Computer Communications 41 (2014) 22–30

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Identifying website communities in mobile internet based on affinity measurement

Jun Liu [a,*], Nirwan Ansari [b]

[a] School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China
[b] Advanced Networking Lab., Electrical and Computer Engineering Department, New Jersey Institute of Technology, NJ, USA

ARTICLE INFO

Article history:
Received 20 August 2013
Received in revised form 26 November 2013
Accepted 30 December 2013
Available online 22 January 2014

Keywords:
Website community
Graph theory
Affinity measurement
Degree distribution

ABSTRACT

With the rapid development of mobile devices and wireless technologies, mobile internet websites play an essential role for delivering networked services in our daily life. Thus, identifying website communities in mobile internet is of theoretical and practical significance in optimizing network resource and improving user experience. Existing solutions are, however, limited to retrieve website communities based on hyperlink structure and content similarities. The relationships between user behaviors and community structures are far from being understood. In this paper, we develop a three-step algorithm to extract communities by affinity measurement derived from user accessing information. Through experimental evaluation with massive detailed HTTP traffic records captured from a cellular core network by high performance monitoring devices, we show that our affinity measurement based method is effective in identifying hidden website communities in mobile internet, which have evaded previous link-based and content-based approaches.

1. Introduction

Mobile internet refers to accessing the internet websites from mobile devices via cellular networks. With the rapid increase of powerful mobile devices, innovative mobile applications and increased cellular spectrum allocation, the traffic volume of mobile internet has been growing continuously [1,2]. Video traffic was 57% of all consumer internet traffic in 2012 and is growing [1], and there have been constant efforts in speeding up content delivery [3,4] and enhancing user experience [5–7]. To cope with the explosive growth of data volume and best serve their customers, mobile network operators need to design and manage their network resources from the overall perspective of the entire mobile internet. Basically, mobile internet services are delivered to users via cooperation among mobile devices, cellular networks and websites. From the viewpoint of service providers, user activities are centered around websites. Therefore, the community structure of websites can be a rich source of information about the circumstance of the mobile internet. In this paper, we propose a new algorithm to extract the community structures of websites in mobile internet and demonstrate its effectiveness by experiments with real-world traffic data. In particular, we focus on how to measure the affinity relationships between websites and use them for identifying important web communities in the mobile internet environment.

Despite the decentralized, unorganized and heterogeneous nature of world wide web, some works have shown that the complex web system can be described by graphs or networks which capture the connections between the entities they are made of, such as clients and websites [8]. In a graph, some nodes maintain closer relationship with each other than the rest of the graph. The set of such nodes is usually referred to as community, cluster or module. It is important and interesting to discover these *a priori* unknown community structures in graphs. The purposes of initial works on investigating the community structures are for structure visualizing [9,10] and content searching [11,12] in the web environment. Subsequently, there have been a growing number of works directed at revealing community structures based on various context information. In general, these works can be categorized into two types. The first type is the link-based approach which extracts community information from the link structure of the hyperlinked web environments. Another type is the content-based approach which defines the relationships between websites in terms of similarities between their contents, such as title, keywords, description, and words in the web pages. Most of content-based works use a vectorial representation of web pages to cluster web sites by related topics. Owing to the semantic relevance between the definition of relationships between websites in the above two types [13], some works have been proposed to combine link and content information for identifying web communities [14].

* Corresponding author. Tel.: +86 10 62283742.
E-mail addresses: liujun@bupt.edu.cn (J. Liu), nirwan.ansari@njit.edu (N. Ansari).

0140-3664/$ - see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2013.12.008

Although there has been tremendous interest in identifying website communities, earlier studies mainly focused on the original information of web pages collected by various crawlers or through web data mining. To a certain extent, the existing relations between website communities and user behaviors have been overlooked and are far from being well understood. In this paper, we carry out the mining of website communities from the user perspective by examining detailed HTTP traffic records in mobile internet. Our traffic records were captured by powerful line-speed monitoring devices which tracked a 10Gbps trunk link in the backbone of a cellular data network. The records consist of complete information of accessing users, used devices and serving websites. The rich information allows us to establish the relations between website communities and accessing users. Based on the examination of these records, we propose a shared user based affinity measurement between websites and build an affinity graph to represent the structure of observed mobile websites. However, the original dense affinity graph may poll too large a set of relations between nodes that will incur high computational workload as well as decrease the fidelity of the mining task. So, we modify a scale-free topology criterion, originally designed for undirected graph, to transform the full directed affinity graph into a sparse one by choosing a threshold parameter value that leads to a graph whose in-degree distribution follows a power law. Then, the influence score, which represents the importance of each website in the sparsified affinity graph, can be calculated based on the out-degree values and weights of its neighbors. At last, all websites are ranked by the calculated influence scores. The k websites with top scores and their neighboring websites in the sparsified affinity graph are identified as top-k website communities. Convincing results based upon real-word traffic data have substantiated the effectiveness of our systematic method.

The main contributions of our work are twofold. First, we have proposed a new algorithm for identifying top-k website communities in mobile internet. Unlike existing solutions, our method can identify the hidden community structure from the user behavior perspective, which cannot be revealed by existing link-based and content-based community identification approaches. Second, we have conducted experiments on a large cellular data network with massive HTTP traffic data to evaluate the effectiveness of our algorithm. Experiments on real-world data show that our algorithm can effectively identify communities in which websites have strong affinity relationships. Two novel types of website communities are identified and explained in the evaluation.

The rest of this paper is organized as follows. Section 2 provides a brief review of the background and related works. Section 3 details our proposed method for identifying website communities. We then present how we evaluate the effectiveness of our method and discuss the results in Section 4. Finally, we conclude the paper in Section 5.

2. Background and related works

Website community identification in the web environment has attracted ample attention in recent years. The most basic and straightforward approach for identifying communities is grouping web objects according to their natural hyperlink structure characteristics. A web environment can be modeled as a graph in which a node represents a website or web page on various contexts and an edge represents a link from one node to another. After abstracting the web as a graph, a set of graph theory based mathematical tools can be applied to detect communities for different purposes. For example, Gibson et al. [15] defined a hyperlinked community on web which contains a core of authoritative central node and linked hub pages. Based on their proposed HITS algorithm [11], community structures can be derived through the link topology. Another graph theory tool applied in identifying web communities is the maximum flow and minimal cut theorem. Flake et al. [16] defined a community as a set of web pages that link to more pages in the community than to pages outside the community. Under this definition, the problem of identifying communities is mapped into a family of graph partitioning problems. The identification procedure is carried out as a loop for a given number of iterations using the maximum flow and minimum cut algorithm. Also, Merelo-Guervs et al. [17] proposed a Self-Organizing Map (SOM) based method to divide a set of blog websites into communities and produce a community navigation map. Besides the graph model, vectors are also used to represent the web objects for both content similarity and hyperlink structure based considerations [18–20]. The advantage of vector representation is that it is suitable for applying clustering technologies based on well studied vector oriented distance measuring algorithms like k-means and k-nearest neighbors clustering. For example, works in [21,22] clustered websites by computed distances between vectors, which represent the topic features and link weights respectively, and both yielded reasonable results.

Owing to the high computational complexity of crawling and parsing contents and links in web pages, the above link-based and content-based methods do not scale well, i.e., computationally expensive in a large scale environment. In addition, absence of user behavior information in these procedures renders the identified communities useless in some situations. We present two motivating examples to understand the importance of identifying user behavior related website communities in mobile internet. The first example deals with the replication of hotspot web contents to reduce cost and improve user experience. With the increasing bandwidth of cellular data networks, users are consuming more resource-hungry and quality sensitive services on their mobile devices, such as video and online gaming applications. To reduce network traffic from the third party for cost saving and to shorten response time for improving user experience, mobile operators tend to build self-owned web servers for replicating hotspot web contents. Because of the copyright issue, they have to negotiate with the owners of such websites for importing hotspot contents one by one. Hence, choosing appropriate websites to replicate in a limited resource condition including space, power and bandwidth is becoming critical. A straightforward method is to replicate web objects based on the rank of accessing popularity [23]. This simple approach, however, requires a long time to accumulate enough request statistics for each object. During the collection time, the hot object may not be popular anymore. Therefore, mobile operators need a more intelligent method to find correlated popular websites to be replicated. The second example is the requirement of identifying the latent service dependency between websites. With advances of web services technologies, some important websites which provide critical web services are hidden behind other websites, such as authentication server for single sign-on, advertising content server for advertising display websites, and streaming servers. The relations between these service providing websites and supporting websites may not be obvious in the hyperlink structure because most of them are implemented by dynamic programming interface rather than static texture link. The user access logs are the only information that can be relied onto infer the relationships between websites.

In above two examples, the common goal is to find some communities in which websites are correlated by shared users rather than hyperlink structure or content similarity. We call such a set of websites as an *affinity community*. In this paper, we propose a means to identify such communities by: (a) considering the relationships between websites based on the user behavior information, (b) quantifying the top-k important communities, and (c) allowing overlapped communities. Towards this end, we design an algorithm with three steps including measuring affinity, sparsifying graph and identifying communities. Our approach, which incorporates the above three features, differs from existing methods of identifying communities in complex networks such as social networks and biological networks.

3. Methodology

In this section, we present details of our methodology for identifying website communities. We first briefly explain notations adopted in this paper. Then, the three steps of our method along with the corresponding algorithms are presented in detail.

3.1. Method overview and preliminaries

Our method consists of three steps, namely, measuring affinity, sparsifying graph and identifying communities. Table 1 lists the main notations used to describe our method. Formally, we model the mobile internet websites as a directed and weighted graph $G = (O, E, W)$, referred to as an *affinity graph*, where O is the set of nodes representing websites and E is the set of directed edges with weights denoting the affinity relationships between websites. The affinity weight from a node to another is derived from the number of common users between two websites and the total number of users of each website $\mu(o_i)$ by an asymmetrical affinity function $aff(o_i, o_j)$. The asymmetric affinity value, which is between 0 and 1, is like the degree of love relationship between two persons: one loves another does not mean the reverse is true. To reduce the computational workload and improve identification quality, the original dense affinity graph is transformed into a sparse graph G', where E' is a set of unweighted edges between nodes. The structure of G' is inferred by optimizing the scale-free fitting index *fit*, which is computed by the frequency distribution of node in-degree d_i^{in}. The influence score $ins(o_i)$ of each node can be calculated by the affinity rates $affr(o_j, o_i)$ of all neighboring nodes o_j around o_i in the sparsified affinity graph. The nodes are ranked by the influence scores and the top-k scorers are selected as the seed set R. Each influential node o_i in R and the set of nodes o_j having a directed edge from o_j to o_i are grouped together as the ith community C_i.

3.2. Measuring affinity

The first step of our algorithm is generating a graph to model the mobile internet websites and relationships between them in a formal way. Affinity measure is a key factor that is used for analyzing complex networks in various domains including physics, biology and sociology. To take advantage of the power of affinity measurement based methodology, we have to define the concept of *affinity* between websites in mobile internet. For this purpose, we first introduce the affinity measurement function $aff: O \times O \rightarrow A$ as: for a given dataset O with n objects $O = \{o_1, o_2, \ldots, o_n\}$, the function $aff(o_i, o_j)$ produces the affinity measurement value between each pair of objects. In general, we assume that the affinity with itself for every object o_i is 0, i.e., $aff(o_i, o_i) = 0$. With the definition of affinity measurement, we can construct a directed and weighted affinity graph $G = (O, E, W)$ with every node in O representing an object and W being the weight value derived by the *aff* function on edge set E. O can represent a large range of objects, such as genes in a biology network, individuals in a social network, nodes on internet, *etc*. Meanwhile, *aff* could take on different forms in different domains. As a well-studied research topic, there are many different ways of computing the affinity of objects [24]. At the same time, the affinity function is not restricted to be a symmetric function. For certain applications, the affinity function *aff* may be asymmetric, i.e., $aff(o_i, o_j) \neq aff(o_j, o_i)$. The asymmetric feature allows the affinity measure to be used in more general graph-based applications. Thus, for every pair of nodes in G, there may exist two directed edges between them, weighted by $aff(o_i, o_j)$ and $aff(o_j, o_i)$. Therefore, the structure of a group of objects can be modeled as a directed and weighted graph based on the asymmetric affinity function.

Table 1
Adopted notations.

Notation	Description
$G = (O, E, W)$	The directed and weighted affinity graph with node set O, edge set E and weight W
$\mu(o_i)$	The set of users accessing node o_i
$aff(o_i, o_j)$	Affinity measurement of o_i for o_j
$spa(o_i, o_j)$	The sparsification function for o_i and o_j
$G' = (O, E')$	The unweighted sparse affinity graph with node set O and edge set E'
d_i^{in}	The in-degree of node o_i in graph G'
fit	The scale-free fitting index of graph G'
$ins(o_i)$	The influence score of node o_i
$affr(o_j, o_i)$	The affinity rate of node o_j for o_i
R	The seed set of k influential nodes
C_i	The ith community

In order to build the graph to reflect the relationships between websites in mobile internet, we have to find a proper resource that websites have in common for establishing the relationship. As we know, the duty of mobile internet website is providing network services to a given set of users. A set of nodes $O = \{o_1, o_2, \ldots, o_n\}$ share a set of users $U = \{u_1, u_2, \ldots, u_l\}$. The set of users accessing each node in O belongs to the power set of U, denoted as $\mathbb{P}(U)$. We define the function $\mu : O \rightarrow \mathbb{P}(U)$ which maps each website to a user set. In other words, $\mu(o_i)$ denotes the set of users accessing node o_i. Clearly, we have $\cup_{i=1}^{n} \mu(o_i) = U$. So, we can define the affinity measurement function in our domain as:

$$aff(o_i, o_j) = \begin{cases} \frac{\mu(o_i) \cap \mu(o_j)}{\mu(o_i)}, & i \neq j \\ 0, & i = j \end{cases} \qquad (1)$$

In Eq. (1), $\mu(o_i) \cap \mu(o_j)$ is the set of users shared by both o_i and o_j. The denominator $\mu(o_i)$ is used to normalize the affinity measurement. From $\mu(o_i) \cap \mu(o_j) \leqslant \mu(o_i)$, we have $0 \leqslant aff(o_i, o_j) \leqslant 1$. Note that $aff(o_i, o_j) = 1$ if all users accessing o_i are contained by the set of users accessing o_j. In contrast, $aff(o_i, o_j) = 0$ if o_i and o_j do not share any user when $i \neq j$.

Having defined the affinity measurement function, we now introduce the concept of an *affinity graph*, which is a directed and weighted graph by generating edges between nodes using the following rule:

(Rule I) Affinity graph edge generation rule: A directed edge $e_{ij} \in E$ from o_i to o_j with weight $aff(o_i, o_j)$ is constructed iff $aff(o_i, o_j) > 0$; otherwise, no edge is constructed.

Under this edge generation rule, each edge in the graph is assigned a weight indicating the affinity relationship between the corresponding node pair. Thus, we get a directed and weighted graph $G = (O, E, W)$ reflecting the affinity relationship between mobile internet websites by their shared users. The affinity graph can be described by an affinity matrix W in which each entry corresponds to the weight of an edge in the graph, say, $w_{ij} = aff(o_i, o_j)$. The algorithm for constructing the affinity graph is summarized as a pseudo-code in Algorithm 1.

Fig. 1 illustrates a mobile web example. Nodes from a to l represent 12 websites. The number in bold text close to each node is the number of users accessing this website. The number in italic type close to each dashed line between two nodes is the number of common users of the two websites. Based on Algorithm 1, the affinity matrix of this example is calculated and shown in Table 2.

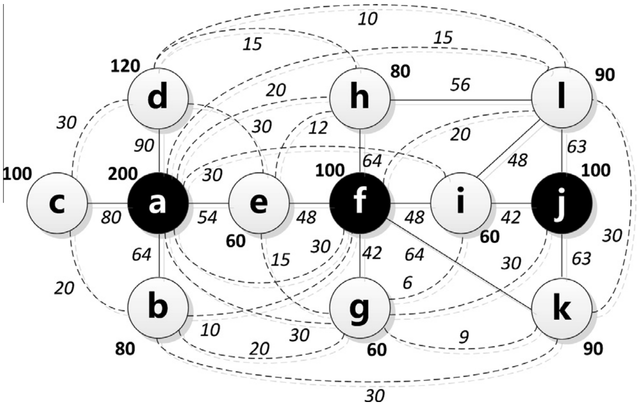


Fig. 1. Websites and user distribution example.

3.3. Sparsifying graph

In the previous section, we have described how to construct an affinity graph where the nodes are websites being studied and the weighted edges represent the affinity relationships between nodes defined by a function based on shared users. However, from the graph point of view, the full affinity graph may poll too large a set of relations between nodes. In reality, although every object is somehow related to many other objects, objects usually have strong relation with a small number of objects and weak relation with most of the other objects. Based on this property, sparsification is proposed to sparsify the dense graph in order to reduce the computational workload and improve the quality of analysis [24]. The definition of sparsification is application specific. Specifically, here we regard sparsification as the interpretation of the affinity matrix entries in order to enhance the quality and reduce the computation of identifying significant communities in the context of our study.

Algorithm I: Measuring affinity

```
Input:
$O = \{o_1, o_2, \ldots, o_n\} \rightarrow$ A set of websites
$\mu = \{\mu(o_1), \mu(o_2), \ldots, \mu(o_n)\} \rightarrow$ Accessing users of each
  website
Output:
$G = \{O, E, W\} \rightarrow$ The affinity graph

$O = \emptyset$
$E = \emptyset$
$W = \emptyset$
for i = 1 to $n$ do
  push $(O, o_i)$
  for j = 1 to $n$ do
    if i $\neq$ j then
      $aff(o_i, o_j) = \frac{\mu(o_i) \cap \mu(o_j)}{\mu(o_i)}$
    else
      $aff(o_i, o_j) = 0$
    end if
    if $aff(o_i, o_j) > 0$ then
      $e_{ij} = 1$
      push $(E, e_{ij})$
      $w_{ij} = aff(o_i, o_j)$
      push $(W, w_{ij})$
    end if
  end for
end for
```

Conceptually, sparsification methods can be categorized into two types [25]. The first one is the global threshold approach which uses a global threshold τ to eliminate edges with a proximity value below this threshold. The second one is the nearest neighbor approach which uses conditions involving the nearest neighbours of a node to sparsify the proximity matrix. The basic idea of the global threshold approach is very simple: all entries in the proximity matrix that are smaller than the given threshold are set to 0, otherwise they are set to 1. As compared with the nearest neighbor approach, the global threshold approach is more acceptable and widely used, especially in complicated situations with a large number of nodes and edges. The global threshold based sparsification function *spa* can be formally defined as:

$$spa(o_i, o_j) = \begin{cases} 1, & aff(o_i, o_j) \geqslant \tau \\ 0, & aff(o_i, o_j) < \tau \end{cases} \tag{2}$$

By adopting the above sparsification function, the original full affinity graph can be transformed into a directed and unweighted graph $G' = (O, E')$, named the *sparsified affinity graph*. The sparsified affinity graph can be regarded as a graph which draws out the "close" affinity relationship between node pairs with affinity measurement larger than the threshold τ. It is built by generating edges between nodes using the following rule:

(Rule II) Sparsified affinity graph edge generation rule: A directed and unweighted edge $e'_{ij} \in E'$ from o_i to o_j is constructed iff $spa(o_i, o_j) > 0$; otherwise, no edge is constructed.

Although the global threshold approach is simple and effective, it comes at the price that the parameter τ is application-dependent and has to be chosen carefully. The choice of threshold τ determines the sensitivity of the edge between node pairs. Obviously, increasing the value of τ leads to fewer edges, which may reduce the noise in the graph. Otherwise, a graph produced by too high τ value may be too sparse to detect expected communities. To choose an appropriate threshold, we leverage the scale-free topological criterion method proposed in [26]. The scale-free topological criterion, originally designed for undirected graph, cannot be straightforwardly used to sparsify our directed graph. Therefore, we will modify and tailor it for directed graphs as described below.

Many studies have shown that the frequency distribution of the in-degree values of nodes is an important characteristic of a graph [27,28]. In our sparsified affinity graph, the in-degree d_i^{in} of o_i equals the number of nodes that have directed edges to o_i. Using the affinity matrix of the sparsified affinity graph, the in-degree of node o_i can be expressed as:

$$d_i^{in} = \sum_{j=1}^{n} e'_{ji} \tag{3}$$

We employ the maximum likelihood method to estimate the frequency distribution of node in-degree in a sparsified affinity graph. Assume that the value of each d_i^{in} is between the minimum value (*minD*) and maximum value (*maxD*). We use the histogram method to partition the interval [*minD*, *maxD*] into equal-width subintervals bins. Specifically, the *r*th bin is defined as:

$$bin_r = [minD + (r-1) \times width, minD + r \times width] \tag{4}$$

where the width is given by:

$$width = (maxD - minD)/M \tag{5}$$

where M is the number of bins. Usually, the default value of M is the square root of the total number of nodes n. Then, the observed relative frequency of the *r*th bin is given by:

$$P(r) = \frac{O_r}{\sum_{r=1}^{M} O_r} \tag{6}$$

Table 2
Affinity matrix of the example.

	a	b	c	d	e	f	g	h	i	j	k	l
a	0	0.32	0.40	0.45	0.27	0.15	0.15	0.1	0.15	0	0	0.08
b	0.80	0	0.25	0	0	0.13	0.25	0	0	0	0.38	0
c	0.80	0.20	0	0.30	0	0	0	0	0	0	0	0
d	0.75	0	0.25	0	0.25	0	0	0.13	0	0	0	0
e	0.90	0	0	0.5	0	0.8	0.25	0.20	0	0	0	0
f	0	0.10	0	0	0.48	0	0.42	0.64	0.48	0	0.64	0.20
g	0.50	0.33	0	0	0.25	0.70	0	0	0.10	0.50	0.15	0
h	0.25	0	0	0.18	0.15	0.80	0	0	0	0	0	0.70
i	0.50	0	0	0	0	0.80	0.10	0	0	0.70	0	0.80
j	0	0	0	0	0	0	0.30	0	0.42	0	0.63	0.56
k	0	0.33	0	0	0	0.71	0.10	0	0	0.70	0	0.33
l	0	0	0	0.11	0	0.22	0	0.62	0.53	0.70	0.33	0

where O_r denotes the observed number of d_i^{in} that fall into the rth bin. Thus, the frequency distribution of the in-degree of the sparsified affinity graph can be estimated as $p(d^{in}) = (P(1), P(2), \dots, P(M))$. Many research studies have shown that the frequency distribution of in-degree in many real graphs especially internet follows a power law [27–29]:

$$P(r) = Cr^{-\alpha} \tag{7}$$

where $C = 1/\sum_{r=1}^{M} r^{-\alpha}$ and α is a positive real number. It is also said this kind of graph exhibits scale-free topology with scaling parameter α. To verify and illustrate the scale-free characteristic, researchers always take log_{10} of both sides of Eq. (7) to show a linear relationship between $log_{10}(P(r))$ and $log_{10}(r)$ as follows:

$$log_{10}(P(r)) = -\alpha \times log_{10}(r) + log_{10}(C) \tag{8}$$

Thus, a scale-free fitting index can be defined to measure how well a graph satisfies the scale-free topology criterion as:

$$fit = corr(log_{10}(p(d^{in})), log_{10}(v))^2 \tag{9}$$

$$= \left(\frac{\sum_{r=1}^{M}((log_{10}P(r) - \overline{log_{10}P(r)}) * (log_{10}(r) - \overline{log_{10}(r)}))}{\sqrt{\sum_{r=1}^{M}(log_{10}P(r) - \overline{log_{10}P(r)})^2} * \sqrt{\sum_{r=1}^{M}(log_{10}(r) - \overline{log_{10}(r)})^2}} \right)^2$$

where $v = (1, 2, \dots, M)$, $\overline{log_{10}P(r)} = \sum_{r=1}^{M} log_{10}P(r)/M$ and $\overline{log_{10}(r)} = \sum_{r=1}^{M} log_{10}(r)/M$. The $corr(\vec{x}, \vec{y})$ stands for the Pearson correlation between $\vec{x}$ and $\vec{y}$. Essentially, the *fit* is the square of the correlation coefficient between $log_{10}(p(d^{in}))$ and $log_{10}(v)$. The closer the *fit* index is to 1, the graph is considered to exhibit more scale-free topology property. In practice, there is a relationship between the *fit* and the threshold parameter τ like a saturation curve (as will be shown in our experiment results in Fig. 5). Meanwhile, the *fit* is not a strictly monotonic function of τ. So, we define the scale-free topology criterion rule according to the recommendation in [26] as follows:

(Rule III) Scale-free topology criterion rule: Choose the first parameter value as the expected threshold τ when the saturation of *fit* is above 0.8.

The reason of choosing the critical *fit* value as 0.8 instead of 1 is to make a trade-off between maximizing scale-free topology fit and maintaining a high mean number of edges in the sparsified affinity graph. Naturally, a high parameter value of τ leading to a *fit* value of 1 may generate a graph with very few edges. This may, however, eliminate useful information for the subsequent step of the method, i.e., identifying influential nodes.

The procedure of constructing the affinity graph is summarized as a pseudo-code in Algorithm 2. Executing this procedure on the output affinity graph shown in Fig. 1 results in the sparsified affinity graph shown in Fig. 2.

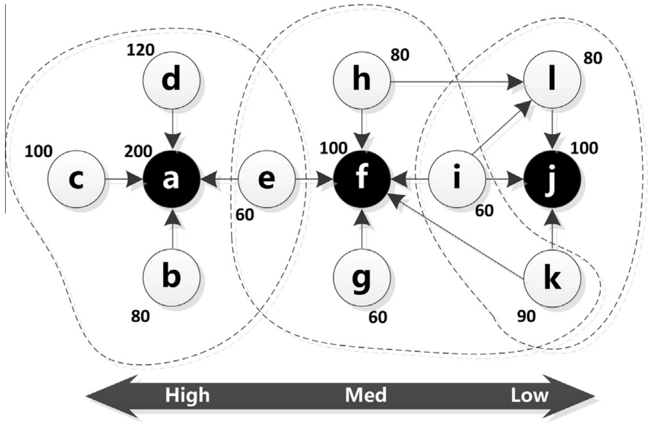


Fig. 2. Example of sparsified affinity graph.

Algorithm II: Sparsifying graph

```
Input:
$G = \{O, E, W\}$ → The affinity graph
Output:
$G' = \{O, E'\}$ → The sparsified affinity graph

$E' = \emptyset$
for $\tau$=0.5 to 1 do
  for i = 1 to $n$ do
    for j = 1 to $n$ do
      if $w_{ij} > \tau$ then
        $e'_{ij}$ = 1
        push $(E', e'_{ij})$
      end if
    end for
  end for
  $G' = (O, E')$
  if $fit(G') > 0.8$ then
    break;
  end if
end if
```

3.4. Identifying communities

As mentioned above, the key task of our study is to identify the top-k important communities, i.e., the k most influential websites and those websites having close relationships with these influential websites. In a graph, the influence of a node is determined based on its position in the structure of the graph.

A straightforward measure of influence may be the degree of a node. The idea behind this simplest approach is that the more neighbors a node has the more influence it may exert. Hence, the problem can be solved in a simple way of selecting the top-*k* nodes which are sorted by their in-degree, the number of edges that point to them. However, as the node degree is an intrinsically local measure, it cannot fully account for the global influence of a node. This shortcoming has motivated the development of more complex measures, such as betweenness centrality [30], closeness centrality [31], HITS [11], and PageRank[32]. Although these improved methods can give better results than simple degree measurement, they are not efficient due to their high computational complexity. Therefore, the design of an effective method to identify influential nodes is still an open issue.

It is not clear how to use the pairwise affinity information to accurately obtain the relative influence of each node on the whole graph. To this end, we propose an influence measurement as a relative *influence score* assigned to each node. Essentially, we assume that the influence score of a node in our domain depends on both the quantity and the dedication of the nodes which have close affinity relationships with it. Specifically, our algorithm makes the following assumptions on measuring the influence score of each node:

1. The influence score of a node depends on the number of nodes which have close affinity relationship with it.
2. The influence score of a node depends on how dedicated nodes having close affinity relationship with it are.

Given a sparsified affinity graph $G' = (O, E')$, we define the *affinity rate* from node o_j to o_i as follows:

$$affr(o_j, o_i) = \begin{cases} \frac{\mu(o_j)}{\sum_{k=1}^{n} e'_{jk}}, & e'_{ji} = 1 \\ 0, & e'_{ji} = 0 \end{cases} \tag{10}$$

where e'_{jk} and e'_{ji} are the edges of the sparsified affinity graph generated according to Rule II, and $\mu(o_j)$ is the number of users accessing website o_j. The affinity rate can be viewed as the dedication of node o_j for node o_i multiplied by the weight of node o_j. It represents the influence of o_i on o_j that is reflected by o_j's dedication and weight. Therefore, the affinity rate function turns the unweighted affinity from o_j to o_i to a weighted influence strength value. Subsequently, the *influence score* of node o_i can be computed by:

$$ins(o_i) = \sum_{j=1}^{n} affr(o_j, o_i) \tag{11}$$

Note that the sum operation in Eq. (11) corresponds to Assumption 1 and each $affr(o_j, o_i)$ corresponds to Assumption 2. The influence score measure is more effective to identify influential nodes than degree measure as it utilizes more information of each node's neighbors, while it has much lower computational complexity than other complex measures.

Algorithm 3 shows the process of computing the influence score of each node and selecting top-*k* website communities. In Algorithm 3, codes from line 3 to 8 calculate the number of edges pointing out from each node. Codes from line 9 to 17 compute the influence score of each node following Eqs. (10) and (11). Then, *k* nodes with the *k* biggest influence scores are inserted into *R*, the set of seed nodes, by codes from line 18 to 25. At last, a seed node and all nodes having an edge pointing from this seed node in *R* are grouped together as a community by codes from line 26 to 34.

If we apply Algorithm 3 on the sparsified affinity graph shown in Fig. 2, we can acquire the top-*3* influential nodes: node *a* with high influence score 330, node *f* with medium influence score 195, and node *j* with low influence score 145, upon which the three identified communities are shown in dashed circles in Fig. 2.

Algorithm III: Identifying communities

```
Input:
$G' = \{O, E'\}$ → The sparsified affinity graph
$k$ → Number of communities to be identified
Output:
$C = \{C_1, C_2, \ldots, C_k\}$ → A set of website communities

1:  $R = \emptyset$
2:  $C = \emptyset$
3:  for j = 1 to $n$ do
4:    $sum(o_j) = 0$
5:    for k = 1 to $n$ do
6:      $sum(o_j)$ += $e'_{jk}$
7:    end for
8:  end for
9:  for i = 1 to $n$ do
10:   $ins(o_i) = 0$
11:   for j = 1 to $n$ do
12:     if $e'_{ji} > 0$ then
13:       $affr(o_j, o_i) = \frac{\mu(o_j)}{sum(o_j)}$
14:       $ins(o_i)$ += $affr(o_j, o_i)$
15:     end if
16:   end for
17: end for
18: for i = 1 to $n$ do
19:   if size $(R) < k$ or $ins(o_i) > minIns(R)$ then
20:     append $(o_i, R)$
21:     if size $(R) > k$ then
22:       remove (elementWithMinIns $(R)$,$R$)
23:     end if
24:   end if
25: end for
26: for i = 1 to $k$ do
27:   $o_m$=pop $(R)$
28:   push $(C_i, o_m)$
29:   for j = 1 to $n$ do
30:     if $e'_{jm} > 0$ then
31:       push $(C_i, o_j)$
32:     end if
33:   end for
34: end for
```

4. Experimental evaluation

4.1. Data set

To evaluate our method, we have conducted an experiment based on the data collected by high performance network traffic monitors placed in the core network of a leading mobile operator in China. The conceptual diagram of the network structure is shown in Fig. 3. The network serving both 2G and 3G subscribers consists of three major parts: (1) various mobile clients including phone, pad, laptop, etc., (2) the access network composed of BTS/BSC (2G) and node-B/RNC (3G), and (3) the core network with SGSN (Serving GPRS Support Node) and GGSN (Gateway GPRS Support Node). A mobile device communicates with a cell tower in the access network which forwards its data service traffic to a SGSN. The SGSN establishes a tunnel on the Gn interface with a GGSN that provides connectivity to external networks. Through this path, the requesting message of a mobile client enters the IP network and reaches the serving server. Data responded from the server to the client traverse in the reversed path.

The high performance traffic monitor system (TMS) devices are placed on the trunks of the Gn interface between SGSN and GGSN to capture all data service traffic. The data cover the activities of 3 million users over 5 days in November 2012 and 2.2 billion HTTP traffic records. For each day, more than 65,000 websites were accessed. Data are saved in a log database and periodically uploaded by an Uploader component to the central massive file system based on Hadoop Distributed File System (HDFS) [33]. Given the sensitivity of the data, privacy related information is masked by the Uploader during the transmission. Each record in the data set is indexed by a hashed subscriber identity and a time stamp. Other columns of each record comprise a summary report of a HTTP request-response transaction including host name, URL, server IP, the total number of packets and bytes of the transaction.

4.2. Result analysis

We carried out the proposed three-step algorithm on top 1,000 important websites, i.e., the 1,000 websites with the most numbers of accessing users. As detailed above, an affinity graph is constructed by Algorithm 1. Each website is represented as a node and has a directed edge to another node if they share some users. Each edge is assigned a weight calculated according to Eq. (1).

Key metrics shown in Table 3 indicate the dense property of the original affinity graph. The average value 659.9 and maximum value 998 of d^{in} imply that there are edges between most of nodes in the graph. Since the original affinity graph is too dense for us to recognize the structure, we obtain the sparsified affinity graph with threshold $\tau = 0.1$ as an example to illustrate the density, which is shown in Fig. 4(a).

To measure how a sparsified affinity graph transformed by Algorithm 2 satisfies the scale-free criterion, we plot the fitting index versus varied threshold τ as shown in Fig. 5. In our experiment, the fitting index of scale-free criterion reaches 0.815 when τ is 0.69. After choosing this optimal threshold based on the scale-free topology criterion rule (Rule III in Section 3), the affinity graph is sparsified into the graph shown in Fig. 4(b). Key metrics of the sparsified affinity graph shown in Table 3 indicate that $\tau = 0.69$ leads to a reasonable trade-off between sparsifying affinity graph and maintaining a reasonable number of edges.

Prior to using the sparsified affinity graph to identify communities, we design and conduct an experiment to evaluate the effectiveness of the scale-free topology based sparsifying operation. In the experiment, a synthetic graph with n nodes following the scale-free structure is generated as the seed graph by a build-in function of the R *igraph* package. Each edge in the seed graph is assigned a random weight ranging from τ_0 to 1.0. Then, $n * (n-1)/4$ edges, each formed by randomly connecting two nodes, are added into the seed graph to produce a reference graph. The effectiveness of the sparsifying method can be measured by the nearness between the identified threshold τ and τ_0, and the $F1$ score, which is produced by comparing edges in the sparsified reference graph with edges in the seed graph. Based on this design, we conduct a number of experiments with n varied from 100 to 600 and a value of τ_0 0.69. For each n, we generate three reference graphs to be sparsified. The average identified τ and $F1$ score are shown in Table 4. Note that the result becomes better when the number of nodes increases. When the number of nodes exceeds 600, which is smaller than the number of websites 1000, we get an accurate sparsified graph which is the same as the seed graph.

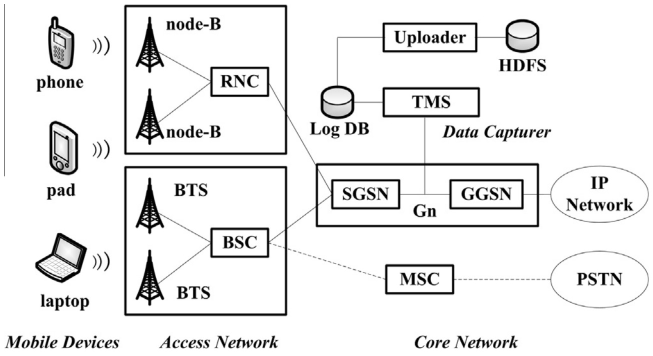


Fig. 3. Network architecture.

After evaluating the effectiveness of the sparsifying method, we calculate the influence score of each node and identify the top-9 communities as shown in Fig. 6. In each community, the central node (red circle) is surrounded by associated nodes (blue circles) that have close relationships with it. The influence score of each central node is displayed in the caption of each sub-figure. Based

Table 3
Metrics of graphs.

Metric	Affinity graph	Sparsified affinity graph
E	1,319,704	902
$avg(d^{in})$	659.9	0.5
$max(d^{in})$	998	46
$\sigma(d^{in})$	225.2	2.7

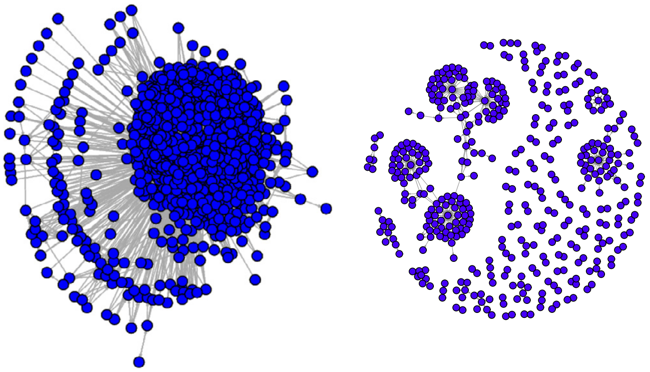

(a) τ = 0.1 (b) τ = 0.69

Fig. 4. Sparsified affinity graphs with different τ.

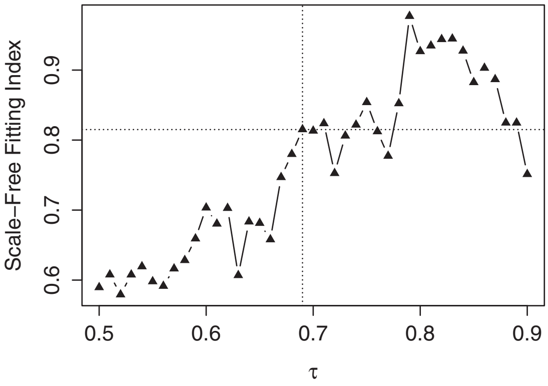


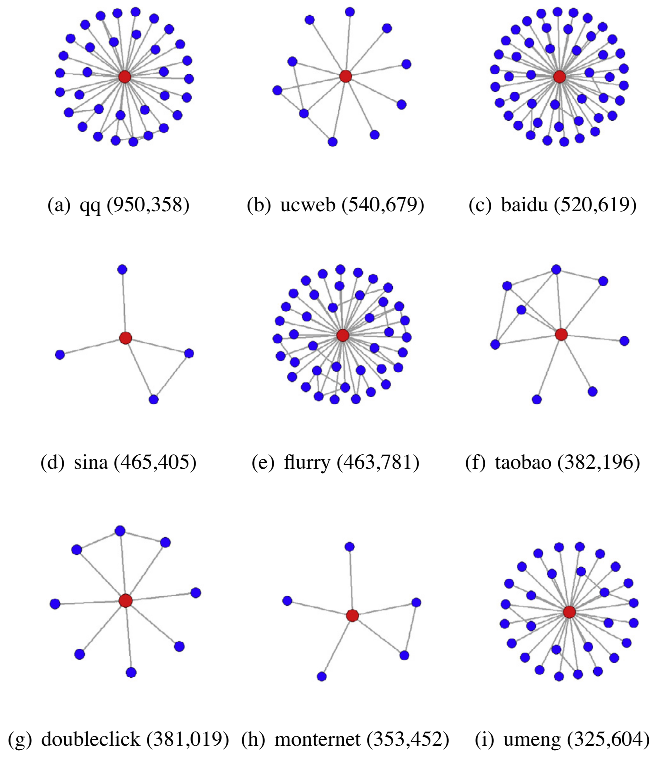
Fig. 5. Scale-free fitting index.

Table 4
F1 score of scale-free method evaluation.

Node	100	200	300	400	500	600
$avg(\tau)$	0.669	0.676	0.682	0.687	0.689	0.690
$avg(F1)$	0.72	0.70	0.72	0.87	0.95	1.00

on human perception, we investigate the real-world relationships between the websites in the same communities. It is not surprising that all central websites are dominant service providers in their domains. Moreover, we observe a number of interesting phenomenons that are not revealed before. The results have substantiated the effectiveness of our method.

The first interesting observation is the identification of some important relationships between websites beyond the link and content relevance. The central node of top 1 community is 'qq.com', the biggest internet service provider in China. The website 'qq.com' is surrounded by a set of websites which possess strong affinities to 'qq' including: (1) 'tencent.com', the official website of the company Tencent which owns 'qq.com'; (2) 'tenpay.com', the payment service portal serving users of 'qq.com' which is developed by Tencent; (3) 'pengyou.com', a social network based on the virtual social relationships in 'qq.com'; (4) 'soso.com', a search engine provided by Tencent; (5) 'qqmail.com', the mail service website for users of 'qq.com'. We can also find close relationships in reality between other websites with 'qq.com' in this community. The 'qq.com' community reveals a group of websites with strong relationships which are owned by the same company Tencent. In addition to this kind of clustering structure under the umbrella of a company, members of community 'flurry.com' in Fig. 6(e) show an ecosystem style structure. 'flurry.com' provides the traffic analysis service for mobile applications. More than 100,000 companies place the analytic tools, which access 'flurry.com', in the mobile applications to accumulate data for user behavior analysis. We find that there are mainly two types of websites in this community: (1) mobile game related websites, such as 'rovio.com' and 'angrybirds.com' for game 'Angry Birds', 'zeptolab.com' for game 'Cut the Rope', and a number of game development studio producing popular games in China like 'appget.cn', 'feidee.com', 'catcap.cn', and (2) advertisement service provider websites, such as 'appads.com', 'adtilt.com', 'immob.cn', *etc.*


(a) qq (950,358) (b) ucweb (540,679) (c) baidu (520,619)

(d) sina (465,405) (e) flurry (463,781) (f) taobao (382,196)

(g) doubleclick (381,019) (h) monternet (353,452) (i) umeng (325,604)

Fig. 6. Top-9 identified communities.

Another interesting phenomenon in the results is that some nodes with small in-degree have higher influence than nodes with large in-degree. Key metrics used to compute the influence score of top-5 websites are shown in Table 5. We choose two well-known websites as examples, 'qq.com' and 'baidu.com'. Although the value of in-degree d_i^{in} of 'baidu.com' is bigger than 'qq.com', the websites influenced by 'qq.com' have more users (bigger $avg(\mu(o_j))$) and are more concentrated (smaller $avg(\sum_{k=1}^{n} e\prime_{jk})$) than the websites influenced by 'baidu.com'. That is why the influence score $ins(o_i)$ of 'qq.com' is bigger than 'baidu.com'.

At last, as mentioned in Section 2, our algorithm should support identifying overlapping communities, i.e., a website may belong to multiple communities. Our algorithm allows a website to join more than one group from different perspectives. A number of websites, which produce popular mobile applications using Flurry statistic API and running on Apple IOS, reside in both communities surrounding 'flurry.com' and 'apple.com', such as 'appget.cn', 'appdriver.cn', and 'catcap.cn'. This kind of identification results prove that our algorithm support identification of overlapping community very well.

In the second step (sparsifying graph) of our method, the original dense weighted affinity graph is sparsified into an unweighted graph. The reason of using the unweighted graph is to reduce the computational complexity of subsequent community identification task. However, the transformation from a weighted graph to an unweighted graph may affect the accuracy of identification. To evaluate this, we conduct an experiment of comparing the difference between identification results from weighted and unweighted sparsified affinity graph. For the weighted affinity graph $G' = (O, E', W)$, we define the *affinity rate* from node o_j to o_i as follows:

$$affr'(o_j, o_i) = \begin{cases} \frac{\mu(o_j)*w_{ji}}{\sum_{k=1}^{n} w_{jk}}, & e'_{ji} = 1 \\ 0, & e'_{ji} = 0 \end{cases} \quad (12)$$

Table 6 shows the identified top-5 influential nodes by the weighted approach. As compared with Table 5, we can see that the nodes and ranks are the same. The top-10 influential nodes remain the same in both approaches. The results show that there is no loss of accuracy in adopting of an unweighted sparsified affinity graph.

To demonstrate how our method is distinguished from previous works, we conduct a comparison experiment with content-based website community identification method. We develop a web crawler to retrieve the title, keywords and description words of homepages of the 94 websites in our top-9 identified communities. The crawling results did not turn out as expected. Forty-six websites, only supporting web services API call, do not return any words to the web crawler, implying that these websites cannot be processed by content-based community identification method.

Table 5
Metrics of top-5 influential nodes.

Metric	qq	ucweb	baidu	sina	flurry
d_i^{in}	33	10	46	4	40
$avg(\mu(o_j))$	58,524	44,224	8,679	80,587	5,099
$avg(\sum_{k=1}^{n} e'_{jk})$	1.4	1.6	2.5	1.4	1.3
$ins(o_i)$	950,358	540,679	520,620	465,405	463,781

Table 6
Metrics of top-5 influential nodes by the weighted approach.

Metric	qq	ucweb	baidu	sina	flurry
$avg\left(\frac{\sum_{k=1}^{n} w_{jk}}{w_{ji}}\right)$	1.19	1.17	1.16	1.14	1.36
$ins(o_i)$	1,921,687	430,627	383,422	305,687	176,882

Table 7
Similarities between websites by two methods.

Website	Affinity-based		Content-based	
	paipai	haodf	paipai	haodf
qq	0.775	0.574	0.573	0.722
baidu	0.632	0.732	0.613	0.607

This is one of the major drawbacks of content-based approach method. For the remaining 48 websites that returned contents, we perform the algorithm in [19] to cluster them to 4 communities. The identified communities, based on human perception, do not provide meaningful information. In Table 7, we illustrate the calculated similarities among the four websites by our method and content-based method, respectively. In our method, 'qq.com' and 'paipai.com', which are owned by the same company and have close relationship, are clearly identified in the same community. 'baidu.com', the leading search engine in China, and 'haodf.com', a well-known health portal mainly obtaining users and clicks from 'baidu.com', are also identified in the same community by our method. However, for the content-based method, the community relationships among them are rather incoherent. We observe that two pairs of websites, 'qq.com' and 'haodf.com', 'baidu.com' and 'paipai.com', are identified in the same community, respectively, but websites in these two pairs do not have any close relationship except a few of common words. The results demonstrate that the quality of identification of our method is much better than that of content-based method.

5. Conclusion

Understanding the website community structure in the web environment is vital for improving service quality, optimizing network resources and devising business strategies. In this paper, we tackle the problem of mining website communities in a mobile internet by our proposed three-step algorithm. The first step is to construct an affinity graph by taking into account of the shared user information between websites. The second step is designed to transform the dense affinity graph into a sparse one to reduce the computational workload and improve the quality of community detection. The last step is performed to select the top-k influential websites and associated websites with strong affinity relationships as the identified communities. Evaluation on the data from a large real-word cellular data network has substantiated the effectiveness of our proposed algorithm in identifying significant website communities. As compared with existing content-based website community identification solutions, our work can reveal the hidden relationships between websites in reality from the user behavior perspective, that cannot be detected by only considering hyperlink structure and content similarity.

References

[1] Cisco Visual Networking Index: Forecast and Methodoloy, 2012-2107; <http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf>.

[2] Sandvine Global Internet Phenomena Report, 2H 2012; <http://www.sandvine.com/downloads/documents/Phenomena_2H_2012/Sandvine_Global_Internet_Phenomena_Report_2H_2012.pdf>.

[3] T. Han, N. Ansari, M. Wu, H. Yu, On accelerating content delivery in mobile networks, IEEE Commun. Surv. Tutor. 3 (3) (2013) 1314–1333.

[4] Y. Zhang, N. Ansari, M. Wu, H. Yu, On wide area network optimization, IEEE Commun. Surv. Tutor. 14 (4) (2012) 1090–1113.

[5] G. Cheng, N. Ansari, S. Papavassiliou, Adaptive QoS provisioning by pricing incentive QoS routing for next generation networks, Comput. Commun. 31 (10) (2008) 2308–2318.

[6] M. Kobayashi, H. Nakayama, N. Ansari, N. Kato, Reliable application layer multicast over combined wired and wireless networks, IEEE Trans. Multimedia 11 (8) (2009) 1466–1477.

[7] J. Zhang, N. Ansari, On assuring end-to-end QoE in next generation networks: challenges and a possible solution, IEEE Commun. Mag. 49 (7) (2011) 185–192.

[8] S.N. Dorogovtsev, J.F. Mendes, Evolution of Networks: From Biological Nets to the Internet and WWW, Oxford University Press, 2003.

[9] C. Chen, Structuring and visualising the www by generalised similarity analysis, in: Proceedings of the Eighth ACM Conference on Hypertext, 1997, pp. 177–186.

[10] S. Mukherjea, Y. Hara, Focus+ context views of World-Wide Web nodes, in: Proceedings of the Eighth ACM Conference on Hypertext, 1997, pp. 187–196.

[11] J.M. Kleinberg, Authoritative sources in a hyperlinked environment, in: Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, 1998, pp. 668–677.

[12] E. Spertus, ParaSite: mining structural information on the web, Comput. Networks ISDN Syst. 29 (8) (1997) 1205–1215.

[13] F. Menczer, Links tell us about lexical and semantic web content, 2001. arXiv:cs/0108004.

[14] Y. Wang, M. Kitsuregawa, On combining link and contents information for web page clustering, Database Expert Systems Applications, Springer, Berlin, Heidelberg, 2002. pp. 902–913.

[15] D. Gibson, J. Kleinberg, P. Raghavan, Inferring web communities from link topology, in: Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia, 1998, pp. 225–234.

[16] G.W. Flake, S. Lawrence, C.L. Giles, Efficient identification of web communities, in: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000, pp. 150–160.

[17] J.J. Merelo-Guervs, B. Prieto, A. Prieto, G. Romero, P.C. Valdivieso, Clustering web-based communities using self-organizing maps. In: IADIS International Conference Web Based Communities, 2004.

[18] M. Ester, H.P. Kriegel, M. Schubert, Web site mining: a new way to spot competitors, customers and suppliers in the world wide web, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 249–258.

[19] F. Ricca, P. Tonella, C. Girardi, E. Pianta, An empirical study on keyword-based web site clustering, in: Proceedings of the 12th IEEE International Workshop on Program Comprehension, 2004, pp. 204–213.

[20] P. Tonella, F. Ricca, E. Pianta, C. Girardi, Using keyword extraction for web site clustering. in: Proceedings of the Fifth IEEE International Workshop on Web Site, Evolution, 2003, pp. 41–48.

[21] H.P. Kriegel, M. Schubert, Classification of websites as sets of feature vectors, Databases Appl. (2004) 127–132.

[22] E. Meneses, Vectors and graphs: two representations to cluster web sites using hyperstructure, in: IEEE Web Congress, LA-Web'06. Fourth Latin American, 2006, pp. 172–178.

[23] J. Kangasharju, J. Roberts, K.W. Ross, Object replication strategies in content distribution networks, Comput. Commun. 25 (4) (2002) 376–383.

[24] P.N. Tan, Introduction to Data Mining, Pearson Education India, 2007.

[25] V. Kumar, An introduction to cluster analysis for data mining, Technical report, University of Minnesota, USA CS Dept., 2000.

[26] B. Zhang, S. Horvath, A general framework for weighted gene co-expression network analysis, Stat. Appl. Genet. Mol. Biol. 4 (1) (2005) 1128.

[27] A. Broder, R. Kumar, F.P. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, J. Wiener, Graph structure in the web, Comput. networks 33 (1) (2000) 309–320.

[28] M. Faloutsos, P. Faloutsos, C. Faloutsos, On power-law relationships of the internet topology, ACM SIGCOMM Comput. Commun. Rev. 29 (4) (1999) 251–262.

[29] C. Pinto, L.A. Mendes, J.A. Machado, A review of power laws in real life phenomena, Commun. Nonlinear Sci. Numer. Simul. 17 (9) (2012) 3558–3578.

[30] L.C. Freeman, A set of measures of centrality based on betweenness, Sociometry (1977) 35–41.

[31] M. Newman, Networks: An Introduction, Oxford University Press, 2009.

[32] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking: bringing order to the web, Technical report, Computer Science Department, Stanford University, 1999.

[33] Apache open source project hadoop, URL <http://hadoop.apache.org/>.