Increasing PageRank through Reinforcement Learning

Adrian K. Agogino
The University of Texas
ECE Department
Austin, Texas 78712
agogino@ece.utexas.edu

Joydeep Ghosh
The University of Texas
ECE Department
Austin, Texas 78712
ghosh@ece.utexas.edu

Abstract

This paper describes a reinforcement learning method, derived from collective intelligence principles, for increasing the combined PageRank for a set of domains. This increased rank is achieved through a set of cooperating reinforcement learners that learn, through exploration, how to add links within the set of domains. We show how reinforcement learners using traditional reward functions perform very poorly at this task. However, reinforcement learners that use rewards based on collective intelligence can achieve good results. The reinforcement learners have an advantage over standard optimization methods in that they can learn with highly non-linear constraints. Additionally we demonstrate that reinforcement learners are robust in that high values of PageRank can be reached, even if all the domains are not cooperating properly.

1 Introduction

The ranking of a page can be very important for the owner of that page. Since many pages are found through search engines, often only the highest ranked pages are read since they tend to be at the top of the list after a search engine query. The popular Google search engine uses an algorithm to rank pages called "PageRank," which assigns a rank to each page, based on the topology of the web, i.e. its link structure [1]. Perterbation analysis shows that PageRank is robust against changes made to the directed web graph [2]. The robustness come from the fact that the PageRank algorithm uses information about the inlinks to a page to determine its rank. Since a page's owner cannot normally change the inlinks to a page, he cannot manipulate the page to increase its rank. However, if a group of domains work together, the average PageRank for pages within the group can be increased significantly by changing the link structure within the group. Since PageRank only counts links between different network domains, this paper will discuss links between domains instead of individual pages.

2 Reinforcement Learning Approach

We implement a collective learning approach where each domain has its own reinforcement learning algorithm that learns which links to add to increase page rank, under a set of constraints. The reinforcement learners "learn," by first doing exploration, where they try a series of actions, consisting of adding links to various domains within the group. After each action, the reinforcement learner receives a reward based on how good the action was. The learner can then exploit the information it has gathered, and choose the actions that will result in the highest expected value of the objective function.

Reinforcement learning has some significant advantages over traditional methods. First of all it can learn arbitrary objective functions. This property becomes very useful when the page owners want to put some soft constraints in the form of nonlinear penalties added to the ranking objective. Another advantage of using reinforcement learners is that they provide more robustness in this naturally distributed system. Since each domain has its own learner, it can adapt to problems caused by other domains. This ability to adapt is important since numerous problems could arise when a diverse set of domain owners is trying to work together. These problems could range from causes such as an owner neglecting to implement the algorithm correctly, or difficulties outside the owners' control such as a failure in a hosting service.

2.1 Choosing Rewards in a Collective

The goal of the collection of learners is to optimize a global objective function based on the actions of all the domains. Let us formally define this global objective function as $G(\zeta)$, where ζ is a vector containing all the information about the problem. In our case, ζ contains the link structure between the domains. We will refer to $G(\zeta)$ as the *global utility*. An example of a global utility would be the average PageRank for a set of domains, $G(\zeta) = \sum_{i \in S} r_i$. In trying to maximize $G(\zeta)$, each reinforcement learner η has it own reward function, $g_\eta(\zeta)$. We will call this function the *personal utility* [3]. A successful collaborative learning system must choose a set of $g_\eta(\zeta)$, such that high values of $G(\zeta)$ are achieved. There are two main difficulties in doing this. The first one is that we need the personal utility to be "aligned" with the global utility. If the personal utilities are chosen poorly, they may cause the learners to work at cross-purposes, where they hurt the global utility while trying to maximize their own utilities. The second problem is that the learner may fail to learn anything at all since ζ contains the actions of the other learners. If chosen poorly, a learner's influence over $g_\eta(\zeta)$ may be dwarfed by the actions of all the other learners.

2.1.1 Factoredness

In this paper we will define a *factored* utility as a personal utility that is aligned with the global utility [1]. Formally a personal utility, $g_\eta(\zeta)$, is factored with respect to $G(\zeta)$, if for any change in ζ resulting in ζ'

$$g(\zeta) \geq g(\zeta') \Leftrightarrow G(\zeta) \geq G(\zeta'). \tag{1}$$

One trivial way of making $g_\eta(\zeta)$ factored is simply to make $g_\eta(\zeta) = G(\zeta)$ for all η. This approach is known as a "Team Game."

2.1.2 Learnability

We will informally define the learnability of a personal utility, $g_\eta(\zeta)$, as the change in $g_\eta(\zeta)$ caused by an action taken by η divided by the change in $g_\eta(\zeta)$ caused by the actions of all the other learners. Note that the utility used in the Team Game approach, $G(\zeta)$, is not very learnable when the number of learners is large, since generally all the learners' actions will have an equal influence on $G(\zeta)$. A more learnable utility for learner η would be one that uses information more local to η, such as the PageRank of the domain associated with η. The difficulty though is that this local utility may not be factored. However a good compromise can be made between factoredness and learnability. This can be done by using a form of the difference utility:

$$DU_\eta(\zeta) = G(\zeta) - G(\zeta_\eta) \tag{2}$$

where ζ_η is the same as ζ except that all the links that have been added by reinforcement learner η are removed. The second term of this utility computes what the global utility would be like without learner η. The entire utility essentially returns the individual learner's contribution to the global utility. This utility tends to be much more learnable than $G(\zeta)$, since the noise caused by all the other learners is subtracted out in the second term of the equation. It can also be proven that this utility is still factored [3, 4].

3 Experiments

We conducted three experiments to determine how effective reinforcement learning is in increasing PageRank for a group of domains. The experiments were conducted using artificial data with 50 domains where the number of outlinks of a domain was uniformly distributed in the range 0 to 25. Artificial data was used since while there are many data sets containing link structure within a few domains, there are not many good publicly available ones that contained a large number of domains. Since PageRank is only calculated between domains, we needed a large number of domains. Also the linkage density is much higher between large domains than between individual pages, so using data from pages is not a viable substitute for domains.

[1] In general factoredness can be defined on any utility with respect to any other utility

In all three experiments, 10 random domains were chosen to comprise the set of cooperating domains. Then all links from a domain within this set to a domain outside of the set were removed. This was done to simulate the practice of many companies of avoiding linking to any page that is not within their own domain or a domain of a closely associated parter. Each of the 10 domains was given a simple single-stage reinforcement learner. The actions of the reinforcement learners was to choose how many links to add to domains within the set. These links were then made randomly since experiments showed that the exact target of the links did not influence PageRank significantly. In each experiment three types of utilities, $g_\eta(\zeta)$, where used. The first utility was the Team Game (TG) utility where $g_\eta(\zeta) = G(\zeta)$. The second utility we called the Selfish Utility (SU), where a learner's reward was equal to its domain's PageRank. The final utility was the difference utility (DU) given by equation 2.

3.1 Experiment 1

In this experiment we let the global utility, $G(\zeta)$, simply be the average page rank, without any penalty on the number of links added. In this case, the optimal solutions was to trivially add as many links as possible. Despite having a staightforward solution, Figure 3.1 shows that learners using TG utilities and SU utilities could not achieve high values of the objective function. The TG utility failed because it was difficult for an individual learner to observe the influence of its action, among all the actions of the other learners. Even if a learner took an action that was good, some of the other learners may have taken bad actions at the same time, resulting in a local utility that had too small a value. The SU utility also failed, because a domain cannot change its own rank significantly by altering its links. However, the reinforcement learners using the difference utility were able to learn to achieve high values for $G(\zeta)$ very quickly.

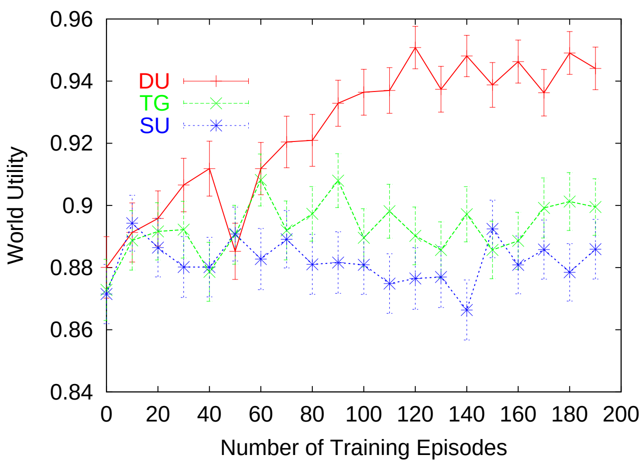

Figure 1: Performance with no link penalty

3.2 Experiment 2

To test reinforcement learning in a more interesting problem, we modified the objective to penalize the number of links added:

$$G(\zeta) = (\sum_{i \in S} r_i)(1 - a \sum_{i \in S} e^{l_i}) \tag{3}$$

where l_i is the number of links added from domain i in set S and a is a scaling factor. Note also that this new objective was nonlinear and difficult to solve analytically. The results in Figure 3.2 show that reinforcement learners using the difference utility were also able to learn the new objective function quickly.

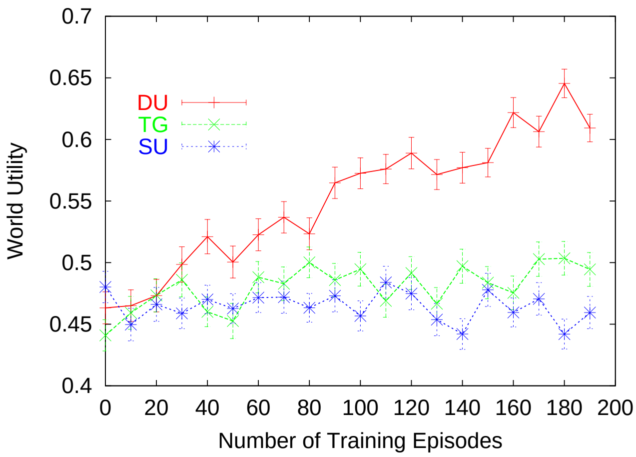

Figure 2: Performance with link penalty e^l

The solution formed by these learners was non-trivial with different domains choosing to add different numbers of links. Figure 3.2 shows outcome of one of the trials. The number of links chosen depends the PageRank of the domain as well as the initial link structure. For example, since the PageRank conferred to a target is divided by the number outlinks from the origin, we would expect that a domain with few outlinks would be more likely to add new links, since each new link would be more valuable to the target's PageRank. This is confirmed by the domain on the upper right of the figure, which started out with no outlinks, but added a large number of new links.

3.3 Experiment 3

As a final experiment, we tested how the reinforcement learners were able to learn when not all the domains cooperated well. In this experiment we used the same non-linear objective function, but this time 20% of the time the learners would not take any action. This would simulate a situation where a domain

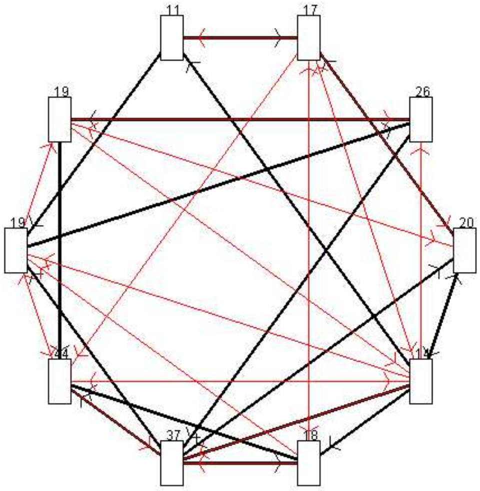

Figure 3: Final link structure when using DU. Thick lines are the original links

owner withdraws from the system unannounced, or neglects to modify his domain. Figure 3.3 shows that again the learners using DU are able to achieve high values of the objective function, even when not every domain is cooperating.

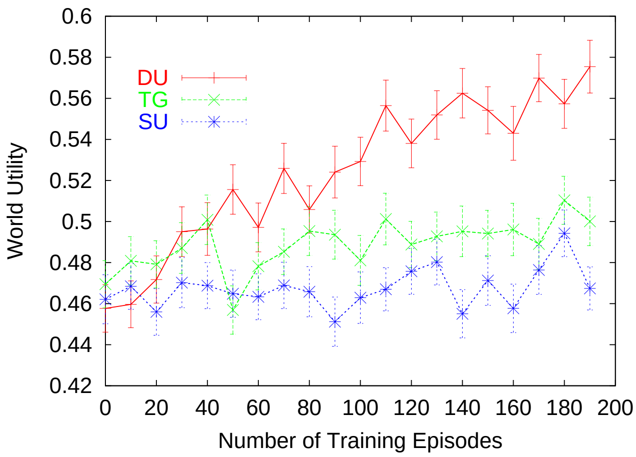

Figure 4: Performance when not all domains cooperate

4 Conclusion

Google proudly proclaims that an individual page cannot significantly change its own PageRank value by adding selected keywords or outlinks. Previous perturbation analysis has supported this claim which helps Google to be robust against manipulation by individual commercial interests. However, it is possible that a set of domains "working together" can significantly improve their average PageRank, but this is a highly complicated problem involving distributed learning. This paper presents a method, based on collective reinforcement learning, where a collection of domains can increase their average PageRank, under a wide set of constraints even while being subject to unreliable cooperation.

References

[1] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," 1998.

[2] A. NG, A. Zheng, and M. Jordan, "Stable algorithms for link analysis," in *Proceedings of SIGIR'01*, 2001.

[3] D. H. Wolpert and K. Tumer, "An Introduction to Collective Intelligence," Tech. Rep. NASA-ARC-IC-99-63, NASA Ames Research Center, 1999. URL:`http://ic.arc.nasa.gov/ic/projects/coin_pubs.html`. To appear in Handbook of Agent Technology, Ed. J. M. Bradshaw, AAAI/MIT Press.

[4] D. H. Wolpert and K. Tumer, "Optimal payoff functions for members of collectives," *Advances in Complex Systems*, vol. 4, no. 2/3, pp. 265–279, 2001.