2008

Ranking web pages using machine learning approaches

Sweah Liang Yong
University of Wollongong

Markus Hagenbuchner
University of Wollongong, markus@uow.edu.au

Ah Chung Tsoi
Hong Kong Baptist University, act@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/infopapers>

Part of the [Physical Sciences and Mathematics Commons](#)

Recommended Citation

Yong, Sweah Liang; Hagenbuchner, Markus; and Tsoi, Ah Chung: Ranking web pages using machine learning approaches 2008.
<https://ro.uow.edu.au/infopapers/3164>

Ranking web pages using machine learning approaches

Abstract

One of the key components which ensures the acceptance of web search service is the web page ranker - a component which is said to have been the main contributing factor to the early successes of Google. It is well established that a machine learning method such as the Graph Neural Network (GNN) is able to learn and estimate Google's page ranking algorithm. This paper shows that the GNN can successfully learn many other Web page ranking methods e.g. TrustRank, HITS and OPIC. Experimental results show that GNN may be suitable to learn any arbitrary web page ranking scheme, and hence, may be more flexible than any other existing web page ranking scheme. The significance of this observation lies in the fact that it is possible to learn ranking schemes for which no algorithmic solution exists or is known.

Disciplines

Physical Sciences and Mathematics

Publication Details

Tsoi, A., Yong, S. & Hagenbuchner, M. 2008, "Ranking web pages using machine learning approaches", IEEE/WIC/ACM international Conference on Web Intelligence and Intelligent Agent Technology, IEEE, Sydney, pp. 677-680.

Ranking Web Pages using Machine Learning Approaches

Sweah Liang Yong
University of Wollongong
sly56@uow.edu.au

Markus Hagenbuchner
University of Wollongong
markus@uow.edu.au

Ah Chung Tsoi
Hong Kong Baptist University
act@hkbu.edu.hk

Abstract

One of the key components which ensures the acceptance of web search service is the web page ranker - a component which is said to have been the main contributing factor to the early successes of Google. It is well established that a machine learning method such as the Graph Neural Network (GNN)[10, 11] is able to learn and estimate Google's page ranking algorithm. This paper shows that the GNN can successfully learn many other web page ranking methods e.g. TrustRank, HITS and OPIC. Experimental results show that GNN may be suitable to learn any arbitrary web page ranking scheme, and hence, may be more flexible than any other existing web page ranking scheme. The significance of this observation lies in the fact that it is possible to learn ranking schemes for which no algorithmic solution exists or is known.

1. Introduction

The World Wide Web (WWW) can be seen as a repository of huge collections of information. Latest estimates [8] the size of the WWW to exceed 40 billion pages. This presents a unique challenge to information retrieval problems. Information retrieval deals with the relevancy of a certain piece of information based on certain retrieval criterion. Web search services provide a means of identifying documents which contain information on a subject area that is relevant to a given query. Due to the size of the WWW, it is very common that a large number of documents can be identified to match a given query string. To help guide users towards finding the best matching documents, a ranking mechanism is employed. Common methods for ranking are either based on *relevancy* such that documents are ordered from most relevant to least relevant, or on *popularity* where documents are ordered from most popular to least popular. It is important to understand that the term *popularity* is normally the result of link analysis and not the result of user feedback. It is found that most of the research performed on page ranking methods for web pages are based on numerical methods such as in [1, 7, 9].

The general purpose of a ranking algorithm is to impose an *order* on relevant Web pages. For example, a Web search

engine responds to a search query with a set of relevant Web pages which are listed in the order of *rank*. Ranking algorithms can be based on various criteria such as link structure only, document property based only, and a combination of the former. For example, PageRank is a structure only ranking scheme, whereas TrustRank uses a combination of structure and a trust value as a property of Web pages.

This paper shows that a suitably trained machine learning method called Graph Neural Network (GNN) can produce a generic model to encompass different types of the numerical page ranking methods. More precisely, the *GNN* can be viewed as a more generic ranking mechanism to the existing numerical ranking method for Web pages.

The structure of this paper is as follows: In Section 2, we will give a brief introduction to *GNN*. Section 3 presents experimental results when training *GNN* on various ranking algorithms, and conclusions are drawn in Section 4.

2 Graph Neural Network and Related Work

In general, neural networks are trained on sample representative patterns and, once trained, can generalize over unseen data. The *GNN* is a relatively new class of neural network based algorithms capable of processing general types of inputs in terms of graphs [3] in a supervised manner. A *GNN* computes the outputs of a graph based on information pertaining to the a node in the graph, its neighborhood nodes, and links. Each node processes information local to the node ¹, and the inputs to the current node. Each node uses a multilayer perceptron (MLP) architecture to process this information as inputs, except that the MLP architecture does not have any output layer. In other words, each node uses the hidden layer neurons as the outputs to other internal nodes. It is only at the output nodes of the graph that there will be the output layer of the MLP. To reduce the number of variables, the same number of hidden layer neurons is used in each MLP pertaining to each node.

The unknown weights of the overall architecture can be trained using an extension of the standard recursive neural network training algorithm [2]. Once trained the *GNN* may be used to compute unknown outputs for any given input

¹In *GNN*, both links and nodes may contain useful information.

from the problem domain. The GNN is suitable to be applied to the web page ranking problem since it is able to process labelled directed graphs, and hence, is able to process information which can be relevant to the ranking of the web documents. In other words, a GNN can process graphs in which nodes represent web pages, and labels attached to a node which may represent the content of the associated web page. It is noted that the GNN is unique in its capabilities. Previous machine learning approaches to the processing of graphs were limited to acyclic graphs, and hence, would be unsuitable to be applied to the WWW domain.

Supervised learning means that a desired network response (the desired output) to a network input is available during the learning process. In other words, the learning process aims to find parameters which lead from a given input to a desired output. A trained network is said to model the underlying task. For instance, when attempting to train a GNN on data for which the desired output is the PageRank value of a web page, then we aim at creating a model of the PageRank algorithm. More specifically, for web page ranking applications, we assign the known rank value to a selected number of web pages as the supervised set. Then, the GNN is trained using the information on these web pages together with the assigned target values. Once trained the GNN may be used to predict the rank value of other web pages for which the ranking has not been determined. A formal description of the GNN training algorithm is presented in [3] and will not be addressed here.

It was found that the application of the GNN to the task of ranking web pages is not a straight-forward task. We found that the GNNs ability to learn a given ranking scheme can greatly depend on domain features being suitably presented. For example, PageRank and OPIC are ranking algorithms which are exclusively based on link analysis, and hence, it could be assumed that a GNN trained successfully on PageRank would also be able to be successful learning OPIC by using the same input dataset. However, it is found that this is not the case. The training dataset need to be designed carefully for any given ranking scheme. This requires an appropriate approach to the:

(a) Labeling of the training dataset: For example, it will be shown that for PageRank it is sufficient to just provide the link matrix of the nodes (without labels) while HITS it is important to provide labels to the edges to represent directional graphs for satisfactory results.

(b) Selection of the training dataset: For example, it is shown that when addressing TrustRank a randomly selected training set results in a GNN only to be able to achieve 55% accuracy while with a balance selection of training set, GNN is able to achieve 86% accuracy.

GNN has been demonstrated to be able to learn and predict PageRank as well as adaptive PageRank effectively[11]. There are also various work performed on unsupervised machine learning methods [5, 6] using Self-

Table 1. Performance of a trained GNN when applied to the test set of 6,835,644 pages.

Ranking Scheme	Performance
PageRank	99.27%
Adaptive PageRank	95.05%
TrustRank (random seeds)	55.63%
TrustRank (balanced seeds)	86.57%
HITS	42.86%
OPIC	88.36%

organizing Map for Structured Data (SOM-SD) which can be used for clustering web pages. To the best of our knowledge, there is no other published work on machine learning for general web page ranking issues.

3 Experimental Results

In this section, experimental results of GNN learning various popular web page ranking algorithms are presented. Results for PageRank and Adaptive PageRank learning using GNN were reported in [11]; it is found to be 99% and 95% accurate respectively. The performance was measured by counting the number of on-target values, where the GNN output is allowed to deviate up to $\pm 5\%$ from the actual PageRank value to be counted as being on-target. We repeated the experiments by using a much newer and larger dataset consisting of 6,835,644 Web pages which is used as the test set, and forms the basis for the experiments given in this paper. The results are summarized in Table 1.

It is shown that the use of the new dataset did not influence the performances obtained when training the GNN to learn PageRank and Adaptive PageRank. The table presents also the results when applying GNN to learn TrustRank, HITS, and OPIC. The results show that the GNN is capable of learning most of the popular ranking schemes. We will find later in this paper that HITS represents a non-linear ranking scheme, and hence, poses a particular challenge for machine learning methods. The experiments are explained in more detail in the following.

3.1 Learning PageRank and Adaptive PageRank using GNN

The results of GNN learning on PageRank and Adaptive PageRank was published in [11]. The experiments in [11] were based on training set and validation set of 4000 pages each, and test set of 1,692,096 pages. In this paper, the training set and validation set were 6000 randomly selected pages each whereas the test set is over 6 million pages in size. Training PageRank is straight forward by setting the training target values equal to the desired PageRank value for the pages in the training (and validation) data sets.

PageRank: The GNN is given the PageRank value for each web page in the training dataset, and hence, the target is

($t = PR$). The results were between 98 - 99.27% on target where on target is defined as $\pm 5\%$ different of *GNN* output versus actually PageRank value expected. It is understood that in web page ranking problems, the overall measurement of performance is on how closely *GNN* is able to predict the overall positioning of a specific web page. However, a measurement on the deviation from target value is also a valid performance measurement.

Adaptive PageRank: Personalization of PageRank [12] is simulated by setting the target to double the PageRank values for pages addressing either of two given topics ($t = 2 \times PR$) while the PageRank for pages addressing both topics or neither of the topics remains unchanged. Once trained, the *GNN* achieved up to 95% on target. This simulates an XOR situation.

Adaptive PageRank using Constraint: The *GNN* is not given the target value but learns using a constraint matrix to define if a particular topic is more important than other topics. *GNN* also demonstrates excellent results in this experiment. In these experiments $> 96\%$ of the pages addressing important topic are ranked higher and/or the less important pages are ranked lower².

3.2 GNN and TrustRank

TrustRank[4] is a web page ranking algorithm which targets to reduce the effect of spamming web pages semi-automatically. For these experiments the same test dataset as in Section 3.1 is used. Subgraphs were selected to form the training set. A subgraph is formed by selecting seed pages, then connect it to the neighborhood at $d = 1$, where d is the distance to the seed node, i.e. it is one link connection away from the seed page. Two different methods of selecting the seed pages are used:

- 1.) Seed pages are selected randomly
- 2.) Balance selection where seed pages are selected through an even sampling of **trusted pages** (as defined in TrustRank) so that in the training dataset there will be an even distribution of web pages of all categories.

Both types of experiments are run with training and validation set size of up to 60,000 subgraphs. When using randomly selected seed pages, the maximum performance obtained by a trained *GNN* is 55%. Experiment showed that the performance does not increase for larger training datasets. Investigations showed that the *GNN* is not able to generalize satisfactorily when the training set is unbalanced. The experiments with randomly selected seed pages indicates a strong unbalance bias in the training dataset. Thus, in the experiments with balanced selection of seed pages, the selection of training set ensures that balanced samples of all types of pages are selected instead of just focusing on

²No target values were used in training, so performance measurement can only be done on comparing the rank of pages addressing important versus pages addressing less important topic

Figure 1. Ranking positional graph for root set R_σ *GNN* Learning on HITS

the seed pages. By utilizing a balanced selection procedure for the seed pages, the *GNN* performance increases to 86% accuracy.

The reason that *GNN* learning on TrustRank is not able to achieve as good performance as *GNN* learning on PageRank is that TrustRank simulates a biased PageRank, with a bias vector θ , $\theta = 0$ for the spam pages. The existence of this bias obviously affected the learning of *GNN*. However, the exact reasons of why this should be so is not yet known.

3.3 GNN and HITS

HITS[7] is a page ranking algorithm which is based on *authority* and *hubs*. The intuition is that a good *authority* points to a good *hub* and vice versa. It is known that HITS is a fragile ranking method in that the rank value of Web pages in the Internet can change dramatically when the topology of the web graph is disturbed. Hence, it is expected that HITS is a particularly challenging task for the *GNN* to learn. In HITS, a root set R_σ is constructed using some information retrieval method; in our case, they are retrieved via a score which is generated by the naïve Bayes classifier. R_σ in the experiments is 120 pages in size. R_σ is then connected with neighborhood pages of $d = 2$ to form a base set S_σ of 6071 pages.

The HITS algorithm is executed on S_σ to obtain the *authority* values of each page. For the experiments, a training set of 1214 seed pages, validation set of 1214 seed pages, and test set of 3643 seed pages together with all the seeds' neighboring pages are used respectively.

Figure 1 shows the ranking positional graph results of S_σ of *GNN* output. The graph is plotted based on the position of pages based on *GNN* ranking (y-axis) versus the position of pages based on HITS ranking (x-axis). The nearer the dots are to the diagonal, the better performance of *GNN*.

In *GNN* learning HITS, the performance measurements used for PageRank are not appropriate. This is due to the fact that the target values of HITS are not stable. It is possible that the *authority* and *hub* values of HITS may reach

Figure 2. Min-Max-Avg of *GNN* Output Error

∞ or $-\infty$. For the same measurement, the on target pages for *GNN* learning HITS is only at 42.86%. However, when evaluating the performance of *GNN* with ranking positional graphs, it can be observed that the pages ranking positions are similar for *GNN* output as well as HITS output.

3.4 *GNN* and OPIC

OPIC[1] is an online ranking algorithm which ranks Web pages as they are crawled. For the experiments, a training set and validation set of 30,000 subgraphs are used. It is noted that unlike other discussed ranking algorithm so far, OPIC does not produce a stable rank value but the value changes as the algorithm progresses. The measure for OPIC rank is taken at the full 10,000 cycle over the whole graph.

Due to the memory limitation of size and learning cycle required for OPIC learning, the experiments are carried out on “islands of subgraph” where a core set of nodes are selected then connected into a size of about 1 million pages. Some nodes exist in multiple islands due to the connectivity³. Figure 2 shows the Min-Max-Avg (in a total of 10 experiments) *GNN* output error with relates to distance from the core set. Due to this, in the case that a node exists in more than 1 island, the *GNN* output of the result with the nearest to the core is used. The performance of *GNN* learning OPIC is measured as shown in Section 3.1 and *GNN* has achieved 88.36% accuracy.

4 Conclusions

From the various experiments conducted using *GNN* to learn different web page ranking mechanisms, it can be concluded that a suitably trained *GNN* can serve as a generic web page ranking framework. Once trained, the *GNN* exhibits a linear computational efficiency, and hence, is suitable to rank large sets of unseen web pages not used in the training and the validation datasets. The significance of the observations made in this paper lies in the fact that *GNN* can be the tool for achieving ranking methods which are hard to realize algorithmically. For example, it is pos-

³In the test dataset, 58% of nodes exist in more than 1 island

sible to conduct a user survey so as to identify user preferences in the ranking of documents. It is a hard task to translate such responses, which involve a cognitive process, into an algorithmic description. The finding of this paper suggests that it is possible to train a *GNN* on arbitrary ranking schemes for the web. Work is currently conducted towards establishing web page ranking using user perceptions on the quality of ranking by using the *GNN* approach. The reader is invited to participate on the survey at <http://vault.uow.edu.au/searchcmp/>

Acknowledgment: The authors wish to acknowledge financial support from the Australian Research Council in the form of Discovery Project grant.

References

- [1] S. Abiteboul, M. Preda, and G. Cobena. Adaptive on-line page importance computation. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 280–290, New York, NY, USA, 2003. ACM Press.
- [2] M. Bianchini, M. Maggini, L. Sarti, and F. Scarselli. Recursive neural networks for processing graphs with labelled edges: theory and applications. *Neural Networks*, 18(8):1040–1050, 2005.
- [3] M. Gori, M. Maggini, and E. Martinelli. Nautilus: Navigate AUTonomously and Target Interesting Links for USers. Technical Report DII-20/99, Dipartimento di Ingegneria dell’Informazione, Università di Siena, Siena, Italy, 1999.
- [4] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating Web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, pages 576–587. Morgan Kaufmann, August 2004.
- [5] M. Hagenbuchner, A. Sperduti, A. Tsoi, F. Trentini, F. Scarselli, and M. Gori. Clustering xml documents using self-organizing maps for structures. In *Lecture Notes in Computer Science*. Springer, 2006.
- [6] M. KC, M. Hagenbuchner, A. Tsoi, F. Scarselli, M. Gori, and S. Sperduti. Xml document mining using contextual self-organizing maps for structures. In *INitiative for the Evaluation of XML Retrieval (INEX)*, December 2006.
- [7] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
- [8] H.-T. Lee, D. Leonard, X. Wang, and D. Loguinov. Irlbot: Scaling to 6 billion pages and beyond. In *WWW '08: Proceedings of the 17th International Conference on World Wide Web*, pages 427–436, Beijing, china, April 2008.
- [9] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.
- [10] F. Scarselli, S. Yong, M. Gori, M. Hagenbuchner, A. Tsoi, and M. Maggini. Graph neural networks for ranking web pages. In *Web Intelligence Conference*, 2005.
- [11] F. Scarselli, S. Yong, M. Hagenbuchner, and A. Tsoi. Adaptive page ranking with neural networks. In *14th International World Wide Web conference*, pages 936–937, Chiba city, Japan, May 2005.
- [12] A. Tsoi, M. Hagenbuchner, and F. Scarselli. Computing customized page ranks. *ACM Transactions on Internet Technology*, volume 6(number 4):381 – 414, November 2006.