A Learning Algorithm for Web Page Scoring Systems

Michelangelo Diligenti, Marco Gori, Marco Maggini
Dipartimento di Ingegneria dell'Informazione, Universita di Siena, Italy
{michi, marco, maggini}@dii.unisi.it

Abstract

Hyperlink analysis is a successful approach to define algorithms which compute the relevance of a document on the basis of the citation graph. In this paper we propose a technique to learn the parameters of the page ranking model using a set of pages labeled as relevant or not relevant by a supervisor. In particular we describe a learning algorithm applied to a scheme similar to PageRank. The ranking algorithm is based on a probabilistic Web surfer model and its parameters are optimized in order to increase the probability of the surfer to visit a page labeled as relevant and to reduce it for the pages labeled as not relevant. The experimental results show the effectiveness of the proposed technique in reorganizing the page ordering in the ranking list accordingly to the examples provided in the learning set.

1 Introduction

PageRank [Page *et al,* 1998] is the most popular algorithm to compute the page relevance in a collection of hyperlinked documents, like the Web. This algorithm exploits the topology of the citation graph in order to determine the authority of each page using the intuitive idea that authoritative documents are linked by many authoritative documents. Thus, the original algorithm computes a ranking which depends only on the graph connectivity without considering the page contents. However, focused versions of PageRank have been proposed in the literature [Haveliwala, 2002; Richardson and Domingos, 2002; Diligenti *et al,* 2002] in order to define more selective rankings for topic-specific search engines. These approaches allow a limited degree of adaptation related to the choice of the specific topic. A text classifier is trained using examples of pages on the topic of interest but no further level of learning is considered in the ranking models. In particular, the classifier is trained independently on the ranking algorithm.

We propose a more general approach in which the learning algorithm is defined directly on the ranking model. In this case there is not a predefined topic, but a supervisor provides examples of pages to be promoted and pages which are considered as not relevant. Thus, the learning algorithm is able to manage both positive examples (relevant pages) and negative examples (not relevant pages). The ranking function is based on the Web surfer model [Diligenti *et al.* 2002]. The relevance of a page is defined by the probability of the surfer to visit the page when considering the stationary probability distribution of the Markov chain which describes the surfer behavior. Basically the learning algorithm computes the parameters which define the surfer model in order to reduce the probability of visiting a negative example and to increase the probability of being in a positive example. In order to reduce the number of free parameters, the surfer model is simplified by dividing the pages among a set of categories and assuming that the surfer behavior depends only on the category of the page where it is located. In particular, the set of categories does not need to be predefined but it is determined by a clustering algorithm based on the document contents. Moreover, whereas the focused versions of the PageRank simply direct the Web surfer to the most promising page (i.e. the page whose content is the most similar to the content of a target page), the proposed algorithm can exploit hierarchies of topics, allowing the surfer to follow complex paths, composed by many steps, leading to relevant pages.

The paper is organized as follows. In the next section the web surfer model is reviewed. Then, in section 3 the learning algorithm is described. Section 4 reports some experiments which show the effectiveness of the proposed learning algorithm. Finally, in section 5 the conclusions are drawn.

2 Random Walks on the Web graph

Random walk theory has been widely used to compute the absolute relevance of a page in the Web [Page *et al,* 1998; Lempel and Moran, 2000]. The Web is represented as a graph G, where each Web page is a node and a link between two nodes represents a hyperlink between the associated pages. A common assumption is that the relevance I_p of page p is represented by the probability of ending up in that page during a walk on this graph.

We consider the model of a Web surfer who can perform one out of two atomic actions while visiting the Web graph: jumping to a node of the graph (action *J)* or following a hyperlink from the current page (action *I).* In general, the action taken by the surfer will depend on the page contents and the links it contains. We can model the user's behavior by a set of probabilities which depend on the current page:

$x(l|q)$ is the probability of following one hyperlink from page *q*, and $x(J|q)$ is the probability of jumping from page *q*. These values must satisfy the normalization constraint, $x(J|q) + x(l|q) = 1$.

The two actions need to specify their targets. Assuming that surfer behavior is time-invariant, then the targets are specified by the probability $x(p|q, J)$ of jumping from page *q* to page p, and the probability $x(p|q, l)$ of selecting a hyperlink from page *q* to page *p*. $x(p|q, l)$ is not null only for the pages *p* linked directly by page *q*, i.e. $p \in ch(q)$, being *ch(q)* the set of the children of node *q* in the graph *G*. These sets of values must satisfy the probability normalization constraints $\sum_{p \in G} x(p|q, J) = 1, \ \forall q \in G$ and $\sum_{p \in ch(q)} x(p|q, l) = 1, \ \forall q \in G$.

The model considers a temporal sequence of actions performed by the surfer and it can be used to compute the probability that the surfer is located in page *p* at time t, $x_p(t)$. The probability distribution on the pages of the Web is updated by taking into account the actions at time $t+1$ using the following equation

$$x_p(t+1) = \sum_{q \in G} x(p|q, J) \cdot x(J|q) \cdot x_q(t) + \quad (1)$$
$$+ \sum_{q \in pa(p)} x(p|q, l) \cdot x(l|q) \cdot x_q(t) ,$$

where $pa(p)$ is the set of the parents of node *p*. The relevance of a page p, x_p is computed using the stationary distribution of the Markov Chain of equation (1). This is a simplified version of the model proposed in [Diligenti *et al,* 2002].

3 Learning the page rank

The random surfer model which is used to compute the page relevance scores depends on the values assigned to the probabilities $x(p|q, J)$, $x(J|q)$, $x(p|q, l)$, and $x(l|q)$. In most of the approaches proposed in the literature, these parameters are predefined or computed from some features describing the page *p* and/or the page *q*. For example, in the original PageRank scheme $x(l|q) = d$, $x(J|q) = 1 - d$, $x(p|q, J) = \frac{1}{N}$, and $x(p|q, l) = \frac{1}{ch(q)}$.

However, in the general case, it would be infeasible to estimate the parameters without any assumption to reduce their number. In particular, we assume that the pages are grouped into *n* clusters according to their content and that the surfer's behavior for any page *p* is dependent only on the cluster which *p* belongs to. In our experiments we used classical clustering techniques (e.g. k-means) on the bag-of-words representation of the pages [Kobayashi and Takeda, 2000]. By this hypothesis, the parameters of the model are computed as:

$$x(p|q, l) = \frac{r(c^q \to c^p|l)}{\sum_{k \in ch(q)} r(c^q \to c^k|l)} \quad (2)$$
$$x(p|q, J) = \frac{x(c^p|c^q, J)}{|c^p|}$$
$$x(l|p) = x(l|c^p)$$
$$x(J|p) = x(J|c^p)$$

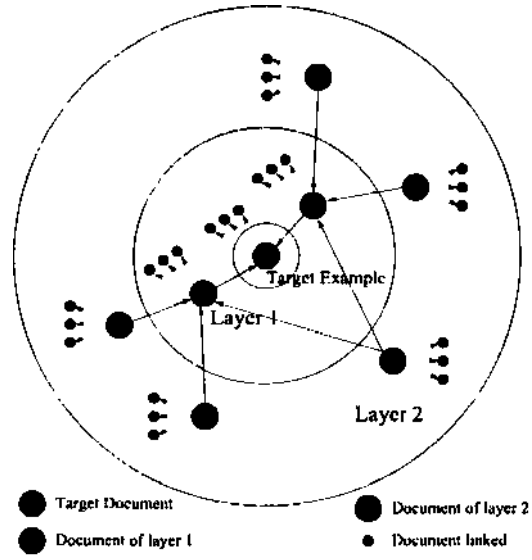

Figure 1: The context graph built by following the links backwards from an example page.

where c^q is the cluster which page *q* belongs to, $|c^p|$ is the number of pages in cluster c^p, the parameter $r(c^q \to c^p|l)$ is the tendency of the surfer to follow a hyperlink from a page in cluster c^q to a page in cluster c^p, and the parameter $x(c^p|c^q, J)$ is the tendency of the surfer to jump from a page of cluster c^q to a page of cluster c^p. The values $r(c^q \to c^p|l)$ are initialized to 1, such that all links are equally likely to be followed. Because of the model assumptions, the stronger is the tendency of the surfer to move to a cluster, the higher are the scores associated to the pages in that cluster.

3.1 The learning set

We assume that a supervisor provides a set pages labeled as *relevant* and *not relevant.* These examples are used to define a *specific* view of the collection of documents contained in a search engine. In particular the supervisor provides:

- a set of pages N^+ which should get a high rank;
- a set of pages N^- which should get a low rank.

In the first case we refer to "positive" or "good" pages, whereas in the latter case to "negative" or "bad" pages.

Each example page is associated to its Context Graph [Diligenti *et al,* 2000], representing how the page can be reached from the Web. The graph is organized into layers as shown in figure 1. The example page is the only node in layer 0. Then, each node in layer $i > 0$ links only nodes in layer $i-1$. Thus, a page belongs to layer *i* if it is at least *i* links away from the example page. The *depth K* of the context graph is defined as the number of layers in the graph including level 0. For a given example, the context graph can be built by backcrawling the Web adding to layer $i+1$ the pages which link the pages in layer i, up to a maximum level *K*. Moreover, all the pages that are linked by at least a page of the graph are also considered.

The learning set consists of the context graphs built from the examples in N^+ and N^-. $\mathcal{CG}$ is the set of all the pages contained in the learning set; L_k^+ (L_k^-) is the union of the pages in the layer *k* of the context graphs of the positive (negative) examples.

3.2 Learning the transition parameters

We define the set of transition probabilities $\mathcal{T} = \{x(p \in c_j | q \in c_i, l)\ i,j = 1,\ldots,n\}$, which represent the probabilities of the surfer to select a hyperlink from a page in the cluster c_i to a page in the cluster c_j. The rank of the pages in the set N^+ is increased, whereas the rank of the pages in N^- is decreased if we increase the probability of the paths leading to pages in N^+ and we reduce the probability of paths leading to pages in N^-. This effect can be obtained by maximizing the following cost function

$$y(\mathcal{T}) = \sum_{k=0}^{K-1} \alpha^k \left[x(p \in L_k^+ | q \in L_{k+1}^+, \mathcal{T}, l) + + (1 - x(p \in L_k^- | q \in L_{k+1}^-, \mathcal{T}, l))\right] , \quad (3)$$

where $0 \leq \alpha \leq 1$ is a discount factor which is used to reduce the impact on the cost function of the links too far from the target page.

The probability $x(p \in L_k^+ | q \in L_{k+1}^+, \mathcal{T}, l)$ of moving one level in the context graph towards a good page, can be rewritten as

$$x(p \in L_k^+ | q \in L_{k+1}^+, \mathcal{T}, l) = \quad (4)$$

$$= \sum_{i=1}^{n} x(p \in L_k^+ | q \in L_{k+1}^+, q \in c_i, \mathcal{T}, l) \cdot \cdot x(q \in c_i | q \in L_{k+1}^+, l) =$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} x(p \in L_k^+ | q \in L_{k+1}^+, q \in c_i, p \in c_j, l) \cdot \cdot x(p \in c_j | q \in c_i, l) \cdot x(q \in c_i | q \in L_{k+1}^+, l) ,$$

where $x(p \in c_j | q \in c_i, l)$ reflects the surfer model assumption that the links are selected depending only on the clusters c_i and c_j. The probability $x(p \in L_k^+ | q \in L_{k+1}^+, q \in c_i, p \in c_j, l)$ of following a link to a page in the cluster c_3 of layer k: from a page in the cluster c_i of layer $k+1$ of a context graph can be estimated as

$$x(p \in L_k^+ \mid q \in L_{k+1}^+, q \in c_i, p \in c_j, l) = \quad (5)$$

$$= x(p \in L_k^+ \cap c_j) | q \in L_{k+1}^+ \cap c_i, l) =$$

$$= \frac{n(L_{k+1}^+ \cap c_i \rightarrow L_k^+ \cap c_j)}{n(L_{k+1}^+ \cap c_i \rightarrow c_j)} ,$$

where $n(A \rightarrow B)$ is the number of links from the pages in the set *A* to the pages in the set *B*.

On the other hand the term $x(q \in c_i | q \in L_{k+1}^+, l)$ represents the probability that the surfer is located in a page of the cluster c_i, given that the page belongs to the layer $k+1$ of a positive context graph. This probability can be estimated using the scores of the pages in cluster c_i and in L_{k+1}^+, computed using the current values of the parameters, as

$$x(q \in c_i | q \in L_{k+1}^+, l) = \frac{\sum_{p \in L_{k+1}^+ \cap c_i} x_p}{\sum_{p \in L_{k+1}^+} x_p} . \quad (6)$$

Similar relationships can be derived for the negative set.

The cost function (3) can be optimized by updating the parameters in $\mathcal{T}$ using gradient ascent. The gradient with respect to the transition parameters $x(p \in c_j | q \in c_i, l)$ can be computed using equation (4), yielding

$$\begin{aligned} \nabla_k^{ij} y(\mathcal{T}) = & \ x(p \in L_k^+ | q \in L_{k+1}^+, q \in c_i, p \in c_j, l) \cdot \\ & \cdot x(q \in c_i | q \in L_{k+1}^+, l) - \quad (7) \\ & - x(p \in L_k^- | q \in L_{k+1}^-, q \in c_i, p \in c_j, l) \cdot \\ & x(q \in c_i | q \in L_{k+1}^-, l) , \end{aligned}$$

where we neglected that $x(q \in c_i | q \in L_{k+1}^+, l)$ and $x(q \in c_i | q \in L_{k+1}^-, l)$ actually depend on the transition parameters. Since the model parameters $r(c_i \rightarrow c_j | l)$ are essentially a not normalized version of the transition parameters $x(p \in c_j | q \in c_i, l)$, we can update them directly as

$$r(c_i \rightarrow c_j | l)' = r(c_i \rightarrow c_j | l) + \eta \sum_{k=0}^{K-1} \alpha^k \nabla_k^{ij} y(\mathcal{T}) , \quad (8)$$

where η is the learning rate.

Once the model parameters are updated, a new score distribution can be computed, thus updating the estimate of the variables $x(q \in c_i | q \in L_{k+1}^+, l)$ using equation (6). Therefore, the function optimization and the score estimation can be iterated, using an EM style algorithm [Dempster *et al.*, 1977], till a termination criterion is satisfied.

3.3 Learning the jump parameters

The transition parameters for a jump can be optimized to increase the probability of jumping to a "good" page and not to a "bad" page. This effect is obtained by maximizing the cost function

$$y(\mathcal{P}) = x(p \in N^+ | \mathcal{P}, J) + \left[1 - x(p \in N^- | \mathcal{P}, J)\right] , \quad (9)$$

where $\mathcal{P} = \{x(p \in c_j | q \in c_i, J)\ i,j = 1,\ldots,n\}$ is the set of the jump parameters.

We start observing that,

$$\begin{aligned} x(p \in N^+ \mid \mathcal{P}, J) = & \\ = & \sum_{i=1}^{n} x(p \in N^+ | q \in c_i, J) \cdot x(q \in c_i | l) = \\ = & \sum_{i=1}^{n}\sum_{j=1}^{n} x(p \in N^+ | q \in c_i, p \in c_j, J) \cdot \\ & x(p \in c_j | q \in c_i, J) \cdot x(q \in c_i | J) . \end{aligned}$$

By assuming that the jump target is independent on the originating cluster c_i, the term $x(p \in N^+ | q \in c_i, p \in c_j, J)$ can be estimated as

$$x(p \in N^+ | q \in c_i, p \in c_j, J) = x(p \in N^+ | p \in c_j, J) = = \frac{|N^+ \cap c_j|}{|\mathcal{CG}|} \quad (11)$$

Moreover, the value $x(q \in c_i | J)$ is computed at each iteration as

$$x(q \in c_i | J) = x(q \in c_i) = \sum_{p \in c_i} x_p \quad (12)$$

Similar relationships can be derived for the negative set.

The cost function (9) is maximized by gradient ascent. The gradient can be computed using equation (10) and the equivalent equation for the negative set, yielding

$$\nabla^{ij}y(\mathcal{P}) = \{x(p \in N^+|q \in c_i, p \in c_j, J) - x(p \in N^-|q \in c_i, p \in c_j, J)\} \cdot x(q \in c_i|J), \quad (13)$$

where we neglected the dependence of $x(q \in d \setminus J)$ on the parameters. At each iteration, the parameters are updated using the direction of the gradient as,

$$x(p \in c_j|q \in c_i, J)' = x(p \in c_j|q \in c_i, J) + \eta \nabla^{ij}y(\mathcal{P}), \quad (14)$$

being r; the learning rate, and then normalized to meet the probabilistic constraints

$$x(p \in c_j|q \in c_i, J) = \frac{x(p \in c_j|q \in c_i, J)'}{\sum_{f=1}^{n} x(p \in c_j|q \in c_f, J)'}. \quad (15)$$

The adapted surfer model can re-surf the Web graph yielding a new estimate of equation (12). Thus, the new gradient estimate can be used to update again the parameters, till the stop criterion is satisfied.

3.4 Learning the surfer action bias

The surfer action bias is represented by the value of $x(l|q \in c_i)$ which is the probability of following a link from the current page rather than jumping to another page $(x(J|q \in c_i) = 1 - x(l|q \in c_i))$. In order to favor the action leading to good pages, we maximize the following cost function which represents a weighted average of the probabilities of ending in the layer k of a positive context graph and not of a negative context graph,

$$y(Q) = \sum_{k=0}^{K-1} \alpha^k \left[x(p \in L_k^+|Q) + (1 - x(p \in L_k^-|Q))\right] \quad (16)$$

$$Q = \{x(l|q \in c_i) \;\; i = 1, \ldots, n\}$$
$$0 \leq \alpha \leq 1$$

$$\begin{aligned} x(p \in L_k^+|Q) &= \sum_{i=1}^{n} x(p \in L_k^+|q \in c_i) \cdot x(q \in c_i) = \\ &= \sum_{i=1}^{n} \big(x(p \in L_k^+|q \in c_i, l) \cdot \\ &\quad x(l|q \in c_i) \cdot x(q \in c_i) + \\ &+ x(p \in L_k^+|q \in c_i, J) \cdot \\ &\quad (1 - x(l|q \in c_i)) \cdot x(q \in c_i)\big). \end{aligned} \quad (17)$$

$$x(p \in L_k^+|q \in c_i, l)$$

$$\begin{aligned} x(p \in L_k^+ \;|\; q \in c_i, l) = \\ = x(p \in L_k^+|q \in L_{k+1}^+, q \in c_i, l) \cdot \end{aligned} \quad (18)$$

$$\begin{aligned} x(q \in L_{k+1}^+, |q \in c_i, l) = \\ = \frac{n(L_{k+1}^+ \cap c_i \to L_k^+)}{n(L_{k+1}^+ \cap c_i \to \mathcal{CG})} \cdot \frac{|L_{k+1}^+ \cap c_i|}{|c_i|}, \end{aligned}$$

where we assumed that all links out of a page are equally likely to be followed. The term $x(p \in L_k^+|q \in c_i, J)$ is the probability of jumping to the layer fc of a positive context graph from a page of cluster i, and can be computed as

$$x(p \in L_k^+|q \in c_i, J) = \frac{|L_k^+|}{|\mathcal{CG}|} \quad (19)$$

Finally, $x(q \in c_i)$ can be computed at each iteration as in equation (12). Similar relationships hold for the negative set.

The gradient of the cost function (16) can be computed using equation (17) and the corresponding relationship for the negative set, yielding

$$\nabla_k^i y(Q) = \{x(p \in L_k^+|q \in c_i, l) - x(p \in L_k^+|q \in c_i, J) - x(p \in L_k^-|q \in c_i, l) + x(p \in L_k^-|q \in c_i, J)\} \cdot x(q \in c_i),$$

neglecting the dependence of $x(q \in c_i)$ on $x(l|q \in c_i)$.

I he parameters are updated by gradient ascent as

$$x(l|q \in c_i)' = x(l|q \in c_i) + \eta \sum_{k=0}^{K-1} \alpha^k \cdot \nabla_k^i y(Q), \quad (20)$$

where the learning rate must be chosen small enough in order to preserve the stochastic model of the surfer.

The adapted surfer model can re-surf the Web graph yielding a new estimate of equation (12). Thus, the new gradient estimate can be used to update again the parameters and the procedure is iterated till the stop criterion is satisfied.

4 Experimental results

In the following experiments, the learning algorithm was applied only to the transition parameters of the random surfer model. The experiments were performed on two datasets, each containing 1.000.000 documents, which were collected by focus crawling the Web on the topics "wine" and "playstation", respectively. The documents were clustered using a hierarchical version of the k-means algorithm [Fukunaga, 1990]. We considered different numbers of clusters generated by the k-means algorithm. In particular, in two different runs, we clustered the pages from the dataset on topic "playstation" into 16 and 25 sets, whereas the dataset on topic "wine" was clustered into 25 and 100 sets.

Two topic-specific search engines were build creating the inverted lists containing the terms of all the documents. Then, the rank of each page in the dataset was computed using the model described in section 2 by setting its parameters to reproduce the original PageRank ranking. We tested the quality of the ranking function by submitting a set of queries to this focused search engine. The documents matching the query were sorted by the scores assigned by the random surfer. By interacting with the search engine, wc found pages which were authorities on the specific topic but were (incorrectly) not inserted in the top positions of the result list. Such pages were selected as examples of pages which should get a higher

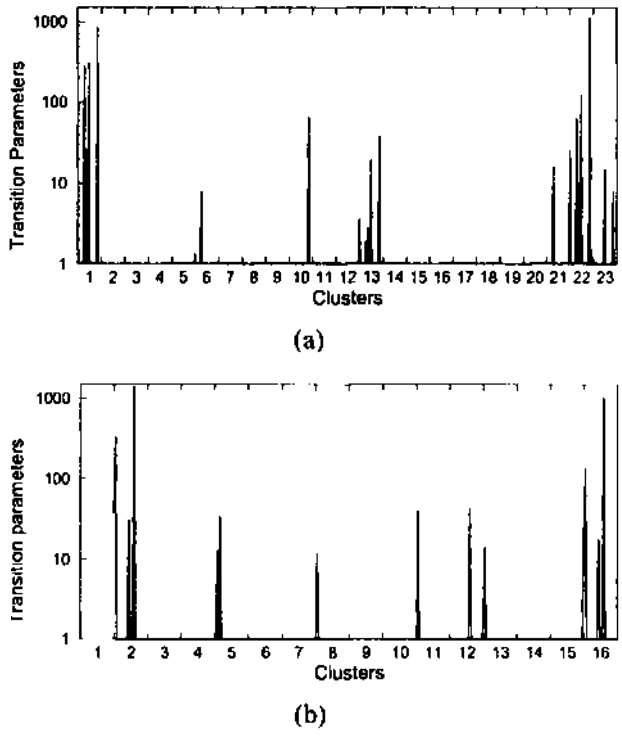

Figure 2: Plots of the values of the transition parameters for each cluster, resulting from the training of the surfer, (a) "wine" dataset. (b) "playstation" dataset.

rank (positive examples). On the other hand, we found pages having a high score which were not authorities on the topic. Such pages were selected as negative examples. The learning algorithm, as described in section 3, was applied to the datasets, using the selected examples. In the experimental setup only a small set of examples (3-15) was typically labelled as positive or negative.

4.1 Analysis of the effects of learning

In a first group of experiments we aimed at demonstrating that the surfer model as learnt by the proposed training algorithm, could not be generated by a simpler (but less powerful) schema as the focused versions of PageRank, proposed in the literature.

In figure 2-(a) 2-(b), we show the values of the transition parameters for each cluster, resulting from the training sessions for a surfer on the topic "wine" and "playstation", respectively. The learnt parameters represent a complex interaction of the resulting surfer with the page clusters. This interaction could not be expressed by a simple schema as, for example, the focused versions of PageRank proposed in the literature [Richardson and Domingos, 2002; Diligenti *et al.,* 2002]. This demonstrates that the model, in order to learn the users' feedback, needed to exploit hierarchies of topics to model the complex path leading either to positive or negative examples.

In figure 3-(a) and 3-(b), we show for each cluster, the values of the probability of the surfer of being located in a page

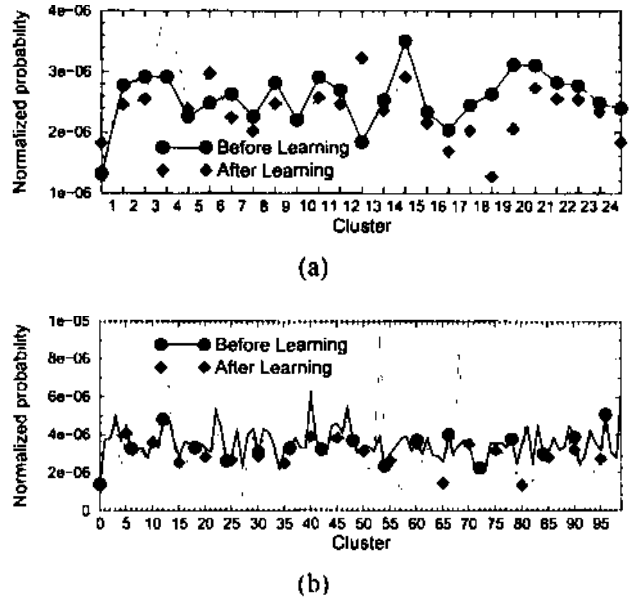

Figure 3: Plots of the probability of the surfer of being located in a page of a given cluster for the topic "wine". The probabilities are normalized with respect to the number of pages in the cluster, (a) 25 clusters, (b) 100 clusters.

of that cluster. In particular we used the dataset on topic "wine" clustered into 25 and 100 groups, respectively. The probabilities arc normalized with respect to the number of pages in the cluster. The learning algorithm is able to increase the likelihood of the random surfer to visit pages belonging to a set of clusters, while decreasing its probability of visiting pages belonging to other clusters.

In figure 4, we report the rank of pages after the learning session with respect of their rank before learning. In particular, on the *x* axis the pages are sorted according to their page rank before the learning session. The closer a page to the origin, the more "relevant" the page is. On the *y* axis it is reported the corresponding page rank after the learning session. The plot shows the generalization capability of the learning algorithm, which is able to change the ranks of a significant percentage of pages, even if a small set of example was provided.

4.2 Qualitative results

After training the surfer and computing the scores for all pages in the dataset, we compared the rank before and after the training process.

Figure 5 reports the pages which obtained the largest variation in their rank. The pages that obtained a negative rank variation were either topic-generic authoritative pages (e.g. www.netscapc.com, www.yahoo.com) or pages relevant for different topics than "wine" (e.g. "www.forgottensoldier-.org"). Only the page "www.yahoo.com" was explicitly provided to the system as a negative example. On the other hand, the pages that obtained a positive rank variation were effectively authoritative pages on the topic "wine" (e.g. "www-.wine-searcher.com" or "webwinery.com"). Among the most

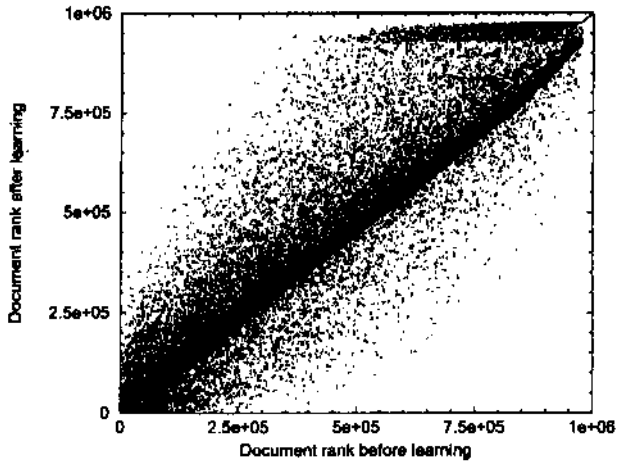

Figure 4: Plot of the page ranks before *(x* axis) versus after learning *(y* axis). The closer a page is to the origin, the most "relevant" it is.

Mos

p://www macromedia.coni/go/getflashplayer
http://wwwt descending pages on topic "wine'
htt.netscape.com
http://www.macromedia.com/downloads
http://www.yahoo.com
http://www.adobe com/products/acrobat/readstep.html
http://help.yahoo.com
http://www.nl.placestostay.com/mdex.html
http://home.netscape.com
http://www.forgottensoldier.com
http://www.inntravels.com

Most ascending pages on topic "wine"

http.//www. winccountry.com
http://webwinery.com
http://www.ny-wine.com/winemfo.asp
http://www.wine-searcher.com
http.//www. insidcwinc.com
http.//webwinery.com
http.//www.vinicsaporidipughacom/en/index.html
http://www.vinitaly.com/home.en.asp
http//www. wmeinstitute.org
http://www.winespectator.com

Figure 5: URLs of the pages that yielded either the largest negative or positive rank changes among the pages that were initially in the top 1000 scoring documents.

ascending documents, only the page "www.winespectator-.com" was explicitly provided to the system as a positive example, whereas all other pages were pushed up in the rank for the capability of the system to generalize from a small set of training data.

5 Conclusions

In this paper we introduced a novel algorithm to learn the ranking function over a collection of Web pages. The proposed framework allows us to tune a previously computed rank distribution, specifying which pages should get a higher or lower score. Like in the original PageRank, we assume that the score of a page is proportional to the probability that a Web surfer is visiting the page. The learning algorithm adapts the parameters of the surfer in order to increase the probability that the surfer is visiting a "good" page, while decreasing the probability that the surfer is visiting a "bad" page. The parameters are estimated from the Context Graphs, which compactly represent the topological context in which a Web page is inserted. The learning algorithm is fast since only close ancestors of example pages are considered in the Context Graphs. Thus, it can be applied to large scale repositories with little overhead (i.e. the ranking function must be computed more many times). Further investigations are needed to compare the accuracy of the learnt ranking functions with respect to the focused versions of the PageRank and to exploit the complete learning algorithm which considers all the model parameters.

References

[Dempster *et al,* 1977] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B,* 39:185-197, 1977.

[Diligenti et al., 2000] M. Diligenti, F. Coetzee, S. Lawrence, L. Giles, and M. Gori. Focus crawling by context graphs. In *Proceedings of the International Conference on Very Large Databases, 11-15 September 2000, II Cairo. Egypt,* pages 527-534, 2000.

[Diligenti *etal,* 2002] M. Diligenti, M. Gori, and M. Maggini. Web Page Scoring Systems for vertical and horizontal search engines. In *Proceedings of the 11-th World Wide Web conference (WWW11),* 2002.

[Fukunaga, 1990] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, San Diego, CA, US, 1990.

[Haveliwala, 2002] T. H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the eleventh international conference on World Wide Web,* pages 517-526. ACM Press, 2002.

[Kobayashi and Takeda, 2000] Mei Kobayashi and Koichi Takeda. Information retrieval on the web. *ACM Computing Surveys,* 32(2):144-173, 2000.

[Lempel and Moran, 2000] R. Lempel and S. Moran. The stochatic approach for link-structure analysis (SALSA) and the TKC effect. In *Proceedings of the 9th World Wide Web Conference,* 2000.

[Page *etal,* 1998] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the Web. Technical report, Computer Science Department, Stanford University, 1998.

[Richardson and Domingos, 2002] Mathew Richardson and Pedro Domingos. The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. In *Advances in Neural Information Processing Systems 14.* MIT Press, 2002.