HITS vs. Non-negative Matrix Factorization

Yuanzhe Cai, Sharma Chakravarthy

Technical Report CSE 2014
HITS vs. Non-negative Matrix Factorization

January 28, 2014

Abstract

Ranking algorithms have been widely used for web and other networks to infer quality/popularity. Both PageRank and HITS were developed for ranking web pages from a web reference graph. Nevertheless, these algorithms have also been applied extensively for a variety of other applications such as question-answer services, author-paper graphs, and others where a graph can be deduced from the data set. The intuition behind HITS has been explained in terms of hubs and spokes as two values are inferred for each node. HITS has also been used extensively for ranking in other applications although it is not clear whether the same intuition carries over. It would be beneficial if we can understand these algorithms mathematically in a general manner so that the results can be interpreted and understood better for different applications. This paper provides such an understanding for applying HITS algorithm to other applications.

In this paper, we generalize the graph semantics in terms two underlying concepts: in-link probability (ILP) and out-link probability (OLP). Using these two, the rank scores of nodes in a graph are computed. We propose the standard non-negative matrix factorization (NMF) approach to calculate ILP and OLP vectors. We also establish a relationship between HITS vectors and ILP/OLP vectors which enables us to better understand HITS vectors associated with any graph in terms of these two probabilities. Finally, we illustrate the versatility of our approach using different graph types (representing different application areas) and validate the results. This work provides an alternative way of understanding HITS algorithm for a variety of applications.

1 Introduction

Empirical studies and theoretical modeling of networks have been the subject of a large body of research work in statistical mathematics and computer science [1, 2, 3]. Network ideas have been widely applied with success to the topics as diverse as the world wide web [4], scientific citation [5], email communication [6], community question answer services (CQAs) [7, 8], epidemiology [9], ecosystems [10, 11] and bioinformatics [12], to name but a few. Since a number of applications need to identify an order (or ranking) of nodes from graphs, several ranking algorithms have been proposed to bring order to these graphs. In 1999, Kleinberg [13] proposed the HITS algorithm to calculate the hub and authority
Table 1: Possible URLs to Introduce the Oracle Company

<table>
<thead>
<tr>
<th>Description</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Main Page</td>
<td><a href="http://www.oracle.com/">www.oracle.com/</a></td>
</tr>
<tr>
<td>Oracle On Twitter</td>
<td><a href="http://twitter.com/#!/oracle">http://twitter.com/#!/oracle</a></td>
</tr>
</tbody>
</table>

score in a web reference graph. Around the same time, Page et al. [4] proposed the PageRank algorithm to identify web page authority in a web reference graph. Since these two algorithms are parameter-free and easy to compute, they have been widely used in numerous real-world applications. Meanwhile, these two algorithms have also been extended to different types of graphs to analyze the order of nodes, such as paper citation graph [5], email graph [6], bioinformatics graph [14], ask-answer graph [8] and so on. Although the intuition behind HITS has been explained in the context of web reference graphs, their intuition for other applications is not so clear. This paper revisits the HITS approach to provide an alternative way to compute it and provide a graph property-based intuition.

As an example, consider a web writer, Steve, who is creating his own personal web page. Because he works at Oracle company, he wants to use a reference to Oracle company in his personal web page. Steve searches on key word “Oracle” and finds a lot of related web pages (See Table 1). Steve can use any of these web page to introduce his company. However, Steve chooses Oracle’s main page for his personal web page because Steve believes that Oracle main page is a better web page to introduce Oracle company than others. In other words, in the web design process a user typically chooses the best web page to link to. We observe two characteristics from the web reference graph. First, links in the reference graph describe the explanation relationship. A web writer creates a URL to a web page because this web writer wants to use this web page to explain an anchor text in his/her web page. In Example 1, web writer, Steve, finds a web page to introduce/explain Oracle company (anchor text). Second, a writer in these graphs always chooses a higher quality web page (in his/her opinion) to create a link. In Example 1, Steve chooses Oracle main page as the target web page because he believes that this web page is a better web page to introduce Oracle company than others. In summary, we have one important observation for these web reference graphs: In the web reference graph writers/web page developers typically choose higher quality web page to explain anchor an text.

For retrieval, users input a few key words and the search engine returns web pages ranked by their relevance. Thus, the global rank (ordered by web page’s quality) can be defined as the probability of the web pages to be used to explain the input key words. Our observation is that in a web reference graph, the web page writer uses a reference (or link) in his/her page based on the quality of the referenced web page (in his/her opinion). Therefore, the global rank can also

1 We do not consider spam user’s behavior because it is more of an exception.
be deemed as the probability of this web page to be chosen as the quality page to explain anchor texts. In other words, this rank score can also be understood as the probability of this web page to receive a link from other web pages.

In general, for any node in a graph we define out-link probability (OLP) and in-link probability (ILP) to describe the rank of that node. The OLP (ILP) of node $a$ describes the probability of node $a$ to create an outgoing (incoming) link to other nodes, respectively. Both ILP and OLP represent different semantics of a graph. A node with high ILP (as determined by the number of in-edges) represents a node with high quality or popularity whether it is web pages or citations or friends etc. In contrast, a node with high OLP represents a node that independently indicates that the node has high connectivity which can be interpreted as a hub for a web reference graph, as a paper with large number of citations, or a person with large number of friends. It is also possible to interpret, in general, high in-degree (ILP) as indicating depth while high out-degree (OLP) as indicating breadth. Of course, it is also possible for a node to be both. Hence, we need to calculate both of these values for each node. We use two vectors $u$ and $v$, respectively, to represent OLP and ILP values of each node. Thus, $uv^T$ describes the probability of creating a link between any two nodes. Let $e$ be the number of edges. Since we can easily calculate the probability of creating a link between any two nodes as $\frac{1}{2}L$ (where $L$ is the adjacency matrix), we obtain the vector $u$ and $v^T$ by solving a cost function $\min \| \frac{1}{2}L - uv^T \|^2_2$. Since we calculate ILP vector $v^T$ and OLP vector $u$ by decomposing the adjacency matrix $L$, we term this approach as ILOD (pronounced Illiad) approach.

We also establish a relationship between HITS and ILOD. The hub vector of HITS has the same rank order of OLP vector $u$ and the authority vector of HITS has the same rank order of ILP vector $v^T$. In Kleinberg’s description [13], we only know the calculation of HITS but do not know the meaning of HITS scores for each node. Since we establish a relationship between HITS and ILOD, we can use OLP and ILP score to explain HITS vectors. We also stress that this understanding is very important because we can clearly know the meaning of these scores when we extend HITS approach to different types of social graphs.

Furthermore, we can apply the ILOD approach to diverse graph types that represent different applications. In the first application (directed graph), we apply this approach to identify experts in Community Question Answering services (CQAs). Here, an edge is drawn from the user who asked a question to the user who answered it. The user’s expertise in the CQAs can also be described as the probability of this user answering many questions so that we use ILP score as the user’s expertise score (See Section 6). However, OLP does not represent anything in this graph (as it only means that the individual has asked a large number of questions that gets translated to incoming edges when those questions are answered by someone)\(^2\). In the second application (bipartite graph), we apply this approach to identify experts (based on the number of accepted answers! However, it is possible to include quality information if weights representing quality are assigned to edges.

\(^2\)Note that this formulation does not account for the quality of answers but only the number of answers! However, it is possible to include quality information if weights representing quality are assigned to edges.
papers in a conference) in an author-paper graph. The user’s expertise score is described as the probability of this user’s paper to be accepted in an ICDM conference (this author-paper graph) and hence we use OLP score as the user’s expertise score (See Section 6) in this application. Again, ILP does not have any significance here. These two applications clearly demonstrate the relevance of ILP and OLP based on application semantics and which one needs to be used and why!

**Contributions:**

- This paper analyzes the ranking problem in a graph. Rather than random traversal (or hubs and spokes), we provide an alternative intuition as to why links in graphs represent qualitative information.
- We propose the concept of ILP and OLP in a graph and use the non-negative matrix factorization (NMF) to calculate ILP and OLP vectors for a graph using its connectivity information. We prove that hub and authority vectors of HITS have the same rank order, respectively, as ILP and OLP vectors. We also argue that HITS vector is the rank-1 approximation of adjacency matrix $L$.
- We demonstrate how the concept of ILP and OLP can be applied to diverse real-world applications (graphs of different characteristics). The experimental results validate the relationship between HITS and ILOD.

**Road Map:** Section 2 represents related work. Section 3 defines the graph models and Section 3.1 describes problem statement. Section 4 defines the probability model on the graph. Section 5 represents our contributions along with the ILOD algorithm and the relationship between ILOD and HITS. Section 6 shows experimental results for different applications and their analysis. Section 7 has conclusions.

## 2 Related Work

We categorize existing work related to our problem into two main categories: ranking approaches and non-negative matrix factorization.

**Ranking Approaches:** PageRank[4] and HITS[13] algorithms are the most widely used approaches to measure a web page’s quality for search. PageRank is an iterative algorithm and in each iteration PageRank simulates a web user randomly surfing the web page. The final PageRank score of a web page describes the probability of this surfer visiting a particular web page. HITS algorithm models the web as two types of pages: hubs and authorities. Hubs are web pages that link to many authoritative pages and authorities are web pages that are linked to by many hub pages. HITS is also an iterative algorithm that updates the hub and authority score of a page based on the scores of pages of its neighboring web page. These two algorithms are also widely applied to a number of applications. For example, the graph generated by a collection of emails describes the email communication relationship between users. Campbell et al. [6] and Dom et al. [15] apply HITS and in-degree approaches to a
synthetic graph as well as a small email graph to rank correspondents according 
to their expertise on subjects of interest. In their experiments, HITS authority 
score shows higher accuracy than that of the in-degree algorithm. PageRank 
and HITS are also applied to the ask-answer graph [8, 16] to measure the users’ 
expertise score and these two algorithms provide more accurate results than 
other algorithms. In addition, PageRank and HITS algorithms have also been 
extended to bring node’s order to the biological graph [14], paper co-citation 
graph [5] and other social graphs. Although these rank algorithms are widely 
used for various applications, the intuition behind these rank algorithm for these 
applications is not as clear. This paper revisits HITS algorithm for social graphs 
and provides an alternative intuition of HITS vectors.

Non-negative Matrix Factorization: Non-negative Matrix Factorization 
(NMF) has been widely studied in the data mining and machine learning areas 
since the initial work of Lee and Seung [17]. It has been applied to a number of 
different areas such as pattern recognition [18], multimedia data analysis [19], 
text mining [20], and DNA gene expression analysis [21]. Extensions of NMF 
have also been developed to accommodate various cost functions as needed in 
different data analysis problems, such as classification [22], collaborative filter-
ing [23] and clustering [24]. In this paper, we will explore the standard NMF 
approach to solve the rank problem in a social graph context and to the best 
of our knowledge, this is the first paper to apply the NMF approach for the 
social graph rank problem. Extensions to this approach by using different cost 
functions to improve rank accuracy are also possible.

In this paper, we establish a relationship between the HITS algorithm and 
the vectors of non-negative matrix factorization. We also show theoretically 
that the HITS hub and authority scores corresponding to the vectors obtained 
by non-negative matrix decomposition. This relationship provides another way 
of understanding the scores computed by the HITS algorithm and provides an 
alternative intuition for many social network problems.

3  Graph Model and Problem Statement

This research focuses on two kinds of graphs: directed and bipartite. The edges 
of a graph can also have weights to reflect preferences, quality etc. For the 
current discussion, we will not be considering them.

Directed Graph: In many applications, objects and relationships are modeled 
as a directed graph \(G = (V, E)\) where each vertex in \(V\) represents an object in a 
particular domain and an edge in \(E\) describes the relationship between objects. 
For example, Figure 1 shows a directed graph that describes the web reference 
graph extracted from Stanford web site (http://www.stanford.edu/). Each 
node is a web page and a directed edge is drawn from web page \(a\) to \(b\) if page \(b\)'s 
URL is used by page \(a\). In addition, we use the adjacency matrix \(L\) to store the 
graph connectivity. The adjacency matrix \(L\) of Figure 1 is shown in Table 2.
Bipartite Graph: In some applications, objects and relationships are modeled as a bipartite graph \( G = (U \cup V, E) \) where nodes can be divided into two disjoint groups \( U \) and \( V \) such that no edge connects the vertices in the same group and an edge in \( E \) describes the relationship between objects in different groups. For example, Figure 2 is a bipartite graph extracted from the ICDM conferences papers (http://www.informatik.uni-trier.de/~ley/db/conf/icdm/). \( U \) is a set of authors and \( V \) is a set of ICDM papers. An edge is drawn from author \( u_1 \) to paper \( p_1 \) if author \( u_1 \) publishes a paper \( p_1 \) in ICDM. We use the biadjacency matrix \( L \) to store the graph structure. The biadjacency matrix \( L \) of Figure 2 is shown in Table 3.

### 3.1 Problem Statement

The global rank of a web page can be deemed as the probability of this web page to be chosen as the quality web page to explain anchor text. In other words, the global rank of a reference graph can also be considered as the probability of this web page to have a link from other web pages. Moreover, in the author-paper graph, researchers are interested in identifying the expert among all authors. User’s expertise score can also be deemed as the probability of this user’s paper to be accepted in ICDM conference (is selected as the author node). In other words, the user’s expertise rank can also be considered as the probability of this user to create a link to the papers. Therefore, these two probabilities (out-link denoted as OLP and in-link denoted as ILP) can be used to describe the semantics of the graphs of diverse real world-applications.

Given a graph \( G \) with nodes and edges, our research focuses on computing the in-link and out-link probabilities of each node that represents the connec-
tivity of that graph. This allows one to understand the characteristics of the graph as well as choose appropriate algorithms for analyzing the graph.

4 Graph Characteristics

In this section, we first define three types of probability that can be associated with a graph and then discuss the relationship among them.

4.1 Types of Edges

Consider that we have a bag which contains all the edges of graph $G$ (See Figure 3). We have three types of edges in this bag for this sample space: (i) Out-Link($a$) Edges: is an edge in which $a$ is the start node, (ii) In-Link($b$) Edges: is an edge in which node $b$ is the end node, and (iii) Link($a,b$) edge: is an edge $<a, b>$ which is from node $a$ to node $b$. Each type is a bag in the most general sense.

![Figure 3: A Sample Space of Figure 1 (18 edges)](image)

We can now define probabilities associated with edges as follows.

**Definition** $P$(Out-Link($a$)), also called as the Out-Link Probability (OLP) of node $a$, is the probability of node $a$ to be connected to other nodes with an outgoing edge.

**Definition** $P$(In-Link($a$)), also called In-Link Probability (ILP) of node $a$, is the probability of node $a$ to be connected from other nodes with an incoming edge.

**Definition** $P$(Link($a,b$)) is the probability of edge $<a, b>$ is the probability of an edge from node $a$ to node $b$. It is computed as the ratio of the number of edges that start with $a$ and end with $b$ to the total number of edges.

All of the above probabilities can be easily obtained from graph $G$. For example, let us select Link($a,b$) from that bag. Let $e$ be the total number of edges in graph $G$. Therefore, if there is one edge in Link($a,b$), $P$(Link($a,b$)) = $\frac{1}{e}$; if there is no link in Link($a,b$), $P$(Link($a,b$)) = $\frac{0}{e} = 0$. Thus, we have the following equation to describe this probability:

$$P(\text{Link}(a,b)) = \frac{\text{number of edges from } a \text{ to } b}{\text{total number of edges}}$$

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0</td>
<td>0.06</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>0.06</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c</td>
<td>0.06</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>0.06</td>
<td>0.06</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>e</td>
<td>0.06</td>
<td>0</td>
<td>0.06</td>
<td>0</td>
<td>0.06</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>f</td>
<td>0.06</td>
<td>0</td>
<td>0</td>
<td>0.06</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>g</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.06</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4: The link probability matrix $LP$ for Figure 1

You need a bag for representing multiples edges between nodes. They can be differentiated, for example, by assigning numbers to each edge.
\[ P(\text{Link}(a, b)) = \begin{cases} 
0, & \text{(no link between node } a \text{ and node } b) \\
\frac{1}{e}, & \text{(1 edge in the bag Link}(a, b)\text{)}
\end{cases} \]  

(1)

Table 4 shows the probability of links between two nodes for Figure 1. We term this matrix as \textbf{Link Probability matrix} \( LP \).

### 4.2 Relationship among Three probabilities

\( \text{Out-Link}(a), \text{In-Link}(a), \text{and Link}(a, b) \) are bags. It is also evident that if and only if both \( \text{Out-Link}(a) \) and \( \text{In-Link}(b) \) is non-empty, then \( \text{Link}(a, b) \) may be non-empty. Therefore, we have the following:

\[
\text{Link}(a, b) = \text{Out} - \text{Link}(a) \cap \text{In} - \text{Link}(b) \quad \text{and} \quad P(\text{Link}(a, b)) = \frac{|\text{Link}(a, b)|}{e} \quad \text{(2)}
\]

The relationship between \( \text{Out-Link}(a) \) and \( \text{In-Link}(b) \) from node \( a \) (or the edge \( \text{Link}(a, b) \)) has an associated semantics. The presence of an edge (i.e., \( \text{Link}(a, b) \)) indicates an an explicit association between the two nodes either as an answer (in an ask-answer graph of example i), a road relationship (in a road map graph of example ii), an authorship (in the bipartite graph of example iii) etc. This explicitly indicates the reasoning for the presence of this link from a specific node \( a \) to a specific node \( b \).

We ignore the semantic meaning of these links\(^4\) as is done in the literature. However, these two events, \( \text{Out-Link}(a) \) and \( \text{In-Link}(b) \), are not independent. The relationship between the two can be captured by vectors whose (matrix) multiplication results in the results in the link probability.

### 5 ILOD Approach

#### 5.1 Motivation

Figure 4 highlights the motivation to calculate \( P(\text{Out-Link}(a)) \) and \( P(\text{In-Link}(b)) \). Vector \( u \) (also called OLP vector) and \( v^T \) (also called ILP vector) are two vectors to store \( P(\text{Out-Link}(*)) \) and \( P(\text{In-Link}(*)) \) score for each node. We expect that \( uv^T \) can match well with link probability matrix \( LP \) (\( LP = \frac{1}{e}L \), see Figure 4). We also assume that the noise data in a graph follows the normal distribution [17], the non-negative vectors \( u \) and \( v^T \) can be obtained by solving the following optimization problem.

\textbf{OLP and ILP problem: Given a link probability matrix } \( LP \) (\( \frac{1}{e}L \)), \textbf{find non-negative vectors } \( u \) \textbf{ and } \( v^T \) \textbf{ to minimize the function}

\[
\frac{1}{2}||LP - uv^T||_F^2
\]

which is the typically used mean square error. Other metrics have also been used for this purpose[17]. The product \( uv^T \) is called a non-negative vector factorization of \( \frac{1}{e}L \), although the product \( uv^T \) is not necessarily equal to the link factorization of \( \frac{1}{e}L \).

\(^4\)Recalled that these link-based algorithms, such as PageRank and HITS ignore the semantic meaning of these links.
probability matrix \( LP \) (equal to \( \frac{1}{2}L \)). Clearly, the product \( uv^T \) is an approximate factorization of rank at one.

Figure 4 also shows the results of matrix decomposition of Figure 1. In this example, since node \( a \) has the highest number of out links (equal to 5), web page \( a \) has a highest probability to create a link to the other nodes (\( u(a) = 0.43 \)); since node \( b \) has the lowest number of in-links (equal to 1), web page \( b \) has the lowest probability to create a link to the other nodes (\( u(b) = 0.07 \)). Similarly, ILP vector \( v^T \) can be explained in the same way.

5.2 ILP and OLP Decomposition for Adjacency Matrix \( L \) (ILOD)

The multiplicative update rules for \( u \) and \( v^T \) in the gradient descent mechanism of Lee and Seung [17] change when the cost function 3 is minimized. \( u^k \) gives a vector of OLP for all nodes on the \( k \)th iteration and \( u^k \) describes the \( i \)th element of vector \( u^k \). Similarly, \( (v^T)^k \) gives a vector of ILP for all nodes on the \( k \)th iteration and \( (v^T)^k \) describes the \( i \)th element of vector \( (v^T)^k \). We successively compute \( u^{k+1} \) and \( (v^T)^{k+1} \) based on \( u^k \) and \( (v^T)^k \). Notice that Lee et al. [25, 17] indicate that Nonnegative Matrix Factorization does not have a global solution (a unique solution). Since solving the standard NMF objective function \( (\min ||LP - v^T u||_F^2, \ s.t. \ LP, v^T, u \geq 0) \) is NP-hard problem, Lee et al. [25, 17] use the gradient descent approach to approximately calculate the matrix \( v^T \) and \( u \). Therefore, with a differently initialized \( v^T \) and \( u \), algorithm may converge to a different solution. Usually, researchers typically initialize the matrix with random values, run the algorithm several times and choose the highest accuracy result. However, OLP and ILP decomposition problem, being a rank-1 NMF problem, has a global minimizer (a unique solution, See Lemma

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0</td>
<td>0.06</td>
</tr>
<tr>
<td>b</td>
<td>0.06</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c</td>
<td>0.06</td>
<td>0.06</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>e</td>
<td>0.06</td>
<td>0</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>f</td>
<td>0.06</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.06</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>g</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.06</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ L \]

\[ v^T \]

\[ \approx \]

\[ u \]

\[ 0.08 \]

\[ 0.11 \]

\[ 0.13 \]

\[ 0.14 \]

\[ 0.08 \]

\[ 0.04 \]

\[ 0.05 \]
Thus, \( u \) and \( v^T \) will converge to the same vectors with any initialized value (excluding \( u^0 = 0 \) and \( (v^T)^0 = 0 \)). We start with \( u^0 \) and \( (v^T)^0 \) which contain all ones:

\[
\begin{align*}
&u^0 = 1 \\
&v^T = 1
\end{align*}
\]

To compute \( u^{k+1} \) and \( (v^T)^{k+1} \) from \( u^k \) and \( (v^T)^k \), we, respectively, use the following equations:

\[
\begin{align*}
&u^{k+1}_i = u^k_i \left( \frac{(LPv^k)_i}{(u^k(v^T)^k)_i} \right) \\
&(v^T)^{k+1}_i = (v^T)_i \left( \frac{(u^T)^k(LPv^k)}{(u^T)^k(u^k(v^T)^k)_i} \right)
\end{align*}
\]

Algorithm 1 outlines the process to calculate the OLP and ILP vectors. It takes in 1 argument link probability matrix \( LP \). In line 1-2, algorithm first initializes variables and sets \( u \) and \( v^T \) vectors as unit vectors. Line 3 is used to stop this iterative algorithm. Although the convergence of iterative non-negative matrix decomposition can be guaranteed in theory (See [17]), practical computation uses a maximum number of iterations (say, \( K \)). In all of our experiments we have seen rapid convergence, which relative rank score stabilizing in 200 iterations. Hence, we have fixed the number of iterations (\( K \)) to 200.

We also analyze the time and space requirements needed for this approach. Because the graph extracted from the real applications is very sparse, we only store the edges; therefore, the space required is only \( O(e) \) where \( e \) is the number of edges in this graph. Let \( n \) be the number of nodes in this graph. The time complexity of ILOD is \( O(Kn^2) \) because in each iteration this algorithm calculates \( LP \times v \) or \( u^T \times LP \) which needs \( O(n^2) \). In fact, ILOD algorithm has the same time and space complexity as the HITS algorithm.

### 5.3 Relationship between HITS and ILOD

In this section, we provide some theoretical analysis of HITS and ILOD.

**Lemma 1** \( LP \) being the link probability matrix of graph \( G \), the pair \( u \) and \( v^T \) are local minimizers of cost equation \( \frac{1}{2} \|LP - uv^T\|_F^2 \) if and only if \( u \) and \( v^T \) are the principle eigenvectors of \( LL^T \) and \( L^T L \) respectively.
Proof See Appendix A.

Based on the above, we establish the relationship between ILOD and HITS algorithm.

**Theorem 1** Given a probability matrix $LP$, the OLP vector $u$ obtained by the NMF algorithm corresponds to the hub vector of the HITS algorithm and the ILP vector $v^T$ obtained by the NMF algorithm corresponds to the authority vector of the HITS algorithm.

**Proof** Kleinberg [13] has indicated that hub rank vector of HITS algorithm is the principle eigenvector of $LL^T$ and the authority rank vector of HITS algorithm is the principle eigenvector of $L^T L$.

According to Lemma 1, OLP vector $u$ is the hub vector of HITS algorithm and ILP vector $v^T$ is the authority vector of HITS algorithm.

6 Experimental Validation

In this section, we apply NMF approach to a number of graph types corresponding to two different applications (corresponding to two types of graphs), illustrating and validating the relationship between HITS and ILOD.

6.1 Directed Graphs

In this application, we first focus on identifying experts from Community Question Answering services (or CQAs). Identifying expertise from a CQA data set is useful in many ways: i) allows one to intrinsically rank (or group) users in the community, ii) this can be beneficially used for identifying good answers, and iii) the CQA service can keep them by providing incentives and route questions to these experts for delivering better answers. Other approaches have been used for this purpose. For example, in order to calculate user’s expertise score, Zhang et al. [8] and Jurczyk et al. [16] extracts the ask-answer graph from these CQAs. Nodes represent users in the QA community and a directed edge is drawn from user $u_1$ to user $u_2$ if user $u_2$ answers one or more questions asked by $u_1$. Table 5 shows a few questions and some of their answers from the Stack Overflow service for the “C” language. Figure 5 shows the ask-answer graph for Table 5 using the ask-answer paradigm.
Table 6: Complete Data set Characteristics

<table>
<thead>
<tr>
<th>Data set</th>
<th>#Ques</th>
<th>#Answers</th>
<th>#Answerers</th>
<th>#Questioners</th>
</tr>
</thead>
<tbody>
<tr>
<td>SO-O</td>
<td>8,644</td>
<td>21,879</td>
<td>4,279</td>
<td>5,722</td>
</tr>
</tbody>
</table>

User Votes Content

| Questioner A | 330 | In C arrays why is this true? a[5] == 5[a] |
| Answerer B   | -6  | Because a[5] will evaluate to: *(a + 5) and 5[a] will evaluate to: *(5 + a) |
| Questioner B | 144 | What is the best tool for creating an Excel Spreadsheet with C#? |
| Answerer C   | 10  | You can use a library called Excel Library. It’s a free, open source library posted on Google Code. |

Table 5: A Sample set of Questions and Answers from Stack Overflow

The user’s expertise score can be described as the probability of this user to answer a question in this CQAs since these CQAs requires us to identify the appropriate user to answer posed questions. Since a directed edge links an asker (a user who asks a question) with an answerer (a user who answers a question), the ILP of user $u_1$ describes a probability of this user $u_1$ to answer the other users’ questions. Therefore, we use ILP score as the user’s expertise score for CQAs. In our experiment, we use Stack Overflow (SO) data set to identify user’s expertise score.

Stack Overflow (SO) data set (http://stackoverflow.com/): This CQAs focuses on computer programming topics. SO allows a user to modify other users answers. In other words, when an answerer wants to answer a question, s/he has two choices: modify an existing answer or provide a new answer. As a result, the average number of answers for each question is only 2.36. In our experiments, we only consider the first user who posts the answer as the answerer, because, in most cases, the first user is likely to provide a significant contribution than other users. Each question in this community is marked with a topic tag (e.g., “Oracle”). We use questions which are marked as “Oracle” as SO-O data set and broader statistical characteristics of this data sets are shown in Table 6.

We compare the HITS (Auth) and ILOD (I) for this ask-answer graph to identify the user’s expertise score and Table 7 shows the top 10 users and their ranks score for ILOD and HITS approaches. In Table 7, if we normalize HITS (Authority) vector and ILOD (ILP) vector, these two vectors match completely. Thus, HITS (Authority) score of a user in this ask-answer graph can be explained as the probability of this user to answer questions.
In this application, we focus on identifying experts from an author-paper graph. This application is similar to some retrieval systems, such as Arnetminer system (http://arnetminer.org) and LinkedIn system (http://www.linkedin.com/). Our ranking model is easy to extend to the bipartite graph.

We use the ICDM data set to represent this type of application. The ICDM data set (http://www.informatik.uni-trier.de/~ley/db/ICDM) contains all the regular and short papers that appeared in the ICDM conference from 2001 to 2011. We extract author-paper graph from these papers which includes 806 papers, 1820 authors and 2625 relationships between papers and authors. The short and regular papers are set as the same weight in this author-paper graph.

For Figure 2, The $u$ vector describes the probability of each user to write a paper and the $v^T$ vector describes the probability of this paper to be written by a user. The probability of a user to write a paper can be used to measure the user’s expertise score because a good expert always publishes research papers in the good conference. In other words, we can intuitively understand that an expert will have authored more papers in the author-paper graph. hence, we...
Table 9: Top 10 Experts from the ICDM Conference (2001 to 2011)

<table>
<thead>
<tr>
<th>ILOD (O)</th>
<th>Score</th>
<th>HITS (Hub)</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zheng Chen</td>
<td>0.025</td>
<td>Zheng Chen</td>
<td>0.57</td>
</tr>
<tr>
<td>Jun Yan</td>
<td>0.021</td>
<td>Jun Yan</td>
<td>0.47</td>
</tr>
<tr>
<td>Lei Ji</td>
<td>0.016</td>
<td>Lei Ji</td>
<td>0.36</td>
</tr>
<tr>
<td>Shuicheng Yan</td>
<td>0.015</td>
<td>Shuicheng Yan</td>
<td>0.35</td>
</tr>
<tr>
<td>Junshi Huang</td>
<td>0.014</td>
<td>Junshi Huang</td>
<td>0.30</td>
</tr>
<tr>
<td>Ning Liu</td>
<td>0.010</td>
<td>Ning Liu</td>
<td>0.24</td>
</tr>
<tr>
<td>Ying Chen</td>
<td>0.004</td>
<td>Ying Chen</td>
<td>0.10</td>
</tr>
<tr>
<td>Zheng Chen</td>
<td>0.003</td>
<td>Zheng Chen</td>
<td>0.06</td>
</tr>
<tr>
<td>Siyu Gu</td>
<td>0.002</td>
<td>Siyu Gu</td>
<td>0.00</td>
</tr>
<tr>
<td>Qiang Yang</td>
<td>0.002</td>
<td>Qiang Yang</td>
<td>0.00</td>
</tr>
</tbody>
</table>

use the OLP vector $u$ to identify an author as an expert in this graph. Similarly, HITS (Hub) can be used to identify an author as an expert in this graph.

We compare HITS (hub) and OLP vector in our experiments. Table 9 shows, respectively, the top 10 experts and ranks score for ILOD and HITS approaches. In Table 9, if we normalize HITS (Hub) vector and ILOD (OLP) vector, these two vectors are the same. Thus, HITS score of author-paper graph can be explained as the probability of an author getting his paper accepted in this conference so that this score can be used as the expertise score.

7 Conclusions

In this paper, we have discussed the ranking problem in web and social networks. We have proposed ILP and OLP of a graph to help understand HITS approach in contexts other than web graphs. We have established that the two probabilities identified in this paper correspond to the hub and authority vectors of the HITS approach. In this paper, we have used the standard non-negative matrix decomposition approach to calculate these two probabilities for each node. Then, we have proved the relationship between HITS vectors and ILP/OLP vectors. In addition, we have applied ILOD to different types of graphs representing applications other than the web graph. Experimental results validate the relationship between ILOD approach and HITS algorithm. This paper provides an alternative intuition for the use of HITS (or NMF) approach which explains the relevance of either the hub vector or the authority vector or both.

8 Appendix A

In this section, we prove Lemma 1.

Lemma 1 LP being an probability matrix of graph $G$, the pair of vectors $u$ and $v^T$ are local minimizer of cost equation \( \frac{1}{2}||LP - u v^T||_2^2 \) if and only if $u$ and $v^T$ are the non-negative eigenvectors of $LL^T$ and $L^T L$ respectively.

Proof Since $LP = \frac{1}{2}L$, $LP$ and $L$ have the same non-negative matrix decomposition vector $u$ and $v^T$. We just need to prove the pair of vectors $u$ and $v^T$
are local minimizer of cost equation $\frac{1}{2}\|L - uu^T\|^2_2$ if and only if $u$ and $v^T$ are the non-negative eigenvectors of $LL^T$ and $L^TL$.

(1) The necessary condition.

Without loss the generality, the vectors $u$ and $v^T$ are respectively partitioned as $(u \ 0)$ and $(0 \ v^T)$ and the adjacency matrix $L$ is partitioned as follows:

\[
L = \begin{pmatrix}
L_{11} & L_{12} \\
L_{21} & L_{22}
\end{pmatrix}
\]  

(4)

To keep gradient descent, Lee et al. [17] indicate that:

\[
uv^Tv - Lv \geq 0, \ \nu uu^T - L^Tu \geq 0
\]  

(5)

Then, we have

\[
\begin{pmatrix}
0 & 0 \\
0 & L_{22}
\end{pmatrix}
\begin{pmatrix}
L_{11} & L_{12} \\
L_{21} & L_{22}
\end{pmatrix}
\begin{pmatrix}
0 \\
u^T
\end{pmatrix}
\geq 0
\]  

(6)

and

\[
\begin{pmatrix}
0 & 0 \\
0 & L_{11}
\end{pmatrix}
\begin{pmatrix}
L_{11} & L_{12} \\
L_{21} & L_{22}
\end{pmatrix}
\begin{pmatrix}
0 \\
u
\end{pmatrix}
\geq 0
\]  

(7)

Then, we have $L_{21}v^T \leq 0, L_{12}^T \geq 0, u(||v||^2u - L_{11}v) = 0$ and $v^T(||v||^2u - L_{11}u) = 0$. Since $L_{21}, L_{12}, u, v^T \geq 0$, we have:

\[
||u||^2||v||^2u = L_{11}L_{11}^Tv^Tu
\]  

(8)

and

\[
||u||^2||v||^2v^Tv = L_{11}^TL_{11}v
\]  

(9)

Thus, $u$ and $v^T$ are, respectively, the eigenvector of $LL^T$ and $L^TL$.

(2) The sufficient condition.

Let $R_{+}^{m \times n}$ be the set of $m \times n$ non-negative matrices and $R^{m \times n}$ be the set of $m \times n$ real matrices. $L \in R_{+}^{m \times n}(m \geq n)$ has a singular value decomposition:

\[
L = UV^T
\]  

(10)

where $U \in R^{m \times m}$ and $V \in R^{n \times n}$ are orthogonal matrices and $\Sigma \in R^{m \times n}$ is an rectangular diagonal matrix with $\lambda_1, \lambda_2, ..., \lambda_n$ on the diagonal where $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$ are the singular values of $L$. $u$ and $v^T$ are the non-negative eigenvector of $LL^T$ and $L^TL$.

Then, for matrix rank $r = 1$, the matrix $L_1 = \lambda_1uv^T$ is a global minimizer of the problem.

\[
\min_{uv^T \in R^{m \times n}} \frac{1}{2}||L - uv^T||^2_{l_2}
\]  

(11)

and its error is $\frac{1}{2}||L - uv^T||^2_{l_2} = \frac{1}{2}\lambda_1^2$. 

16
References


