分 类 号 TP391 密级

基于聚类分析的网络流量分类研究

研 究 生 姓 名： 何震凯

指导教师姓名、职称 ： 阳爱民教授

学 科 专 业 ： 计算机应用技术

研 究 方 向 ： 智能信息处理

湖 南 工 业 大 学

二〇〇九年五月三十日

分 类 号 TP391　　密级

基于聚类分析的网络流量分类研究

Research on Network Traffic Classification Based on Clustering Analysis

研 究 生 姓 名：　何震凯

指导教师姓名、职称：　阳爱民教授

学 科 专 业：　计算机应用技术

研 究 方 向：　智能信息处理

论文答辩日期＿＿＿＿＿＿　答辩委员会主席

湖 南 工 业 大 学

二〇〇九年五月三十日

湖南工业大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者签名：何震凯　　日期：2009年 5月 30日

湖南工业大学论文版权使用授权书

本人了解湖南工业大学有关保留、使用学位论文的规定，即：学校有权保留学位论文，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以采用复印、缩印或其他手段保存学位论文；学校可根据国家或湖南省有关部门规定送交学位论文。

作者签名：何震凯 导师签名：阳爱民 日期：2009年 5月 30日

摘　要

随着互联网技术的快速发展，新的应用类型(如FTP、DNS、P2P等)不断涌现，特别是一些采用非标准端口和协议加密形式进行通信的应用的出现，使得传统的基于端口和基于有效载荷的网络流量分类方法效率降低。这激发国内外很多研究者以应用类型作为类别，以网络中通信时所产生的流的统计特性作为特征，运用机器学习来进行网络流量分类研究。这篇论文也是采用机器学习方法研究网络流量分类以及相关技术。包括网络流量数据采集、特征产生、样本标识、特征选择，以及应用机器学习方法对网络流进行分类等技术。

在基于机器学习方法的网络流量分类中，网络流量样本，包括训练样本、测试样本的获取是非常重要的一步。文章首先通过校园网络的中心交换机端口映射方法捕获网络报文，然后将采集到的报文按五元组(源IP地址、源端口号、目的IP地址、目的端口号、协议)解析为流，并统计报文大小、个数、时间、标志位等特征，形成了代表网络流的特征向量。最后结合基于端口、基于有效载荷和协议等多种方法，实现样本的自动标识，形成流样本，采用该方法进行标注正确率高。

在特征选择方面，文章使用基于主成分分析(Principal Component Analysis, 简称PCA)和基于信息增益等两种特征选择方法对两个数据集的候选特征集进行了特征优选，并得到了各自的最优特征子集。实验结果表明提出的方法可以减少特征的数量以便减少学习和分类的时间，同时还可以去掉不相关或冗余特征，提高分类的准确性。

最后，文章应用DBSCAN(*Density Based Spatial Clustering of Application with Noise*)和K-Means两种聚类算法对经过特征选择的网络流进行聚类分析，根据聚类结果产生基于聚类分析的网络流量分类规则，并构建基于聚类分析的网络流量分类器。用实验结果验证了所提出和使用的方法对网络流量分类的有效性和可行性，达到了较高的查准率和总准确度；而且实现简单，算法效率高，是很好的网络流量分类研究方法，具有很强的研究意义和实用价值。

关键词：网络流量分类，特征选择，PCA，信息增益，DBSCAN算法，K-Means算法，聚类分析

ABSTRACT

With the rapid development of Internet technology, many application types of Internet (such as FTP, DNS, P2P, etc.) appeared. The traditional based-port and payload-based methods according to application types become inefficient on network traffic classification because of communications using non-standard port and encrypted protocol. This situation motivated many domestic and foreign researchers to study classify network traffic by machine learning methods. Those methods classify network traffic according application types and use the flow statistical characteristics of applications when they communicate on a network. This thesis is also adopting machine learning methods the related technologies to research network traffic classification. The work include network traffic data collection, generating the statistical features, mark the flow example, feature selection, and classifying application type of network traffic.

In network traffic classification based on the machine learning methods gather the network flows sample (including training example and test example) is very important. Firstly, to capture network packet, the method of port mapping on the center of the campus network switch are used. And analysis those messages in accordance with the five-tuple(source IP address、source port number、source IP address、source port number 、protocol)to flow after they are collected. And then, characteristics (such as the packet size, number, time, sign bit, etc.) of these packets are statistic to generated the feature vector which represents each network flow. Finally, implement auto identifying flow style by port-based, payload-based and protocol methods and form a sample flow.

In network traffic feature selection， two feature selection methods called principal component analysis and information gain are introduced to select feature on candidate feature set of two dataset, and have got their optimized feature subset. The experimental results show that the method can reduce the number of characteristics in order to reduce the learning and classify time, and also can remove irrelevant or redundant features, increase the accuracy of classification.

Finally, the two clustering algorithms DBSCAN (Density-Based

Spatial Clustering of Application with Noise) and K-Means were applied to clustering analysis of the network flow after they have reduced dimension. The Clustering-Based classification rules according to the clustering results are established, and a large number of experiments have done. The experimental results show that these methods which applied to network traffic classification can lead to a higher precision and overall accuracy; and the algorithm of high efficiency and easy implementation, so it is a good method for classification of network traffic. And has a strong research significance and practical value.
Key words: network classification, feature selection, principal component analysis, information gain, DBSCAN algorithm, K-Means algorithm, clustering analysis

目　录

第一章　绪　论

近年来,由于增长惊人的P2P应用对网络带宽的吞噬，利用带宽管理工具来优化网络性能和提供服务质量(*QoS*)保证的需求大幅增加。有研究在最新的文献中表明，*P2P*的流量已经占到网络总流量的50-70%[1、2]。因此，毫不奇怪很多网络管理者千方百计的寻找网络管理工具优化他们的网络。本文针对原有网络流量分类方法的不足，提出本文网络流量分类研究方法。本章主要介绍了研究背景、研究意义以及相关研究工作的优缺点，并在提出了本文的研究方法——基于聚类分析的网络流量分类方法。

1.1 研究背景

最初，网络流量分类通常使用基于端口的网络流量识别方法。这种方法曾经取得过巨大的成功，这是因为传统的应用类型只是根据IANA(*Internet Assigned Numbers Authority*)分配的端口进行通信，因此根据端口号就能确定网络流量的应用类型。随着应用技术的发展，有些应用已不再使用标准的端口进行通信[1、3、4]，而且有些P2P应用使用短暂的端口甚至使用周知端口(如Ftp、Web的端口)伪装流量进行通信，例如KazaA 就可能使用Web的保留端口80进行通信。这就使得基于端口的识别方法的准确度严重削弱，已不能满足网络分类要求。

基于包有效载荷内容(*packet contents*)[5、6、7、8、9、10、11]的检测技术是针对基于端口的识别方法的效率降低而提出的。这种方法根据已知的应用类型报文所包含的特征码来判断应用类型。研究表明，这种方法能够很好的应用于网络流量分类，甚至包括P2P流的分类，因为每种应用类型在网络中通信时可能都有期特有的特征码，只需经过特征码匹配就能正确的对网络流量的应用类型区别出来。实际上，很多商用的带宽管理工具也是使用应用的特征码来增强分类器的鲁棒性。

但是，基于包有效载荷内容的检测方法也有几个局限性。第一，这个技术只能依据获得的特征码来识别流量类型，维护和更新特征码表是一个繁重的任务。同时，一种应用类型可能有多种特征码，而且特征码随时间的推移会发生改变。第二，这种方法进行包的深层检测，可能因为只捕获到少量的有效载荷的字节数不充分而匹配失败，甚至还会涉及到隐私问题。而且这种方法要求在带宽管理工具上占用较多的存储空间，大量的处理工作，以至于开销过大。第三，包检测技术不能应用于加密的应用。有些应用类型使用了协议加密技术，采用这种技术就是为了网络管理人员不能收集应用类型的特征码。

由于基于端口号和基于有效载荷的网络流量分类技术的效率降低，激发人们

去探寻一种仅使用应用类型在通信时所特有的行为模式来进行网络流量分类。在此基础上，大多数研究者使用 *Packet-Level* 和 *Flow_Level* 两个层面上的信息的统计特性来进行网络流量分类[12,13,14,15]。这些分类技术基于这样一个事实，不同的应用类型在网络中进行通信时有其独特的行为模式。例如，使用 FTP 进行大量文件传送时，报文之间的交互到达时间非常小，而且平均报文大小会比一个客户端向另一个客户端发送即时消息的报文要大很多。而且根据这些传输时的特定模式，一些 P2P 应用(例如 BitTorrent)可以与 FTP 区分出来，因为 P2P 常常是持久连接而且是双向的传输数据，而 FTP 的连接不是持久的和单向传输数据的。

本文中提出一种基于聚类分析的网络流量分类的方法，这种方法将仅使用传输层的统计特性(如双向报文到达时间、双向报文长度等)，对包含了 P2P(如：BitTorrent 等)和非 P2P(如：WEB、FTP、DNS 等)等应用类型的网络流量进行分类。

1.2 研究意义

基于聚类分析(*Clustering Analysis*)的网络流量分类是指在基于 TCP/IP 协议的互联网中，按照 5 元组源 IP 地址、源端口号，目标 IP 地址、目标端口号及 IP 协议) 的定义，将报文(*Packets*)分成双向 TCP 或 UDP 流(Flow)，抽取与协议和端口无关的流的特征，形成特征向量，用特征向量来表示流，以流的应用类型(如 FTP，P2P，网络游戏等)作为流的类别，引入基于聚类分析的流量特征选择策略，研究基于聚类的网络流量分类的技术，力图找到一种快速高效的网络流量分类方法。建立一种能够满足当前网络技术快速发展、网络管理人员能够根据需要对网络流量进行监控和管理、性能卓越的网络流量分类体系。

鉴于基于端口的分类方法和基于有效载荷的分类方法的效率的降低，采用机器学习的方法对网络流量自动分类是一种有效途径。本研究提出基于聚类分析的研究方法是机器学习领域中一个重要方法。针对网络流量的实际问题，可以对已知的和未知的网络流量都可以进行聚类分析。可以动态的识别和分类网络流量类型，对网络规划、*QoS*、动态访问控制、入侵检测等具有很强的实用价值。

1.3 国内外研究现状

1.3.1 基于端口(Port-based)的分类

历史上，使用周知端口号对 Internet 网络的流量分类技术取得了很大的成功，这是由于很多传统的应用类型使用 IANA 分配的固定端口号进行通信。表 1-1 列出了部分 IANA 分配给常见的应用类型的端口号。例如 email 应用通常在端口 25

使用 *SMTP* (*Simple Mail Transfer Protocol*)发邮件，而在 110 端口使用 POP3 协议 (*the Post Office Protocol version 3*) 收邮件。

近来，很多最新开发的应用不再使用标准的端口[7、10、16]通信使得基于端口的分类方法的效率日渐低下。特别是，由于一些新的P2P应用不再使用固定的和事先预知的端口号，而且现在许多应用并没有IANA分配或注册的端口号，它们使用周知的端口号，而这些端口号与IANA分配的端口号可能存在交迭，导致基于端口的方法无法正确识别流量的应用类型。甚至周知的或注册的端口的应用，也会下列原因使用不同端口号，而无法正确识别和分类。即，非特权用户通常使用1023以上的端口；用户可能故意隐藏他们的存在或绕过基于端口的过滤器；若干服务器共享单个IP地址（主机）；许多应用，如被动FTP或视频/声音通讯 (*Video/Voice*)，使用不可知的动态端口。

表 1-1 IANA 分配给几种常见应用类型的端口号

应用类型	使用的端口号	应用类型	使用的端口号
FTP Data	20	FTP Control	21
SSH	22	Telnet	23
SMTP	25	DNS	53
HTTP	80	POP3	110
IRC	113	NNTP	119
SOCKS	1080	HTTPS	443

1.3.2 基于有效载荷(payload-based)的分类方法

另一种网络流量分类的方法可以有效地避免基于端口的分类方法的不足，这种方法通过分析包的有效载荷对网络流量类型进行识别，该方法也被称为“深层包检测”。在此方法中，对数据包的有效载荷进行分析，以确定是否含有已知应用程序的特征签名。应用有效载荷分析用于网络流量分类的一个典型例子是Moore[16]等人的研究，他们描述为基于内容的网络流量分类。他们的分类方法的第一步是根据 IANA 分配的端口号来建立一个初始分类，然后是一个迭代过程，也就是根据有效载荷来分类流量。在文献[2,4,17]中，Sen 等人发现了一种可以精确识别 P2P 应用的方法。他们的方法是基于应用层签名。这种方法在针对 P2P 网络流量分类时十分有效，因为多数 P2P 协议都有固定的特征串，例如 BitTorrent protocol = “13 42 69 74 54 6F 72 72 65 6E 74 20 70 72 6F 74 6F 63 6F 6C”，只要通过特征串匹配就能轻松地对 P2P 进行分类。基于有效载荷的识别方法曾经得到广泛的研究，有些研究成果甚至还投入到商用。

尽管基于有效载荷识别这种技术避免依赖于固定端口号，但它增加了网络识别设备的复杂性和处理的负担，如随着P2P应用的增加，特征串的数量也相应增加，使得该方法每检测一个报文所需要匹配的特征串越来越多，从而识别的效率逐渐降低。这种方法必须保持与广泛的应用语义和网络级语法知识的一致性，必须有能力对潜在的大量流进行并发分析。这种方法当遇到处理私有协议或加了密的流量时就十分困难或是不可能。另一个问题是直接分析应用层内容可能触及侵犯个人隐私等法律问题。

1.3.3 基于机器学习的网路流量分类方法现状

随着基于端口号和基于有效载荷的网络流量分类方法的缺点日益显现，使用机器学习方法对网络流量进行分类是一种很有前途的方法。国内外一些学者开始使用机器学习(Machine Learning)的方法[18]来进行网络流量分类的研究。这些研究方法大多数是在 Flow-level 的层次上展开研究的，认为不同的应用具有不同的传输数据的模式，因此根据这些模式可以对流量进行分类。这些方法的特点是抽取与协议和端口无关的统计特性(如报文长度，持续时间等)，形成特征向量，用特征向量表示流，以流的应用类型(如 WEB, FTP，DNS 等)作为流的类别，然后用机器学习方法构造分类器，对网络流量进行分类。

对流分类使用机器学习技术的思想在文献[19]中的入侵检测中初次提出。用四个属性作为流的特征，分类数据以 24 小时为周期来进行采集的。McGregor[15]等人使用传输层属性对网络流量进行了聚类分析，但哪些属性能够得到最好的聚类结果，没有进行分析 Moore[11] 采用监督的 *Naive Bayes* 分类方法进行流量分类与应用识别，首先将网络流量数据手动分类，确定了流量的具体应用类型，并将流量数据分成训练集和测试集。利用 *Naive Bayes* 方法进行分类，平均分类准确率超过了 83%。Roughan 采用最近邻和线性判别分析的方法[12] ，仅使用连接持续时间和包的平均大小作为流量分类的特征，采用 Bayes 的方法进行分类，成功地把网络应用映射到不同的 *QoS* 类别。然而只采用两个属性的统计信息并不能区分所有的应用类别，因此获得的准确度很低。S. Zander 等人采用了 Autoclass 的方法[20]，并通过特征选取技术 SFS(*Sequential forward search*)来选取较优的流属性集，并评价了不同的特征集对结果的影响。为了验证其方法的有效性，使用从不同的网络位置收集的网络数据来进行测试，得到了较好的分类结果。Jeffrey Erman 等人采用无监督的 EM(*Expectation Maximization*)方法[25]，识别不同应用的网络流量，使用 *Total Number Packets*、*Mean Packet Size*、*Mean Data Packet Size*、*Flow Duration* 和 *Mean Inter-Arrival Time of Packets* 这五个流量统计特征来标识每个连接。通过与 Bayes 的分类方法进行比较，获得了更为准确的分类结果。但

该方法的缺点是训练时间较长。

聚类分析作为机器学习领域的重要方法，采用聚类分析的方法对网络流量进行分类也越来越受到多方学者的重视。文献[21]采用K-Means和ADC(*approximate distance clustering*)两种聚类算法应用到入侵检测和网络监控中，收到了明显效果。Shi Zhong[22]等在基于聚类的网络入侵检测的研究中采用*k-means, Mixture-Of-Spherical Gaussians, Self-Organizing Map, Neural-Gas*等方法的无监督学习中体现了聚类方法在该应用的可行性，而这些方法的有监督学习方法则成功发现了网络入侵检测中未知的攻击类型。Gerhard Munz等人在文献*"Traffic Anomaly Detection Using K-Means Clustering"*[23]一文中，使用K-Means聚类算法对异常流量进行检测。他们没有对训练数据集预先进行标注，最后成功的将规则的流量和异常流量进行划分。Jeffrey Errman[24、25]等人在其连续的研究中使用K-Means、DBSCAN、AutoClass三种半监督聚类算法，用少量的标注样本来训练分类器，达到了非常理想的分类效果，并且成功地实现了流到应用类型的映射。

国内对基于机器学习的网络流量分类研究还不多见，文献[26]使用粗糙集理论(*Rough Set Theory*)和遗传算法(*Genetic Algorithm*)来构建和优化分类器，对网络流量进行分类研究。文献[27]的作者利用自组织映射图(SOM)的人工神经网络，对网络流量进行聚类分析研究。邓河[28]等使用 SVM 方法对文件共享中的BitTorrent,流媒体中的PPLive,网络电话中的Skype,即时通讯中的MSN 4种P2P网络流量进行分类研究.介绍了基于SVM的P2P流量分类的整体框架,描述了流量样本的获取及处理方法,并对分类器的构建及实验结果进行了介绍.实验结果验证了提出方法的有效性,平均分类精确率为92.38%.

本文采用若干聚类分析算法,仅使用传输层的统计特性(如双向报文到达时间、双向报文长度等),对包含了P2P(如BitTorrent 等)和非P2P(如WEB、FTP、DNS)等的网络流量进行分类。实验表明，这些方法对网络流量进行分类，查准率和总精确度均较高。

1.4 文章主要内容及组织

文章的组织结构安排如下：

第一章介绍了网络流量分类的研究动机、国内外的研究现状以及存在的问题，同时，给出了这篇论文所研究的主要内容。

第二章主要讨论了网络流量样本集采集与特征产生，对捕获的报文采取五元组的方式规则为网络的应用流，计算网络流的相关统计特征，采用综合方法进行自动地标识网络流的应用类型。

第三章详细说明了基于 PCA 和基于信息增益的网络流量特征选择方法。

第四章主要分析了基于聚类的网络流量分类的两种类型算法：基于密度的 DBSCAN 算法和基于划分的 K-Means 算法；并对提出的网络流量分类方案进行了实验测评。

第五章描述了基于聚类方法的网络流量分类系统的设计，包括各个功能模块的设计及实现。

第六章对论文研究工作进行总结，介绍了文章研究工作的创新点，分析将来进一步的研究工作。

第二章 数据采集及网络流的形成

网络流量样本集的获取是基于机器学习方法网络流量分类的基础。本章主要介绍了网络数据的捕获方法，如何将捕获到的网络数据解析为网络流，网络流的自动标识以及网络流量候选特征的形成。对网络报文的捕获，采用了包过滤机制仅截取包前 N 个字节，主要是考虑到对报文深层扫描的流的标识。针对网络应用流的标识，采取了基于端口、基于有效载荷等多种方法的综合形式进行标识。网络流的特征主要是计算独立于报文的协议和有效载荷的相关统计特征，如报文长度、持续时间等。

2.1 网络数据采集

网络数据采集是指在TCP/IP网络模型下，采集Internet中的TCP报文和UDP报文。TCP/IP网络架构模型可以分为五个逻辑结构，如图2-1所示；通常情况网络中数据报文被主机的网卡得到后，就直接交给了操作系统的协议栈。协议栈按照数据链路层、网络层、传输层、应用层的顺序一层层地识别、丢弃包头并进行分析，最后将得到的数据包内容交给目的应用程序。但是，这些经过协议栈处理的数据包内容对我们进行流量分类来说是没有意义的，因为包含流量信息的包头在协议栈的分析处理过程中就被丢弃了，网络流量分类需要得到的是原始的网络数据包。这样就要求有一种能够捕获原始数据包，包括发送到正在运行的主机上的数据包和在其它主机在共享媒介上交换的数据包的网路数据捕获工具。WinPcap 就是这样一种很好的网路数据捕获工具。

图 2-1 TCP/IP 网络架构模型

WinPcap[29]是一个在 Windows 操作系统下的免费的、公开的可以直接访问网络的系统。WinPcap 包含了一个最优化的内核模式驱动——称作 Netgroup

Packet Filter(NPF)，和一套与 libpcap 兼容的用户级函数库。WinPcap 使 Unix 平台下的应用程序能方便地与 Win32 平台下的程序联系，并且它能使一套很大的函数库只需通过简单的重新编译就立刻在 Win32 平台下使用。而且，由于网络监听的重要性，WinPcap 还为此提供了特殊的系统调用函数。WinPcap 可以捕获原始数据包，包括发送到正在运行的主机上的数据包和在其它主机在共享媒介上交换的数据包；将数据包发送给应用程序之前按用户规定对捕获的数据包进行过滤；向网络发送原始数据包；对网络通信进行统计。

此外，WinPcap 可以独立于主机的协议(如 TCP-IP 协议)进行接收和发送数据包。这意味着 WinPcap 不能阻塞、过滤或处理本机上其它程序产生的数据：它仅仅能嗅探在网线上传输的数据包。因此，WinPcap 不能在 traffic shapers、QoS schedulers 和个人防火墙这类应用程序中使用。

鉴于 WinPcap 的强大功能，因而受到了广泛的是用。国内外众多的网络安全软件都使用了这个函数库进行开发，并取得了良好的效果。本文的捕获系统也是基于 WinPcap 的函数库进行的开发。

2.1.1 捕获方法

为了获得充足有效的实验数据，我们通过校园网络中心交换机(Cisco 6509)的端口镜像的方式来采集网络流量数据。然后在采集终端运行WinPcap程序就可以采集到网络中的原始数据。虽然我们的分类方法只使用流的统计特征，但是应用层信息可以帮助我们训练分类器和应用类型的确定。因此，我们在采集数据的时候适当的保留了部分应用层信息。此外，本研究所使用的数据时间跨度要求很大，以检验我们构建分类器的有效性。这就要求我们克服以下困难：

(1) 保留一些必要的应用层信息以辅助标识网络流属于哪种应用类型，那么就要捕获相关的应用层报文的报头，然而应用层报头的长度会随着应用的不同而不同，因此为简单起见，我们截取每个报文的前128个字节内容组织成Libpcap (*.dmp)格式的网络流量踪迹文件(Trace Files)。这样既包含了所需要的报头信息，又节省了存储空间。

(2) 由于数据量大，时间跨度长，需要足够的网络踪迹文件存储空间。

另外，由于采集的数据是报文的前 128 个字节，含有丰富的信息，也可以满足基于端口和有效载荷方法对网络流的分析和标识。图 2-2 是网络数据采集界面图，图 2-3 是 dmp 文件的组织形式。

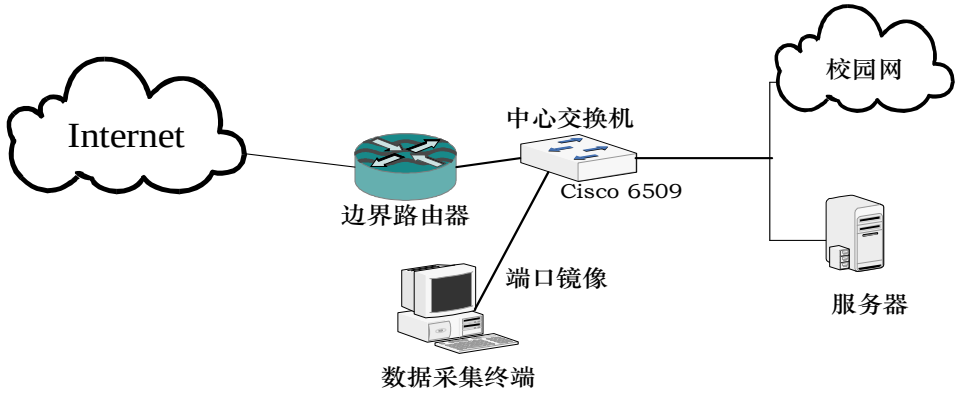

图 2-2 网络流量数据采集示意图

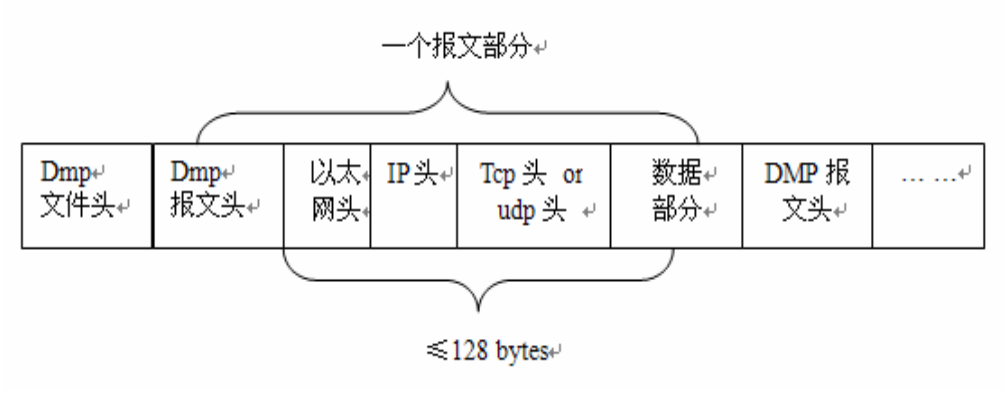

图 2-3 *.dmp 文件格式示意图

2.1.2 数据集

为了保证采集的网络踪迹文件满足我们的研究，我们历时 18 个月，在不同的时间段采集了大约共采集了 180GB 的网络踪迹文件。并且我们还制定了多种采集方案。如每隔 8 小时采集一小时的网络数据，每隔 24 小时采集一小时，每隔 48 小时采集一小时和从晚上 20：00 到次日 8：00 连续采集等方案。并将采集好的文件按采集时间标注好。表 2-1 列出了部分数据集。

表 2-1 通过捕获报文而得到的部分数据集

数据集	开始时间	结束时间	持续时间(s)	数据大小(GB)
SubSet1	2008-09-17 15:29:30	2008-09-17 21:06:10	20200s	7.25
SubSet2	2008-08-21 21:06:10	2008-08-21 03.24.59	20470s	7.60

Subset3	2008-09-12 03.24.59	2008-09-12 09.47.43	23099s	8.20
Subset4	2008-09-15 09.47.43	2008-09-15 15.30.22	20559s	7.63
Subset5	2008-10-09 14:05:25	2008-10-09 16：08：13	14565	4.98
Subset6	2008-10-10 19:57:48	2008-10-11 08:31:01	45193	7.56
Subset7	2008-10-11 22:18:18	2008-10-12 07:54:35	34013	4.14

2.2　网络流的定义及特征计算

2.2.1 流的定义

本文以网络流为研究对象，将流定义如下：在基于 TCP/IP 协议的互联网中，按照报文(Packet)的五元组(Five Tuple) ，即，源 IP 地址、源端口号，目标 IP 地址、目标端口号及 IP 协议，将报文分成双向 TCP 或 UDP 流(Flow)。规定流与流之间的空闲时间(Idle Timeout)为 60 秒，超过 60 秒被认为是不同的流。

按照上述流的定义将流及其相关信息我们形式化描述如下：$F=\{F_1,\cdots,F_i,\cdots,F_N\}$ $(i=1,2,\ldots,N)$ 表示样本流集合，F_i 表示第 i 条流，其中 $F_i=\{f_{i1},\ldots,f_{ij},\ldots,f_{iM}\}$ $(j=1,2,\ldots,M)$，F_{ij} 表示 i^{th} 流的 j^{th} 的属性值。上述表示中，N 是样本流的数目，M 是流的属性数目。在我们的研究中，流看成是一个 M 维向量。在网络流量分类中，流的属性由流的统计特性，如持续时间，双向字节传输数，数据包的总数等信息产生。设 $L=\{L_1,\ldots,L_p,\ldots,L_P\}$ 为流的应用类型的标签集合，P 表示是应用类型的数量，P 的取值代表不同的应用类型。

2.2.2 流特征分析

基于机器学习方法的流量分类的一个重要目标是应独立于协议、通信端口的统计特征，找到能很好适合于机器学习的网络流量方法，克服基于端口和基于有效载荷方法的缺陷，提高网络流量的分类准确度。流特征分析主要考查流的流的时间特征、流中报文的标志位信息特征，流的报文个数及大小特征。由这些特征产生的流的候选特征集。

(1)流的时间特征

流的时间特征主要有报文间隔到达时间(*Packet Inter-Arrival Time*)，流的持

续时间(*Duration*)，空闲时间(*Idle Time*)，活动时间(*Active Time*)等。文中仅给出流的持续时间示意图。图 2-4(a)是我们考查 403 条 DNS 流的持续时间分布图，图 2-4(b)是我们考查 455 条 SMTP 流的持续时间分布图。

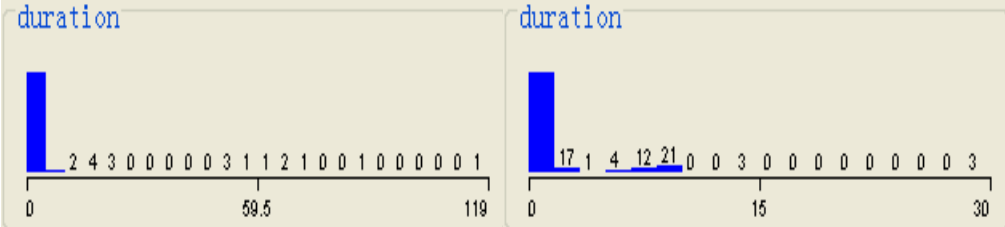

图 2-4(a) SMTP 流 Duration 分布　　图 2-4(b) DNS 流 Duration 分布

(2)流的报文个数及大小特征

网络流是由一系列的报文所组成，所以可以抽取流的报文个数等相关属性作为流的特征。流的报文个数及报文大小特征主要有前向、后向总的报文的个数、字节数、最小长度、最大长度、平均长度、均方差等属性。而这些属性根据应用类型的不同，差别很大。这里比较了 BitTorrent、DNS 、FTP 和 WEB 四种应用类型的后向包个数的差别，如图 2-5 所示。从图中可以看出四种不同应用类型的网络流的后向包的个数有很大的差别，BitTorrent 流的后向包个数最多，WEB 流次之。说明 BitTorrent 在下载的同时不断的向其他用户传输数据。

流的报文个数及报文大小的统计特征对于基于机器学习的网络流量分类的分类器构建很有帮助，不同应用类型的这些统计都有所不同，有些甚至差别很大，如报文的平均长度大小，这些特征都可以作为机器学习的候选特征。

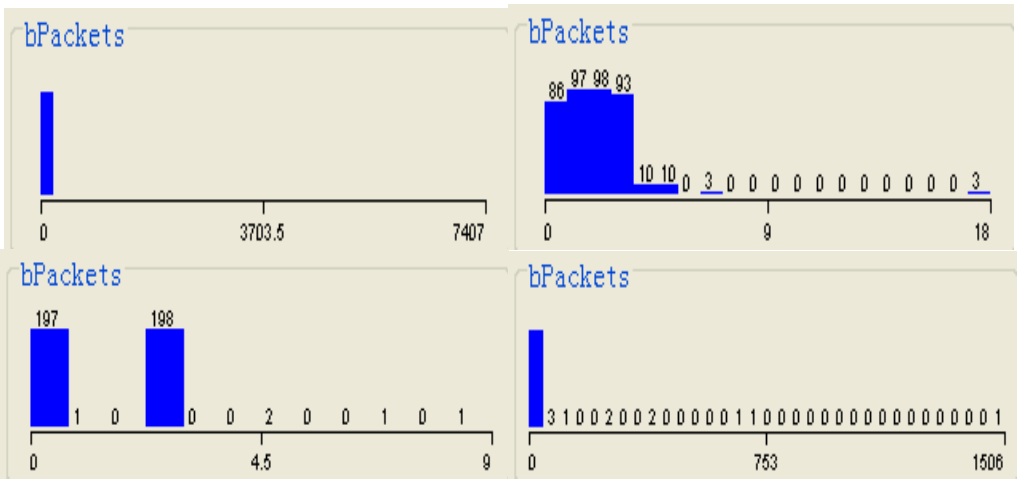

图 2-5 BitTorrent、DNS 、FTP 和 WEB 四种类型后向包的数量比较

(3)流中报文的标志位信息特征

报文的标志位信息特征主要包括前向、后向报文中 *Ack* 、*Rst*、*Psh*、*Urg*、*Syn* 等标志位特征。图 2-6 图列出了 FTP 类型的标志位的相关信息。2-6 表示前(后)

向报文中有推送比特位的包的个数。

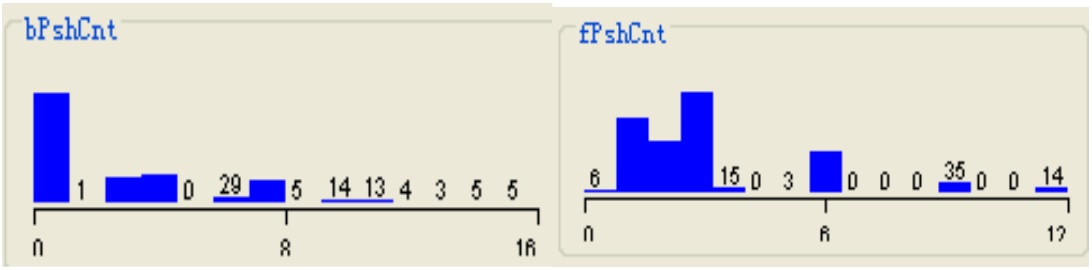

图 2-6 前(后)向报文中有推送比特位的包的个数

(4)双向报文长度的范围区间

双向报文长度范围区间属性包括前向、后向报文长度小于 100、500、1000、大于 1000 的报文特性。图 2-7 列出 WEB 类型的相关信息。

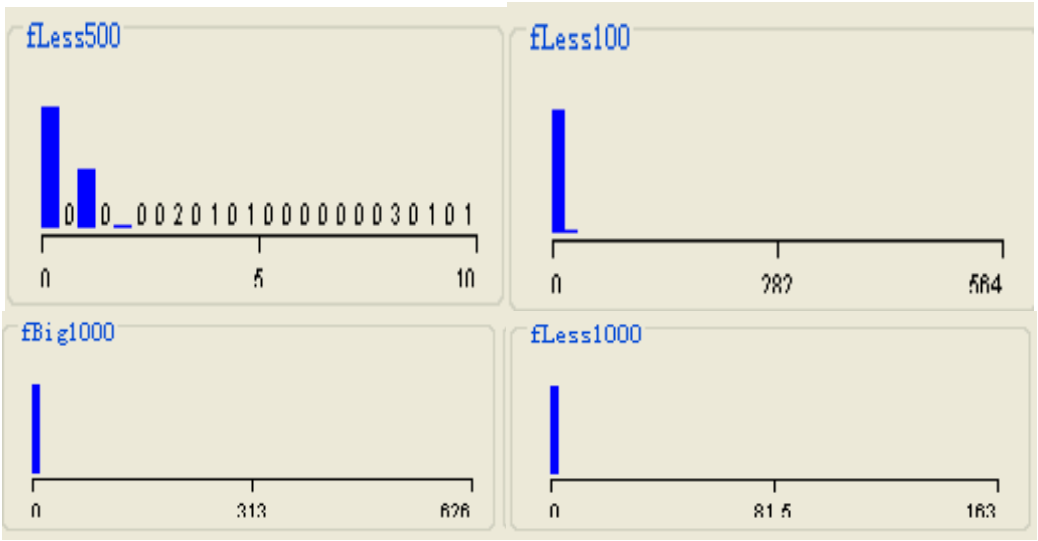

图 2-7 WEB 应用前向报文长度属性

2.2.3 流的候选特征集

文献[30]的研究表明，潜在的流的特征有很多，本文通过上述方法产生了 34 个属性作为候选特征。表 2-2 中所列出了这 34 个候选特征。

表 2-2 候选特征集

序号	符号表示	特征描述	备注
1	*fPackets*	前向总的报文的个数	流的报文个数及大
2	*fBytes*	前向总的报文的大小	
3	*bPackets*	后向总的报文的个数	
4	*bBytes*	后向总的报文的大小	
5	*minFpktLen*	前向报文的最小长度	
6	*maxFpktLen*	前向报文的最大长度	
7	*meanLenFsm*	前向报文的平均长度	

8	*stdLenFqsm*	前向报文的均方差	小特征
9	*minBpktLen*	后向报文的最小长度	
10	*maxBpktLen*	后向报文的最大长度	
11	*meanLenBsm*	后向报文的平均长度	
12	*stdLenBqsm*	后向报文的均方差	
13	*duration*	流的持续时间	时间特征
14	*fLess100*、	正向报文长度小于100	双向报文长度的范围区间
15	*bLess100*	反向报文长度小于100	
16	*fLess500*	正向报文长度小于500	
17	*bLess500*	反向报文长度小于500	
18	*fLess1000*	正向报文长度小于1000	
19	*bLess1000*	反向报文长度小于1000	
20	*fBig1000*	正向报文长度大于1000	
21	*bBig1000*	反向报文长度大于1000	
22	*avePktPerSecond*	每秒钟平均报文大小	
23	*fframNum*	前向报文分片标志	双向分片标志位特征
24	*bframNum*	后向报文分片标志	
25	*fAckCnt*	前向报文中ACK标志位	双向报文的TCP标志位特性
26	*bAckCnt*	后向报文中ACK标志位	
27	*fRstCnt*	前向报文Rst标志位	
28	*bRstCnt*	前向报文Rst标志位	
29	*fPshCnt*	前向报文推送标志位	
30	*bPshCnt*	后向向报文推送标志位	
31	*fUrgCnt*	前向报文紧急标志位	
32	*bUrgCnt*	后向报文紧急标志位	
33	*fSynCnt*	前向报文Syn标志位	
34	*bSynCnt*	后向报文Syn标志位	

2.3 流量应用类型的自动标识

互联网上的应用纷繁复杂、协议繁多，并且随着技术的革新将有更多的应用类型出现。表2-3列出了目前常见的几种应用类型。而传统的基于端口的分类方法和基于有效载荷的分类方法已不能正确将它们分类。机器学习方法虽然为网络流量分类开辟了新的道路，但是这要求研究者为其提供有利于机器学习的样本。为了形成有利于机器学习的样本，首先要对网络流量类型进行正确的标注。对样本流的应用类型的正确标注，不但可以提高分类器的学习效率，还可以检验分类器的查准率。

表 2-3 互联网上常见的应用类型

类型	应用协议
Internet	*FTP，HTTP，SSH，WEB，HTTP2，DNS* 等
E-Mail	*POP3,SMTP*等
Games	*Half-life，MSN Zone， Kali，Yahoo Games* 等
P2P	*EDonkey， Emule，Gnutella， KazzaA，Direct Connect，BitTorrent，BtSprite, fasttrack*等
MultiMedia	*NetMeeting, QuickTime, RealAudio, WindowsMedia, PPLive, UUsee， RTSP*等
Messageing	*ICQ， OICQ，MSN Messenger， Yahoo！Messenger*等
Database	*FileMake Pro， MS SQL, Oracle*等

分类中的样本要涉及到互联网上的多种服务应用，采用单一方法进行标识样本或人工的方法已不能满足当前的需要。针对以上这些按照不同通信模式或会话结构的互联网络应用，我们综合了基于端口识别、基于特征码识别和基于协议解析算法等多种算法，设计了样本自动标识系统。

2.3.1 基于端口的识别

端口识别法是根据 TCP 数据包或 UDP 数据包首部的源端口号或目的端口号识别一些常见应用类型的流量，如 HTTP，SMTP，Telnet，HTTPS 等。 这些协议及应用类型使用固定端口进行通信大致分三种情况：

(1) IANA 组织分配的公认端口，例如，web 应用、E-mail 应用，DNS 协议等。公认端口(Well Known)主要由超级用户进程或特权用户程序使用，由 IANA 统一分配，在 0 到 1023 之间。

(2) 企业开发的专用协议或应用使用在 IANA 组织注册的登记端口，例如，MS SQL、Oracle database 等应用；登记端口(Registered Ports)由普通用户进程使用，在 1024 到 49151 之间；

(3) 开发的一些流行协议或应用也使用专用端口，尽管这些端口未在 IANA 组织注册，但可以通过流量分析得到这些网络应用的常用端口。

2.3.2 基于特征串的识别

基于特征串识别方法是检查一个流前几个数据包的负载部分，确定是否存在

预定义的应用特征码。应用特征码有两种类型，一般是固定长度字符串，另一种是可变长度字符串，可用正规表达式表示。一些网络应用及其特征见表2-4。

表 2-4 常见的网络应用及其特征

应用名称	特征串	说明
BitTorrent	BitTorrent 客户端之间的握手消息格式： 数据包的第一个字节开始，内容为： <0x13><BitTorrent protocol>，特征串有： BT_CHOCK, BT_UNCHOKE,BT_UNINTERESTED, BT_HAVE, BT_BITFIELD, BT_REQUEST, BT_PIECE, BT_CANCEL, BT_KEEP_ALIVE, AZ_PEER_EXCHANGE	使用正规表达式来表示特征码
directconnect	TCP 数据是一系列命令，格式为： $command_type field1 field2…\|；以“$”开头，“|”结束。Command type 有：Send Get ， Dir，ConnectT， Supports，Hello， MyINFO， Search，MyNick，Quit， Key，RevConn， Version，Lock，HubName	使用TCP协议通信。
edoneky	\xe3, \xc5	\x为16进制
gnutella	TCP 报头后第一个字符串包含：‘GNUTELLA’，‘HTTP’或者‘GET’。如果第一个字符串为‘HTTP’或者‘GET’，后面必须有下面一些字符串之一： User-Agent: <Name> UserAgent: <Name> Server: <Name> 其中，<Name>为下面一些字符串之一： GNUTELLA CONNECT，LimeWire, Mactella, Morpheus, Mutella等	使用TCP协议通信。
kazaa	请求消息和应答消息中都包含“X-Kazaa-supernodeIP”字符串。如果TCP报头后跟‘GET’或‘HTTP’之一，并且必须有一个字符串为‘X-Kazaa’。如 .*Kazaa	
rtsp	.*rtsp)	目的端口为554
real	.*GET ， GETSon3077	目的端口为3077
ssh	.*SSH	目的端口为22

基于特征识别方法的特点是不管网络应用使用什么端口，都可以准确识别流量的应用类型。其缺点是特征码匹配开销很大，同时获取一个网络应用的特征码需要大量流量数据的分析。

2.3.3 基于协议的解析

基于协议的解析方法适用于识别一些使用动态端口的网络应用产生的流量。比如，VoIP、流媒体、多媒体等互联网应用。这类应用的特点是先通过一个公开的固定端口建立一个控制会话(Control Session)，在控制会话中协商出后面的数据会话(Data Session)的动态端口，有些应用则是在协商出数据会话的动态端口之前，还有一个二级控制会话(Second Control Session)的端口协商过程，后者的典型应用如 H.323 体系的多媒体通信，这些应用除了关注流量和流向外，往往还需要连接时长、适用的何种音视频编解码算法、音视频质量等信息，因此，就有必要对 Payload 进行解析。

为分析上述互联网应用，我们需要先建立一个描述控制会话端口的表(CPT)，DSPP 算法的描述为：

步骤 1、如果网络包的主端口在 CPT 中存在，并且没有设置 FIN 标志，则执行步骤 5；否则执行步骤 6；

步骤 2、由分派函数指派相应的协议解析函数，对 Payload 进行协议分析；

步骤 3、如果协议分析结果正确，则为该应用建立一个会话流(Session Flow)，并且解析出后续动态会话(Dynamic Session)端口；否则，该网络包视为其他应用，参与其他应用的分析；

步骤 4、如果所属应用有二级控制会话，则仿照 2、3 的步骤，继续进行 Payload 的分析，直至解析出数据会话(Data Session)端口；

步骤 5、所有属于二级控制会话和数据会话的网络包，都属于该应用的这个会话流；

步骤 6、如果收到 Disconnect 的网络包，或者在一定的超时时间内会话流的数据包不再到达，则该会话流结束。

基于协议解析方法的特点是可以准确识别流量的应用类型，缺点是需要解析信令数据包，处理开销大，同时只适用于已知协议类型。

2.3.4 样本自动标识过程

综合以上分析，我们设计了网络流量应用类型自动标注的流程。由四个步骤组成：

步骤 1： 查看流的端口是否用常用的端口，如果是的则采用基于端口的标注方法进行标注，否则的话进入步骤 3。

步骤 2：对标注为 WWW 的流进行 BitTorrent 的特征串匹配。实验中我们发现很多 BitTorrent 流量使用了 IANA 分配给 WWW 的端口 80，所以在步骤 1 之

后，对 WWW 的流进行 BitTorrent 的特征串匹配。以区分伪装端口的流量。

步骤 3：对于没有使用基于端口的方法标注的网络流采取基于特征串匹配方法进行标注，如果没有符合的特征串匹配，即转入步骤 4。

步骤4：使用基于协议解析的方法进行标注。样本的类型自动标注流程图见图2-8。

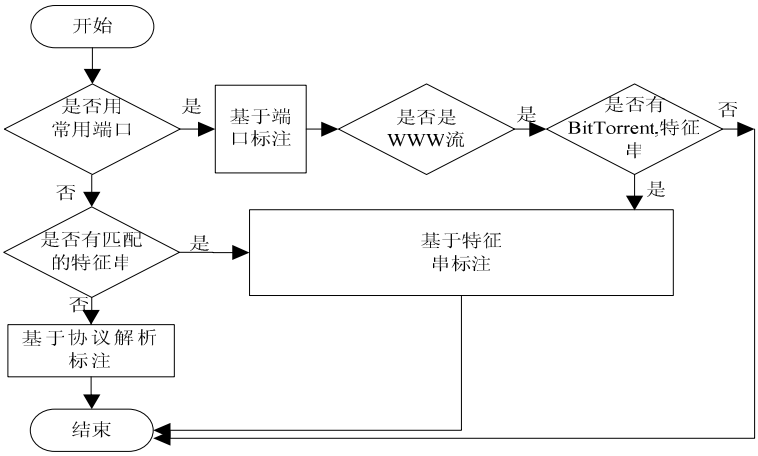

图 2-8 网络流样本自动标识过程

2.4 小结

本章主要对网络流量数据样本采集做了详细介绍，对采集的数据根据五元组的规则将数据解析为流。根据流的定义我们计算了流的统计特征，并由这些特征产生候选的特征集。为了形成有利于机器学习的样本，我们结合端口号、协议解析、基于特征串等多种方法设计了流量样本类型自动标注流程。为网络流量流量分类工作做好了准备。

第三章　网络流量的特征选择

聚类分析和其他机器学习算法一样，有一个重要问题需要解决，即选择什么样的特征来训练分类器，这就是特征选择问题。本章主要讨论了特征选择问题，介绍了特征选择的意义、分类、国内外研究现状。此外，还着重介绍了基于 PCA 和信息增益的网络流量最优特征子集的选择方法，通过对 2.2 产生的候选特征集的计算，得到了算法各自的最优特征子集。

3.1 特征选择概述

描述网络流的潜在特征有很多，可能达到成百上千个特征。特征集合的描述质量直接影响机器学习算法的效率。特征选择可以看作是对样本的特征空间一个优化，若在特征集合中存在大量冗余或不相关的特征，不但会在基于机器学习的网络流量分类中分类准确度降低，而且会增加机器学习算法的搜索空间，导致算法效率降低。因此，选择和提取能准确刻画网络行为的特征对网络流量分类就显得至关重要。特征选择就是根据给定的准则从一组特征中挑选出一些最有效的特征以降低特征空间维数，无用的、冗余的以及最少使用的特征将被从特征集合中删除。由此，特征选择的目的是要找出特征的一个子集，此子集在特征表述性能上比得上整个特征集。

3.1.1 特征选择的分类

从使用方法上特征选择可以别分为两大类：一类叫做过滤器[31,32](Filter)的特征选择算法，它独立于分类器对数据进行处理，即在训练开始之前去除不必要的属性。算法使用基于数据一般特点的启发性规则对特征子集进行评价。另一类方法叫做嵌入方式[33,34](Wrapper)，这种方法在特征选择时考虑具体机器学习算法的特点。它使用某一归纳算法结合重复统计抽样技术(比如交叉确认)来评价特征子集的准确性。Filter 方法数据处理速度比较快；Wrapper 方法准确性很高，但速度慢，且丧失了数据的一般特性。在数据量比较小，且分类器已经确定的情况下应该选择 Wrapper 方法，而对于大数据量的应用应该选择 Filter 方法。

3.1.2 特征空间的搜索方向

搜索方向也就是要评价的特征子集产生的次序。搜索的方向有从空集开始的前向搜索、从全集开始的后向搜索、双向搜索和随机搜索等。

(1) 前向搜索 SFG(*Sequential Forward Generation*)。前向搜索从空集 S 开始，随着搜索的进行，依据某种评价标准从未被包含在 S 里的特征集中选择最佳的属性不断加到 S。通常分析人员预先知道有些属性是与目标相关的，在这种情况下搜索是从这些属性构成的子集开始的，而非空集。

(2) 后向搜索 SBG(*Sequential Backward Generation*)。后向搜索从全集 S 开始，依据某种评价标准不断从 S 中选择最不重要的属性，直到达到某种停止标准。它是对前向搜索的补充，因为有时候评价最不重要的特征比评价最有用的特征要容易。

(3) 双向搜索 BG(*Bidirectional Generation*)。搜索到特征子集空间的中部时，需要评价的子集数就会急剧增加，当最佳属性子集不是在属性子集空间的中部时，使用单向搜索前向或后向)，如果搜索要通过子集空间的中部就会消耗掉大量的搜索时间。双向搜索同时向两个方向开始搜索。一般当其中一个方向搜索到最佳子集或两个方向在中部相遇时停止搜索，而在中部相遇的可能性是很小的。所以双向搜索是比较常用的搜索方法。

(4) 随机搜索 RG(*Random Generation*)。随机搜索从任意的方向开始，对属性的增加和删除也有一定的随机性。这样做是为了克服局部极小。与 SFG 和 SBG 不同，尽管能够察觉到维数不断减小或增加的趋势，它的下一个产生的属性子集的维数是不可预测的。

3.1.3 搜索策略

由于搜索空间的大小各异，我们需要使用不同的搜索策略。选择正确的搜索策略对这两个方面都有帮助，正确的搜索策略应该能够根据具体情况得到正确性和时间消耗之间的平衡点。搜索策略可以大致分为三种：

(1) 完全搜索：为了不丢失最优解，通常它会搜索到每一个属性子集，空间复杂度是 O(2N)。如果我们指定了最小结果集的维数 M，前向搜索的搜索空间大小是 $C_n^0 + C_n^1 + \cdots + C_n^M$，对后向搜索，空间的大小是 $C_n^n + C_n^{n-1} + \cdots + C_n^M$。

(2) 启发性搜索：是在搜索的过程中使用启发性信息。通常启发性搜索的搜索空间只是在空集和全集之间的一条路径，这比完全搜索快得多，因为它只沿着一条特殊的路径处理数据，并且得到的解是近似最优解。

(3) 不确定性搜索：这种策略随机产生下一个待评价的子集，而不是顺序产生。新产生的子集要在维度、准确性等方面比当前的最佳子集更好，才会被记录下来。

3.1.4 评价方法

对于评价方法，可以把它看成给特征子集或某一属性打分。这里用 U(S) 表示对特征子集 S 的评价值(U 的值越大就认为 S 越好)，用 D(S)表示 S 的维数。当 $U(S_1)>U(S_2)$且 $D(S_1)>D(S_2)$或 $U(S_1)\geq U(S_2)$且 $D(S_1)<D(S_2)$时，称 S_1 比 S_2 好。可供选择的评价方法比较多，常用的有:信息增益、距离、依赖性、一致性、准确性等方法。本文此处仅介绍一致性评价。

一致性评价方法具有单调性、计算复杂度低和一定的噪声处理能力，因而被广泛使用。一致性方法是通过考察数据集的不一致率来评价的。设 x 和 y 是数据集 D 中的两个实例，如果 x 和 y 除了决策属性不同，条件属性的取值都相同，则说 x 和 y 是不一致的。为得到一个数据集 D 的不一致率，先做如下定义：把 D 分割成 $FD=\{D_1,D_2,\cdots,D_i,D_j\}$($i,j$ 是条件属性集合所有可能的取值数)，其中 D_i，是指对任意的 $x,y\in D_i$,x 和 y 条件属性的取值均相同;对于 FD 中的任意 $D_l,D_k(l\neq k)$，不存在 $x\in D_l$ 和 $y\in D_k$，使得 x 和 y 条件属性的取值均相同。再对 FD 中的每一个元素 D； 做同样性质的分割，D_j 被分割成 $FD_j=\{D_{j1},D_{j2},\cdots D_{jj},\cdots,D_{jn}\}$(n 是决策属性所有可能取值数)，其中 D_{jj} 是指对任意的 $x,y\in D_{jj}$,x 和 y 决策属性的取值相同；对于 FD_j 的任意 D_{lj} 和 D_{lk} ($l\neq k$)，不存在 $x\in D_{lj}$ 和 $y\in D_{lk}$，使得 x 和 y 决策属性的取值相同。根据以上定义数据集 D 的不一致率 U 为：

$$U=\frac{\sum_{i=1}^{j}\left(|D_i|-\max\left(|D_{i1}|,|D_{i2}|,\cdots,|D_{in}|\right)\right)}{|D|} \tag{3-1}$$

上式中的$|D|$是数据集 D 中的实例数。数据集 D 在属性子集 S_j 上的不一致率是指 D 在属性子集 S_j 的投影 D'的不一致率。

一致性评价方法有大多数评价方法不具备的特性:单调性。单调性可以被有效地看成启发性信息，来大大减小搜索空间。对于数据集 D 的两个属性子集 S_i 和 S_j，如果 S_i 属于 S_j，则$(US_i)\geqslant(US_j)$。在对属性子集空间进行搜索的过程中，假设已经对 S_j 评价过，且知道 S_j 不满足一致性要求，如果下一个要评价的是 S_i，而 S_i 属于 S_j，没必要对 S_i 进行评价就可以知道 S_i 也是不满足一致性要求的；同样在搜索的过程中，假设已经对 S_i 评价过，且知道 S_i 满足一致性要求，如果下一个要评价的是 S_j，而 S_i 属于 S_j，则没必要对 S_j 进行评价就可以知道 S_j 也是满足一致性要求的。

3.2 基于 PCA 的网络流量特征选择

主成分分析[35](*Principal Component Analysis*, 简称PCA)是常用的特征选择技术，是基于变量的协方差矩阵对信息进行处理、压缩和抽提的有效方法。其目的是希望找到一个或少数几个综合指标来代替原来的统计指标，而且希望新的综合指标能够尽可能地保留原有信息，并具有最大的方差。也即压缩变量个数，用

较少的变量去解释原始数据中的大部分变量，剔除冗余信息。从而将许多相关性很高的变量转化成个数较少、能解释大部分原始数据方差且彼此互相独立的几个新变量，也就是所谓的主成分。简而言之，就是用新的变量替代原来的变量，实现特征空间的维度降低，而且新的变量能够完全代表原变量的信息。

3.2.1 PCA 降维原理

由 2.2.1 流的定义，$F=\{F_1,\ldots,F_i,\ldots,F_N\}(i=1,2,\ldots,N)$表示样本流集合，$F_i$ 表示第 i 条流，其中 $F_i=\{f_{i1},\ldots,f_{ij},\ldots,f_{iM}\}(j=1,2,\ldots,M)$，$F_{ij}$ 表示 i^{th} 流的 j^{th} 的属性值。上述表示中，N 是样本流的数目，M 是流的属性数目。在我们的研究中，流看成是一个 M 维向量。那么 N 个流样本就可以表示为 $M\times N$ 训练样本流的矩阵，用 F_Matrix 表示。

$$F\_Matrix=\begin{bmatrix} f_{11} f_{21} \cdots f_{N1} \\ f_{12} f_{22} \cdots f_{N2} \\ \vdots \quad \vdots \quad \cdots \vdots \\ f_{1M} \ f_{2M} \cdots f_{NM} \end{bmatrix} \tag{3-2}$$

然后计算 F_Matrix 的协方差矩阵。

令 $\overline{FF}=\frac{1}{N}\sum_{i=1}^{N}F_i$，$\Phi_i=F_i-\overline{FF}$，$A=[\Phi_1,\Phi_2,\ldots,\Phi_n]$，则对应矩阵 F_Matrix 的协方差矩阵为：

$$Q=\frac{1}{N-1}\sum_{i=1}^{N}\Phi_i\Phi_i^T=AA^T \tag{3-3}$$

这里 Q 是 $N\times N$ 矩阵。于是，可以求出矩阵 Q 的特征值λ_j $(j=1,2,\ldots,M)$，并按其大小顺序排列好，$\lambda_1\geqslant\lambda_2\geqslant\ldots\lambda_M$。同样，我们可以计算出特征值对应的特征向量，$\mu_j(1\leqslant j\leqslant M)$。$Q$ 是对称的，这样$\mu_1,\ldots,\mu_j,\ldots,\mu_M$ 为主成分，并且可以线性表示为 $F_i-\overline{FF}$，即

$$F_i-\overline{FF}=\sum_{j=1}^{M}b_j\mu_j \tag{3-4}$$

最后，就要确定优选的特征值了。即保留前 H 个值最大的特征值，并使得式(3-5)成立

$$\sum_{j=1}^{H}\lambda_j\Big/\sum_{j=1}^{M}\lambda_j>Threshold \tag{3-5}$$

这里 $Threshold$ 取值一般为大于 0.85。

3.2.2 基于 PCA 的网络流量特征选择算法

根据 3.2.1 介绍的 PCA 算法原理，我们设计了基于 PCA 的网络流量特征选择算法。算法如下：

算法 3-1 *PCA(F, Threshold)*

输入：训练样本流集合 *F*(用候选特征表示)及阈值 *Threshold*;

输出：流较优的特征子集

步骤：

Step1: 训练样本流的矩阵产生。将 F_i 表示为$[f_{i1},\ldots,f_{ij},\ldots, f_{iM}]^T$，那么 F 可以表示为$[F_1,\ldots,F_i,\ldots,F_N]^T$。这样可以产生一个 $M\times N$ 训练样本流的矩阵，用 *F_Matrix* 表示。

Step2: 计算 *F_Matrix* 的协方差矩阵 Q，如式(3-3)所示。

Step3: 计算协方差矩阵 Q 的特征值和特征向量。

求出 Q 的特征值λ_j $(j=1,2,\ldots,M)$,并将特征值按大小顺序排列,$\lambda_1\geqslant\lambda_2\geqslant\ldots\lambda_M$ 及特征值对应的特征向量$\mu_j(1\leqslant j\leqslant M)$。$Q$ 是对称的，$\mu_1,\ldots,\mu_j,\ldots,\mu_M$ 即为主成分，$\mu_1,\ldots,\mu_j,\ldots,\mu_M$ 线性相关，可以表示为式(3-4)；

Step4: 确定优选的特征值。保留 H 个最大的特征值，并使得 *Threshold* 取值满足式(3-5)。

3.2.3 PCA 特征选择算法实验

实验中，以表 2-2 中的候选特征作为流的初始特征，以解析好的样本流集合为对象，运用 PCA 方法进行特征选择实验。实验中，使用的各种类型的样本流总数为 2000 条，数据集中的应用类型包含有 *WWW*，*DNS*，*POP3*，*SMTP*，*FTP*，*SOCKS* 等常见的应用类型和 *BitTorrent*，*BtSprit* 等几种 P2P 类型，并取阈值 *Threshold* =0.90。

由 3.2.1，首先计算协方差矩阵 Q 的特征值，并按由大到小的顺序排列，见图 3-1，其中最大的特征值为 11.27808，最小的为 0。

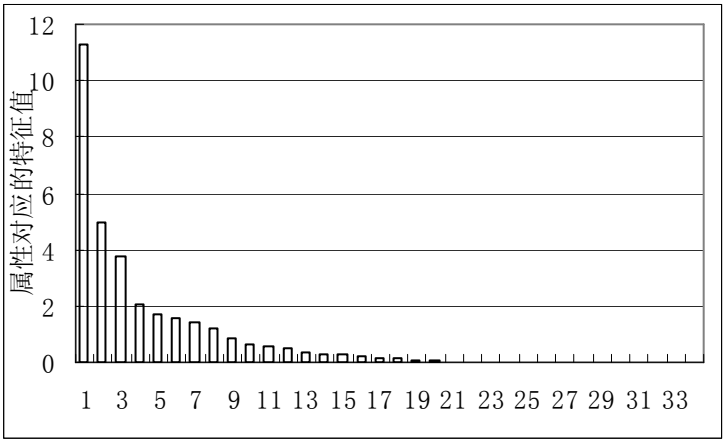

图 3-1 流属性对应的特征值

表 3-1 值最大的十个特征值

序号	特征值	所占比例	累计比例
1	11.27808	0.35244	0.35244
2	4.9383	0.15432	0.50676
3	3.73123	0.1166	0.62336
4	2.04497	0.06391	0.68727
5	1.69398	0.05294	0.7402
6	1.54472	0.04827	0.78848
7	1.40537	0.04392	0.83239
8	1.20868	0.03777	0.87017
9	0.84087	0.02628	0.89644
10	0.62026	0.01938	0.91583

通过计算，$\sum_{j=1}^{34}\lambda_j$ =31.9465，前十个特征值之和为29.3032，根据 $\sum_{j=1}^{H}\lambda_j \Big/ \sum_{j=1}^{M}\lambda_j > Threshold$ 的原则，前十个特征值的比重达到了91.583%，具体信息见表3-1。选取与它们对应的特征代替原始的特征集，于是这些被选的特征即为较优特征子集。这些特征是*fPackets*，*fBytes*， *bPackets*， *bBytes*，*minFpktLen*，*maxFpktLen*，*meanLenFsum*，*minBpktLen*，*maxBpktLen*，*meanLenBsum*。

3.3 基于信息增益的网络流量特征分组及选择

1948年,Shannon提出并发展了信息论,研究以数学的方法来度量信息,提出了信息增益等基本概念,并得到广泛的应用。信息增益又称为互信息。样本中属性的信息增益越大,其包含的信息量也越大。也就是说,在特征选择时,应计算各个属性的信息增益,具有最高信息增益值的属性是给定集合中具有最高区分度的属性。

3.3.1 基于信息增益的网络流量特征分组及选择基础

特征分组是进行特征选择及降维的有效方法之一,其主要思想是使用约定的相似性度量，对特征进行分组，使得分在同一组的特征具有很强的相似性，而不同组的特征具有较大的差异，然候选出各组的代表特征作为精简后的特征子集，从而在一定程度上消除特征冗余，实现降维。在本文中，采用信息增益作为特征之间的相似性度量，并采用一种基于密度的分组方法进行特征分组，实现特征的精简。信息增益[36,37](*Information Gain*)是指期望信息或者信息熵的有效减少量，根据它能够确定在什么样的层次上选择什么样的变量来分类。

网络流量的测量数据来自流量信息采集,包括了时间特征、流中报文的标志位信息特征，流的报文个数及大小特征等属性。这样,把样本数据当作离散信息源,把样本数据中的各个属性看作是一组随机事件(向量空间),就可以对它的信

息熵[38]进行分析。

随机事件 $X=\{n_i,\ i=1,\ \cdots,N\}$表示在测量数据中属性 i 发生了 n_i 次，那么信息熵的定义如下：

$$H(X)=-\sum_{i=1}^{N}p(x_i)\log_2(x_i) \tag{3-6}$$

其中 $p(x_i)=(n_i/S)$，$S=\sum n_i$ 表示某个属性 i 发生的总次数。同样，$Y=\{n_j, j=1, \cdots,N\}$ 表示在测量数据中属性 j 发生了 n_j 次，那么通过观测随机变量 Y，随机变量 X 的信息熵变为:

$$H(X\mid Y)=-\sum_{j=1}^{N}p(y_j)\sum_{j=1}^{N}p(x_i\mid y_j)\log_2\left(p\left(x_i\mid y_j\right)\right) \tag{3-7}$$

其中 $p(y_j)=(n_j/S)$，$S=\sum n_j$ 表示某个 j 属性发生的总次数。$p(x_i)$代表代表随机变量 X 的先验概率, $p(x_i|y_j)$ 代表观测到随机变量 Y 后随机变量 X 的后验概率。引入随机变量 Y 的信息后，随机变量 X 的信息熵(即互信息)$H(X|Y)\leq H(X)$，　X 的不确定程度会变小或保持不变。若 Y 与 X 不相关，　$H(X|Y)=H(X)$；若 Y 与 X 相关，则 $H(X|Y)<H(X)$, 而差值 $H(X)-H(X|Y)$越大, Y 与 X 的相关性越强。因此定义信息增益 $IG(X|Y)$为：

$$IG(X\mid Y)=H(X)-H(X\mid Y) \tag{3-8}$$

由此，我们根据以上三式，可以计算出待选特征的信息增益值。而且, 可以证明信息增益具有对称性, 即 $IG(X|Y)=IG(Y|X)$ 。另外, 为了对信息增益进行归一化, 可采用式(3-9) , 同理 y 有 $SU(X,Y)=SU(Y,X)$。

$$SU(X,Y)=2\left[\frac{IG(X|Y)}{H(X)+H(Y)}\right] \tag{3-9}$$

在相似度定义的基础上, 就可以基于特征之间的相似度进行特征分组。本研究首先针对每个特征分别统计与该特征相似度大于某个阈值的其它特征的个数, 然后找出与该特征相似度大于指定阈值的其它特征数最大的特征, 将该特征及与其相似度大于指定阈值的其它特征归为一组(该特征即为此组特征的代表特征)；然后将该组特征从原特征集合中删除, 继续上述过程, 直至所有特征都被归到某一特定组为止；最后各组特征的代表特征即形成精简后的特征子集, 具体算法流程如算法 3-2 所示。

3.3.2 基于信息增益的网络流量特征分组及选择算法

算法 3-2 *Partition Features* (F,δ)

输入:原始特征集合 F, 阈值 δ

输出:精简后的特征子集 *FS*

步骤：

Step 1 初始化特征子集 $FS=\{\}$，$FW=F$；

Step 2 根据式(3-6) ～ (3-9) 计算每个特征与其它特征的信息增益，形成特征相似度(信息增益) 矩阵 SU；

Step 3 针对每个特征 f_i 在特征集合 FW 中搜索与其信息增益大于阈值 δ 的其它特征，形成特征子集 F_i 如下：$F_i=\{f_k|SU(f_i,f_k)\geq\delta, f_k\in FW, k\neq i\}$ $(i=1,2,\cdots,|FW|)$；

Step 4 令 $S_i=|F_i|,(i=1,2,\cdots,|FW|)$；

Step 5 令 $S_m=Max(S_i)$，(i=1，2，…，|FW|)；

Step 6 将特征 f_m 选入代表特征子集 FS，即 $FS=FS\{f_m\}$；

Step 7 从特征集合 FW 中剔除特征子集 F_m 及特征 f_m，即 $FW=FW-(F_m\cup\{f_m\})$；

Step 8 重复步骤(3)～(7)，直至 $FW=\{\}$时结束，输出精简的代表特征集合 FS 。

3.3.3 基于信息增益的网络流量特征分组及选择实验

我们还是以表 2-2 中的候选特征作为流的初始特征，以采集的样本流集合为对象，从样本流中随机抽取 14200 条记录用于实验。这些流样本中包含了九种应用类型。具体数据信息见表 3-2。实验时，我们根据公式(3-6)～(3-8)计算每个流属性的信息增益值，并将这些属性的信息增益值用图 3-2 表示。然后将阈值设定 δ=0.5，对网络流量的特征分组，实现降维。

表 3-2 数据集中应用类型的分布

应用类型	流的数量	比例(%)
WWW	1000	25.0
DNS	200	5.0
POP3	200	5.0
SMTP	300	7.5
BitTorrent	1000	25.0
BtSprit	100	2.5
FTP	1000	25.0
Xunlei	100	2.5
SOCKS	100	2.5
总数	4000	100

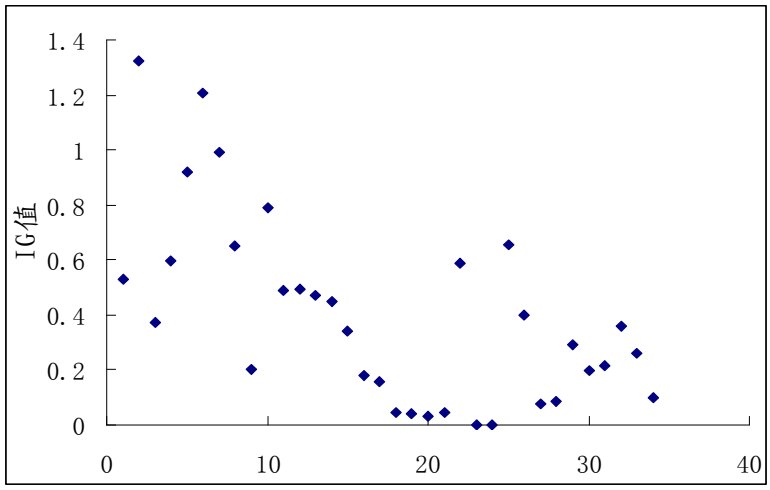

图 3-2 34 个特征 IG 值分布图

图 3-2 中序号与表 2-2 所列出的特征一一对应。从图中可以看出 IG 值最大的是 *fBytes* 达到了 1.3224，而最小的 IG 值为 0。我们按照 IG 值的大小顺序，选择值大于阈值 δ 的特征作为一组。IG 值大于阈值 δ 的特征一共有十个，这十个特征详细信息见表 3-3。

表 3-3 信息增益值最大的 10 个特征

序号	特征名称	IG 值	描述
1	*fBytes*	1.3224	前向总的报文的大小
2	*maxFpktLen*	1.209	前向报文的最大长度
3	*meanLenFsum*	0.9907	前向报文的平均长度
4	*minFpktLen*	0.9204	前向报文的最小长度
5	*maxBpktLen*	0.7094	后向报文的最大长度
6	*fAckCnt*	0.655	前向报文中 ACK 标志位
7	*stdLenFsqsum*	0.6509	前向报文的均方差
8	*bBytes*	0.5952	后向总的报文的大小
9	*avePktPerSecond*	0.5871	前向总的报文的个数
10	*fPackets*	0.5283	前向总的报文的个数

3.4 小结

本章介绍了特征选择的基础知识，阐述了特征选择的一般方法。并使用基于 PCA 的特征选择算法和基于信息增益的特征分组和选择两种方法对网络流量数据的特征进行了筛选，得到各自的最优特征子集。这两个特征子集将在第四章中结合机器学习算法进行实验，验证两种特征选择算法的有效性。

第四章 基于聚类的网络流量分类及实验测评

聚类技术作为机器学习中重要研究内容，在图像、医药、生物得到了广泛的研究与应用。聚类的目的是把性质相似的空间数据聚集在一起，而让不同的聚类(簇)之间的数据差异很大。本章主要介绍使用聚类方法对网络流量数据进行聚类与分类的研究。使用基于距离的 K-Means 算法和基于密度的 DBSCAN 的算法对网络流量数据进行聚类分析，然后根据聚类的结果构建分类器，对网络流量数据进行分类。为了验证方法的有效性，本章进行了大量试验来评测分类器的性能。

4.1 聚类技术概述

迄今为止，聚类还没有一个学术界公认的定义。这里给出 *Everitt*[39]在 1974 年关于聚类所下的定义：一个类(簇)内的实体是相似的，不同类(簇)的实体是不相似的；一个类(簇)是测试空间中点的会聚，同一类(簇)的任意两点之间的距离小于不同类(簇)的任意两个点间的距离；类(簇)可以描述为一个包含密度相对较高的点的多维空间的连通区域，他们借助包含密度相对较低的点的区域相分离。

典型的聚类过程主要包括数据(或称之为样本或模式)准备、特征选择和特征提取、相似度计算、聚类(或分组)、对聚类结果进行有效性评估等步骤[39,40,41,42]。即聚类过程可以描述为：

(1) 数据准备，包括特征标准化和降维。

(2) 特征选择，从最初的特征中选择最有效的特征,并将其存储于向量中。

(3) 特征提取，通过对所选择的特征进行转换形成新的突出特征。

(4) 聚类(或分组)，首先选择合适特征类型的某种距离函数(或构造新的距离函数)进行相似程度的度量；而后执行聚类或分组。

(5) 聚类结果评估，是指对聚类结果进行评估。

4.1.1 聚类算法的类别

没有任何一种聚类技术(聚类算法)可以普遍适用于揭示各种多维数据所呈现出来的多种多样的结构[42]。根据数据在聚类中的积聚规则以及应用这些规则的方法，有多种聚类算法。聚类算法有多种分类方法，本文将聚类算法大致分为层次聚类算法、划分式聚类算法、基于密度的聚类算法和网格的聚类算法和其他聚类算法。

4.1.2 层次聚类

层次聚类算法又称为树聚类算法[43, 44]，它使用数据的联接规则，透过一种层次架构方式，反复将数据进行分裂或聚合，以形成一个层次序列的聚类问题解。层次聚类算法由树状结构的底部开始逐层向上进行聚合，假定样本集 $S=\{F_1, F_2, \cdots, F_n\}$共有 n 个样本。算法表述为：

算法 4-1： 层次聚类算法

输入： 样本集 S

输出： 满足条件的类

Step1： 初始化，将每个样本 F_i 作为一类； /*共生成 n 个类*/

Step2： 找出距离最近的两个类，$dist(F_r, F_k) = \min_{\forall F_u, F_v \in S, F_u \neq F_v} dis(F_u, F_v)$；/*从现有类中找出两个最相似的类(距离最近的类)*/

Step3： 合并这两个类，即将类 F_r，F_k 合并一个新类 F_{rk}； /*将现有类数减一*/

Step4： 若所有的样本都属于同一个类，则终止本算法；否则，返回 *Step2*。

算法中*dist*()代表两个类中的距离，两个类之间距离的度量方法是传统层次聚合算法的重要组成部分，它主要包括两个重要参数相似性度量方法和联接规则。这里采用欧式距离(*Euclidean distance*)作为相似性度量方法, 联接规则主要包括单联接规则、完全联接规则、类(簇)间平均联接规则、类(簇)内平均联接规则和沃德法。这几种联接规则可以定义如下[45](其中||*x-y*||是欧几里德范数，n_i和n_k分别指类F_r和F_k中的样本个数，$C(n_i+n_k,2)$表示从n_i+n_k个元素中抽出两个元素的不同组合方法总数)。

单联接聚合规则： $dist(F_r, F_k) = \min_{x \in F_r, y \in F_k} \| x - y \|$;

全联接聚合规则： $dist(F_i, F_k) = \min_{x \in F_i, y \in F_k} \| x - y \|$;

类(簇)间平均联接聚合规则： $dist(F_i, F_k) = (1/n_i n_k) \Sigma_{x \in F_i, y \in F_k} \|x-y\|$;

类(簇)内平均联接聚合规则： $dist(F_i, F_k) = (1/C(n_i + n_k, 2) \Sigma_{x \in F_i, y \in F_k} \|x-y\|$

沃德法： $dist(F_i, F_k) = (1/(n_i + n_k)) \Sigma_{x \in (F_i, F_k)} \|x-n\|^2$，其中$n$是融合聚类的中心。

层次聚类算法目前具有代表性的有*CURE*[46]、*ROCK*[47] 、*BIRCH*[48]等。*CURE*算法以聚类间的相似度为合并依据，它有别于以往中心点或重心点的聚类算法，是以一个代表点来取代一个簇，而*MST*(*Minimum spanning tree*)方法以全部的数据样本皆为代表点。*CURE*算法则是固定选择*C*个点(*2≤C≤N*，*N*为全部样本集之和)代表簇，接着将全部的代表点向重心收缩以此将相似的簇合并，直到达到需要为止。CURE算法适合于大型数据集，可以发现任意形状的簇，对离群值处理较健全，但不适合处理类别属性的数据。

4.1.3 划分式聚类

划分式聚类算法是发展最早的聚类技术，这一类算法用户必须预先决定要分割的聚类数目，再以重心点(*Centroid–based*)或中心点(*Mediod-based*)的方式进行分群。划分式聚类算法是以距离作为评估标准，通常有曼哈顿距离(*Manhattan distance*)与欧几里德距离(*Euclidean distance*)，目前较重要的方法有*PAM*[49]、*CLARA*[49]、*CLARANS*[50] 以及*K-Means*[51] 。

*K-Means*算法是最典型的以中心基础的划分式聚类算法，它是以群的中心为聚类的代表点，但代表点不一定是类(簇)中的一点，所以不一定可以找到最佳的聚类。然而，此方法所得到的聚类质量很容易受到噪声(*Noise*)或是离群值(*Outliers*)所影响。另一种方法是以中心点作为代表点(如*PAM*算法)，这些聚类技术对于小型的数据集都有不错的处理能力，但是随着数据集的增加，处理的效率也越来越差。所以，通常在处理大型数据集的时候采用取样的方式来解决。然而取样的算法也会受到样本的数量及取样方法所影响，倘若样本数量太少，则聚类的结果不足以代表整个数据集的分布情况及意义；再者，若取样的方法不佳，则会影响到聚类的质量。*CLARANS*是架构于*PAM*与*CLARA*上的以中心点为基础的划分式聚类算法，它是第一个针对于空间数据集所设的算法，但只能发现简易形状的分布，也无法有效率地针对高维数据进行聚类分析。

4.1.4 基于密度的聚类

在一个样本数据集中，假设有某些数据点分布密度相当密集，则这些数据点形成一个群聚，也就是说，在群聚的内部的分布密度应大于群集外的数据分布密度。目前，较为重要的方法有 *DBSCAN*[52]，*OPTICS*[53]。

DBSCAN 算法是较早利用密度概念处理聚类问题的算法，使用者需要设定邻域半径(*Eps*)以及至少在该邻域的数据点数(*MinPts*)两个参数，只要在半径 *Eps* 邻域内的数据点数大于阈值 *MinPts*，则形成类(簇)。接着 *DBSCAN* 开始由核心点向外扩展群聚范围，由使用者所定的 *Eps* 的区域可直接包含且形成群聚的数据点，称为密度直接可达(*Directly density-reachable*)。藉由边缘点向外扩展可间接包含到的数据点，称为密度间接可达(*Density-reachable*)。利用密度的方法的特性，*DBSCAN* 可以有效分辨并控制离群值，但是对于处理任意形状群聚的辨别效果较差，尤其在参数的设定，会严重影响到聚类的结果。这些参数必须要根据经验或是观测判断，使用者很难设定参数大小。

4.1.5 基于网格的聚类

基于网格的聚类算法将数据空间量化成许多格子，大量的减少聚类的时间。在这类算法中，具有代表性的*STING*[54]、*WaveCluster*[55]。

*STING*算法是将数据空间切割成格子状，其聚类方式是由上而下的，利用广度搜索将格子内的簇合并。*STING*搜索存在格子的统计数据，然后进行聚类。其缺点是分群边缘的形状不是水平就是垂直，尽管有快速聚类的特点，但会有损其聚类的质量。此外，*STING*是呈现阶层式架构，较高的层会储存较低层次的数据，所以查询速度非常快，其时间复杂度为$O(k)$（k为最底层格子的数）。

4.1.6 聚类算法比较与参数分析

以上几种类型的聚类算法因为聚类的方式不同，所以皆存在一些限制与缺点。进来有些学者尝试以混合的方法，结合多种聚类技术，取其算法的有点，如*BRIDGE*，它有效的结合 *K-means* 快速容易执行且可以找到最佳的聚类的特性，与 *DBSCAN* 可以摒除受噪声影响的优点，成为更好的聚类算法。聚类算法因为有先天的限制或缺点，我们可以借助几项评价标准来判断聚类算法的优劣。

① 可扩展性(*Scalabilty*): 聚类算法必需能有效的处理大型数据库。

② 需要极少的领域知识去决定输入参数：大多数聚类算在进行聚类前都会要求用户输入参数，而这些参数会影响聚类的质量和结果，用户可能要通过查阅使用手册或是专业领域的知识才能找出最合适的参数之，这不只是造成用户的困扰，也容易因为输入不当的参数影响聚类的质量。

③ 能处理不同类型的属性：许多聚类算法被设计为适合处理数值型或类别型的数据，然而在许多应用上，数据库中可能包含各种类型的属性，故聚类算法必须具备处理不同类型属性的能力。

④ 能处理的维度：一个数据库中可能包含多个维度与属性，许多聚类算法对于低维度的数据有良好的聚类能力，而对于高维度的数据的处理能力不佳，好的聚类算法必须具备处理多维或高维数据的能力。

⑤ 能辨别任意形状的聚类：一般的聚类算法是以聚类来衡量数据的相似度，这一类型的算法对于任意形状的聚类辨别能力有限。好的聚类算法必须要能很有效而且正确的发现任意形状的聚类。

⑥ 处理噪声的能力：聚类算法必须要有能去除或是过滤噪声数据的能力，在实际的数据库中，数据库中可能包含噪声，遗漏值(*missing Value*)，这会影响到聚类结果的质量。

⑦ 可解释性：用户通过聚类方式所产生的聚类结果，必须可以让用户理解其中所代表的意义，进而将该聚类进行定义后再使用。

⑧ 对于数据的输入不敏感：有些聚类算法对数据的输入顺序有影响，不同的数据输入顺序会产生不同的聚类结果。

表 4-1 列出了一些重要的聚类算法对适用数据类型和参数需求表。

表 4-1 重要聚类算法与参数需求比较表

算法类型	算法名称	复杂度	适用数据类型	是否需要参数
划分式聚类算法	*k-means*	$O(nkt)$，n 代表数据总数，k 为分群数，t 为程序循环次数。	数值型	分类数 k
	PAM	$O((kn-k)^2)$	数值型	分类数 k
	CLARA	$O(k10-k)^2+kn-k))$	数值型	分类数 k
	CLARANS	$O(kn)^2$	空间数据	最大邻居数(*Maxneighbor*)局部最小值(*Numlocal*)
层次式聚类算法	*CURE*	低维度：$O(n^2)$ 高维度 $O(n^2logn)$	数值型	收缩参数 α
	ROCK	$O(n^2+nm_mm_a+n^2logn)$，m_m 最大邻居数，m_a 为平均邻居数。	类别型	需要参数
密度聚类算法	*DBSCAN*	$O(nlogn)$	空间数据	临域半径(*Eps*)，数据点数阈值(*MinPts*)
网格聚类算法	*CLIQUE*	$O(cd+nd)$ d 为维度，c 为场数	空间数据	ξ 代表格子的宽度，τ 表示投影到该格子数量的阈值
	STING	$O(k)$,k 为最底层格子数	空间数据	需要参数

4.2 一种基于 DBSCAN 的网络流量分类

*DBSCANDensity-Based Spatial Clustering of Application with Noise)*算法由 Martin. Ester等人[52]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*(neighborhood)*内包含的数据点数目必须不小于某一给定值(用*MinPts*表示)。在我们的研究中，将利用*DBSCAN*算法聚类得到簇构建网络流量分类器。

4.2.1 基于 DBSCAN 算法的网络流量聚类的相关定义

定义 4-1（**流的邻域**）流 F_i 的邻域是以该流为中心、以 *Eps* 为半径的超圆区域内包含的流集合，记作 $N_{Eps}(F_i)$, $N_{Eps}(F_i)=\{F_k \in F | dist(F_i,F_k) \leqq Eps\}$,$F$ 是样本流集合，

$dist(F_i,F_k)$是流 F_i 和 F_k 之间的距离，用式(4-1)来计算。

$$dist(F_i,F_k)=\left[\sum_{j=1}^{M}\left(f_{ij}-f_{kj}\right)^2\right]^{1/2} \qquad (4-1)$$

定义 4-2（直接密度可达）给定 *Eps* 和 *MinPts*，若流 F_i 从 F_k 直接密度可达，则满足：

①F_i 处于 F_k 的临域中，即 $F_i \in N_{Eps}(F_k)$；

②F_k 是核心点，即$|(N_{Eps}F_k)| > MinPts$。

定义 4-3（密度可达到）给定流集合 F，当存在一个流对象链 $F_1,F_2,\ldots,F_N$，$F_1=F_k$，$F_N=F_i$，对于 F_{i+1} 是 F_i 关于 *Eps*、*MinPts* 直接密度可达的，则称 F_i 从 F_k 关于 *Eps*、*MinPts* 密度可达。

定义 4-4（密度连接）如果样本流集合 F 中存在一条流 F_l 使得流 F_j 和 F_k 是从 F_l 关于 *Eps*、*MinPts* 密度可达，那么流 F_j 和 F_k 关于 *Eps*、*MinPts* 密度连接。

定义 4-5(簇 *Cluster*) F 是样本流集合，簇 L_p 是 F 的一个关于 *Eps*、*MinPts* 的非空子集，当且仅当 L_p 满足：

①对于$\forall F_i,F_k$，若 $F_i \in L_n$，且 F_k 从 F_i 密度可达，则；$F_k \in L_n$

②对于$\forall F_i,F_k \in L_p$，则 F_i 从 F_k 是关于 *Eps*、*MinPts* 密度连接的。

定义4-6 (噪声*Noise*))设$L_1,\cdots,L_n,\cdots L_p$是样本流集合$F$中满足参数$Eps_n$，$MinPts_n(p=1,\cdots,P)$的簇。则定义噪声为，样本流集合中不属于任何簇$L_n$的点；即$Noise=\{F_o \in F|\forall p:F_o \notin L_p\}$。

4.2.2 基于 DBSCAN 的网络流量聚类方法

DBSCAN算法检查样本流集合F中每个流数据点的邻域$N_{Eps}(F_i)$。若一条流F_i的邻域$N_{Eps}(F_i)$包含多于 *MinPts*条流，就要创建包含流F_i的新簇(*Cluster*),并将F_i看成核对象流。然后根据这些核对象流，循环收集“直接密度可达”的流，其中可能涉及进行若干“密度可达”簇的合并。当各个簇再无新的流加入时聚类进程结束。

DBSCAN 算法聚类过程可以从流数据集中的任一条流开始，对于聚类结果没有任何影响。算法中要强调的是，只要一条流的 *Eps* 邻域的密度到达了 *MinPts*，才能成为一个核对象流。只有核对象流才可以将周围的流数据点聚成一个簇。

下面是基于 DBSCAN 的网络流量聚类方法的伪代码。

```
DBSCAN (SetOfPoints,Eps,MinPts)
    //SetOfPoint 表示待聚类的训练样本流集合
ClusterId:= nextId(NOISE)；
FOR i FROM 1 TO SetOfPoints.size DO
```

```
 Point:=SetOfPoints.get(i);
    IF Point.ClId:=UNCLASSIFIED THEN
    IF ExpandCluster(SetOfPoints,Point,
ClusterId,Eps,MinPts) THEN
        ClusterId:=nextId(ClusterId)
    END IF
    END IF
END FOR
END;//DBSCAN
```

上述伪代码中 *SetOfPoints* 表示待聚类的训练样本流集合，*Eps* 和 *MinPts* 是两个全局密度参数，这两个参数往往不能唯一的确定，它们要根据数据集的具体分布情况作合理的选择。函数 *SetOfPoints.get*(*i*)返回数据点集中的第 *i* 条流。

聚类函数 *ExpandCluster* 的伪代码如下：

```
ExpandCluster (SetOfPoints,Point,
            ClusterId,Eps,MinPts):Boolean;
 seeds:=SetOfPoints.regionQuery(Point,Eps);
 IF seeds.size<MinPts THEN //没有核对象流
 SetOfPoint.changeClId(Point,NOISE);
 RETURN False;
 ELSE //all point in seeds are
      //density reachable from Point
 SetOfPoints.changeClIds(seeds,ClId);
 seeds.delete(Point);
 WHILE seeds<>Empty DO
 currentP:=seeds.first();
result:=SetOfPoints.regionQuery(currentP,Eps);IF result.size>=MinPts THEN
 FOR i FROM 1 TO result.size DO
 resultP:=result.get(i);
 IF resultP.ClIdIN{UNCLASSIFIED,NOISE} THEN
  IF resultP.ClId:=UNCLASSIFIED THEN
   seeds.append(resultP);
  END IF;
  SetOfPoints.changeClIds(seeds,ClId);
 END IF;//UNCLASSIFIED or NOISE
```

END FOR;
END IF;//result.size>=MinPts
seeds.delete(currentP);
END WHILE;//seeds<>Empty
RETURN TRUE;
END IF;
END;

上述伪代码中 *SetOfPoints.regionQuery(Point, Eps)*返回对 *Point* 这一点周围邻域内的所有点。

4.3 基于 K-Means 算法的网络流量分类

4.3.1 经典 K-Means 算法

*K-Means*聚类算法具有简单快速等特点，是一种典型的基于划分的聚类方法。该算法的核心思想是找出K 个聚类中心C_1，C_2，…C_k使得每一个数据点X_i 和与其最近的聚类中心C_γ 的平方距离和最小化(该平方距离和被称为偏差D).

设 K 是 *K-Means* 算法的输入参数，代表该算法在流样本数据集上分割并计算后输出的簇的数量。流样本数据集是 n 个模式(*Patterns*)组成的，模式与数据点的概念是相同的在初始化时，根据输入的参数 K 随机地从 n 个模式$\{i_1, i_2, \cdots, i_n\}$中找出 K 个原型(*Prototypes*)$\{W_1,W_2,\ldots,W_k\}$。因此 $W_j=i_l, j \in \{1,2,\cdots,K\}$，$l \in \{1,2,\cdots,n\}$。聚类的质量是由式(4-2)的错误函数确定

$$E=\sum_{j=1}^{k}\sum_{i\in c_j}\left|i_1-W_j\right|^2 \tag{4-2}$$

合适的 K 值选择是一个比较困难的问题。通常，这与问题域相关，用户需选择若干个 K 试验。下面我们给出 *K-Means* 算法：

算法 4-2：*The K-Means Algorithm*

输入：训练样本流集合 F 和簇的个数 K.

输出：簇 $C_1,C_2,\ldots,C_k$.

步骤：

Step1: 初始化，输入 F 并随机指定 K 个聚类中心($\gamma_1, \gamma_2,\ldots\gamma_k$)；

Step2: 分配每个流 F_i，对每个样本流 F_i 找到离它最近的聚类中心 γ_v，并将其分配到 γ_v 所标明的簇中；

Step3： 修正聚类中心；

Step4： 计算式(4-2)所示的错误函数；

Step5： 如果 E 值收敛，返回($C_1,C_2,\ldots,C_k$)；否则，返回到 *Step2*。

K-Means算法的优点是能对大型数据集进行高效分类,其计算复杂性为*O*(*tKmn*),其中,*t*为迭代次数,*K* 为聚类数,*m*为特征属性数,*n*为待分类的对象数,通常,*K*,*m*,*t*＜＜*n*。在对大型数据集聚类时,*K*-Means算法比层次聚类算法快得多。但算法通常会在获得一个局部最优值时终止；仅适合对数值型数据聚类。

4.3.2 改进的 K-Means 算法

由于经典K-Means算法的不足，以经典*K-Means* 算法为基础，研究者们提出了很多新的改进的*K-Means*算法，结合网络流量分类这里只介绍一致性保留*K-Means* 算法(*K-Means-CP*)。

一致性保留*K-Means* 算法(*K-means-CP*)由*Ding*等人提出[56]，他们在该算法中引入了模式识别中的一个重要概念——最近邻一致性。他们将这个概念扩展到数据聚类，对一个簇中的任意数据点，要求它的*k*最近邻和*k*互最近邻都必须在该簇中。他们研究了簇的*k*最近邻一致性的性质，提出了*kNN*和*kMN*一致性强制和改进算法，并提出了将类*k*最近邻或类*k*互最近邻一致性作为数据聚类的一种重要质量度量方法。他们选用互联网上20个新闻组数据集进行了实验，结果表明，*k*最近邻一致性、*k*互最近邻一致性以及算法聚类的正确率都得到显著改善。同时,这也表明局部一致性信息可帮助全局聚类目标函数优化。算法如下：

算法 4-3：*The K-Means-CP Algorithm*

输入: 训练样本流集合 *F* 和簇的个数 *K*.

输出: 簇 $C_1,C_2,\ldots,C_k$.

步骤:

Step1: 初始化，输入 *F* 并随机指定 *K* 个点作为初始的簇中心($\gamma_1,\gamma_2,\ldots\gamma_k$)；

Step2: 分配一个近邻集 *S*；

/*将 *S* 分配到离其最近的簇 C_p 中， $p=\arg\min_{v=1,\cdots,k}\sum_{F_i\in S}(F_i-m_v)^2$ */

Step3： 更新簇中心，置 $m_v=\sum_{F_i\in C_v}F_i/n_v$；

/*更新聚类中心即质心，m_v 是类 C_v 的中心，$n_k=|C_k|$*/

Step4： 质心不再移动，则终止算法；否则返回 Step2。

/* 收敛判别式是 $J_{Km}=\sum_{v=1,\cdots,K}\sum_{F_i\in C_v}(F_i-m_v)^2$ */

4.4 簇所属的应用类别的确定及分类器的分类规则

4.4.1 簇所属的应用类别的确定

训练样本流集合 *F* 经过上述聚类方法，可得到多个簇以及这些簇的核对象流数据点、簇所包含的训练样本流。设聚类后的簇用 C_k 表示,用 Cf_k 表示簇的核对象点(可以看作簇的中心)，每个簇所包含的样本流数据集合用 $F^k(F^k\in F)$表示，其

中 k=1,2,…,K，K 表示簇的数目。在 2.2.1 的介绍中，L={L_1,…, L_p,…, L_P}表示流的应用类型的标签集合。簇所属的类别采用简单多数投票的方式来确定，即按式(4-3)进行。

$$C_k \in \arg\max_{p=1}^{P}(vote(F^k \in L_p)) \tag{4-3}$$

式(4-3)中 *vote*()表示 F^k 中属于类别 *Lp* 的流数据的数目。

4.4.2 基于聚类的分类器的分类规则

利用聚类产生的簇及簇所对应的类别，分类器的分类规则如式(4-4)所示。

$$\begin{aligned} &If \quad F_i \quad is \quad the \quad closest \quad C_b, \\ &Then \quad F_i \in L_p \; and \quad C_b \in L_p \end{aligned} \tag{4-4}$$

利用式(4-4)的分类规则，对于待分类的流 F_x 可以按式(4-5)的判别式进行分类。

$$F_x \in \left(C_b = \arg\min_{k=1}^{K} dist\left(F_X, Cf_k\right) \rightarrow L_p \right) \tag{4-5}$$

式(4-5)中 $C_b \rightarrow L_p$ 表示簇所对应的应用类型，*dist*()表示欧氏距离。

4.5 基于聚类的分类器实验评测

4.5.1 分类器评测标准

任何一个领域中的技术研究都需要有与之相对应的一套评价标准，好的评价标准将引导这一领域沿着正确的方向发展。如何评价基于聚类方法的网络流量分类系统是一个关键问题。由于网络流量在不同的层面上的类别各不相同，因此评价基于聚类方法的网络流量分类系统性能的优劣并不容易，不失一般性我们使用查准率(*precision*)和总精确度(*overall accuracy*)指标来评价分类器的性能。其计算公式如式(4-6)和式(4-7)。

$$precision = \frac{TP}{(TP+FP)} \tag{4-6}$$

$$overall\ accuracy = \frac{\sum_{p=1}^{P} TP_p}{total\ number\ of\ flows} \tag{4-7}$$

式中 *TP*(*True Positives*)是指给定一个类别，正确分类的流数量，*FP*(*False Positives*)被错误分类的流的数量。

4.5.2 实验数据集

实验中我们从采集到的网络踪迹文件中选择数据子集，按照第二章介绍的方法解析为样本流，然后再使用 2.3 所介绍的方法对样本流进行标注。样本流中包含了 *WWW，DNS，POP3，SMTP，FTP，SOCKS* 等常见的应用类型和 *BitTorrent，BtSprit* 等几种 *P2P* 类型流量。此外，由原始踪迹文件解析成流的每个数据子集中包含 *WWW* 数量最多，一般所占比例达到了 70%以上。为了使机器学习算法对每种类型的流有较公正的辨别能力，所以有必要适当减少那些所占比重大的流量类型的数量。而且，考虑到机器学习算法的效率，我们把试验数据集控制在 8000～20000 个流样本之间。表 4-2 和表 4-3 给出了本文用于实验的两个样本数据集。

表 4-2 样本数据集 1 应用类型的分布

应用类型	流的数量	比例(%)
WWW	4200	29.58
DNS	400	2.82
POP3	500	3.52
SMTP	400	2.82
BitTorrent	3000	21.13
BtSprit	1200	8.45
FTP	3000	21.13
Xunlei	1200	8.45
SOCKS	300	2.11
总数	14200	100

表 4-3 样本数据集 2 应用类型的分布

应用类型	流的数量	比例(%)
BtTorrent	3443	35.81
FTP	1000	10.40
WEB	3251	33.81
SOCK	379	3.9
NNTP	88	0.92
SMTP	455	4.73
BtSprit	1003	10.43
总数	9619	100

4.5.3 DBSCAN 聚类与分类实验

我们使用 3.2 所介绍的特征选择算法对数据集 1 进行处理，得到一个最终的

试验数据集；同样我们应用 3.3 介绍的基于信息增益的特征选择算法对数据集 2 进行处理，得到第二个样本集。对于第一个样本数据集，采用 *DBSCAN* 聚类进行实验，实验中我们取 *Eps* 为 0.02，0.03，0.04，*MinPts* 为 4，8，12 进行实验。当 *MinPts*=4 时，聚类算法产生的簇明显多于其它值所产生的簇，这说明 *MinPts* 的越小越有利于簇的形成，形成了很多的小簇。图 4-1 是不同输入参数时的 *Overall Accuracy* 值，在 *MinPts*=4，*Eps* =0.04 时取得最大值达到 94.38%。从图 4-1 中可以看到，当 *MinPts* 确定时，*Eps* 增大，*Overall Accuracy* 的值也增大，说明 *DBSCAN* 聚类算法对 *Eps* 值越大且 *MinPts* 越小越容易形成聚类。

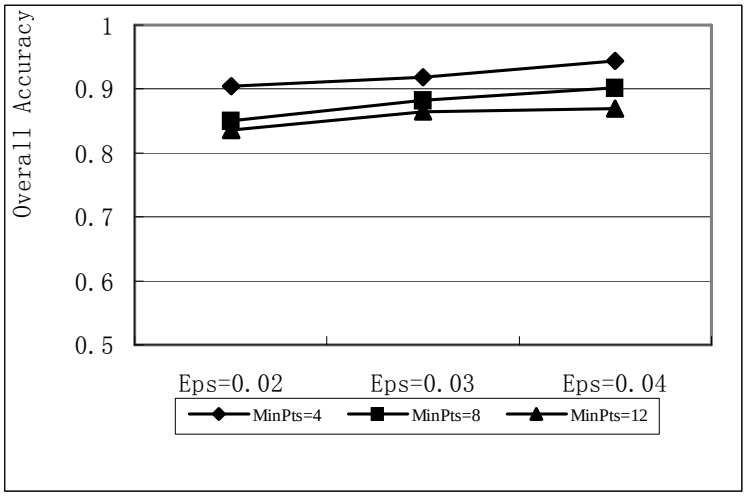

图 4-1 *Eps*、*MinPts* 取不同值数据集 1 的总精确度

同样，*Eps*、*MinPts* 的取值不同对查准率的值也有影响。图 4-2 给出的是 *Eps*=0.04 时，*Eps* 取不同值时各种应用类型的查准率。从图中可以看出，*Eps* 确定，*MinPts* 越小，各种类型的查准率越高。在 *Eps* 取其它值时，情况极为类似。

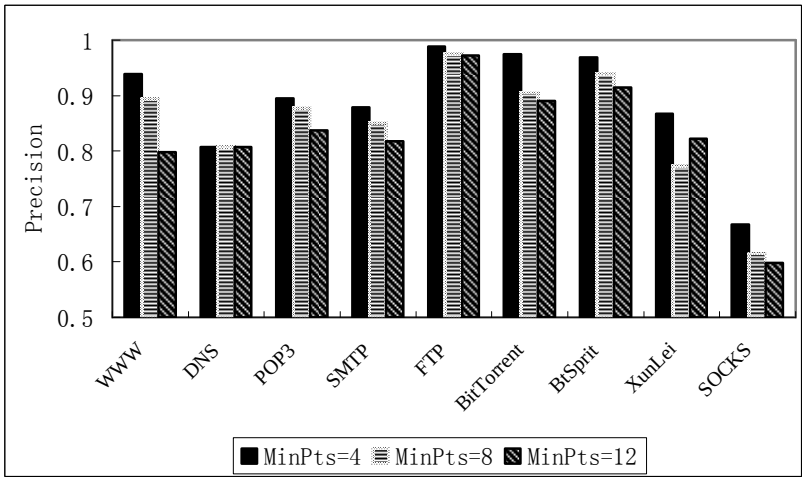

图 4-2 Eps=0.04，MinPts 取不同值的查准率

对于第二个数据集我们也作了相似的实验，图 4-3 是 *Eps* 和 *MinPts* 取不同

值是的 *Overall Accuracy*，从图中我们也可以得到与图 4-1 相似的结论：即 *Eps* 取值小且 *MinPts* 小时最容易形成聚类，聚类效果最好。图 4-4 给出了数据集 2 中各种类型在 *Eps*=0.02，*MinPts* 值不同时的 *Precision*。其中仍然以 *FTP* 的最高，达到 93%；而 *SOCKs* 的 *Precision* 最低仅 67%左右。

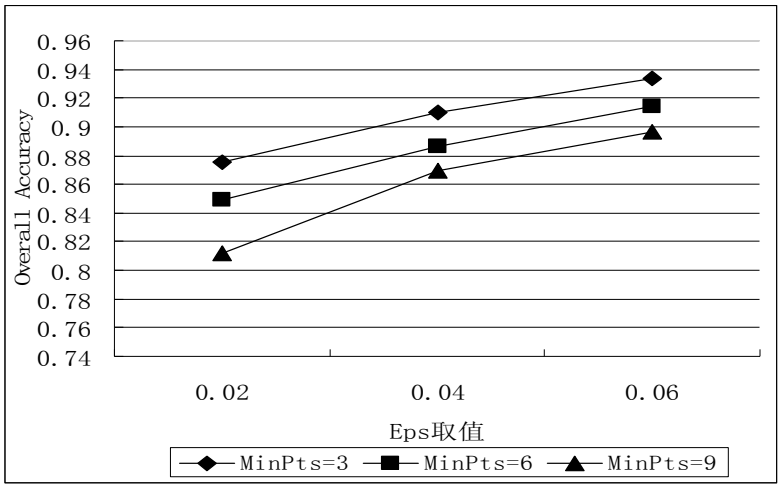

图 4-3 Eps、MinPts 取不同值数据集 2 的总精确度

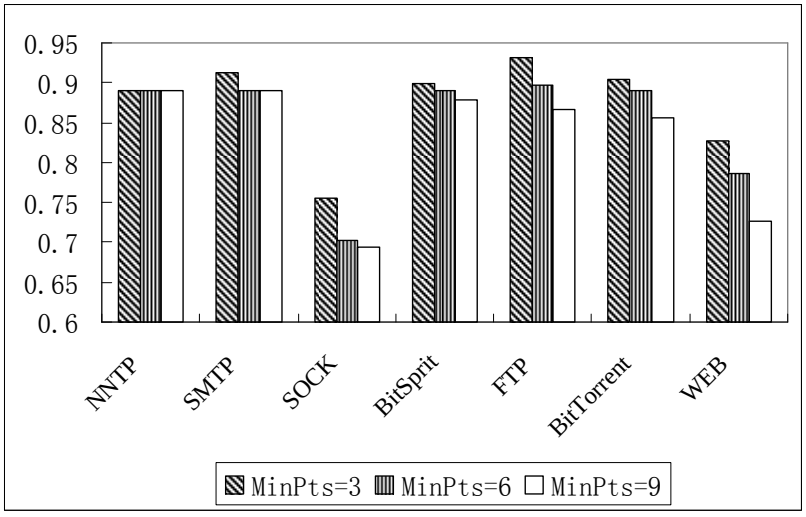

图 4-4 *Eps=0.02，MinPts* 取不同值数据集 2 的各种应用类型的 Precision

4.5.4 K-Means 聚类分类实验

同样，我们使用 *K-Means* 聚类算法对两个数据集进行实验，考查使用 *K-Means* 算法的聚类分类效果。*K-Means* 聚类算法对参数 *K* 是敏感的，*K* 值就是算法把样本集划分成簇的个数。在我们的试验中，我们希望每种应用类型至少产生一个簇；此外，由于某种应用类型内部流样本的差异，这样该应用类型就可能形成多个簇。为了得到一个合适的 *K* 值，在我们的实验中 *K* 的取值从 10 到 200，每次按 10 递增。因为，*K-Means* 算法最初是按照 *K* 的值随机分配簇，随着 *K* 的值的变化，*K-Means* 算法可能不会形成同样的聚类结果。所以，为了得到一个局部最优的结

果，*K-Means* 算法必须经过多次运算才能得到一个满足式(4-2)的错误函数最小值。图 4-5 给出了两个数据集在不同 *K* 值时的 *Overall Accuracy* 值。从图中可以看出，数据集 1 从 *K*=10 到 *K*=110 之间其 *Overall Accuracy* 值都在逐步增加，而当 *K*＞110 后就几乎不再变化；对于数据集 2，当 K＞60 后，*Overall Accuracy* 值就不再明显变化。这说明 *K* 值过大，可能导致过度拟合的可能性增大。图 4-6 给出了 *K*=40 时，数据集 2 中各种应用类型的 *Precision* 值。可以看出除了 *NNTP* 和 *SMTP* 两种类型的 *Precision* 较低，其他类型的都超过了 80%，而 *FTP* 和 *SOCK* 这两种类型甚至超过了 90%。此外，在实验中我们发现当 *K*＜40 时，*K-Means* 算法将 *NNTP* 和 *SMTP* 两种类型的测试样本流都划分到了 *BitTorrent* 类型中，当 *K*=30 时才发现少量 *SMTP* 样本形成单独的簇。这可能是由于训练样本不足形成的误分类情况。

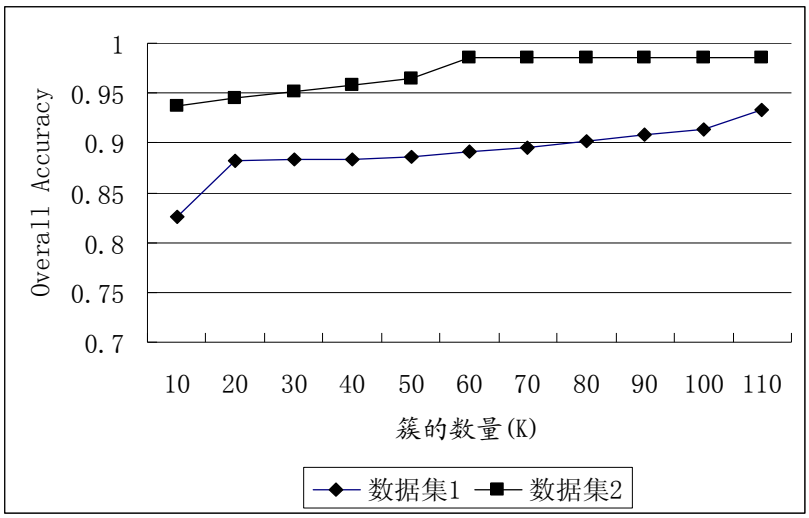

图 4-5 数据集 1、2 在不同 K 值下的 Overall Accuracy

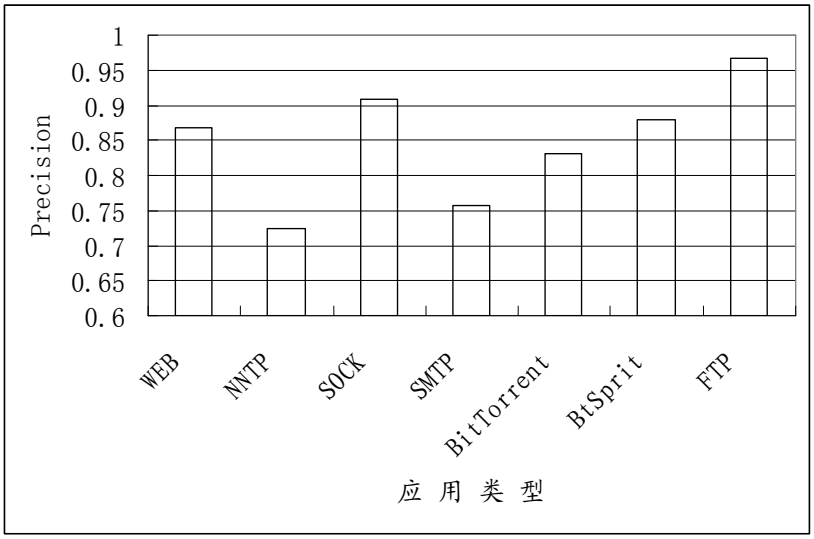

图 4-6 数据集 2 在 K=40 时各种类型的 Precision

4.6 小结

本章采用两种类型的聚类算法，基于密度的聚类算法——DBSCAN 和基于划分的聚类算法——K-Means 算法对两个数据集进行了聚类分类的实验。实验结果表明，本文提出的特征选择和聚类的分类规则的方案应用于网络流量分类其查准率和总精确度都达到了较高的水平。而且实现简单，算法效率高，是一个很好的网络流量分类研究方法，为网络流量分类的在线分类研究奠定了坚实的基础。

第五章 网络流量分类系统设计及实现

本章主要介绍了基于前面章节的研究内容设计和实现网络流量分类系统*Network Traffic Classification System*(*NTCS*)。着重介绍了系统的框架，各个功能模块的设计与实现。

5.1 系统整体框架

基于上述研究内容，将基于机器学习方法的网络流量分类的分类过程用图5-1来表示。分类过程分为网络数据采集、网络流特征集生成、机器学习及分类器评测四个阶段。因此我们的网络流量分类系统大致可以分为数据采集模块、网络流量分析及特征生成模块和流量分类模块。各个模块组成见图5-2。

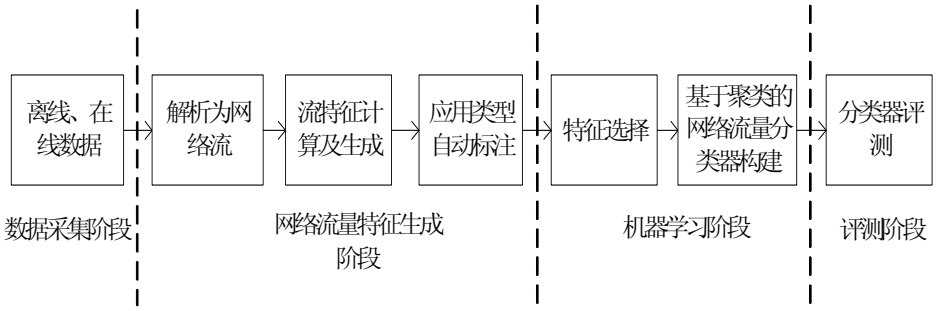

图 5-1 网络流量分类过程

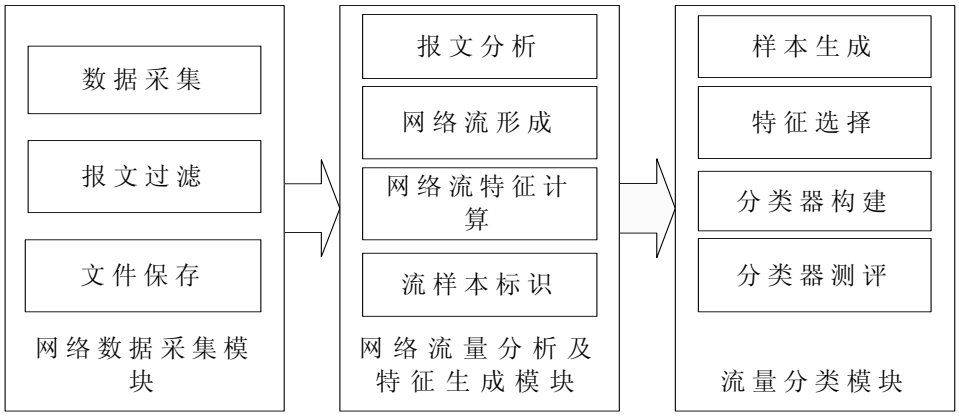

图 5-2 各模块主要功能及组成

5.2 数据采集模块

该模块主要实现数据采集、报文过滤、网络踪迹文件保存等功能。在本文中主要利用WinPcap的函数库进行的开发。WinPcap是一个在Windows操作系统下的免费的、公开的用于直接访问网络的系统。它包含了一套与libpcap兼容的捕获

数据包的函数库。WinPcap能够被各种不同的网络分析工具、发现并修理故障工具、安全工具和监听工具使用。WinPcap可以独立于主机的协议(如TCP-IP协议)进行接收和发送数据包。这意味着WinPcap不能阻塞、过滤或处理本机上其它程序产生的数据：它仅仅能嗅探在网线上传输的数据包。本文的捕获系统也是基于WinPcap的函数库进行的开发。在开发中首先要导入jpcap的开发包，利用该API可以获得有关网卡的信息。图5-3是数据采集的界面图。

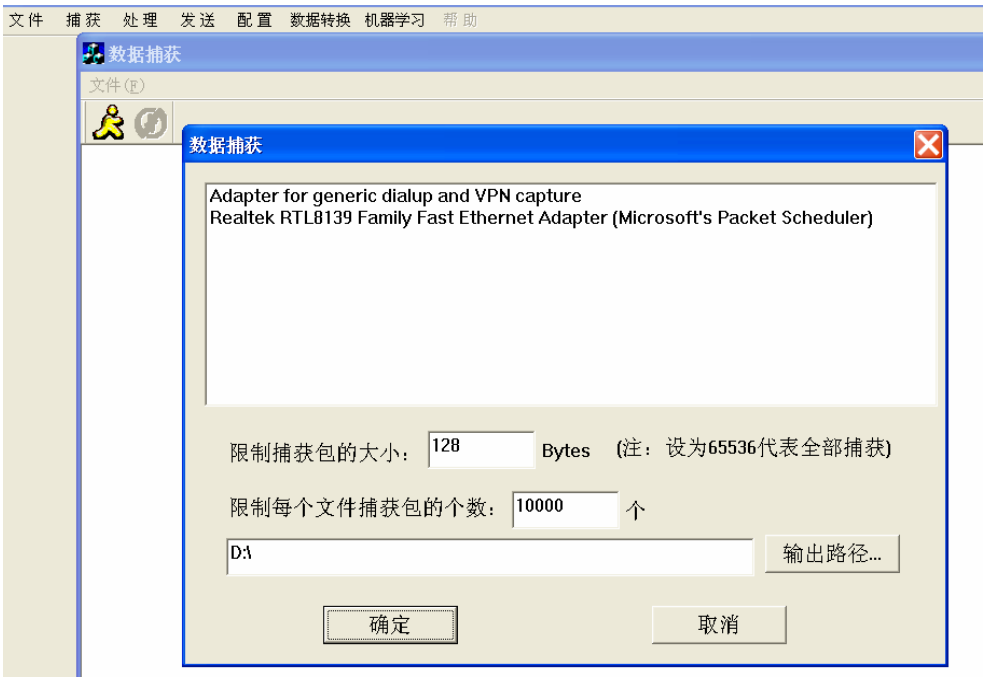

图 5-3 系统采集数据时的界面

数据采集首先要确定网卡信息，只有选对相应网卡才能采集到数据。所以系统应该显示网卡的相关信息，获取网卡信息代码如下：

```
public class GetNetworkDevice //获取网卡信息
{
    public String getDevice()
    {String DevicesStr= "";
        NetworkInterface[] devices = JpcapCaptor.getDeviceList();
        for (int i = 0; i < devices.length; i++)
        {
            DevicesStr = DevicesStr + devices[i].description + "@";
        }
        return DevicesStr;
    }
    for (int i = 0; i < devices.length; i++)//显示网卡列表
        {
            //显示网卡名称
            System.out.println(i+": "+devices[i].name + "(" + devices[i].description+")");
            //显示MAC地址
            System.out.print(" MAC address:");
            for (byte b : devices[i].mac_address)
                System.out.print(Integer.toHexString(b&0xff) + ":");
            System.out.println();
```

```
            //显示IP地址、网络掩码及广播地址
            for (NetworkInterfaceAddress a : devices[i].addresses)
            System.out.println(" address:"+a.address + " " + a.subnet + " "+ a.broadcast);
        }
}
```

Jpcap 提供的开发包不但可以捕获网络数据包还可以对其进行解析。在 JAVA 开发中我们在导入 Jpacp 相关开发包后，利用继承线程的办法实现数据报文的捕获、存储和解析，给我们提供了极大的方便。一旦网卡被打开，就可以使用 *CapturePacket* 类进行数据捕获，储存以及处理。图 5-4 给出了 *CapturePacket* 类的类图。

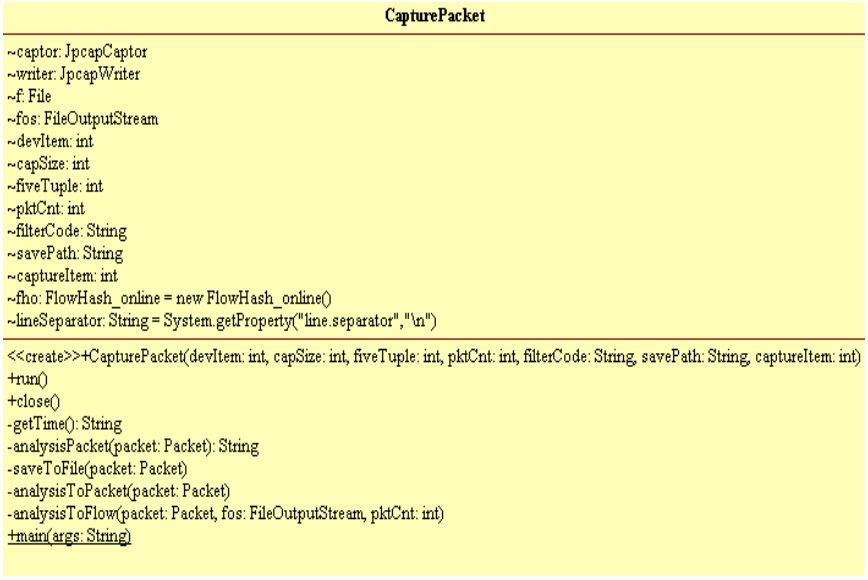

图 5-4 CapturePacket 类图

CapturePacket 类中函数说明：

Public void run (): 启动新的线程，捕获网络原始数据，并以时间为文件名保存网络数据。

*Public void close ():*结束数据捕获。

*Private String getTime ():*获取当前系统时间，返回结果传递给 *run()*函数，用于生成文件名。

*Private String analysisPacket (Packet pachet):*解析报文信息。

saveToFile (Packet pachet): 保存文件。

analysisToPacket(Packet packet): 解析报文具体信息。

analysisToFlow(Packet packet, FileOutputStream fos, int pktCnt): 将报文初步解析为流。

5.3 网络流量分析及特征生成模块

本部分主要介绍网络报文分析，生成流，网络流特征产生和流样本自动标注等几个功能模块的设计。捕获的网络报文数据经过这些处理后形成标准的arff格式文件。Arff格式文件的特点是各个记录相互独立、没有顺序要求，同时各个记录间不存在关系。每条arff文件记录就是一条流样本，流样本的属性用逗号隔开。

1. 网络报文分析模块

该模块主要涉及对以dmp格式的网络踪迹文件的分析处理，即对捕获的数据报文进行解析。解析的过程就是提取报文中的内容信息，包括IP报头所包含的新，TCP或UDP报头所包含的信息，以及标志位和提取包的有效载荷的内容。这里我们使用AnalysisDmpPacket类来处理dmp文件。图6-4就是AnalysisDmpPacket类图。

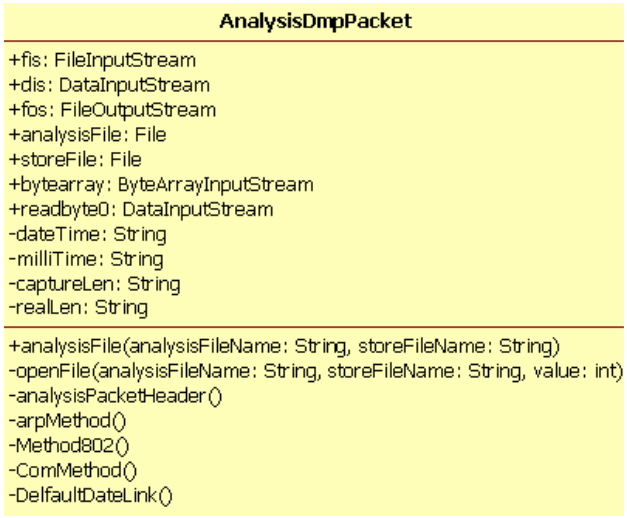

图 5-5 AnalysisDmpPacket 类图

其中*analysisFile*()函数用来解析包的信息，这些信息包括了TCP报头，IP报头，UDP报头及其他标志位信息。再利用*analysisPacketHeader*()函数就可以得到源、目的IP地址，端口号，报文数目，报文的时间特征及相关的标志位信息。*ArpMethod*(),*DefaultDateLink*()分别是处理arp和数据链路层函数。

2. 网络流的形成

网络流的形成主要由*FeatureStruct*，*AnalysisDmp*，*FlowHash*等类来实现。*FeatureStruct*类中定义了流的属性，*AnalysisDmp*类似*AnalysisDmpPacket*类也是用来解析报文信息，解析后的信息传递给*FlowHash*形成网络流。图5-6为*AnalysisDmp*和*FlowHash*两个类的类图。这里仅对*FlowHash*进行详细说明。

DealPacketInformation(String s):传进的变量为*AnalysisDmp*分析报文后的信息。通过前向五元组匹配和后向无原组匹配的方法，计算报文的前向、后向报文个数、大小、最大值、最小值、均方差及时间和标记位属性的值。

ListHashMap():输出流的特征。当处理完一个dmp文件后通过遍历*HashMap*

方法将各特征值写入文件，并以流的形式保存。*FeatureResultFile(String StoreFileName)*:用来暂时保存流特征。

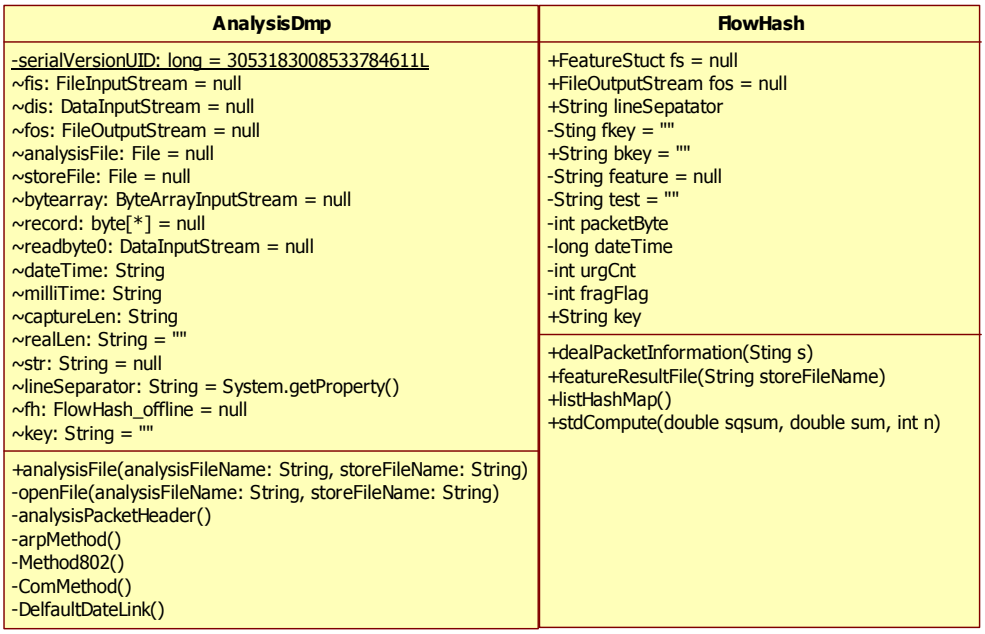

图 5-6 AnalysisDmp、 FlowHash 类图

图 5-7 是报文解析为流的流程图，图中 Pf 代表前向五元组 Pb 代表后向五元组。这是因为我们的研究对象是双向的 TCP 流，因此有前向和后向的分别。

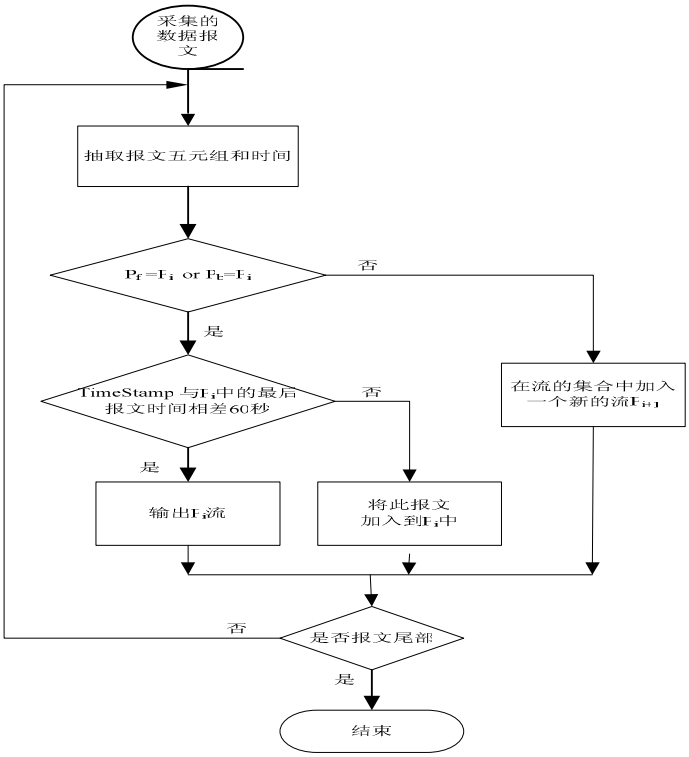

图 5-7 报文解析为流的流程图

在将报文解析为流过程中我们要进行流特征计算，从而产生流的候选特征集。流特征计算主要是通过提取报文各部分信息头进行分析，统计相应的信息。图 5-8 是流特征计算时的流程图。对于要处理的报文，首先采用网络流形成的方法找到匹配的流 F_i，接着判断报文的时间与 F_i 中的最后一个报文的时间差，如果大于预先设定的 IdleTimeThreshold，则计算与时间相关的一些特征；如果报文的协议是 TCP 协议，就计算有关标志位特征；最后还得根据报文是否属于 F_i 流中的前向报文来计算双向流的报文个数及大小特征，见表 2-2 中的特征。

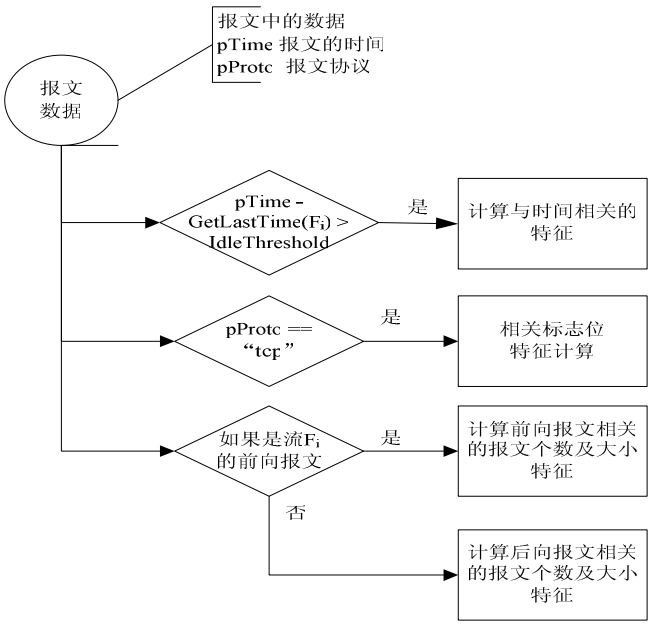

图 5-8 流量统计特征计算过程图

3. 样本自动标注

在2.3.2中我们详细介绍了流样本自动标注的方法和流程。这里我们通过*FlowHash*类中的*dealPacketInformation*()方法中的相关信息结合端口、协议和有效载荷等方法直接将流标注好，以文件的形式保存。图5-9就是流量分析后形成的流样本在文件中的表示形式。

```
2,120,1,60,60,60,60,0.0,60,60,60,0.0,1,1,0,0,0,0,0,0,0,3,0,0,0,0,2,1,0,0,0,0,0,0,WEB
3,186,0,0,62,62,62,0.0,0,0,0,0.0,2,0,0,0,0,0,0,0,9,0,0,0,0,0,0,0,0,0,0,3,0,WEB
1,60,1,60,60,60,60,0.0,60,60,60,0.0,0,1,0,0,0,0,0,0,0,2,0,0,0,0,1,1,0,0,0,0,0,0,WEB
1,66,0,0,66,66,66,0.0,0,0,0,0.0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,WEB
3,186,1,60,62,62,62,0.0,60,60,60,0.0,2,1,0,0,0,0,0,0,9,0,0,0,0,0,0,1,0,0,0,1,3,0,WEB
3,186,0,0,62,62,62,0.0,0,0,0,0.0,2,0,0,0,0,0,0,0,9,0,0,0,0,0,0,0,0,0,0,3,0,WEB
3,186,0,0,62,62,62,0.0,0,0,0,0.0,2,0,0,0,0,0,0,0,9,0,0,0,0,0,0,0,0,0,0,3,0,WEB
1,60,0,0,60,60,60,0.0,0,0,0,0.0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,WEB
1,60,0,0,60,60,60,0.0,0,0,0,0.0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,WEB
1,66,0,0,66,66,66,0.0,0,0,0,0.0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,WEB
1,60,1,60,60,60,60,0.0,60,60,60,0.0,0,1,0,0,0,0,0,0,0,2,0,0,0,0,1,1,0,0,0,0,0,0,SMTP
1,60,1,60,60,60,60,0.0,60,60,60,0.0,0,1,0,0,0,0,0,0,0,2,0,0,0,0,1,1,0,0,0,0,0,0,SMTP
1,60,1,60,60,60,60,0.0,60,60,60,0.0,0,1,0,0,0,0,0,0,0,2,0,0,0,0,1,1,0,0,0,0,0,0,SMTP
1,60,1,60,60,60,60,0.0,60,60,60,0.0,0,1,0,0,0,0,0,0,0,2,0,0,0,0,1,1,0,0,0,0,0,0,SMTP
1,60,1,60,60,60,60,0.0,60,60,60,0.0,0,1,0,0,0,0,0,0,0,2,0,0,0,0,1,1,0,0,0,0,0,0,SMTP
1,60,1,60,60,60,60,0.0,60,60,60,0.0,0,1,0,0,0,0,0,0,0,2,0,0,0,0,1,1,0,0,0,0,0,0,SMTP
```

图 5-9 流量分析后形成的样本流

5.4 流量分类模块

该模块的实现主要借助Waikato大学开发的著名的公开的数据挖掘工作平台Weka。该平台集合了大量能承担数据挖掘任务的机器学习算法，开发者则可使用Java语言，利用Weka[57]的架构上开发出更多的数据挖掘算法。在Weka中集成自己的算法甚至借鉴它的方法自己实现可视化工具并不是件很困难的事情。本文中特征选择功能模块和分类模块都借助了Weka中的Weka.attributeSelection和Weka.clusterers两个包。这里我们仅简要介绍与特征选择和聚类相关的几个类。

图 5-10 给出了本文中提到的两种特征选择算法的类图。我们前面已经对相关函数作了较详细的说明，这里不再赘述。同样，我们可以得到聚类模块的类图，这里不再列举。

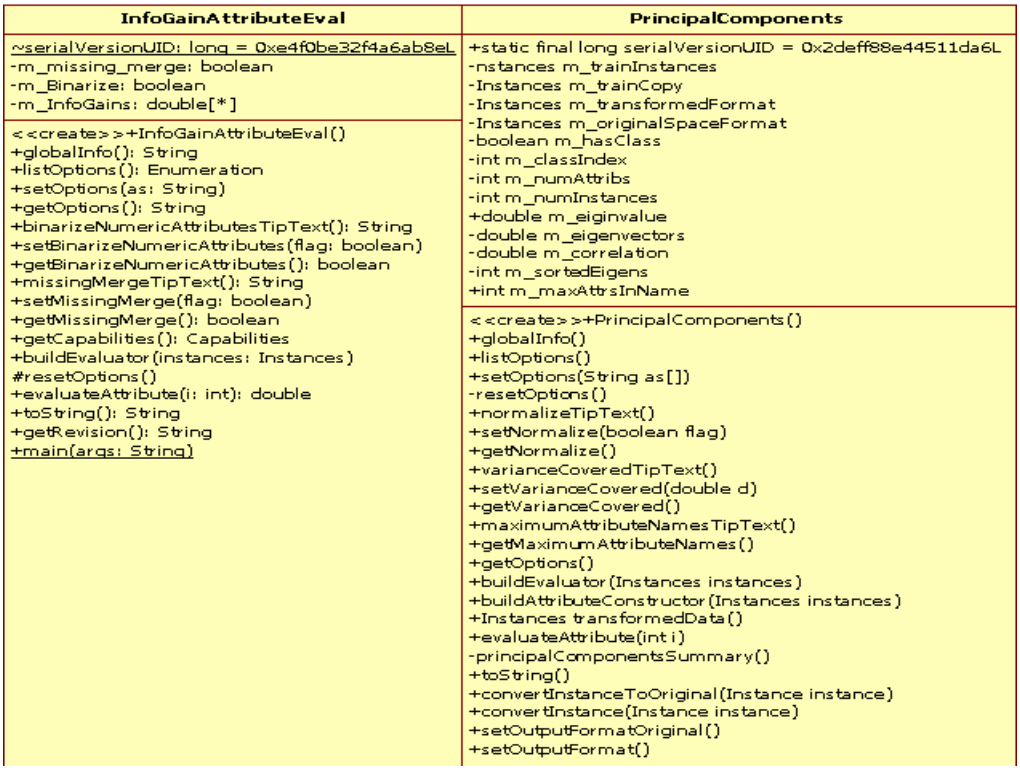

图 5-10 InfoGainAttributeEval 和 PrincipalComponents 的类图

从分析跟踪文件的结果文件中按照所分类别提取出将要用于分类的样本，形成 arff 格式的文件，形成这种格式文件的好处在于我们可以使用 weka 中的各种机器学习算法来对样本进行分类评测。流样本经过特征选择和聚类处理后，其处理结果仍然可以以标准的 Arff 格式保存。图 5-11 为经过 *Principal Components Analysis* 算法特征选择后的流记录，图 5-12 为使用 K-Means 聚类算法聚类后将流样本划分到对应的簇的情况。

在构建分类器模块中，为了构成一个可伸展性的平台，方便以后一些新的分类方法的加入，在实现中采用了接口技术。另外将 weka[]中的大部分算法导入到

了系统当中。

```
124,62,62,62,0,0,0,0,0,0,NNTP
124,62,62,62,0,0,0,0,0,0,NNTP
124,62,62,62,0,0,0,0,0,0,NNTP
124,62,62,62,0,0,0,0,0,0,NNTP
124,62,62,62,0,0,0,0,0,0,NNTP
124,62,62,62,0,0,0,0,0,0,NNTP
62,62,62,62,0,0,0,1,0,0,NNTP
186,62,62,62,0,0,0,0,0,0,NNTP
124,62,62,62,0,0,0,0,0,0,NNTP
186,62,62,62,0,0,0,0,0,0,NNTP
66,66,66,66,66,0,0,0,254,2.598076,WEB
60,60,60,60,0,0,1,1,0,0,WEB
1021,779,204,60,576,287.401044,4,8,698,242.774701,WEB
66,66,66,66,0,0,1,1,0,0,WEB
5128,1506,639,60,62,677.045789,7,0,362,0.745356,WEB
304,122,75,60,62,26.570661,3,0,122,1,WEB
120,60,60,60,60,0,2,3,60,0,WEB
```

图 5-11　特征选择后的流记录

```
0,1,60,0,0,60,60,60,0,0,0,FTP,cluster16
1,1,60,0,0,60,60,60,0,0,0,WWW,cluster3
2,1,60,0,0,60,60,60,0,0,0,FTP,cluster16
3,4,284,0,0,71,71,71,0,0,0,Bittorrent,cluster18
4,5,476,3,186,60,218,94,60,66,62,WWW,cluster8
5,1,60,0,0,60,60,60,0,0,0,FTP,cluster16
6,3,182,1,62,60,62,60,62,62,62,WWW,cluster8
7,4,308,4,246,60,122,76,60,66,61,WWW,cluster8
8,7,902,5,868,66,313,127,66,592,173,BtSprit,cluster7
9,3,553,8,1821,60,431,184,60,1401,227,WWW,cluster5
10,3,182,2,120,60,62,60,60,60,60,WWW,cluster8
11,3,234,0,0,78,78,78,0,0,0,Bittorrent,cluster18
12,2,286,0,0,143,143,143,0,0,0,BtSprit,cluster6
13,1,60,0,0,60,60,60,0,0,0,WWW,cluster3
14,5,406,2,120,60,164,80,60,60,60,SOCK,cluster8
15,1,60,0,0,60,60,60,0,0,0,FTP,cluster16
16,1,74,0,0,74,74,74,0,0,0,WWW,cluster3
17,4,304,4,242,60,122,75,60,62,60,WWW,cluster8
```

图 5-12 聚类后将流样本划分到对应的簇的情况

5.5 小结

本章主要对网络流量分类项目各个功能模块进行了详细的说明，并对其中比较重要的类作了详细的设计。系统实现主要借助了 jpcap 提供的 API 开发包和 weka 中相关算法的功能实现。

第六章 总结与展望

6.1 总结

本文针对基于端口和基于有效载荷的网络流量分类方法的缺点，提出了一种基于聚类分析的网络流量分类框架。该框架以网络流为研究对象，仅使用网络流在通信时产生的统计特征对网络流量按照应用类型进行分类。在研究过程中，实现了网络流特征的产生、选择及分类等重要步骤。经过我们对提出的网络流量分类框架的评估，认为这些研究有一定实用价值，并对其他分类问题有一定借鉴意义。本论文主要的工作及创新点如下：

1、提出基于聚类的网络流量分类方法，网络流经过聚类分析后，簇所属的类别采用简单多数投票的方式来确定；并给出基于聚类的分类器的分类规则。实验表明，该方法能够达到较高的查准率和总精确度。

2、提出的框架能够很好的适用多类分类问题，能够成功地将包含多种应用类型实验数据分类，具有较强的普遍适用性；

3、使用两种特征选择方法所选出的各自最优特征子集能够有效代表整个特征空间，提高了分类算法的效率；

4、提出采用独立于端口和协议的网络流量统计特征进行分类。对采用非固定端口以及加密数据等手段的服务也有很强的识别能力。基于这方面考虑，文章中产生了 34 个独立于端口和协议的候选特征。

6.2 进一步的研究工作

虽然在研究过程中，我们提出了一些新的观点和方法，取得了一定的研究成果。但在基于机器学习的网络流量分类研究中，还有很多方面值得进一步研究。具体表现在：

1、本文主要针对网络中 TCP 流的研究，但对 UDP 流的研究甚少。UDP 流量同样有其对应的流特征，同样可以采用机器学习方法来研究。这样有利于拓展我们的分类框架，实现基于 UDP 通信的流量分类。

2、探索选择新的聚类算法设计我们的分类器。本文中我们所使用的几种聚类算法都只是把流样本划分到固定的簇当中，而选用基于概率模型的聚类算法就有可能根据概率把流样本划分到不同的簇之中，这样可能对提升分类器的分类精度有所帮助。而且这些被称为软聚类的聚类算法在机器学习相关的文献中可查阅到。

3、设计通用的网络流量分类器。本文中我们仅对常见应用类型的流量进行了实验，而对小样本流量没有进行相关的实验；此外，我们也没有对网络上异常流量进行研究。因此有必要对此深入的研究，设计出通用的网络流量分类器，这种分类器既能有效分类网络流量，也能发现网络中的异常流量。

4、在线网络流量分类。本文中只是对网络中的数据进行捕获并形成踪迹文件后进行解析分类，没有实现实时的在线的网络数据分类。而高性能的实时在线网络分类能及时反映网络状态，可以让网络管理员及时地对网络流量进行控制，提高网络服务质量。因此在这一方面的研究很有实用和商业价值，值得进一步研究。

参考文献

[1] Cache Logic. Peer-to-Peer in 2005. http://www.cachelogic.com/home/pages/research/p2p2005.php, 2005.

[2] S. Sen and J. Wang. Analyzing Peer-to-Peer Traffic across Large Networks [C].IEEE/ACM Transactions on Networking, 2004, 12(2):219-232.

[3] T. Karagiannis, A. Broido, M. Faloutsos, and KC Claffy. Transport Layer Identification of P2P Traffic[C]. In IMC'04, Taormina, Italy, October 2004.

[4] S.Sen, O. Spatscheck, and D. Wang.Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures[C]. In WWW'04, New York, USA, May 2004.

[5] T. Choi, C. Kim, S. Yoon, J. Park, H. Kim, H. Chung, and T. Jesong.Content Aware Internet Application Traffic Measurement and Analysis[C]. In IEEE/IFIPNOMS'04, Seoul, Korea, April 2004.

[6] P. Haffiner, S. Sen, O. Spatscheck, and D. Wang. ACAS: Automated Construction of Application Signatures [C]. In SIGCOMM'05 MineNet Workshop, Philadelphia, USA, August 2005.

[7] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: Multilevel Traffic Classification in the Dark [C]. In SIGCOMM'05, Philadelphia, USA, August 2005.

[8] J. Ma, K. Levchenko, C. Krebich, S. Savage, and G. Voelker.Unexpected Means of Protocol Inference [J]. In IMC'06, Rio de Janeiro, Brasil, October 2006.

[9] A. Moore and K. Papagiannaki.Toward the Accurate Identification of Network Applications [J]. In PAM'05, Boston, USA, March 2005.

[10] A. McGregor, M. Hall, P. Lorier, and J. Brunskill.Flow Clustering Using Machine Learning Techniques [C]. In PAM'04, Antibes Juan-les-Pins, France, April 2004.

[11] A. Moore and D. Zuev.Internet Traffic Classification Using Bayesian AnalysisTechniques [C]. In SIGMETRIC'05, Banff, Canada, June 2005.

[12] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield. Class-of-Service Mapping for QoS: A Statistical Signature-based Approach to IP Traffic Classification [C]. In IMC'04, Taormina, Italy, October 2004.

[13] Y. Zhang and V. Paxson. Detecting Backdoors.In USENIX Security Symposium, Denver [J], USA, August 2000.

[14] T. Karagiannis, A. Broido, M. Faloutsos, and kc. claffy. Transport Layer Identification of P2P Traffic [C]. In IMC'04, Taormina, Italy, October 2004.

[15] A. McGregor, M. Hall, P. Lorier, and J. Brunskill. Flow Clustering Using Machine Learning Techniques [C].In PAM'04, Antibes Juan-les-Pins, France, April 2004.

[16] A. Moore and D. Zuev.Internet Traffic Classification Using Bayesian Analysis

Techniques [J]. In SIGMETRIC'05, Banff, Canada, June 2005.

[17] M. Roughan, S. Sen, O. Spatscheck, and N. Dueld. Class-of-Service Mapping for QoS: A Statistical Signature-based Approach to IP Traffic Classification [J]. In IMC'04, Taormina, Italy, October 2004.

[18] Tom M. Mitchell .Machine Learning. [M] McGraw-Hill Education (ISE Editions), December 1997.

[19] J. Frank.Machine Learning and Intrusion Detection: Current and Future Directions [C].Proceedings of the National 17th Computer Security Conference, 1994.

[20] S. Zander, T.Nguyen, G. Armitage.Self-learning IP Traffic Classification based on Statistical Flow Characteristics [C]. Passive & Active Measurement Workshop (PAM) 2005, Boston, USA, March/April 2005.

[21] Yingqiu Liu, Wei li, Yun-chun li. Network Traffic Classification Using K-Means Clustering [J].Proceedings of the Second International Multi-Symposiums on Computer and Computational Sciences, Pages 360-365, 2007.

[22] SHI ZHONG, TAGHI KHOSHGOFTAAR, and NAEEM SELIYA.CLUSTERING-BASED NETWORK INTRUSION DETECTION [J].Department of Computer Science and Engineering Florida Atlantic University, Boca Raton, FL 33431, USA Santa Clara, California, USA, April, 1999：9－12.

[23] Gerhard Munz, Sa Li, Georg Carle.Traffic Anomaly Detection Using K-Means Clustering [C].Computer Networks and Internet Wilhelm Schickard Institute for Computer Science University of Tuebingen, Germany.2007.

[24] Jeffrey Erman， Anirban Mahantie，Martin Arlitt Ira Coheny，Carey Williamson Semi-Supervised Network Traffic Classification. [C] SIGMETRICS'07, San Diego, California, USA. June, 2007:12－16.

[25] Jeffrey Erman. Offline/Realtime Network Traffic Classication Using Semi-Supervised Learning [C].CALGARY, ALBERTA April, 2007.

[26] Ning Li, Zilong Chen, and Gang Zhou.Network Traffic Classification Using Rough Set Theory and Genetic Algorithm [J].D.-S. Huang, K. Li, and G.W. Irwin (Eds.): ICIC 2006, LNCIS 344,2006: 945-950.

[27] 杨哲.基于 SOM 人工神经网络的网络流量聚类分析[J]计算机工程,2006,32(6): 103-105.

[28] 邓河，阳爱民，刘永定. 一种基于 SVM 的网络流量分类方法. [J]计算机工程与应用,2008, 44(14): 122-126.

[29] http://www.mirrorservice.org/sites/ftp.wiretapped.net/pub/security/packet-capture/winpcap/.

[30] Andrew Moore, Denis Zuev and Michael Crogan.Discriminators for use in flow-based classification [M]. ISSN 1470-5559 August 2005.

[31] H Liu, R Setiono. A Probabilistic Approach to Feature Selection: A filter Solution [J].Proc of Int'l Conf on Machine Learning[C].1996.319-327.

[32] Sanmay Das.Filters Wrappers and a Boosting Based Hybrid for Feat ure Selection. [C] l Proc of t he 8t h Int' l Conf on Machine Learning.2001:74-81.

[33] Huang Yuan , Shian-Shyong Tseng , Wu Gangshan , et al.A Two-Phase Feature Selection Met hod Using Both Filter and Wrapper [C].Proc of 1999 IEEE Inter' l Conf on Systems ,Man , and Cybernetics,Vol 2].1999,132 -136.

[34] R Kohavi, G H John.Wrappers for Feature Subset Selection [J].Artificial Intelligence Journal, 1997 (1-2): 273-324.

[35] I. T. Jolliffe.Principal Component Analysis [M].New York, Springer-Verlag, chs. 2,3.

[36] YU Lei, Liu Huan. Efficient Feature Selection via Analysis of Relevance and Redundancy. [J] Journal of Machine Learning Re2 search , 2004 (5) :1205～1224.

[37] Mitra P, Murthy C A. Pal S K. Unsupervised Feature Selection Using Feature Similarity. [J] IEEE Transactions on Pat tern Analysis and Machine Intelligence , 2002 ,24 (3) :301～312.

[38] Wang B, Pan H D, Li J H. A Secure (T, N) Threshold Signature Scheme [J]. Journal of Shanghai Jiaotong University, 2002, 36 (9):133321336.

[39] Jain AK, Murty MN, Flynn PJ. Data clustering: A review [C]. ACM Computing Surveys, 1999, 31(3):264-323.

[40] Jain AK, Dubes RC.Algorithms for Clustering Data [J].Prentice-Hall Advanced Reference Series, 1988. 1-334.

[41] Jain AK, Duin RPW, Mao JC. Statistical pattern recognition: A review [J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2000,22(1):4-37.

[42] Sambasivam S, T documents. Issues in Informing Science and Information Technology [J], 2006,(3):563-579.

[43] Marques JP, Written; Wu YF, Trans. Pattern Recognition Concepts, Methods and Applications [M]. 2nd ed., Beijing: Tsinghua University Press, 2002. 51-74 (in Chinese).

[44] Fred ALN, Leitão JMN.Partitional vs hierarchical clustering using a minimum grammar complexity approach [C].In: Proc. of the SSPR&SPR 2000.LNCS 1876, 2000.193-202. http://www.sigmod.org/dblp/db/conf/sspr/sspr2000.html.

[45] Marques JP, Written; Wu YF, Trans. Pattern Recognition Concepts, Methods and Applications [M]. 2nd ed., Beijing: Tsinghua University Press, 2002.51-74 (in Chinese).

[46] Guha S., Rastogi R. and Shim K., "CURE: An efficient clustering algorithm for large database, "[C] In Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data(SIGMOD' 98), Seattle, WA, June 1998:73-84.

[47] Guha S., Rastogi R. and Shim K., “ROCK: A Robust Clustering Algorithm For Categorical Attribute,”[C]. In Proc. 1999 Int. Conf. Data Engineering (ICDE’99), Sydney, Australia, Mar. 1999: 512-521.

[48] Zhang T., Ramakrishnan R. and Livny M., “BIRCH: An efficient clustering Method for large database，” [C] In Proc. 1996 ACM-SIGMOD Int. Conf.Management of Data (SIGMOD’96) 1996:103-114.

[49] Kaufman L. and Rousseeuw PJ, “Finding Groups in Data: an Introduction to cluster Analysis，”[C] John Wiley & Sonw, 1990.

[50] Ng R. and Han J., Efficient and Effective Clustering Method for Spatial Data Mining,” [C] In Proc. 1994 Int. Conf. Very Large Database (VLDB’94), Santiago, Chile, Sept, 1994:144-155.

[51] MacQueen J., “Some Methods for Classification and Analysis of Multivariate Observations,” [C] In Proc. 5th Berkeley Symp. Math. Stat. and Prob., Vol .1, pp. 181-297, 1967.

[52] Martin. Ester, H Peter Kriege, J Sander, et al. A density-based algorithm for discovering clusters in large spatial databases with noise [C].The 2nd International Confabulation Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA, 1996.

[53] Ankerst M., Breunig M., Kriegel H.P. and Sander J., “OPTICS: Ordering points to identify the Clustering strcture,”[C] In Proc. 1999 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD’99), pp. 49-60, Philadelphia,PA, Jume 1999.

[54] Wang W., Yang and Muntz R., “STING: A Statistical Information grid approach to Spatial Data Mining,”[C] In Proc. 1997 Int. Conf. Very Large Database (VLDB’97), pp. 185-195, Athens, Greece, Aug, 1997.

[55] Sheikholeslami G., Chatterjee s., and Zhang A., “WaverCluster: A multi-resolution clustering approach for very large spatial database,”[C] In Proc. 1998 Int. Conf. Very Large Database (VLDB’98), pp. 428-439, New York, Aug, 1998.

[56] Ding C, He X. K-Nearest-Neighbor in data clustering: Incorporating local information into global optimization [J]. In: Proc. of the ACM Symp.On Applied Computing.Nicosia: ACM Press, 2004. 584-589. http://www.acm.org/ conferences/ sac/sac2004/.

[57] http://www.cs.waikato.ac.nz/ml/weka/.

附 录

参与的科研项目：

[1]中国博士后科学基金， 题目:互联网上基于应用类型的网络流量分类研究。编号：No.20070410299

[2]广东省自然科学基金博士科研启动基金，题目：基于机器学习的网络流量分类研究。编号:No.7300450

论文发表情况：

[1] 一种使用DBSCAN聚类的网络流量分类方法.计算机应用研究2009第8-9期。

致　谢

时光如水，岁月如梭，三年的研究生时光很快就要过去，在这篇论文完成之际，忽然意识到两年半研究生生涯将就此结束，心中难免不舍。回顾这一程求学路，记忆里满是老师的悉心指导和同学的快乐相伴，他们让我的生活充实而富有活力，让我在生命的又一里程碑上刻下了重要的篇章，在此我要向他们表达最诚挚的感谢。

首先，衷心感谢我的导师阳爱民教授！感谢他教导了我学术研究的方法，也为我指引了研究的领域和方向。在本论文的撰写过程中，阳老师从选题直至成稿一直给予我重要的指导和帮助，为我解开了无数的困惑，提供了很多关键性的建议。他实事求是的科学态度，严谨的治学精神，渊博的学识，诚恳待人的品格，深刻地影响了我，让我在专业知识和科研水平上进步的同时，也领悟很多为人处世的道理。师恩永生难忘！

感谢李长云教授、金可音教授、朱艳辉教授，他们渊博的知识、敏捷的思维和废寝忘食的敬业精神给我留下了深刻的印象，并始终是我学习的榜样；满君丰老师、王志兵老师以及所有的任课老师，谢谢他们在三年中传授我新的知识以及给予的帮助。

感谢同实验室的同学刘永定、邱密、汪彦及智能信息研究所的全体同学，在学习生活当中他们给了我热情的帮助和真诚建议。和他们一起对课题的讨论使我开阔了眼界，扩宽了思路！

感谢06级计算机研究生同学，一起渡过了这三年美好的时光！

最后，深深感谢我的父母，感谢父母对我的养育之恩，我的每一个成长脚步都倾注着他们的心血，他们为我付出的一切我终身难以报答!感谢我的爱人，谢谢她在这三年来对我学业上的支持以及生活上的关爱。深深感谢我最亲爱的家人们，他们永远是我最坚强的后盾！

再次衷心感谢所有关心、爱护、支持和帮助过我的师长、亲人和朋友们。