Properties of Hostgraph Enriched with IP Information

© Y. Pritykin, V. Koshelev

Yandex
{pritykin, vakoshelev}@yandex-team.ru

Abstract

Networks have been extensively studied from different perspectives in a lot of contexts. However, most of research uses a simple representation of networks with all edges having equal weights. We propose to consider networks arising in Internet research with weights on the edges. We consider a very simple example where weights are assigned to edges of the hostgraph based on similarity of IP addresses of the hosts. We show that some simple host-wise features of this weighted hostgraph may contain interesting signal, in particular, for ranking.

1. Introduction

For a lot of large systems and structures of different nature, network can be a good model capable of representing a lot of key properties. Examples are well known in biology (networks of molecular interactions, neuron or cell communication), transportation (networks of roads, flights, public transportation roots), communication architecture (telephone or Internet infrastructure), social interactions (social networks). Networks have been extensively studied from many different perspectives [1]. In most cases networks are represented as simply a collection of vertices and edges between them, directed or undirected. However, often weights can be assigned to edges, so that the resulting weighted network contains more useful information and thus is more representative. Even simple weight distributions can be quite informative [2].

In studying Internet, networks have proved especially prominent. At a high level of abstraction, the World Wide Web can be represented as a network of all pages and hyperlinks between them represented as edges. At even a more abstract level, hostgraph is a directed graph of hosts (owners) with edges in the graph being hyperlinks between pages of the hosts. Probably, the most well known example of usage of such a graph is the PageRank algorithm for the web graph [3] that uses a very basic model of user behavior, where a user starts from a random page on the web and then randomly surfs via hyperlinks. Historically, this algorithm was a breakthrough in ranking, and later a lot of variations were proposed, including applications of this algorithm to other graphs, e.g., hostgraph, and varying different parameters, e.g., playing with different user's initial distributions on pages of the web [4]. However, it seems that in all current versions all edges (i.e., hyperlinks) are usually assumed to have equal weight. Later, as more data became available and sophisticated learning-to-rank methods were extensively introduced, PageRank and its many variations lost their leading positions, as well as many other methods that rely only on the information from the webgraph.

At the same time, the underlying structure of the Internet at a lower level of abstraction can be represented as a collection of IP addresses. Understanding the structure of the Internet in this sense is not an easy task [5]. An interesting approach to email spam detection was recently proposed [6] that is based on learning on lightweight features from large collections of data. Some of these features are based just on IP addresses of senders and do not use any email content. This method proved to achieve comparable accuracy to existing blacklisting methods.

We argue that these two kinds of Internet data can be integrated. We propose to look at the hostgraph where edges are weighted with respect to similarity of IP addresses of the source and the destination of the edge. The idea is to capture different properties of different IP density areas in the hostgraph. We show that it can be advantageous to consider certain measures on this graph, even very simple ones, with respect to such weights. We show that corresponding host-wise features may be useful for ranking [7].

2. Data

The main our dataset is a hostgraph constructed by one of the versions of robot developed in Yandex [8]. Vertices of this graph are hosts, i.e., collections of documents aggregated from one owner. We say there is an edge between two hosts if a document from one host has a hyperlink to a document from another host. For our experiments, we consider only partial data from the older version of the robot, so our graph is far from complete representation of the whole Internet, and mostly contains hosts from Russian segment of the web. Moreover, we clean it further, filtering out large blog hosts. Also, we leave only the nodes with at least one incoming degree. The final hostgraph for our experiments contains about 19M hosts and has total of

Proceedings of the 13th All-Russian scientific conference "Digital Libraries: Advanced Methods and Technologies, Digital Collections" - RCDL'2011, Voronezh, Russia, 2011

276M edges (14.7 outcoming edges, i.e., hyperlinks, per host on average).

IP addresses are collected by Yandex robot, with one IP address per host. Of course, in reality the correspondence between hosts and IP addresses may not be one-to-one for a variety of reasons. However, for majority of hosts this approximation seems to be appropriate, and is suitable for our needs.

3. Results

3.1 Computations

First, we define a measure of similarity between IP addresses. We want it to be a number between 0 and 1 that is 0 for equal IP addresses and is getting larger as IP addresses are getting farther apart. We use numeric distance, based on the one defined in [6]. If two IP addresses are treated as numbers between 0 and $2^{32} - 1$, let n be the most significant bit where they differ, e.g., 31 if they differ in the least significant bit, as in 192.108.0.0 and 192.108.0.1, and 0 if they differ in the most significant bit, as in 77.88.21.11 and 213.180.204.11. Then let a^{-n} be the distance between these IP addresses, where $a > 1$ is some parameter. In what follows, we choose $a = 1.1$ (the idea is to make the distance not as sensitive as say when using the standard value $a = 2$; although the results should not depend much on this parameter). Now assign weight to each edge in the hostgraph to be the above distance between source and target IP addresses.

Now we consider several different host-wise features based on the above edge weights. The simplest one is the strength of a host `IPStrength`, and is defined as the sum of weights of all incoming edges in the graph. (It is more useful to consider only incoming edges, i.e., hyperlinks *to* the host, as they are less controllable by the host.) We also consider the average of weights of all incoming edges for a host (i.e., `IPStrength` divided by indegree), and denote it by `IPStrAv`. We also denote the host in-degree by `HostInDegree`.

We want to emphasize that the above features are very simply defined and their parameters are chosen somewhat arbitrarily. For checking robustness, we tried some changes in the above scheme (edge distance function, parameters of this function like a above, etc.) and obtained comparable results (not shown here).

3.2 Statistics

As it is well known, many different "real world" networks are scale-free, i.e., their degree distribution follows a power law asymptotically [1]. Fig. 1 shows distributions for `HostInDegree` and `IPStrength` on a log-log plane. Indeed, `HostInDegree` very clearly follows the power law distribution, as `IPStrength` does in the range of relatively large values. However, the distribution of `IPStrength` deviates from power law in the range of small values of `IPStrength` (and thus small `HostInDegree`). This difference indicates that `IPStrength` has potential to contain a useful signal even in presence of `HostInDegree`.

For checking if indeed `IPStrength` contains a useful signal, we use a standard approach in statistics, permutation test, in the following way. We compute the same measure as `IPStrength`, for the real hostgraph used in the initial computation, but with all IP addresses in the hostgraph permuted among hosts uniformly at random. We repeat this experiment 100 times and thus obtain for each host the null distribution of 100 values (for real hostgraph and a new random permutation of IP addresses on the whole set of its vertices in each experiment). Let us denote the average of this distribution for a host as a host feature `IPStrengthRand`. The distribution of `IPStrengthRand` is shown on Fig. 1 (bottom, green).

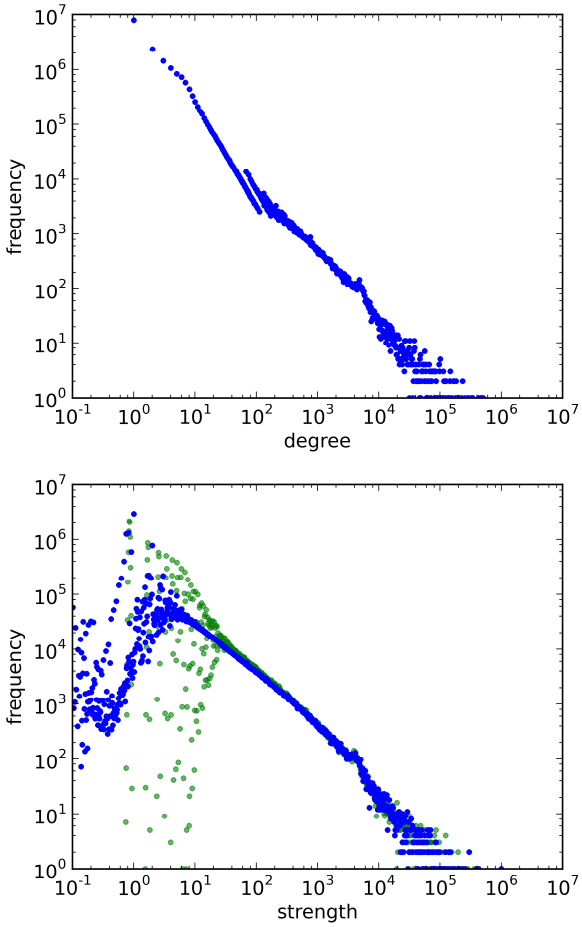

Figure 1: Degree (`HostInDegree`) and weighted degree (`IPStrength`) distributions on log-log scale. Each point shows frequency of the value from x axis in the hostgraph. With strength (blue), also randomized strength (`IPStrengthRand`, green) is shown for comparison (see text for details).

A clear difference between distributions of `IPStrength` and `IPStrengthRand`, again especially in the range of small values, is another indication of potential useful signal in `IPStrength`. Note that actually most of the hosts in the hostgraph are in the

range of small IPStrength values: 98.7% of the hosts have IPStrength less than 100, and 89.3% of the hosts have IPStrength less than 10.

For each host, we compute the *z*-score of its IPStrength with respect to the distribution of its randomized values. If *v* is the value of IPStrength for this host, *m* and *s* are the mean and the standard deviation of all randomized values for this host, respecitvely, then *z*-score is defined as $z = (v - m) / s$. We denote this feature of a host as IPStrZScore. There are 16.7% of the hosts with IPStrZScore < –2, that is, in the hostgraph with actual distribution of IP addresses there are many hosts (much more than expected by chance) with links from the hosts of similar IP addresses, as compared with the random distribution of IP addresses in the hostgraph. The distribution of *z*-scores is shown on Fig. 2.

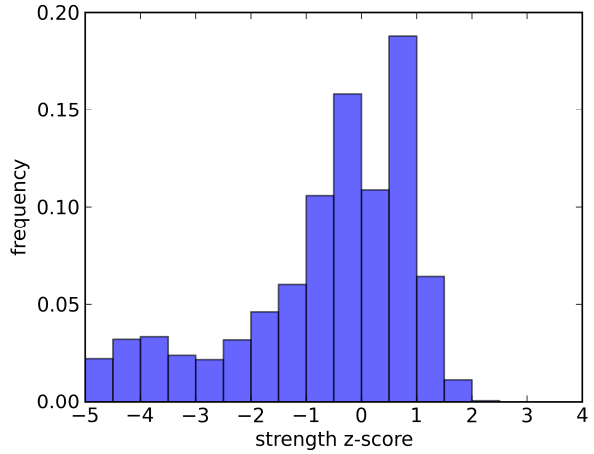

Figure 2: Strength z-score (IPStrZScore) normalized distribution.

For finer comparison with randomized values, for each host we compute empirical percentile of its IPStrength with respect to the null distribution of randomized values for this host. As we only have 100 samples, at the highest precision we can use the 1% and 99% levels. It turns out that for 14.8% of the hosts their IPStrength is significantly *lower* than IPStrengthRand at significance level < 1%, and for 17.2% of the hosts their IPStrength is significantly *higher* than IPStrengthRand at significance level > 99%. Thus for 32.0% of the hosts IPStrength significantly deviates from its randomized counterpart.

Overall, this comparison of IPStrength with corresponding randomized data indicates that the weights defined above are indeed informative for the hostgraph and are far from random.

3.2 Comparison

It is interesting to compare the IP-similarity-based host-wise features that we defined above, with other common static link-based host-wise features. MatrixNet is a new method of machine learning that was developed in Yandex [7]. One possible way for feature comparison is to compare the contribution while learning relevance function for ranking. For a certain learning sample where web pages are scored by assesors as "good" and "not so good" for certain queries [7], we can run MatrixNet on a certain collection of features and compare the success of optimizing the relevance function using all features and all features plus the new feature in question. This gain can serve as a proxy for the contribution of the feature to ranking, and gains of different features can be compared with each other.

It should be noted, however, that simple features such as defined above are unlikely to directly contribute to current state-of-the-art ranking pipeline, as it uses a complicated highly optimized model with hundreds of features, many of them quite sophisticated. Even if the features do contribute positively, it is not an easy task to detect this contribution, especially in the production setting.

This is why we do a simplified experiment, more as proof of concept, to show that the features of the hostgraph defined above are of interest. From the whole collection of ranking features we choose a representative set of 20 static features that depend mostly on the hostgraph or the pagegraph, and in certain cases on some additional information. This set includes HostRank and its variants, quotation index [9], and several models of link relevance developed in Yandex. Some of these features are among the strongest even if compared with the whole collection of all features. We compare with this set each of the features IPStrength, IPStrAv, IPStrZScore, and also HostInDegree. For each of these features we do the following test: for two learning samples - with and without the feature - we run optimization of relevance using MartixNet. We do 200-fold cross-validation where in each fold, randomly chosen 90% of the learning sample are used for learning and the remaining 10% are used for testing. The same subdivision of folds is used in both tests, and thus the two 200-length vectors of gain values obtained in folds of these two tests can be compared with each other. Thus we can compare not only the difference in average gain over folds, but also how significantly consistent among folds this difference is, as by some standard statistical test (we use Wilcoxon rank test). This experiment, as described, still has a lot of parameters that should be indicated and that we omit here as otherwise it would require the detailed description of MatrixNet algorithm and properties of learning samples used, which is not our focus.

After repeating this experiment with different parameters, we conclude that certain our features are at least as good, and some of them even better than those static features used for comparison. For example, in one reasonable setting, addition of IPStrAv to the sample shows the improvement of 0.0062% to the error in optimization, and addition of IPStrZScore shows the improvement of 0.0060%, and both are significantly consistent over folds as by Wilcoxon rank test, $p < 0.0002$. For comparison, in exactly the same setting, addition of HostInDegree shows improvement of –0.0013% (in other words, loss of 0.0013%). In another setting, IPStrength also proves useful: it shows

relative improvement of 0.0021% ($p < 0.037$) even after `HostInDegree` is already added to the sample. This confirms that `IPStrength` has useful signal even in presence of `HostInDegree`.

Overall, observations from these experiments again support the idea of considering weighted networks instead of plain ones where edges may only be present or not present.

Note that here we choose only 20 representative static features to compare with, as without doubt more complicated dynamic features can be much more useful for ranking, and the effect of static features then is tiny. However, we tend to think that even the above simplified experiment is good enough for direct comparison of certain collections of features. At the same time, even static features of the hosts are interesting to study, since they are relatively easy to compute, for hostgraph, even though looking large, is only a tiny portion of all the data that is constantly being collected by large search engines like Yandex. Unlike many other features useful and optimized exclusively for ranking, the ones that we consider here could potentially be used for other tasks such as spam detection and web crawling. Specialized ranking or ranking in absence of some data critical for other more efficient features (for example, when developing search engine for a new market) could be another application.

4. Conclusions

We argue that considering graphs of the web with weighted edges may be useful. As a proof of concept, we proposed to integrate the hostgraph with IP addresses of hosts, namely, to weigh edges in the hostgraph with respect to similarity of IP addresses of their endpoint hosts. We gave some evidence why such consideration may be useful. Potentially, other graphs representing the Internet and the web (webpage graph, social graph, internal page graph of the host) may be interesting to consider with weights assigned to their edges, thus integrating with networks all kinds of Internet data that now are increasingly being collected.

References

[1] D. Easley and J. Kleinberg. Networks, Crowds, and Markets: Reasoning about a Highly Connected World. Cambridge University Press, 2010.

[2] A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. Proceedings of National Academy of Sciences, 101(11):3747–3752, 2004.

[3] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

[4] R. Baeza-Yates, P. Boldi, and C. Castillo. Generalizing PageRank: damping functions for link-based ranking algorithms. In Proc. SIGIR'06, pages 308–315, 2006.

[5] J. Heidemann, Y. Pradkin, R. Govidan, C. Papadopoulos, G. Bartlett, and J. Bannister. Census and survey of the visible internet. In Proc. Internet Measurement Conference 2008, pages 169-182, October 2008.

[6] S. Hao, N. Syed, N. Feamster, A. Gray, and S. Krasser. Detecting spammers with SNARE: Spatio-temporal network-level automatic reputation engine. In Proc. 18th USENIX Security Symposium, pages 101–118, August 2009.

[7] Matrixnet: New level of search quality. http://company.yandex.com/technologies/matrixnet.xml
Матрикснет - новое качество поиска Яндекса. http://company.yandex.ru/technology/matrixnet/

[8] Yandex. http://www.yandex.com
Яндекс. http://www.yandex.ru

[9] Тематический индекс цитирования (thematic quotation index). http://help.yandex.ru/catalogue/?id=873431