ISSN : 0377-2063

The Institution of Electronics and Telecommunication Engineers

IETE Journal of Research

Volume 59 • No. 4 • July-August 2013

www.ietejournals.org

Flow Cluster Algorithm Based on Improved K-means Method

Shi Dong[1,2,4,5], Dingding Zhou[3], Wei Ding[2,4,5], and Jian Gong[2,4,5]

[1]School of Computer Science and Technology, [3]Department of Laboratory and Equipment Management, Zhoukou Normal University, Zhoukou, 466001, [2]School of Computer Science and Engineering, Southeast University, [4]Jiangsu Key Laboratory of Computer Networking Technology, [5]Key Laboratory of Computer Network and Information Integration, Ministry of Education, Nanjing, 211189, China

ABSTRACT

Traffic classification is currently an important challenge for network management. In recent years, some traffic classification and identification algorithms have been proposed; identifying encrypted application traffic represents an important issue for many network tasks including quality of service. Port number-based classifiers work only for well-known applications and signature-based classifiers are not suitable for encrypted packet payloads. So researchers tend to identify network traffic based on behaviors observed in network application. But the results are so far limited in scope and frequently disappointing. In this paper, flow identification method is proposed to identify network flows based on traffic statistic, which adopt improved k-means cluster algorithm (SA-k-means) to classify traffic, and analyze the impact factor of cluster. Also, experiment results show SA-k-means method is effective.

Keywords:
Application behavior characteristic, Application identification, Metric correlation, Network management, Network measurement.

1. INTRODUCTION

Traffic classification is crucial for classic network management tasks, such as traffic engineering and capacity planning. However, traffic identification is a difficult problem that requires the use of very complex identification techniques. For many years, the use of port identification was widely used. The Internet Assigned Numbers Authority (IANA) [1] provides a port mapping table. Network traffic of a particular port belongs to a particular network application. However, in recently years, many new applications occur with different port. Port identification method is not suitable for traffic identification. With the variety of applications emerging, besides the traditional applications (e.g., http, email, web, and ftp), new applications such as P2P have gained strong momentum. So it will be an interesting work to classify traffic and identify applications. A number of areas, such as trend analysis and dynamic access control, can benefit from it. At the same time, accurate classification and identification of internet traffic is an important basis of network security and traffic engineering. Traffic statistics of different applications include Web, P2P file-sharing, and file transfer, reflecting user behavior while using the network, so it can be useful to help network administrators to control traffic such that traffic critical to business is given higher priority service on their network. On the other hand, the traditional classification methods that are based on supervised learning have limitations in practice, for they can only construct classifiers for the network flows whose types are already known to them, and new network applications are not recognized. In contrast, the unsupervised learning method, also known as cluster analysis, produces meaningful division according to the degree of similarity within the data sets. According to [2], for the definition of clustering, the entities within a cluster are similar, the entities of different types of clusters are dissimilar; a cluster is the point of convergence in test space, the distance between any two points with the same cluster is less than the distance between any two points in different clusters; the class cluster can be described as a connected region of multi-dimensional space that contains the points set with relatively high density, which separates other class cluster that contain points set with relatively low density from the whole original data sets. Traffic classification based on the unsupervised learning method overcomes the drawbacks of the supervised learning method: It divides the network flows into different clusters in line with the similarity of data sets, in such a way that the new unknown type of network applications can be identified on the basis of some different clusters. Given that the network traffic itself has a complex and dynamic nature, according to the principle of unsupervised learning, this paper also constructs the flow classifier based on the clustering results, and it can be used to determine the categories of other traffic flows. Ours is a work-in-progress. Preliminary results indicate that clustering is indeed a useful technique for traffic identification. Our goal is to build an efficient and accurate classification tool using clustering techniques as

the building block. Such a clustering tool would consist of two stages: A model building stage and a classification stage. In the first stage, an unsupervised clustering algorithm clusters training data. This produces a set of clusters that are then labeled to become our classification model. In the second stage, this model is used to develop a classifier that has the ability to label both online and offline network traffic. We note that offline classification is relatively easier compared to online classification, as flow statistics needed by the clustering algorithm may be easily obtained in the former case; the latter requires use of estimation techniques for flow statistics. We should also note that this approach is not a panacea, for the traffic classification problem. While the model building phase does automatically generate clusters, we still need to use other techniques to label the clusters (e.g., payload analysis, manual classification, port-based analysis, or a combination thereof). This task is manageable because the model would typically be built using small data sets.

The remainder of this paper is structured as follows. The different Internet traffic classification methods including those using cluster analyses are reviewed in Section 2. Section 3 outlines the theory and methods employed by the clustering algorithms studied in this paper. Section 4 and Section 5 present our methodology and outline our experimental results, respectively. Section 6 discusses the experimental results. Section 7 presents our conclusions.

2. RELATED WORKS

Currently, there are several traffic classification techniques. One of the most used techniques [3] is payload identification. It uses signature of traffic, which can be simple strings or complex regular expressions; one or more signatures are used for each application. Another interesting method for traffic identification is based on statistical properties. Such methods assume that the statistical properties of traffic are unique for different applications and can be used to distinguish applications from each other. The commonly used statistical features, for example, include flow duration, packet inter-arrival time, packet size, etc., The method normally adopts machine learning which can be divided into supervised learning and unsupervised learning technologies. Moore *et al.* [4] use a supervised machine learning algorithm called Naive Bayes as a classifier. Moore *et al*. showed that the Naive Bayes approach has a high accuracy classifying traffic. Supervised learning requires the training data to be labeled before the model is built. We believe that an unsupervised clustering approach offers some advantages over supervised learning approaches. One of the main benefits is that new applications can be identified by examining the connections that are grouped to form a new cluster. The supervised approach cannot discover new applications and can only classify traffic for which it has labeled training data. Another advantage occurs when the connections are being labeled. Due to the high accuracy of our clusters, only a few of the connections need to be identified in order to label the cluster with a high degree of confidence. Also consider the case where the data set being clustered contains encrypted P2P connections or other types of encrypted traffic. Karagiannis *et al*. proposed a technique that uses the unique behaviors of P2P applications when they are transferring data or making connections to identify this traffic [5]. Their results show that this approach is comparable with that of payload-based identification in terms of accuracy. More recently, Karagiannis *et al*. developed another method that uses the social, functional, and application behaviors to identify all types of traffic [2]. This approach focuses on higher level behaviors such as the number of concurrent connections to an IP address and does not use the transport layer characteristics of single connection that we utilize in this paper. Bermolen *et al.* [6] uses support vector machines, accurately identifies P2P-TVtraffic as well as traffic that is generated by other kinds of applications. Keralapura *et al.* [7] proposed a novel two-stage p2p traffic classifier, called Self-Learning Traffic Classifier (SLTC), which can accurately identify p2p traffic in high-speed networks. Xu *et al.* [8] proposed a novel approach to identify P2P traffic by leveraging the data transfer behavior of P2P applications. Molnar *et al*. [9] proposed a new method to identify skype traffic. Mcgregor *et al*. [10] have explored using the EM (expectation maximization) algorithm to break the traffic trace down into clusters with different characteristics. Zander *et al.* [11] have used AutoClass (based on EM algorithm) for traffic clustering, and the clusters have been transformed into classifiers. Erman *et al.* [12] have showed that the AutoClass approach could achieve higher accuracy than the supervised Naive Bayes method, and the clustering approach also had the advantage of discovering previously unknown applications. Erman *et al.* [13] and Erman *et al.* [14] have further compared three clustering algorithms and proposed a hybrid approach called semi-supervised learning.

3. FEATURE METRIC

Moore *et al.* [15] collected 249 kinds of attributes of the network flow by measuring it directly, while many attributes were interrelated, leading to large quantities of computation and low detection accuracy. So in this paper, we define some flow metrics as shown from Table 1. This metric feature is composed of network behavior characteristic with label which is labeled by l7 filter software. And, we extract the unidirectional flow from packet and change the unidirectional flow into bi-directional flow. So, flow metrics of Table 1 are all composed of bi-directional flow characteristic. Many flow metrics directly get from NETFLOW, for

example metric tcpflags and tos. The other metrics are statistics information of packets. These features allow for discrimination between the different traffic classes.

4. THE SA-K-MEANS ALGORITHM

Jain *et al*. [16] proposed that there are a variety of partition-based clustering algorithms available. The K-Means algorithm partitions objects in a data set into a fixed number of K disjoint subsets. For each cluster, the partitioning algorithm maximizes the homogeneity within the cluster by minimizing the square-error. The formula for the square error is:

$$E = \sum_{i=1}^{K} \sum_{j=1}^{n} |dist(x_j, c_i)|^2$$

The square error is calculated as the distance squared between each object x and the center (or mean) of its cluster. Object c represents the respective center of each cluster. The square error is minimized by K-Means using the following algorithm. The centers of the K clusters are initially chosen randomly from within the subspace. The objects in the data set are then partitioned into the nearest cluster. K-Means iteratively computes the new centers of the clusters that are formed and then repartitions them based on the new centers. The K-Means algorithm continues this process until the membership within the clusters stabilizes, thus producing the final partitioning. The basic purpose of the K-means algorithm is to find the K division which can minimize objective function value, the cluster algorithm idea is simple, it is easy to implement, and the convergence speed is faster. Rational choice of the cluster center and the threshold will get the correct clustering results as shown in Figure 1. Figure 1a shows that the data are correctly divided into four clusters. However, when cluster center threshold is increased, the data are wrongly divided into two clusters in Figure 1b, at last Figure 1c shows that the whole data is one cluster. If the apparent differences between clusters, and dense data distribution, the algorithm is more effective, but if each cluster shape and size is not very different, it may appear larger cluster segmentation. In addition, the K-means algorithm for clustering; the optimal clustering results by the extreme points correspond to the objective function; the objective function may exist many local minima points, this will lead to algorithms converge at local minimum points. Therefore, the initial cluster centers were randomly selected and may cause the solution into a local optimal solution; it is difficult to obtain the global optimal solution. The main limitations of the algorithm in the following aspects: (1) The final clustering result depends in the first division. (2) First, number of clusters M should be known beforehand. (3) Cluster size is sensitive to the noise and isolated point. (4) The algorithm often makes clusters result to partial optimization. (5) It is not suitable for the cluster of non-convex shape or small difference between clusters.

Table 1: Predominant feature used to describe

Feature	Feature description
lport	Low port number
hport	High port number
during	Flow during
transproto	Transport protocol used (TCP/UDP)
TCP flags 1	TCP header flag, or (OR), transport layer protocol is UDP, the feature is 0
TCP flags 2	TCP header flag, or (OR), transport layer protocol is UDP, the feature is 0
pps	Packets/duration
bps	Bytes/duration
Mean packets arrived time	Duration/packets
Biodirection Packets ratio	Forward packets/backward packets
Biodirection Bytes ratio	Forward bytes/backward bytes
Biodirection Packet length ratio	Biodirection packets length ratio
Biodirection packets	Forward packets+backward packets
Biodirection bytes	Forward bytes+backward bytes
tos	Biodirection TOS OR from NETFLOW
Mean packet length	Biodirection bytes/Biodirection packets

The simulated annealing algorithm is a heuristic random search algorithm; parallel and asymptotic convergence has been proved in theory that it is a probability 1, converge to the global optimal solution of global optimization algorithms, with simulated annealing algorithm K-means clustering algorithm to optimize the limitations of K-means clustering algorithm can be improved to improve the performance of the algorithm. Based on the simulated annealing-improved

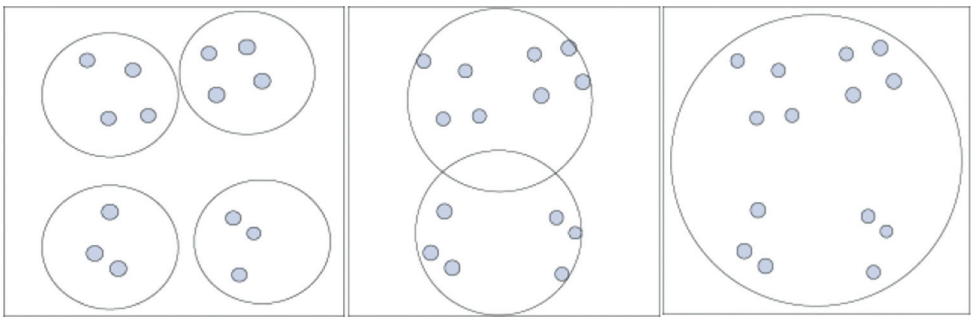

Figure 1: Cluster process (a) Correct cluster; (b) Bigger threshold cause bad cluster and (c) Too large threshold cause bad cluster.

K-means clustering algorithm, the internal energy E is considered as the objective function value and the cluster results based on K-means clustering algorithm will be considered as the initial solution; the initial objective function value is considered as the initial temperature T_0. The method process is an iterative process (generate a new solution → calculating the objective function → accept or discard the new solution), and gradually reduce the T value, the algorithm terminates when the current solution is the approximate optimal solution. The beginning of this algorithm is faster to find a relatively optimal areas and then through more precise search and eventually find the global optimal solution. The objective function selects the dispersion in the current clustering by general category as the objective function, such as

$$J_w = \sum_{i=1}^{M} \sum_{x \in (w_i)} d(X, \overline{X(w_i)}) \quad (1)$$

The initial temperature: Under normal circumstances, in order to initially produce a new solution, which is accepted, the algorithm should reach quasi-equilibrium at the beginning. So the basic K-means clustering algorithm clustering results are considered as the initial solution, the initial temperature $T_0 = J_w$.

Perturbation method: The generation of new solutions is a result of disturbance in the current solution. The algorithm uses a random perturbation method, i.e., immediately change a cluster sample of the current category, thereby creating a new category, so that the algorithm may jump out of local minima. To summarize,the SA-kmeans algorithm is given in Algorithm 1.

5. EVALUATIONS

In this paper, we use the routine evaluation standard for verifying the effectiveness of our classification algorithm. The effectiveness of the current flow identification algorithm has the following three evaluation criteria. The classification capabilities of the model will be estimated for unknown data sets based on the experimental results for test data sets. If the classification model M has been established, together with the test data set $f=\{f_1, f_2., f_n\}$ and the class attributes collection $l=\{l_1, l_2., l_n\}$, where the network flows f corresponds to the nth class of network application, the corresponding confusion matrix can take the form shown in Table 2.

Algorithm 1: SA-kmeans algorithm

```
1.  //Preprocessing stage
2.  use kmeans cluster network flow as w;
3.  T0←kmeanscluster; Jw←f (w);
4.  a=0:99;K←0;
5.  T0=Jw and initialize Annealing speed a and max Annealing cycles;
6.  generate new cluster w';
7.  compute new object function Jw';
8.  while Jw'!=optimal object function do
9.  ΔJ=Jw-Jw';
10. if then ΔJ<0
11. w←w'
12. if ΔJ≥0 then if (p (w, w', T) > random() then
13. w←w;
14. k=k + 1;
15. end while;
16. return w;
```

Where, c_{ij} is the number of the instances that truly have type i among all those classified as type j by the classification model. Obviously, the larger is the values of the diagonal elements of the confusion matrix; the better is the classification accuracy of the model. The following are some evaluation metrics that are used for the study, and the concepts involved are as follows:

- FP (false positive): The flows not in A are misclassified as A. For example, a non-P2P flow is misclassified as a P2P flow. FP will produce false warnings for the classification system
- FN (false negative): The flows in A are misclassified as some other category. For example, a true P2P flow is not identified as P2P. FN will result in identification accuracy loss.

The calculating methods are as follows:
Precision:
The percentage of samples classified as A that are really in class A

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

Recall:
The percentage of samples in class A that are correctly classified as A

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

Overall accuracy:
The percentage of samples that are correctly classified

$$Overall\ accuracy = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + FN_i)} \quad (4)$$

Table 2: Confusion matrix of *n* class

	Class 1	Class 2	…	Class *n*
Class1	C_{11}	C_{12}	…	C_{1n}
Class2	C_{21}	C_{22}	…	C_{2n}
⋮	⋮	⋮	⋮	⋮
class *n*	C_{n1}	Cn_2	…	C_{nn}

6. EXPERIMENTAL RESULTS

6.1 Dataset

In order to validate the method and analyze the impact factor, we adopt NOC_SET as dataset. As shown from Table 3, we collected data at southeast university, and use ourselves l7_filter_modify software to label the flow. L7_filter_modify is developed based on L7filter [17]. At last, we generated NOC_SET dataset. A basic requirement of traffic classification is that the flow types are correctly identified. Table 1 also shows the frequently used application classes of the data sets used in this study. An application class may contain different kinds of data, for example, the class Mail includes IMAP, SMTP, and POP3. TCP/IP traffic flows are the fundamental objects for classification, which is represented as a flow of one or more packets between two hosts of a network using network communication protocols. The flow is clarified by the IP five-tuple consisting of the source-IP, destination-IP, source-port, destination-port, and the protocol type. In order to focus on the traffic classification process itself, the semantically complete TCP connections are selected to make up the training sets and testing sets, where semantically complete TCP flow is defined as: A bi-directional flow for which one can observe the complete connection set-up (SYN-ACK) and another complete connection tear-down (FIN-ACK).

6.2 Impact of K

The K-Means algorithm has an input parameter of K. This input parameter, as mentioned in Section 4, is the number of disjoint partitions used by K-Means. In our data sets, we would expect that there would be at least one cluster for each traffic class. In addition, due to the diversity of the traffic in some classes such as HTTP (e.g., browsing, bulk download, streaming), we would expect even more clusters to be formed. Therefore, based on this, in Figure 2, the K-Means algorithm was evaluated with K initially being 10 and K being incremented by 10 for each subsequent clustering.

Table 3: NOC SET dataset

AppID	Application	Protocol	Flow number
1	WWW	HTTP, https, etc	904572
2	Bulk	FTP	5483
3	Mail	Pop3, Imap, Smtp	385
4	P2P	BitTorrent, eDonkey, Xunlei, etc	11186
5	Service	DNS, NTP	3035
6	Interactive	SSH, CVS, pcAnywhere, etc	6
7	Multimedia	RTSP, Real, etc	20
8	Voice	SIP, Skype, etc	276
9	Others	Games, attacks, etc	26500

The minimum, maximum, and average results for the K-Means clustering algorithm are shown in Figure 1. Initially, when the number of clusters is small the overall accuracy of K-Means is approximately 49% for the Auckland IV data sets and 67% for the Calgary data sets. The overall accuracy steadily improves as the number of clusters increases. This continues until K is around 100 with the overall accuracy being 74% and 76% on average, for the NOCSET data sets, respectively. At this point, the improvement is much more gradual with the overall accuracy only improving by an additional 1.0% when K is 120 in both data sets. When K is greater than 120, the improvement is further diminished with the overall accuracy improving to the high 78% range when K is 150. However, the overall accuracy of improved kmeans method (Sa-kmeans) has minimal change with the increasing of values of K.

6.3 Cluster Weights

For the traffic classification problem, the number of clusters produced by a clustering algorithm is an important consideration. The reason being that once the clustering is complete, each of the clusters must be labeled. Minimizing the number of clusters is also cost effective during the classification stage. One way of reducing the number of clusters to label is by evaluating the clusters with many connections in them. For example, if a clustering algorithm with high accuracy places the majority of the connections in a small subset of the clusters, then by analyzing only this subset a majority of the connections can be classified. Figure 3 shows the percentage of connections represented as the percentage of clusters increases, using the NOCSET data sets. The 12 largest clusters produced by k-means only contain 50% of the connections. In contrast, for the SA-Kmeans the 7 largest clusters contain over 50% of the connections. It will show the SA-Kmeans has good cluster weights.

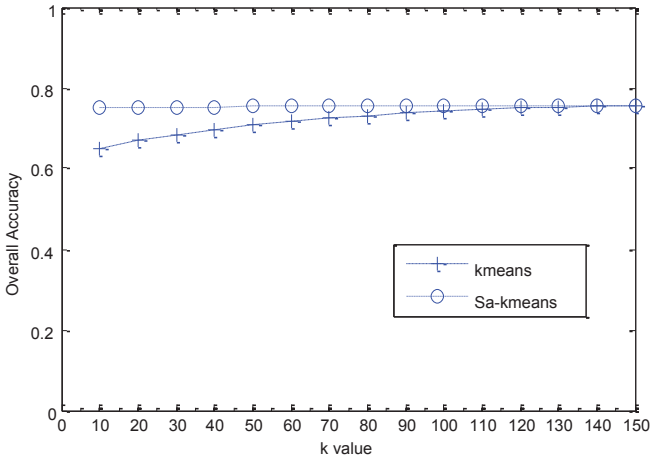

Figure 2: K value impact on overall accuracy.

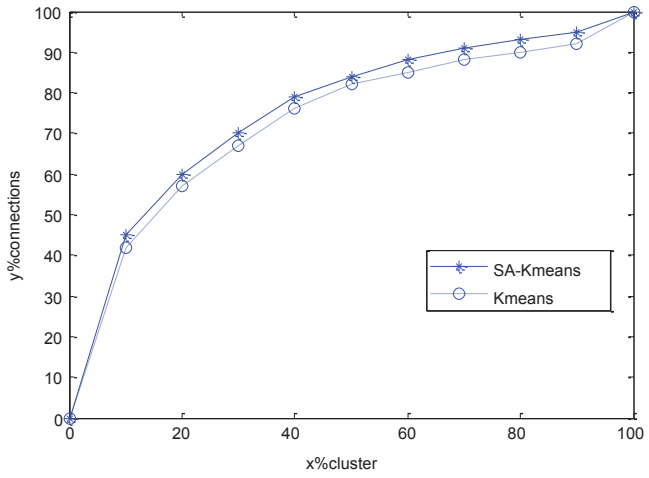

Figure 3: CDF of cluster weights.

6.4 Impact of Cluster Distance Method

Distance calculation methods used by different clustering results below for three different distance calculation methods were analyzed and compared. (1) Euclidean distance, (2) The cosine distance, and (3) Tanimoto measure method.

In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points that one would measure with a ruler, and is given by the Pythagorean formula. By using this formula as distance, Euclidean space (or even any inner product space) becomes a metric space. The associated norm is called the Euclidean norm. Older literature refers to the metric as Pythagorean metric.

The Euclidean distance between points p and q is the length of the line segment connecting them $(\overline{pq})$.

In Cartesian coordinates, if p = $(p_1, p_2..., p_n)$ and q = $(q_1, q_2..., q_n)$ are two points in Euclidean n-space, then the distance from p to q, or from q to p is given by:

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \ldots + (q_n - p_n)^2}$$

$$= d(q, p) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

Angular similarity is often used to compare documents in text mining. In addition, [18] it is used to measure cohesion within clusters in the field of data mining.

The cosine of two vectors can be easily derived by using the Euclidean dot product formula:

$$a.b = \| a \| \| b \| \cos \theta$$

Given two vectors of attributes, A and B, the cosine similarity, θ, is represented using a dot product and magnitude as

$$angle = \cos(\theta) = \frac{A \bullet B}{\| A \| \| B \|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

The resulting similarity ranges from -1 meaning exactly opposite, to 1 meaning exactly the same, with 0 usually indicating independence, and in-between values indicating intermediate similarity or dissimilarity.

For text matching, the attribute vectors A and B are usually the term frequency vectors of the documents. The cosine similarity can be seen as a method of normalizing document length during comparison.

In the case of information retrieval, the cosine similarity of two documents will range from 0 to 1, since the term frequencies (tf-idf weights) cannot be negative. The angle between two term frequency vectors cannot be greater than 90°.

Tanimoto Distance is often referred to, erroneously, as a synonym for Jaccard Distance ($1-T_s$). This function is a proper distance metric. "Tanimoto Distance" is often stated as being a proper distance metric, probably because of its confusion with Jaccard Distance.

If Jaccard or [19] Tanimoto Similarity is expressed over a bit vector, then it can be written as

$$f(A, B) = \frac{A \bullet B}{|A|^2 + |B|^2 - A \bullet B}$$

where the same calculation is expressed in terms of vector scalar product and magnitude. This representation relies on the fact that for a bit vector (where the value of each dimension is either 0 or 1) then

$$A \bullet B = \sum_i (A_i \wedge B_i) \; and \; |A|^2 = \sum_i (A_i).$$

k-means algorithm with different cluster distance method is given in Algorithm 2. From Figures 4 and 5, we can see k-means-angle cluster method have best precision on every type.

Figure 6 shows that different k values have impacted the accuracy classification, and with the increasing of K, overall accuracy also is increasing, and k-means-angle method is most accurate compared with others. When k > 50, k-means-angle is the same to the k-means-ED.

7. DISCUSSION

By the above section study, we can see that SA-k-means

Algorithm 2: SA-kmeans with different cluster distance algorithm

```
1.  //Preprocessing stage
2.  select k as cluster center from n;
3.  for n−k do
4.  for disttype ∈ X1 do
5.  x=distED (n−k; k);
6.  for disttype ∈ X2 do
7.  x=distangle (n−k; k);
8.  for disttype ∈ X3 do
9.  x=disttanimoto (n−k; k);
10. if then x < σ
11. wx→w;
12. use kmeans cluster network flow as w;
13. T0←kmeanscluster; Jw←f (w);
14. a=0:99;K←0;
15. T0=Jw and initialize Annealing speed a and max Annealing cycles;
16. generate new cluster w′;
17. compute new object function Jw′;
18. while Jw′!=optimal object function do
19. ΔJ=Jw – Jw′;
20. if then ΔJ<0
21. w←w′
22. if ΔJ≥0 then if (p (w, w′, T) > random() then
23. w←w′;
24. k=k + 1;
25. end while;
26. return w;
```

algorithm can overcome the defects of k-means method which choose deference center k value to get bad cluster results, while k-means method has many distance methods. From the above experiment, we can conclude that angular seminary method will get better cluster results. So this section will mainly discuss SA-k-means with angular seminary. And according to the impact factor of packet sampling, deeply analyze influence of the sampling ratio on cluster results. And the experiment shows that the sampling ratio is related to sample number. When the sample number is larger, change of cluster accuracy is obviously.

8. CONCLUSION

From our theoretical analysis and experimental results, we conclude that ED and angle are similar when applied to high-dimensional k-means queries. For normalized data and clustered data, ED and angle becomes even more similar. And we proposed the SA-K-means method which can overcome the partial optimization and cannot influence on k value; meanwhile, SA-k-means have better classification accuracy compared with k-means method.

9. ACKNOWLEDGMENTS

This paper is supported by National 973 Plan Projects (2009CB320505) and National Science and Technology Plan Projects (2008BAH37B04).

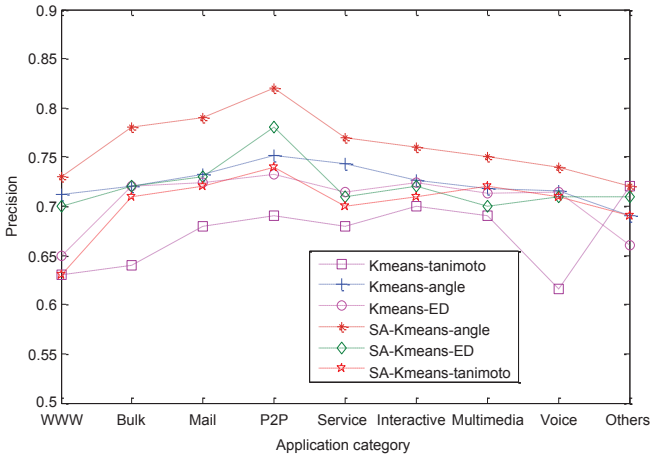

Figure 4: K-means method's precision with cluster distance method.

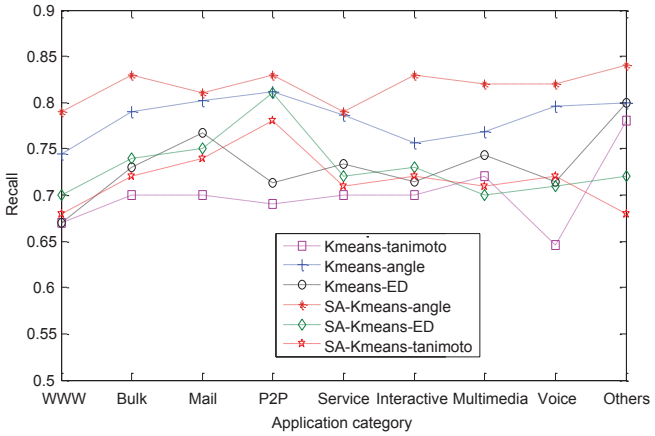

Figure 5: K-means method's recallwith cluster distance method.

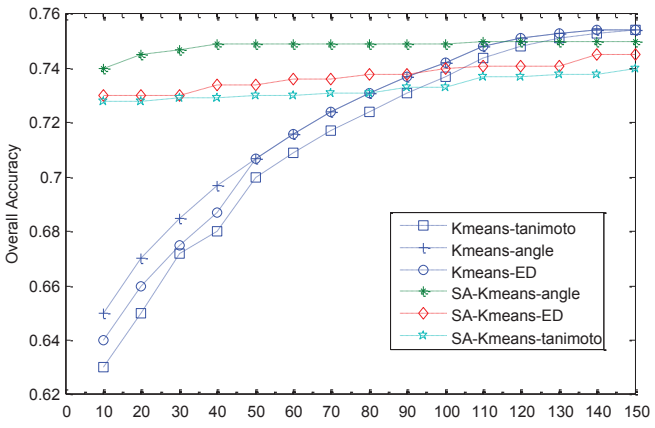

Figure 6: Cluster distance method of SA-K-means and K-means.

REFERENCES

1. Available from: http://www.iana.org/assignments/port-numbers

[Last accessed on 2011].

2. T Karagiannis, K Papagiannaki, and M Faloutsos, "Blinc: Multilevel traffic classification in the dark," In: *ACM SIGCOMM Computer Communication Review*, Vol. 35, pp. 229-40, 2005.
3. Y Wang, Y Xiang, W Zhou, and S Yu, "Generating regular expression signatures for network traffic classification in trusted network management," *Journal of Network and Computer Applications*, Vol. 35, pp. 992-1000, 2012.
4. A Moore, and D Zuev, "Internet traffic classification using bayesian analysis techniques," In: *ACM SIGMETRICS Performance Evaluation Review*, Vol. 33, pp. 50-60, 2005.
5. T Karagiannis, A Broido, M Faloutsos, "Transport layer identification of p2p traffic," In: *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pp. 121-34, 2004.
6. P Bermolen, M Mellia, M Meo, D Rossi, and S Valenti, "Abacus: Accurate behavioral classification of p2p-tv traffic," *Computer Networks*, Vol. 55, pp. 1394-11, 2011.
7. R Keralapura, A Nucci, and C Chuah, "A novel self-learning architecture for p2p traffic classification in high speed networks," *Computer Networks*, Vol. 54, pp. 1055-68, 2010.
8. K Xu, M Zhang, M Ye, D Chiu, and J Wu, "Identify p2p traffic by inspecting data transfer behavior," *Computer Communications,* Vol. 33, pp. 1141-50, 2010.
9. S Moln'ar, and M Per'enyi, "On the identification and analysis of skype traffic," *International Journal of Communication Systems*, Vol. 24, pp. 94-117, 2011.
10. A McGregor, M Hall, P Lorier, and J Brunskill, "Flow clustering using machine learning techniques," *Passive and Active Network Measurement,* pp. 205-14, 2004.
11. S Zander, T Nguyen, and G Armitage, "Automated traffic classification and application identification using machine learning," *In: Local Computer Networks, 30th Anniversary. The IEEE Conference on,* 2005. pp. 250-7,2005.
12. J Erman, A Mahanti, and M Arlitt, "Qrp05-4: Internet traffic identification using machine learning," In: *Global Telecommunications Conference, 2006. GLOBECOM'06. IEEE*, pp. 1-6, 2006.
13. J Erman, M Arlitt, and A Mahanti, "Traffic classification using clustering algorithms," In: *Proceedings of the 2006 SIGCOMM workshop on Mining network data*, pp. 281-6, 2006.
14. J Erman, A Mahanti, M Arlitt, I Cohen, and C Williamson, "Semi-supervised network traffic classification," In: *ACM SIGMETRICS Performance Evaluation Review*, Vol. 35, pp. 369-70, 2007.
15. A Moore, and K Papagiannaki, "Toward the accurate identification of network applications," *Passive and Active Network Measurement*, pp. 41-54, 2005.
16. A Jain, and R Dubes, "Algorithms for clustering data," New jersey, USA: Prentice-Hall, Inc.; 1988.
17. J Levandoski, E Sommer, and M Strait, "Application layer packet classifier for linux," 2008. Available from: http://www.l7-filter.sourceforge.net.
18. P Tan, M Steinbach, and V Kumar, "Introduction to data mining," India: Pearson Addison Wesley Boston; 2006.
19. D Rogers, and T Tanimoto, "A computer program for classifying plants," *Science* Vol. 132, no. 3434, pp. 1115, 1960.

AUTHORS

Shi Dong is a Ph.D. candidate in school of computer science and engineering at Southeast University. His major research interests include network security, network management, and network measurement.

E-mail: njbsok@gmail.com

Dingding Zhou is a lecturer of Zhoukou Normal University. Her major research interests include network management and network measurement.

E-mail: zdd@zknu.edu.cn

Wei Ding received B.S degree in the computer soft from Nanjing University in 1982 respectively, she received M.S, Ph.D degree from Southeast University, in 1987, 1995. Nowadays she is a professor in Southeast University. Her major research interests are in high speed communications, network management, and network security.

E-mail: wding@njnet.edu.cn

Jiam Gong received B.S degree in the Computer soft from Nanjing University in 1982. He received Ph.D degree from Southeast University in 1996. Nowadays he is a professor in Southeast University, and his major interests are network management, network security.

E-mail: jgong@njnet.edu.cn

DOI: 10.4103/0377-2063.118021; Paper No JR 621_12;