ZoomRank: Bridging PageRank and HITS

Siheng Chen
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA
sihengc@andrew.cmu.edu

Qin Gao
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA
qingao@andrew.cmu.edu

Chao Li
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA
chaoli1@andrew.cmu.edu

Jelena Kovačević
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA
jelenak@cmu.edu

Christos Faloutsos
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA
christos+@cs.cmu.edu

ABSTRACT

How do we find important nodes in a large-scale graph? We propose a novel ranking framework that works for large, undirected graphs, handles the degree dilemma and includes PageRank and HITS as special cases. The main idea behind our framework is a principled way to merge multiscale information on graphs. The advantages of our algorithm are (a) generality: it includes PageRank and HITS as special cases; (b) quality: it outperforms PageRank in the tasks of ranking movies, it outperforms HITS on disconnected graphs, and it detects local information that is ignored by both HITS and PageRank; (c) scalability: it is efficient for large-scale graphs.

Categories and Subject Descriptors

H.2.8 [**Database management**]: :Database applications-Data mining

Keywords

Ranking algorithm; PageRank; HITS

1. INTRODUCTION

Ranking is valuable in numerous settings, including ranking of web results [6, 22, 17, 10, 32, 21], personalized ranking [18, 3, 33], importance and influence propagation in blogs [20], recommendation systems and node-proximity estimation [26], strength and influence of friendships [8, 24], ranking in relational databases [2, 14, 1], ranking in citation networks [9, 15], and document ranking [27], among others.

PageRank [6] and HITS [22] are two well-known ranking algorithms; they both have shortages, however. When a graph is undirected, the solution of PageRank is approximately proportional to the degree distribution of the graph [16]; HITS only considers a subgraph that focuses on the given search term and runs at search time. Moreover, these algorithms conflict; for example, when we want to compare the importance between Mike and Tom in Figure 1, PageRank with flyout claims that Mike is more important than Tom, and HITS claims the opposite result. For PageRank, a high-degree neighbor dilutes the turning-back probability, which diminishes the importance; for HITS, a high-degree neighbor accumulates the hub scores, which promotes the importance. In other words, a significant difference between the two algorithms is how to treat the degree of a graph; we call this the *degree dilemma*.

Over the years, many ranking algorithms have been proposed to improve PageRank and HITS from different perspectives. For example, [31] proposed a nonlinear dynamical system to replace the previous linear dynamical systems; [29] proposed a measure to quantify the goodness of a ranking list and optimize the results based on the proposed measure; [12, 23] proposed SimRank, which measures similarity by computing the first-meeting probability of two random surfers, based on the random surfer model; [11] studied rankings that allow ties; [10, 16] studied the degree heuristic; [10, 16] proposed a ranking algorithm based on a matching models learning from training data; [30] proposed a ranking algorithm that works for local graphs. None of these extended algorithms acknowledge the degree dilemma.

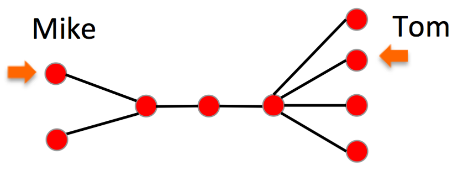


Figure 1: Is Mike more important than Tom? Degree dilemma: PageRank ranks Mike higher, but HITS ranks Tom higher.

To handle the degree dilemma, we revist PageRank and HITS and propose a simple, and unified ranking framework that bridges PageRank and HITS; we name it *ZoomRank*. We further optimize the parameters of the framework and propose a new algorithm ZoomRankOpt, which provides competitive empirical performance on

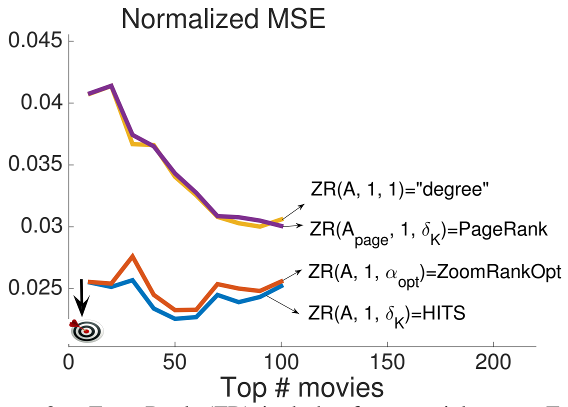


Figure 2: ZoomRank (ZR) includes four special cases; ZoomRankOpt ties HITS and outperforms PageRank in rankinng the Nexflix movies. Lower normalized MSE means better performance.

various datasets; see Figure 2 for an example.

The concept behind ZoomRank (responsible for its name as well) is the fact that we treat a graph structure as a scene in which the algorithm works as a camera capturing important information. ZoomRank provides a general understanding to bridge PageRank and HITS; it is also flexible and allows us to handle the degree dilemma. The main idea behind ZoomRank is a principled way to merge multiscale information on graphs, which is similar to switching the focal length of a camera. The advantages of ZoomRank are

- **Generality**: it provides a general understanding of ranking algorithms and includes PageRank and HITS as special cases; it can switch between the local focus and the global focus on graphs;
- **Quality**: it outperforms PageRank in the tasks of ranking movies; it outperforms HITS on disconnected graphs; and it detects local information that is ignored by both HITS and PageRank;
- **Scalability**: it is efficient for large-scale graphs.

The outline of the paper is typical: we give the survey (Section 2), the proposed method (Section 3), experiments (Section 4), and conclusions (Section 5).

2. BACKGROUND AND RELATED WORK

We consider an undirected graph, denoted as $G = (\mathcal{V}, \mathrm{A})$, where $\mathcal{V} = \{v_n\}_{n=1}^N$ is the set of nodes and $\mathrm{A} \in \mathbb{R}^{N \times N}$ is a symmetric adjacency matrix. The adjacency matrix represents the connections between nodes and the edge weight $\mathrm{A}_{i,j}$ between nodes v_i and v_j is a quantitative expression of the underlying relation between the ith node and the jth node, such as a similarity, a friendship, a dependency, or a communication pattern. When $\mathrm{A}_{i,j}$ is zero, it means that there is no connection between the ith node and the jth node. Let D be a diagonal degree matrix with $\mathrm{D}_{i,i} = \sum_j \mathrm{A}_{i,j}$.

Let $\widetilde{\mathrm{A}} = \mathrm{D}^{-\gamma} \, \mathrm{A} \, \mathrm{D}^{-\beta}$ be a normalized adjacency matrix, where γ is the out-degree influence factor and β is the in-degree influence factor. The influence factors α, β control how we normalize the adjacency matrix. When $\gamma, \beta = 0$, then $\widetilde{\mathrm{A}}$ is an unnormalized adjacency matrix; when $\gamma = 0, \beta = 1$, then $\widetilde{\mathrm{A}} = \mathrm{A} \, \mathrm{D}^{-1}$ is normalized by the in-degrees, which turns out to be a transition matrix. $\widetilde{\mathrm{A}}_{i,j}$ denotes the probability transiting from the ith node to the jth node and the sum of each column in $\widetilde{\mathrm{A}}$ is one; when $\gamma = 1, \beta = 0$, then $\widetilde{\mathrm{A}} = \mathrm{D}^{-1} \, \mathrm{A}$ is normalized by the out-degrees, which turns out to be a consensus matrix; when $\gamma = 1/2, \beta = 1/2$, $\widetilde{\mathrm{A}} = \mathrm{D}^{-1/2} \, \mathrm{A} \, \mathrm{D}^{-1/2}$ is normalized by both in-degrees and out-degrees. The other combinations of γ and β have rarely been considered previously.

The importance score is defined as a map that assigns a value $x_i \in \mathbb{R}$ to the node v_i, where x_i reflects the importance of the node v_i; the larger the x_i, the more important the node v_i. We collect the importance scores of all the nodes as a vector

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \ldots & x_N \end{bmatrix}^T \in \mathbb{R}^N,$$

where the ith vector element x_i is indexed by the node v_i. The goal of a ranking algorithm is to design an importance score based on the graph structure; in other words, we aim to design an algorithm to automatically obtain $\mathbf{x}$ from A, where $\mathbf{x}$ reflects the importance of all the nodes. The final ranking is $\mathbf{x}$ sorted in a descending order. Table 1 summarizes the list of symbols we use.

Symbols	Definitions
A	adjacency matrix
$\widetilde{\mathrm{A}}$	normalized adjacency matrix
P	lens matrix (defined in Section 3.2)
I	identity matrix
D	degree matrix
$\mathbf{x}$	importance score
$\mathbf{e}$	initialization score
$\mathbf{x}^{(k)}$	result of the kth iteration
K	total number of iterations
δ_k	indicator vector

Table 1: Symbols and definitions.

As mentioned in Section 1, many ranking algorithms based on graphs are extended from PageRank and HITS. In this paper, we revisit these two seminal works and study their connections to the degree heuristic. We start with random walk with restart, which provides a family of algorithms and includes PageRank as a special case.

2.1 Random Walk with Restart

Random walk with restart uses a steady-state distribution of random particles to represent the importance of a node [28]. We obtain the steady-state distribution iteratively as follows:

$$\mathbf{x}^{(k+1)} \leftarrow c \, \widetilde{\mathrm{A}} \, \mathbf{x}^{(k)} + (1-c)\mathbf{e}, \tag{1}$$

where c is a probability that return to the original node, $\widetilde{\mathrm{A}}$ is a normalized adjacency matrix, and $\mathbf{e}$ is a starting vector, where each element initializes the prior importance of each node. We call c as a return probability and $1-c$ as a restart probability. The final score is

$$\mathbf{x} = (1-c)(\mathrm{I} - c\widetilde{\mathrm{A}})^{-1}\mathbf{e}. \tag{2}$$

When $\widetilde{\mathrm{A}} = \mathrm{A} \, \mathrm{D}^{-1}$, the solution to (2) is the PageRank; when a graph is undirected, the solution to (2) is approximately the degree, which means that the normalization and the iterative computations do not benefit to the solutions. We will show this in Section 3.3.

As shown in Section 1, because of the normalization of the adjacency matrix, random walk with restart sometimes underestimates the influence of the high-degree nodes.

2.2 HITS

HITS assigns two scores, a hub score and an authority score to each node in a graph [22]. Let $\mathbf{x}_h, \mathbf{x}_a \in \mathbb{R}^N$ be vectors representing the hub scores and authority scores of all the nodes, respectively. These are computed iteratively as follows:

$$\mathbf{x}_h^{(k+1)} \leftarrow \mathrm{A}\,\mathbf{x}_a^{(k)}, \text{ and } \mathbf{x}_a^{(k+1)} \leftarrow \mathrm{A}^T\,\mathbf{x}_h^{(k+1)}, \quad (3)$$

where k denotes the kth iteration, and $\mathbf{x}_h$ and $\mathbf{x}_a$ are normalized after each iteration. Intuitively, the hub score of a node is the sum of the authority scores of all the nodes to which it points, while the authority score of a node is the sum of the hub scores of all the nodes that point to it. For undirected graphs, since A is symmetric, the hub score and the authority score for each node are the same. Under appropriate conditions, $\mathbf{x}_h$ converges to the first left singular vector of A and $\mathbf{x}_a$ converges to the first right singular vector of A. In other words, HITS finds the largest subgraph, or community, and only ranks the nodes in this subgraph. This restricts HITS to run at search time. As shown in Section 1, HITS uses the unnormalized adjacency matrix and sometimes overestimates the importance of the high-degree nodes.

3. OUR UNIFYING FRAMEWORK

In this section we present the proposed method, we analyze it and provide the reader with several interesting observations, at least in our opinion.

3.1 Intuition

The main idea behind our algorithm is to merge multiscale information on graphs (see Figure 3 for illustration). The blue region shows the neighbors of node a (first-order neighbors) and the red region shows the neighbors of neighbors of node a (second-order neighbors). To evaluate the importance of node a, the degree only considers the blue region and counts the number of neighbors of node a, which represents local information. Since local information is limited, we aim to obtain a flexible tradeoff between local focus and global focus; we also consider the red region and other far-away regions, which represents more global information. In our method, for each node, we partition the graph into multiple scales based on the geodesic distance to the node and then we weigh the number of neighbors from multiple scales. Intuitively, the more local scale gets higher weight.

3.2 Algorithm

Let $\mathrm{P} \in \mathbb{R}^{N \times N}$ be a lens matrix of a graph G. It is not necessarily an adjacency matrix; it just reflects the graph structure. Let $\mathbf{e} \in \mathbb{R}^N$ be an initialization score, whose element represents the prior importance of each node. Let $\mathbf{x}^{(k)} = \mathrm{P}^k\,\mathbf{e}$ be the kth-order generalized degree based on the lens matrix P. When P is an unnormalized adjacency matrix, $\mathrm{P} = \mathrm{A}$, and $\mathbf{e} = \mathbf{1}$, the ith element in $\mathbf{x}^{(1)}$ counts the total number of neighbors of the ith node, which is the degree and the ith element in $\mathbf{x}^{(k)}$ counts the total number of nodes that the ith node accesses in k steps. We call $\mathbf{x}^{(k)}$ the *kth-order generalized degree* because P is not necessarily the unnormalized adjacency matrix and $\mathbf{e}$ is not necessarily constant. Each element in the kth-order generalized degree represents an accumulation of information from a certain geodesic distance. When k is small, $\mathbf{x}^{(k)}$ considers local information; as k increases,

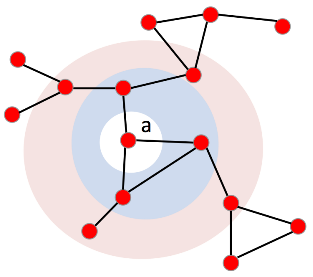


Figure 3: Intuition behind the proposed algorithm. For each node, we partition the graph into multiple scales based on the geodesic distance to the node and weigh information from these scales. Blue region denotes the nodes with geodesic distance 1 from node a, and pink region denotes the nodes with geodesic distance 2 from node a. These two regions contribute to the importance of node a differently.

$\mathbf{x}^{(k)}$ starts to consider more global information. We combine information from multiple scales by weighting the generalized degrees from multiple orders as

$$\mathbf{x} = \sum_{k=0}^{K} \alpha_k \mathbf{x}^{(k)} = \sum_{k=0}^{K} \alpha_k\,\mathrm{P}^k\,\mathbf{e} = h_\alpha(\mathrm{P})\mathbf{e}, \quad (4)$$

where α_k denotes the zoom factor of the kth order and $h_\alpha(\cdot)$ denotes a polynomial function with zoom factors α. The solution $\mathbf{x}$ is a vector representation of the importance scores for all the nodes.

We first equally activate all the nodes and then propagate their activation information on the graph through the lens matrix P. Each time, when the activation information propagates from a node to its neighbors, the importance of activation information diminishes by some constant factor. We finally accumulate all the the activation information within K steps of the propagation at each node to be its importance. We call (4) *ZoomRank* because it allows zooming in to look at local information or zooming out to look at global information on the graph, which is similar to switching the focal length of a camera. ZoomRank can be implemented in an iterative fashion, as shown in Algortihm 1.

```
Data: P: lens matrix
Result: $\mathbf{x}$: importance score
initialization: $k \leftarrow 0, \mathbf{x}^{(k)} \leftarrow \mathbf{1}, \mathbf{x} \leftarrow \mathbf{x}^{(k)}$ ;
while $k < K$ do
    $\mathbf{x}^{(k+1)} \leftarrow \mathrm{P}\,\mathbf{x}^{(k)}$ ;
    $\mathbf{x} \leftarrow \mathbf{x} + \alpha_k \mathbf{x}^{(k+1)}$ ;
    $k \leftarrow k + 1$ ;
Output: $\mathbf{x}$;
```

Algorithm 1: ZoomRank: a simple linear dynamic system.

ZoomRank includes three tuning parameters: a lens matrix P, an initialization score $\mathbf{e}$, and a series of zoom factors $\alpha = \{\alpha_k\}_{k=1}^K$. We treat a graph structure as a scene in which the algorithm works as a camera capturing important information. The lens matrix works as a lens that determines the quality of images; the initialization score determines which part is more important to focus; the zoom factors determine the focal length (vicinity or infinity). Without loss of generality, we set the lens matrix as the normalized adjacency matrix, that is, $\mathrm{P} = \widetilde{\mathrm{A}}$.

The choice of initialization score is useful for personalized ranking [18, 33]. Without knowing any prior knowledge about user preferences, initialization score is constant, $\mathbf{e} = \mathbf{1}$. Depending on specific applications, we choose how to normalize $\widetilde{\mathrm{A}}$ and how to choose the zoom factors, which will be discussed in Section 4. We then parameterize ZoomRank as ZR(P, $\mathbf{e}$, α).

Previously, we consider ZoomRank for general graphs. Here we consider a special case for bipartite graphs, which will be used in Section 4. A bipartite graph is a graph whose nodes can be divided into two disjoint node sets, where any two nodes within the same node set are not connected.

Let the sizes of two sets be N_1 and N_2 ($N = N_1 + N_2$), the importance score be

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \in \mathbb{R}^N,$$

where $\mathbf{x}_1 \in \mathbb{R}^{N_1}$ and $\mathbf{x}_2 \in \mathbb{R}^{N_2}$ corresponds to the importance scores for two node sets, respectively, and the lens matrix is represented as

$$\mathrm{P} = \begin{bmatrix} \mathbf{0} & \mathrm{P}_1 \\ \mathrm{P}_2 & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{N \times N},$$

where $\mathrm{P}_1 \in \mathbb{R}^{N_1 \times N_2}$ and $\mathrm{P}_2 \in \mathbb{R}^{N_2 \times N_1}$. Since the diagonal blocks are zeros, we can update $\mathbf{x}_1$ and $\mathbf{x}_2$ separately, which reduces the computational cost.

3.3 Theoretical Analysis

Here we provide some theoretical analysis for ZoomRank. In Section 2.1, we mention that the solution of random walk with restart is approximately the degree. Here we show the result.

THEOREM 1. (**Random walk with restart is equivalent to "degree" under some assumptions.**) *When* A *is symmetric, the normalized adjacency matrix is* $\widetilde{\mathrm{A}} = \mathrm{D}^{-(1-\beta)}\,\mathrm{A}\,\mathrm{D}^{-\beta}$*, and the return probability is* $c = 1$*. When* (2) *has a unique solution, the solution is* $\mathbf{x} = \mathbf{d}^{\beta}$.

Proof: We just need to check that $\mathbf{x} = \mathbf{d}^{\beta}$ satisfies the steady-state requirement, $\mathbf{x} = \widetilde{\mathrm{A}}\,\mathbf{x}$.

$$\begin{aligned} \widetilde{\mathrm{A}}\,\mathbf{x} &= \mathrm{D}^{-(1-\beta)}\,\mathrm{A}\,\mathrm{D}^{-\beta}\,\mathbf{d}^{\beta} \;=\; \mathrm{D}^{-(1-\beta)}\,\mathrm{A}\cdot\mathbf{1} \\ &= \mathrm{D}^{-(1-\beta)}\,\mathbf{d} \;=\; \mathbf{d}^{\beta} \;=\; \mathbf{x}. \end{aligned}$$

■

Theorem 1 shows that when the adjacency matrix A represents undirected graphs, without considering the restart probability, the solutions of random walk with restart are nothing but the degrees. Even considering the restart probability, the solutions are approximately the degrees [16]. This means that the normalization of adjacency matrix and the iterative computations do not benefit to the solutions of random walk with restart.

We next show that random walk with restart and HITS are special cases of ZoomRank.

THEOREM 2. (**Random walk with restart is a special case of ZoomRank.**) *Random walk with restart has the same solution with ZoomRank when the lens matrix be* $c\,\widetilde{\mathrm{A}} + \frac{1-c}{N}\mathbf{e}\mathbf{1}^T$*, the initialization score be* $\mathbf{1}$*, and the zoom factor is* δ_K*, where* K *is the total number of iterations and* δ_K *is an indicator vector with 1 at* K *and 0, otherwise.*

Proof: When we do not consider the restart probability, Theorem 1 shows that the ranking results are nothing but the degrees, which is ZR(A, $\mathbf{1}$, δ_1).

When we consider the restart probability. Let $\mathrm{P} = \mathrm{A}_{\text{page}} = c\,\widetilde{\mathrm{A}} + \frac{1-c}{N}\mathbf{e}\mathbf{1}^T$. Then, (1) can be reformulated as

$$\begin{aligned} \mathbf{x}^{(k+1)} &= c\,\widetilde{\mathrm{A}}\,\mathbf{x}^{(k)} + (1-c)\mathbf{e} \\ &= \left(c\,\widetilde{\mathrm{A}} + \frac{1-c}{N}\mathbf{e}\mathbf{1}^T\right)\mathbf{x}^{(k)} \\ &= \mathrm{P}\,\mathbf{x}^{(k)} \;=\; \mathrm{P}^{k+1}\,\mathbf{e}. \end{aligned}$$

This is ZR(A_{page}, $\mathbf{1}$, δ_K). ■

Theorem 2 shows that random walk without restart is a special case of ZoomRank that focuses on the vicinity, and random walk with restart is a special case of ZoomRank with a normalized adjacency matrix as the lens matrix. Note that because of the normalization, the solutions are still close to the degree, which still focuses on the vicinity.

THEOREM 3. (**HITS is a special case of ZoomRank.**) *HITS has the same solution with ZoomRank when the lens matrix be*

$$\begin{bmatrix} \mathbf{0} & \mathrm{A} \\ \mathrm{A}^T & \mathbf{0} \end{bmatrix},$$

the initialization score be $\mathbf{1}$*, and the zoom factor is* δ_K.

Proof: Let

$$\mathrm{P} = \begin{bmatrix} \mathbf{0} & \mathrm{A} \\ \mathrm{A}^T & \mathbf{0} \end{bmatrix}, \text{ and } \mathbf{x} = \begin{bmatrix} \mathbf{x}_h \\ \mathbf{x}_a \end{bmatrix}.$$

Then, (3) can be reformulated as

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \begin{bmatrix} \mathbf{x}_h^{(k+1)} \\ \mathbf{x}_a^{(k+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathrm{A} \\ \mathrm{A}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_h^{(k)} \\ \mathbf{x}_a^{(k)} \end{bmatrix} \\ &= \mathrm{P}\,\mathbf{x}^{(k)} \;=\; \mathrm{P}^{k+1}\,\mathbf{e}. \end{aligned}$$

This is ZR(P, $\mathbf{1}$, δ_K). ■

Theorem 3 show that HITS records the steady-state distribution and is a special case of ZoomRank that focuses on infinity. Comparing to both random walk with restart and HITS, ZoomRank records all the propagation processes, which include both local and global information on a graph.

THEOREM 4. (**ZoomRank has a closed-form solution.**) *Let* $\alpha_k = a^k$ *with* $a < 1/\lambda_{\max}(\mathrm{P})$ *and* K *goes to infinity, where* $\lambda_{\max}$ *is the largest eigenvalue of P. The closed-form solution of* (4) *is* $(1 - a\,\mathrm{P})^{-1}\mathbf{e}$.

Proof: The closed-form solution is

$$\begin{aligned}\mathbf{x} &= \lim_{K\to\infty}\sum_{k=0}^{K}\alpha_k\,\mathrm{P}^k\,\mathbf{e}\\ &= \lim_{K\to\infty}\sum_{k=0}^{K}(a\,\mathrm{P})^k\mathbf{e} \;=\; (1-a\,\mathrm{P})^{-1}\mathbf{e}.\end{aligned}$$

■

As mentioned in Section 3.2, the core idea of ZoomRank is to merge information from multiple geodesic distances. We quantify the information from geodesic distance k (k steps away) by using the kth-order generalized degrees. Here we show that ZoomRank follows the ranking principles based on generalized degrees.

AXIOM 1. *Let $\alpha_k \to 0$, ZoomRank satisfies the following principles:*

- *Two nodes have the same importance when the generalized degrees of two nodes are the same in all the orders, that is, $\mathbf{x}_a^{(k)} = \mathbf{x}_b^{(k)}$ for all k; ;*
- *Node a is more important than node b when there exists an integer $k > 0$, such that the kth-order generalized degree of node a is larger than that of b and at the same time, for all $i \le k$, the ith-order generalized degrees of node a are the same with that of node b, that is, $\mathbf{x}_a^{(k)} > \mathbf{x}_b^{(k)}$ and $\mathbf{x}_a^{(i)} = \mathbf{x}_b^{(i)}$, for all $i \le k$.*

Axiom 1 shows a principle to compare the importance of nodes based on generalized degrees: when two nodes have the same generalized degree in an order, we compare their generalized degrees in the next order, until they are different. When P is an unnormalized adjacency matrix, the kth-order generalized degree of node a is the total number of nodes that are k steps away from node a. Following Axiom 1, more connections, larger generalized degree and higher importance.

3.4 Complexity Analysis

THEOREM 5. *The computational complexity of ZoomRank is $O(K\,\|\mathrm{P}\|_0)$.*

Proof: In each iteration, the bottleneck of computation is a matrix-vector multiplication. The computational complexity depends on the number of nonzero elements in P, which is $\|\mathrm{P}\|_0$. We have K iterations in total. ■

4. EXPERIMENTS

Here we report experiments to answer the following questions

- Q1: How are parameters for ZoomRank chosen?
- Q2: How well does ZoomRank work on real data?

All experiments were performed on a laptop with 2.3 GHz Intel Core i7 CPU and 16 GB 1600 MHz DDR3 memory.

4.1 Q1: parameter choice - ZoomRankOpt

ZoomRank includes three tuning parameters: the lens matrix P, initialization score $\mathbf{e}$, and the zoom factors α.

Lens matrix. From Theorem 2, we see that when the lens matrix is the normalized adjacency matrix, the solutions of ZoomRank focus on the vicinity, which are approximately the degrees. Instead of iterating several times, we can set the lens matrix be an unnormalized adjacency matrix and propagate only once. For this reason, we consider the unnormalized adjacency matrix as the lens matrix here.

Initialization score. Initialization score provides the prior importance before ranking. The choice is based on personalized information. Without any prior information, we treat each node equally, that is, $\mathbf{e} = \mathbf{1}$. When we have the prior information, we can easily adapt $\mathbf{e}$ and achieve personalized ranking.

Zoom factors. When zooming into the vicinity, we activate the first zoom factor in (4), which is degree; when zooming out infinitely, we activate the last zoom factor in (4), which is HITS; when we want to combine both the local and global information, we set

$$\alpha_k^{\mathrm{opt}} \;=\; \left(\frac{1-\epsilon}{\lambda_{\max}(\mathrm{A})}\right)^k, \tag{5}$$

where $\epsilon = 0.05$. We denote $\alpha_{\mathrm{opt}} = \{\alpha_k^{\mathrm{opt}}\}_{k=0}^{K}$. As shown in Theorem 4, since $\epsilon > 0$, ZoomRank converges to a closed-form solution. As the step k grows, the zoom factor α_{opt} has an exponential decay and ZoomRank mainly collects local information. Since ϵ is small, the zoom factor decays slowly to capture sufficient global information. Thus, ZoomRank collects multiscale information. We call this setting ZoomRankOpt, that is $\mathrm{ZR}(\mathrm{A}, \mathbf{1}, \alpha_{\mathrm{opt}})$.

4.2 Quality

We validate the quality by studying two applications. The graphs we used in our experiments are described in the table 2.

Description	Nodes	Edges
MovieLens 100K	2,625	10,000
Netflix	497,959	100,480,507
NELL	1,971,555	2,041,006

Table 2: Summary of real-world graphs used.

MovieLens. MovieLens dataset was collected by the GroupLens Research Project at the University of Minnesota [19]. The dataset was collected through the MovieLens web site (movielens.umn.edu) during the seven-month period from September 19th, 1997 through April 22nd, 1998. The dataset contains 100,000 ratings (1-5) from 943 users on 1682 movies and each user has rated at least 20 movies.

The goal here is to rank the importance of the movies without any rating. We only know which movies users watch; for example, user i watches movie j. The intuition behind ranking here is that (a) a movie with more views tends to be better; (b) a user who watches more movies tends to be more experienced and more important. When we use the degree heuristic, we simply consider (a).

To use the ranking algorithms, we first construct a bipartite graph to represent the the movie-user relation and the corresponding adjacency matrix is

$$\mathrm{A} = \begin{bmatrix} \mathbf{0} & \mathrm{A}_b \\ \mathrm{A}_b^T & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{N \times N},$$

where $\mathrm{A}_b \in \mathbb{R}^{N_1 \times N_2}$ $(N_1 = 1682,\ N_2 = 943, N = N_1 + N_2)$ with

$$(\mathrm{A}_b)_{i,j} = \begin{cases} 1, & j\text{th user watches } i\text{th movie;} \\ 0, & \text{otherwise.} \end{cases}$$

We then consider four settings of ZoomRank, including "degree" (the degree heuristic), PageRank, HITS and ZoomRankOpt. For "degree", we rank the movies by counting how many users watch each movie; for PageRank, we set the return probability is 0.85. These four settings of ZoomRank focus on different information on a graph: "degree" considers the local information by only counting the number of the first-order neighbors; PageRank diminishes the influence from neighbors by normalizing the degree, and the results are similar to "degree" ; HITS considers the global information by only considering the steady-state result; and ZoomRankOpt collects multiscale information and has a closed-form solution, which is shown in Theorem 4. For PageRank, HITS and ZoomRankOpt, we iterate 100 times.

A good ranking algorithm should rank good movies to the top of the ranking list. In the dataset, 943 users already provided their opinions as the ratings. We thus simply compare the results of the ranking algorithms to the average opinions of 943 users. Let $\mathbf{x} \in \mathbb{R}^N$ be a vector containing ratings of all the movies averaged over 943 users; for example, x_i is the rating of movie i averaged over 943 users. A subscript select a subset from $\mathbf{x}$; for example, $\mathbf{x}_{\text{groundtruth}} \in \mathbb{R}^M$ be a vector containing ratings of top M movies with the highest averaged ratings in $\mathbf{x}$, and $\mathbf{x}_{\text{rank}} \in \mathbb{R}^M$ be a vector containing ratings of top M most important movies provided by a ranking algorithm. We evaluate a ranking algorithm by the normalized mean square error.

$$\text{NormalizedMSE} = \frac{\|\mathbf{x}_{\text{groundtruth}} - \mathbf{x}_{\text{rank}}\|_2^2}{\|\mathbf{x}_{\text{groundtruth}}\|_2^2}.$$

A smaller normalized mean square error means the ranking algorithm is better because its result is closer to the groundtruth ratings. Figure 4(a) compares the performance of four algorithms. We see that ZoomRankOpt and HITS have similar performances, which are much better than "degree" and PageRank. Recall that "degree" only considers that a good movie tends to have more views, but ignores that the ratings from different users may have different importance: users who watch more movies tend be more importance. On the other hand, ZoomRankOpt and HITS consider both aspects by operating on movie data then user data iteratively. The difference of ZoomRankOpt and HITS is the zoom factors during the interaction.

Table 3 shows the top 10 most important movies given by four ranking algorithms in the MovieLens 100K dataset. The average ratings of top 10 important movies provided by "degree", PageRank, HITS and ZoomRankOpt are 3.75, 3.75, 4.05, and 4.05, respectively, which again shows that ZoomRankOpt and HITS outperforms "degree" and PageRank. Note that the average rating of all the movies is 3.07, this means that all four of these algorithms detect fairly good movies just from the movie-user relations. From Table 3, we also see that the results provided by "degree" and PageRank are similar, which validates Theorem 1. The results provided by ZoomRankOpt and HITS are also similar, which implies that global information quickly dominates.

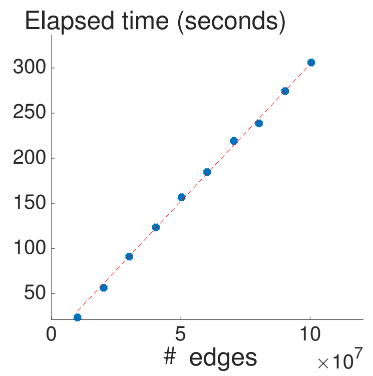


Figure 6: Elapsed time grows with the size of dataset linearly. ZoomRank ranks the Netflix dataset with $2,667,199$ nodes within a few minutes on a laptop.

Netflix. The Netflix dataset was provided by Netflix, an online DVD-rental and online video streaming service for conducting the Netflix competition [5, 4]. The dataset contains $100,480,507$ ratings $(1-5)$ from $480,189$ users on $17,770$ movies. Similarly to the MovieLens 100K, we aim to rank the movies without knowing any ratings. We use the same four algorithms for ranking.

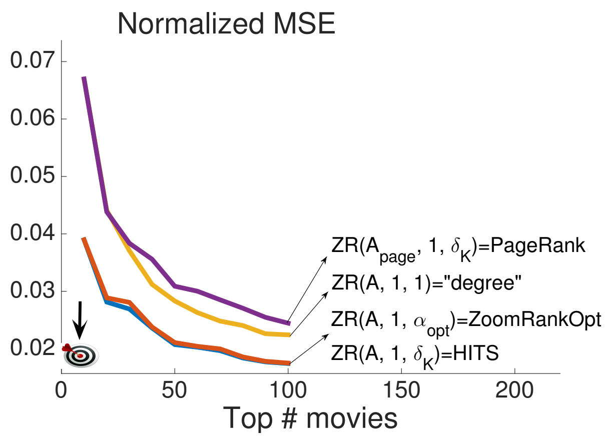
Figure 4(b) compares the performance of four algorithms. We see that ZoomRankOpt and HITS have similar performances, which are better than "degree" and PageRank. Table 3 shows the top 10 most important movies in the MovieLens 100K dataset. The average ratings of top 10 most important movies provided by "degree", PageRank, HITS and ZoomRankOpt are 3.78, 3.78, 3.97, and 3.97, respectively. The average rating of all the movies is 3.23. Similarity to MovieLens 100K, results provided by "degree" and PageRank are similar and the results provided by ZoomRankOpt and HITS are similar. We also show the scalability in Figure 6. We see the elapsed time grows with the size of dataset linearly, which validates Theorem 5.

4.3 Discussion

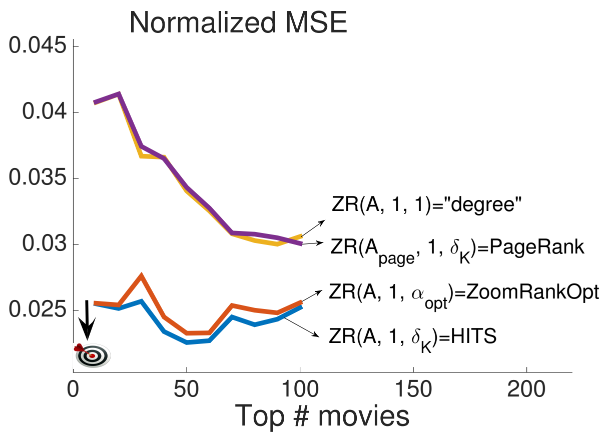
We next use a synthetic example and a dataset from the never ending language learning system to further illustrate how ZoomRank works under various parameter settings.

Synthetic data. Previously, we see both ZoomRankOpt and HITS perform well on MovieLens 100K and Netflix. Here we show that HITS fails in a simple synthetic dataset, but ZoomRankOpt still works well.

We generate a graph with 200 nodes as shown in Figure 5(a). The First 10 nodes form an Erdős-Rényi graph with connectivity probability 0.8; the other 190 nodes form a preferential-attachment graph with one adding edge in each iteration. Figures 5(b) and (c) show the importance scores of HITS and ZoomRankOpt, respectively. We see that HITS only provides the scores for the first 10 nodes and cannot distinguish the importance of the other 190 nodes. On the other hand, ZoomRankOpt successfully detects the importance of the first 10 nodes and also ranks the other 190 nodes well. Since HITS only records the steady-state distribution, which is the first eigenvector of the adjacency matrix when the graph is barely con-



(a) MovieLens 100K.



(b) Netflix.

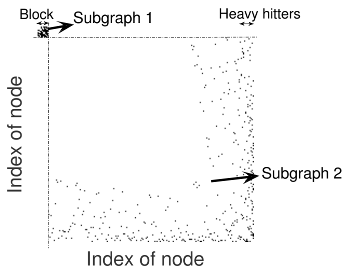
Figure 4: ZoomRanksOpt ties HITS and outperforms PageRank in the MovieLens 100K dataset. Lower Normalized MSE means better performance.

ZR$(A, \mathbf{1}, \delta_1)$ ="degree"	ZR$(A_{\text{page}}, \mathbf{1}, \delta_K)$ = PageRank	ZR$(A, \mathbf{1}, \delta_K)$ = HITS	ZR$(A, \mathbf{1}, \alpha_{\text{opt}})$ = ZoomRankOpt
Star Wars (4.36)	Star Wars (4.36)	Star Wars (4.36)	Star Wars (4.36)
Contact (3.80)	Contact (3.80)	Return of the Jedi (4.01)	Return of the Jedi (4.01)
Fargo (4.16)	English Patient (3.66)	Raiders of the Lost Ark (4.25)	Raiders of the Lost Ark (4.25)
Return of the Jedi (4.01)	Liar Liar (3.16)	Fargo (4.16)	Fargo (4.16)
Liar Liar (3.16)	Scream (3.44)	Pulp Fiction (4.06)	Pulp Fiction (4.06)
English Patient (3.65)	Fargo (4.16)	Silence of the Lambs (4.29)	Silence of the Lambs, The (4.29)
Scream (3.44)	Return of the Jedi (4.01)	Independence Day (3.44)	Independence Day (3.44)
Toy Story (3.88)	Air Force One (3.63)	Empire Strikes Back (4.20)	Toy Story (3.87)
Air Force One (3.63)	Toy Story (3.87)	Toy Story (3.87)	Empire Strikes Back (4.20)
Independence Day (3.44)	Independence Day (3.44)	Back to the Future (3.83)	Back to the Future (3.83)

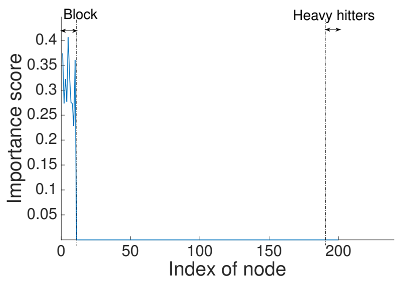
Table 3: ZoomRank ranks top movies in the MovieLens 100K dataset. The groundtruth rating over 943 users is shown in the parentheses. HITS and ZoomRankOpt outperform "degree" and PageRank.

ZR$(A, \mathbf{1}, \delta_1)$ ="degree"	ZR$(A_{\text{page}}, \mathbf{1}, \delta_K)$ = PageRank	ZR$(A, \mathbf{1}, \delta_K)$ = HITS	ZR$(A, \mathbf{1}, \alpha_{\text{opt}})$ = ZoomRankOpt
Miss Congeniality (3.36)	Miss Congeniality (3.36)	Pirates of the Caribbean (4.15)	Pirates of the Caribbean (4.15)
Independence Day (3.72)	Independence Day (3.72)	Independence Day (3.72)	Miss Congeniality (3.36)
The Patriot (3.78)	The Patriot (3.78)	Miss Congeniality (3.36)	Independence Day (3.72)
The Day After Tomorrow (3.44)	The Day After Tomorrow (3.78)	Forrest Gump (4.30)	Forrest Gump (4.30)
Pirates of the Caribbean (4.15)	Pirates of the Caribbean (4.15)	Ocean's Eleven (3.89)	Pretty Woman (3.91)
Pretty Woman (3.91)	Pretty Woman (3.91)	Pretty Woman (3.91)	The Patriot (3.78)
Forrest Gump (4.30)	The Green Mile (4.31)	The Patriot (3.78)	Ocean's Eleven (3.88)
The Green Mile (4.31)	Con Air (3.45)	The Sixth Sense (4.33)	The Green Mile (4.31)
Con Air (3.45)	Forrest Gump (4.30)	The Bourne Identity (3.97)	The Sixth Sense (4.33)
Twister (3.41)	Twister (3.41)	The Green Mile (4.31)	The Bourne Identity (3.97)

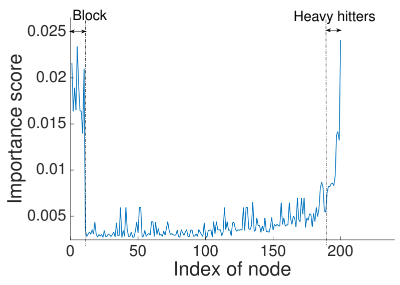
Table 4: ZoomRank ranks top movies in the Netflix dataset. Average ratings are shown in the parentheses. HITS and ZoomRankOpt outperform "degree" and PageRank.



(a) Adjacency matrix.



(b) HITS.



(c) ZoomRankOpt.

Figure 5: HITS fails to detect heavy hitters and ZoomRankOpt detects both block and heavy hitters. The simulated graph contains two subgraphs: subgraph 1 is a Erdős-Rényi graph with 10 nodes and connectivity probability 0.8; subgraph 2 is a preferential-attachment graph with 190 nodes and one adding edge in each iteration.

nected, HITS may assign an important score of zero to many nodes. ZoomRankOpt avoids this by keeping track of the iterative processes.

NELL. The never ending language learning system (NELL) is a semantic machine learning system to develop means of answering questions posed by users in natural language with no human intervention in the process [25, 7]. Here we study a dataset collected by NELL that contains $2,041,006$ facts. Each fact is a triple containing a subject, a verb and an object. The NELL dataset includes $1,632,580$ subjects and $338,975$ objects in total [13]. The goal is to rank the importance of the subjects and objects from those facts. For example, we find the most important city in the NELL data by looking for the most important subject related to the object "city". The basic assumption is that (a) important facts appear more frequently and (b) a fact tends to be more important when it associates with an important subject or object.

We construct a bipartite graph to represent the the subject-object relations and the corresponding adjacency matrix is

$$\mathrm{A} = \begin{bmatrix} \mathbf{0} & \mathrm{A}_b \\ \mathrm{A}_b^T & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{N \times N},$$

where $\mathrm{A}_b \in \mathbb{R}^{N_1 \times N_2}$ ($N_1 = 1,632,580, N_2 = 338,975, N = N_1 + N_2 = 1971555$) with

$$(\mathrm{A}_b)_{i,j} = \begin{cases} 1, & i\text{th subject and } j\text{th object coappear;} \\ 0, & \text{otherwise.} \end{cases}$$

To compare four algorithms, we show the top 10 most important subjects related to the object of actor given by four algorithms in Tables 5. Both "degree" and PageRank rank Tom Cruise as the most important actor; HITS ranks Christian Bale as the most important actor; and ZoomRankOpt ranks Kevin Bacon as the most important actor. People may have their own preferences and there is no ground-truth of the ranking. Here we provide some analysis of the connections of Tom Cruise, Christian Bale and Kevin Bacon to understand how these ranking algorithms work.

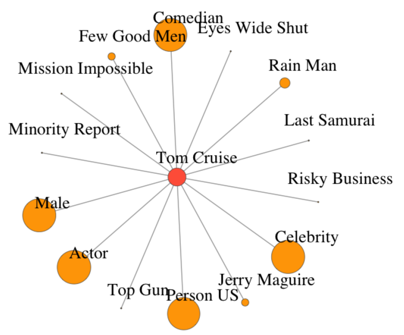
Figure 7(a) shows the ego-graph of Tom Cruise. As a subject, Tom Cruise directly connects to 14 objects, including 5 categories and 9 movies. When we look at the second-order neighbors, the categories of actor, celebrity, comedian, male and person US contributes to 25602, 19242, 15637, 18141, and 10333 connections, respectively; while 9 movies only contribute to 5 connections in total.

Figure 7(c) shows the ego-graph of Christian Bale. As a subject, Christian Bale directly connects to 11 objects, including 5 categories, which are the same with Tom Cruise, and 6 movies. Since the number of the first-order neighbor of Christian Bale is smaller than that of Tom Cruise, "degree" ranks Tom Cruise in the first place and Christian Bale in the 6th place. However, when we look at the second-order neighbors, Christian Bale's 6 movies contributes to 9 connects, which are larger than that of Tom Cruise. This leads to HITS ranking Christian Bale as more important than Tom Cruise.

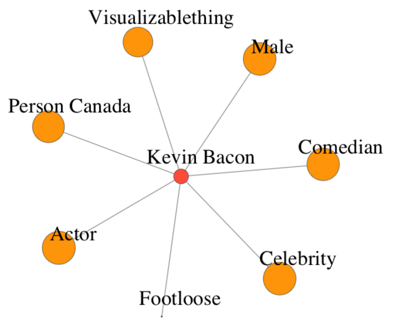
Figure 7(b) shows the ego-graph of Kevin Bacon. As a subject, Kevin Bacon directly connects to 7 objects, including 6 categories and only 1 movies. The number of the first-order neighbor of Kevin Bacon is smaller than that of Tom Cruise and Christian Bale, thus "degree" ranks Kevin Bacon in the 88th place. However, Kevin Bacon connects to one more category, visualizablething, which brings additional 2306 second-order connections. Kevin Bacon thus has more connections in the second order and this leads to ZoomRankOpt ranking Kevin Bacon in the first place.

Figure 7(d) shows who has more connection in different orders, that is, we compare who has larger total number of k-step-away connections. We see that Tom Cruise has the most 1-step-away connections; Christian Bale has the most far-away connections; and Kevin Bacon is in-between. Recall that "degree" only focuses on the vicinity, which is the number of the first-order connections; HITS focuses on the infinity, which is the number of large-order connections; ZoomRankOpt merges multiscale information and detects the information in-between. This explains why both "degree" and HITS treat Kevin Bacon as less important than Christian Bale, but ZoomRankOpt treats Kevin Bacon as more important than Christian Bale. Based on different users' needs, ZoomRank is flexible, providing personalized results by focusing on different information on graphs.

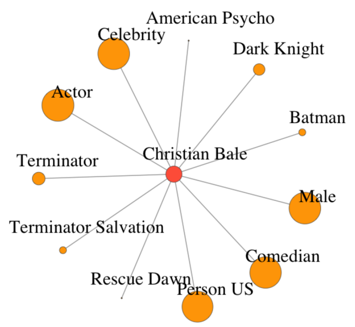
Why ZoomRank? As a general algorithm, ZoomRank works as a plugin framework, that is, it allows for the selection of different parameters to adapt to various real-world needs. ZoomRank not only includes random walk with restart and HITS as special cases, but also provides a conceptual bridge connecting random walk with restart and HITS through the concept of generalized degree. It reveals that the normalization of the adjacency matrix leads to the degree dilemma, and that the choice of zoom factors allows for the detection of different information on the graph. We also propose a new special case of ZoomRank, called ZoomRankOpt. It weighs



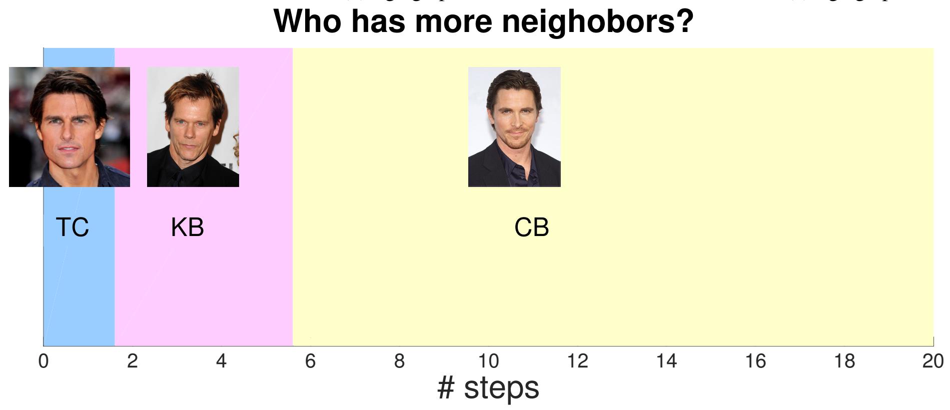
(a) Ego graph of Tom Cruise.



(b) Ego graph of Kevin Bacon.



(c) Ego graph of Christian Bale.



(d) Who has more k-step-away connections.

Figure 7: ZoomRank is flexible. Tom Cruise (TC) has many 1-step-away connections, and wins out "degree" (=ZR$(\mathrm{A}, \mathbf{1}, \delta_1)$; Christian Bale (CB) has many far-away connections, and wins out HITS (=ZR$(\mathrm{A}, \mathbf{1}, \delta_K)$; Kevin Bacon (KB) is in-between, and wins out ZoomRankOpt.

multiscale information and works as a supplement of random walk with restart and HITS. When a graph is not well connected, HITS only provides important scores for the nodes in the largest component and the importance scores of all the other nodes are zero, which makes HITS an online algorithm; see Figure 5. On the other hand, PageRank and ZoomRankOpt are offline algorithms, providing important scores for all the nodes.

In practice, we usually do not have a groundtruth for ranking algorithms without an expensive user study; it is then hard to evaluate whether one algorithm is better than the other. In this paper, we consider evaluating the ranking algorithms in the task of rating movies. The user study is implicitly provided by the ratings from a large number of users; we rank the movies by sorting the users' ratings and use this ranking as the groundtruth. The results are thus less biased because of the large number of users. ZoomRank provides good performance in both MovieLens 100K and Netflix dataset, as shown in Figure 4. Since ZoomRank is a linear dynamic system, it has nice convergence properties and is easily scalable.

5. CONCLUSIONS

We presented ZoomRank, which addresses the ranking problem for large, undirected graphs. ZoomRank provides a general understanding to two seminal ranking algorithms, PageRank and HITS, and includes them as special cases; it is also flexible in handling the degree dilemma. The main idea behind ZoomRank is a principled way to merge multiscale information on the graph. ZoomRank works like a camera that allows zooming in to look at local information or zooming out to look at global information on graphs. The advantages of our algorithm are

- **Generality.** ZoomRank provides a general ranking framework that bridges random walk with restart and HITS, see Theorems 2 and 3.
- **Quality**. ZoomRankOpt outperforms PageRank in ranking movies, see Figure 4; ZoomRankOpt outperforms HITS in disconnected graphs, see Figure 5; ZoomRankOpt detects the local information that is ignored by both HITS and PageRank, see Figure 7.
- **Scalability**. ZoomRank has the similar computational cost with PageRank and HITS, see Theorem 5 and Figure 6.

Reproducibility: We have already open-sourced our code, at `http://users.ece.cmu.edu/~sihengc/research.html`

6. REFERENCES

[1] R. Agrawal, R. Rantzau, and E. Terzi. Context-sensitive ranking. In *Proceedings of the ACM SIGMOD, Chicago, Illinois, USA, June 27-29, 2006*, pages 383–394, 2006.

[2] S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis. Automated ranking of database query results. In *CIDR*, 2003.

[3] B. Bahmani, A. Chowdhury, and A. Goel. Fast incremental and personalized pagerank. *PVLDB*, 4(3):173–184, 2010.

[4] R. M. Bell, Y. Koren, and C. Volinsky. The bellkor solution to the netflix prize. *Technical report, AT&T Labs*, 2007.

[5] J. Bennett and S. Lanning. The netflix prize. *Proceedings of KDD Cup and Workshop 2007*, Aug. 2007.

ZR($\mathrm{A}, \mathbf{1}, \delta_1$) ="degree"	ZR($\mathrm{A}_{\mathrm{page}}, \mathbf{1}, \delta_K$) = PageRank	ZR($\mathrm{A}, \mathbf{1}, \delta_K$) = HITS	ZR($\mathrm{A}, \mathbf{1}, \alpha_{\mathrm{opt}}$) = ZoomRankOpt
Tom Cruise	Tom Cruise	Christian Bale	Kevin Bacon
Mel Gibson	Mel Gibson	Eddie Murphy	Christian Bale
Brad Pitt	Ferrell	Jim Carrey	Eddie Murphy
Clint Eastwood	Movie Actors	Daniel Craig	Ron Silver
Jack Nicholson	Eddie Murphy	Harrison Ford	Jim Carrey
Christian Bale	Tom Hanks	Adam West	Seth Green
Eddie Murphy	Johnny Depp	Val Kilmer	Denzel Washington
Johnny Depp	Jim Carrey	Denzel Washington	Harrison Ford
Tom Hanks	Brad Pitt	Jeff Bridges	Bruce Willis
Marlon brando	Kimora Lee Simmons	Keanu Reeves	Sean Penn

Table 5: ZoomRank efficiently provides top actors in the NELL dataset .

[6] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

[7] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI 2010*, 2010.

[8] P. DeScioli, R. Kurzban, E. N. Koch, and D. Liben-Nowell. Best friends: Alliances, friend ranking, and the myspace social network. *Perspectives on Psychological Science*, 6(1):6–8, Jan. 2011.

[9] Y. Ding, E. Yan, A. R. Frazho, and J. Caverlee. Pagerank for ranking authors in co-citation networks. *JASIST*, 60(11):2229–2243, 2009.

[10] D. Donato, S. Leonardi, and P. Tsaparas. Stability and similarity of link analysis ranking algorithms. *Internet Mathematics*, 3(4):479–507, 2006.

[11] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, and E. Vee. Comparing partial rankings. *SIAM J. Discrete Math.*, 20(3):628–648, 2006.

[12] D. Fogaras and B. Rácz. Scaling link-based similarity search. In *WWW 2005, Chiba, Japan, May 10-14, 2005*, pages 641–650, 2005.

[13] M. Gardner, P. P. Talukdar, B. Kisiel, and T. Mitchell. Improving learning and inference in a large knowledge-base using latent syntactic cues. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013.

[14] F. Geerts, H. Mannila, and E. Terzi. Relational link-based ranking. In *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004*, pages 552–563, 2004.

[15] R. Ghosh, T. Kuo, C. Hsu, S. Lin, and K. Lerman. Time-aware ranking in dynamic citation networks. In *ICDMW, Vancouver, BC, Canada, December 11, 2011*, pages 373–380, 2011.

[16] V. Grolmusz. A note on the pagerank of undirected graphs. *Inf. Process. Lett.*, 115(6-8):633–634, 2015.

[17] Z. Gyöngyi, H. Garcia-Molina, and J. O. Pedersen. Combating web spam with trustrank. In *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004*, pages 576–587, 2004.

[18] T. H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Trans. Knowl. Data Eng.*, 15(4):784–796, 2003.

[19] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, Aug. 1999.

[20] D. Ienco, F. Bonchi, and C. Castillo. The meme ranking problem: Maximizing microblogging virality. In *ICDMW*, pages 328 – 335, Dec 2010.

[21] S. Kashoob, J. Caverlee, and K. Y. Kamath. Community-based ranking of the social web. In *HT'10, Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, Toronto, Ontario, Canada, June 13-16, 2010*, pages 141–150, 2010.

[22] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46:604âĂŞ632, Sept. 1999.

[23] P. Li, H. Liu, J. X. Yu, J. He, and X. Du. Fast single-pair simrank computation. In *SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA*, pages 571–582, 2010.

[24] D. Liben-Nowell, C. Knipe, and C. Coalson. Indifferent attachment: the role of degree in ranking friends. In *ASONAM '13, Niagara, ON, Canada - August 25 - 29, 2013*, pages 1416–1417, 2013.

[25] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. Never-ending learning. In *AAAI*, 2015.

[26] J. Ni, H. Tong, W. Fan, and X. Zhang. Inside the atoms: ranking on a network of networks. In *KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 1356–1365, 2014.

[27] J. Petterson, T. S. Caetano, J. J. McAuley, and J. Yu. Exponential family graph matching and ranking. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, pages 1455–1463, 2009.

[28] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. *ICDM*, pages 613–622, Dec. 2006.

[29] H. Tong, J. He, Z. Wen, R. Konuru, and C. Lin. Diversified ranking on large graphs: an optimization viewpoint. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 1028–1036, 2011.

[30] M. Trevisiol, L. M. Aiello, P. Boldi, and R. Blanco. Local ranking problem on the browsegraph. In *SIGIR, Santiago, Chile, August 9-13, 2015*, pages 173–182, 2015.

[31] P. Tsaparas. Using non-linear dynamical systems for web searching and ranking. In *Proceedings of the Twenty-third ACM SIGACT-SIGMOD-SIGART, June 14-16, 2004, Paris, France*, pages 59–70, 2004.

[32] Y. V. Volkovich. *Stochastic analysis of web page ranking*. PhD thesis, University of Twente. CTIT, 2009.

[33] T. Zhao, J. J. McAuley, and I. King. Leveraging social connections to improve personalized ranking for collaborative filtering. In *CIKM, Shanghai, China, November 3-7, 2014*, pages 261–270, 2014.