BASSET: Scalable Gateway Finder in Large Graphs

Hanghang Tong[1], Spiros Papadimitriou[2], Christos Faloutsos[1],
Philip S. Yu[3], and Tina Eliassi-Rad[4]

[1] Carnegie Mellon University
{htong,christos}@cs.cmu.edu
[2] IBM T.J. Watson
spapadim@us.ibm.com
[3] University of Illinois at Chicago
psyu@cs.uic.edu
[4] Lawrence Livermore National Laboratory
eliassi@llnl.gov

Abstract. Given a social network, who is the best person to introduce you to, say, Chris Ferguson, the poker champion? Or, given a network of people and skills, who is the best person to help you learn about, say, wavelets? The goal is to find a small group of 'gateways': persons who is close enough to us, as well as close enough to the target (person, or skill) or, in other words, are crucial in connecting us to the target.

The main contributions are the following: (a) we show how to formulate this problem precisely; (b) we show that it is sub-modular and thus it can be solved near-optimally; (c) we give fast, scalable algorithms to find such gateways. Experiments on real data sets validate the effectiveness and efficiency of the proposed methods, achieving up to *6,000,000x* speedup.

Keywords: Gateway, Sub-modularity, Scalability, Graph-Mining.

1 Introduction

What is the best gateway between a source node and a target node, in a network? This is a core problem that appears under several guises, with numerous generalizations. Motivating applications include the following:

1. In a corporate social network, which are the key people that bring or hold different groups together? Or, if seeking to establish a cross-division project, who are the best people to lead such an effort?
2. In an immunization setting, given a set of nodes that are infected, and a set of nodes we want to defend, which are the best few 'gateways' we should immunize?
3. Similarly, in a network setting, which are the gateway nodes we should best defend against an attack, to maximize connectivity from source to target.

M.J. Zaki et al. (Eds.): PAKDD 2010, Part II, LNAI 6119, pp. 449–463, 2010.

The problem has several, natural generalizations: (a) we may be interested in the top k best gateways (in case our first few choices are unavailable); (b) we may have more than one source nodes, and more than one target nodes, as in the immunization setting above; (c) we may have a bi-partite graph with relationships (edges) between different node types, as in the last example above. Our main contributions in this paper are:

- A novel 'gateway-ness' score for a given source and target, that agrees with human intuition. Its generalization to the case where we have a group of nodes as the source and the target;
- Two algorithms to find a set of nodes with the highest 'gateway-ness' score, which (1) are fast and scalable; and (2) lead to *near-optimal* results;
- Extensive experimental results on real data sets, showing the effectiveness and efficiency of the proposed methods.

The rest of the paper is organized as follows: We give the problem definitions in Section 2; present 'gateway-ness' scores in Section 3; and deal with the computational issues in Section 4. We evaluate the proposed methods in Section 5. Finally, we review the related work in Section 6 and conclude in Section 7.

Table 1. Symbols

Symbol	Definition and Description
$\mathbf{A}, \mathbf{B}, \ldots$	matrices (bold upper case)
$\mathbf{A}(i,j)$	the element at the i^{th} row and j^{th} column of matrix $\mathbf{A}$
$\mathbf{A}(i,:)$	the i^{th} row of matrix $\mathbf{A}$
$\mathbf{A}(:,j)$	the j^{th} column of matrix $\mathbf{A}$
$\mathbf{A}'$	transpose of matrix $\mathbf{A}$
$\mathbf{a}, \mathbf{b}, \ldots$	column vectors
$\boldsymbol{p}, \boldsymbol{q}, \ldots$	ordered sequences
$\mathcal{S}, \mathcal{T}, \ldots$	sets (calligraphic)
n	number of nodes in the graph
m	number of edges in the graph
$\mathrm{g}(s,t,\mathcal{I})$	the 'Gateway-ness' score for the subset of nodes $\mathcal{I}$ wrt s and t
$\mathrm{g}(\mathcal{S},\mathcal{T},\mathcal{I})$	the 'Gateway-ness' score for the subset of nodes $\mathcal{I}$ wrt $\mathcal{S}$ and $\mathcal{T}$
$\mathbf{r}(s,t)$	the proximity score from s to t
$\mathbf{r}_{\mathcal{I}}(s,t)$	the proximity score from s to t by setting the subset of nodes indexed by $\mathcal{I}$ as sinks

2 Problem Definitions

Table 1 lists the main symbols we use throughout the paper. In this paper, we focus on directed weighted graphs. We represent the graph by its normalized adjacency matrix ($\mathbf{A}$). Following standard notation, we use capital bold letters for matrices (e.g., $\mathbf{A}$), lower-case bold letters for vectors (e.g., $\mathbf{a}$), and calligraphic fonts for sets (e.g., $\mathcal{S}$). We denote the transpose with a prime (i.e., $\mathbf{A}'$

is the transpose of $\mathbf{A}$). We use arrowed lower-case letters for paths on the graph (e.g., $\boldsymbol{p}$), which are ordered sequences. We use parenthesized superscripts to represent source/target information for the corresponding variables. For example $\boldsymbol{p}^{(s,t)} = \{s = u_0, u_1, ..., u_l = t\}$ is a path from the source node s to the target node t. If the source/target information is clear from the context, we omit the superscript for brevity. A sink node i on the graph is a node without out-links (i.e., $\mathbf{A}(:, i) = 0$). We use subscripts to denote the corresponding variable after setting the nodes indexed by the subscripts as sinks. For example, $\boldsymbol{p}_{\mathcal{I}}^{(s,t)}$ is the path from the source node s to the target node t, which does not go through any nodes indexed by the set $\mathcal{I}$ (i.e., $u_i \notin \mathcal{I}, i = 0, ..., l$). With the above notations, our problems can be formally defined as follows:

Problem 1. **(Pair-Gateway)**

Given: A weighted directed graph $\mathbf{A}$, a source node s, a target node t, and a budget (integer) k;

Find: A set of at most k nodes which has the highest 'gate-way-ness' score wrt s and t.

Problem 2. **(Group-Gateway)**

Given: A weighted directed graph $\mathbf{A}$, a group of source nodes $\mathcal{S}$, a group of target nodes $\mathcal{T}$, and a budget (integer) k;

Find: A set of at most k nodes which has the highest 'gate-way-ness' score wrt $\mathcal{S}$ and $\mathcal{T}$.

In both Problem 1 (Pair-Gateway) and Problem 2 (Group-Gateway), there are two sub-problems: (1) how to define the 'gateway-ness' score of a given subset of nodes $\mathcal{I}$; (2) how to find the subset of nodes with the highest 'gateway-ness' score. In the next two sections, we present the solutions for each, respectively.

3 Proposed 'Gateway-ness' Scores

In this section, we present our definitions for 'Gateway-ness'. We first focus on the case of a single source s and a single target t (Pair-Gateway). We then generalize to the case where both the source and the target are a group of nodes (Group-Gateway)

3.1 Node 'Gateway-ness' Score

Given a single source s and a single target t, we want to measure the 'Gateway-ness' score for a given set of nodes $\mathcal{I}$. We first give the formal definitions in such a setting and then provide some intuitions for our definitions.

For a graph $\mathbf{A}$, we can use random walk with restart to measure the proximity (i.e., relevance/closeness) from the source node s to the target node t, which is defined as follows: Consider a random particle that starts from node s. The particle iteratively transits to its neighbors with probability proportional to the corresponding edge weights. Also at each step, the particle returns to node s with some restart probability $(1-c)$. The proximity score from node s to node t is defined as the steady-state probability $\mathbf{r}(s,t)$ that the particle will be on node t [15]. Intuitively, $\mathbf{r}(s,t)$ is the fraction of time that the particle starting from node s will spend on node t of the graph, after an infinite number of steps.

Intuitively, a set of nodes $\mathcal{I}$ are good gateways wrt s and t if they play an important role in the proximity measure from the source to the target. Therefore, our 'Gateway-ness' score can be defined as follows:

$$\mathrm{g}(s,t,\mathcal{I}) \triangleq \varDelta\mathbf{r}(s,t) \triangleq \mathbf{r}(s,t) - \mathbf{r}_{\mathcal{I}}(s,t) \tag{1}$$

where $r_{\mathcal{I}}(s,t)$ is the proximity score from source s to t after setting the subset of nodes indexed by $\mathcal{I}$ as sinks.

3.2 Group 'Gateway-ness' Score

Here we consider the case where the source and/or target consist of more than one nodes. Suppose we have a group of source nodes $\mathcal{S}$ and a group of target nodes $\mathcal{T}$. Then, the 'Gateway-ness' score for a given set of nodes $\mathcal{I}$ can be defined in a similar way:

$$\mathrm{g}(\mathcal{S},\mathcal{T},\mathcal{I}) \triangleq \sum_{s\in\mathcal{S},t\in\mathcal{T}} \varDelta\mathbf{r}(s,t) \triangleq \sum_{s\in\mathcal{S},t\in\mathcal{T}} (\mathbf{r}(s,t) - \mathbf{r}_{\mathcal{I}}(s,t)) \tag{2}$$

where $r_{\mathcal{I}}(s,t)$ is the proximity score from s to t by setting the subset of nodes indexed by $\mathcal{I}$ as sinks (i.e., delete all out-edges, by setting $\mathbf{A}(:,i) = 0$ for all $i \in \mathcal{I}$).

4 BASSET: Proposed Fast Solutions

In this section, we address how to quickly find a subset of nodes of the highest 'Gateway-ness' score. We start by showing that the straight-forward methods (referred to as 'Com-RWR') are computationally intractable. Then, we present the proposed BASSET (BASSET-N for Pair-Gateway and BASSET-G for Group-Gateway). For each case, we first present the algorithm and then analyze its effectiveness as well as its computational complexity.

4.1 Computational Challenges

Here, we present the computational challenges and the way we tackle them. For the sake of succinctness, we mainly focus on BASSET-N.

There are two main computational challenges in order to find a subset of nodes with the highest 'Gateway-ness' score. First of all, we need to compute the proximity from the source to the target on different graphs, each of which is a perturbed version of the original graph. This essentially means that we cannot directly apply some powerful pre-computational method to evaluate the proximity from the source to the target (after setting the subset of nodes indexed by $\mathcal{I}$ as sinks). Instead, we have to rely on on-line iterative methods, whose computational complexity is $O(m)$. The challenges are compounded by the need to evaluate $\mathrm{g}(s,t,\mathcal{I})$ (eq. (1)) or $\mathrm{g}(\mathcal{S},\mathcal{T},\mathcal{I})$(eq. (2)) an exponential number of times ($\binom{n}{k}$). Putting these together, the straightforward way to find k nodes with the highest 'Gateway-ness' score is $O(\binom{n}{k}m)$. This is computationally intractable. Suppose on a graph with $1,000,000$ nodes, we want to find the best $k = 5$ gateway nodes. If computing each proximity score takes 0.001 seconds, then 2.64×10^{17} years are needed to find the gateways. This is much longer than the age of the universe.[1]

To tackle such challenges, we resort to two main ideas, which are summarized in Theorem 1. According to Theorem 1, in order to evaluate the 'Gateway-ness' score of a given set of nodes, we do not need to actually set these nodes as sinks and compute the proximity score on the new graph. Instead, we can compute it from the original graph. In this way, we can utilize methods based on pre-computation to accelerate the process. Furthermore, since $\mathrm{g}(s,t,\mathcal{I})$ and $\mathrm{g}(\mathcal{S},\mathcal{T},\mathcal{I})$ are sub-modular wrt $\mathcal{I}$, we can develop some greedy algorithm to avoid exponential enumeration, and still get some *near-optimal* solution. In Theorem 1, $\mathbf{A}$ is the normalized adjacency matrix of the graph. It is worth pointing out that The proposed methods (BASSET-N and BASSET-G) we will introduce are orthogonal to the specific way of normalization. For simplicity, we use column-normalization throughout this paper. Also, $\mathbf{Q}(\mathcal{I},\mathcal{I})$ is a $|\mathcal{I}| \times |\mathcal{I}|$ matrix, containing the elements in the matrix $\mathbf{Q}$ which are at the rows/columns indexed by $\mathcal{I}$. Similarly, $\mathbf{Q}(t,\mathcal{I})$ is a row vector with length $|\mathcal{I}|$, containing the elements in the matrix $\mathbf{Q}$ which are at the t^{th} row and the columns indexed by $\mathcal{I}$. $\mathbf{Q}(\mathcal{I},s)$ is a column vector with length $|\mathcal{I}|$, containing the elements in the matrix $\mathbf{Q}$ which are at the s^{th} column and the rows indexed by $\mathcal{I}$.

Theorem 1. Core Theorem. *Let $\mathbf{A}$ be the normalized adjacency matrix of the graph, and $\mathbf{Q} = (1-c)(\mathbf{I}-c\mathbf{A})^{-1}$. For a given source s and target t, the 'Gateway-ness' score of a subset of nodes $\mathcal{I}$ defined in eq.* (1) *satisfies the properties P1 and P2. For a given source group $\mathcal{S}$ and target group $\mathcal{T}$, the 'Gateway-ness' score of a subset of nodes $\mathcal{I}$ defined in eq.* (2) *satisfies the properties P3 and P4, where $s \neq t$, $s,t \notin \mathcal{I}$, $\mathcal{S} \bigcap \mathcal{T} = \emptyset$, $\mathcal{S} \bigcap \mathcal{I} = \emptyset$, and $\mathcal{T} \bigcap \mathcal{I} = \emptyset$.*

P1. $g(s,t,\mathcal{I}) = \mathbf{Q}(t,\mathcal{I})\mathbf{Q}(\mathcal{I},\mathcal{I})^{-1}\mathbf{Q}(\mathcal{I},s)$;
P2. $g(s,t,\mathcal{I})$ *is sub-modular wrt the set* $\mathcal{I}$.
P3. $g(\mathcal{S},\mathcal{T},\mathcal{I}) = \sum_{s\in\mathcal{S},t\in\mathcal{T}} \mathbf{Q}(t,\mathcal{I})\mathbf{Q}(\mathcal{I},\mathcal{I})^{-1}\mathbf{Q}(\mathcal{I},s)$;
P4. $g(\mathcal{S},\mathcal{T},\mathcal{I})$ *is sub-modular wrt the set* $\mathcal{I}$.

Proof of P1: WLOG, we assume that $\mathcal{I} = \{n-k+1, ...n\}$. Let $\mathbf{A}$ and $\tilde{\mathbf{A}}$ be the normalized adjacency matrices of the graph before/after we set the subset of nodes in $\mathcal{I}$ as sinks. Write $\mathbf{A}$ and $\tilde{\mathbf{A}}$ in block form:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{pmatrix}, \ \tilde{\mathbf{A}} = \begin{pmatrix} \tilde{\mathbf{A}}_{1,1} & \tilde{\mathbf{A}}_{1,2} \\ \tilde{\mathbf{A}}_{2,1} & \tilde{\mathbf{A}}_{2,2} \end{pmatrix} = \begin{pmatrix} \mathbf{A}_{1,1} & \mathbf{0} \\ \mathbf{A}_{2,1} & \mathbf{0} \end{pmatrix} \tag{3}$$

[1] According to Wikipedia, (http://en.wikipedia.org/wiki/Age_of_the_universe), the age of the universe is about 1.4×10^{10} year.

where $\mathbf{0}$ is a matrix with all zero elements.

Let $\tilde{\mathbf{Q}} = (1-c)(\mathbf{I} - c\tilde{\mathbf{A}})^{-1}$. We can also write $\tilde{\mathbf{Q}}$ and $\mathbf{Q}$ in block form:

$$\mathbf{Q} = (1-c)(\mathbf{I} - c\mathbf{A})^{-1} = \begin{pmatrix} \mathbf{Q}_{1,1} & \mathbf{Q}_{1,2} \\ \mathbf{Q}_{2,1} & \mathbf{Q}_{2,2} \end{pmatrix} = (1-c)\begin{pmatrix} \mathbf{I} - c\mathbf{A}_{1,1} & -c\mathbf{A}_{1,2} \\ -c\mathbf{A}_{2,1} & \mathbf{I} - c\mathbf{A}_{2,2} \end{pmatrix}^{-1}$$
$$\tilde{\mathbf{Q}} = \begin{pmatrix} \tilde{\mathbf{Q}}_{1,1} & \tilde{\mathbf{Q}}_{1,2} \\ \tilde{\mathbf{Q}}_{2,1} & \tilde{\mathbf{Q}}_{2,2} \end{pmatrix} = (1-c)\begin{pmatrix} \mathbf{I} - c\mathbf{A}_{1,1} & \mathbf{0} \\ -c\mathbf{A}_{2,1} & \mathbf{I} \end{pmatrix}^{-1}$$

Applying the block matrix inverse lemma [12] to $\tilde{\mathbf{Q}}$ and $\mathbf{Q}$, we get the following equations:

$$\begin{aligned}
\tilde{\mathbf{Q}}_{1,1} &= (1-c)(\mathbf{I} - c\mathbf{A}_{1,1})^{-1}, \quad \tilde{\mathbf{Q}}_{1,2} = \mathbf{0} \\
\tilde{\mathbf{Q}}_{2,1} &= c(1-c)\mathbf{A}_{2,1}(\mathbf{I} - c\mathbf{A}_{1,1})^{-1}, \quad \tilde{\mathbf{Q}}_{2,2} = (1-c)\mathbf{I} \\
\mathbf{Q}_{1,1} &= (1-c)(\mathbf{I} - c\mathbf{A}_{1,1})^{-1} + c^2(\mathbf{I} - c\mathbf{A}_{1,1})^{-1}\mathbf{A}_{1,2}\mathbf{Q}_{2,2}\mathbf{A}_{2,1}(\mathbf{I} - c\mathbf{A}_{1,1})^{-1} \\
\mathbf{Q}_{1,2} &= c(\mathbf{I} - c\mathbf{A}_{1,1})^{-1}\mathbf{A}_{1,2}\mathbf{Q}_{2,2} \\
\mathbf{Q}_{2,1} &= c\mathbf{Q}_{2,2}\mathbf{A}_{2,1}(\mathbf{I} - c\mathbf{A}_{1,1})^{-1}
\end{aligned} \tag{4}$$

Therefore, we have

$$\tilde{\mathbf{Q}}_{1,1} = \mathbf{Q}_{1,1} - \mathbf{Q}_{1,2}\mathbf{Q}_{2,2}^{-1}\mathbf{Q}_{2,1} \tag{5}$$

On the other hand, based on the properties of random walk with restart [15], we have $\mathbf{r}(i,j) = \mathbf{Q}(j,i)$, and $\mathbf{r}_{\mathcal{I}}(i,j) = \tilde{\mathbf{Q}}(j,i)$,$(i,j = 1,...,n)$. Together with eq. (5), we have

$$\mathrm{g}(s,t,\mathcal{I}) = \mathbf{r}(s,t) - \mathbf{r}_{\mathcal{I}}(s,t) = \mathbf{Q}_{1,1}(t,s) - \tilde{\mathbf{Q}}_{1,1}(t,s) = \mathbf{Q}_{1,2}(t,:)\mathbf{Q}_{2,2}^{-1}\mathbf{Q}_{1,2}(:,s) \tag{6}$$

which completes the proofs of P1. □

Proof of P3: Since P1 holds, we have

$$\mathrm{g}(\mathcal{S},\mathcal{T},\mathcal{I}) = \sum_{s\in\mathcal{S},t\in\mathcal{T}} \Delta r(s,t) = \sum_{s\in\mathcal{S},t\in\mathcal{T}} \mathrm{g}(s,t,\mathcal{I}) = \sum_{s\in\mathcal{S},t\in\mathcal{T}} \mathbf{Q}(t,\mathcal{I})\mathbf{Q}(\mathcal{I},\mathcal{I})^{-1}\mathbf{Q}(\mathcal{I},s) \tag{7}$$

which completes the proofs of P3. □

Proof of P2: Let $\mathcal{I},\mathcal{J},\mathcal{K}$ be three subsets and $\mathcal{I} \subseteq \mathcal{J}$. We will first prove by induction that, for any integer power j, the following inequality holds element-wise.

$$\mathbf{A}_{\mathcal{I}}^{j} - \mathbf{A}_{\mathcal{I}\bigcup\mathcal{K}}^{j} \geq \mathbf{A}_{\mathcal{J}}^{j} - \mathbf{A}_{\mathcal{J}\bigcup\mathcal{K}}^{j} \tag{8}$$

It is easy to verify the base case (i.e.,$j = 1$) for eq. (8) holds. Next, assume that eq. (8) holds for $j = 1,...,j_0$, and we want to prove that it also holds for the case $j = j_0 + 1$:

$$\begin{aligned}
\mathbf{A}_{\mathcal{I}}^{j_0+1} - \mathbf{A}_{\mathcal{I}\bigcup\mathcal{K}}^{j_0+1} &= \mathbf{A}_{\mathcal{I}}^{j_0+1} - \mathbf{A}_{\mathcal{I}\bigcup\mathcal{K}}^{j_0}\mathbf{A}_{\mathcal{I}} + \mathbf{A}_{\mathcal{I}\bigcup\mathcal{K}}^{j_0}\mathbf{A}_{\mathcal{I}} - \mathbf{A}_{\mathcal{I}\bigcup\mathcal{K}}^{j_0+1} \\
&= (\mathbf{A}_{\mathcal{I}}^{j_0} - \mathbf{A}_{\mathcal{I}\bigcup\mathcal{K}}^{j_0})\mathbf{A}_{\mathcal{I}} + \mathbf{A}_{\mathcal{I}\bigcup\mathcal{K}}^{j_0}(\mathbf{A}_{\mathcal{I}} - \mathbf{A}_{\mathcal{I}\bigcup\mathcal{K}}) \\
&\geq (\mathbf{A}_{\mathcal{J}}^{j_0} - \mathbf{A}_{\mathcal{J}\bigcup\mathcal{K}}^{j_0})\mathbf{A}_{\mathcal{I}} + \mathbf{A}_{\mathcal{I}\bigcup\mathcal{K}}^{j_0}(\mathbf{A}_{\mathcal{J}} - \mathbf{A}_{\mathcal{J}\bigcup\mathcal{K}}) \\
&\geq (\mathbf{A}_{\mathcal{J}}^{j_0} - \mathbf{A}_{\mathcal{J}\bigcup\mathcal{K}}^{j_0})\mathbf{A}_{\mathcal{J}} + \mathbf{A}_{\mathcal{J}\bigcup\mathcal{K}}^{j_0}(\mathbf{A}_{\mathcal{J}} - \mathbf{A}_{\mathcal{J}\bigcup\mathcal{K}}) = \mathbf{A}_{\mathcal{J}}^{j_0+1} - \mathbf{A}_{\mathcal{J}\bigcup\mathcal{K}}^{j_0+1}
\end{aligned} \tag{9}$$

In eq. (9), the first inequality holds because of the induction assumption. The second inequality holds because $\mathbf{A}_{\mathcal{I}} \geq \mathbf{A}_{\mathcal{J}} \geq 0$ holds element-wise, and $\mathbf{A}_{\mathcal{I} \bigcup \mathcal{K}} \geq \mathbf{A}_{\mathcal{J} \bigcup \mathcal{K}} \geq 0$ holds element-wise.

Since $\tilde{\mathbf{Q}} = (1-c)(\mathbf{I} - c\tilde{\mathbf{A}})^{-1} = (1-c)\sum_{j=0}^{\infty}(c\tilde{\mathbf{A}})^j$, we have

$$\begin{aligned}
\mathrm{g}(s,t,\mathcal{I}\cup\mathcal{K}) - \mathrm{g}(s,t,\mathcal{I}) &= (1-c)\sum_{j=0}^{\infty}((c\mathbf{A}_{\mathcal{I}})^j(t,s) - (c\mathbf{A}_{\mathcal{I}\bigcup\mathcal{K}})^j(t,s)) \\
&\geq (1-c)\sum_{j=0}^{\infty}((c\mathbf{A}_{\mathcal{J}})^j(t,s) - (c\mathbf{A}_{\mathcal{J}\bigcup\mathcal{K}})^j(t,s)) \\
&= \mathrm{g}(s,t,\mathcal{J}\cup\mathcal{K}) - \mathrm{g}(s,t,\mathcal{J}) \qquad (10)
\end{aligned}$$

Therefore, $\mathrm{g}(s,t,\mathcal{I})$ is sub-modular, which completes the proof of P2. □

Proof of P4: Since $\mathrm{g}(\mathcal{S},\mathcal{T},\mathcal{I}) = \sum_{s\in\mathcal{S},t\in\mathcal{T}}\mathrm{g}(s,t,\mathcal{I})$ (In other words, $\mathrm{g}(\mathcal{S},\mathcal{T},\mathcal{I})$ is a non-negative linear combination of sub-modular functions) , according to the linearity of sub-modular functions [8], we have that $\mathrm{g}(\mathcal{S},\mathcal{T},\mathcal{I})$ is also sub-modular, which completes the proof of P4. □

4.2 BASSET-N for Problem 1

BASSET-N Algorithm. Our fast solution for Problem 1 is summarized in Alg. 1. In Alg. 1, after initialization (step 1), we first pick a node i_0 with the highest $\frac{\mathbf{r}(s,i)\mathbf{r}(i,t)}{\mathbf{r}(i,i)}$ (step 3). Then, in steps 4-14, we find the rest of the nodes in a greedy way. That is, in each outer loop, we try to find one more node while keeping the current $\mathcal{I}$ unchanged. According to P1 of theorem 1, $\mathbf{v}(i)$ computed in step 7 is the gateway score for the subset $\mathcal{J}$.[2] If the current subset of nodes $\mathcal{I}$ can completely disconnect the source and the target (by setting them as sinks), we will stop the algorithm (step 12). Therefore, Alg. 1 always returns no more than k nodes. It is worth pointing out that in Alg. 1, all the proximity scores are computed from the original graph $\mathbf{A}$. Therefore, we can utilize some powerful methods based on pre-computation to accelerate the whole process. To name a few, for a medium size graph $\mathbf{A}$ (e.g., a few thousands of nodes), we can pre-compute and store the matrix $\mathbf{Q} = (1-c)(\mathbf{I} - c\mathbf{A})^{-1}$; for large unipartite graphs and bipartite graphs, we can use the NB_LIN and BB_LIN algorithms, respectively [15].

Analysis of BASSET-N. In this subsection, we analyze the effectiveness and the efficiency of Alg. 1. First, the effectiveness of the proposed BASSET-N is guaranteed by the following lemma. According to Lemma 1, although BASSET-N is a greedy algorithm, the results it outputs are *near-optimal.*

Lemma 1. Effectiveness of BASSET-N. *Let $\mathcal{I}$ be the subset of nodes selected by Alg. 1 and $|\mathcal{I}| = k_0$. Then, $g(s,t,\mathcal{I}) \geq (1-1/e)max_{|\mathcal{J}|=k_0}\ g(s,t,\mathcal{J})$, where $g(s,t,\mathcal{I})$, and $g(s,t,\mathcal{J})$ are defined by eq.* (1)*.*

Proof: It is easy to verify that the node i_0 selected in step 10 of Alg. 1 satisfies $i_0 = \mathrm{argmax}_{j\notin\mathcal{I},j\neq s,j\neq t}\mathrm{g}(s,t,\mathcal{I}\bigcup j)$. Also, we have $\mathrm{g}(s,t,\phi) = 0$, where ϕ is an

[2] This is because in random walk with restart, we have $\mathbf{r}(i,j) = \mathbf{Q}(j,i)$ for any i,j [15].

Algorithm 1. BASSET-N

```
Input: the normalized adjacency matrix $\mathbf{A}$, the source node $s$, the target node $t$, the
    budget $k$ and the parameter $c$
Output: a set of nodes $\mathcal{I}$, where $|\mathcal{I}| \leq k$.
 1: initialize $\mathcal{I}$ to be empty.
 2: compute the proximity score $\mathbf{r}(s,t)$ from the source node $s$ to the target node $t$.
 3: find $i_0 = \text{argmax}_i \frac{\mathbf{r}(s,i)\mathbf{r}(i,t)}{\mathbf{r}(i,i)}$, where $i = 1, ..., n$ and $i \neq s, i \neq t$. add $i_0$ to $\mathcal{I}$.
 4: for $j = 2$ to $k$ do
 5:   for $i = 1$ to $n$, and $i \neq s, i \neq t$ and $i \notin \mathcal{I}$ do
 6:     let $\mathcal{J} = \mathcal{I} \cup i$.
 7:     compute $\mathbf{v}(i) = \mathbf{r}(\mathcal{J},t)'\mathbf{r}(\mathcal{J},\mathcal{J})^{-1}\mathbf{r}(s,\mathcal{J})'$
 8:   end for
 9:   if $\max_i \mathbf{v}(i) \leq \mathbf{r}(s,t)$ then
10:     find $i_0 = \text{argmax}_i \mathbf{v}(i)$; add $i_0$ to $\mathcal{I}$.
11:   else
12:     break;
13:   end if
14: end for
15: return $\mathcal{I}$
```

empty set. On the other hand, according to Theorem 1, $\text{g}(s,t,\mathcal{I})$ is sub-modular wrt the subset $\mathcal{I}$. Therefore, we have $\text{g}(s,t,\mathcal{I}) \geq (1-1/e)\max_{|\mathcal{J}|=k_0} \text{g}(s,t,\mathcal{J})$, which completes the proof. □

Next, we analyze the efficiency of BASSET-N, which is given in Lemma 2[3]. We can draw the following two conclusions, according to Lemma 2: (1) the proposed BASSET-N achieves a significant speedup over the straight-forward method ($O(n \cdot k^4)$ vs. $O(\binom{n}{k}m)$). For example, in the graph with 100 nodes and 1,000 edges, in order to find the gateway with $k = 5$ nodes, BASSET-N is more than *6 orders of magnitude* faster, and the speedup quickly increases wrt the size of the graph; (2) the proposed BASSET-N is applicable to large graphs since it is linear wrt the number of the nodes.

Lemma 2. Efficiency of BASSET-N. *The computational complexity of Alg. 1 is upper bounded by $O(n \cdot k^4)$.*

Proof: The cost for steps 1-2 is constant. The cost for step 3 is $O(n)$. At each inner loop (steps 6-7), the cost is $O(nj^3 + nj^2)$. The cost for steps 9-13 is $O(n)$. The outer loop has no more than $k-1$ iterations. Putting these together, the computational cost for BASSET-N is:

$$\begin{aligned}\text{Cost(BASSET-N)} &\leq n + \sum_{j=1}^{k}(nj^3 + nj^2 + n) \\ &= n + nk + n\frac{k(k+1)(2k+1)}{6} + n\frac{k^2(k+1)^2}{4} = O(nk^4) \quad (11)\end{aligned}$$

which completes the proof. □

[3] Here, we assume that the cost to get one proximity score is constant, which can be achieved with pre-computation methods [15].

4.3 BASSET-G for Problem 2

BASSET-G Algorithm. Our fast solution for Problem 2 is summarized in Alg. 2. It works in a similar way as Alg. 1: after initialization (step 1), we first pick a node i_0 with the highest $\sum_{s\in\mathcal{S},\ t\in\mathcal{T}} \frac{\mathbf{r}(s,i)\mathbf{r}(i,t)}{\mathbf{r}(i,i)}$ (step 3). Then, in steps 4-14, we find the rest of the nodes in a greedy way. That is, in each outer-loop, we try to find one more node while keeping the current $\mathcal{I}$ unchanged. If the current subset of the nodes $\mathcal{I}$ can completely disconnect the source group and the target group (by setting them as sinks), we will stop the algorithm (step 10). As in Alg. 1, all the proximity scores are computed from the original graph $\mathbf{A}$. Therefore, we can again utilize those powerful pre-computation based methods to accelerate the whole process.

Algorithm 2. BASSET-G

```
Input: the normalized adjacency matrix $\mathbf{A}$, the source group $\mathcal{S}$, the target group $\mathcal{T}$,
    the budget $k$ and the parameter $c$
Output: a set of nodes $\mathcal{I}$, where $|\mathcal{I}| \leq k$.
 1: initialize $\mathcal{I}$ to be empty.
 2: compute the proximity score $\sum_{s\in\mathcal{S},\ t\in\mathcal{T}} \mathbf{r}(s,t)$ from the source group $\mathcal{S}$ to the target
    group $\mathcal{T}$.
 3: find $i_0 = \text{argmax}_i \sum_{s\in\mathcal{S},\ t\in\mathcal{T}} \frac{\mathbf{r}(s,i)\mathbf{r}(i,t)}{\mathbf{r}(i,i)}$, where $i = 1, ..., n$ and $i \neq s, i \neq t$; add $i_0$
    to $\mathcal{I}$.
 4: for $j = 2$ to $k$ do
 5:   for $i = 1$ to $n$, and $i \neq s, i \neq t$ and $i \notin \mathcal{I}$ do
 6:     let $\mathcal{J} = \mathcal{I} \cup i$.
 7:     compute $\mathbf{v}(i$ as $\mathbf{v}(i) = \sum_{s\in\mathcal{S},\ t\in\mathcal{T}} \mathbf{r}(\mathcal{J},t)'\mathbf{r}(\mathcal{J},\mathcal{J})^{-1}\mathbf{r}(s,\mathcal{J})'$
 8:   end for
 9:   if $\max_i \mathbf{v}(i) \leq \sum_{s\in\mathcal{S},\ t\in\mathcal{T}} \mathbf{r}(s,t)$ then
10:     find $i_0 = \text{argmax}_i \mathbf{v}(i)$; add $i_0$ to $\mathcal{I}$.
11:   else
12:     break;
13:   end if
14: end for
15: return $\mathcal{I}$
```

Analysis of BASSET-G. The effectiveness and efficiency of the proposed BASSET-G are given in Lemma 3 and Lemma 4, respectively. Similar as BASSET-N, the proposed BASSET-G is (1) *near-optimal*; and (2) fast and scalable for large graphs.

Lemma 3. Effectiveness of BASSET-G. *Let $\mathcal{I}$ be the subset of nodes selected by Alg. 2 and $|\mathcal{I}| = k_0$. Then, $g(\mathcal{S},\mathcal{T},\mathcal{I}) \geq (1 - 1/e)max_{|\mathcal{J}|=k_0}\ g(\mathcal{S},\mathcal{T},\mathcal{J})$, where $g(\mathcal{S},\mathcal{T},\mathcal{I})$, and $g(\mathcal{S},\mathcal{T},\mathcal{J})$ are defined by eq.* (2).

Proof: Similar as for Lemma 1. Omitted for brevity. □

Lemma 4. Efficiency of BASSET-G. *The computational complexity of Alg. 2 is upper bounded by $O(n \cdot (max(k, |\mathcal{S}|, |\mathcal{T}|))^4)$.*

Proof: Similar as for Lemma 2. Omitted for brevity. □

5 Experimental Evaluations

In this section we present experimental results. All the experiments are designed to answer the following questions:

- *Effectiveness:* how effective are the 'Gateway-ness' scores in real graphs?
- *Efficiency:* how fast and scalable are BASSET-N and BASSET-G?

5.1 Data Sets

We used three real data sets, which are summarized in table 2.

Table 2. Summary of the data sets

Name	n	m	**Weight**
PolBooks	105	882	No
AA	418,236	2,753,798	Yes
NetFlix	2,667,199	56,919,190	No

The first data set (*PolBooks*) is a co-purchasing book network.[4] Each node is a political book and there is an edge between two books if purchased by the same person. Overall, we have $n = 105$ nodes and $m = 882$ edges.

The second data set (AA) is a co-authorship network, where each node is an author and the edge weight is the number of the co-authored papers between the two corresponding persons. Overall, we have $n = 418,236$ nodes and $m = 2,753,798$ edges.

The last data set (*NetFlix*) is from the Netflix prize[5]. Rows represent users and columns represent movies. If a user has given a particular movie positive ratings (4 or 5), we connect them with an edge. In total, we have 2,667,199 nodes (2,649,429 users and 17,770 movies), and 56,919,190 edges.

5.2 Effectiveness

Here, we evaluate the effectiveness of the proposed 'Gateway-ness' scores. We first compare with several candidate methods in terms of separating the source from the target. And then, we present various case studies.

Quantitative Comparisons. The basic idea of the proposed 'Gateway-ness' scores is to find a subset of nodes which collectively play an important role in measuring the proximity from the source node (or source group) to the target node (or target group). Here, we want to validate this basic assumption. We compare it with the following alternative choices: (a) selecting k nodes with the highest center-piece AND score (CePS-AND) [13]; (b) selecting k nodes with the highest center-piece OR score (CePS-OR) [13]; (c) randomly selecting k

[4] `http://www.orgnet.com/`

[5] `http://www.netflixprize.com/`

Table 3. Effectiveness comparison between BASSET-N and alternatives. x-axis is the budget k and y-axis is the normalized decay of proximity. Higher is better. The proposed BASSET-N is the best.

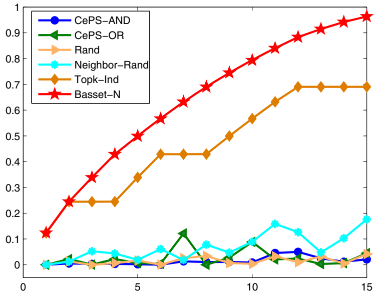


Table 4. BASSET-N on *PolBooks* Graph. ('c' for 'conservative', 'l' for 'liberal', and 'n' for 'neutral')

Node	Book Title	Label
10	Bush country	c
13	Off with their heads	c
103	Back up such up	l
5	Sleeping with the devil	n
8	Ghost wars	n
77	Plan of attack	n
78	Bush at war	c
59	Rise of the vulcanes	c
52	Allies	c
42	The Bushes	c

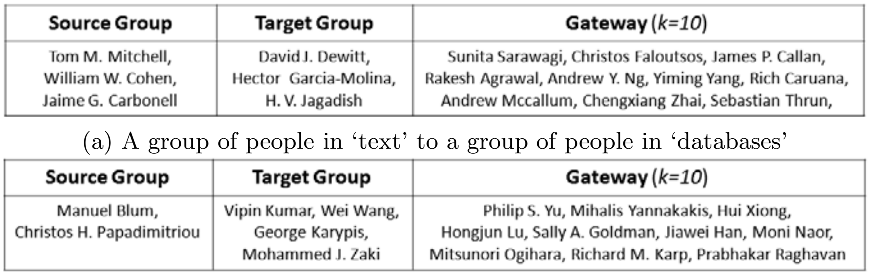
nodes (Rand); (d) randomly selecting k nodes from the neighboring nodes of the source node and the target node (Neighbor-Rand); (e) selecting k nodes with the highest $\frac{\mathbf{r}(s,i)\mathbf{r}(i,t)}{\mathbf{r}(i,i)}$(Topk-Ind). We randomly select a source node s and a target node t,[6] and then use the different methods to select a subset $\mathcal{I}$ with k nodes. Figure 2 presents the comparison results, where the x-axis is the number of nodes selected (k), and the y-axis is the normalized decay in terms of the proximity score from the source node s to the target node t ($\frac{\mathbf{r}(s,t)-\mathbf{r}_{\mathcal{I}}(s,t)}{\mathbf{r}(s,t)}$). The resulting curves are averaged over 1,000 randomly chosen source-target pairs. From table 3, we can see that (1) the proposed BASSET-N performs best in terms of separating the source from the target; (2)Topk-Ind, where we simply select k nodes with highest $\frac{\mathbf{r}(s,i)\mathbf{r}(i,t)}{\mathbf{r}(i,i)}$, does not perform as well as BASSET-N, where we want to find a subset of k nodes which *collectively* has the highest score $\mathbf{r}(\mathcal{I},t)'\mathbf{r}(\mathcal{I},\mathcal{I})^{-1}\mathbf{r}(s,\mathcal{I})'$.

Case Studies. Next, we will show some case studies, to demonstrate the effectiveness of BASSET-N and BASSET-G.

PolBooks. For this data set, the nodes are political books and the existence of the edge indicates the co-purchasing (by the same person) of the two books. Each book is annotated by one of the following three labels: 'liberal', 'conservative' and 'neutral'. We pick a 'liberal' book ('The Price of Loyalty') as the source node, and a 'conservative' book ('Losing Bin Laden') as the target node. Then, we ran the proposed BASSET-N to find the gateway with 10 nodes. The result is presented in table 4. The result is again consistent with human intuition, - the resulting gateway books are either popular books in one of the two communities ('conservative' vs. 'liberal') such as, 'Bush country' from 'conservative', 'Back up suck up' from 'liberal', etc; or those 'neutral' books which are likely to be purchased by readers from both communities (e.g., 'Sleeping with the devil', etc).

[6] The result when source and target are a group of nodes is similar, and omitted for brevity.

Table 5. BASSET-G on AA Network



Source Group	Target Group	Gateway (*k=10*)
Tom M. Mitchell, William W. Cohen, Jaime G. Carbonell	David J. Dewitt, Hector Garcia-Molina, H. V. Jagadish	Sunita Sarawagi, Christos Faloutsos, James P. Callan, Rakesh Agrawal, Andrew Y. Ng, Yiming Yang, Rich Caruana, Andrew Mccallum, Chengxiang Zhai, Sebastian Thrun,

(a) A group of people in 'text' to a group of people in 'databases'

Source Group	Target Group	Gateway (*k=10*)
Manuel Blum, Christos H. Papadimitriou	Vipin Kumar, Wei Wang, George Karypis, Mohammed J. Zaki	Philip S. Yu, Mihalis Yannakakis, Hui Xiong, Hongjun Lu, Sally A. Goldman, Jiawei Han, Moni Naor, Mitsunori Ogihara, Richard M. Karp, Prabhakar Raghavan

(b) A group of people in 'theory ' to a group of people in 'bioinfomatics'

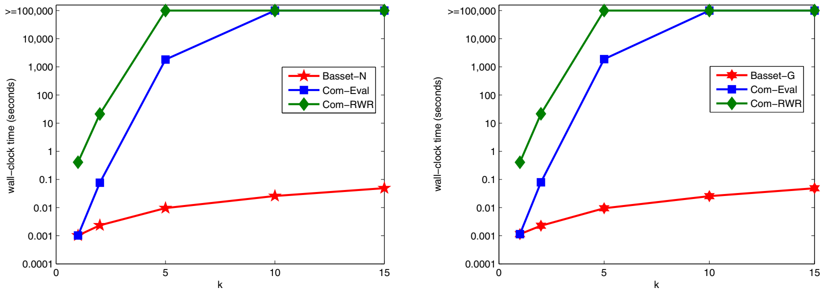
AA. We use this data set to perform case studies for the proposed BASSET-G. We choose (1) a group of people from a certain field (e.g., 'text', 'theory', etc) as the source group $\mathcal{S}$; and (2) another group of people in some other field (e.g., 'databases', 'bioinfomatics', etc) as the target group $\mathcal{T}$. Then, we ran the proposed BASSET-N to find the gateway with $k = 10$ nodes. Table 5 lists some results. They are all consistent with human intuition, - the resulting authors are either productive authors in one of the two fields, or multi-disciplinary, who have close collaborations to both the source and the target groups of authors.

5.3 Efficiency

We will study the wall-clock running time of the proposed BASSET-N and BASSET-G here. Basically, we want to answer the following two questions:

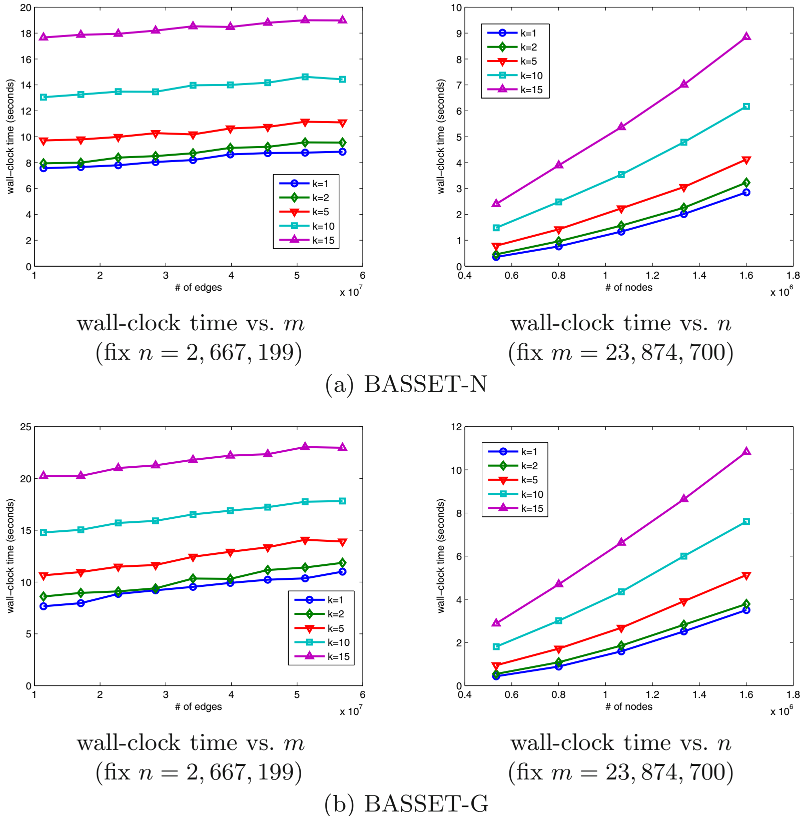
1. *(Speed)* What is the speedup of the proposed BASSET-N and BASSET-G over the straightforward methods?
2. *(Scalability)* How do BASSET-N and BASSET-G scale with the size of the graph (n and m)?

First, we compare BASSET-N and BASSET-G with two straightforward methods: (1) 'Com-RWR', where we use combinatorial enumeration to find the gateway and, for each enumeration, we compute the proximity from the *new* graph; and (2) 'Com-Eval', where we use combinatorial enumeration to find the gateway, and for each enumeration, we compute the proximity from the *original* graph. Figure 1 shows the comparison on *PolBooks* graph. We can draw the following conclusions. (1) Straightforward methods ('Com-RWR' and 'Com-Eval') are computationally intractable even for a small graph. For example, it takes more than $1,000$ seconds and $100,000$ seconds to find the $k = 5$ gateway by 'Com-Eval' and by 'Com-RWR', respectively. (2) The speedup of the proposed BASSET-N and BASSET-G over both 'Com-Eval' and 'Com-RWR' is significant - in most cases, we achieve *several (up to 6) orders of magnitude* speedups. (3) The speedup of the proposed BASSET-N and BASSET-G over both 'Com-RWR' and 'Com-Eval' quickly increases wrt the size of the gateway k. Note that we stop running the program if it takes more than 100,000 seconds (i.e., longer than a day).



(a) BASSET-N (b) BASSET-G

Fig. 1. Comparison of speed on *PolBooks* graph. Wall-clock time vs. k. Lower is better. Time is in logarithm scale. The proposed BASSET-N and BASSET-G (red star) are significantly faster.



wall-clock time vs. m
(fix $n = 2,667,199$)

wall-clock time vs. n
(fix $m = 23,874,700$)

(a) BASSET-N

wall-clock time vs. m
(fix $n = 2,667,199$)

wall-clock time vs. n
(fix $m = 23,874,700$)

(b) BASSET-G

Fig. 2. Scalability of BASSET. Wall-clock time vs. the size of the graph. Lower is better. $|\mathcal{S}| = |\mathcal{T}| = 5$.

Next, we evaluate the scalability of the proposed BASSET-N and BASSET-G wrt the size of the graph, using the largest data set (*NetFlix*). From figure 2, we can make the following conclusions: (1) if we fix the number of nodes (n) in the graph, the wall-clock time of both BASSET-N and BASSET-G is almost *constant* wrt the number of edges (m); and (2) if we fix the number of edges (m) in the graph, the wall-clock time of both BASSET-N and BASSET-G is *linear* wrt the number of nodes (n). Therefore, they are suitable for large graphs.

6 Related Work

In this section, we review the related work, which can be categorized into three parts:

Betweenness centrality. The proposed 'Gateway-ness' scores relate to measures of betweenness centrality, both those based on the shortest path [4], as well as those based on random walk [10]. When the gateway set size is $k = 1$, the proposed 'Gateway-ness' scores can be viewed as *query-specific* betweenness centrality measures. Moreover, in the proposed BASSET-N and BASSET-G, we aim to find a subset of nodes *collectively*, wherein traditional betweenness centrality, we usually calculate the score for each node *independently* (and then might pick k nodes with the highest individual scores).

Connection subgraphs. In the proposed BASSET-N, the idea of finding a subset of nodes wrt the source/target is also related to the concept of connection subgraphs, such as [3,7,13]. However, in connection subgraphs, we aim to find a subset of nodes which have *strong* connections among themselves for the purpose of visualization. While in the proposed BASSET-N, we implicity encourage the resulting subset of nodes to be disconnected with each other so that they are able to *collectively disconnect* the target node from the source node to the largest extent (if we set them as sinks).

Graph proximity. The basic idea of the proposed BASSET-N and BASSET-G is to find a subset of nodes which will bring the largest decrease of the proximity score from the source node (or the source group) to the target node (or the target group). Graph proximity itself is an important building block in many graph mining settings. Representative work includes the BANKS system [1], link prediction [9], content-based image retrieval [6], cross-modal correlation discovery [11], pattern matching [14], ObjectRank [2], RelationalRank [5], etc.

7 Conclusion

In this paper, we study how to find good 'gateway' nodes in a graph, given one or more source and target nodes. Our main contributions are: (a) we formulate the problem precisely; (b) we develop BASSET-N and BASSET-G, two fast (up to *6,000,000x* speedup) and scalable (*linear* wrt the number of the nodes in the graph) algorithms to solve it in a provably near-optimal fashion, using sub-modularity. We applied the proposed BASSET-N and BASSET-G on real data sets to validate the effectiveness and efficiency.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grants No. DBI-0640543 IIS-0705359 CNS-0721736 IIS0808661 iCAST and was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. DE-AC52-07NA27344. This work is also partially supported by an IBM Faculty Award, a SPRINT gift, with additional funding from Intel, and Hewlett-Packard. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or other funding parties.

References

1. Aditya, B., Bhalotia, G., Chakrabarti, S., Hulgeri, A., Nakhe, C., Parag, S.S.: Banks: Browsing and keyword searching in relational databases. In: VLDB, pp. 1083–1086 (2002)
2. Balmin, A., Hristidis, V., Papakonstantinou, Y.: Objectrank: Authority-based keyword search in databases. In: VLDB, pp. 564–575 (2004)
3. Faloutsos, C., McCurley, K.S., Tomkins, A.: Fast discovery of connection subgraphs. In: KDD, pp. 118–127 (2004)
4. Freeman, L.C.: A set of measures of centrality based on betweenness. Sociometry, 35–41 (1977)
5. Geerts, F., Mannila, H., Terzi, E.: Relational link-based ranking. In: VLDB, pp. 552–563 (2004)
6. He, J., Li, M., Zhang, H.-J., Tong, H., Zhang, C.: Manifold-ranking based image retrieval. ACM Multimedia, 9–16 (2004)
7. Koren, Y., North, S.C., Volinsky, C.: Measuring and extracting proximity in networks. In: KDD, pp. 245–255 (2006)
8. Krause, A., Guestrin, C.: Near-optimal nonmyopic value of information in graphical models. In: UAI, pp. 324–331 (2005)
9. Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. In: Proc. CIKM (2003)
10. Newman, M.: A measure of betweenness centrality based on random walks. Social Networks 27, 39–54 (2005)
11. Pan, J.-Y., Yang, H.-J., Faloutsos, C., Duygulu, P.: Automatic multimedia cross-modal correlation discovery. In: KDD, pp. 653–658 (2004)
12. Piegorsch, W., Casella, G.E.: Inverting a sum of matrices. SIAM Review 32, 470 (1990)
13. Tong, H., Faloutsos, C.: Center-piece subgraphs: problem definition and fast solutions. In: KDD, pp. 404–413 (2006)
14. Tong, H., Faloutsos, C., Gallagher, B., Eliassi-Rad, T.: Fast best-effort pattern matching in large attributed graphs. In: KDD, pp. 737–746 (2007)
15. Tong, H., Faloutsos, C., Pan, J.-Y.: Random walk with restart: Fast solutions and applications. Knowledge and Information Systems: An International Journal, KAIS (2008)