Understanding Crowds' Migration on the Web

Yong Wang*, Komal Pal† and Aleksandar Kuzmanovic†
*University of Electronic Science and Technology of China, Chengdu, China
†Northwestern University, Evanston, IL, USA

***Abstract*—Consider a network where nodes are websites and the weight of a link that connects two nodes corresponds to the average number of users that visits both of the two websites over longer timescales. Such *user-driven* Web network is not only invaluable for understanding how crowds' interests collectively spread on the Web, but also useful for applications such as advertising or search. In this paper, we manage to construct such a network by 'putting together' pieces of information publicly available from the popular analytics websites.**

Our contributions are threefold. First, we design a crawler and a normalization methodology that enable us to construct a user-driven Web network based on limited publicly-available information, and validate the high accuracy of our approach. Second, we evaluate the unique properties of our network, and demonstrate that it exhibits small-world, seed-free, and scale-free phenomena. Finally, we build an application, *website selector*, on top of the user-driven network. The core concept utilized in the website selector is that by exploiting the knowledge that a number of websites share a number of common users, an advertiser might prefer displaying his ads only on a subset of these websites to optimize the budget allocation, and in turn increase the visibility of his ads on other websites. Our website selector system is tailored for ad commissioners and it could be easily embedded in their ad selection algorithms.

I. Introduction

The World Wide Web attracts millions of users on a daily basis. This has created an unprecedented opportunity to study the properties of online social networks that are formed around popular websites and services (*e.g.*, [1]). Such information is invaluable for understanding fascinating individual and collective online user properties. Moreover, the knowledge acquired in this way is useful for developing advanced socially-aware Web and Internet services [2], [3].

In this paper, we explore users' association with different websites *at scale* in an attempt to understand how groups of users collectively 'walk the Web'. In particular, we study a network in which nodes are websites, while a weight of a link that connects two nodes represents the average number of users that visits both of the two websites over longer timescales. Such information is invaluable in applications such as search and advertising. In particular, we build an application, *website selector*, on top of such user-driven network. The core concept utilized in the website selector is that by exploiting the knowledge that a number of websites share a number of common users, an advertiser might prefer displaying his ads only on a subset of these websites to optimize the budget allocation, and in turn increase the visibility of his ads on other websites.

We obtain our network by crawling the popular `Google Trends` website [4], which combines information from a variety of sources, such as aggregated Google search data, aggregated opt-in anonymous Google Analytics data, and opt-in consumer panel data. We then study the properties of the user-driven network and build the website selector on top of such a network. Our contributions are the following.

First, we provide a comprehensive methodology for generating a normalized user-driven Web network by strategically extracting and combining pieces of publicly-available information. In particular, our goal is to generate globally meaningful link weights based on relative local weights available from individual Google Trends' snippets. We design a method that combines crawling and normalization procedures to achieve this goal. We validate the high accuracy of our approach.

Second, we evaluate the properties of the network. We show that our network has a small average path length and a strong clustering coefficient, which demonstrate that our network exhibits a small-world phenomenon. Moreover, we prove that the properties are independent from network features in terms of its size, as well as the seed where we start crawling the network. Finally, by comparing our network with the Web and online social networks, we realize that our network significantly differs from the Web, but to some extent, is closer to online social networks.

Third, we build an application, website selector, to optimize advertisers' campaigns. The selection is performed automatically based on advertisers' budget. We extensively evaluate our application and the results demonstrate that website selector can increase the visibility of ads by more than 22% and consequently help increase revenues. Finally, our website selector system is tailored for ad commissioners and it could be easily embedded in their ad selection algorithms.

This paper is structured as follows. We first systemically crawl, normalize and validate the network in Section II. We then evaluate the properties of the network and compare it with Web as well as online social networks in Section III. We next propose website selector in Section IV. We finally discuss related work in Section V and conclude the paper in Section VI.

II. Crawling and Normalization

Here, we explain our methodology for obtaining a user-driven Web network. We first introduce the basic mechanisms provided by Google Trends in Section II-A. Then, we explain the construction of the website-affinity-based networks in Section II-B. Next, we provide a methodology to effectively normalize such networks, so that all websites in the network

This project is supported by the National Science Foundation (NSF) via grant CNS-1064595.

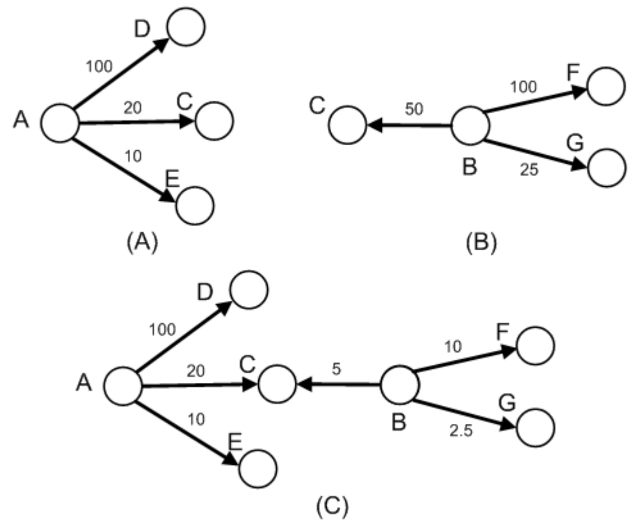

Fig. 1. Background (The numbers are the relative values given by Google trends)

are scaled to the same unified $'global'$ scale. In addition, because the affinity in the normalized network does not reflect the absolute traffic information, we bring such absolute values into the networks in Section II-D, and unify two scale systems in Section II-E. Finally, we validate our approach in Section II-F.

A. Background

For a website (*e.g.*, www.nytimes.com), Google trends [4] provides a list of up to 10 of other websites that its users have also visited. Moreover, Google trends also shows the relative likelihood that users who visited this site also visited other websites. Each website that is searched on Google trends is considered a *parent* node in our network, and the corresponding list returned by Google trends consists of the *children* of the parent website. An edge between a parent node and a child node means that these two websites have common users. The weights of edges reflect the likelihood of a parent node's users to migrate to its children websites, which in turn shows the affinity among them.

The likelihood of users migration between the parent website and its first child (the website which the parent website's users most likely visit) is always scaled to 100% by Google trends. The likelihood of visiting other children (websites that the parent website's users also visit) is scaled to the affinity of the parent and its first child. For example, in Figure 1(A), the users who visit website A also visit other 3 websites (D, C, and E) with a decreasing order based on the number of common users between A and its children. Assume that the number of common users shared by A and D is 100K. (This number is scaled to 100% by Google trends and shown close to the edge AD in Figure 1(A)). The number of common users between A and C is 20K, which will be scaled to 20%. Similarly, the likelihood of edge AE is also scaled to the value of AD.

In addition, Google trends accepts the queries with more than one website (*e.g.*, 'www.nytimes.com, www.cnn.com'). If two websites are searched together orderly, for the second website, Google trends will provide the affinity ratio between the second website and the first website's children, rather than between itself and its children. In addition, such relation is scaled to the value between the first website and its first child as well. As we will explain in Section II-C, such mechanism is very helpful when normalizing the network.

B. Crawling the Network

We start crawling the network in a breadth-first way from three different seeds: the first one is nytimes.com, a very popular newspaper website with 55M unique visits monthly, the second one is timesofindia.com, an Indian newspaper website of 13M unique visits monthly, and the third one is sina.com.cn, a Chinese newspaper website of over 73M unique visits monthly. Starting from three seeds in different regions to construct the networks enables us to examine the differences and similarities of these networks' properties in Section III.

In every step of the crawling, we record the website as a node in the network, and the relation between this website and its children as the edges. The weights of the edges are also recorded in a straightforward way. Since Google trends could not provide children information for some websites due to the insignificant relations between these websites and their children, the crawler is terminated by itself when there are no nodes that have children information to further expand the network. By running for a period of around 15 days, the crawler collects 297,457 nodes and 2,807,496 edges for the US-centric network initiating from nytimes.com, 297,443 nodes and 2,807,396 edges for the India-centric network initiating from timesofinda.com, and 290,532 nodes and 2,700,852 edges for the China-centric network starting from sina.com.cn.

C. Normalizing the Network

Google trends provide invaluable information for us to crawl the network and understand the relation among websites. However, the key issue is that the weights of the edges represent affiliation values relative to each individual query only. Hence, such values are insufficient to analyze the affiliation properties of this network at a global level.

Consider the following example shown in Figure 1(B). The absolute traffic between B and C is 5K, and this value is set as 50% since it is scaled to the relation between B and its first child F whose absolute traffic is 10K. From the relative values' perspective, website B has stronger affinity with C than A does. However, the number of common users between A and C is 20K, twice as large than that between B and C. Apparently, not only do such relative values fail to reflect the real traffic among websites, but also hinder the analyzes of the network's real properties. Thus we introduce a series of approaches to normalize the network in this section, and validate the correctness of the normalization in Section II-F.

As we explained above, a basic mechanism supported by Google trends is that when two websites A and B are searched together orderly, Google trends will provide the relation between B and A's children. In the above example, since A and B

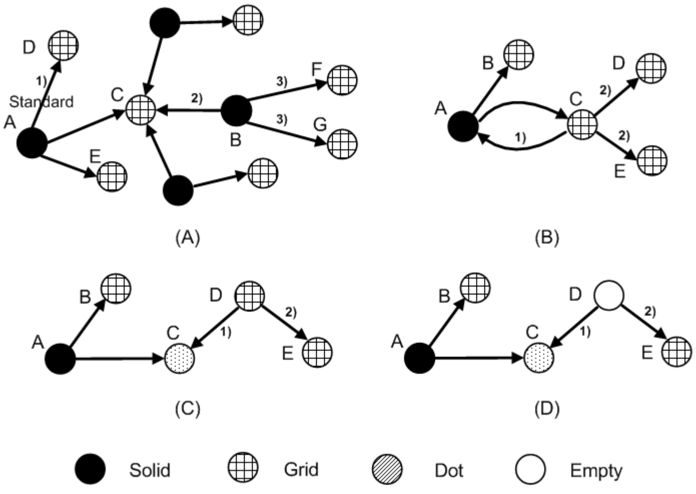

Fig. 2. Normalization (The numbers are the sequences of normalization)

share a same child C, the relation of BC would be scaled to 5% based on the weight of AD, if A and B are searched together (refer to Figure 1 (C)). Although this still does not show the absolute traffic, such mechanism of utilizing child C as a bridge to bring A and B together could exhibit the real traffic relationship between both websites. Later, we will introduce an approach to providing absolute traffic for the networks in Section II-D and unifying two scale systems in Section II-E. Once the weight of BC is normalized, the weights of B and its other children (F and G) can also be normalized based on the ratio of BC to BF and BG. Continuing the example above, the weight of BG was 25%, which means the ratio of BC to BG is 2:1, thus the weight of BG will be normalized to 2.5% shown at Figure 1 (C).

Phase 1: Selecting a starting point. In order to start the normalization with a relatively large subset of the network, which in turn could improve the normalization efficiency, as shown in Figure 2(A), we first choose the node (C) with the maximum in-degree. Note that this node is also the one whose number of parents is the largest. We then randomly choose a parent node of C's (say A), and consider the relation between this parent node (A) and its first child (D) as the standard weight (step 1 in Figure 2(A)), to which weights in the network will be scaled. We next normalize other parent nodes by searching them separately with A in Google trends. Once the relation between the parent nodes (say B) and C is scaled to the weight of AD (step 2 in Figure 2(A)), we normalize B's other edges correspondingly based on the ratio of BC to these edges (step 3 in Figure 2(A)). Once the nodes and their edges are normalized, we label these nodes as solid to indicate that their edges are normalized, and their children as grid to express that their edges are not normalized yet. Interestingly, the nodes with the maximum unnormalized in-degree in three networks that we investigate are `facebook.com`. This means that online users who visit other websites are most likely to visit facebook.com as well. We find this makes sense due to Facebook's increasing popularity [5].

Phase 2: A back link from a grid node to its parent. If there is a back link from a grid node to its ancestor that is a solid node, the weight of the back link must be the same as that of the forward link from the solid node to its child. Thus, in the example of Figure 2 (B), the weight of CA equals to that of AC. This is due to the fact that the users who look at website A as well as website C are the same population. Since node A is a solid node whose edges are normalized, the weight of CA can be normalized to that of AC (step 1 in Figure 2 (B)). C's other edges (CD and CE) can also be normalized based on the ratio of CA to them (step 2 in Figure 2 (B)).

Phase 3: A grid node shares a child node with a solid node. If a grid node has a child that can be either a solid node or a grid node (illustrated as Dot in Figure 2), and this child is also a descendant of a solid node, we can search this grid node and the solid node together to normalize this grid node. For example, in Figure 2 (C), A is already normalized and it shares a same child C, with D. Entering A and D together in Google trends can normalize the edge of DC (step 1 in Figure 2 (C)). Moreover, D's other edges (say DE) can also be normalized subsequently (step 2 in Figure 2 (C)).

Phase 4: An empty node shares a child node with a solid node. An empty node in Figure 2 indicates that this node is neither a solid node nor a grid node. In the network, if such nodes have a child that is also a descendant of other solid nodes, they can also be normalized. In the example of Figure 2 (D), D is an empty node and has a child C. C is at the same time also a child of the solid node A. Consequently, the edge of DC is normalized by that of AC (step 1 in Figure 2 (D)), and the edges of DE is scaled to that of DC correspondingly (step 2 in Figure 2 (D)).

Repeat phase 2 to 4. If there is no node that satisfies a previous phase's requirements, the normalization enters the next step. Furthermore, if the normalization could not go further with any nodes in phase 4, it starts from the step 2 again in a new loop.

We run the normalization experiment on three networks, and we find that the above rules help us achieve high normalization rates. In particular, we manage to normalize (label nodes as solid) over 97.65% nodes of each of the three networks in 17 days. We are unable to achieve 100% normalization rates mainly for the following two reasons: (i) Nodes that are leaves in the network can not be normalized due to the absence of links. (ii) Nodes have children (links with other nodes), but they neither point back to their parents, nor share their children with other solid nodes.

Google trends regularly updates its data for websites every month. Building the network in the first cycle, and normalizing it in the other cycle, might induce errors in the normalization procedure. To avoid any effects of such errors, we decide to use the portion of the nodes that are normalized within the *same* cycle.[1] In particular, we normalize 217,515 (73.12%) nodes of the US-centric network, 214,405 (72.08%) nodes of the India-centric network, and 218,566 (75.23%) nodes of the China-centric network within the same cycle (12 days for normal-

[1] Google trends provides the month/year information for its data. In this way, we are able to realize that the crawling and normalization are executed within the same cycle.

ization and 27 days in total for crawling and normalization). While we necessarily reduce the normalized network size in this case, we assure that we induce no normalization error in the process. Moreover, we find that the network exhibits a *scale-free* phenomenon, as we explain in detail in Section III below.

D. Adding the absolute traffic into the network.

By normalizing the networks, we manage to adjust the weights of the edges in the networks to the same scale system. However, these weights do not reflect the absolute traffic values yet. Because one of our goals is to build a website selection application (Section IV), we strive to estimate the total number of unique users that an advertiser could reach based on its budget. Thus, our next goal is to convert the relative affinity weights into absolute weights.

Driven by this goal, we further crawl the number of unique users for each website in the dataset from Doubleclick ad planner [6]. For example, `www.nytimes.com` is tagged with the traffic of 55M unique visits monthly. We find that the figures shown in Doubleclick are the same as that shown in Google trends. This makes sense because Doubleclick [7] was acquired by Google [8], and hence its data is in line with Google's. This experiment is executed for 7 days in parallel with the normalization (Section II-C), and is just after crawling the networks (Section II-B). In this way, for each website in the network, we are able to label the number of unique users information with it. We use such absolute traffic values to convert the relative weights of edges in the network into absolute weights in Section II-E below.

E. Unifying two scale systems

So far we have a normalized website-affinity-based network where the weights of edges are in a *relative* scale system (Section II-C). Also, we have the number of unique users information in the *absolute* scale system for each website (Section II-D). Now, we are about to use this absolute traffic information to convert the relative weights of edges in the network into absolute numbers. Before moving to such normalization of two scale systems, we first verify one issue: For a website, are its top 10 children websites sufficient to represent the relations among itself and all its children? The answer to this question would help us to correctly conduct the effort of normalizing two scale systems later.

Top 10 children are sufficient. For a website, we compare the traffic from other websites and the traffic from its top 10 children to check the proportion of the traffic that its top 10 children take up. As we discussed in Section II-A, if two websites are searched together orderly in Google trends, for the second website, Google trends will provide the affinity ratio between the second website and the first website's children, instead of between the second website and its children. Based on this mechanism, we conduct the experiment in the following steps. We first randomly choose 500 candidates for each network. We then select a website from the pool of all the *remaining* websites (217,015 = 217,515 -500 in the US-centric network, 213,905 = 214,405 - 500 in the India-centric network and 218,066 = 218,566 -500 in the China-centric network) in the normalized networks. Next, we search this website with a candidate orderly at the same time to normalize the weights of edges between its children and the candidate. We repeat this process until the pool of other websites is exhausted for each of all candidates.

Certainly, such experiment yields a large amount of requests to Google trends (total traffic in each case of three networks = 500 * the number of non-leaf nodes in the network), and takes up another three days (27 days for crawling and normalization, 30 days in total which are in the same cycle). These are exactly the reasons why we limit ourselves to a reasonable size of testset, *500* in particular. Finally, for each of the candidates, we examine the proportion of the traffic from its top children to its total traffic. We find that the top (up to) 10 children account for over 77.16% of traffic, which is a *lower bound* for all 500 web sites that we explored. Thus, we show that a node has strong affiliation only with its top few children. The aggregation of normalized weights of edges between a node and its children beyond the top 10 is certainly small. This implies that our result is in line with the well-known Pareto principle [9], [10], [11] (also known as the 80-20 rule). Indeed, our results show that approximately 20% of the children websites with strong affinity provide 80% traffic volume for the parent website. Consequently, for a website, the traffic from its significant children are sufficient enough to represent its total traffic.

Converting the relative weights to absolute weights. We conduct the normalization of two scale systems in the following steps. Given the fact that for each website, its top 10 children account for over 77.16% of its total absolute traffic. We first obtain the total absolute traffic that its top (up to) 10 children websites take up in a straightforward way. Then, for each edge between a child and its patent, we calculate its absolute value based on the proportion of the weight of this edge to the weights of edges between its parent to all children. For example, in Figure 1, assume that for website A, its number of unique users is 168.48K, then the absolute traffic that its top (up to) 10 children take up is 130K. Next, the absolute weight of AD is 100K = 168.48K * 77.16% * 100 / 130. Accordingly, the absolute weights of AC and AE are 20K and 10K respectively.

F. Validation

In this section, we verify the issue: Does the normalized network accurately reflect the affinities of websites on the Internet? For this purpose, we utilize the number of unique users information from Doubleclick ad planner as the criterion to verify the correctness of our normalized network. To provide a baseline, we also evaluate our un-normalized network. In particular, for each node in both networks, we consider the sum of the weights of its edges as its total traffic. We realize that such substitution is not fully accurate, since the two networks do not consider the traffic from weak affinity websites. However, as we explained above, it is sufficient for us to evaluate the network features.

Next, we rank the websites in both un-normalized and normalized networks in terms of their traffic. In particular, for each website in the un-normalized network, we simply add the (un-normalized) weights of edges, while in the normalized one, since the traffic has been scaled to the standard weight, we add the normalized weights of edges. We then rank the website based on its total number of unique users obtained from the Doubleclick ad planner. We finally compare the lists of ranking in both cases with the ranking in Doubleclick. We find that most websites in the ranking list of normalized network have the exact sequence in Doubleclick's list, *e.g.*, 159,460 (73.31%) websites in the normalized US-centric trace (we explain this number on more detail below). However, only 32,126 (14.77%) websites in the un-normalized US-centric trace have the correct sequence. Such a huge discrepancy between two networks justifies the efforts of normalization.

The reasons for the normalized network being unable to achieve 100% ranking match lies in our approach's limitation in normalizing the leaf nodes. Indeed, our additional experiments show that the ranking for *non-leaf* nodes in our network follows the Doubleclick popularity list in as many as **91.66%** of all cases. In summary, we show that our normalization efforts yield highly accurate results, and hence we proceed with evaluating the properties of our normalized networks below.

III. Network Structure Analysis

In this section, we attempt to completely characterize our network. For the purpose of this analysis, we use three datasets, which are a US-centric network, an India-centric network, and a China-centric network. The reasons for using the three datasets is to study regional-based variations, *e.g.*, if the network properties are different based on their sampling seeds. Later, we show that the network properties change very little with a change in the network seed and therefore, it is possible to estimate similar properties for other networks with different sampling seeds.

A. Network properties

1) Weighted Degree Distribution: Degree distribution is defined as the distribution of the number of links that the nodes in a network have to other nodes. We plot the weighted degree distribution of all nodes for each dataset. The results are displayed in Figure 3 that is presented in a log-normal scale.

Our results show that the weighted degree distribution for all networks is almost the same and follows a *log-normal* curve. Another noticeable fact is that most degree weights are concentrated between 10 and 10,000 for all networks (the large portion shown from 1 to 4 of the x-axis in Figure 3). The distribution for three networks shows a certain diversity in terms of degrees. The negative x-axis extends to -4 in the case of the US-centric network indicating that there are nodes that display very small degrees. For the China-centric network, however, the positive x-axis is over 8 showing that there exist nodes that have very large degrees. Moreover, the peak of the curves are seen between 2 and 3 for all three networks. Furthermore, the number of nodes reaching peak in the US-centric network is larger than that in India-centric network and China-centric network. As we will explain later, such nodes with very high degrees are strongly connected in the network. Clearly, despite the similar distributions, we can notice some distinguishing elements in three networks. More of these similarities and differences are highlighted in the following sub-sections.

TABLE I
Summary of average path length and diameter results

	US	India	China	WEB	Online social networks
Nodes	217,515	214,405	218,566		
Average Path	7.033	7.141	7.318	16.12	4.25 to 5.88
Diameter	15	15	15	905	9 to 27

Compared with Web [12] and online social networks [13] that uniformly follow a power-law distribution, our network follows a log-normal distribution. Such a distribution is due to the fact that the traffic of user migration among websites are usually similar. Likewise, our network does not show web sites with insignificant traffic volumes, because such sites are filtered out by `Google Trends`. Hence, only very few websites have very large traffic, and very few websites exhibit very small traffic too.

2) Average Path Length and Diameter: Average path length for a network is defined as the average of shortest path lengths between every pair of nodes. It is a measure of the efficiency of information transfer on the network. Diameter, on the other hand, is defined as the maximum of all shortest path lengths between any two nodes. The results are captured in Table I.

Comparing with US-centric and India-centric networks, the China-centric network has a relatively larger average path length due to the way these networks are clustered (Section III-A3). With the experiments conducted, we discover that Chinese dataset consists of clusters containing nodes with high degree (high-degree clusters) in the core, and the clusters with less frequent, low degree nodes (low-degree clusters) on the fringes. The reason for the larger average path length displayed at the China-centric network is because high-degree clusters and low-degree clusters are loosely connected.

The US dataset exhibits the shortest average path length among the three networks. This attributes to the fact that the nodes with high degree are clustered in the core, and in turn well connect with other nodes. As we will demonstrate in Section III-A3, the India-centric network tends to be uniformly clustered. Thus, its average path length is larger than US's, but shorter than China's. Despite the variance among the three networks, we can notice that the difference is not significant, and it is independent from the seed where we start crawling. Furthermore, such seed-free phenomenon is even more clear on the diameter property, due to the networks' similar sizes.

The Web [12] has been shown to have a diameter of 905 and average path length of 16.12. Each of the four social networks studied in [13] have average path lengths lying between 4.25 and 5.88, and diameters ranging from 9 to 27. Given our network structure's similarity to the construct of social networks, the similar values are not unexpected. In the Web domain, websites in different regions or with different topics may not be connected which makes the Web's diameter and average path length incredibly larger. However, websites in our networks are connected by users' migration, even though they do not have a physical link at the network-level. Thus, our

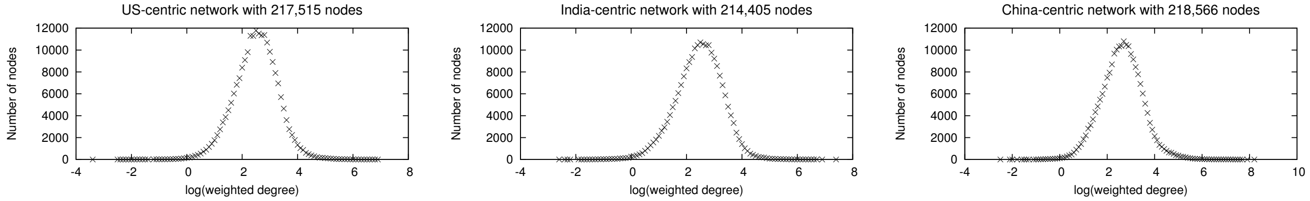

Fig. 3. Weighted degree distribution of three networks

network is more connected than Web. (We will demonstrate this fact in Section III-A3 below.)

TABLE II
SUMMARY OF CLUSTERING COEFFICIENT RESULTS

	US	India	China	WEB	Online social networks
Nodes	217,515	214,405	218,566		
Clustering	0.2332	0.2336	0.2443	0.081	0.136 to 0.33

3) Clustering Coefficient: The clustering coefficient of a node with N neighbors is defined as the number of links that exists between the nodes N neighbors divided by the number of possible links that could exist between the nodes neighbors (N*(N-1)). The clustering coefficient of a graph is, then, defined as the average clustering coefficient of all its nodes. The results are listed in Table II along with parameters in Web and online social networks.

It is clear from the table that the clustering coefficients are almost identical irrespective of the sampling location. The slightly lower values for the US-centric network and India-centric network indicate that they are not as strongly clustered as the China-centric network. Combining this result with the large number of nodes with high degree observed in Section III-A1, and the larger average path length learned in Section III-A2, we confirm that the nodes with very high degrees and these with low degrees are separately clustered and loosely connected with each other in the China-centric network.

Nodes in the US dataset, on the other hand, are clustered in a different way. In particular, nodes with high degrees are clustered in the core, while low degree nodes are not well clustered and scattered in the network. Thus, the clustering coefficient is smallest. The India-centric network sits in between. In particular, the nodes are more uniformly clustered, compared with two other networks.

Table III shows the ratios of the observed clustering coefficient of the proposed networks to those of their corresponding Erdös-Rényi (ER) random graphs [14], with the same number of nodes and edges. ER graphs have no link bias towards local nodes. Hence, they can be considered as a baseline for the degree of clustering in the proposed networks. In particular, the clustering coefficient of the three networks in Table III are four orders of magnitude larger than their corresponding ER random graphs. Such a high clustering coefficient, suggesting the presence of strong clustering, can be explained as follows. A group of online users usually visit a number of same websites that are in turn strongly connected by 'virtual' links from users' perspective.

TABLE III
THE RATIOS OF THE OBSERVED CLUSTERING COEFFICIENT OF THE PROPOSED NETWORKS TO THESE OF THEIR CORRESPONDING ERDÖS-RÉNYI GRAPHS

	US	India	China
Nodes	217,515	214,405	218,566
Clustering	0.2332	0.2336	0.2443
Ratio	2814.23	2790.19	2856.09

Combining the large clustering coefficient and the small average path length and diameter observed in Section III-A2, we recognize that our network is a *small-world* network [15]. The values we obtained are also close to the online social networks clustering coefficients which range from 0.136 to 0.33 [13], but far away from Web's clustering coefficient that is 0.081 [16]. The higher value of our networks clustering coefficient as compared to the Web is clearly because of the strong relationships between the nodes in our datasets, due to construction. The nodes in our datasets exhibit close relationships because the linkages represent the weights of user movement from one node to another. Online social networks on the other hand would tend to have even stronger relationships and, as a result, higher clustering coefficients.

Seed-Independent properties. For the parameters we have investigated, they do have a small difference in terms of their values displayed on different networks. However, as we pointed out previously, such variance is not significant, and thus, we conclude that our networks exhibit a *seed-free* phenomenon.

Scale-Independent properties. Here, we investigate if the network properties change significantly along with the increase in network size. We present the results for the India-centric network due to the space limitation. In particular, the experiments conducted in previous sections are based on the India-centric network with 214,405 nodes (72.08 % of the network normalized after 12 days). Next, we compare these results with the ones obtained from the same network that consists of 290,465 nodes (97.65% of the network normalized after 17 days). In this bigger network, the most degree weights again are concentrated between 10 and 10,000 and the peak is observed between 100 and 1,000. The average path length of 7.531 (was 7.141), the diameter of 16 (was 15), and the clustering coefficient of 0.2454 (was 0.2336) demonstrate that the network properties change very little with a change in the network size and therefore, it is possible to estimate similar properties for a bigger, more informative network.

IV. Website Selector

In this section, we explore the utility of our user-driven network. Given the structure of this network, the most obvious use lies in e-commerce and the specific area that we target is online advertising. In particular, advertisers want maximum ”bang for the buck” or the maximum number of visitors they can get for their ads within their budget. By providing the shared user information, our graph ensures that the advertisers are not paying double for the same set of visitors and gives them the optimum visibility for their budget.

A. CPM revenue model

Currently, there are three popular revenue models in use for online advertisements - CPM (Cost Per Mille), CPC (Cost Per Click), and CPA (Cost Per Acquisition). For the purpose of our research, we restrict our attention on the CPM model. Under this model, the advertiser is charged in multiples of thousand impressions. An impression is defined as a load of an advertisement. (This excludes page refreshes or reloads.) This is similar to other traditional advertising schemes, where the advertiser has to pay for his advertisement irrespective of whether he can generate any mindshare or revenue with it.

B. Cost Optimization

Under the traditional model of advertising, an advertiser or the commissioner has three main decision criteria when creating a list of websites that the advertiser may want to work with. These are - $i)$ his budget, $ii)$ the cost associated with advertising on each of these websites, and $iii)$ the popularity/ranking of websites. Simply put, the goal he seeks to achieve is getting the maximum viewership for his advertisement within his budget. With our research, we show that by including a fourth input in the decision process - $iv)$ the number of shared users - the advertiser can get maximum ”bang for his buck”. The improvement achieved over the traditional optimization problem can be as high as 22-26%.

We model the problem using simple non-linear optimization. As inputs to the problem we have - $i)$ the advertiser’s overall budget, $ii)$ the CPM for each website, $iii)$ the number of unique users (or popularity) of each website, and $iv)$ the number of shared users between every two websites. The first two inputs help us define the constraint for the optimization problem and the last two help in defining the objective function. Mathematically, the problem is stated as follows.
Let:

- Y_i : the binary decision variable representing selection/rejection of a node i from the advertising plan
- U_i : the number of unique visitors associated with node i
- C_{ij} : the number of shared visitors between node i and node j
- S_i : the CPM associated with node i
- B : the advertiser’s budget

TABLE IV
Performance improvements realized using the ’*sub-optimization*’ and the ’*website selector*’, keeping the ’*greedy approach*’ as baseline

	Sub-optimization	**Website selector**
US network	0.06%	24.85%
India network	0.18%	22.38%
China network	0.12%	25.79%

Then the objective function can be defined as

$$f = Max(\sum_{i=1}^{n} U_i \times Y_i - \sum_{\substack{i,j=1,\\ i<j}}^{n} C_{ij} \times Y_i \times Y_j)$$

$$Subject\ to \sum_{i=1}^{n} S_i \times Y_i \leq B$$

, where n is the number of websites in our dataset. This makes it an optimization problem with a non-linear objective function and a linear constraint. There can be a concern that we are ignoring higher order relationships by assuming that the shared user base between any nodes i and j is completely independent of the shared user base between another set of nodes i and k. The concern is well founded, but, because we restrict our attention to publicly available data only, we have to model all binary relationships as exclusive of each other. An ad commissioner can have a more holistic view of the network traffic and would be able to explore the higher order relationships easily. Moreover, it is easy to prove that this approach, by ignoring higher order relationships, is underestimating the objective function. (Clearly, by reducing the effective number of shared users, we will only increase the value of f.) Our results are, therefore, far from being overly optimistic. Given more information about user relationships between nodes, the model will only better itself.

C. Experiment

We design a tool, which we call the ”website selector”, around the presented model. We test it on three datasets - US-centric network, India-centric network, and China-centric network. To solve the optimization problem, we use Knitro [17], a commercially available optimization tool. The problem is modeled using Matlab. A random normal distribution is used to compute the CPM values for the websites, since this information is not publicly available. But it is assumed that this information will be readily available to the concerned decision makers.

We compare our results with those achieved using two other techniques - $i)$ a simple greedy approach where we choose the websites in ascending order of their CPM until we reach our budget constraint (greedy approach), $ii)$ a linear optimization approach to maximize the number of unique visitors subject to the budget constraint, ignoring the shared users information (sub-optimization).

The Table IV presents a clear picture of the effectiveness of the website selector in maximizing viewership for the advertisers. In particular, the viewership advantage provided by the website selector is as high as 24.85% for the US-centric network, 25.79% for the India-centric network, and 22.38% for the China-centric network, when compared to the greedy approach, constrained by the same budget of $50,000. The results achieved by the sub-optimal approach and greedy approach are almost identical, as shown in Table IV.

The impact of the solution is also evident in the fact that for the US-centric dataset approximately 17% of the top 1000 extremely high volume traffic sites (monthly traffic $>$ 4.5M) are eliminated because of their shared user relationships with other websites, while only 0.1% are eliminated using the sub-optimal approach, which does not use the shared user data. The figures are 17% and 0% for the top 1000 in the Indian dataset, and 16% and 0% respectively for the top 1000 websites in the Chinese dataset. Intuitively, 0% of the top websites will be eliminated under the greedy approach.

By eliminating the high volume, high cost websites, based on the shared relationships, our methodology is able to deliver a net higher number of unique viewers within the same budget. The results prove conclusively that shared user information should be an integral part of decision making for advertisers seeking to achieve a $'true'$ maximum visibility within their budgets.

V. Related work

Here, we present the related work in the following two areas: (i) using Google Trends data and (ii) optimizing online advertising. In the Google Trends domain, our work relates to the previous work [18], [19] in the sense that the authors also utilize the information from Google trends. In particular, authors in [18], [19] could track and predict flu-like illness in a population by analyzing health-seeking behavior in the form of online web search queries. A detection of disease activity in a place can be made, if a volcanic increase of searching on flu-related keywords is observed. While we share the same source of data, our work differs from theirs in that we exploit information about the affinities among websites, which is another valuable resource given by Google trends, to investigate the properties of a user-driven network.

In the online advertising domain, our work reflects a frequency capping requirement for the advertisers [20]. Few major ad commissioners implemented this requirement in a different way. In particular, individual users are tracked across websites in terms of their cookies. Thus, the same ad is not shown to the user, once the number of its appearance (to that user) is more than the specified frequency capping. Besides the fact that such cookie-based approach is still being challenged [21], this method is incapable of predicting the number of individual viewership an advertiser can reach in advance, and selecting the best website candidates with which an advertiser wants to collaborate.

VI. Conclusions

In an attempt to understand how users' interests collectively spread on the Web, we design a crawler which gathers the information about the affinities among websites from Google trends to form a user-driven network. We propose, utilize, and evaluate a series of approaches to globally normalize this network and validate its high accuracy. Based on this user-driven network, we design an optimization application, website selector, that can be used for advertisers' campaigns. The application effectively selects a group of websites whose number of common users is minimal subject to advertiser's budget and targeting preferences.

Our key contributions are the following. (i) We discover that the user-driven Web network exhibits an evident small-world phenomenon; it has a small average path length and a large clustering coefficient. We also find that the network properties are both scale-free, *i.e.*, the properties are irrespective of network size, and seed-free, *i.e.*, the properties are independent from the seed where we start crawling the network. (ii) In comparison with the Web and online social networks, we find that the user-driven network substantially differs from the Web network, but to some extent is closer to online social networks. Hence, we conclude that the human component dominantly affects the properties of this human-driven Web network. (iii) We extensively evaluate website selector and the results demonstrate that it is able to increase the visibility of ads by more than 22% and consequently increase revenues for advertisers. (iv) Our website selector system is tailored for ad commissioners and it could be easily embedded in their ad selection algorithms.

References

[1] T. Karagiannis and M. Vojnović, "Behavioral Profiles for Advanced Email Features," in *WWW '09*.

[2] "Google latitude," http://www.google.com/latitude.

[3] A. Mislove, A. Post, P. Druschel, and K. P. Gummadi, "Ostra: Leveraging Trust to Thwart Unwanted Communication," in *NSDI '08*.

[4] "Google trends," http://trends.google.com/websites.

[5] "Facebook popularity eats company bandwidth," http://www.networkworld.com/news/2010/081710-facebook-popularity-eats-company.html.

[6] "Doubleclick ad planner — google," www.google.com/adplanner/.

[7] "Double click," http://www.doubleclick.com/.

[8] "Internet marketing news: Doubleclick deal means Google controls 69% of the online ad market," http://www.browsermedia.co.uk/2008/04/01/doubleclick-deal-means-google-controls-69/-of-the-online-ad-market/.

[9] A. Bookstein, "Informetric distributions, part i: Unified overview," *Journal of the American Society for Information Science*, 1990.

[10] O. S. Klass, O. Biham, M. Levy, O. Malcai, and S. Solomon, "The forbes 400 and the pareto wealth distribution," *Economics Letters*, 2006.

[11] R. Koch, *The 80/20 Principle: The Secret of Achieving More with Less*. Nicholas Brealey Publishing, 2007.

[12] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, "Graph structure in the web," in *WWW '00*.

[13] A. Mislove, M. Marcon, K. Gummadi, P. Druschel, and B. B. charjee, "Measurement and Analysis of Online Social Networks," in *IMC '07*.

[14] P. Erdös and A. Rényi, "On random graphs. i." *Publicationes Mathematicae Debrecen*, 1959.

[15] Duncan J. Watts and Steven H. Strogatz, "Collective dynamics of śmall-worldńetworks," *Nature*, 1998.

[16] L. A. Adamic, "The small world web," in *Research and advanced technology for digital libraries '99*.

[17] "Knitro," http://www.ziena.com/knitro.htm.

[18] "Flu trends," http://www.google.org/flutrends/.

[19] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant, "Detecting influenza epidemics using search engine query data," *Nature*, 2009.

[20] A. Farahat, "Privacy preserving frequency capping in internet banner advertising," in *WWW '09*.

[21] "Recent lawsuits challenge use of flash cookies to track online behavior," http://privacylaw.proskauer.com/2010/09/articles/behavioral-marketing/recent-lawsuits-challenge-use-of-flash-cookies-to-track-online-behavior/.