Using a Layered Markov Model for Distributed Web Ranking Computation*

Jie Wu
School of Comp. and Comm. Sciences
Ecole Polytechnique Fédérale de Lausanne
1015 Lausanne, Switzerland
jie.wu@epfl.ch

Karl Aberer
School of Comp. and Comm. Sciences
Ecole Polytechnique Fédérale de Lausanne
1015 Lausanne, Switzerland
karl.aberer@epfl.ch

Abstract

The link structure of the Web graph is used in algorithms such as Kleinberg's HITS and Google's PageRank to assign authoritative weights to Web pages and thus rank them. Both require a centralized computation of the ranking if used to rank the complete Web graph. In this paper, we propose a new approach based on a Layered Markov Model to distinguish transitions among Web sites and Web documents. Based on this model, we propose two different approaches for computation of ranking of Web documents, a centralized one and a decentralized one. Both produce a well-defined ranking for a given Web graph. We then formally prove that the two approaches are equivalent. This provides a theoretical foundation for decomposing link-based rank computation and makes the computation for a Web-scale graph feasible in a decentralized fashion, such as required for Web search engines having a peer-to-peer architecture. Furthermore, personalized rankings can be produced by adapting the computation at both the local layer and the global layer. Our empirical results show that the ranking generated by our model is qualitatively comparable to or even better than the ranking produced by PageRank.

1. Introduction

Applying the peer-to-peer architectural paradigm to Web search engines has recently become a subject of intensive research. Whereas proposals have been made for the decomposition of content-based retrieval techniques, such as classical text-based vector space retrieval or latent semantic indexing, it is much less clear of how to decompose the computation for ranking methods based on the link structure of the Web.

*The work presented in this paper was carried out in the framework of the EPFL Center for Global Computing and supported by the Swiss National Funding Agency OFES as part of the European FP 6 STREP project ALVIS (002068).

We first briefly review the two most prominent link-based ranking algorithms - HITS [10] and PageRank [14], and then describe our main contribution towards enabling link-based ranking in peer-to-peer Web search engines.

1.1. Link-based Web Document Ranking

HITS is a query-dependent approach, which first obtains a subgraph of the Web relevant to a query result, and then applies the algorithm to this subgraph. On the other hand, PageRank is query-independent and operates on the whole Web graph directly. Disregarding this difference, both rely on the same principles of linear algebra to generate a ranking vector, by using a principal Eigenvector of a matrix generated from the (sub) Web graph to be studied.

It has been shown [4] that HITS is often instable such that the returned Eigenvectors depend on variations of the initial seed vector and that the resulting Eigenvectors inappropriately assign zero weights to parts of the graph. In short, HITS lacks strong theoretical basis assuring certain desirable properties of the resulting rankings.

In PageRank the process of surfing the Web is used as an intuitive model for assessing importance of Web pages and is modeled as a random walker on the Web graph. The computation of PageRank (and similarly of HITS if it were applied to the complete Web graph) is then performed on a matrix representation of the walker's stochastic state transition on the complete Web graph. It is inherently difficult to decompose as it would be required for a distributed computation in a peer-to-peer architecture.

1.2. SiteRank for Computation Distribution

Various methods [6, 7, 8] have been proposed to speed up the computation of PageRank by either more efficient implementations or the application of optimized numerical algorithms. However, all these attempts have a limited potential of keeping up with the Web growth, as the complexity of centralized computation of Page rank is inherently

related to the size of the link matrix of the Web, which is extremely large and growing. A different direction has been taken in [2, 9], where structural features of the Web graph related to the hierarchical organization of the Web are used to simplify the link matrix and thus the computation of PageRank. In [2] the Web graph is aggregated at the Web site level in order to improve the efficiency of PageRank computation. Even though these algorithms make use of the internal structure of the Web in ranking computation, they are typically designed to be centralized.

Recently several research groups, including ours, have thus investigated the possibility to compute PageRank in a distributed fashion [17, 19, 20]. Common to these approaches is an idea similar to the one used in [2]: the Web graph is aggregated at the Web site level and a ranking is computed at this granularity, which we call *SiteRank*. For each Web site an independent ranking is computed for the local document collection, which we call a local *DocRank*. The different DocRanks are then aggregated by using the SiteRank to weight the relative importance of the Web sites. Apparently this computation can be performed in a distributed manner since all DocRanks can be computed independently. The computation of SiteRank is of a comparably low complexity and can be either performed centrally or in a distributed fashion, depending on the architectural requirements. First experimental studies provide empirical evidence that rankings computed in this way, not only allow a widely distributed and thus scalable computation, but also produce ranking results comparable (and sometimes even more reasonable) than applying global PageRank. However, as opposed to the standard PageRank method, a solid theoretical foundation for this approach has not yet been provided. This is the gap that we close in this paper.

The model for distributed link-based web rank computation that we introduce is based on Layered Markov Models taking into account the Web site structure as a higher level of abstraction in the standard random surfer model of the Web. It serves the following purposes:

- It clarifies the relationship between a standard PageRank computation and distributed computation by composing SiteRank with DocRank. Thus it provides a precise semantics to SiteRank methods relative to PageRank.

- It provides sufficient conditions on SiteRank and DocRank computation to ensure non-degeneracy of the global ranking resulting from their composition. One of the important observation on PageRank was that ensuring primitivity of the link matrix by adding a random transition matrix is required. We show that primitivity of the global Web site graph and the local document graphs is sufficient.

- It shows the equivalence of distributed rank computation to a specific global ranking computation. Thus we prove that the distributed rank computation is equivalent to a well-understood centralized algorithm.

In addition, our model provides a foundation for a whole class of ranking methods, e.g. by replacing the PageRank algorithm by any other methods for the computation of DocRank and/or SiteRank at different layers, including methods that exploit user relevance feedback, of which the models that have been proposed in [2, 17, 20, 19] are specific instances. It thus generalizes the existing proposals.

1.3. Contribution

In our Layered Markov Model the Web is no longer considered as a flat graph of Web documents, but characterized by a multi-layer hierarchical structure. In the analysis, a two-layer structure is used: the graph of Web sites at the higher layer, and the graphs of Web documents at the lower layer. A transition from one Web document to another is mapped to both transitions between Web sites at the higher layer and transitions between Web pages on the same Web site at the lower layer. We will show in this paper that this model has the following important properties:

1. The derived ranking method satisfies basic properties required for consistently producing rankings. In particular, the ranking is well-defined and produces a probability distribution over the Web pages.

2. We provide a Partition Theorem for Rank Computation showing that by using the model we can provide a distributed algorithm for computing the ranking that is equivalent to result from a well-defined global algorithm.

3. Empirical experiments demonstrate that the ranking result produced by our approach is qualitatively comparable to or even better than that of PageRank. Link spamming is also defeated to a satisfiable degree.

4. While the model provides an alternative to existing link-based ranking methods allowing for distributed computation, it also introduces the possibility to generate in an elegant way personalized rankings by including into the computational personal preferences at both the Web site layer and the Web page layer.

2. Layered Markov Model

In this section, we first summarize the classical PageRank algorithm, then we present our new model.

2.1. The Classical Web Ranking

In the classical PageRank model, a surfer performs random walks on the flat graph generated by the Web pages, by either following hyperlinks on Web pages or jumping to a random page if no such link exists. A damping factor is defined to be the probability that a surfer does follow a hyperlink contained in the page where the surfer is currently located. Suppose the damping factor is f, then the probability that the surfer performs a random jump is $1 - f$. PageRank first generates a transition matrix $\mathbf{M}$ based on the original link-based adjacency matrix. However, the matrix $\mathbf{M}$ does not ensure the existence of the stationary vector of the Markov chain which characterizes the surfer behavior, i.e. the PageRank vector. As widely accepted, the unaltered Web creates a reducible Markov chain [1]. Thus, PageRank enforces the following adjustment to make a new irreducible transition matrix:

$$\hat{\mathbf{M}} = f\mathbf{M} + \frac{1-f}{N_D}\mathbf{e}\mathbf{e}' \tag{1}$$

where N_D is the total number of Web documents, $\mathbf{e}$ is the column vector of all 1s and $\mathbf{e}'$ is $\mathbf{e}$ transposed. $\hat{\mathbf{M}}$ is then primitive [2], thus the power method will finally produce the stationary PageRank vector. The adjustment is called *maximal irreducibility* [11].

We also use $M(G)$ and $\hat{M}(G)$ to denote the function of generating such matrices for a given graph G. Remember that in the function body of $\hat{M}(G)$, personalization of rankings can be obtained by replacing $\mathbf{e}'$ with a personalized distribution vector $\mathbf{v}_p$'s transposed $\mathbf{v}'_p$ in equation (1).

2.2. Why Layered Markov Model

Hierarchical Hidden Markov Models are used in [1] and similar work to determine optimal parameters for a Hidden Markov Model (HMM) given observed outputs from the hidden states. The main purpose of this work is to reduce the complexity of learning a hidden model for large-scale and highly complex application domains, such as analysis of traffic data from ISDN traffic, Ethernet LAN's, Common Channel Signaling Network (CCNS) and Variable Bit Rate (VBR) video, etc.. Studies have shown that for such problems, applying the standard HMM learning algorithm does not generate acceptable results.

The study in this paper is fundamentally different from that work both in the model and the problem itself: In our model, we do not have observed outputs while a hidden model has. The purpose is also different. We do not aim to learn system parameters, but to mine the link structure and obtain a ranking for all global system states related to the Web graph.

While PageRank assumes that the Web is a flat graph of documents and the surfers move among them without exploiting the hierarchical structure, we consider the Layered Markov Model as a suitable replacement for the flat chain to analyze the Web link structure for the following reasons:

- The logical structure of the Web graph is inherently hierarchical. No matter, whether the Web pages are grouped by Internet domain names, by geographical distribution, or by Web sites, the resulting organization is hierarchical. Such a hierarchical structure does definitely influence the patterns of user behavior.
- The Web is shown to be self-similar [3] in the sense that interestingly, part of it demonstrates properties similar to those of the whole Web. Thus instead of obtaining a snapshot of the whole Web graph, introducing substantial latency, and performing costly computations on it, bottom-up approaches, which deal only with part of the Web graph and then integrate the partial results in a decentralized way to obtain the final result, seem to be a very promising and scalable alternative for approaching such a large-scale problem.

Figure 1 illustrates a Layered Markov Model structure. The model consists of 12 sub-states (small circles) and 3 super-states (big circles), which are referred to as *phase*s in [1]. There exists a transition process at the upper layer among phases and there are three independent transition processes happening among the sub-states belonging to the three super-states.[3]

When applying the Web surfer paradigm, a phase could be considered as a surfer's staying within a specific Web site or a particular group of Web pages. The transition among phases corresponds to a surfer's moving from one Web site or group to another. The transition among sub-states corresponds to a surfer's movement within the site or group. Thus a comprehensive transition model should be a function of both the transitions among phases and the transitions among sub-states. In other words, the global system behavior emerges from the behaviors of decentralized and cooperative local sub-systems.

We consider a *two-layer* model in the following to keep explanations simple, but the analysis can be extended to *multi-layer* models using similar reasoning. We introduce now the notations to describe the two-layer model.

[1] When represented by the stochastic transition matrix, reducibility means the matrix is not strongly connected.

[2] This property pertains to whether a unique ranking vector exists. A primitive matrix is a nonnegative irreducible matrix that has only 1 eigenvalue on its spectral circle. A nonnegative matrix M is primitive if and only if $M^p > 0$ for some $p > 0$ [13]. A positive matrix is thus always primitive.

[3] Please note the figures of layered models here are only for the purpose of illustration, and the transition probabilities in the matrices used in our examples later are not necessarily related to the edges of the graph shown in this figure.

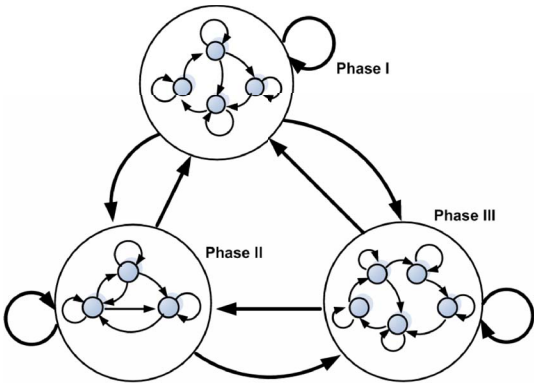

Figure 1. Phases and sub-states in Layered Markov Model.

- Given the number of phases $N_{\mathbb{P}}$, we use $\{1, 2, \cdots, N_{\mathbb{P}}\}$ to label the individual phases and denote the phase active at time t as a variable $Z(t)$. The set of phases is denoted by $\mathbb{P} = \{\mathbb{P}_1, \mathbb{P}_2, \cdots, \mathbb{P}_{N_{\mathbb{P}}}\}$.

- For each phase $\mathbb{P}_I$ the number of its sub-states is n_I. We use $\{1, 2, \cdots, n_I\}$ to label the individual sub-states and denote the state at time t as a variable $z^I(t)$. The set of sub-states of phase $\mathbb{P}_I$ is denoted by $O^I = \{o^I_1, o^I_2, \cdots, o^I_{n_I}\}$. The overall set of sets of sub-state is denoted by $\mathbb{O} = \{O^1, O^2, \cdots, O^{N_{\mathbb{P}}}\}$.

- The transition probability at the phase layer is given by $\mathbf{Y} = \{y_{IJ}\}$ where $y_{IJ} = P(Z(t+1) = J | Z(t) = I)$ and $1 \leq I, J \leq N_{\mathbb{P}}$. The initial state distribution vector is denoted by $\mathbf{v}_Y$.

- For each phase I, the transition probability at the sub-state layer is given by $\mathbf{U}^I = \{u^I_{ij}\}$ where $u^I_{ij} = P(z^I(t+1) = j | Z(t+1) = I, Z(t) = I, z^I(t) = i)$ and $1 \leq i, j \leq n_I$. In addition, $\mathbb{U}$ is defined to be the set of all sub-state transition matrices: $\mathbb{U} = \{\mathbf{U}^1, \mathbf{U}^2, \cdots, \mathbf{U}^{N_{\mathbb{P}}}\}$. There exists a one-to-one mapping between $\mathbb{P}$ and $\mathbb{U}$, namely each phase $\mathbb{P}_I$ has its sub-state transition matrix U^I, $1 \leq I \leq N_{\mathbb{P}}$. The set of initial state distribution vectors is denoted by $\mathbf{v}_{\mathbb{U}} = \{\mathbf{v}^1_U, \mathbf{v}^2_U, \cdots, \mathbf{v}^{N_{\mathbb{P}}}_U\}$.

When context is clear, we also use the index of a phase or a sub-state to designate the phase or sub-state. For example, phase 2 for $\mathbb{P}_2$ and its sub-state 3 for o^2_3 in O^2. An overall system state is denoted by a (phase,sub-state) pair like (2,3) which means the system is at the sub-state 3 of phase 2. In addition, $\mathcal{N}_{\mathbb{P}} = \sum_{I=1}^{N_{\mathbb{P}}} n_I$ is used to denote the total number of overall system states. An overall system state is also called a global system state in contrast to a local sub-state (i.e. a sub-state local to a phase).

Definition 1. *A (two-layer)* Layered Markov Model *is a 6-tuple* $\mathbb{LMM} = (\mathbb{P}, \mathbf{Y}, \mathbf{v}_Y, \mathbb{O}, \mathbb{U}, \mathbf{v}_{\mathbb{U}})$ *where each dimension has the meaning explained above.*

2.3. LMM for Ranking Global System States

We want to use the Layered Markov Model to compute a ranking for all global system states, i.e., a stationary (if possible) distribution vector for all global system states. Such a ranking also should be uniquely defined.

We assume that state transition between two global system states is always abstracted as first an inter-phase transition, and then an intra-phase transition.

As an example, suppose we have a phase transition matrix $\mathbf{Y}$, and three sub-state transition matrices $\mathbf{U}^1$ of the four-sub-state phase I, $\mathbf{U}^2$ of the three-sub-state phase II, and $\mathbf{U}^3$ of the five-sub-state phase III as follows:

$$\mathbf{Y} = \begin{bmatrix} .1 & .3 & .6 \\ .2 & .4 & .4 \\ .3 & .5 & .2 \end{bmatrix} \qquad \mathbf{U}^1 = \begin{bmatrix} .3 & .3 & .2 & .2 \\ .5 & .1 & .1 & .3 \\ .1 & .2 & .6 & .1 \\ .4 & .3 & .1 & .2 \end{bmatrix}$$

$$\mathbf{U}^2 = \begin{bmatrix} .2 & .1 & .7 \\ .1 & .8 & .1 \\ .05 & .05 & .9 \end{bmatrix} \qquad \mathbf{U}^3 = \begin{bmatrix} .6 & .02 & .2 & .1 & .08 \\ .05 & .2 & .5 & .05 & .2 \\ .4 & .1 & .2 & .1 & .2 \\ .7 & .1 & .05 & .1 & .05 \\ .5 & .2 & .1 & .1 & .1 \end{bmatrix}$$

We want to rank the 12 global system states according to the general authority implied by the transition link structure. To do so, we need to obtain a global transition probability matrix for the 12 global system states. To derive such a matrix, we first introduce our notion of *layer-decomposability*.

2.3.1 Layer-Decomposability

Informally, the property of layer-decomposability ensures the legitimacy of decomposing the transition between two global system states into the two steps of first inter-phase transition then intra-phase transition.

In order to define the decomposability between layers, we first introduce the concept of *gatekeeper sub-state*.

Definition 2. *A gatekeeper sub-state o^I_G of a phase $\mathbb{P}_I$ is a* virtual *sub-state appended to the phase, such that it connects to every other sub-state and every other sub-state is connected to it.*

After the introduction of gatekeeper sub-states for phases, the decomposability of a Layered Markov Model is defined as below.

Definition 3. *Layers in a Layered Markov Model are* decomposable *if the transition probability between two given non-gatekeeper sub-states in their two corresponding*

phases satisfies:

$$P(Z(t+1)=J, z(t+1)=j | Z(t)=I, z(t)=i) \quad (2)$$
$$= P(Z(t+1)=J | Z(t)=I) P(z^J(t+1)=j | z^J(t)=o_G^J)$$

The definition basically assures that whenever a phase transition takes place, it has to go through the gatekeeper sub-state of the destination phase. The gatekeeper sub-state functions as the boundary between inter-phase transitions and intra-phase transitions.

Denoting the transition probability in phase $\mathbb{P}_J$ from the gatekeeper sub-state o_G^J to sub-state o_j^J by u_{Gj}^J, the elements of the resulting global transition matrix W are computed as follows:

$$w_{(I,i)(J,j)} = y_{IJ} u_{Gj}^J \quad (3)$$

W.l.o.g., we assume that for each $J \in [1, N_{\mathbb{P}}]$, we have

$$\sum_j u_{Gj}^J = 1 \quad (4)$$

as this is the sum of the transition probabilities from the gatekeeper sub-state o_G^J to all the sub-states o_j^J, $j = 1, 2, \cdots, n_J$ within phase $\mathbb{P}_J$.

We show the following.

Lemma 1. *The resulting transition matrix* $\mathbf{W}$ *satisfies the raw stochastic property.*

Proof. Given an overall system state (I, i),

$$\sum_J \sum_j w_{(I,i)(J,j)} = \sum_J \sum_j y_{IJ} u_{Gj}^J = \sum_J y_{IJ} \sum_j u_{Gj}^J = 1$$

□

2.3.2 Transition Probabilities of Gatekeeper Sub-states

To compute (3), for each phase J we have to obtain the u_{Gj}^J values of all $j \in [1, n_J]$.

We already have the Markovian (not necessarily irreducible) transition matrix $\mathbf{U}^J$. After adding the new virtual gatekeeper sub-state, we need to make the new $(n_J+1) \times (n_J+1)$ matrix $\hat{\mathbf{U}}^J$ Markovian as well. A possible method of applying such a change is:

$$\hat{\mathbf{U}}^J = \left[\begin{array}{c|c} \alpha \mathbf{U}^J & (1-\alpha)\mathbf{e} \\ \hline (\mathbf{v}_U^J)^T & 0 \end{array} \right]$$

where $0 < \alpha < 1$ is an adjustable parameter, $\mathbf{e}$ is the column vector of all 1s and $\mathbf{v}_U^J$ is the initial state distribution vector for all the non-gatekeeper sub-states within $\mathbb{P}_J$, as we have described before. The new matrix $\hat{\mathbf{U}}^J$ is not only Markovian, but also irreducible and primitive.

This method is actually known as the approach of *minimal irreducibility* in the context of PageRank computation. In detail, applying the power method on $\hat{\mathbf{U}}^J$ will eventually produce its principal Eigenvector. After that, the last element of the vector, which corresponds to the appended gatekeeper sub-state in our case, is removed and the remaining n_J elements are re-normalized to sum up to 1. The resulting vector π_U^J is considered as the stationary distribution over all the non-gatekeeper sub-states within the given phase J. We take the n_J elements of the stationary distribution vector π_U^J as the values of all u_{Gj}^J, $j \in [1, n_J]$.

Interestingly enough, it is shown in [11] that this method is equivalent in theory and in computational efficiency to Google's method of maximal irreducibility. Thus, given the adjustable factor α, we actually take the *PageRank value*s of the local sub-states of $\mathbb{P}_J$ as their u_{Gj}^J values, $j \in [1, n_J]$.

To compute a ranking for the system states, we need to ensure the primitivity of the new global transition matrix.

Lemma 2. *If* $\mathbf{Y}$ *is primitive and the PageRank values of the local sub-states of* $\mathbb{P}_J$ *are taken as their* u_{Gj}^J *values,* $j \in [1, n_J]$*, the global transition matrix* $\mathbf{W}$ *is also primitive.*

Proof. This is a natural consequence of all the u_{Gj}^J values' being positive. □

Thus $\mathbf{W}$ has only one Eigenvalue on its spectral circle. The corresponding Eigenvector is used to rank the states in the overall system. However, we do not make the assumption that both $\mathbf{Y}$ and $\mathbf{U}$ are primitive, we are only sure that both of them are Markovian. Even if they are not primitive, we can make the resulting $\mathbf{W}$ primitive by adopting the same approach as taken in PageRank, the so-called method of maximal irreducibility, by connecting every pair of nodes via random jumps. Once the primitivity is achieved, we can always compute the ranking of the system states.

We now compute the $\mathbf{W}$ for our example given by the four Markovian matrices $\mathbf{Y}, \mathbf{U}^1, \mathbf{U}^2$ and $\mathbf{U}^3$. First, we compute the PageRank vectors for three phases (denoted by π_G^J,$J = 1, 2, 3$):

$$\pi_G^1 = \begin{pmatrix} 0.3054 \\ 0.2312 \\ 0.2582 \\ 0.2052 \end{pmatrix} \quad \pi_G^2 = \begin{pmatrix} 0.1191 \\ 0.2691 \\ 0.6117 \end{pmatrix} \quad \pi_G^3 = \begin{pmatrix} 0.4557 \\ 0.1038 \\ 0.2014 \\ 0.1106 \\ 0.1285 \end{pmatrix}$$

Then we use equation (3) to obtain the new $\mathbf{W}$ [4]. The elements of this global system transition matrix are the probabilities of transitions among global system states. The elements of both the rows and columns are in the order of (1,1), (1,2), (1,3), (1,4), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3), (3,4), (3,5). $1 \cdots 12$ are assigned as their corresponding global

[4]The actual value is not included here due to the limit of space. It can be found in [18].

system state index. For example, the element $w_{(12)(7)} = w_{(3,5)(2,3)}$ is the transition probability from the sub-state 5 of phase 3 (global system state 12) to the sub-state 3 of phase 2 (global system state 7). Layer decomposability assures that $w_{(3,5)(2,3)} = y_{32}u^2_{G3} = 0.5 \times 0.6117 = 0.3059$.

As equation (3) does not depend on i anymore given a global system state (I, i), we can find that in the matrix $\mathbf{W}$ rows pertaining to a particular value I are constant.

At this point, we are able to compute a ranking for the global system states. There are two possible approaches.

Approach 1: We apply the standard PageRank algorithm to $\mathbf{W}$ to rank all states, i.e. we apply the method of maximal irreducibility to $\mathbf{W}$ before we launch the power method to compute the principal Eigenvector. We obtain $\pi_{\mathbf{W}}$ as follows:

$$\begin{array}{l} 1:(1,1) \\ 2:(1,2) \\ 3:(1,3) \\ 4:(1,4) \\ 5:(2,1) \\ 6:(2,2) \\ 7:(2,3) \\ 8:(3,1) \\ 9:(3,2) \\ 10:(3,3) \\ 11:(3,4) \\ 12:(3,5) \end{array} \quad \pi_{\mathbf{W}} = \begin{pmatrix} 0.0682 \\ 0.0547 \\ 0.0596 \\ 0.0499 \\ 0.0545 \\ 0.1073 \\ 0.2281 \\ 0.1562 \\ 0.0452 \\ 0.0760 \\ 0.0474 \\ 0.0530 \end{pmatrix} \begin{array}{l} 5 \\ 7 \\ 6 \\ 10 \\ 8 \\ 3 \\ 1 \\ 2 \\ 12 \\ 4 \\ 11 \\ 9 \end{array} \quad \tilde{\pi}_{\mathbf{W}} = \begin{pmatrix} 0.0658 \\ 0.0498 \\ 0.0556 \\ 0.0442 \\ 0.0495 \\ 0.1118 \\ 0.2541 \\ 0.1683 \\ 0.0383 \\ 0.0744 \\ 0.0408 \\ 0.0474 \end{pmatrix} \begin{array}{l} 5 \\ 7 \\ 6 \\ 10 \\ 8 \\ 3 \\ 1 \\ 2 \\ 12 \\ 4 \\ 11 \\ 9 \end{array}$$

Figure 2. Ranking results of Approach 1 & 2

The first column in Figure 2 above is the list of global system states with their index numbers on the left-hand side. The middle vector $\pi_{\mathbf{W}}$ gives the rank values (PageRank values) we computed based on the transition matrix $\mathbf{W}$, and the column neighboring to the vector on the right-hand side gives the order numbers of the states ranked by their rank values.

Approach 2: On the other hand, as $\mathbf{Y}$ is already primitive, hence $\mathbf{W}$ is primitive as well. We can compute directly its stationary state distribution without applying the Maximal Irreducibility method. The resulting ranking is shown by the right vector $\tilde{\pi}_{\mathbf{W}}$ in Figure 2. We notice that apart from minor differences in the absolute values, the two results rank all system states in an identical order.

The results imply that, in the Layered Markov Model defined by $\mathbf{Y}, \mathbf{U}^1, \mathbf{U}^2$ and $\mathbf{U}^3$, the top three (highly ranked) overall system states are number 7, 8 and 6, namely (2,3), (3,1) and (2,2).

In both Approach 1 and Approach 2, we have to compute in advance the global transition matrix $\mathbf{W}$ in order to derive the ranking of the global system states, thus we consider these two as *centralized approaches* for computing the global system state ranking. The difference between them is that if the maximal irreducibility adjustment has to be performed on W.

2.3.3 Partition Theorem for Rank Computation

A natural question is now that given the PageRank ranking for all four matrices, $\mathbf{Y}, \mathbf{U}^1, \mathbf{U}^2$ and $\mathbf{U}^3$, is it possible to obtain the stationary distribution for the global system states without deriving a new matrix $\mathbf{W}$ and applying the PageRank algorithm to it?

We introduce now such an algorithm step-by-step:

1. At the phase level, if $\mathbf{Y}$ is already primitive, we can compute its stationary distribution $\tilde{\pi}_{\mathbf{Y}}$ without applying the maximal irreducibility method to $\mathbf{Y}$ before the power method is applied. The element for phase I in the distribution vector is denoted by $\tilde{\pi}_{\mathbf{Y}}(I)$.

   Certainly, we can also compute the slightly different $\pi_{\mathbf{Y}}$ by applying the maximal irreducibility method to $\mathbf{Y}$ even if $\mathbf{Y}$ is already primitive. We will see later on why we don't make this choice.

2. At the sub-state level within phases, for each phase I, we compute its stationary distribution π^I_G by applying the PageRank algorithm to $\mathbf{U}^I$. Remember this resulting vector is related to our introduced gatekeeper sub-state of each phase $\mathbb{P}_I$. We denote the element for sub-state i in the distribution vector by $\pi^I_G(i)$.

3. For each global system state (I, i), we assign it a value as follows:

$$\tilde{\pi}(I, i) = \tilde{\pi}_{\mathbf{Y}}(I)\pi^I_G(i) \tag{5}$$

   The assignments to all global system states form a state distribution π.

We call this the *Layered Method* of rank computation. The result of this computation has the following (expected) property.

Theorem 1. *The resulting vector of the Layered Method of rank computation is a probability distribution.*

Proof.

$$\sum_I \sum_i \tilde{\pi}(I, i) = \sum_I \sum_i \tilde{\pi}_{\mathbf{Y}}(I)\pi^I_G(i) = \sum_I \tilde{\pi}_{\mathbf{Y}}(I) \sum_i \pi^I_G(i) = 1$$

□

We give an example illustrating the computation: we want to compute the ranking value assigned to the global system state $7 : (2, 3)$.

Approach 3: The PageRank vector $\pi_{\mathbf{Y}}$ for Y is:

$$\pi_{\mathbf{Y}} = (0.2315, 0.4015, 0.3670)'$$

We can replace $\tilde{\pi}_{\mathbf{Y}}(I)$ in (5) with $\pi_{\mathbf{Y}}(I)$ and the result is still a probability distribution. The corresponding multiplication becomes:

$$\pi(2,3) = \pi_{\mathbf{Y}}(2)\pi_G^2(3) = 0.4015 \times 0.6117 = 0.2456$$

Unsurprisingly, this value is different from $\pi_{\mathbf{W}}(2,3)$ that we've computed before.

Approach 4 (the Layered Method): The vector $\tilde{\pi}_{\mathbf{Y}}$ for Y is:

$$\tilde{\pi}_{\mathbf{Y}} = (0.2154, 0.4154, 0.3692)'$$

Thus:

$$\tilde{\pi}(2,3) = \tilde{\pi}_{\mathbf{Y}}(2)\pi_G^2(3) = 0.4154 \times 0.6117 = 0.2541$$

Notice that this value is equal to that of $\tilde{\pi}_{\mathbf{W}}(2,3)$ we have obtained previously.

We call Approach 3 and Approach 4 the *decentralized approaches* for computing the global system state ranking, as we do NOT have to compute in advance the global transition matrix $\mathbf{W}$. Instead we compute the ranking for the phases (or Web sites for the case of Web document ranking), the individual rankings for the sub-states in each phase (or the individual Web document rankings for each Web site), which can be done in a parallel or decentralized fashion. The differences between Approach 3 and 4 is also whether the maximal irreducibility adjustment has to be performed on W.

Now we want to show the equality of the values obtained from Approach 2 and Approach 4 in the example is not accidental.

Corollary 1. *Approach 2 and Approach 4 (the Layered Method) are equivalent.*

This corollary results from the following theorem.

Theorem 2. *Give $\mathbb{LMM} = (\mathbb{P}, \mathbf{Y}, \mathbf{v}_Y, \mathbb{O}, \mathbb{U}, \mathbf{v}_{\mathbb{U}})$ as a Layered Markov Model where $\mathbf{Y}$ is primitive. The following vectors are first computed: the stationary state distribution vector $\tilde{\pi}_{\mathbf{Y}}$ of $\mathbf{Y}$, the PageRank vectors $\pi_G^I, I \in [1, N_{\mathbb{P}}]$. A new matrix $\mathbf{W}$ and a new vector $\tilde{\pi}$ are derived in the following fashion:*

1. *Both the size of $\mathbf{W}$ and the length of $\tilde{\pi}$ are $\mathcal{N}_{\mathbb{P}} = \sum_{I=1}^{N_{\mathbb{P}}} n_I$, i.e., the total number of the global system states in the model $\mathbb{LMM}$. Every element of $\mathbf{W}$ and every element of $\tilde{\pi}$ correspond to a global system state (I, i) ordered by $I \in [1, N_{\mathbb{P}}]$ and $i \in [1, n_I]$.*
2. *Every element of $\mathbf{W}$ is defined by $w_{(I,i)(J,j)} = y_{IJ}\pi_G^J(j)$.*
3. *Every element of $\tilde{\pi}$ is defined by $\tilde{\pi}(I,i) = \tilde{\pi}_{\mathbf{Y}}(I)\pi_G^I(i)$.*

Then $\mathbf{W}$ is also primitive and its stationary state distribution vector is exactly $\tilde{\pi}$.

Proof. For a primitive matrix, we know its stationary state distribution vector is the principal Eigenvector of its transposed matrix. Lemma 2 assures that $\mathbf{W}$ is primitive. Lemma 1 says $\mathbf{W}$ is Markovian, thus the principal Eigenvalue of $\mathbf{W}$ is 1. Then it remains to show

$$\mathbf{W}'\tilde{\pi} = \tilde{\pi}$$

which is equivalent to that, given (I, i),

$$\begin{aligned}
& \textstyle\sum_J \sum_j w_{(J,j)(I,i)}\tilde{\pi}(J,j) = \tilde{\pi}(I,i) \\
\Leftrightarrow \quad & \textstyle\sum_J \sum_j y_{JI}\pi_G^I(i)\tilde{\pi}_{\mathbf{Y}}(J)\pi_G^J(j) = \tilde{\pi}_{\mathbf{Y}}(I)\pi_G^I(i) \\
\Leftrightarrow \quad & \pi_G^I(i)\textstyle\sum_J y_{JI}\tilde{\pi}_{\mathbf{Y}}(J)\sum_j \pi_G^J(j) = \tilde{\pi}_{\mathbf{Y}}(I)\pi_G^I(i) \\
\Leftrightarrow \quad & \pi_G^I(i)\textstyle\sum_J y_{JI}\tilde{\pi}_{\mathbf{Y}}(J) = \tilde{\pi}_{\mathbf{Y}}(I)\pi_G^I(i) \\
\Leftrightarrow \quad & \textstyle\sum_J y_{JI}\tilde{\pi}_{\mathbf{Y}}(J) = \tilde{\pi}_{\mathbf{Y}}(I)
\end{aligned}$$

The last equality is guaranteed by the fact that $\tilde{\pi}_{\mathbf{Y}}$ is the stationary state distribution vector of $\mathbf{Y}$. □

We call Theorem 2 the *Partition Theorem for Rank Computation* as the rank computation for the global system states in a Layered Markov Model can be decomposed into several steps that can be performed in a decentralized or/and parallel fashion, if decomposability is assumed and the phase transition matrix is primitive.

The computation proceeds as follows: (1) At the phase layer, computation of the stationary distribution for the phase transition matrix. (2) At the sub-state layer, computation of the PageRank for individual sub-state stationary distribution for the sub-state transition matrix. (3) The aggregation of those vectors where only $O(\mathcal{N}_{\mathbb{P}})$ multiplications are necessary. In contrast, previous methods require a large number of multiplications of two $\mathcal{N}_{\mathbb{P}} \times \mathcal{N}_{\mathbb{P}}$ matrices until the resulting vector converges.

3. Application to Web Information Retrieval

We now discuss how the obtained theoretical results can be applied in the context of Web Information Retrieval. We know that search engines take into consideration both query-based ranking (for example, distances between queries and documents based on the Vector Space Model) and link-structure-based ranking (typically PageRank in Google and HITS-derived algorithm in Teoma) when ordering search results. We focus on the second aspect.

3.1. Different Abstractions for the Web Graph

Previous research work focused on the page granularity of the Web, i.e., a graph where the vertices are Web pages

and the edges are links among pages. We propose to model the Web graph at the granularity of Web site. We call the graph at the document level the *DocGraph*, and the graph at the Web site level the *SiteGraph*. We also use the notion of *SiteLink* to designate hyperlinks among Web sites and *DocLink* for those among Web documents. When the SiteGraph is created, to count the number of Sitelinks between two sites, we add the number of outgoing edges from any node in the first site to any node in the second site.

Given the graph of Web documents $G_D(V_D, E_D)$ with N_D pages as a DocGraph, we assume its corresponding SiteGraph is $G_S(V_S, E_S)$ with N_S Web sites in total, $v_s \in V_S$ is a Web site and $e_s \in E_S$ is a SiteLink. We use the notations $G_D(V_D, E_D), v_d, e_d$ for a DocGraph. We also use the shorthand d and s to represent a Web document and a Web site respectively. Taking one page d, we denote its corresponding site as $s = site(d)$ with $n_s = size(s)$ local Web documents in total. $V_d(s) \subseteq V_D$ is the set of all local Web pages of the particular Web site s. $E_d(s) \subseteq E_D$ is defined to be the set of those e_d whose both originating and destination documents are members of $V_d(s)$. $G_d^s = (V_d(s), E_d(s))$ is defined to be the subgraph restricted with the Web site s.

We call the ranking of Web sites the *SiteRank* for the SiteGraph and the ranking of Web documents the *DocRank* for the DocGraph. PageRank is an example of DocRank, but DocRank can be computed in a way other than PageRank, for example, as in our approach in a decentralized fashion. We use the notions SiteRank(G_S) and DocRank(G_D) to refer to the SiteRank result of G_S and DocRank result of G_D respectively. When we are using the matrix representations $\hat{\mathbf{M}}_S$ of G_S and $\hat{\mathbf{M}}_D$ of G_D, we also use SiteRank($\hat{\mathbf{M}}_S$) and DocRank($\hat{\mathbf{M}}_D$) to denote the rankings.

3.2. Layered Method for DocRank

Having the analytical results above, we compute the DocRank for a given Web graph in the following steps:

1. Derive the global DocGraph $G_D(V_D, E_D)$ from the given Web graph. Typically, DocLinks are processed.

2. Derive the global SiteGraph $G_S(V_S, E_S)$ from the DocGraph. Nodes in the SiteGraph are the Web sites. Edges are grouped together according to Web sites. The numbers of SiteLinks are counted.

3. For each Web site s, derive the subgraph G_d^s, its matrix representation $\hat{\mathbf{M}}_d^s = \hat{M}(G_d^s)$ and compute its $\pi_D(s)$ =DocRank($\hat{\mathbf{M}}_d^s$) using the classical PageRank algorithm. This step can be completely decentralized in a peer-to-peer search system.

4. For the global SiteGraph $G_S(V_S, E_S)$, we first derive a primitive transition matrix and then compute its principal Eigenvector. The primitivity of the transition probability matrix is required by Theorem 2. In practice, we compute $\hat{\mathbf{M}}_S = \hat{M}(G_S)$ which is primitive and its principal Eigenvector $\pi_S = (\pi_S(s_1), \cdots, \pi_S(s_{N_S}))'$ as the SiteRank.

5. For $i = 1, \cdots, N_S$, we list the N_D DocRank vectors $\pi_D(s_i)$ and create an aggregate vector from them:

   $$\pi_D = (\pi_D(s_1)', \cdots, \pi_D(s_{N_S})')'$$

   By applying Theorem 2, we perform a weighted product to obtain the final global ranking for all documents in the DocGraph $G_D(V_D, E_D)$:

   $$DocRank(G_D) = (\pi_S(s_1)\pi_D(s_1)', \cdots, \pi_S(s_{N_S})\pi_D(s_{N_S})')'$$

Personalization of rankings can be easily implemented in our layered method for DocRank. Personalization at the lower layer, i.e., the layer of local Web documents within specific Web sites, can be realized in Step 3 by providing different personalized vectors in the function body of $\hat{M}(G_d^s)$. Similarly, personalization at the higher layer, i.e., the layer of Web sites, can be realized in Step 4. Of course, it can also be applied to both layers at the same time.

When considering a peer-to-peer architecture the strategy for computing SiteRank and DocRank need to be considered. In a flat peer-to-peer architecture SiteRank could be a shared resource among all peers, i.e. globally available, which is realistic as its value changes less rapidly. DocRank computations are performed by individual peers, which would ideally map to Web servers. This would in particular open the possibility to obtain access to the hidden Web. Alternatively, super-peer architectures can be considered, where rank aggregation is only performed at super-peers and individual peers provide their local DocRanks.

Worthy of notice is the difference between the *block graph* in [9] and our SiteGraph although they look similar. The main difference is the weight assignment to edges among blocks. In BlockRank such an assignment is dependent on earlier stage of computation. The weight of the single edge between two blocks is the sum of local PageRank values of the source pages in the source block. In other words, the edge weights are bound with the local PageRank values and the computation has to be serialized. In our model, only the number of SiteLinks is used thus the computations of SiteRank and local DocRanks can be done in parallel.

3.3. Empirical Results

We made some initial experiments on a recently crawled snapshot of our campus Web. We started from the home

page of the university, *www.epfl.ch*, and let the crawler follow the hyperlinks and retrieve Web pages. Different from many other previous published experiments, we did not exclude dynamic Web pages generated by server-side scripts. The reason is that nowadays most Web sites use them as a powerful vehicle to provide dynamic and fresh information. Without including them, the captured Web graph would be a rather skewed one. However, crawling dynamic pages often causes an infinite loop for all kinds of possibilities. To avoid this, researchers usually let the crawler run and then stop it after it has been running for a period of time.

Our partial Campus Web graph was captured in late 2003. In this graph there are 218 Web sites and 433707 Web pages altogether. We follow the steps described in Section 3.2 to compute the SiteRank of the SiteGraph of this partial Campus Web, the DocRanks of every site, and finally the global DocRank for all pages in this partial Campus Web. The result is presented in Figure 4. To make comparison, we also apply the PageRank algorithm to the set of all Web pages to obtain the PageRank for them. The result containing the top 15 entries is presented in Figure 3. The left columns are the lists of the document identifiers with their corresponding URLs in the right columns. Documents are listed in descending order of the computed rank values.

16	http://www.epfl.ch/
1737	http://research.epfl.ch/
73612	http://research.epfl.ch/research/Webdriver?LO=...
73613	http://research.epfl.ch/research/Webdriver?MIval=...
73614	http://research.epfl.ch/research/Webdriver?LO=...
18282	http://research.epfl.ch/research/Webdriver?MIval=...
677	http://www.epfl.ch/place.html
570	http://www.epfl.ch/styles/dynastyle.php
459683	http://dmawww.epfl.ch/roso.mosaic/ismp97/...
73635	http://research.epfl.ch/research/Webdriver?LO=...
73636	http://research.epfl.ch/research/Webdriver?MIval=...
73637	http://research.epfl.ch/research/Webdriver?LO=...
122990	http://lamp.epfl.ch/~linuxsoft/java/jdk1.4/docs/...
90330	http://lampwww.epfl.ch/~linuxsoft/java/jdk1.4/docs/...
614	http://sti.epfl.ch/

Figure 3. Result by PageRank

In Figure 3, the top entries of the PageRank result are dominated by some pages which share an identical URL prefix. Further investigation shows that all of them have a huge in-degree number. For example, the dynamic page 73612 has 17004 incoming links and most of its originating pages have the same URL prefix

```
http://research.epfl.ch/research/Webdriver?
```

which means they are generated by the same server-side script and heavily linked among each other. Similarly, the static page 122990 has 6425 incoming links and most of its originating pages have as well the same URL prefix

```
http://lamp.epfl.ch/~linuxsoft/java/jdk1.4/docs/
```

which means they are all javadocs of jdk1.4 and also heavily linked among each other.

It seems that the agglomerate structure of these document sets boosts drastically their PageRank values and this fact has been widely exploited by spammers such that even a new business has been created to make the most out of it.

16	http://www.epfl.ch/
570	http://www.epfl.ch/styles/dynastyle.php
677	http://www.epfl.ch/place.html
73324	http://satellite.epfl.ch/
2196	http://lcsmwww.epfl.ch/
153	http://cssa.epfl.ch/
572	http://www.epfl.ch/150/
2884	http://sti.epfl.ch/news/AG/AG-Faculte-STI08.html
73446	http://mysearch.epfl.ch/help/?la=fr
678	http://www.epfl.ch/niceberg/content/1/
581517	http://smte.epfl.ch/francais/impressum.php
71973	http://spi.epfl.ch/Jahia/site/spi/cache/offonce/pid/...
71975	http://spi.epfl.ch/page33282.html
681	http://www.epfl.ch/impressum.html
71961	http://vpf.epfl.ch/

Figure 4. Result by LMM-based Method

On the other hand, the ranking result computed by our Layered Method based on LMM gives a very neat list of entries which really cover many authoritative aspects of the university, such as central place (677), student bar (73324), student organization (153), 150 anniversary page (572), faculty of engineering (2884), search (73446), news (678), internal journal (71973 and 71975), press information (681), vice presidency of education (71961), etc..

In the Layered Method, the role played by the entangled cross links has been made much less important due to the effect of introducing the SiteRank of the owner Web site as a crucial part of the final global ranking for a particular Web page. It shows that the global page ranking algorithm is not necessarily the best possible ranking method. It demonstrates the capability of the LMM model to defeat link spamming to a very satisfiable degree.

4. Summary and Future Work

Applying the peer-to-peer architectural paradigm to Web search engines has recently become a subject of intensive research [5, 12, 16, 15]. Most recent work has been focusing on key-based retrieval. Among those, there are two main

categories: P2P-IR using structured overlay networks and P2P-IR using unstructured overlay networks. To make an IR system scalable, especially at the Web scale, is one of the key arguments why P2P approaches are adopted.

Whereas for the decomposition of content-based retrieval techniques, such as classical text-based vector space retrieval or latent semantic indexing, various proposals have been made, the decomposition of rank computation based on the link structure of the Web is less clear. This is also one of the reasons that the combination of link-based ranking with content-based methods in P2P search has not been investigated closely yet.

In this paper, we introduce a novel link-structure analysis method based on a Layered Markov Model. Our model differs substantially from the classic rank computation models that consider a flat Web graph. Our model makes use of the inherent hierarchical logical structure of the Web and the self-similar character of the Internet. We provide a strict analysis of our model for the Web ranking problem and give the Partition Theorem for Rank Computation. Such a formal result backs up theoretically the rank computation of the Internet-scale Web graph in a completely distributed way. This removes the radical obstacle and limitation that the existing algorithms have to suffer in terms of requiring global computation. Empirical experiments give good results and show that link spamming which has been a headache for some global ranking algorithm is also nicely defeated to a very satisfiable degree.

Future work includes in particular the investigation of the retrieval performance of LMM-based algorithms. Though it seems this has not been properly addressed for the standard, centralized methods like PageRank and HITS either, it seems important to improve insight into this question. To that end experiments with the TREC collection are planned and work of combining query-based ranking and link-based ranking will also be carried out.

References

[1] J. Adibi and W.-M. Shen. Self-similar layered hidden Markov models. *Lecture Notes in Computer Science*, 2168:1–15, 2001.

[2] A. Z. Broder, R. Lempel, F. Maghoul, and J. Pedersen. Efficient pagerank approximation via graph aggregation. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 484–485. ACM Press, 2004.

[3] S. Dill, S. R. Kumar, K. S. McCurley, S. Rajagopalan, D. Sivakumar, and A. Tomkins. Self-similarity in the web. In *The VLDB Journal*, pages 69–78, 2001.

[4] A. Farahat, T. LoFaro, J. C. Miller, G. Rae, and L. A. Ward. Existence and Uniqueness of Ranking Vectors for Linear Link Analysis Algorithms. *SIAM Journal on Scientific Computing*, (submitted), 2003.

[5] O. D. Gnawali. A keyword-set search system for peer-to-peer networks. Master's thesis, Department of Electrical Engineering and Computer Science, MIT, May 2002.

[6] T. H. Haveliwala. Efficient computation of pageRank. Technical Report 1999-31, Stanford University Database Group, Sept. 1999.

[7] S. Kamvar, T. Haveliwala, and G. Golub. Adaptive methods for the computation of pagerank. Technical report, Stanford University, 2003.

[8] S. Kamvar, T. Haveliwala, C. Manning, and G. Golub. Extrapolation methods for accelerating pagerank computations. In *Proceedings of the Twelfth International World Wide Web Conference*, 2003.

[9] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub. Exploiting the block structure of the web for computing pagerank. Technical report, Stanford University, Mar. 2003.

[10] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[11] A. N. Langville and C. D. Meyer. Deeper inside pagerank. *Internet Mathematics*, Feb. 2004.

[12] J. Li, B. T. Loo, J. M. Hellerstein, M. F. Kaashoek, D. R. Karger, and R. Morris. On the feasibility of peer-to-peer web indexing and search. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems*, Berkeley, California, USA, 2003.

[13] C. D. Meyer, editor. *Matrix analysis and applied linear algebra*. Society for Industrial and Applied Mathematics, 2000.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, Jan. 1998.

[15] C. Tang, S. Dwarkadas, and Z. Xu. On scaling latent semantic indexing for large peer-to-peer systems. In *Proceedings of the 27th Annual International ACM SIGIR Conference*, Sheffield, UK, July 2004.

[16] C. Tang, Z. Xu, and S. Dwarkadas. Peer-to-peer information retrieval using self-organizing semantic overlay networks. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 175–186. ACM Press, 2003.

[17] Y. Wang and D. J. DeWitt. Computing pagerank in a distributed internet search system. In *Proceedings of the 30th International Conference on Very Large Data Bases*, pages 420–431. Morgan Kaufmann Publishers Inc., 2004.

[18] J. Wu and K. Aberer. Using a layered markov model for decentralized web ranking. Technical Report IC/2004/70, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, Aug. 2004.

[19] J. Wu and K. Aberer. Using siterank for decentralized computation of web document ranking. In *Proceedings of the third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Eindhoven, The Netherlands, Aug. 2004.

[20] J. Wu and K. Aberer. Using siterank in P2P information retrieval. Technical Report IC/2004/31, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, Mar. 2004.