Bayesian Nonparametric Clustering of Network Traffic Data

Barış Kurt
Department of Computer Engineering
Boğaziçi University
P.K.2, 34342, Istanbul, Turkey
baris.kurt@boun.edu.tr

A. Taylan Cemgil
Department of Computer Engineering
Boğaziçi University
P.K.2, 34342, Istanbul, Turkey
taylan.cemgil@boun.edu.tr

Muhittin Mungan
Department of Physics
Boğaziçi University
North Campus, KB Building, 34342,
Istanbul, Turkey
mmungan@boun.edu.tr

Ece Saygun
Ericsson Turkey
Uso Center NO:61, 34398 Maslak
Istanbul, Turkey
ece.saygun@ericsson.com

Abstract

We introduce a Bayesian nonparametric method for the clustering of network flows, sequences of packets observed during the communication between pairs of hosts. Our goal is to infer the application types without inspecting in detail the content of each packet, avoiding the so called DPI (Deep Packet Inspection). Instead, we only use simple to derive features such as packet size or direction (upload/download) and represent each flow by a sequence of symbols from a finite alphabet. The flows are naturally modeled as a mixture of Markov models, generated by a Dirichlet process. We have implemented a blocked Gibbs sampler for inferring cluster assignments by integrating out the model parameters. The clustering results obtained on data captured from a real network seems to be coherent.

1 Introduction

Network traffic classification is an important and challenging problem. It plays a crucial role in network management, such as quality of service, security, and trend analysis. The goal is to infer the applications that generate a certain group of packets, such as video, peer-to-peer, gaming, email etc. Traditionally, application type could be inferred simply via investigation of the IP-port number, however this approach is no longer effective as there are now many different kinds of network applications, some of which deliberately change their behavior in order not to be detected, such as peer-to-peer protocols. Another difficulty is that due to privacy requirements and computational burden, it is desirable that classification algorithms are allowed to use only partial information present in the network data and avoid deep packet inspection (DPI).

While there are several different approaches to traffic classification [1, 2, 3], in this work, we are interested in *flow-based clustering* [4, 5, 6]. We are going to use flow statistics between two communicating pairs, ignoring everything else such as the payload of the network packages, DNS information, connectivity patterns of hosts, etc. We collected our real world data by monitoring our own network activity. The individual flows labels in the real world data are not known, but we know which flows belong to which application. We also generated synthetic data according to our generative models, for the verification of our method.

We modeled the network flows as Markovian time series and assumed that they are generated by a Dirichlet process mixture model [7]. We implemented a Gibbs sampler, to infer the number of clusters and cluster assignments of the flows in both synthetic and real world data, and observed coherent results.

2 Methodology

A network flow f is a chain of packets $s_{1:T}$ transmitted according to a protocol between a source node and destination node. Each packet has the following properties: arrival time, protocol, up/down flag, and size. Currently, we use only up/down information and size, $s_t = \{up/down, size\}$. Moreover, the size in bytes is quantized into S levels, so each packet s_t is an element of a state space with cardinality $2 \times S$. Each flow is generated by a first order Markov model $p(s_t|s_{t-1};\theta)$ where θ are the model parameters (initial state distribution and transition matrix). The complete data set $F = \{f^1, f^2, \ldots, f^N\}$ is generated by a Dirichlet process mixture.

$$\begin{aligned} f^n|\theta_i &\sim \text{MarkovModel}(\theta_i) \\ \theta_i|G &\sim G \\ G &\sim \text{DirichletProcess}(G_0, \alpha) \end{aligned}$$

G_0 is the prior distribution of Markov model parameters, that we take as a independent Dirichlet for each rows of the transition matrix. At each step, a new flow is generated by either one of the available Markov models, or a new Markov model is introduced with a probability dictated by a Polya-urn scheme. The parameter α is the concentration parameter, which affects the tendency to generate new Markov models. In the process, flows that are generated by the same Markov model forms a cluster. Let's call $c_i = j$ if flow i is in cluster j. The (marginal) likelihood of F conditional on the partitioning c is:

$$\begin{aligned} p(F|c) &= \int p(F|c,\theta)p(\theta)d\theta = \prod_m \int p(F^{[c=m]}|\theta)p(\theta)d\theta \\ F^{[c=m]} &= \{f^n \in F|c_n = m\} \end{aligned}$$

Due to the conjugacy of $p(\theta)$, for each c, this expression can be evaluated in closed form. As the number of flows increase, the inference for the cluster assignments $c = \{c_1, \ldots, c_N\}$ becomes intractable (the number assignments c_i represent different partition of the flows into clusters, and the number of possible partitions grows super exponentially with the total number of flows given by the Bell numbers). We use a collapsed Gibbs sampler for sampling from $p(c|F)$, by integrating out the Markov parameters, with the following full conditionals:

$$p(c_i = j|c_{-i}, F) \propto p(c_{-i}, c_i = j) \int p(F|c_{-i}, c_i = j, \theta)p(\theta)d\theta, \quad (c_i \text{ belongs to cluster j})$$

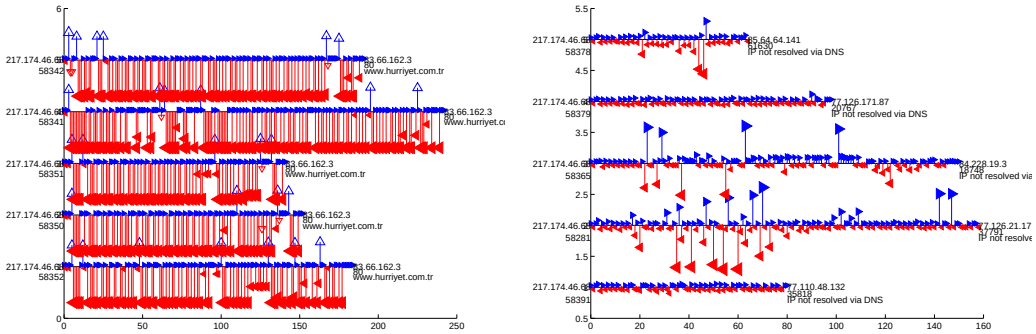
$$p(c_i = c_{new}|c_{-i}, F) \propto p(c_{-i}, c_i = c_{new}) \int p(F|c_{-i}, c_i = c_{new}, \theta)p(\theta)d\theta, \quad (c_i \text{ belongs to a new cluster})$$

where, c_{-i} is the all cluster indicators except c_i.

3 Results

In the experiments done with synthetic data, we observed that the number of clusters formed by the Gibbs sampler are close to the ground truth, and the clusters successfully group together flows, generated from the same Markov models. However we cannot elaborate the success of our method quantitatively on the real world data without the ground truth. But, Figure 1 shows two examples of clusters found in the real world data and it is observed that the clusters group together the flows with similar characteristics.

Figure 2 shows a clustering result on real world data which contains 4 different applications. We did not use DPI ground truth, but we separately collected the flows for each application in order to match the flows with their applications. It's observed that each application contains many types of flows; some of these types are shared, and some are unique for the application.



(a) A cluster of 5 flows with large down packets.

(b) A cluster of 5 flows with small up and down packets.

Figure 1: Visualization of 2 network flow clusters found on the real world data. Each line represents a flow. The x-axis represents the time, blue upward arrows shows up packets, red downward arrows shows down packets. Arrow lengths represent packet sizes. It can be seen that the flows in each cluster have similar properties. In part (a), long red arrows shows that the flows have large down packets. In part (b), sizes of up and down packets are mostly small.

4 Discussions

The clusters formed by our method do not immediately determine an application's type, but we can consider that they form a dictionary which may be helpful in the type inference. As the future work, we plan to develop new models which determine an applications type by investigating the distribution of its flows among the clusters.

Whenever a new type of application is observed, the nonparametric method allows the introduction of new clusters, which means adding new words to the dictionary. This adaptivity is important for the analysis of the network traffic data, since new types of applications emerge frequently.

5 Acknowledgments

This research has been partly supported by Ericsson Telekomnikasyon A.Ş. within the scope of research being conducted in CELTIC project MEVICO.

References

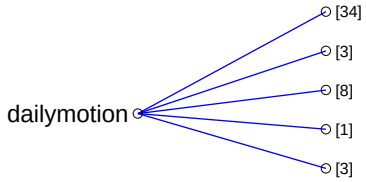
[1] Alberto Dainotti, W. de Donato, and A. Pescapé. TIE: a community-oriented traffic classification platform. *Traffic Monitoring and Analysis*, pages 64–74, 2009.

[2] Marios Iliofotou, Prashanth Pappu, Michalis Faloutsos, Michael Mitzenmacher, Sumeet Singh, and George Varghese. Network traffic analysis using traffic dispersion graphs (TDGs): techniques and hardware implementation. Technical report, 2007.

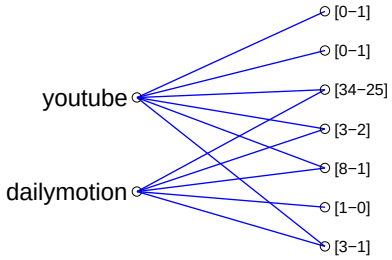
[3] Thomas Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: multilevel traffic classification in the dark. *ACM SIGCOMM Computer Communication Review*, 35(4):229–240, 2005.

[4] Charalampos Rotsos, Jurgen Van Gael, Andrew W. Moore, and Zoubin Ghahramani. Probabilistic graphical models for semi-supervised traffic classification. *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference on ZZZ - IWCMC '10*, page 752, 2010.

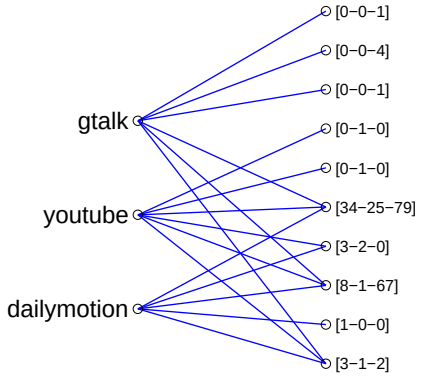
[5] A.W. Moore and Denis Zuev. Internet traffic classification using bayesian analysis techniques. In *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 50–60. ACM, 2005.



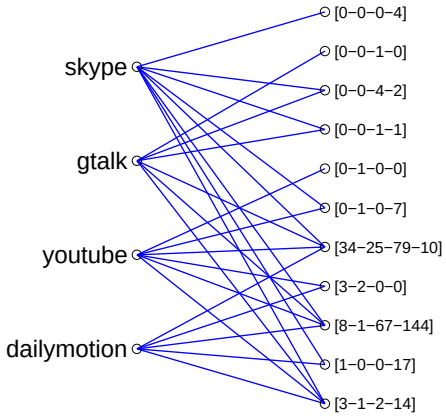
(a) Clusters formed by a video application (dailymotion).



(b) Clusters from two video applications (dailymotion & youtube).



(c) A voip application is introduced (gtalk).



(d) Second voip application is introduced (skype).

Figure 2: Visualization of the clustering process. The left nodes are applications, which is a collection of many flows. The right nodes represent the clusters. The numbers next to the clusters are the number of flows in that cluster. The order of the numbering is the order of the applications. (a) shows a single video application which is composed of 5 types of flows. In (b) two video applications are shown. They share 4 clusters. In (c), a voip application is added. It shares 3 clusters with video applications and forms 3 new clusters. In (d), the second voip application forms one new cluster. It also shares 2 clusters with the voip application, and 5 with video applications.

[6] Alberto Dainotti, Walter de Donato, Antonio Pescape, and Pierluigi Salvo Rossi. Classification of Network Traffic via Packet-Level Hidden Markov Models. *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, pages 1–5, 2008.

[7] Radford M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.