공학석사학위논문

Topic-model based Automatic Signature Extraction of Internet Applications

토픽 모델링을 이용한 자동

인터넷 응용 프로그램 시그니쳐 추출

2013 년 2 월

서울대학교대학원

전기·컴퓨터공학부

윤효진

Topic-model based Automatic Signature Extraction of Internet Applications

지도교수 김종권

이 논문을 공학석사학위논문으로 제출함

2013 년 2 월

서울대학교대학원

전기·컴퓨터공학부

윤효진

윤효진의 석사학위 논문을 인준함

2012 년 12 월

위원장 전화숙(인)

부위원장 김종권(인)

위원 권태경(인)

Abstract

Topic-model based Automatic Signature Extraction of Internet Applications

HyojinYoon
School of Computer Science and Engineering
The Graduate School
Seoul National University

Classifying network traffic according to the application that generated it has attracted significant interests among Internet researchers and operators, as it is an essential task for understanding, operating, optimizing, planning, and financing the Internet. Although content-analysis based Deep Packet (Payload) Inspection technique has been found very accurate once given a set of known payload signature strings for corresponding applications, it is very time consuming and challenging to manually derive and construct the signatures.

In this paper, we propose a new, automatic payload content-analysis based traffic classification method called TASTE(Topic-model based Automatic Signature Extraction). TASTE adopts the Latent Dirichlet Allocation (LDA) topic model, which is one of the most popular probabilistic text modeling techniques for extracting latent semantic information from

text corpa. Our evaluation with a broad range of data sets demonstrates that TASTE can automatically detect and identify signatures for a range of applications without any prior knowledge, with 96-98% of overall accuracy.

Keywords: Network traffic monitoring, Traffic classification,Topic model, Deep packet inspection, Contents analysis, LDA, Signature extraction

Student Number: 2011-20891

Contents

List of Figures

List of Tables

Chapter 1Introduction

1.1 Background

Traffic classification, particularly according to the causing applications, has gained substantial attention within the Internet research and operational community, as it is a fundamental and essential task for understanding, optimizing, securing, and financing the current network infrastructure as well as planning improvements in future network architectures [28]. Internet traffic had traditionally been classified using well-known transport layer port numbers. However, this approach has become less reliable as an increasing number of applications hide their identity by using dynamically assigned random port numbers to communicate [15].

As a more accurate and reliable alternative, traffic contents analysis based Deep Packet (Payload) Inspection (DPI) technique was proposed, which inspects packet payloads to find specific application-level signature strings of known applications. While DPI has been found very accurate once given a set of known payload signatures for corresponding applications, it is very time consuming and challenging to manually derive and construct application signatures; (i) due to the lack of openly available standard protocol specifications for many applications, (ii) there are various implementations not fully complying with the specifications in the available documentation, if any, and (iii) application signature strings may change over time as the causing applications or protocols evolve; thus the application signature construction process has to be repeatedly performed to keep up with the changes [15].

1.2 Main Idea and Contributions

To solve the challenges, this paper proposes a novel method that automatically performs traffic contents analysis and develops accurate payload signatures for a range of different Internet applications, even without a priori knowledge on the application signatures. Our proposed method, named TASTE (Topic-model based Automatic Signature Extraction), uses the Latent Dirichlet Allocation (LDA)-based topic model [3], which has become one of the most popular text modeling techniques for extracting and classifying latent semantic information from text corpora. LDA has been shown to be effective in some text-related tasks such as document classification; the feasibility and effectiveness of using LDA in classifying network traffic still remains unknown. By using the LDA-based topic model on a broad range of real traffic traces, this paper empirically shows the capability of automatic extraction of a list of keywords (i.e., payload signatures) per LDA-derived topic (i.e., a group of words or payload strings generated from an application) and utilization of them for accurate traffic classification.

We highlight the main contributions from this paper:

- Proposed traffic classification is based on topic modeling. Latent DirichletAllocation (LDA) algorithm for traffic clustering without any prior knowledge and application signatures is used..
- The approach produces highly pure clusters from traffic by extracting topic signatures (topic word lists) and matching them to flow payloads.

- The approach uses only a signal packet payload of each flow. Only inspecting a packet for a flow can improve CPU resource consumption and speed.
- Experimentations have been done with five campus and careful analyses have been performed on the impacts of each parameter required for the processing. Final results indicate that this approach can achieve highly accurate classification results with low error rates.

1.3 Thesis Organization

The rest of the paper is organized as follows. Section 2 is dedicated to the related works. Section 3 introduces the concepts of LDA topic modeling and proposes a simple methodology for applying traffic trace to topic modeling and getting pure clusters. Section 4 explains the data sets and the processes of parameter settings. Section 5 includes classification using offline files and fair comparison with the existing classification techniques and also shows the classification results with raw traces. Finally, Section 6 and 7 conclude our work with future research direction.

Chapter 2Related Work

Traditionally, network-level application analysis has depended heavily on identification via well-known ports. New application patterns, particularly P2P use, undermined thisassumption, leading measurement researchers to seek workarounds. Several researches [22, 8] have shown that using port numbers can only classify traffic with 70% accuracy, which is not sufficient. One class of solutions focuses on deeper structural analyses of communication patterns, including the graph structure between IP addresses, protocols and port numbers over time, and the distribution of packet sizes and inter-arrival times across connections [17, 16, 18, 25, 1, 11, 27, 12, 9]. These approaches depend on the uniqueness of specific communication structures within a particular application. While this approach has been shown to work well for separate application classes (e.g., Mail vs. P2P), it is most likely unable to distinguish between application instances (e.g., one P2P system vs. another).

Techniques that rely on inspection of packet contents [22, 24, 14, 23, 29] have been proposed to address the diminished effectiveness of typical traffic classification. An alternative approach, application layer signature mapping, involves the exhaustive search of reliable signatures with more promising accuracy. Payload-based classification approaches inspect packet payloads to find targeted application signatures. This approach assumes that most application traffic contains signaling packets, and the signatures in signaling packets are unique. By identifying these signatures, we can classify which flow is generated by which application.

Payload-based approaches show reliable accuracy if the signatures are exactly and uniquely extracted. In fact, commercial bandwidth management tools use application signaturematching to enhance robustness of

classification.Early efforts focused on using hand-craft string classifiers to overcome the limitations of port-based classification for various classes of applications. Existing approaches to application signature identification (e.g.,[24, 7, 22]) involved a labor-intensive process combining information from available documentation with information gleaned from analysis of packet-level traces to develop potential signatures. Such a painstaking manual approach will not scale if it has to be applied individually to the growing range and number of diverse Internet applications.

Even for well known protocols, constructing good signatures is a delicate job, requiring expressions that have a high probability of matching the application and few falsematches to instances of other protocols. The problem has been addressed by Haffner et al.[14], who automate the construction of protocol signatures by employing a supervisedmachine learning approach on traffic containing known instances of each protocol. The system, ACAS[14] uses the first N byte payload as the input to train a machine learning model and uses it to classify flows. Their results are quite good, frequently approaching the performance of good manual signatures. Park et al.[23] have generated the signatures of not only traditional applications, such as HTTP, FTP, and more, but also newer applications like P2P by analyzing the packetcontents. P2P applications are the popular choice of target applications in other similar research because of its high traffic usage, complexity of traffic dynamics, and communication via undisclosed proprietary protocols. Obtaining the signatures of traditional applications, such as HTTP, FTP, and more, is relatively easy while their protocol format and communication behavior are publicly available. They propose LASER algorithm which tries to find the longest common subsequence among samples. The method is sensitiveto the noise in the samples and the comparing order. Moreover, there is a big challenge in generating

signature if different common substrings exist in the given application.Autosigsystem [29] solves the challenges as follows. First, to filter the noises in extracting common short shingles (A contiguous data block is called a shingle), shingles are divided into different groups according to their positions. Second, redundant and short shingles will increase the false positive when they are used directly as signatures. They are merged into common substrings in AutoSig. The adaptive shingle merging algorithm is proposed to overcome the over merge problem which incurred by greedy merge policy. Last,AutoSig generated multiple common substring sequences as the application signature to increase the accuracy of the signatures.Above these methods [29, 23, 14] try to extract the specific substring in the payload while our work is aim to classify Internet traffic automatically without prior knowledge.Chung et al.[4, 5] focus on how to utilize the payload data without heavy packet inspection while achieving reasonable accuracy. The work provides a classification method withutilizing cosine similarity between network flows. They converted payloads into a vector representation, and then calculated the similarity between payload vectors to classifythem. This method does not need deep level packet inspection or exhaustive signature extraction.Even if these approaches are guaranteed to automatic signature extraction and traffic classification, these approaches present several drawbacks with respect to real time traffic classification. it presupposes that network managers know what protocols they are looking for. In fact, new application protocols come into existence at an alarming rate and many network managers would like to be alerted that there is "a new popular application on the block" even if they have no prior experience with it.Levchenko et al. build several probabilistic models on payload[20] including the statistical model treating each n byte flow distribution as a product of N independent byte-

distribution and the Markov process model which relies on introducing independence between bytes. They builds further upon this approach by removing the requirement thatthe protocols be known in advance. By simply using raw network data, the unsupervised algorithms classify traffic into distinct protocols based on correlations between theirpacket content. They try to use the byte distribution in the payload. Thus, using no a priori information the method is able to create classifiers that can then distinguish between protocols. In this sense(i.e., of being unsupervised), our approach is similar in spirit to that of Levchenko et al.[20], who suggest using the initial payloads of the first packets in a session as the protocol signature. However, even though the alternative is extremely accurate, the method needs a high storage and computational cost to study every packet that traverses a link (in particular on very high-speed links). Additionally, they evaluated the automated schemes only on conventional applications such as FTP, SMTP, HTTP, HTTPS, SSH, DNS, and NTP, not on newer applications such as P2P, Games, and Streaming.

Chapter 3Topic Model Based Classification

We first begin with a brief review of the concept of modeling latent topics. Latent topic models such as Latent Dirichlet Allocation (LDA) [3] provide a means to take advantage of the statistical structure of the corpus itself. LDA assumes that each observed document has been generated by weighted mixtures of unobserved (latent) topics, which are learned from the documents and often correspond to meaningful semantic themes present in the corpus. As a statistical topic model for discovering topics in large text document corpus, LDA has been quite popular in the realm of text document classification, identifying latent topics fromtext documents, computer vision, and social network analysis [2]. But its applicability in extracting application signature strings from Internet traffic data has not been explored so far. To the best of our knowledge, this is the first attempt at applying LDA in the context of network traffic classification.

3.1 LDA

In machine learning and natural language processing, a topic model is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents. Probabilistic topic models [14, 2] are a suite of algorithms, the goal of which is to discover the hidden thematic structure in large archives of document. Topic modeling algorithms are statistical methods that analyze the words of the original texts to discover the themes that run through them and do not require any prior annotations or labeling of the documents in which the topics emerge from the analysis of the original texts.

The topic model for traffic classification that this paper proposes utilizes LDA algorithm. In LDA [3], it is assumed that observed words in each document are generated by a document-specific mixture of topics and these topics are corpus-wide latent structures. We define our corpus of length N with the flat word vector $\vec{w} = w_1 \ldots w_N$. At corpus position i, the element c_i in $\vec{C} = c_1 \ldots c_N$ designates the document containing observed word w_i. Similarly, the vector $\vec{z} = z_1 \ldots z_N$ defines the hidden topic assignments of each observed word. The number of latent topics is fixed to some K, and each topic $k = 1 \ldots K$ is associated with a topic-word multinomial β_k over W, vocabulary words. Each β multinomial is generated by a conjugate Dirichlet prior with parameter η. Each document $d = 1 \ldots D$ is associated with a multinomial θ_d over K topics, which is also generated by a conjugate Dirichlet prior with parameter α. With this notation, the full generative model is then given by

$$\left(\prod_{k=1}^{K} P\left(\beta_k \mid \eta\right)\right)\left(\prod_{d=1}^{D} P\left(\theta_d \mid \alpha\right)\right)\left(\prod_{i=1}^{N} \beta_{z_i}\left(w_i\right)\theta_{d_i}\left(z_i\right)\right) \quad (1)$$

where $\beta_{z_i}\left(w_i\right)$ is the w_i-th element in vector β_{z_i}, and $\theta_{d_i}\left(z_i\right)$ is the z_i-th element in vector θ_{d_i}. Given an observed corpus $(\vec{w}, \vec{C})$ and model hyperparameters (α, η), the typical modeling goal is to infer the latent variables $(\vec{z}, \beta, \theta)$. The algorithm computes the conditional distribution of the topic structure given the observed documents (that is called the posterior).

While exact inference is intractable for LDA, a variety of approximate inference algorithms have been developed [3, 29]. These approximation schemes generally fall into two categories, sampling based algorithms and variational algorithms. Sampling based algorithms attempt to collect samples from the posterior to approximate it with an empirical distribution. On the other hand, as a deterministic alternative to sampling based algorithms, variational methods posit a parameterized family of distributions overthe hidden structure and then find the member of that family closest to the posterior (rather than approximating the posterior with samples), thus transform the inference problem to an optimization problem [2]. This paper applies the variational methods presented in [3], to which readers are referred for more details due to space limitations.

3.2Applying LDA to Traffic Classification

3.2.1Overview

The basic idea of LDA is that a document can be viewed as a mixture of a limited number of topics and each meaningful word in the document can be associated with one of these topics [23]. Given a corpus of documents, LDA attempts to discover the followings: (i) it identifies a set of topics, (ii) it associates a set of words with a topic, and (iii) it defines a specific mixture of these topics for each document in the corpus [23]. To apply LDA in traffic classification and identification, we consider a traffic trace to be a collection of traffic flows (with payload data) and each traffic flow is associated with the application (i.e., a dominant topic) that generated it. Thus a traffic flow can be thought of as and mapped to a document with a

dominant topic, which is asignature string of the corresponding application. Given this mapping, application of LDA to traffic data sets is depictedin Figure 1.

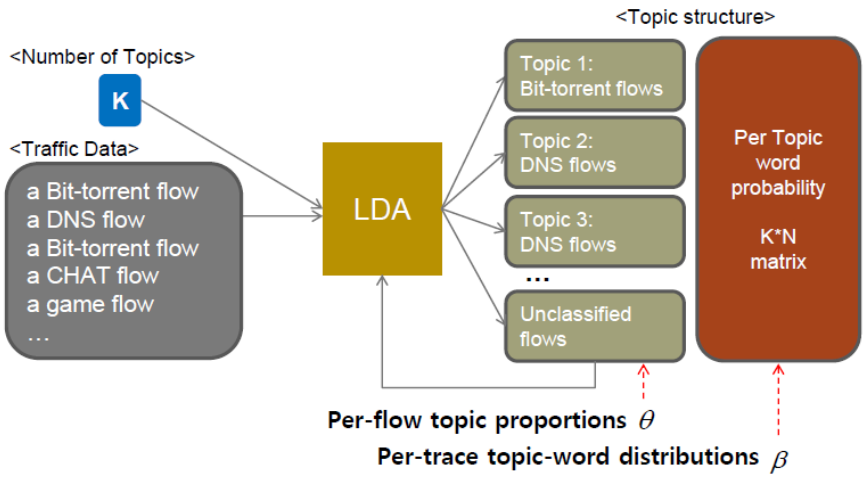


Figure 1.Applying LDA to Traffic Classification

3.2.2Inputs: Data pre-processing and Parameters

As in Figure 1, LDA takes input in the form of a dataset of flow instances. An instance refers to an individual, independent flow payload vector, characterized by the <word:frequency> pairs contained in the flow's payload. In the context of classical document classification, words are typically defined as consecutive bytes separated by delimiters such as spaces or punctuation marks. However, the same definition cannot be applied in this case because traffic payload data, containing both hex and ASCII strings, do not have such separators. With the absence of such delimiters, we define word as a sequence of consecutive bytes within a sliding window covering i bytes, which slides one byte at a time from the first to the n-i+1

th byte for a traffic flow with n bytes payload data. We discuss observed classification accuracies against varying i values in section 4.3.

In extracting words as well as their observed frequencies from a given set of traffic traces, TASTE considers initial payload data only; i.e., words contained in the first n bytes (of the first data packet) of a traffic flow, since (i) the earlier the better, for identifying application traffic in an operational network, (ii) for most applications (except Gnutella2), payload signatures are located at the beginning of a data flow [22, 24, 14, 23, 29, 18, 25, 17], and (iii) signature matching is a resource-intensive and expensive task whose cost is proportional to the length of the signatures to compare, thus we want to minimize the length of payload data that have to be processed. To satisfy these desirable requirements for early application identification, TASTE inspects only (i) the first packet of UDP application flows and (ii) the first (payloaded) data packet exchanged right after the three-way handshaking process of TCP applications. We discuss observed classification accuracies against varying n values in section 4.3.

As LDA does not know the number of topics of the corpus beforehand, the number of topics (K) is required as an input parameter of the model. We have observed that varying the number of topics has a significant impact on the classification accuracy, as discussed in Section 4.3.

3.2.3Outputs: Topics and their Distributions

The result of LDA is a set of topics and a distribution of these topics in each flow. A topic is a collection of automatically extracted words (i.e., payload strings) along with the importance of each string to the topic represented as a numeric fraction. The model is then used for clustering

flows and extracting most probable words per topic, i.e., signature strings per application. The distribution of words for each topic from the topic-word multinomial is used to extract the highest probable “Top L" words per topic. We define the L most probable words for topic z as Wz, given by the following expression:

$$W_z = L\text{-}\arg\max_{w} \beta_z(w)$$

Finally, we adopt majority voting to assign each flow to a dominant topic, with which the flow shares the largest number of “Top L" words in its payload data. When there is no such topic, the flow is left unclassified. This is often the case (i) when with the data/class imbalance problem [27,10], where some minority classes are heavily outnumbered by others in the data set thus their words have failed to be included in the lists of “Top L" words, or (ii) when the flow contains an equal number of “Top L" words from multiple different topics. Unclassified flows are collected until all the input flows are processed by LDA, then forwarded again as input data to the next round of LDA classification, as shown in Figure 1. The multi-round classification processes are repeated until every flow is assigned to a topic, with max 10 rounds. We measure the accuracy of classification results obtained after all the rounds.

Chapter 4METHODOLOGIES

This section describes our methodology for LDA-based traffic classification, including performance metrics, data set, establishing reference benchmark (i.e., ground truth), and experimental setup for applying LDA to traffic classification.

4.1. Performance metrics

To measure the performance of the proposed scheme, we use three metrics: overall accuracy, precision, and recall.

- Overall accuracy: the ratio of the number of correctly classified traffic flows to the total number of all flows in a given trace. We apply this metric to measure the accuracy of a classifier on a set of whole trace. The following two metrics are to evaluate the quality of classification results for each application class.

- Precision: the ratio of True Positives over the sum of True Positives and False Positives or the percentage of flows that are properly attributed to a given application.

- Recall: the ratio of True Positives over the sum of True Positives and False Negatives or the percentage of flows in an application class that are correctly identified.

4.2. Data set and reference benchmark

Table1.Characteristics of analyzed traces

Set	Data	Day	Start	Duration	Link type	SrcIP	DstIP	Packets	Bytes	Avg.Flows(/5min)
Keio-I	2006/08/06	Tue	19:43	30m	edge	73K	310K	27M	16G	155K
Keio-II	2006/08/10	Thu	01:18	30m	edge	54K	110K	25M	16G	77K
WIDE-I	2007/01/09	Tue	08:02	2h	Backbone	318K	480K	159M	53G	267K
WIDE-II	2012/03/30	Fri	00:02	45m	Backbone	340K	530K	214M	82G	419K
KAIST	2008/03/18	Tue	19:00	9h	edge	135K	103K	278M	98G	36K

Our data sets consist of five anonymized payload traces with payload data collected at two edge links and one backbone link located in the Japan and Korea (Table 1). The WIDE traces were captured at a 150 Mbps Ethernet US Japan Trans-Pacific backbone link that carries commodity traffic for WIDE member organizations. The Keio traces were collected on a 1 Gb/s Ethernet link in Keio UniversityShonan-Fujisawa campus. The KAIST trace was captured at one of four external links connecting a 1Gb/s KAIST campus network and a national research and network in Korea.

To establish a reference point in evaluating the algorithms, we use the payload-based classifier crl_pay, of which application signature sets have been developed and used in [19, 28, 18, 26, 10, 32, 21, 16]. The resulting classifier includes payload signatures of various popular applications, summarized in Table 2. The payload classification procedure examines the payload contents of each packet against signature strings, and in case of a match, this procedure classifies the corresponding flow with an application-specific tag. Previously classified flows are not re-examined again unless they have been classified as HTTP, in which case re-examination may allow identification of non-Web traffic relayed over HTTP (e.g., Streaming, P2P, etc.)[18]. Flows that could not be classified with the

crl_pay code's signature matching process (i.e., flows (i) without payload data or (ii) whose ground truth is not available to us) as well as attack flows such as port or address scanning ones are excluded from our analysis.

Table2.Application Categories

Category	Application/protocol
Web	HTTP, HTTPS
P2P	FastTrack, eDonkey, BitTorrent, Ares, Gnutella, WinMX, OpenNap, MP2P, SoulSeek, Direct Connect, GoBoogy, Soribada, PeerEnabler, Napster Blubster, FileBEE, FileGuri, FilePia IMESH, ROMNET, HotLine, Waste
FTP	FTP
DNS	DNS
Mail/News	BIFF, SMTP, POP, IMAP, IDENTD, NNTP
Streaming	MMS(WMP), Real, Quicktime, Shoutcast, Vbrick Streaming, Logitech Video IM Backbone Radio, PointCast, ABACast
Network Operation	Netbios, SMB, SNMP, NTP SpamAssasin, GoToMyPc, RIP ICMP, BGP, Bootp, Traceroute
Encryption	SSH, SSL, Kerberos, IPSec, ISAKMP
Games	Quake, HalfLife, Age of Empires, DOOM Battle field Vietnam, WOW, Star Sieze Everquest, Startcraft, Asherons, HALO
Chat	AIM, IRC, MSN Messenger, Yahoo messenger IChat, QNext, MS Netmeet, PGPFone, TALK

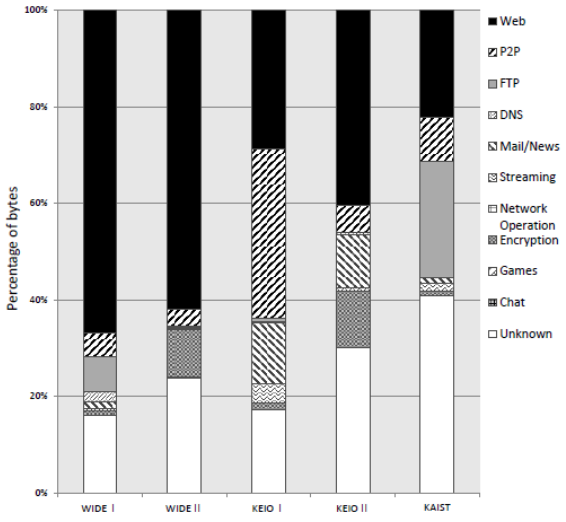


Figure 2.Percentage of flows for five traces

4.3. Parameter Setting

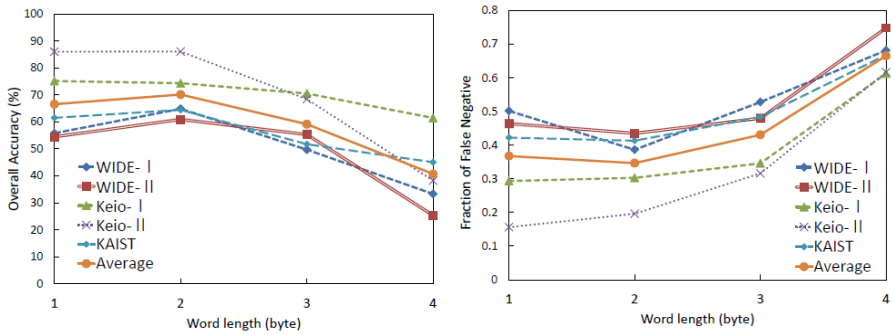
This section measures the impact of each parameter in applying traffic to LDA topic modeling and then finds the appropriate settings of parameters that can generally achieve good result. Five notations for each parameter are defined as follows: the number of topics K, the payload inspection range of a flow signal packet P, the length of a word W, threshold T, the number of words per topic L.

To figure out the impacts of parameters and general trend, and to avoid the class imbalance problem [27, 10], sampled data sets from each trace in 1 are used because this unequal distribution from raw data does not allow for equal testing of the different classes. All these data sets consist of 1000 random flow samples of each traffic class. In addition, to achieve a greater confidence in the results, five different data sets for each trace are generated. Each of these data sets was then, in turn, used to evaluate the topic model based algorithms. The average results from the data sets of each trace were documented, and then final selection of general parameter settings for five traces were made, considering overall accuracy, false negative ratio and speed rather than selecting optimal settings for each trace, because, in real classification, the magic number cannot be known and this number depends on each traces.

A. Word length

The impact of word length was evaluated with W initially being 1byte and W being incremented by one byte for each trace. Considering performance and efficiency, the remaining initial parameters are N=5(bytes), K=10, L=4, T=100 based on measurements from [18, 25, 23, 18]. The minimum, maximum, and average results for the word length are shown

in Figure 3. According to the figure, when the word length was 2-byte, better results were produced in terms of overall accuracy and false negative ratio. It is acknowledged that fixing the word length to 2-bytes gives the best result for traffic clustering by TASTE. The larger the word length W is, the more specific the extracted words are. However, common words (application signatures) are usually very short. The long words may represent the payloads of flows look differently even if the flows are generated by the same application, which drops overall accuracy and increases false negative errors. From this measurement, W is set to 2-bytes in the system.



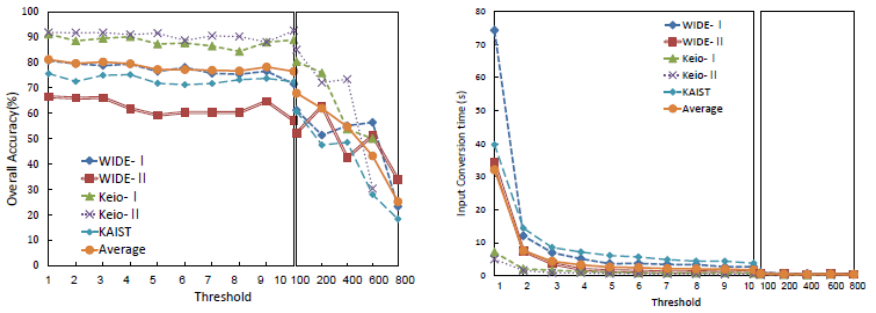
(a) Overall Accuracy (b) Fraction of False Negative

Figure 3.Overall accuracy and FNs by word length

B. Threshold

The impact of threshold T was measured. Using the all words consumes a huge amount of processing time especially when converting traffic data to the LDA input form (Vectors composed of word:frequency elements in each payload). For example, with the 2byte word length, about 62599 words were normally extracted from five byte range of about 740000 flow payloads. For this reason, the threshold T was applied to filter unnecessary words and to reduce processing time of input data. For example,

when T=2, words with less than 2 times total frequency are eliminated. The value of threshold T was tested between 1 and 10 by increment of 1, and between 100 and 800 by increment of 200. Measurements of overall accuracy and processing time were taken for input for converting payload data to the form of LDA input. The other parameters are K=10, L=4, N=5(byte) and W=2(byte). The impact on overall accuracy and processing time for input conversion of threshold values is shown in Figure4. As seen in the result, some necessary words are missing if T is too big, meaning the reduction of accuracy. In contrast, small threshold increases the time consumption for the processing time. Taking the balance of both overall accuracy and speed into account, it is determined to be the best to set the threshold to 2.



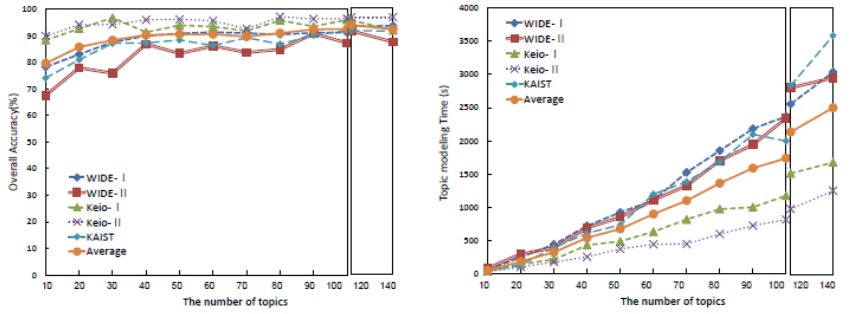
(a) Overall Accuracy (b) Processing time for input conversion

Figure 4.Overall accuracy and
Input data processing time by threshold T

C. The number of topics

The LDA algorithm takes an input parameter of K. For different applications, the optimum value of K varies. In addition, due to the diversity of the traffic in some classes such as HTTP (e.g., streaming, browsing, bulk download) it is expected that even more clusters to be formed. Therefore,

based on this, our approach based on LDA algorithm was evaluated with K initially being 10 and K being incremented by 10 for clustering. Measurements of the trend of overall accuracy and the topic modeling time with varying K are shown in Figure 5. The number of topics K has a strong impact on the result. Initially, when the number of clusters is small, the overall accuracy of our approach is approximately 70-90% and the average overall accuracy is 80%. The overall accuracy steadily improves as the number of clusters increases. This continues until K is approaching around 100 or 120with overall accuracy being 87-96% and 93-94% on average for each five data set, respectively. However, the large values of K increase processing time for LDA and reduce efficiency. Selecting optimal value of K is one of the big challenges of clustering algorithms. In practical real traffic classification, the optimal value of K is dependent on each trace and it is hard to determine the optimum value of K. From the results, the value K is set to 100 for our further experiments. Although not optimal, the general value across five data-sets is expected to bring high accuracy and moderate processing time for larger size data sets.

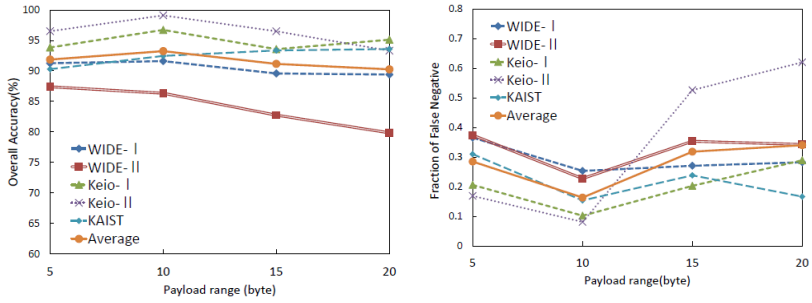


(a) Overall Accuracy (b) Processing time for topic modeling

Figure 5.Overall accuracy and
Topic modeling timeby number of topics K

D. Payload byte range

In experimentation of the impact of N, the byte range of a flow's payload was used. The value of parameter N was varied 5bytes to 20bytes by the increment of five bytes in each step. The remaining parameters are W=2(byte), K=100, T=2 and L=4 and the results in terms of overall accuracy and false negative ratio are shown in Figure 6. In Figure 6(a), the payload range of initial 10bytes brings the best overall accuracy and minimum false negatives across all the five data sets. Therefore, the payload inspection range N was set to initial 10bytes.



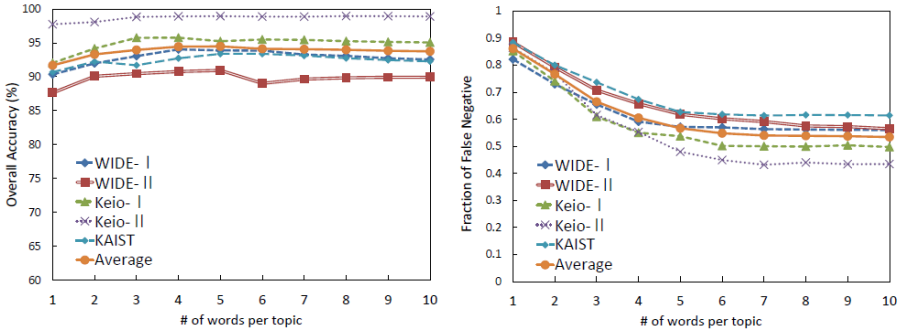
(a) Overall Accuracy (b) Fraction of False Negative

Figure 6.Overall accuracy and False Negative ratio by payload range to be examined

E. The number of words per topic

The parameter L is used for our direct-matching strategy in Section 3.2. This strategy creates pure clusters by checking whether a flow of payload includes the most probable L words for each topic and by assigning the flow to the topic when they are matched. Experimentation was performed on the overall accuracy and False Negatives with L initially being 1 and L being incremented by 1 for assigning flows. At each setting, the most

probable L words(Wz) were extracted for each topicz according to the topic-word distribution. The remaining parameters are K=100, W=2(byte), N=10(byte) and T=2. The result of one step clustering was measured to figure out the only impact of the parameter L and shown in Figure 7. Figure 7 indicates that the overall accuracy steadily improves until the value of L approaches to 5. The number of false negatives decreases as the value of L increases. However, after the value of L gets closer to 5, accuracy reductions and false negative improvements were found to be little. The reason is that as the value L gets larger, the words which isnot related to the corresponding topic are extracted more. Therefore, the number of words per topic L is set to 5.



(a) Overall Accuracy (b) Fraction of False Negative

Figure 7.Overall accuracy and False Negative ratio
by the number of words per topic

Chapter 5RESULTS

We evaluated the performance of TASTE. First, the performance of other classification approaches was compared with the TASTE classifier. In this case, the overall accuracy metric is used to figure out the difference of accuracy, and the sampled traces were examined for equal comparison. Also an analysis was done on the overall accuracy and the classification quality of TASTE with raw data traces. The parameter setting is equal to above experimental settings(W=2,N=10,K=100,T=2 and L=5).

5.1. Performance Comparison

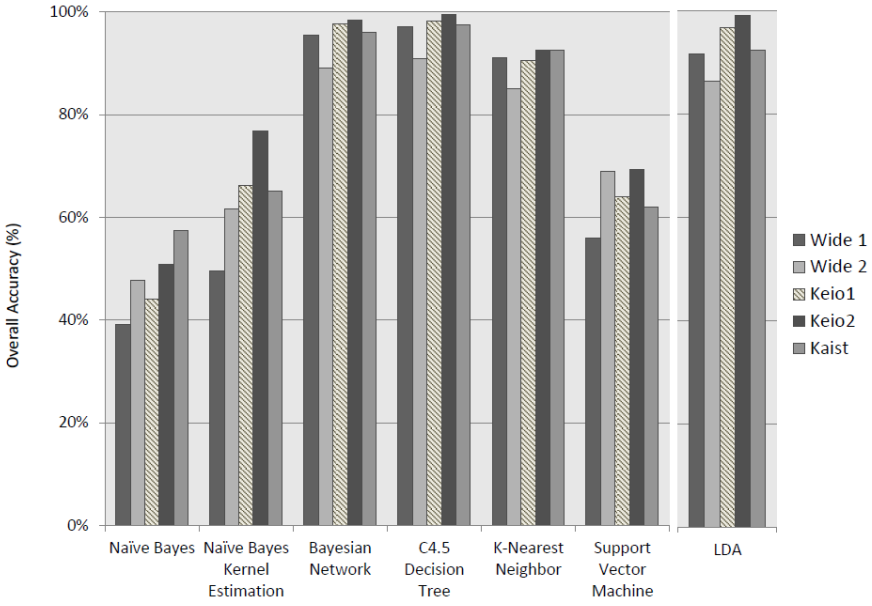


Figure 8.Overall Accuracy comparisons of feature-based approach with-supervised ML.s and LDA topicmodel based approach

The performance of other classification approaches was compared with TASTE classifier. The WEKA machine learning software suite is used for other classification schemes [18]. Due to the high classification accuracy, Flow-feature based classification approaches with supervised learning were chosen as the comparison schemes. We experimented the schemes by using various machine learning algorithms such as Naïve Bayes(with Kernel Estimation), Bayesian Network, C4.5, K-Nearest Neighbor and Support Vector Machine algorithm. The results were estimated by 10-fold cross-validation. In this case, the overall accuracy metric is used to figure out the difference of accuracy and the sampled traces were examined for equal comparison. Figure 8 shows the classification results of the approach and existing schemes. In Figure 8, Flow-feature based approaches with supervised learning can guarantee more than 90% accuracy except for few machine learning algorithms because supervised classifier can be turned in detail resulting in good accuracy. However these approaches need sufficient amount of training examples and are available only in case with the prior knowledge of applications to be classified [18]. Clustering can solve the limitation of supervised mode. TASTE has a merit of unsupervised approach and can achieve highly accurate classification results comparable with supervised learning based approach. As a result, the overall accuracy of TASTE(LDA) can reach about 93-99% depending on each trace, exhibited in Figure 8.

5.2. Classification Quality

Additional analysis was performed on the classification quality of TASTE by using two metrics, precision and recall with raw traces in Table 1. Because the above results are the classification results for sampled traces

balanced with equal traffic volume of each application, real traffic trace may have

data-imbalance and reduce classification performance. In this experiments with large raw data sets, application traffic using fixed standard port numbers that can classify easily was excluded for the purpose of efficiency. Table 3 shows the overall accuracy and classification quality for raw traces. The results are highly accurate even with real data trace which have different distribution of application flows.

Table3.Per-application precision (%) and recall (%) of TASTE

(a) Classification Quality of Keio1, Keio2 traces

	Keio-I			Keio-II		
App	flows(%)	Precision	Recall	flows(%)	Precision	Recall
SpamAssasin	34.5	98.16	96.61	42.9	94.02	99.45
NTP	30.5	97.87	99.36	40.6	98.22	100
CHAT	15	98.6	92.09	3.9	93.33	88.96
BitTorrent	10.3	98.31	85.73	-	-	-
SNMP	3.6	100	98.52	7.3	100	100
Streaming	2.9	92.76	87.31	2	99.16	75.64
IRC	0.8	100	67.02	0.9	90.32	42.42
GoToMyPC	0.5	96.08	94.23	0.3	-	-
others	1.9	-	-	2	-	-
Overall	97.95%			96.13%		

(b) Classification Quality of Wide1, Wide2 and KAIST traces

	WIDE-I			WIDE-II			KAIST		
App	flows(%)	Precision	Recall	flows(%)	Precision	Recall	flows(%)	Precision	Recall
BitTorrent	26.4	97.1	84.01	61.9	96.56	98.65	52.7	99.28	92.74
SpamAssasin	24.2	99.91	62.32	0.4	99.61	41.88	-	-	-
CHAT	15.7	96.56	68.43	1.2	71.32	21.58	2.8	88.32	94.68
NTP	11.3	93.49	95.44	27.9	96.7	99.9	19.4	99.58	98.79
SMB	5.8	97.06	66.34	3.2	94.76	96.08	8.2	93.46	98.97
eDonkey	4	99.75	84.13	1.6	65.84	80.52	4.9	85.87	73.75
IRC	2.8	93.25	94.73	1.4	78.57	2.24	2.8	99.38	88.61
Gnutella	2.7	86.09	26.13	0.2	91.52	96.28	0.1	93.85	79.22
Goboogy	1.2	94.77	96.53	0.01	-	-	0.8	90.16	32.42
SNMP	1	98.8	91.29	1.3	90.41	99.14	-	-	-
News	0.3	96.35	34.02	-	-	-	0.001	-	-
Streaming	0.2	78.02	20.94	0.001	-	-	8.2	99.54	95.94
others	4.3	-	-	0.93	-	-	0.11	-	-
Overall	96.84%			95.69%			97.86%		

Chapter 6DISCUSSION

The list of most probable words for each topic can serve as key signatures for clustering with high purity because the words are extracted from initial payloads in which the most part is composed of signatures. As such, the word lists of topics have the possibility of getting general application signatures. The original limitation of the payload based approach, even if its classification result is highly accurate, still exists in this topic model based approach. The payload based sequence matching has the fundamental challenges with time complexity and may not be feasible when the payload is encrypted. However, scalability problem can be eased by employing fast and simple port-based approach as pre-processing step. It curtails the payload based works by reducing the size of data and only unclassified flows could be processed to our approach.

Chapter 7Conclusion

In this paper, we propose a new, automatic payload content analysis classification method called TASTE (Topic-model based Automatic Signature Extraction) for the network traffic classification problem. Our method is based on the Latent Dirichlet Allocation (LDA), which is one of the most popular probabilistic document modeling techniques for extracting latent semantic information from text corpora.

We employed LDA to traffic classification as a simple suggestion for the mapping and clustering strategies. TASTE is unsupervised learning based classification using payload data and has an advantage of clustering raw traffic trace without any prior knowledge like application signature library or full flow statistics. Our analysis is based on TASTE classifier's ability to produce clusters that have a high predictive power of a single traffic class, and the strategy of the parameter combination that can generally produce the best clustering results. The method can get not only highly pure clusters, but also a set of key words for each topic from LDA topic model. Additionally, the results showed that, without a priori knowledge, TASTE produces the high overall accuracy compared with the different supervised classification approaches. Thus, labeling a single protocol instance is sufficient to classify all such traffic. In effect, we have substituted the painful process of manual flow analysis and classifier construction with the far easier task of recognizing a protocol instance. TASTE is guaranteed to produce highly accurate classification results, in which the overall accuracy can reach about 96-98% with high classification quality using real-world raw traffic traces.

Bibliography

[1] L. Bernaille, R. Teixeira, and K. Salamatian. Early application identification. In CoNEXT, page 11, 2006.

[2] D. M. Blei. Introduction to probabilistic topic models. Communications of the ACM, 2011.

[3] D. M. Blei and A. Y. Ng. Latent dirichlet allocation. In Journal of Machine Learning Research, pages 993-1022, 2003.

[4] J. Y. Chung, B. Park, Y. J. Won, J. Strassner, and J. W. Hong. Traffic classification based on flow similarity. In IPOM, pages 65-77, 2009.

[5] J. Y. Chung, B. Park, Y. J. Won, J. Strassner, and J. W. Hong. An effective similarity metric for application traffic classification. In NOMS, pages 286-292, 2010.

[6] K. C. Claffy, H.-W. Braun, and G. C. Polyzos. A parameterizable methodology for internet traffic flow profiling. IEEE Journal on Selected Areas in Communications, 13(8):1481-1494, 1995.

[7] A. Dainotti, W. de Donato, and A. Pescape. Portload: Taking the best of two worlds in traffic classification. In INFOCOM, pages 1-5, 2010.

[8] C. Dewes, A. Wichmann, and A. Feldmann. An analysis of internet chat systems. In Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, IMC '03, pages 51-64, New York, NY, USA, 2003. ACM.

[9] H. Dreger, A. Feldmann, M. Mai, V. Paxson, and R. Sommer. Dy-

namic application-layer protocol analysis for network intrusion detection. In USENIX-SS, volume 15, 2006.

[10] J. Erman, M. Arlitt, and A. Mahanti. Traffic classification using clustering algorithms. In Proceedings of the MineNet, pages 281-286, 2006.

[11] J. Erman, A. Mahanti, and M. Arlitt. Byte me: a case for byte accuracy in traffic classification. In Proceedings of the 3rd annual ACM workshop on Mining network data, MineNet '07, pages 35-38, New York, NY, USA, 2007. ACM.

[12] J. Erman, A. Mahanti, M. Arlitt, and C. Williamson. Identifying and discriminating between web and peer-to-peer traffic in the network core. In WWW, pages 883-892, 2007.

[13] J. Erman, A. Mahanti, M. F. Arlitt, I. Cohen, and C. L. Williamson. Offine/realtime traffic classification using semi-supervised learning. Perform. Eval., 64(9-12):1194-1213, 2007.

[14] M. Girolami and A. Kaban. On an equivalence between plsi and lda. In Proceedings of SIGIR, 2003.

[15] P. Haffner, S. Sen, O. Spatscheck, and D. Wang. Acas: automated construction of application signatures. In MineNet, pages 197-202, 2005.

[16] M. Iliofotou, H. chul Kim, M. Faloutsos, M. Mitzenmacher, P. Pappu, and G. Varghese. Graption: A graph-based p2p traffic classifi-

cation framework for the internet backbone. Computer Networks, 55:1909-1920, 2011.

[17] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh, and G. Varghese. Network monitoring using traffic dispersion graphs (tdgs). In Internet Measurement Comference, pages 315-320,2007.

[18] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. Blinc: multilevel traffic classification in the dark. In SIGCOMM, pages 229-240, 2005.

[19] H. Kim, K. C. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee. Internet traffic classification demystified: myths, caveats, and the best practices. In CoNEXT, page 11, 2008.

[20] S. S. Kim and A. L. N. Reddy. Image-based anomaly detection technique: Algorithm, implementation and effectiveness. IEEE Journal on Selected Areas in Communications, 24(10):1942-1954, 2006.

[21] S. Lee, H. Kim, D. Barman, S. Lee, C. kwon Kim, T. T. Kwon, and Y. Choi. Netramark: a network traffic classification benchmark. Computer Communication Review, 41(1):22-30, 2011.

[22] J. Ma, K. Levchenko, C. Kreibich, S. Savage, and G. M. Voelker. Unexpected means of protocol inference. In Internet Measurement Conference, pages 313-326, 2006.

[23] G. Maskeri, S. Sarkar, and K. Heafield. Mining business topics

in source code using latent dirichlet allocation. In Proceedings of ISEC, pages 113-120,2008.

[24] A. W. Moore and K. Papagiannaki. Toward the accurate identification of network applications. In PAM, pages 41-54, 2005.

[25] B.-C. Park, Y. J. Won, M.-S. Kim, and J. W. Hong. Towards automated application signature generation for traffic identification. In NOMS, pages 160-167,2008.

[26] S. Sen, O. Spatscheck, and D. Wang. Accurate, scalable in-network identification of p2p traffic using application signatures. In WWW, pages 512-521, 2004.

[27] S. Singh, C. Estan, G. Varghese, and S. Savage. Automated worm fingerprinting. In OSDI, pages 45-60, 2004.

[28] Y. sup Lim, H. chul Kim, J. Jeong, C. kwon Kim, T. T. Kwon, and Y. Choi. Internet traffic classification demystified: On the sources of the discriminative power. In CoNEXT, page 11, 2010.

[29] Y. W. Teh, D. Newman, and M. Welling. A collapsed variationalbayesian inference algorithm for latent dirichlet allocation. In NIPS, pages 1353-1360, 2006.

[30] Y. Wang, Y. Xiang, and S.-Z. Yu. Automatic application signature construction from unknown traffic. In AINA, pages 1115-1120, 2010.

[31] N. Williams, S. Zander, and G. Armitage. A preliminary per-

formance comparison of five machine learning algorithms for practical ip traffic flow classification. In ACM SIGCOMM Computer Communication Review, volume 36, pages 5-16, 2006.

[32] Y. J. Won, B.-C. Park, H.-T. Ju, M.-S. Kim, and J. W. Hong. A hybrid approach for accurate application traffic identification. In E2EMON, pages 1-8, 2006.

[33] M. Ye, K. Xu, J. Wu, and H. Po. Autosig-automatically generating signatures for applications. In Proceedings of the IEEE CIT, pages 104-109, 2009.

초록

최근 인터넷 사용이 대중화되면서 인터넷 유저들은 다양한 응용 프로그램을 사용하고 있다. 이메일이나 웹 서비스 같은 전통적인 인터넷 서비스뿐만 아닌 P2P 파일 공유, 멀티미디어 스트리밍 서비스 등 다양한 인터넷 응용프로그램이 등장하고 있으며 트래픽 또한 급증하고 있다. 따라서 다량으로 발생하는 트래픽을 정확히 파악하여 효율적인 네트워크 관리를 위해 트래픽 모니터링 및 분석의 기술이 점점 더 요구되고 있다.

네트워크 트래픽 분류 기술이 점점 더 발전해가고 있는 가운데 페이로드 기반 분석 방법은 패킷의 컨텐츠를 깊게 분석하는 방법이며 여전히 확실하고 정확한 트래픽 분류 방법 중 하나이다. 기존의 페이로드 기반 분석 방법은 미리 알고 있는 시그니쳐를 이용하여 트래픽을 분류한다. 이 방법은 확실한 반면 정확한 시그니쳐 셋을 미리 알고 있어야 한다. 즉 시그니쳐 추출을 위해서는 사전 전문 지식이 필요하며 시간 복잡도가 높은 단점이 있다.

본 논문에서는 자연어 처리 프로세싱 및 문서 검색 및 분류 분야에 널리 사용되는"토픽 모델링 기법"을 이용해 트래픽을 페이로드 기반으로 어떠한 사전 정보도 필요 없이 클러스터링하고 분석하는 기술을 제안한다. 또한 토픽 모델링을 이용해 토픽 당 유력한 워드를 추출하여 응용 프로그램에 대한 키워드, 즉 시그니쳐를 추출하는 기법을 소개한다. 마지막으로 실제 트레이스를 이용해 토픽 모델링 기반 트래픽 분류를

실험한 결과 95%~98%의 우수한 정확도로 결과를 얻을 수 있음을 증명하였다.

주요어: 네트워크 트래픽 모니터링, 트래픽 분류 기술, 깊은 패킷 분석, 컨텐츠 분석, 시그니쳐 추출, 토픽 모델, LDA

학번: 2011-20891