基于复杂网络社团划分的网络流量分类

蔡 君[1,2] 余顺争[1]

(中山大学电子与通信工程系 广州 510275)[1] (广东技术师范学院电子与信息学院 广州 510665)[2]

摘 要 随着网络的高速发展以及各种应用的不断涌现,采用端口号映射或有效负载分析的方法进行流量分类与应用识别已难以满足应用的需求。以流为网络节点、流之间统计特征的相似度为边,构建流相关网络模型,利用Newman快速社团划分算法(NFCD)对流相关网络模型进行社团划分,得到了流的聚类结果,实现了网络流量的分类,并与先前的两种无监督的流量分类算法(K-Means,DBSCAN)进行了对比。实验结果显示,利用NFCD算法具有更高的准确率,并能产生更好的聚类效果,且不受输入参数影响。

关键词 流量分类,无监督聚类,社团划分,复杂网络

中图法分类号 TN919.26 **文献标识码** A

Internet Traffic Classification Based on Detecting Community Structure in Complex Network

CAI Jun[1,2] YU Shun-zheng[1]

(Department of Electronic and Communication Engineering, Sun Yat-Sen University, Guangzhou 510275, China)[1]

(School of Electronic and Information, Guangdong Polytechnic Normal University, Guangzhou 510665, China)[2]

Abstract In recent years, Internet traffic classification using port-based or payload-based methods is becoming increasingly difficult with peer-to-peer(P2P) applications using dynamic port numbers, masquerading techniques, and encryption to avoid detection. Because supervised clustering algorithm needs accuracy of training sets and it can not classify unknown application, we introduced complex network's community detecting algorithm, a new unsupervised classify algorithm, which has previously not been used for network traffic classification. We evaluated this algorithm and compared it with the previously used unsupervised K-means and DBSCAN algorithm, using empirical Internet traces. The experiment results show complex network's community detecting algorithm works very well in accuracy and produces better clusters, besides, complex network's community detecting algorithm need not know the number of the traffic application beforehand.

Keywords Traffic classification, Unsupervised clustering, Community detecting algorithm, Complex network

1 引言

网络流量分类(Network Traffic Classification)是指在基于TCP/IP协议的互联网(Internet)中,按照网络的应用类型(如HTTP,DNS,FTP,P2P等)将网络通信产生的双向TCP流和UDP流进行分类。它有助于网络研究人员和网络管理者进行容量规划、趋势分析、服务质量控制(QoS)管理以及安全检测。近年来,随着互联网技术的不断发展,网络应用的快速增长和变化给网络流量分类带来了一系列的挑战。很多新的网络服务(如P2P,在线游戏)采用动态端口、协议加密以及其他技术,使传统的基于IANA(Internet assigned numbers authority)定义的协议端口[1]和深层数据包探测技术(DPI, deep packet inspection)的分类方法[2]已不能保证进行正确的网络流量分类和统计。使用机器学习的方法统计流量的特征,对网络上的流量进行分类,成为当前流量测量领域内一个新兴的研究热点。目前,应用于流量分类的机器学习方法有分类(有监督的学习方法)和聚类(无监督的学习方法)。有监督的分类模型构造方法有贝叶斯[3]、决策树[4]、关联规则学习[5]、神经网络[6]等。所有有监督的学习方法是在已知类别的网络流量中进行训练,根据已有的准确类别来判断其分类的准确性。这些方法无法发现新的应用模式,只能在训练集已有的应用类型的基础上对未知的流量进行分类,并且稳定性不高,需经常重复训练模型。而无监督的方法克服了有监督的缺点,它只根据网络流量的相似程度划分成不同的应用,无需准确的训练集,可以发现新的网络应用,典型的方法是2006年加拿大卡尔加里大学的Erman等人引入的K-means和DBSCAN的聚类方法[7]。由于这两种方法都需要提前指定某些参数的值,当选取不当时,算法的性能会大幅度地下降。本文以流为节点、流之间统计特征的相似度为边,构建流相关网络模型,然后利用Newman快速社团划分算法(NF-

到稿日期:2010-04-20 返修日期:2010-07-19 本文受国家高技术研究发展计划(863)(2007AA01Z449),国家自然科学基金(60970146)和国家自然科学基金-广东联合基金重点项目(U0735002)资助。

蔡 君(1981-),男,博士生,主要研究方向为计算机网络安全、复杂网络,E-mail:gzhcaijun@gmail.com;余顺争(1958-),男,教授,博士生导师,主要研究方向为网络安全、网络行为分析、网络测量等。

CD)对流相关网络模型进行社团划分，得到对应于不同应用的社团。实验结果与先前的两种无监督的流量分类算法(K-Means,DBSCAN)进行的对比显示，利用 NFCD 算法进行流量分类具有更高的准确率和能产生更好的聚类效果，且无需提前输入参数。

本文第 2 节根据流之间统计特征的相似度构建流相关网络模型；第 3 节介绍本文应用于流量分类的复杂网络社团划分算法(NFCD)；第 4 节通过实验结果的分析论证了本文所提方法的有效性；最后总结了全文的工作。

2 流相关网络模型

在数学上网络 $G=(V,E)$ 是指由一个点集 $V(G)$ 和一个边集 $E(G)$ 组成的一个图，需要从网络流量中的流抽象出网络中的点、边对应关系。本文中将网络流量中的流(由分组首部的五元组(源地址，目的地址，源端口，目的端口，协议)唯一确定)映射为节点 V_i，所有节点构成节点集 $V(G)$。边的映射方法如下：

1) 计算流之间的相关度，采用皮尔逊(Pearson)相关系数计算[8]：

$$r=\frac{\sum_{i=1}^{n}(X_i-\overline{X})(Y_i-\overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\overline{X})^2\sum_{i=1}^{n}(Y_i-\overline{Y})^2}} \tag{1}$$

式中，n 为样本量，$X_i(Y_i)$ 和 $\overline{X}(\overline{Y})$ 分别为流的统计特征的观测值和均值。r 描述的是两个变量间线性相关强弱的程度，r 的取值在 -1 与 $+1$ 之间，若 $r>0$，表明两个变量是正相关，即一个变量的值越大，另一个变量的值也会越大；若 $r<0$，表明两个变量是负相关，即一个变量的值越大另一个变量的值反而会越小。r 的绝对值越大表明相关性越强。若 $r=0$，表明两个变量间不是线性相关。流之间的相关性用流相关矩阵 FSA 存储，矩阵元素 S_{ij} 表示流 i 与流 j 之间的相关程度。

2) 我们将 r 值转换为下列形式：

$$r_1=\sqrt{(m-2)r^2/1-r^2} \tag{2}$$

式中，r 表示两个流之间的皮尔逊相关性，m 表示计算流之间相关性所用的统计特征的个数。转换后的 r_1 值服从自由度为 $n-2$ 的 t 分布。可以用 t 统计量对总体相关系数为 0 的原假设进行检验。若 t 检验显著，则拒绝原假设，即两个变量是线性相关的，在对应的流相关网络模型中，连接这两个流之间的边被赋予权重“1”，表示这条边出现；若 t 检验不显著，则不能拒绝原假设，即两个变量不是线性相关的，在对应的流相关网络模型中，连接这两个流之间的边被赋予权重“0”，表示这条边不出现。即：如果两个流之间的 r_1 值大于预设的 p-value 对应的 t 分布值，连接这两个流之间的边被赋予权重“1”，否则为“0”。在对所有数据集的实验中，我们设置 p 值均为 10^{-5}。流相关网络模型使用邻接矩阵 FRA 存储，矩阵元素 R_{ij} 为流 i 与流 j 之间边的连接关系。

3 流聚类算法

根据复杂网络社团划分算法，对流相关网络模型进行社团划分，可以将网络流根据应用进行聚类。Newman 等人基于同类匹配(assortative Mixing)[9]定义了一个衡量网络社团划分质量的标准——模块度(Modularity)。定义如下[10]：

$$Q=\sum_r(e_{rr}-a_r^2)=Tre-\|e^2\| \tag{3}$$

考虑某种社团划分形式，将网络划分为 k 个社团。定义一个 $k\times k$ 维的对称矩阵 $e=(e_{ij})$，其中元素 e_{ij} 表示网络中连接两个不同社团的节点的边在所有边中所占的比例，这两个节点分别位于第 i 个社团和第 j 个社团。矩阵中对角线上各元素之和为 $Tre=\sum_i e_{ii}$，它给出了网络中连接某一个社团内部各节点的边在所有边的数目中占的比例。每行中各元素之和为 $a_i=\sum_j e_{ij}$，它表示与第 i 个社团中的节点相连的边在所有边中所占的比例。式(1)的物理意义是：网络中社团内实际连接数目与随机连接情况下社团内的期望连接数目之差。Q 的上限为 1，Q 越接近 1，说明社团结构越明显，划分质量越好。Newman 在式(1)的基础上提出了一种快速算法，它可以用于分析节点数达 100 万的复杂网络。这种快速算法实际上基于贪婪算法思想的一种凝聚算法，算法如下：

①初始化网络为 n 个社团，即每个节点就是一个独立社团。初始的 e_{ij} 和 a_i 满足：

$$e_{ij}=\begin{cases}1/2m, & \text{如果节点 } i \text{ 和 } j \text{ 之间有边相连}\\ 0, & \text{其它}\end{cases}$$

$$a_i=k_i/2m$$

式中，k_i 为节点的度，m 为网络中总的边数。

②依次合并有边相连的社团对，并计算合并后的模块度增量：

$$\Delta Q=e_{ij}+e_{ji}-2a_ia_j=2(e_{ij}-a_ia_j)$$

根据贪婪算法的原理，每次合并应该沿着使 Q 值增大或者减少的方向进行。

③重复执行步骤②，不断合并社团，直到整个网络都合并成为一个社团。在这些社团结构中，选择一个对应着局部最大 Q 值的社团划分，就可得到最好的网络社团结构。该算法的总复杂度为 $O((m+n)n)$，对于稀疏矩阵则为 $O(n^2)$，m 和 n 分别为网络中总的边数和节点数。

4 实验

4.1 实验数据

为分析和比较这 3 种算法，本文使用两个不同地点、不同时期采集的实际网络数据集，第一个是 Auckland-VIII 数据集(简称 auck8)[11]。第二个是由本文采集网络流量记录而来，简称 zsdx09。

auck8 数据集：NLANR 测量与网络分析组织的被动测量与分析项目(PAM)提供了许多公开的网络分组 Trace 数据集，用以研究 Internet 的操作、行为和健康度。本文使用其中的 auck8 一周的数据，采自 2003 年 12 月。

Zsdx09：该数据采集于广州市某区教育机构的网络出口，整个网络拥有大约 1600 多台主机。我们捕获了 2009 年 3 月为期 2 个星期经过该网络接口的所有网络流量。

在 auck8 和 zsdx09 数据集中，使用 tcptrace 工具从中找出并提取双向均发送一个分组以上的双向流。对于有状态的 TCP 协议，双向流等价于一个连接；而 UDP 没有显式的结束，故使用通用的 90 秒超时即空闲超过 90 秒认为该流结束。

通常认为发起应用会话的一方为客户端，响应的一方为服务器，故一个双向流可分为客户到服务器(c2s)和服务器到到客户(s2c)两个方向。对于每个流我们考虑了 3 个方面流的统计属性：分组数量相关属性；分组长度相关属性和时间相关属性。应选取最能反应网络应用本质区别的流量特征属

性，本文通过主成份分析[12]（principal component analysis，PCA）方法来选择合适的特征子集。具体的属性说明见表1，包含31项属性。

表1 网络流统计特征属性描述

No.	Abbreviation	Description
1	pkts	Total number of packets between service and client
2	pkts_c2s	Total number of packets from client to server
3	pkts_s2c	Total number of packets from server to client
4	byte	Total number of bytes between service and client
5	bytes_c2s	Total number of bytes from client to server
6	bytes_s2c	Total number of bytes from server to client
7	psavg	Mean packets between service and client
8	psavg_c2s	Mean packets from client to server
9	psavg_s2c	Mean packets from server to client
11	psvar	Standard variance packets between service and client
12	psvar_c2s	Standard variance packets from client to server
13	psvar_s2c	Standard variance packets from server to client
14	psmin	Minimum packets between service and client
15	psmin_c2s	Minimum packets from client to server
16	psmin_s2c	Minimum packets from server to client
17	psmax	Maximum packets between service and client
18	psmax_c2s	Maximum packets from client to server
19	psmax_s2c	Maximum packets from server to client
20	duration	Duration of the flow
21	iptavg	Mean inter-arrival time between service and client
22	iptavg_c2s	Mean inter-arrival time from client to server
23	iptavg_s2c	Mean inter-arrival time from server to client
24	ipvar	Standard variance inter-arrival time between service and client
25	ipvar _c2s	Standard variance inter-arrival time from client to server
26	ipvar _s2c	Standard variance inter-arrival time from server to client
27	ipmin	Minimum inter-arrival time between service and client
28	ipmin _c2s	Minimum inter-arrival time from client to server
29	ipmin _s2c	Minimum inter-arrival time from server to client
30	ipmax	Maximum inter-arrival time between service and client
31	ipmax _c2s	Maximum inter-arrival time from client to server
32	ipmax _s2c	Maximum inter-arrival time from server to client

在auck8数据集中，本文考虑DNS，FTP（control），FTP（data），HTTP，IRC，NNTP，LIMEWIRE（使用Gnutella协议的P2P应用的），POP3和SOCKS应用。由于公开的数据集中只有分组首部信息，无法准确获知真实应用类别。实验中使用IANA定义的端口号进行分类，先前的研究中典型的应用大部分都和端口号对应一致[13]。

在zsdx09trace数据集中，我们采集了分组的全部载荷，采用Moore[14]等人提出的基于特征字段的流量分类方法对数据集进行正确分类。对该数据集我们仅考察了HTTP，DNS，POP3，SMTP，FTP，XunLei和SOCKS应用。

4.2 测试方法

K-means和DBSCAN算法输入的是流相似矩阵FSA存储的数据；NFCD算法输入的是FRA存储的数据。在auck8和zsdx09数据集中，HTTP流占有很大的比例，超过50%。这种不均匀的分布导致不能进行不同应用的均匀测试。为解决这个问题，对auck8数据集，从每种应用中随机抽取1000个样本构成auck8数据子集；对zsdx09数据集，从每种应用中随机抽取2000个样本构成zsdx09数据子集。这种数据处理方法使得测试结果能够证明算法具有聚类所有流的能力，而不是仅局限于HTTP流。另外，为进一步提高测试结果的可信度，每个数据集产生10个不同的数据子集。每个数据子集轮流评估这3种聚类算法，重复实验10次。我们对每个数据集的平均结果进行了分析。

4.3 实验结果与分析

在这一部分，我们分析和比较了K-means，DBSCAN和NFCD 3种算法的整体准确率（Overall Accuracy）和对各个应用类别的精确度（Precision）以及它们的聚类效果。另外，我们还考察了3种算法的效率性能，即聚类时间。

4.3.1 评估策略

聚类模型的整体准确率应用最广，几乎为所有研究人员所采纳，它反映了聚类模型正确预测样本数在预测总数中的比例。精确率（Precision）是针对每一个应用类别分别独立考察，是聚类为一个应用类别的实例当中确实属于该类别的实例的百分比。它们用以下形式描述：

$$OA=\frac{\sum TP \text{ for all clusters}}{\text{total number of flows}} \tag{4}$$

$$precision=\frac{TP}{TP+FP} \tag{5}$$

其中，真正TP（true positive）表示实际类型为某一应用的样本中被聚类算法正确预测的样本数。假正FP（false positive）表示实际类型不是某一应用的样本中被聚类算法误判为该应用的样本数。

4.3.2 算法整体准确率和精确率

K-Means算法以k（最终聚类的个数）为参数，把n个数据对象聚为k个类，在聚类过程中，希望每个类对应一种应用。此外，由于一些应用中的多样性，例如HTTP包括浏览、bulk download、流媒体等，将形成很多的类，因此从$k=10$初始值开始，以10的整数倍逐次增加k值，评估K-Means算法与输入参数k的关系，其测试结果如图1所示。当k值为10时，数据集auck8和zsdx08的整体准确率分别为60%和51%，相对较低，随着k值的增大，其整体准确率得到改善，当k值到达100时，数据集auck8和zsdx08的整体准确率分别达到85%和81%，再增加k，其整体准确率基本保持稳定。实验结果显示，该算法整体准确率不高，并且很大程度上受到输入参数k值的影响。

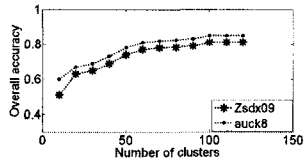

图1 K-Means整体准确率

DBSCAN算法由Ester等人[15]提出。它是利用类的高密度连通性，快速发现任意形状的簇。其基本思想是：对于簇中的每个数据点，在给定的半径（用eps表示）的邻域内包含的数据点数目必须不小于某一给定值（用minPts表示）。因此，该算法有2个输入参数，分别为minPts和eps，其整体准确率结果如图2所示。以auck8数据集进行实验，实验中调整它们的值，取eps为0.03，minPts为12时，该算法的整体准确率取得最大值，到达72.3%。从实验过程中发现，DBSCAN算法对输入参数Eps，minPts极度敏感。

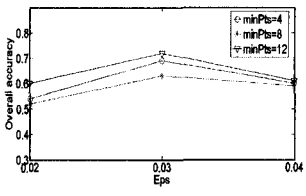

图2 DBSCAN整体准确率

（下转第86页）

[11] Chen Y D, Lu P Z. Constructions of Even-variable Boolean Function with Optimum Algebraic Immunity[EB/OL]. http://eprint.iacr.org/2009/130

[12] Bracken A, Preneel B. On the algebraic immunity of symmetric Boolean functions[C]//Progress in Cryptology-Indocrypt 2005. Berlin: Springer-Verlag, 2005: 35-48

[13] Dalai D K, Maitra S, Sarkar S. Basic Theory in Construction of Boolean Functions with Maximum Possible Annihilator Immunity[J]. Design, Codes and Cryptography, 2006, 40(1): 41-58

[14] Qu L, Li C. On the 2^m-variable symmetric Boolean functions with maximum algebraic immunity[J]. Science in China Information Sciences, 2008, 51(2): 120-127

(上接第 82 页)

NFCD 算法的整体准确率如表 2 所列，对于该算法，聚类结果的数目和聚类参数是自动确定的。从表 2 中的实验结果可以看出，相对于 K-Means 和 DBSCAN 算法，NFCD 具有最高的准确率。对 zsdx09 和 auck8 数据集，算法的平均准确率分别到达 94.6%和 95.7%。

表 2 NFCD 算法的整体准确率(OA)

Data set	Average	Minimum	Maximum
auck8	95.7%	93.2%	97.6%
zsdx09	94.6%	91.3%	98.0%

图 3 给出了使用数据集 zsdx09 时，DBSCAN，K-Means 和 NFCD 算法对各个应用的精确率。对于这几种应用，NFCD 算法在各应用类别中都取得了最佳的结果。

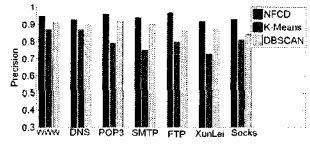

图 3 DBSCAN，K-Means 和 NFCD 的精确率

4.3.3 聚类效果

在处理流量聚类过程中，聚类算法所产生的聚类数量应加以重点考察。因为当聚类完成后，每一个类都要进行标记，聚类的数量越小越容易标记，所以减少聚类数量非常关键。图 4 使用 zsdx09 数据集，表示了流数目的百分比和聚类数目的百分比的关系。在这次聚类过程中 K-Means 算法的输入参数 k 设置为 100。DBSCAN 算法的输入参数设置为 eps=0.03，minPts=12。从图可知，NFCD 相对于 K-Means 和 DBSCAN 算法，可产生更好的聚集效果。对于 NFCD 算法，最大的 6 个社团包含了接近 70%的流。这 6 个社团分别是 HTTP，DNS，PoP3，SMTP，FTP，Socks，其整体准确率也高达 96%。对 auck8 数据集进行类似实验，得到相似的结果。

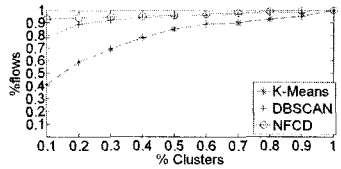

图 4 聚类效果的 CDF

另一个显著的不同就是算法的聚类时间，若使用 auck8 数据集子集进行测试，K-Means 算法是最快的，平均时间不到 56 秒，NFCD 算法用时也较短，平均时间为 72 秒，DBSCAN 算法的时间相对较长，平均为 150 秒。

结束语 网络流量分类与应用识别对于网络管理、安全、研究等非常重要。简单的基于端口号分类的方法逐渐失去效用，基于深度分组的分类技术也存在各种缺点。本文根据不同类别应用产生的网络流量具有独特的统计特征的性质，提出了一种基于复杂网络社团划分算法的无监督的网络流量分类方法，并与先前应用于流量分类的 K-Means 和 DBSCAN 算法进行了比较。通过实际数据集验证了本文提出的方法在准确性和聚类效果上具有明显的优越性。

参 考 文 献

[1] Sen S, Spatscheck O, Wang Dongmei. Accurate, scalable in network identification of p2p traffic using application signature[C]// Proceedings of the 13th international conference on World Wide Web. 2004: 512-521

[2] Haffner P, Sen S, Spatscheck O, et al. ACAS: Automated Construction of Application signatures[C]//Proceedings of the 2005 ACM SIGCOMM Workshop on Mining Network Data. 2005: 197-202

[3] Moore A, Zuev D. Internet traffic classification using Bayesian analysis techniques [C] // ACM International Conference on Measurement and Modeling of Computer Systems (SIGMERTRICS). 2005: 50-60

[4] 徐鹏，林森. 基于 C4.5 决策树的流量分类方法[J]. 软件学报，2009(10): 2692-2704

[5] Williams N, Zander S, Armitage G. A preliminary performance comparison of five machine learning algorithmsfor practical IP traffic flow classification[J]. Special Interest Group on Data Communication Computer Communication Review, 2006: 5-15

[6] Auld T, Moore A W, Gull S F. Bayesian neural networks for Internet traffic classification[J]. IEEE Transactions on Neural Networks, 2007: 223-239

[7] Erman J, Alitt M, Mahanti A. Traffic classification using clustering algorithms[C]// ACM SIGCOMM MineNet. 2006: 281-286

[8] Strehl A, Ghosh J, Mooney R. Impact of similarity measures on web-page clustering [D]. AI for Web Search, 2000

[9] Newman M E J. Mixing patterns in networks[J]. Phys. Rev. E, 2003, 67: 026126

[10] Clauset A, Newman M E J, Moore C. Finding community structure in very large networks[J]. Phys. Rev. E, 2004, 70: 066111

[11] NLANR-Passive Measurement and Analysis[OL]. http://pma.nlanr.net

[12] Iffeit J. Principal component analysis (2^{nd}) [M]. New York: Springer-Verlag, 2003

[13] Zander S, Nguyen T, Armitage G. Automated Traffic Classification and Application Identification Using Machine Learning [C]// The IEEE Conference on Local Computer Networks 30th Anniversary. 2005: 250-257

[14] Moore A W, Papagiannaki K. Toward the accurate identification of network applications[C]//Dovrolis C, ed. Proc. of the PAM 2005. LNCS 3431. Heidelberg: Springer-Verlag, 2005: 41-54

[15] Ester M, Kriege H P, Sander J. A density-based algorithm for discovering clusters in large spatial databases with noise[C]// Proc. of the 2^{nd} International Confabulation Knowledge Discovery and Data Mining Portland. 1996: 226-231