A Generalized Co-HITS Algorithm and Its Application to Bipartite Graphs

Hongbo Deng
Department of CSE
The Chinese University of HK
Shatin, NT, Hong Kong
hbdeng@cse.cuhk.edu.hk

Michael R. Lyu
Department of CSE
The Chinese University of HK
Shatin, NT, Hong Kong
lyu@cse.cuhk.edu.hk

Irwin King
Department of CSE
The Chinese University of HK
Shatin, NT, Hong Kong
king@cse.cuhk.edu.hk

ABSTRACT

Recently many data types arising from data mining and Web search applications can be modeled as bipartite graphs. Examples include queries and URLs in query logs, and authors and papers in scientific literature. However, one of the issues is that previous algorithms only consider the content and link information from one side of the bipartite graph. There is a lack of constraints to make sure the final relevance of the score propagation on the graph, as there are many noisy edges within the bipartite graph. In this paper, we propose a novel and general Co-HITS algorithm to incorporate the bipartite graph with the content information from both sides as well as the constraints of relevance. Moreover, we investigate the algorithm based on two frameworks, including the iterative and the regularization frameworks, and illustrate the generalized Co-HITS algorithm from different views. For the iterative framework, it contains HITS and personalized PageRank as special cases. In the regularization framework, we successfully build a connection with HITS, and develop a new cost function to consider the direct relationship between two entity sets, which leads to a significant improvement over the baseline method. To illustrate our methodology, we apply the Co-HITS algorithm, with many different settings, to the application of query suggestion by mining the AOL query log data. Experimental results demonstrate that CoRegu-0.5 (i.e., a model of the regularization framework) achieves the best performance with consistent and promising improvements.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*retrieval models, search process*; H.2.8 [**Database Management**]: Database Applications—*data mining*

General Terms

Algorithms, Performance, Experimentation

KDD'09, June 28–July 1, 2009, Paris, France.

Keywords

Co-HITS, bipartite graphs, mutual reinforcement, score propagation, regularization

1. INTRODUCTION

Bipartite graphs have been widely used to represent the relationship between two sets of entities (which we refer to as two kinds of data to avoid ambiguity) for Web search and data mining applications. The Web offers rich relational data which can be represented by bipartite graphs, such as queries and URLs in query logs, authors and papers in scientific literature, and reviewers and movies in a movie recommender system. Taking the query-URL bipartite graph as an example, although there is no direct edges between two queries, the edges of the bipartite graph between queries and URLs may lead to hidden edges within the query set as shown in Fig. 1. Previous work [5] shows that there is a natural random walk on the bipartite graph, which demonstrates certain advantages comparing with the traditional approaches based on the content information. Many link analysis methods have been proposed, such as HITS [12] and PageRank [4], to capture some semantic relations within the bipartite graph.

The problem we address is how to utilize and leverage both the graph and content information, so as to improve the precision of retrieved entities. One good example is the query suggestion by mining a query log, in which we have a query-URL bipartite graph, and the queries and URLs. In addition, the queries and URLs can be represented as term vectors with the content information. The objective of the query suggestion is to find semantically similar queries for the given query q. Traditionally, we can identify initial similar queries based on the content information, then utilize HITS or personalized PageRank [10] for further mutual reinforcement on the bipartite graph. However, one of the issues is that there is a lack of constraints to make sure the final relevance of the score propagation on the graph, as there are many noisy edges within the bipartite graph. For example, let us consider the following two queries: *map* and *Yahoo*, where they may be co-linked by some URLs such as "www.yahoo.com" (*Yahoo!*). As the general URL *Yahoo!* is associated with many queries, it can aggregate large relevance scores by the mutual reinforcement, which may propagate the score to the highly connected query *Yahoo* and lead to the high relevance score between *map* and *Yahoo*. In this case, if we consider the content information of the URL *Yahoo!*, the relevance score of the URL *Yahoo!* against the query *map* will be very low. Thus, when incorporating the

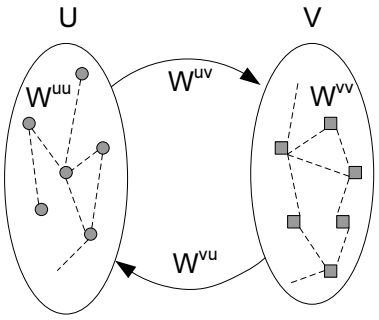


Figure 1: Example of a bipartite graph. The edges between U and V are represented as the transition matrices W^{uv} and W^{vu}. Note that the dashed lines represent hidden links when considering the vertices in one side, where W^{uu} and W^{vv} denote the hidden transition matrices within U and V respectively.

low relevance of the URL into the mutual reinforcement on the bipartite graph, the final relevance score between *map* and *Yahoo* would be constrained to a lower, but more reasonable score. In order to avoid the adverse effect of noisy data, we argue that the initial relevance scores, from both sides of the bipartite graph, provide valuable and reinforced information as well as the constraints of relevance, which should all be incorporated in a unified framework.

In this paper, we propose a novel and general algorithm, namely generalized Co-HITS, to incorporate the bipartite graph with the content information from both sides. Consequently, we investigate the following two frameworks, i.e., iterative framework and regularization framework, for the generalized Co-HITS algorithm from different views. The basic idea of the iterative framework is to propagate the scores on the bipartite graph via an iterative process with the constraints from both sides. The iterative framework contains HITS, personalized PageRank, and the one-step propagation algorithm as the special cases. Furthermore, we develop a joint regularization framework instead of the above iterative algorithm. In the regularization framework, we successfully build the connection with HITS, and develop a new cost function to consider the direct relationship between two entity sets, which leads to a significant improvement over the baseline method. To illustrate our methodology, we apply the generalized Co-HITS algorithm with different settings to the query suggestion task using the real-world AOL query log data [20]. Experimental results show that the CoRegu-0.5 (i.e., a model of the regularization framework) achieves the best performance, and its improvements are consistent and promising.

In a nutshell, our major contributions of this paper are: (1) the introduction of the generalized *Co-HITS* algorithm to incorporate the bipartite graph with the content information from both sides; (2) the investigation of two frameworks, including the iterative and the regularization frameworks, for the generalized Co-HITS algorithm from different perspectives; and (3) a new smoothness function in the regularization framework to consider the direct relationship between two entity sets as well as the smoothness within the same entity set, which leads to a significant improvement over the baseline method.

The rest of this paper is organized as follows. We first introduce the preliminaries in Section 2. In Section 3 we present the proposed Co-HITS algorithm, including the iterative framework and the regularization framework. Section 4 describes the application to bipartite graphs. We then describe and report the experimental evaluation in Section 5, and briefly review some related work in Section 6. Finally, we present our conclusions and future work in Section 7.

2. PRELIMINARIES

Consider a bipartite graph $G = (U \cup V, E)$, its *vertices* can be divided into two disjoint sets U and V such that each *edge* in E connects a vertex in U and one in V; that is, there is no edge between two vertices in the same set. Let $U = \{u_1, u_2, ..., u_m\}$ and $V = \{v_1, v_2, ..., v_n\}$ be the two sets of m and n unique entities. Generally, a bipartite graph can be modeled as a weighted directed graph. Given $i \in U$ and $j \in V$, if there is an edge connecting u_i and v_j, the transition probabilities w_{ij}^{uv} and w_{ji}^{vu} are positive, where w_{ij}^{uv} denotes the transition probability from u_i to v_j, and w_{ji}^{vu} denotes the transition probability from v_j to u_i; otherwise, $w_{ij}^{uv} = w_{ji}^{vu} = 0$. Since the transition probability from state i to some state must be 1, we have $\sum_{j \in V} w_{ij}^{uv} = 1$ and $\sum_{i \in U} w_{ji}^{vu} = 1$.

For a bipartite graph, there is a natural random walk on the graph with the transition probability as shown in Fig. 1. Let $W^{uv} \in \mathbb{R}^{m \times n}$ denote the transition matrix from U to V, whose entry (i, j) contains a weight w_{ij}^{uv} from u_i to v_j. Let $W^{vu} \in \mathbb{R}^{n \times m}$ be the transition matrix from V to U, whose entry (j, i) contains a weight w_{ji}^{vu} from v_j to u_i. To consider the vertices in one side, such as the query-to-query graph in query logs, then a hidden transition probability w_{ij}^{uu} from u_i to u_j, corresponding to a dashed line in Fig. 1, can be introduced as:

$$w_{ij}^{uu} = \sum_{k \in V} w_{ik}^{uv} w_{kj}^{vu}, \tag{1}$$

and

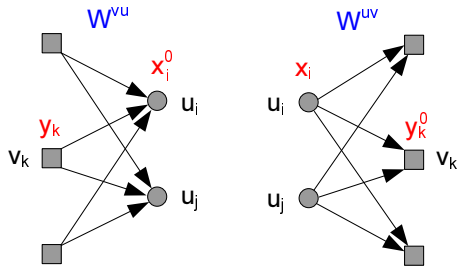
$$\begin{aligned} \sum_{j \in U} w_{ij}^{uu} &= \sum_{j \in U} \sum_{k \in V} w_{ik}^{uv} w_{kj}^{vu} = \sum_{k \in V} \left(w_{ik}^{uv} \sum_{j \in U} w_{kj}^{vu} \right), \\ &= \sum_{k \in V} w_{ik}^{uv} = 1. \end{aligned} \tag{2}$$

Similarly, for the transition probability from v_i to v_j, we can show that $w_{ij}^{vv} = \sum_{k \in U} w_{ik}^{vu} w_{kj}^{uv}$ and $\sum_{j \in V} w_{ij}^{vv} = 1$. We use $W^{uu} \in \mathbb{R}^{m \times m}$ and $W^{vv} \in \mathbb{R}^{n \times n}$ to denote the hidden transition matrices within U and V, respectively.

In addition to the graph information, each entity (such as a query or a document) may be represented as a term vector with its content information. For a given query q, the relevance scores of the entities can be calculated using a text relevance function f, such as the vector space model [1] and the statistical language model [23, 27]. The initial relevance scores x_i^0 and y_j^0 are respectively defined by $x_i^0 = f(q, u_i)$, and $y_j^0 = f(q, v_j)$ for u_i and v_j.

3. GENERALIZED CO-HITS ALGORITHM

Given a query q and the above information, the ultimate goal is to find a set of entities which are most relevant to the query q. The problem we address is how to utilize and leverage both the graph and content information, so as to



(a) (b)

Figure 2: Score propagation on the bipartite graph: (a) score y_k is propagated to u_i and u_j, and (b) score x_i is propagated to v_k.

improve the precision of the results. In this section, we propose a novel and general algorithm, namely generalized Co-HITS, to incorporate the bipartite graph with the content information from both sides.

3.1 Iterative Framework

The basic idea of our method is to propagate the scores on the bipartite graph via an iterative process. As shown in Fig. 2(a), the score y_k of v_k is propagated to u_i according to the transition probability. Similarly, additional scores are propagated from other vertices of V to u_i, then the score of u_i is updated to get a new value x_i. In Fig. 2(b), it shows that the new value x_i is propagated to v_k. The intuition behind the score propagation is the mutual reinforcement to boost co-linked entities on the bipartite graph. In addition, the initial relevance scores based on the content information provide invaluable information, which should also be considered in the framework.

In order to incorporate the bipartite graph with the content information, the generalized Co-HITS equations can be written as

$$x_i = (1-\lambda_u)x_i^0 + \lambda_u \sum_{k\in V} w_{ki}^{vu} y_k, \quad (3)$$

$$y_k = (1-\lambda_v)y_k^0 + \lambda_v \sum_{j\in U} w_{jk}^{uv} x_j, \quad (4)$$

where $\lambda_u \in [0,1]$ and $\lambda_v \in [0,1]$ are the personalized parameters, x_i^0 and y_k^0 are the initial scores for u_i and v_k respectively. In this model, the initial scores are normalized to be $\sum_{i\in U} x_i^0 = 1$ and $\sum_{k\in V} y_k^0 = 1$. Thus, after the updating operation, the sum of x_i and the sum of y_k will also be equal to 1 without further normalization. If only considering the vertices in one side, by substituting Eq. (4) for y_k in Eq. (3), the generalized Co-HITS equation can be represented as the following

$$\begin{aligned} x_i &= (1-\lambda_u)x_i^0 + \lambda_u(1-\lambda_v)\sum_{k\in V} w_{ki}^{vu} y_k^0 \\ &\quad + \lambda_u\lambda_v \sum_{j\in U}\left(\sum_{k\in V} w_{jk}^{uv} w_{ki}^{vu}\right) x_j, \\ &= (1-\lambda_u)x_i^0 + \lambda_u(1-\lambda_v)\sum_{k\in V} w_{ki}^{vu} y_k^0 \\ &\quad + \lambda_u\lambda_v \sum_{j\in U} w_{ji}^{uu} x_j. \end{aligned} \quad (5)$$

The final scores of every entities can be obtained through an iteratively updating process. From our empirical testing, we find in most cases the equation can converge after about 10 iterations.

The proposed Co-HITS framework is general, and it contains a large algorithm space as shown in Table 1, in which HITS and personalized PageRank are actually two special cases in this space. If λ_u is set to be 0, the algorithm returns the initial scores as the *baseline*. If λ_u and λ_v are all equal to 1, Eq. (5) becomes the ordinary HITS equation,

$$x_i = \sum_{j\in U} w_{ji}^{uu} x_j. \quad (6)$$

If one of the parameters λ_u and λ_v is set to be 1, it can be regarded as the personalized PageRank (PPR) algorithm [10]. Suppose $\lambda_v = 1$, it becomes

$$x_i = (1-\lambda_u)\cdot x_i^0 + \lambda_u \sum_{j\in U} w_{ji}^{uu} \cdot x_j. \quad (7)$$

When λ_v is set to be 0, the algorithm becomes a general hybrid method which aggregates the initial scores X^0 and Y^0 as follows,

$$x_i = (1-\lambda_u)\cdot x_i^0 + \lambda_u \sum_{k\in V} w_{ki}^{vu} \cdot y_k^0, \quad (8)$$

which can be viewed as an one-step propagation algorithm.

3.2 Regularization Framework

Here we investigate a joint regularization framework for the above iterative framework. Let us first consider the vertices in one side, and imagine the personalized PageRank algorithm within the graph U as Eq. (7). For each iteration, every node receives the score from its neighbors (second term), and also retain its initial score (first term). The iteration process continues, and finally converges with the scores that are determined by their neighbors on the graph and their initial scores. A regularization framework can be developed for the personalized PageRank algorithm, by regularizing the smoothness of relevance scores over the graph along with a regularizer on the initial ranking scores. The cost function R_1, associated with U, is defined to be

$$R_1 = \frac{1}{2}\sum_{i,j\in U} w_{ij}^{uu}\left\|\frac{x_i}{\sqrt{d_{ii}}} - \frac{x_j}{\sqrt{d_{jj}}}\right\|^2 + \mu\sum_{i\in U}\left\|x_i - x_i^0\right\|^2, \quad (9)$$

where $\mu > 0$ is the regularization parameter, and D is a diagonal matrix with entries $d_{ii} = \sum_j w_{ij}$ for normalization. Intuitively, the first term of the cost function defines the global consistency of the refined ranking scores over the graph, while the second term defines the constraint to fit the initial ranking scores, and the trade-off between each other can be controlled by the parameter μ. When $\mu \to +\infty$, R_1 puts all weights on the second term, and the regularization framework boils down to the *baseline* which corresponds to $\lambda_\mu = 0$ in Eq. (7). If $\mu = 0$, the regularization framework discards the initial ranking scores, and only takes into account the global consistency on the graph, which corresponds to $\lambda_\mu = 1$ in Eq. (7) (i.e., HITS as Eq. (6)). Similarly, for the cost function R_2 associated with V, we can show that

$$R_2 = \frac{1}{2}\sum_{i,j\in V} w_{ij}^{vv}\left\|\frac{y_i}{\sqrt{d_{ii}}} - \frac{y_j}{\sqrt{d_{jj}}}\right\|^2 + \mu\sum_{i\in V}\left\|y_i - y_i^0\right\|^2.$$

The intuition behind this framework is the global consistency, i.e., similar entities are most likely to have similar relevance scores with respect to a query.

Until now, R_1 and R_2 have defined the consistency based on the hidden links within U and V individually. However, the direct links between U and V may have more significant effect on the score propagation and mutual reinforcement. In this paper, we investigate and develop a new cost function R_3 to consider the direct relationship between U and V:

$$R_3 = \frac{1}{2}\sum_{i\in U, j\in V} w_{ij}^{uv}\left\|\frac{x_i}{\sqrt{d_{ii}}} - \frac{y_j}{\sqrt{d_{jj}}}\right\|^2 + \frac{1}{2}\sum_{j\in V, i\in U} w_{ji}^{vu}\left\|\frac{y_j}{\sqrt{d_{jj}}} - \frac{x_i}{\sqrt{d_{ii}}}\right\|^2. \quad (10)$$

The intuition behind R_3 is the smoothness constraint between two entity sets, which penalizes large differences in relevance scores for vertices between U and V that are strongly connected.

Formally, the cost function R, associated with both U and V, is defined to be

$$R = \lambda_r(R_1 + \alpha R_2) + (1-\lambda_r)R_3, \quad (11)$$

where $\alpha > 0$ and $\lambda_r \in [0,1]$. By minimizing the cost function R, we obtain the general regularization framework associated with the general Co-HITS equation as Eq. (5). In this paper, we simply set $\alpha = 1$ and focus on investigating the effect of parameter λ_r. Then the original optimization problem $\min_F(R)$ can be rewritten as follows:

$$\begin{aligned} \min_F \quad & \frac{1}{2}\sum_{i,j=1}^{m+n} w_{ij}\left\|\frac{f_i}{\sqrt{d_{ii}}} - \frac{f_j}{\sqrt{d_{jj}}}\right\|^2 + \mu\sum_{i=1}^{m+n}\left\|f_i - f_i^0\right\|^2 \\ s.t. \quad & W = \begin{bmatrix} W^{uu} & \beta\cdot W^{uv} \\ \beta\cdot W^{vu} & W^{vv} \end{bmatrix} \\ & F = \begin{bmatrix} X \\ Y \end{bmatrix} \\ & \beta = (1-\lambda_r)/\lambda_r, \end{aligned} \quad (12)$$

where X and Y are the score vectors for U and V respectively. Differentiating Eq. (12) [30, 32], we have

$$\frac{dR}{dF}\Big|_{F=F^*} = F^* - SF^* + \mu(F^* - F^0) = 0, \quad (13)$$

where $S = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$, then Eq. (13) can be transformed into

$$F^* - \frac{1}{1+\mu}SF^* - \frac{\mu}{1+\mu}F^0 = 0. \quad (14)$$

After simplifying, a closed-form solution can be derived,

$$F^* = \mu_\beta(I - \mu_\alpha S)^{-1}F^0, \quad (15)$$
$$\mu_\alpha = \frac{1}{1+\mu}, \text{ and } \mu_\beta = \frac{\mu}{1+\mu},$$

where I is an identity matrix. Note that μ_α ranges from 0 to 1, and $\mu_\alpha + \mu_\beta = 1$. In this paper, we consider the normalized Laplacian in [30], and S is positive-semidefinite. Details about how to calculate the matrix W and S will be introduced in Section 4.1. Given the initial ranking scores F^0 and the matrix S, we can compute the refined ranking scores F^* directly.

Table 1: Connections with other methods

Iterative Framework		
λ_u	λ_v	Description
$=0$	$\in[0,1]$	Initial scores $x_i = x_i^0$
$=1$	$=1$	Original HITS as Eq. (6)
$\in(0,1)$	$=1$	Personalized PageRank as Eq. (7)
$\in(0,1)$	$=0$	One-step propagation as Eq. (8)
$\in(0,1)$	$\in(0,1)$	General Co-HITS as Eq. (3)
Regularization Framework		
μ_α, λ_r		Description
$\mu_\alpha = 0$		Initial scores $x_i = x_i^0$
$\mu_\alpha = 1$		Corresponding to HITS
$\mu_\alpha \in (0,1)$		General regularization framework
$\lambda_r = 1$		Single-sided regularization
$\lambda_r \in (0,1)$		Double-sided regularization
$\lambda_r = 0.5$		$R = 0.5(R_1 + R_2) + 0.5R_3$

3.3 Connections and Justifications

In this section, we establish connections between the generalized Co-HITS algorithm and other methods in Table 1. The iterative framework contains HITS, personalized PageRank, and the one-step propagation algorithm as the special cases. When looking at the regularization framework, its variations are controlled by the parameters μ_α and λ_r. When $\mu_\alpha = 0$ ($\mu \to +\infty$), R puts all weights on the second term, and the regularization framework boils down to the *baseline*. If $\mu_\alpha = 1$ ($\mu = 0$), the regularization framework discards the initial ranking scores, and only takes into account the global consistency on the graph, which corresponds to the HITS algorithm. Moreover, a different selection of λ_r leads to a different smoothing strategy. If $\lambda_r = 1$, it only considers the single-side regularization within U and V. If $\lambda_r \in (0,1)$, it utilizes the double-side regularization to make full use of the bipartite graph.

For the large-scale information retrieval, the matrix S is usually very large but sparse, which can be loaded in a relatively small storage space. However, the inverse matrix $(I - \mu_\alpha S)^{-1}$ will be very dense, and may need a huge space to save it. To balance the storage space and the computation time of the inverse matrix, we suggest to approximate the Eq. (15) in a specific subgraph with a submatrix $\hat{S}$, which consists of the top-n entities according to the initial ranking scores $\hat{F}^0$. It can be found that the top ranking scores usually outnumber the very low ranking scores. Theoretically, if the ranking scores after n are close to 0, the following approximate solution is equivalent to Eq. (15),

$$\hat{F}^* = (I - \mu_\alpha \hat{S})^{-1}\hat{F}^0. \quad (16)$$

In this equation, we eliminate the parameter μ_β as it does not change the ranking. Accordingly, it needs to calculate the inverse matrix $(I - \mu_\alpha\hat{S})^{-1}$ online. Fortunately, the matrix is usually very sparse, then the complexity time of the sparse matrix inversion can be reduced to be linear with the number of nonzero matrix elements. In our experiments, we extract the top 5,000 entities for approximation.

4. APPLICATION TO BIPARTITE GRAPHS

To illustrate our proposed method, we use the statistical language model as the baseline to calculate the initial relevance scores based on the content information, and specify

the application in query suggestion base on the query-URL bipartite graph. In this section we introduce the bipartite graph construction and the statistical language model, then show the overall algorithm of our framework.

4.1 Bipartite Graph Construction

Bipartite graphs are widely used to describe the relationship between queries U and URLs V when mining the query logs, such as query suggestion and classification. The edges of the query-URL bipartite graph can capture some semantic relations between queries and URLs. For each edge $(q_i, d_j) \in E$ we associate a numeric weight c_{ij}, known as the *click frequency*, that measures the number of times the URL d_j was clicked when shown in response to the query q_i. The transition probability w_{ij}^{uv} [5, 22] from the query q_i to the URL d_j is defined by normalizing the click frequency from the query q_i as $w_{ij}^{uv} = \frac{c_{ij}}{\sum_{j \in V} c_{ij}}$, while the transition probability w_{ji}^{vu} from the URL d_j to the query q_i is defined as $w_{ji}^{vu} = \frac{c_{ij}}{\sum_{i \in U} c_{ij}}$. Thus, we can easily obtain the transition matrices W^{uv}, W^{vu}, W^{uu} and W^{vv}.

In practice, it is sometimes unnecessary to apply our learning algorithms to a very large bipartite graph constructed from the entire collection. Since our task is to find the most relevant queries as suggestion for a given query, it would be more efficient to apply our algorithm only to a relatively compact query-URL bipartite graph that covers the relevant queries and related URLs. We utilize the same method used in [15] for building a compact query-URL bipartite graph and iteratively expanding it in the following,

1. Initialize a query set $\hat{U} = U_L$ (seed query set), and initialize a URL set $\hat{V} = V_L$ (seed URL set);
2. Update $\hat{V}$ to add the set of URLs that are connected with $\hat{U}$;
3. Update $\hat{U}$ to be the set of queries that are connected with $\hat{V}$;
4. Iterate 2 and 3 until $\hat{U}$ and $\hat{V}$ reach a desired size;

The final bipartite graph $\hat{G}$ to which the algorithms are applied consists of $\hat{U}$, $\hat{V}$ and edges $\hat{E}$ connecting them. According to the relevance scores, we initialize the top-10 relevant queries and top-10 relevant URLs as the seed sets. Generally, it only needs one iteration to reach 5,000 entities in our experiments. In this paper, we employ the widely used k-nearest neighbor (k-NN) graph, where each node is connected to its k nearest neighbors under the transition probability measure and the edges can be weighed by the transition matrices. It has been shown to be effective when $k = 10$ in [7]. Then, the matrix $\hat{W}$ is constructed with maximum 50,000 $(5,000 \times 10)$ entries. After normalization, we can obtain the matrix $\hat{S}$. Fortunately, the matrix is usually very sparse, and the complexity time of the sparse matrix inversion can be reduced to be linear with the number of nonzero matrix elements.

4.2 Statistical Language Model

Using language models for information retrieval has been studied extensively in recent years [23, 27, 28]. To determine the probability of a query given a document, we infer a document model θ_d for each document in a collection. With query q as input, retrieved documents are ranked based on the probability that the document's language model would generate the terms of the query, $p(q|\theta_d)$. The ranking function $f^0(q, d)$ can be written as

$$f^0(q, d) = p(q|\theta_d) = \prod_{t \in q} p(t|\theta_d)^{n(t,q)}, \tag{17}$$

where $p(t|\theta_d)$ is the maximum likelihood estimation of the term t in a document d, and $n(t, q)$ is the number of times that term t occurs in query q. The likelihood of a query q consisting of a number of terms t for a document d under a language model with Jelinek-Mercer smoothing [28] is $p(t|\theta_d) = 0.5p(t|d) + 0.5p(t)$. With the language model, we calculate the initial ranking scores of the documents with respect to a query.

In our proposed method, we employ the language model to determine the initial relevance scores F^0 for the queries and URLs. Note the queries from the query log are very short, but it still can be viewed as a document in the language model. We can get better initial relevance scores if we perform the query expansion and construct the document model with the expanded queries. For each URL, although its exact content information is not included in the query log, it can be represented as a document by the aggregation of connected queries [21].

Algorithm 1 Generalized Co-HITS Algorithm

Input: Given a query q and the bipartite graph
Perform:

1. Calculate the initial ranking scores based on the statistical language model and extract the top-ranked U_L and V_L as the seed sets;
2. Expand and extract the compact bipartite subgraph $\hat{G} = (\hat{U} \cup \hat{V}, \hat{E})$;
3. Get the weight matrix $\hat{W}$ or $\hat{S}$, and normalize the corresponding initial scores F^0;
4. Solve Eq. (5) or Eq. (16) and get the final scores $\hat{F}^*$.

Output: Return the ranked queries

4.3 Overall Algorithm

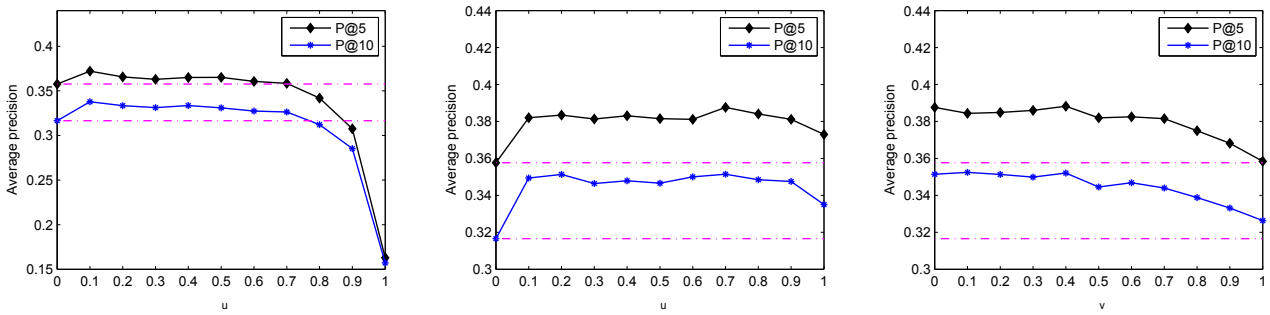
By unifying the Co-HITS algorithm in Section 3 and the application to bipartite graphs, we summarize the proposed algorithm in Algorithm 1. In the algorithm, note that we first perform preprocessing in a collection to construct the bipartite graph, and calculate the transition matrices. In the algorithm, we calculate the initial ranking scores using the language model, extract the compact bipartite subgraph, and perform the Co-HITS algorithm.

To implement the Co-HITS algorithm, we employ a sparse matrix package, i.e., CSparse [6], to solve the sparse matrix inversion efficiently. To deploy the efficient implementations of our scheme, all of the other algorithms used in the study are programmed in the C# language. We have implemented the language modeling approach to obtain the initial relevance scores with the Lucene.Net[1] package. For these experiments, the system indexes the collection and does tokenization, stopping and stemming in the usual way. The

[1]http://incubator.apache.org/lucene.net/

Table 2: Samples of the AOL query log dataset.

UserID	Query	Time	Rank	ClickURL
2722	yahoo	2006-04-25 13:03:23	1	http://www.yahoo.com
121537	map	2006-05-25 18:28:58	1	http://www.mapquest.com
123557	travel	2006-03-13 01:09:53	2	http://www.expedia.com
1903540	cheap flight	2006-05-15 00:31:43	1	http://www.cheapflights.com



(a) $\lambda_v = 1$ (PPR) (b) $\lambda_v = 0$ (OSP) (c) $\lambda_u = 0.7$ (CoIter)

Figure 3: The effect of varying parameters (λ_u and λ_v) in the iteration framework: (a) personalized PageRank, (b) one-step propagation, and (c) general Co-HITS. The dashed lines denote the baseline results.

testing hardware environment is on a Windows workstation with 3.0GHz CPU and 1GB physical memory.

5. EXPERIMENTAL EVALUATION

In the following experiments we compare our proposed algorithm with other methods on the tasks of mining query logs through an empirical evaluation. We define the following task: Given a query and a query-URL bipartite graph, the system has to identify a list of queries which are most similar or semantically relevant to the given query. In the rest of this section, we introduce the data collection, the assessments and evaluation metrics, and present the experimental results.

5.1 Data Collection

The dataset that we study is adapted from the query log of AOL search engine [20]. The entire collection consists of 19,442,629 user click-through records. These records contain 10,154,742 unique queries and 1,632,789 unique URLs submitted from about 650,000 users over three months (from March to May 2006). As shown in Table 2, each record of the click contains the same information: UserID, Query, Time, Rank and ClickURL. This dataset is the raw data recorded by the search engine, and contains a lot of noises. Hence, we conduct a similar method employed in [26] to clean the raw data. We clean the data by removing the queries that appear less than 2 times, and by combining the near-duplicated queries which have the same terms without the stopwords and punctuation marks (for example, "google's image" and "google image" will be combined as the same query). After cleaning, our data collection consists of 883,913 queries and 967,174 URLs. After the construction of the click graph, we observe that a total of 4,900,387 edges exist, which indicates that each query has 5.54 distinct clicks, and each URL is clicked by 5.07 distinct queries. Moreover, taken as a whole, this data collection has 250,127 unique terms which appear in all the queries.

5.2 Assessments and Evaluation Metrics

It is difficult to evaluate the quality of query similarity/relevance rankings due to the scarcity of data that can be examined publicly. For an automatic evaluation, we utilize the same method used in [2] to evaluate the similarity of retrieved queries, but engage the Google Directory[2] instead of the Open Directory Project[3]. When a user types a query in Google Directory, besides site matches, we can also find *category* matches in the form of paths between directories. Moreover, these categories are ordered by relevance. For instance, the query "United States" would provide the hierarchical category "Regional > North America > United States", while one of the results for "National Parks" would be "Regional > North America > United States > Travel and Tourism > National Parks and Monuments". Hence, to measure how similar two queries are, we can use a notion of similarity between the corresponding categories provided by the search results of Google Directory. In particular, we measure the similarity between two categories Ca_i and Ca_r as the length of their longest common prefix $P(Ca_i, Ca_r)$ divided by the length of the longest path between Ca_i and Ca_r. More precisely, the similarity is defined as:

$$Sim(Ca_i, Ca_r) = \frac{|P(Ca_i, Ca_r)|}{\max(|Ca_i|, |Ca_r|)}, \qquad (18)$$

where $|Ca_i|$ denotes the length of a path. For instance, the similarity between the above two queries is 3/5 since they share the path "Regional > North America > United States" and the longest one is made of five directories. We evaluate the similarity between two queries by measuring the similarity between the aggregated categories of the two queries, among the top 5 answers provided by Google Directory.

To give a fair assessment, we randomly select 300 distinct queries from the data collection, then retrieve a list of similar queries using the proposed methods for each of these

[2]http://directory.google.com/
[3]http://www.dmoz.org/

queries. For the evaluation of the task, we adopt the precision at rank n to measure the relevance of the top n results of the retrieved list with respect to a given query q_r, which is defined as

$$P@n = \frac{\sum_{i=1}^{n} Sim(q_i, q_r)}{n}, \quad (19)$$

where $Sim(q_i, q_r)$ means the similarity between q_i and q_r. In our experiments, we report the precision from $P@1$ to $P@10$, and take the average over all the 300 distinct queries.

5.3 Experimental Results

We consider the question whether our proposed method can boost the performance using the generalized Co-HITS algorithm for query suggestion. First the experiments are performed to compare the iterative framework of Co-HITS with different parameters λ_u and λ_v. Then we examine the performance of the regularization framework by varying the parameters μ_α and λ_r. Finally, we investigate and compare the detailed results of different methods, which shows that the regularization framework CoRegu-0.5 achieves the best results.

5.3.1 Comparison of Iterative Framework

For the iterative framework, the generalized Co-HITS contains HITS, personalized PageRank (PPR), and the one-step propagation (OSP) algorithms as the special cases. In this subsection, we compare the performance of general Co-HITS (CoIter) with the above special cases, and report the precisions of P@5 and P@10 in Fig. 3.

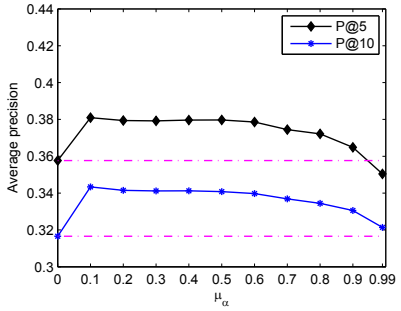
First of all, we evaluate the performance of personalized PageRank after setting $\lambda_v = 1$. Figure 3(a) illustrates the experimental results for different λ_u, in which the solid curves indicate the precisions of P@5 and P@10 for different parameters, and the dashed curves denote the precisions for the *baseline*. We can see that the performance has only a slight increase when compared to the baseline if λ_u is set close to 0. With the increase of λ_u, the performance becomes worse, and even underperforms the baseline. It is because of the lack of relevance constraints from both sides of the bipartite graph, so the score propagation on the graph may be influenced easily due to some noise edges. When λ_u is equal to 1, it corresponds to the HITS algorithm that discards the initial relevance scores.

When $\lambda_v = 0$, the Co-HITS algorithm boils down to simply aggregation of the initial scores from both sides. As shown in Fig. 3(b), we notice that the simple aggregation method (i.e., one-step propagation when λ_u is set from 0.1 to 0.9) benefits from both sides, and outperforms the method that only considers from one side. This observation supports the intuition of our Co-HITS algorithm that the initial relevance scores from both sides provide valuable and reinforced information as well as the constraints of relevance.

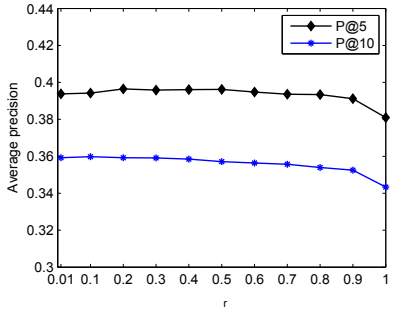
To illustrate the performance of general Co-HITS algorithm, we choose to set $\lambda_u = 0.7$ and vary the parameter λ_v from 0 to 1, and then show the results in Fig. 3(c). From this figure, we can observe that its improvement over the baseline is promising when compared to the personalized PageRank, and it is comparable with the one-step propagation when λ_v is set to be 0.4.

5.3.2 Comparison of Regularization Framework

For the regularization framework, we first evaluate the single-sided regularization (SiRegu) by varying the parameter μ_α, then we fix μ_α and perform the double-sided regularization (CoRegu) with different λ_r.



(a) $\lambda_r = 1$ (SiRegu)



(b) $\mu_\alpha = 0.1$ (CoRegu)

Figure 4: The effect of varying parameters (μ_α and λ_r) in the regularization framework: (a) single-sided regularization, and (b) double-sided regularization.

As mentioned in Table 1, the parameter μ_α is used to control the balance between the global consistency and the initial ranking scores in the unified regularization framework as Eq. (9), and it ranges from 0 to 1. The experimental results for the single-sided regularization are illustrated in Fig. 4(a). When $\mu_\alpha = 0$, SiRegu boils down to the initial *baseline*. We can see that the performance is improved over the *baseline* when incorporating the global consistency ($\mu_\alpha > 0$) in the framework. With the increase of μ_α, the performance becomes better until it puts too much weight on the term of global consistency ($\mu_\alpha \to 1$). If $\mu_\alpha \to 1$, SiRegu discards the initial ranking scores, and only takes into account the global consistency on the graph. As shown in Fig. 4(a), when the parameter μ_α is equal to 0.99, the performance of our method becomes worse than the initial *baseline* due to the overweighted global consistency. According to the theoretical analysis in Section 3.2, SiRegu corresponds to the personalized PageRank in the iteration framework. By comparing Fig. 4(a) with Fig. 3(a), both results are improved first and then degraded with the increase of μ_α and λ_u, which shows that the parameters μ_α and λ_u have similar impact on SiRegu and PPR, respectively.

We have shown that SiRegu can improve the performance over the initial baseline, and achieves the best performance when μ_α is set to be 0.1. Now we fix $\mu_\alpha = 0.1$, and examine whether CoRegu can further boost the performance by incorporating a direct smoothness constraint between two entity sets. According to Fig. 4(b), it is obvious that CoRegu ($\lambda_r < 1$) performs better than SiRegu ($\lambda_r = 1$). The improvement over the SiRegu method owes to the direct

Table 3: Comparison of different methods by P@5 and P@10. The mean precisions and the percentages of relative improvements are shown in the table.

Method	Para		Evaluation metrics	
Iter	λ_u	λ_v	P@5	P@10
Baseline	0	×	0.358 (0%)	0.317 (0%)
PPR-0.1	0.1	1	0.372 (4.0%)	0.338 (6.7%)
OSP-0.7	0.7	0	0.388 (8.4%)	0.351 (11.0%)
CoIter-0.4	0.7	0.4	0.388 (8.6%)	0.352 (11.2%)
Regu	λ_r	μ_α	P@5	P@10
SiRegu-0.1	1	0.1	0.381 (6.5%)	0.343 (8.5%)
CoRegu-0.5	0.5	0.1	0.396 (10.8%)	0.357 (12.8%)

smoothness constraint as Eq. (10) which is incorporated in the CoRegu framework. This observation supports the theoretical analysis of the proposed regularization framework. Moreover, CoRegu is relatively robust and may achieve the best results when the parameter λ_r is set to be 0.2-0.6.

5.3.3 *Detailed Results*

To gain a better insight into the proposed Co-HITS algorithm, we compare the best results of different models using P@5 and P@10 in Table 3. The mean precisions and the percentages of relative improvements over the baseline are shown in the table. A quick scan of the table reveals that CoRegu-0.5 achieves the best performance. When looking at the relative improvements of those models, we can see that CoRegu-0.5 improves over the baseline by 10.8% (for P@5) and 12.8% (for P@10) respectively, while CoIter-0.4 over the baseline by 8.6% and 11.2%. In addition, SiRegu-0.1 performs better than PPR-0.1. These results confirm that the regularization framework outperforms the iterative framework.

Figure 5 illustrates the precisions of six models from P@1 to P@10. In general, we can see that the performances of all the models, except the PPR-0.1, are better than the baseline. It is comparable for the precisions of OSP-0.7, CoIter-0.4 and SiRegu-0.1. The double-sided regularization model, i.e., CoRegu-0.5, achieves the best performance, whose improvements are consistent. After looking into the details, one important observation is that the improvements of our method over the baseline are increased for larger n (of the evaluation matric P@n). This is because the mutual reinforcement can boost the semantically relevant entities which have low initial scores. According to all the the experimental results, we can argue that it is very essential and promising to consider the double-sided regularization framework for the bipartite graph.

6. RELATED WORK

The work is related to the category of link analysis methods. In [9], the authors have tried to model a unified framework for link analysis, which includes the two popular ranking algorithms HITS [12] and PageRank [4]. Several normalized ranking algorithms are studied which are intermediate between HITS and PageRank. Our method differs from this unified framework as we integrate the graph information with the content information.

According to some generalization of PageRank and HITS, a family of work on the structural re-ranking paradigm over

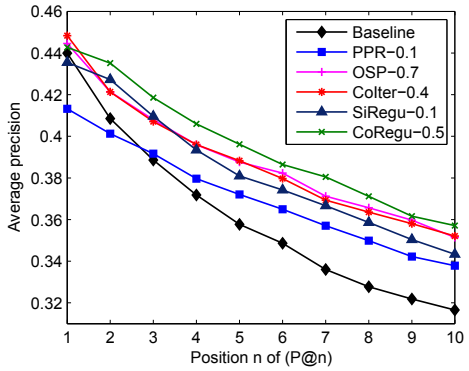


Figure 5: Comparison of six models.

a graph was proposed to refine the initial ranking scores. Kurland and Lee performed re-ranking based on centrality within graphs, through PageRank-inspired algorithm [13] and HITS-style cluster-based approach [14]. Zhang et al. [29] proposed a similar method to improve Web search results based on a linear combination of results from text search and authority ranking. In addition, PopRank [19] is developed to extend PageRank models to integrate heterogenous relationships between objects. Another approach suggested by Minkov et al. [18] has been used to improve an initial ranking on graph walks in entity-relation networks. However, those methods does not make full use of the content and graph information as they treat the content and graph information individually.

The regularization framework we proposed is closely related to graph-based semi-supervised learning [32, 30, 25, 31], which usually assume label smoothness over the graph. Mei et al. [17] extend the graph harmonic function [32] to multiple classes. However, our work is different from theirs, as their tasks are mainly used in query-independent settings (i.e., semi-supervised classification, topic modeling), while we focus on query-dependent ranking problems. With the advance of machine learning, graph-based models have been widely and successively used in information retrieval and data mining. Diaz [8] use score regularization to adjust adhoc retrieval scores from an initial retrieval. Deng et al. [7] propose a method to learn a latent space graph from multiple relationships between objects, and then regularize the smoothness of ranking scores over the latent graph. More recently, Qin et al. [24] use relational objects to enhance learning to rank with parameterized regularization models. But those three methods only consider the regularization from one side of the bipartite graph or within a single graph, while our regularization framework takes into account not only the smoothness within the same entity set but also the direct relationship between two entity sets.

This work is also related to query log analysis [2], as we apply our Co-HITS algorithm to the application of query suggestion by mining the query logs. A common model for utilizing query logs from search engines is in the form of a query-URL bipartite graph (i.e., click graph) [5]. Based on the click graph, many research efforts in query log analysis have been devoted to query clustering [3], query suggestion [11, 16] and query classification [15]. Craswell and Szummer [5] used click graph random walks for relevance rank in image search. Li et al. [15] presented the use of click

graphs in improving query intent classifiers. In this work, we combine the click graph with the content information from queries and URLs to improve the precisions of the results, which differs from the previous methods.

7. CONCLUSIONS

In this paper we have presented the generalized Co-HITS algorithm for bipartite graphs, whose basic idea is to incorporate the bipartite graph with the content information from both sides. We not only formally define the iterative framework, but also investigate the regularization framework for the generalized Co-HITS algorithm from different views. For the iterative framework, it has been shown that HITS, personalized PageRank, and the one-step propagation algorithm are special cases of the generalized Co-HITS algorithm. In the regularization framework, we successfully build the connection with HITS, and develop a new cost function to consider the direct relationship between two entity sets, which leads to a significant improvement over the baseline method. We have applied the proposed algorithm to mine the query log and compare with many different settings. Experimental results show that the improvements of our proposed model are consistent, and CoRegu-0.5 achieves the best performance. In future work, it would be interesting to investigate the performance of our Co-HITS algorithm in other bipartite graphs to see if the proposed method might have an impact on any bipartite graphs.

8. ACKNOWLEDGMENTS

This work described in this paper is supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4128/08E and Project No. CUHK4158/08E). This work is also affiliated with the Microsoft-CUHK Joint Laboratory for Human-Centric Computing and Interface Technologies.

9. REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern information retrieval*. Addison-Wesley Harlow, 1999.

[2] R. A. Baeza-Yates and A. Tiberi. Extracting semantic relations from query logs. In *KDD*, pages 76–85, 2007.

[3] D. Beeferman and A. L. Berger. Agglomerative clustering of a search engine query log. In *KDD*, pages 407–416, 2000.

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

[5] N. Craswell and M. Szummer. Random walks on the click graph. In *SIGIR*, pages 239–246, 2007.

[6] T. Davis. *Direct Methods for Sparse Linear Systems*. Society for Industrial Mathematics, 2006.

[7] H. Deng, M. R. Lyu, and I. King. Effective latent space graph-based re-ranking model with global consistency. In *WSDM*, pages 212–221, 2009.

[8] F. Diaz. Regularizing ad hoc retrieval scores. In *CIKM*, pages 672–679, 2005.

[9] C. H. Q. Ding, X. He, P. Husbands, H. Zha, and H. D. Simon. Pagerank, hits and a unified framework for link analysis. In *SIGIR*, pages 353–354, 2002.

[10] T. Haveliwala, S. Kamvar, and G. Jeh. An analytical comparison of approaches to personalizing PageRank. *Preprint, June*, 2003.

[11] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *WWW*, pages 387–396, 2006.

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

[13] O. Kurland and L. Lee. Pagerank without hyperlinks: structural re-ranking using links induced by language models. In *SIGIR*, pages 306–313, 2005.

[14] O. Kurland and L. Lee. Respect my authority!: Hits without hyperlinks, utilizing cluster-based language models. In *SIGIR*, pages 83–90, 2006.

[15] X. Li, Y.-Y. Wang, and A. Acero. Learning query intent from regularized click graphs. In *SIGIR*, pages 339–346, 2008.

[16] H. Ma, H. Yang, I. King, and M. R. Lyu. Learning latent semantic relations from clickthrough data for query suggestion. In *CIKM*, pages 709–718, 2008.

[17] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *WWW*, pages 101–110, 2008.

[18] E. Minkov, W. W. Cohen, and A. Y. Ng. Contextual search and name disambiguation in email using graphs. In *SIGIR*, pages 27–34, 2006.

[19] Z. Nie, Y. Zhang, J.-R. Wen, and W.-Y. Ma. Object-level ranking: bringing order to web objects. In *WWW*, pages 567–574, 2005.

[20] G. Pass, A. Chowdhury, and C. Torgeson. A picture of search. In *Infoscale*, 2006.

[21] B. Poblete and R. A. Baeza-Yates. Query-sets: using implicit feedback and query patterns to organize web documents. In *WWW*, pages 41–50, 2008.

[22] B. Poblete, C. Castillo, and A. Gionis. Dr. searcher and mr. browser: a unified hyperlink-click graph. In *CIKM*, pages 1123–1132, 2008.

[23] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281, 1998.

[24] T. Qin, T.-Y. Liu, et al. Learning to rank relational objects and its application to web search. In *WWW*, pages 407–416, 2008.

[25] A. Smola and R. Kondor. Kernels and regularization on graphs. *COLT*, 2003.

[26] X. Wang and C. Zhai. Learn from web search logs to organize search results. In *SIGIR*, pages 87–94, 2007.

[27] C. Zhai and J. D. Lafferty. Two-stage language models for information retrieval. In *SIGIR*, pages 49–56, 2002.

[28] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004.

[29] B. Zhang, H. Li, Y. Liu, L. Ji, W. Xi, W. Fan, Z. Chen, and W.-Y. Ma. Improving web search results using affinity graph. In *SIGIR*, pages 504–511, 2005.

[30] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, 2003.

[31] D. Zhou, B. Schölkopf, and T. Hofmann. Semi-supervised learning on directed graphs. In *NIPS*, 2004.

[32] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.