Journal of Engineering and Applied Sciences 6 (2): 135-140, 2011
ISSN: 1816-949X

An Enhanced Mechanism for Profiling and Searching the Internet Endpoints by Clustering the Endpoints Using Fuzzy C-Means Algorithm

S. Kannan, T. Kalaikumaran, S. Karthik and V.P. Arunachalam
Department of Computer Science and Engineering, SNS College of Technology,
Sathy Main Road, 641035 Coimbatore, Tamil Nadu, India

Abstract: Understanding and using the internet in worldwide is a challenging problem that is typically addressed by analyzing network traces. However, obtaining such traces presents its own set of challenges owing to either privacy concerns or to other operational difficulties. The key hypothesis of the research here is that most of the information needed to profile, the internet endpoints is already available around us on the web. We implement and deploy a Google-based profiling tool which accurately characterizes endpoint behaviour by collecting and strategically combining information freely available on the web. Unconstrained endpoint profiling approach is used to profile and classify the endpoints. The websites are classified and clustered based on the search hits which contain the hit text and URL. On querying, it matches the domain name and URL if it does not match then it verifies the key words. The key words in the web cache are clustered using Fuzzy C-means algorithm which enhances the speed of the search engine.

Key words: Clustering, Unconstrained Endpoint Profiling (UEP), search hits, network, internnet, India

INTRODUCTION

In this study, we focus on web-oriented UEP approach that aims to characterize endpoint behaviour by strategically combining information from a number of different sources Google search available on the web. The key idea is to query the engine with IP addresses corresponding to arbitrary endpoints. In particular, we search on text strings corresponding to the standard dotted decimal representation of IP addresses and then characterize endpoints by extracting information from. The core components of the the responses returned by methodology are:

- A rule generator that operates on top of the Google search engine
- An IP tagger that tags endpoints with appropriate features based solely on information collected on the web

The key challenge, lies in automatically and accurately distilling valuable information from the web and creating a semantically-rich endpoint database. The rule generator returns the search hits, containing URL and hit text. We use the clustering algorithm to cluster the key words (Cheeseman and Stutz, 1996). Cluster analysis or clustering is the assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar in some sense. Clustering can be considered the most important unsupervised learning problem so as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data. A cluster is therefore, a collection of objects which are similar between them and are dissimilar to the objects belonging to other clusters. We can divide the cluster based on two criterions:

Distance clustering: Two or more objects belong to the same cluster if they are close according to a given distance (in this case geometrical distance). This is called distance-based clustering.

Conceptual clustering: Two or more objects belong to the same cluster if this one defines a concept common to all that objects (Chen and Trajkovic, 2004).

Goals of clustering: The major goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. But how to decide what constitutes a good clustering. It can be shown that there is no absolute best criterion which would be independent of the final aim of the clustering. Consequently, it is the user which must supply this criterion in such a way that the result of the clustering will suit their needs. For instance, we could be interested

Corresponding Author: S. Kannan, Department of Computer Science and Engineering, SNS College of Technology,
Sathy Main Road, 641035 Coimbatore, Tamil Nadu, India

in finding representatives for homogeneous groups (data reduction) in finding natural clusters and describe their unknown properties (natural data types) in finding useful and suitable groupings (useful data classes) or in finding unusual data objects (outlier detection) (Ellacoya Networks, 2007).

APPLICATIONS OF CLUSTERING

Clustering algorithms can be applied in many fields for instance:

Marketing: Finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records.

Biology: Classification of plants and animals given their features.

Libraries: Book ordering.

Insurance: Identifying groups of motor insurance policy holders with a high average claim cost; identifying frauds.

City-planning: Identifying groups of houses according to their house type, value and geographical location.

Earthquake studies: Clustering observed earthquake epicenters to identify dangerous zones.

WWW: Document classification; clustering weblog data to discover groups of similar access patterns.

Search result grouping: In the process of intelligent grouping of the files and websites, clustering may be used to create a more relevant set of search results compared to normal search engines like Google. There are currently a number of web based clustering tools such as clusty (Jeffrey *et al*., 2006).

Problems with clustering: There are a number of problems with clustering. Among them:

- Current clustering techniques do not address all the requirements adequately (and concurrently)
- Dealing with large number of dimensions and large number of data items can be problematic because of time complexity
- The effectiveness of the method depends on the definition of distance (for distance-based clustering)
- If an obvious distance measure does not exist we must define it which is not always easy, especially in multi-dimensional spaces
- The result of the clustering algorithm (that in many cases can be arbitrary itself) can be interpreted in different ways (Karagiannis *et al*., 2005)

CLUSTERING ALGORITHMS

Clustering algorithms may be classified as below:

Density-based methods: It introduces the notion of uncertainty based on density while it handles efficiently arbitrarily shaped clusters. Both the clustering criterion and the membership function are based on the density distribution of the data.

Hierarchical methods: This algorithm finds successive clusters using previously established clusters. These algorithms usually are either agglomerative (bottom-up) or divisive (top-down). Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters (Karagiannis *et al*., 2007).

Partitioning methods: This algorithm typically determines all clusters at once but can also be used as divisive algorithms in the hierarchical clustering (Kim *et al*., 2008). The main requirements that a clustering algorithm should satisfy are:

- Scalability
- Dealing with different types of attributes
- Discovering clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Ability to deal with noise and outliers
- Insensitivity to order of input records
- High dimensionality
- Interpretability and usability

We mainly use the Partition method of clustering which includes the following algorithms (Liang *et al*., 2005). K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and

an early group age is done. At this point, we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop, we may notice that the k centroids change their location step by step until no more changes are done.

In other words, centroids do not move any more. K-means is a simple algorithm that has been adapted to many problem domains. As we are going to see, it is a good candidate for extension to research with fuzzy feature vectors. The main disadvantage of K-means is that it is applicable only when mean is defined then what about categorical data and need to specify k, the number of clusters in advance (Madhyastha *et al*., 2006).

Automatic classification or clustering auto class: Solves the problem of automatic discovery of classes in data (sometimes called clustering or unsupervised learning) as distinct from the generation of class descriptions from labelled examples (called supervised learning). It aims to discover the natural classes in the data. Auto class is applicable to observations of things that can be described by a set of attributes without referring to other things. The data values corresponding to each attribute are limited to be either numbers or the elements of a fixed set of symbols. With numeric data, a measurement error must be provided (Mai *et al*., 2006).

Fuzzy C-means: FCM is a method of clustering which allows one piece of data to belong to two or more clusters (McDaniel *et al*., 2006).

Related work: Recent studies have addressed the key words for each domain are clustered using auto class algorithm. The main disadvantage of auto class algorithm is that each data item must belong to one and only one cluster. Also, we need to traverse through many clusters to find a single search. This increases the searching time of the search engine.

Hence in the proposed system, we cluster all the related websites for a single key word. Fuzzy C-means algorithm is used for clustering the websites since each site can belong to >1 cluster (Mislove *et al*., 2007; Plissonneau *et al*., 2005; Rexford *et al*., 2002).

Web-based endpoint profiling tool: Figure 1 shows the web-based endpoint profiling tool which gives the frame work for the survey. At the functional level, the goal is straight forward: we query Google search engine by searching on text strings corresponding to the standard dotted decimal representation of IP addresses. For a given

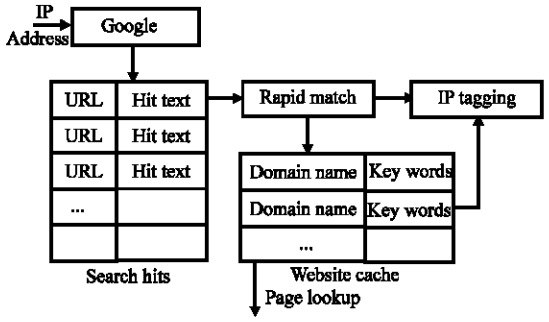

Fig. 1: Existing framework

input in the form of an IP address, e.g., 200.101.18.182 we collect search hits returned by Google and then extract information about the corresponding endpoint. The output is a set of tags (features) associated with this IP address. For example, forum user, game abuser, mail server, etc. In general, an endpoint could be tagged by a number of features, e.g., a forum user and a p2p client. Such information can come from a number of different URLs. The profiling methodology involves the following three modules:

- Rule generation
- Web classification
- IP tagging

We present a new framework which involved in web classification phase of the proposed work (Spring *et al*., 2002).

PROPOSED WORK

We construct clusters for each key word containing all the related websites. For clustering, we use the Fuzzy C-means algorithm which allows a single site to be present in >1 cluster. This reduces the searching speed and also the memory in the web cache is reduced. This process can be achieved by the following 3 phases:

Rule generation: The process starts by querying and then obtaining the set of search hits. Each search hit consists of a URL and corresponding hit text, i.e., the text surrounding the word searched. By meaningfully used, we mean that the key word found implies an application or application class associated with network activity. We construct a set of rules that map key words to an interpretation for the functioning of that website, i.e., the website class.

For example, the rules we develop in this step capture the intelligence that presence of one of the following key

words; counter strike, world of war craft, age of empires or game abuser in either the URL or the text of a website implies that it is a gaming website (either gaming server list or abuse list). For instance if a website only contains the key word mail server from the set of key words then it is classified as a site containing list of mail servers. However, if a website contains one of the following words, spam or dictionary attacker besides mail server then it is classified as one containing list of malicious mail servers, e.g., one that is known to originate spam. Similar rules are used to differentiate between websites providing gaming servers list and gaming abuse list. (Trestian *et al*., 2008).

Web classifier: Extracting information about endpoints from the web is a nontrivial problem. The approach is to first characterize a given webpage (returned by Google), i.e., determine what information is contained on a website. This approach significantly simplifies the endpoint tagging procedure (Verkaik *et al*., 2006).

Rapid URL search: Some websites can be quickly classified by the key words present in their domain name itself. Hence after obtaining a search hit we first scan the URL string to identify the presence of one of the key words from the key word set in the URL. If the rapid URL search succeeds, we proceed to the IP tagging phase. If rapid match fails, we initiate a more thorough search in the hit text as we explain next.

Hit text search: To facilitate efficient webpage characterization and endpoint tagging, we build a website cache (Fig. 2). The key idea is to speed-up the classification of endpoints coming from the same key word.

The procedure is as follows. If we get a match in the website cache (for the specific URL, we are currently trying to match), we check if any of the key words associated with that domain match in the search hit text and then it finds out the related websites of that cluster. In rapid match, we map the key words entered in the search engine with the key words in the web cache. From the matched key word, all the relevant websites belong to that cluster will be tagged and displayed (Verkaik *et al*., 2006; Zander *et al*., 2005).

IP tagging: The final step is to tag an IP address, based on the collected information. We distinguish between three different scenarios, URL based tagging. In some scenarios, an IP address can be directly tagged when the

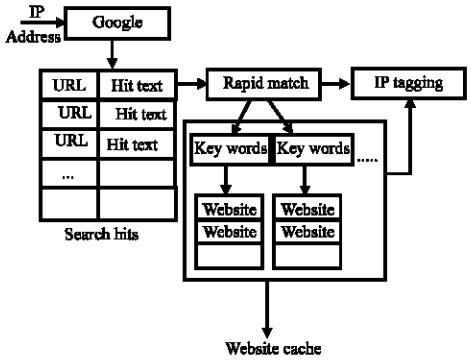

Fig. 2: Proposed framework

URL can be classified via rapid search for key words in the URL itself. One example is classifying eMule p2p servers based on the emule-project.net domain name. In majority of the cases such rapid tagging is not possible and hence we have to examine the hit text for additional information.

General hit text based tagging: For most of the websites, we are able to accurately tag endpoints using a key word based approach. The procedure is as follows. If we get a match in the website cache (for the specific URL we are currently trying to match) we check if any of the key words associated with that domain match in the search hit text (Verkaik *et al*., 2006).

Hit text based tagging for forums: The key word based approach fails when a URL maps to an Internet forum site. This is because a number of non correlated key words may appear at a forum page. Likewise, forums are specific because an IP address can appear at such a site for different reasons. Either, it has been automatically recorded by a forum post or because a forum user deliberately posted a link (containing the given IP address) for various reasons. In the case of forums, we proceed as follows.

First, we use a postdate and user name in the vicinity of the IP address to determine if the IP address was logged automatically by a forum post. Hence, we tag it as the forum user.

If this is not the case, the presence of the following key words: http: \, ftp: \, pp- stream:\, mms:\, etc., in front of the IP address string in the hit text suggests that the user deliberately posted a link to a shared resource on the forum.

Consequently, we tag the IP address as an http or ftp or as a streaming sup-porting a given protocol (ppstream, mms, tvants, sop, etc.). Because each IP address

generates several search hits, multiple tags can be generated for an IP address (Zander *et al*., 2005).

CONCLUSION

In this study, we proposed a novel approach to the endpoint profiling problem. The key idea is to shift the research focus from mining operational network traces to extracting the information about endpoints from elsewhere, e.g., web or p2p systems. We developed and deployed a profiling tool that operates on top of the Google search engine.

It is capable of collecting, automatically processing and strategically combining information about endpoints and finally tagging the same with extracted features. We demonstrated that the proposed approach can accurately predict application and protocol usage trends even when no network traces are available, outperform state-of-the-art classification tools such as BLINC when packet traces are available and retain high classification capabilities even when only sampled flow-level traces are available.

We applied the approach to profile endpoints at four different world regions and provided a unique and comprehensive set of insights about network applications and protocols used in these regions, characteristics of endpoint classes that share similar access patterns and clients' locality properties.

Finally, we demonstrated that complementary UEP approaches such as p2p-or DNS-based schemes can further improve the web-based UEP performance.

REFERENCES

Cheeseman, P. and J. Stutz, 1996. Bayesian Classification (AutoClass): Theory and Results. In: Advances in Knowledge Discovery and Data Mining, Fayyad, U.M., G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy (Eds.). MIT Press, Cambridge, Massachusetts, pp: 153-180.

Chen, H. and L. Trajkovic, 2004. Trunked radio systems: Traffic prediction based on user clusters. Proceedings of the International Symposium on Wireless Communication Systems, Sept. 20-22, USA., pp: 76-80.

Ellacoya Networks, 2007. Web traffic overtakes P2P as largest bandwidth on the network. http://www.circleid.com/posts/web_traffic_overtakes_p2p_bandwidth/.

Jeffrey, E., M. Arlitt and A. Mahanti, 2006. Traffic classification using clustering algorithms. Proceedings of the ACM SIGCOMM 2006 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, Sept. 11-15, ACM Press, Pisa, Italy, pp: 281-286.

Karagiannis, T., K. Papagiannaki and M. Faloutsos, 2005. BLINC: Multilevel traffic classification in the dark. Proceedings of the Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, Aug. 22-26, ACM Press, Philadelphia, Pennsylvania, USA., pp: 229-240.

Karagiannis, T., K. Papagiannaki, N. Taft and M. Faloutsos, 2007. Profiling the end host. Proceedings of the 8th international conference on Passive and Active Network Measurement, (PANM'07), Springer-Verlag, Berlin, Heidelberg, pp: 186-196.

Kim, H., K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos and K. Lee, 2008. Internet traffic classification demystified: Myths, caveats and the best practices. Proceedings of the ACM CONEXT Conference, Dec. 10-12, Madrid, Spain, pp: 1-12.

Liang, J., R. Kumar, Y. Xi and K.W. Ross, 2005. Pollution in P2P file sharing systems. Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, March 13-17, USA., pp: 1174-1185.

Madhyastha, H.V., T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy and A. Venkataramani, 2006. IPlane: An information plane for distributed services. Proceedings of the 7th Symposium on Operating Systems Design and Implementation, November 2006, Seattle, Washington, pp: 367-380.

Mai, J., C.N. Chuah, A. Sridharan, T. Ye and H. Zang, 2006. Is sampled data sufficient for anomaly detection. Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, Oct. 25-27, Rio de Janeiro, Brazil, pp: 165-176.

McDaniel, P., S. Sen, O. Spatscheck, J. van der Merwe, W. Aiello and C. Kalmanek, 2006. Enterprise security: A community of interest based approach. Proceedings of the NDSS, San Diego, CA, Feb. 2006.

Mislove, A., M. Marcon, K.P. Gummadi, P. Druschel and B. Bhattacharjee, 2007. Measurement and analysis of online social networks. Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, Oct. 23-26, San Diego, CA. USA., pp: 29-42.

Plissonneau, L., J.L. Costeux and P. Brown, 2005. Analysis of peer-to-peer traffic on ADSL. Passive Active Network Measurement, 3431: 69-82.

Rexford, J., J. Wang, Z. Xiao and Y. Zhang, 2002. BGP routing stability of popular destinations. Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment, Nov. 6-8, Marseille, France, pp: 197-202.

Spring, N., R. Mahajan and D. Wetherall, 2002. Measuring ISP topologies with rocketfuel. Proceedings of the ACMSIGCOMM Conference, Aug. 19-23, Pittsburgh, Pennsylvania, USA., pp: 133-145.

Trestian, I., S. Ranjan, A. Kuzmanovi and A. Nucci, 2008. Unconstrained endpoint profiling (Googling the Internet). Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication, Aug. 17-22, Seattle, WA. USA., pp: 279-290.

Verkaik, P., O. Spatscheck, J. van der Merwe and A.C. Snoeren, 2006. PRIMED: Community-of-interest-based DDoS mitigation. Proceedings of the 2006 SIGCOMM Workshop on Large-Scale Attack Defense, Sept. 11-15, Pisa, Italy, pp: 147-154.

Zander, S., T. Nguyen and G. Armitage, 2005. Automated traffic classification and application identification using machine learning. Proceedings of the IEEE Conference on Local Computer Networks, (LCN'05), Sydney, Australia, USA., pp: 250-257.