Unsupervised Ranking using Graph Structures and Node Attributes

Chin-Chi Hsu [‡]
chinchi@iis.sinica.edu.tw

Yi-An Lai [†]
b99202031@ntu.edu.tw

Wen-Hao Chen [†]
b02902023@ntu.edu.tw

Ming-Han Feng [†]
b00902001@ntu.edu.tw

Shou-De Lin [†]
sdlin@csie.ntu.edu.tw

[†] Department of Computer Science and Information Engineering, National Taiwan University
[‡] Institute of Information Science, Academia Sinica

ABSTRACT

PageRank has been the signature unsupervised ranking model for ranking node importance in a graph. One potential drawback of PageRank is that its computation depends only on input graph structures, not considering external information such as the attributes of nodes. This work proposes AttriRank, an unsupervised ranking model that considers not only graph structure but also the attributes of nodes. AttriRank is unsupervised and domain-independent, which is different from most of the existing works requiring either ground-truth labels or specific domain knowledge. Combining two reasonable assumptions about PageRank and node attributes, AttriRank transfers extra node information into a Markov chain model to obtain the ranking. We further develop approximation for AttriRank and reduce its complexity to be linear to the number of nodes or links in the graph, which makes it feasible for large network data. The experiments show that AttriRank outperforms competing models in diverse graph ranking applications.

Keywords

unsupervised learning; node ranking; PageRank

1. INTRODUCTION

The ranking of nodes based on their importance is widely used in applications such as web search and social network. PageRank [14] is a well-known unsupervised solution for such problem. It assumes that a highly-ranked node is more likely to be pointed to by other highly-ranked nodes. Based on the assumption, PageRank gives each node a ranking score by modeling itself using the Markov chain framework, to obtain the unique converged ranking result. Formally, PageRank runs the update rule:

$$\pi^{(t+1)} = (1-d)\frac{1}{N}\mathbf{1} + dP\pi^{(t)}$$

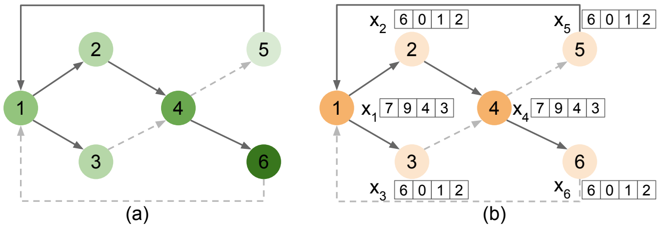


Figure 1: On figure (a), node 1 and 4 should obtain identical PageRank values, and so are nodes 5 and 6, assuming their edges (3,4) and (4,5) are not missing. However, if those links are missing, then with the attribute values as shown in (b), it is possible to recover the similarity of rankings between those nodes.

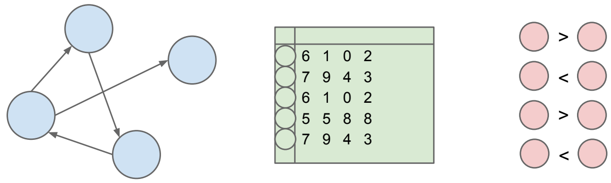
where vector π denotes the ranking scores of all N nodes, P denotes the transition matrix from graph edges, $\mathbf{1}$ is the vector with all elements equal to 1, and d is the damping factor which is recommended to be set to 0.85. PageRank updates π for sufficient iterations to approach the converged output.

One potential drawback of PageRank is that only the graph information is considered in the ranking. Relying only on the graph information is problematic since in real world data the graph information might contain errors or missing evidence due to data collection biases. For instance, in a social network, friendship connections are very likely to be missing, which can affect the ranking results of a PageRank-based model. Nonetheless, the nodes in a graph are very likely to contain certain profile information or attributes. Take social networks as an example, we can obtain some extra information (e.g. name, demographic features) about the nodes or persons, and it is reasonable to assume that information (e.g. job title) can be used to boost the ranking performance. Such observation motivates our research aiming to create a better yet still unsupervised ranking model that considers both graph structure and external node attributes.

We take Figure 1 as an example. Assume that the link between node 3 and node 4 is missing while collecting the graph information. In a regular PageRank model, the importance of node 1 cannot be fully propagated to node 4. However, the similarity of attributes between node 1 and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WSDM 2017, February 06-10, 2017, Cambridge, United Kingdom
© 2017 ACM. ISBN 978-1-4503-4675-7/17/02...$15.00
DOI: http://dx.doi.org/10.1145/3018661.3018668



6	1	0	2
7	9	4	3
6	1	0	2
5	5	8	8
7	9	4	3

(a) Graph (b) Node attributes (c) Ranking labels

Figure 2: Possible training data of graph-based ranking models. Previous works use either (a), (a + c) or (a + b + c), while AttriRank utilizes only (a + b).

node 4 could provide a hint that the ranking of these two nodes should be closer.

Another potential issue for PageRank is that the ranking score of a node depends only on the score of its neighbors due to the 1st-order Markov assumption. In other words, PageRank models itself with node adjacency relations and does not directly consider non-adjacent nodes. However, there exists other information from the graph that is highly relevant to a node's importance. To name a few: the total number of 2nd-degree neighbors of a node; or the total number of shortest paths traversing through this node. Our model allows the modeling of such complex graph information as attributes attaching to the nodes and yields a more trustable ranking that considers more relevant information. Since data outside graphs are not always available, making use of graph attributes is another direction to enhance PageRank.

We design an *unsupervised* learning framework, *AttriRank*, to improve PageRank performance by transferring the knowledge of node attributes. Given arbitrary node attributes, we propose a random walk-based framework to integrate graph and attribute information for ranking. Such unsupervised ranking problem has gain significant attention these years. For instance, WSDM Cup 2016 focuses on unsupervised ranking of authors on an academic graph. Unfortunately, the solutions from the winners in the competition are designed specifically for academic graph rather than general graphs.

As Figure 2 illustrates, related works incorporating attribute information try to fit given ranking labels. The supervised framework might not be as effective since creating supervised ranking ground truth is itself a challenging task. To elaborate, since the goal is to rank nodes based on the graph and the node attributes, human annotators have to look into these two types of information and provide the ranking for the nodes. When the graph size grows, it would be extremely hard for human to generate faithful and unbiased rankings. That motivates us to work on unsupervised ranking in such problem.

Overall, our technical contributions include:

1. Exploiting both link structures and node attributes, we propose an unsupervised learning framework AttriRank to improve the quality of node importance ranking.

2. We address the efficiency problem by proposing two approximation tricks based on eigenvectors and Taylor expression, which brings the time complexity linear to the number of nodes. Experiments report no significant performance difference between the exact and approximate AttriRank.

3. Theoretical justifications are provided for the selection of AttriRank parameters. It is essential as there is no labeled validation set available to tune parameters in unsupervised learning.

We release our source code of AttriRank in our website [1].

2. RELATED WORK

The earliest solutions to rank nodes of a network might be the Centrality-based metrics [7] , such as closeness and betweenness centrality derived in the field of social network analysis. A few years later, PageRank [14] and HITS [12] were proposed. PageRank gives ranking by computing the mixture of authority score and reset probability while HITS considers authority scores and hub scores separately. Weighted PageRank [21] introduces weighted transition matrix where each entry in the matrix is proportional to the number of inlinks and outlinks of the innode. N-step PageRank [23] is another derivative of PageRank, which replaces the original transition matrix with a matrix whose entry is proportional to the in-node's N-step neighbor count. All the models mentioned above mainly focus on the link structure of the graph, ignoring other information embedded in the graph or the external attributes which are the focus of this paper.

Recently, some semi-supervised PageRank-related methods have been proposed, for example, TrustRank [10] , Adaptive PageRank [18] and Semi-supervised PageRank [8]. TrustRank focuses on detecting spam websites. It first computes hub score for each node. Nodes with top k scores form a seed set. Human experts evaluate these web pages as spam or non-spam. The result forms the initial score vector and then a PageRank-based method is used to produce the final rank. TrustRank is simple but heuristic: scores of human labeled websites propagate to other unlabeled websites. A website that is farther from a non-spam website receives a lower score than a website closer to it. Adaptive PageRank is an enhancement of PageRank. It transforms the original PageRank formula into an objective function which minimizes the norm-2 distance between the optimal PageRank score under certain constraints and the original PageRank. The constraints, for example, can be pairwise preferences in websites. Semi-Supervised PageRank further expands Adaptive PageRank. It is also a general version of many PageRank-like algorithms such as NetRank [1], LiftHITS [6] and Laplacian Rank [25, 2, 16, 20]. It includes node features and edge features into the objective function. Node feature vector is included in the reset probability matrix and edge feature vector is included in the transition matrix. In addition, there is a weight vector for all node feature vectors and a weight vector for all edge feature vectors. Both weight vectors and the rank for each node can be learned during the optimization process. Nevertheless, these methods are either supervised or semi-supervised and have to rely on labels to adjust their scores.

[1] https://github.com/ntumslab/AttriRank

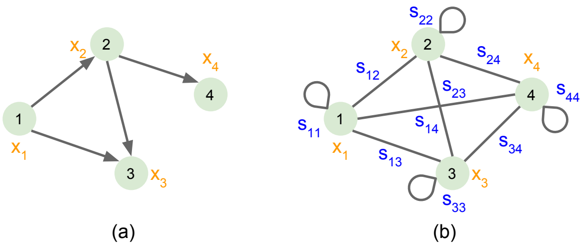


Figure 3: Example of two graphs involved in AttriRank. (a) Graph G with nodes $\{1, 2, 3, 4\}$ is given as input. The nodes have attribute vectors $\{x_1, x_2, x_3, x_4\}$. (b) Graph H is an imaginary, fully connected graph with the same set of nodes. Each edge (i, j) is assigned a positive similarity weight s_{ij}, including self-loops.

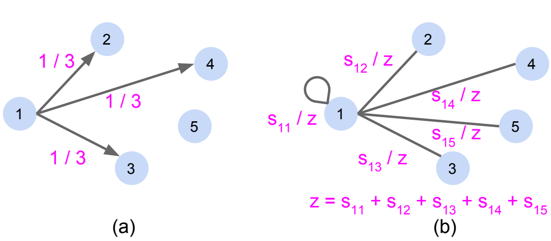


Figure 4: Random walk interpretation of AttriRank. Consider a case with node 1 linking nodes $\{2, 3, 4\}$, but not $\{5\}$ in graph G. (a) In graph G, like PageRank, a walker at node 1 moves to one of the direct successors uniformly at random. (b) In graph H, a walker at node 1 chooses one node out of all nodes with probability proportional to similarity weights.

3. METHODOLOGY

3.1 Problem Definition

We are given an N-node graph, denoted by $G = (V, E)$, as input to our model. V and E denote the set of nodes and edges respectively. Each node $i \in V$ is associated with a K-dimensional attribute vector x_i. Matrix $X = [x_1 x_2 \dots x_N] \in \mathbb{R}^{K \times N}$ denotes all the node attribute vectors. Our goal is to build an unsupervised extension of PageRank that, with the help of node attributes, outputs a more reliable ranking score π_i for each i.

3.2 AttriRank

3.2.1 Model Assumptions

Similar to every unsupervised learning algorithm, AttriRank needs to rely on some assumptions for ranking:

1. **PageRank assumption**. A node receives higher ranking if it is linked by many other high-score nodes.
2. **Attribute assumption**. If a pair of nodes (i, j) have similar attribute values $x_i \approx x_j$, then they should receive similar ranking scores $\pi_i \approx \pi_j$.

The PageRank assumption is exactly the same as that of a conventional PageRank model. The attribute assumption is commonly utilized in machine learning. That is, for two instances with similar attributes, their classification or regression outcomes should be similar.

3.2.2 AttriRank Overview

Based on the two assumptions, AttriRank adopts a scenario where a random walker simultaneously moves in two graphs G and H. H is a *fully connected undirected* graph sharing the same node set V of G, as shown in Figure 3. Each edge weight between nodes (i, j) in H represents the similarity $s_{ij} > 0$ between the corresponding node attributes (x_i, x_j). The choice of s_{ij} is discussed in Section 3.2.3. Recall that the random walk in PageRank can be interpreted by a Markov chain model. AttriRank adopts the same interpretation with the following update rule:

$$\pi^{(t+1)} = (1 - d)Q\pi^{(t)} + dP\pi^{(t)}, \tag{1}$$

where

$$P_{ij} \equiv \begin{cases} \frac{1}{\delta_j} & \text{if directed edge } (j, i) \in E \\ \frac{1}{N} & \text{if } \delta_j = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$Q_{ij} \equiv \frac{s_{ij}}{\sum_{k \in V} s_{kj}}. \tag{3}$$

Under Markov chain framework, the vector $\pi \in \mathbb{R}^N$ of node ranking scores is modeled as a probability distribution. That is, for each node i, its ranking score $\pi_i \geq 0 \ \forall \ i$ and $\sum_{i \in V} \pi_i = 1$. $P \in \mathbb{R}^{N \times N}$ and $Q \in \mathbb{R}^{N \times N}$ denote the corresponding transition matrices for graph G and H respectively. If G is undirected, an edge (i, j) in G is treated as two directed edges (i, j) and (j, i). δ_j is the number of outgoing links, or out-degree, of node j. Like PageRank, we have to deal with dangling nodes ($\delta_j = 0$) to avoid $\pi = 0$. Parameter $d \in (0, 1)$ controls the random-walk preference ratio between graph G and H. Figure 4 draws the visual interpretation of (1). We repeatedly assign vector π to the right-hand side of (1), until convergence.

AttriRank can be treated as the linear combination of our two proposed assumptions where d determines their ratio. $P\pi$ obeys the PageRank assumption, while $Q\pi$ follows the attribute assumption. Assume that two nodes (i, j) have similar attribute values $x_i \approx x_j$. Then similarity $s_{ik} \approx s_{jk} \ \forall \ k$. Therefore, $\pi_i = \sum_{k \in V} \frac{s_{ik}}{\sum_{l \in V} s_{lk}} \pi_k \approx \sum_{k \in V} \frac{s_{jk}}{\sum_{l \in V} s_{lk}} \pi_k = \pi_j$ as $d \to 0$.

Below we prove the convergence property of AttriRank:

Lemma 1. *AttriRank as a Markov chain model converges to the unique stationary probability distribution, regardless of the initial probability distribution π.*

Proof. It has been known that if a Markov chain model is formulated by an aperiodic and irreducible transition matrix, then it will converge to the unique stationary probability distribution after infinitely many iterative updates. We have the transition matrix of AttriRank $R \equiv (1-d)Q + dP$. Since $d \in (0, 1)$ and $s_{ij} > 0 \Rightarrow Q_{ij} > 0$ by (3), R is ensured to be

- aperiodic. Since $Q_{ii} > 0 \Rightarrow R_{ii} > 0$, a random walker at any state (node) i has non-zero chance to stay at i

forever; hence, each state i is likely to be returned at any particular timestamp.

- irreducible. Any state j transits to any other state i with probability $Q_{ij} > 0 \Rightarrow R_{ij} > 0$.

□

3.2.3 Similarity between Attributes

Previously we have defined graph H with similarity weight s_{ij} for each edge (i, j). s_{ij} represents the scale of similarity between the attributes of two adjacent nodes i and j. Mathematically, Radial Basis Function (RBF) kernel is selected as our similarity definition in AttriRank. That is,

$$s_{ij} \equiv e^{-\gamma \|x_i - x_j\|_2^2} \tag{4}$$

where positive parameter γ controls the influence of attribute distances. RBF kernel has been employed as a similarity definition in various occasions such as [13, 24]. There are three major benefits for us to adopt such similarity measurement:

- $s_{ij} = s_{ji}$. It is the general requirement for any distance metrics, and we will later exploit this property to derive an efficient approximation in Section 3.3.1.
- $0 < s_{ij} \leq 1$. $s_{ij} > 0$ to satisfy Lemma 1. The range between 0 and 1 can let s_{ij} be explained with probability $\Pr(x_i = x_j)$ where $s_{ii} = \Pr(x_i = x_i) = 1$.
- RBF kernel is equivalent to the inner product $\phi(x_i)^T \phi(x_j)$ of two infinitely dimensional vectors projected from x_i and x_j. Thus s_{ij} could catch non-linear similarity between x_i and x_j.

3.2.4 Internal Attributes

Here we propose that the attributes of nodes not only can be derived from external data, but also from internal graph structure information. One potential drawback for the original PageRank algorithm is that it only propagates the near-by node scores due to the 1st-order Markov assumption. More complex structure information such as the sum of degrees of neighbor nodes 2 steps away are not directly considered. AttriRank allows us to directly model such information into attributes to enhance the performance. To distinguish sources of attributes, we call those extracted from the input graph itself as *internal attributes* and attributes unrelated to graph structure as *external attributes*. Our experiments show that both types of attributes can boost the performance. For each node, we pick the following 13 internal attributes and will show their effectiveness in the experiment section: (1) assortativity [2]; (2) in-degree; (3) out-degree; (4-5) the sum and the mean of in-degrees of direct successors; (6-7) the sum and the mean of out-degrees of direct predecessors; (8-10) the number of successors at distance $\{2, 3, 4\}$; (11-13) the number of successors at distance $k \in \{2, 3, 4\}$ divided by the number of successors at distance $k - 1$. To avoid large numerical scale, we take the logarithmic value of all the internal attributes.

3.2.5 Time Preprocessing

Many publicly available graph datasets contain the timestamps of its node being added into the graph. Here we propose a trick to encode time information into our model. Normally it takes time for a new coming node to build its connection. Thus, the importance of cold-start nodes is being overlooked in PageRank. A similar observation is reported by teams competing for WSDM Cup Challenge 2016 [3]. Our preprocessing on attributes is as below:

$$x'_i = \frac{1}{1 + t_i - \min_k t_k} x_i \tag{5}$$

where t_i is the timestamp of node i and $\min_k t_k$ means the earliest time in the graph. With the adjustment of (5), the similarity scores between two cold-start nodes becomes higher, meaning that it is more likely a random walker will surf from one cold-start node to another, raising their PageRank scores.

3.3 Efficient Model Approximation

Since H is assumed to be a fully connected graph, it takes at least $O(N^2)$ time and space to generate matrix Q as well as calculate matrix multiplication $Q\pi$. That says, for large graphs, it is infeasible to compute (1). To address the scalability problem, we propose two approximation tricks in this section.

3.3.1 Surrogate of $Q\pi$

Based on the definition of Q in (3), we discover the following.

Lemma 2. *If we define a N-dimensional vector r where each element r_i is defined as*

$$r_i = \frac{1}{z} \sum_{j \in V} s_{ij} \tag{6}$$

with the normalization term $z = \sum_{i \in V} \sum_{j \in V} s_{ij}$, then we have

$$r = Qr. \tag{7}$$

Proof. For each node i,

$$\begin{aligned}
(Qr)_i &= \sum_{j \in V} \frac{s_{ij}}{\sum_{k \in V} s_{kj}} r_j \\
&= \sum_{j \in V} \frac{s_{ij}}{\sum_{k \in V} s_{kj}} \frac{\sum_{k \in V} s_{jk}}{z} \\
&= \frac{1}{z} \sum_{j \in V} s_{ij} \\
&= r_i.
\end{aligned}$$

The proof requires property $s_{jk} = s_{kj}$ in (4). □

Since Q serves as a stochastic matrix under Markov chain framework, vector r is the stationary probability distribution for Q. In fact r is the corresponding eigenvector for the largest absolute eigenvalue 1 and captures the major transformation direction of Q. Hence, instead of calculating $Q\pi$ for each update, we put vector r into AttriRank:

$$\pi^{(t+1)} = (1 - d)r + dP\pi^{(t)}. \tag{8}$$

We can generate sparse matrix P and vector r using (2) and (6) as a preprocessing step. P and r are kept as fixed during the update process of π, such that we can avoid the calculation on $Q\pi$. The approximate formulation also meets

[2] Degree / Average degree of neighbors

[3] https://wsdmcupchallenge.azurewebsites.net/

our attribute assumption: if an attribute vector pair $x_i \approx x_j$, then $r_i \approx r_j$ such that $\pi_i \approx \pi_j$ when $d \to 0$. Lemma 1 also holds because the transition matrix becomes $R' \equiv (1-d)r\mathbf{1}^T + dP$ where $\mathbf{1}$ is the vector of all elements equal to 1. Since $r\mathbf{1}^T$ consists of all positive elements, we can verify aperiodicity and irreducibility of R' in the same way.

3.3.2 Approximation of r

Despite the replacement of $Q\pi$ with r, the generation of the vector r still takes $O(N^2K)$ due to pairwise similarity computation in (6). To avoid the quadratic time relative to the number of nodes, we first approximate the unnormalized element $\hat{r}_i = \sum_{j \in V} s_{ij}$. Since RBF kernel similarity includes the exponential function, Taylor expression offers an inspiration to eliminate complex computation. As variable $y \to 0$, Taylor expression $e^y \approx 1 + y + \frac{1}{2}y^2 > 0$. $\hat{r}_i$ can be derived as follows:

$$
\begin{aligned}
\hat{r}_i &= \sum_{j \in V} e^{-\gamma \|x_i - x_j\|_2^2} = \sum_{j \in V} e^{-\gamma (\|x_i\|_2^2 + \|x_j\|_2^2 - 2x_i^T x_j)} \\
&\approx e^{-\gamma \|x_i\|_2^2} \sum_{j \in V} e^{-\gamma \|x_j\|_2^2} \left(1 + 2\gamma x_i^T x_j + \frac{1}{2}(2\gamma x_i^T x_j)^2\right) \\
&= e^{-\gamma \|x_i\|_2^2} \Bigg[\sum_{j \in V} e^{-\gamma \|x_j\|_2^2} + x_i^T \left(2\gamma \sum_{j \in V} e^{-\gamma \|x_j\|_2^2} x_j\right) \\
&\quad + x_i^T \left(2\gamma^2 \sum_{j \in V} e^{-\gamma \|x_j\|_2^2} x_j x_j^T\right) x_i \Bigg] \\
&\equiv w_i \left[a + x_i^T b + x_i^T C x_i \right].
\end{aligned}
$$

Let $\{w_i \forall i \in V, a, b, C\}$ be the corresponding dummy variables. After obtaining the values of these variables, we can compute $\hat{r}_i$ for each node i and then normalize the entire vector to avoid directly computing z. Each of the above variables requires $O(NK^2)$ computation time at most. Since $N \gg K$ in most real-world large networks, the N-linear time complexity is sufficiently scalable in practice.

A concern lies in the value of $y \equiv 2\gamma x_i^T x_j \ \forall\ i, j$. Taylor expression achieves high approximation accuracy as $y \to 0$. Fortunately, the assumption is reasonable in AttriRank. As long as we apply common *standardization (Z-score)* technique for each dimension of an attribute vector, the mean of these attribute dimensions must be 0 and their variance be 1. Lemma 3 roughly summarizes the consequence of standardization.

LEMMA 3. *Suppose that all elements, as random variables with zero mean and unit variance, in attribute matrix X are independent of each other, then mean $E(y) = 0$ and variance $Var(y) = 4\gamma^2 K$.*

PROOF. For any pair of attribute vectors (x_i, x_j),

$$
\mathrm{E}(y) = \sum_{k=1}^{K} \mathrm{E}(2\gamma x_{ik} x_{jk}) = 2\gamma \sum_{k=1}^{K} \mathrm{E}(x_{ik})\,\mathrm{E}(x_{jk}) = 0
$$

$$
\mathrm{Var}(y) = \sum_{k=1}^{K} \mathrm{Var}(2\gamma x_{ik} x_{jk}) = 4\gamma^2 \sum_{k=1}^{K} \mathrm{E}(x_{ik}^2)\,\mathrm{E}(x_{jk}^2) = 4\gamma^2 K.
$$

□

Here we remark the connection between RBF kernel parameter γ and $\mathrm{Var}(y)$. In real implementation we suggest setting $\gamma = \frac{1}{K}$. Thus by Lemma 3, we have $\mathrm{Var}(y) = \frac{4}{K}$ such that any sample of y is more concentrated toward $\mathrm{E}(y) = 0$ when K is larger. It guarantees the Taylor approximation accuracy.

3.4 Modelling Parameter d

AttriRank has another parameter d that determines the preference of two model assumptions. PageRank consists of a similar parameter called "damping factor", which is usually recommended to be 0.85. However, as shown in our experiments, $d = 0.85$ does not produce the best performance for AttriRank.

Without training labels, it is not possible to adopt the validation technique to choose a suitable d. Here we propose to view $d \in (0, 1)$ as a random variable and model its probability distribution. Then AttriRank returns expectation $\mathrm{E}(\pi)$ over d, which is less sensitive to individual d values. We refer readers to [9] which has a complete investigation of this topic.

At first, we express vector π to be a function of d by (8):

$$
\begin{aligned}
&\pi = (1-d)r + dP\pi \\
&\Rightarrow \pi = (1-d)(I - dP)^{-1} r = (1-d) \sum_{k=0}^{\infty} d^k P^k r.
\end{aligned}
$$

The last equation holds due to Neumann series. Then the expected value of π is written as:

$$
\mathrm{E}(\pi) = \sum_{k=0}^{\infty} \Big(\mathrm{E}(d^k) - \mathrm{E}(d^{k+1}) \Big) P^k r, \tag{9}
$$

where $\mathrm{E}(d^k)$ is the k-th order moment. The right-hand side of (9) is composed of infinitely many terms; hence, we approximate $\mathrm{E}(\pi) \equiv \pi^{(\infty)}$ by iteratively summing the terms:

$$
\begin{aligned}
&\rho^{(0)} = \pi^{(0)} \equiv (1 - \mathrm{E}(d))r \qquad (10) \\
&\rho^{(k)} \equiv (\mathrm{E}(d^k) - \mathrm{E}(d^{k+1}))P^k r = \frac{\mathrm{E}(d^k) - \mathrm{E}(d^{k+1})}{\mathrm{E}(d^{k-1}) - \mathrm{E}(d^k)} P \rho^{(k-1)} \\
&\pi^{(k)} \equiv \sum_{t=0}^{k} \rho^{(t)} = \pi^{(k-1)} + \rho^{(k)}.
\end{aligned}
$$

Since P, r are non-negative variables and $\mathrm{E}(d^k) - \mathrm{E}(d^{k+1}) = \mathrm{E}(d^k(1-d)) > 0$ as $d \in (0,1)$ and $k \geq 0$, all elements in vector $\rho^{(k)}$ are non-negative. As our novel contribution, Lemma 4 ensures that $\pi^{(k)}$ should be summed in the increasing order of k regardless of probability distributions, while [9] derives its convergence rate $\|\pi^{(k)} - \mathrm{E}(\pi)\| \leq \mathrm{E}(d^{k+1})$.

LEMMA 4.

$$
\mathbf{1}^T \rho^{(k)} < \mathbf{1}^T \rho^{(k-1)} \ \forall\ k \geq 1,
$$

where inner product $\mathbf{1}^T \rho^{(k)}$ equals to the sum of elements in $\rho^{(k)}$, $\mathbf{1}$ is the N-dimensional vector with all 1's.

PROOF. First, we have a few observations:

- $\mathbf{1}^T(P^k r) = 1 \ \forall\ k \geq 0$ because $P^k r$ is always a vector of Markov chain probability distribution among N nodes.
- $\Big(\mathrm{E}(d^{k-1}) - \mathrm{E}(d^k)\Big) - \Big(\mathrm{E}(d^k) - \mathrm{E}(d^{k+1})\Big) = \mathrm{E}(d^{k-1}(d-1)^2) > 0$ as $d \in (0,1)$ and $k \geq 1$.

Based on these observations, for all $k \geq 1$,

$$\begin{aligned}
\mathbf{1}^T \rho^{(k)} &= \mathbf{1}^T \left(\left(\mathrm{E}(d^k) - \mathrm{E}(d^{k+1}) \right) P^k r \right) \\
&= \left(\mathrm{E}(d^k) - \mathrm{E}(d^{k+1}) \right) \mathbf{1}^T (P^k r) \\
&< \left(\mathrm{E}(d^{k-1}) - \mathrm{E}(d^k) \right) \mathbf{1}^T (P^{k-1} r) \\
&= \mathbf{1}^T \rho^{(k-1)}.
\end{aligned}$$

□

Finally, we comment on the choice of the probability distribution. If d follows a uniform distribution, then (9) is equivalent to TotalRank [4], with [3] reporting its convergence rate. [9] establishes the probabilistic viewpoint of TotalRank, and then analyzes it with more general beta distribution (uniform = beta($\alpha = \beta = 1$)). In Section 4.2, we conduct experiments to examine AttriRank performance with respect to both distributions.

3.5 Complexity Analysis

Integrating all the proposals in the previous sections, we present the pseudo code of AttriRank as Algorithm 1. The pseudo code is also used for our complexity analyses.

3.5.1 Time

As Section 3.3.2 mentions, implementation of vector r takes overall $O(NK^2)$ time. Using sparse matrix structure, the generation of P requires $O(|E|)$ time only. Since the total number iterations to convergence depend on the input data, here we consider time consumption of one update. Given fixed P and r, it takes $O(|E|)$ time to calculate the most dominant multiplication $P\rho$. The overall complexity is linear to the number of nodes $N = |V|$ or the number of edges $|E|$. Thus, AttriRank is feasible for big datasets.

3.5.2 Space

To store the input data, it takes $O(N + |E|)$ for graph G and $O(NK)$ for attribute matrix X. Sparse matrix P needs $O(|E|)$ space; vector r and π require $O(N)$ only. Hence, the space complexity of AttriRank is also linear to N or $|E|$, suitable for large training data.

4. EXPERIMENTS

4.1 Setup

4.1.1 Datasets

To verify whether our proposed model adapts to various graph-ranking applications, we collect overall four datasets with distinct ranking goals. Among them, one contains external features and we generate internal features for all of them.

- **Webspam** [4]: In 2008, Webspam Challenge competition uses this dataset to evaluate anti-spam ranking systems. A qualified ranking model should give low ranking scores to spam webpages. Beside a large network of 114529 webpages and 1836441 hyperlinks, the competition provides 138 transformed link-based attributes (internal, extracted inside the graph) and 96 content-based attributes (external, extracted outside the graph). The top-ranked solutions in this competition confirmed the effectiveness of these attributes. The dataset contains overall 122 labeled spam webpages and 1933 labeled non-spam webpages.

Algorithm 1 AttriRank

```
Input: Graph $G = (V, E)$ where each node $i \in V$ has attribute vector $x_i$, RBF kernel parameter $\gamma$, tolerance $\epsilon$, distribution parameters for parameter $d$
Output: Ranking score vector $\pi$
Standardize $x_i$ for each $i \in V$
Scalar $w_i \leftarrow e^{-\gamma \|x_i\|_2^2}$ for each $i \in V$
Scalar $a \leftarrow \sum_{j \in V} w_j$
Vector $b \leftarrow 2\gamma \sum_{j \in V} w_j x_j$
Matrix $C \leftarrow 2\gamma^2 \sum_{j \in V} w_j x_j x_j^T$
Scalar $\hat{r}_i \leftarrow w_i(a + x_i^T b + x_i^T C x_i)$ for each $i \in V$
Vector $r \leftarrow \frac{1}{z}\hat{r}$ where $z = \sum_{i \in V} \hat{r}_i$
Generate matrix $P$ by (2)
Vector $\pi \leftarrow \rho \leftarrow (1 - \mathrm{E}(d))r$    ▷ $\frac{1}{2}r$ for uniform distribution, $\frac{\beta}{\alpha+\beta}r$ for beta distribution
$k \leftarrow 1$
while $\|\rho\| > \epsilon$ do
    $\rho \leftarrow \frac{\mathrm{E}(d^k) - \mathrm{E}(d^{k+1})}{\mathrm{E}(d^{k-1}) - \mathrm{E}(d^k)} P\rho$    ▷ $\frac{k}{k+2}P\rho$ for uniform distribution, $\frac{k+\alpha-1}{k+\alpha+\beta}P\rho$ for beta distribution
    $\pi \leftarrow \pi + \rho$
    $k \leftarrow k + 1$
end while
```

- **Hep-Ph** [5]: The dataset includes 34546 papers and 421578 citations from 1993 to 2003. Prior works such as [19] rank paper importance depending on the citation links. There is no external attribute or label in this dataset. We follow [19] which counts the number of citations after the year 2000 as the ground truth for the importance of papers; citations before the year 1999 are kept for model training. Also, we extract 13 internal attributes as described previously. Paper publishing time is used in our preprocessing stage as mentioned in Section 3.2.5.

- **FB Friendship and Wall Post** [6]: The authors in [11] apply a weighted PageRank algorithm for active-user detection on Facebook (FB) activities, including friendship and posts (i.e. individual posts an article on the other's wall), from the city of New Orleans, in 2009. Following the design in [11], we label every user a binary class: a user is justified as an "active user" if the user writes at least one post in next three weeks, and "inactive user" otherwise. The dataset is composed of 63731 users who have total 817090 friendship links and 831401 wall post edges.

  Both friendship and wall post networks share the 14862 positive labels and 48869 negative labels. Each network is extracted 13 internal attributes. A user's joining time is defined as the average timestamp of all the user's posts.

[4] http://chato.cl/webspam/datasets/uk2007/

[5] http://snap.stanford.edu/data/cit-HepPh.html

[6] http://socialnetworks.mpi-sws.org/data-wosn2009.html

4.1.2 Evaluation

In datasets Webspam, FB Friendship and FB Wall Post, we take Area under ROC Curve (AUC) due to their binary ground-truth labels. That says, a better ranking model shall rank the positive labels higher than the negative ones. This evaluation metric is the same as the one used in Webspam competition [11]. For the citation network, since the ground truths are real values instead of binary, we follow the experiments of [19] to use Spearman's rank correlation coefficient as the evaluation metric.

4.1.3 Compared Models

Since AttriRank is an unsupervised ranking model designed for arbitrary graphs, we choose several unsupervised graph ranking models to compare with:

- **PageRank (PR)** [14]: It is our baseline solution to node importance ranking.
- **Closeness and Betweenness Centrality** [7]: Centrality metrics is defined to identify the most important nodes in a social network. Here we compare with two common centralities: closeness and betweenness. Closeness centrality assumes that the most important nodes should have shorter path lengths to the other nodes. Betweenness centrality claims that the most important nodes must be involved in more shortest paths.
- **Semi-Supervised PageRank (SSP)** [8]: To our knowledge, it is a state-of-the-art semi-supervised general graph ranking model. It consists of a supervised component and an unsupervised component. We adopt its unsupervised part as one of our competitors. The objective function of the unsupervised SSP is shown as below:

  $$\arg\min_{\pi\geq 0,\phi\geq 0} \|(1-d)X\phi + dP\pi - \pi\|_2^2,$$

  where $\phi \in \mathbb{R}^K$ is the attribute weight vector. Since SSP accepts non-negative X, we normalize attributes to $[0,1]$ instead of performing standardization. The optimization problem is solved using projected gradient descent, as suggested in the original paper. Note that similar to AttriRank, SSP also exploits the attribute information.
- **Weighted PageRank (WPR)** [21]: Utilizing the in-degree I_i and out-degree O_i information of a node i, WPR determines the edge weights to improve ranking reliability. The update rule of WPR is as follows:

  $$\pi^{(t+1)} = (1-d)\frac{1}{N}\mathbf{1} + dP\pi^{(t)}$$

  $$P_{ij} = \frac{1}{z_j}\frac{I_i}{\sum_{k\in F_j} I_k}\frac{O_i}{\sum_{k\in F_j} O_k},$$

  where F_j is the set of nodes pointed to by node j, and z_j denotes the normalization term for column j.

4.2 Results

We will verify a few hypotheses about AttriRank in this section.

H1: Does AttriRank outperform other unsupervised ranking models? At first, let us compare the performance of experimented models with suggested parameters. For PageRank, SSP and WPR, we follow what the original papers suggested to fix $d = 0.85$. For AttriRank, we suggest d follows Beta($\alpha = 2, \beta = 3$) to generate the expected ranking of nodes as described previously. RBF kernel parameter is set to $\gamma = \frac{1}{K}$, as has shown previously to guarantee the validity of our approximation. We confirm three observations from the results in Table 1 and 2. First, Table 1 shows that AttriRank significantly outperforms other competitors with Webspam dataset while using only external attributes. The performance is further improved when the internal attributes are added. Second, for the other three datasets, AttriRank utilizes internal attributes to outperform the other models significantly. Third, comparing to SSP which models attributes through regression, AttriRank with pairwise attribute similarities is a superior strategy.

We draw Figure 5 to show the model performance with different parameter d. The experiments confirm that regardless of the value of the parameter d, AttriRank almost always performs the best. The best AttriRank performance in Webspam occurs as $d = 0.02$, meaning that the attributes contribute much more significantly than the graph structure. It is reasonable since the real-world spam pages might try to obtain as many connections as possible to raise their PageRank values. Finally, compared to the other attribute-aware solution SSP, AttriRank is relatively insensitive to parameter selections.

H2: Is using external attributes better than internal attributes? Table 3 shows that for Webspam dataset the external attributes seem to be more useful than internal. Nevertheless, combining both of them always yields the best results, regardless of the distribution of the parameter d.

H3: Do the two approximation tricks affect the performance? Due to page limits, we only report the results on Webspam dataset, but similar observations are made on the other datasets. Figure 6 shows that the ranking produced by the approximated solution is almost identical to the ranking without approximation, regardless which d is used. It demonstrates that both tricks have produced approximates fairly close to the true values.

H4: What is a better distribution for d? Table 3 shows two common distribution choices for $d \in (0, 1)$. Without prior knowledge, uniform distribution is the natural choice. The results show that uniform distribution is a better choice than common recommendation $d = 0.85$. Furthermore, Figure 5 shows that the performance of our model seems to follow a right-skewed shape in regards of d. Thus, we tried a right-skewed beta distribution of hyperparameters $\alpha = 2, \beta = 3$. The results confirmed such choice, and eventually we recommend users to use such beta distribution as the default selection for AttriRank.

H5: Is our current selection for RBF Kernel parameter $\gamma = \frac{1}{K}$ appropriate? We examine the performance of different setup of γ in Table 4. We found that $\gamma = \frac{1}{K}$ does bring better performance than extreme selections $\{1, \frac{1}{K^2}\}$. Although our approximation prefers smaller γ, values too small will cause the RBF-kernel similarities between all node pairs to be close to 1, and AttriRank is thus reduced to PageRank where vector $r = \frac{1}{N}\mathbf{1}$. Besides, we notice the competitive alternative $\frac{1}{\sqrt{K}}$, which is also reasonable since according to Lemma 3 the variance than becomes a constant. Finally, though there exist a well-known method [22] to automatically determine γ, we do not implement it due to its inefficient $O(N^2)$ time complexity.

Table 1: Model performance comparison (AUC) using Webspam. (e) denotes using external node attributes; (e)(i) denotes using both external and internal attributes.

PageRank	Closeness	Betweenness	SSP(e)	SSP (e)(i)	WPR	AttriRank (e)	AttriRank (e)(i)
0.553	0.577	0.556	0.558	0.559	0.509	0.659	**0.666**

Table 2: Model performance comparison with internal attributes available.

Dataset	Evaluation	PageRank	Closeness	Betweenness	SSP	WPR	AttriRank
Hep-Ph	Rank Corr.	0.434	0.286	0.445	0.252	0.406	**0.605**
FB Friendship	AUC	0.741	0.674	0.708	0.722	0.730	**0.796**
FB Wall Post	AUC	0.775	0.755	0.765	0.786	0.765	**0.810**

Table 3: AUC of AttriRank with different attribute sets and parameter d choices with Webspam dataset.

Parameter d	Internal	External	In. and Ex.
$d = 0.85$	0.609	0.616	0.619
$d \sim$ Uniform	0.641	0.649	0.654
$d \sim \text{Beta}(\alpha = 2, \beta = 3)$	0.648	0.659	0.666

Table 4: AttriRank with different values in γ . Parameter $d \sim$ Beta$(\alpha = 2, \beta = 3)$; both internal and external attributes are used for Webspam dataset.

γ	1	$\frac{1}{\sqrt{K}}$	$\frac{1}{K}$	$\frac{1}{K^2}$
Webspam	0.626	**0.704**	0.666	0.562
Hep-Ph	0.546	0.601	**0.605**	0.497
FB Friendship	0.741	0.792	**0.796**	0.753
FB Wall Post	0.802	**0.828**	0.810	0.795

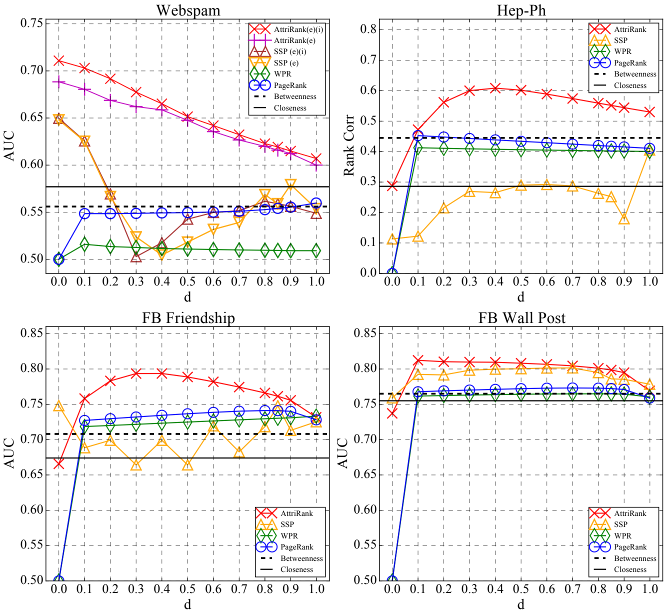


Figure 5: Model performance comparison with different parameters d (i.e. damping factor in PageRank, SSP and WPR).

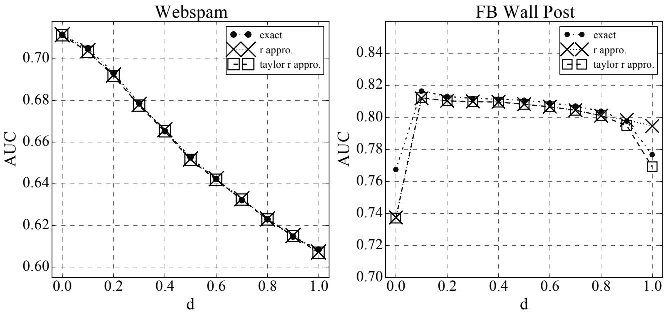


Figure 6: We draw the results on Webspam(e)(i) and FB Wall Post datasets. There are three curves in each feature, representing the original AttriRank model and two approximation models. The results demonstrate that the approximation tricks show no significant effects on the outputs.

5. CONCLUSION

We believe that PageRank could gain better ranking quality while considering the contribution from external attributes, and thus propose a general PageRank-attribute model, AttriRank, to achieve this goal. By constructing a random-walk model based on our PageRank and attribute assumptions, we have incorporated the node-attribute information into the PageRank framework without sacrificing the original theoretical benefits of PageRank. We further suggest and verify that our model can be applied to include internal attributes even when external information is unavailable. Another major contribution lies in the two approximation tricks we have proposed that allow the whole model to perform in linear time without sacrificing the performance. Acknowledging the challenge of parameter selection in an unsupervised model, in this paper we provide not only practical suggestions but also theoretical analyses on every parameter in our model. Finally, we conduct a series of experiments to verify the validity of the arguments throughout this paper.

Our future works are two-fold. First, we would like to extend the model to edge attributes. Second, we would like to investigate the qualify of ranking when integrating our model with node attributes learned from more complex models such as [5, 15, 17].

6. REFERENCES

[1] A. Agarwal, S. Chakrabarti, and S. Aggarwal. Learning to rank networked entities. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 14–23, New York, NY, USA, 2006. ACM.

[2] S. Agarwal. Ranking on graph data. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 25–32, New York, NY, USA, 2006. ACM.

[3] R. Baeza-Yates, P. Boldi, and C. Castillo. Generalizing pagerank: Damping functions for link-based ranking algorithms. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 308–315, New York, NY, USA, 2006. ACM.

[4] P. Boldi. Totalrank: Ranking without damping. In *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*, WWW '05, pages 898–899, New York, NY, USA, 2005. ACM.

[5] S. Cao, W. Lu, and Q. Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 891–900. ACM, 2015.

[6] H. Chang, D. Cohn, and A. McCallum. Learning to create customized authority lists. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 127–134, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[7] L. C. Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1978.

[8] B. Gao, T.-Y. Liu, W. Wei, T. Wang, and H. Li. Semi-supervised ranking on very large graphs with rich metadata. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 96–104, New York, NY, USA, 2011. ACM.

[9] D. F. Gleich. *Models and Algorithms for Pagerank Sensitivity*. PhD thesis, Stanford, CA, USA, 2009. AAI3382730.

[10] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pages 576–587. VLDB Endowment, 2004.

[11] J. Heidemann, M. Klier, and F. Probst. Identifying key users in online social networks: A pagerank based approach. 2010.

[12] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[13] D. Kuang, S. Yun, and H. Park. Symnmf: nonnegative low-rank approximation of a similarity matrix for graph clustering. *Journal of Global Optimization*, 62(3):545–574, 2015.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[15] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[16] D. Rao and D. Yarowsky. Ranking and semi-supervised classification on large scale graphs using map-reduce. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, TextGraphs-4, pages 58–65, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[17] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. ACM, 2015.

[18] A. C. Tsoi, G. Morini, F. Scarselli, M. Hagenbuchner, and M. Maggini. Adaptive ranking of web pages. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 356–365, New York, NY, USA, 2003. ACM.

[19] Y. Wang, Y. Tong, and M. Zeng. Ranking scientific articles by exploiting citations, authors, journals, and time information. In *AAAI*, 2013.

[20] M. Xie, J. Liu, N. Zheng, D. Li, Y. Huang, and Y. Wang. Semi-supervised graph-ranking for text retrieval. In *Proceedings of the 4th Asia Information Retrieval Conference on Information Retrieval Technology*, AIRS'08, pages 256–263, Berlin, Heidelberg, 2008. Springer-Verlag.

[21] W. Xing and A. Ghorbani. Weighted pagerank algorithm. In *Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on*, pages 305–314. IEEE, 2004.

[22] L. Zelnik-manor and P. Perona. Self-tuning spectral clustering. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1601–1608. MIT Press, 2005.

[23] L. Zhang, T. Qin, T.-Y. Liu, Y. Bao, and H. Li. N-step pagerank for web search. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 653–660, Berlin, Heidelberg, 2007. Springer-Verlag.

[24] H. Zhao, P. Poupart, Y. Zhang, and M. Lysy. Sof: Soft-cluster matrix factorization for probabilistic clustering, 2015.

[25] D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 1036–1043, New York, NY, USA, 2005. ACM.