**网络流量分类识别**

**数据挖掘课程报告**

**15S103163 宋博宇 网络安全实验室**

# 课题内容概述

网络安全实验室的课题中包含对网络流量进行分类识别的任务。

对网络流量按照应用类型准确地识别和分类是许多网络管理任务的重要组成部分，如流量优先级控制，流量定形、监管、诊断监视等。比如说，网络管理员可能需要识别并节流来自P2P协议的文件共享流量来管理自己的带宽预算，确保其他应用的网络性能。与网络管理任务类似，许多网络工程问题，如负载特征提取和建模，容量规划，路由配置也得益于准确地识别网络流量。

实时的流量统计有能力帮助网络服务提供商和他们的设备供应商解决困难的网络管理问题。网络管理员需要随时知道什么流量穿过了他们的网络，才能迅速采取应对措施来保障多样的商业服务目标。流量分类可能是自动入侵检测系统的核心组成部分，用来检测拒绝服务攻击，可以触发针对优先客户的自动网络资源重分配，或者识别哪些违背了服务条款的网络资源使用。

如今各种不同的网络应用层出不穷，网络流量的复杂性和多样性给流量分类问题带来了巨大的挑战。很多研究人员开始寻找接近于数据挖掘的技术来解决流量分类问题。

# 流量识别任务中数据挖掘技术的应用

## 流量识别任务流程

如图2-1所示，基于机器学习的流量分类主要分为三个阶段，预处理阶段，学习阶段和预测阶段。预处理阶段包括对原始网络数据的整流，特征值计算以及特征值约简，学习阶段是训练模型学习规则的过程，预测阶段是对实际流量进行分类的过程。机器学习方法重点研究通过特征选择和训练进行分类模型的构造，即分类器的学习阶段。

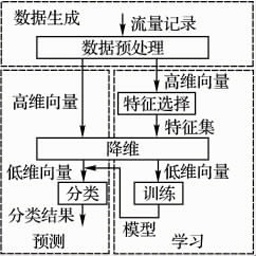


图 2-1 机器学习的流量分类

1. **数据预处理**

原始的网络数据集记录了每个数据包的到达时间和数据包内容，在预处理阶段首先要根据五元组进行整流，在每个TCP或UDP流上区分流量方向，然后在每个流上计算感兴趣的流量特征，如数据包大小的分布，数据包间隔时间，连接持续时间等。

1. **降维**

经过数据预处理后的网络流是一个有各项特征值的向量，可以作为机器学习算法的输入，但网络流特征冗余会影响分类结果的准确性，也会增加训练的计算开销，可以将高维向量投影到低维空间中，再用以训练。

1. **特征约简**

将可获得的特征都用来训练分类器并不一定是最好的选择，因为不相关的特征和冗余的特征会对算法的性能产生负作用。可以通过一些算法进行评估，选择具有很强代表性的特征子集，来训练模型。

1. **训练**

从训练数据集中构建分类模型的过程，主要任务是建立一个从网络流特征到应用类别的映射，有不同的分类模型可以选择。

1. **测试**

依据训练的分类模型，对未知的网络流进行预测，得出网络流所属的应用类别。该阶段涉及到对分类模型的评估，有很多流量分类度量指标可以选择。评估还可以分为以流计算和以字节计算两个方向，前者侧重于对流识别能力的评估，后者侧重于识别那些占据主要通信量的大流。

## 2.2特征值归约方法

采用**信息增益率评估，**特征集合为S，假设根据特征A划分训练集，划分前后信息量差值就是信息增益，见公式2-3：

 （2-3）

内在信息（Intrinsic Information）定义为公式2-4：

 （2-4）

信息增益率定义为公式2-5：

 （2-5）

对于特征集合S中的每一项特征，分别计算其信息增益率，从高到低进行排序，就是信息增益率评估的主要过程。

根据信息增益率来评估特征集合中的不同特征，可以知道哪些特征最有区分度，对训练分类器最有贡献。

## 2.3流量分类模型度量指标

评价一个分类器有多准确，一般是通过假正（False Positives），假负（False Negatives），真正（True Positives）和真负（True Negatives）这些度量指标来衡量的。

流量分类任务中使用准确率（Accuracy）作为度量指标。准确率通常定义为正确分类的样本数量在所有样本中的百分比。

## 2.4 K-Means聚类

在多个簇基分类器集成的系统中，训练阶段，网络流向量首先要经过一个聚类处理过程，我们选择的聚类算法是K-Means。

机器学习和数据挖掘领域中有许多基于划分的聚类算法，如K-Means算法、K-Medoids算法、CLARANS算法等。选择K-Means算法是因为它是这些算法中最快和最简单的。K-Means算法将数据集中的对象划分到事先指定的K个互斥的子集中，这些子集称为簇。在每一个簇中，划分算法都通过最小化簇内部的平方误差来最大化簇的同一性。平方误差见公式3-1：

 （3-1）

在每一个簇内部，都计算所有对象和簇中心点的距离的平方，然后累加这些平方距离。就代表了第个簇的中心。这里的距离度量是欧几里得距离，即公式3-2：

 （3-2）

两个n维向量a与b的欧几里得距离就是它们对应项与平方差的总和。

初始状态，K个簇的中心是在子集空间中随机选取的。数据集中的对象随后被划分到距离最近的簇中，K-Means迭代地计算每个簇新的中心点，然后根据这些中心点再次划分所有的对象。K-Means算法重复这个过程直到所有簇中的对象都稳定不再变化，这样就产生了最终的一个划分。

K-Means算法的一个关键问题是如何确定聚类个数K。在集成分类系统中，聚类个数K也是一个重要的参数。

## 2.5集成的分类算法

在每个簇上单独训练的簇基分类器，可以选择各种有监督的分类算法。

**（1）支持向量机**

SVM将低维的输入向量投影到高维的向量空间中，通过升维的方式将非线性可分问题转换为线性可分问题。

本文实验中选择了顺序最小优化（SMO）的SVM实现方案，这种高效的实现使用成对分类方法，将多类别分类问题分解成一系列的二分类子问题，从而消除了数值最优化的需求。

**（2）朴素贝叶斯**

朴素贝叶斯分类器建立在贝叶斯理论之上。这种分类技术解析每个属性和类别的关系，得到类别在该属性下的条件概率。

一个实例x属于类别c的概率可以根据公式3-1计算：

 （3-1）

在测试阶段，一个未知的实例属于哪种类别的概率最大，就会被认为属于哪种类别。

**（3）贝叶斯网络**

贝叶斯网络是有向无环图和一个条件概率表的结合。有向无环图中的结点表示特征或类别，而连接两个结点的边表示二者之间的关系。

条件概率表决定了这些边的连接强度。对每一个结点，条件概率表定义了给定父结点时的概率分布。如果一个结点没有父结点，那么概率分布就是无条件的。如果一个结点有多个父结点，那么概率分布就是给定父结点时的条件概率。

贝叶斯网络的学习分为两个步骤，首先网络结构形成（结构化学习），然后概率表被估算出来（概率分布计算）。

# 流量分类识别实验

## 实验环境

集成分类器的流量识别系统是在Java中使用Weka库编写的，Weka限制在GNU通用公共证书的条件下发布，可以运行在大多数操作系统平台上。Weka集成了许多用于数据挖掘任务的机器学习算法。这些算法既可以通过GUI直接用于数据集，用可以在java代码中调用提供给用户的API使用。Weka提供的工具包括数据的预处理，分类，回归，聚类，关联规则和可视化。同时，Weka也很适合用于开发新的机器学习方法。

## 实验数据

实验数据集为NIMS（Network Information Management and Security Group）捕捉。

表3-1 应用构成

|  |  |  |
| --- | --- | --- |
| **应用** | **流数量** | **百分比(%)** |
| TELNET | 1251 | 0.17 |
| FTP | 1728 | 0.24 |
| HTTP | 11904 | 1.66 |
| DNS | 38016 | 5.32 |
| lime | 646271 | 90.53 |
| local | 2557 | 0.35 |
| remote | 2422 | 0.33 |
| scp | 2444 | 0.34 |
| sftp | 2412 | 0.33 |
| x11 | 2355 | 0.32 |
| shell | 2491 | 0.34 |
| TOTAL | 713851 | 100 |

他们采集实验室内部的客户机通过SSH连接外部的四台SSH服务器产生的流量。通过SSH运行了六种不同的服务：Shell login，X11，Local tunneling，Remote tunneling，Scp和Sftp，也捕捉了几种主要的背景流量，如DNS，HTTP，FTP，P2P（limewire）和TELNET，它们的构成如表3-1所示。

## 特征选择

数据集中选取的网络流特征如表3-2所示。

为了得到训练集和测试集，首先将NIMS数据集分为两个数据子集，分别为NIMS1和NIMS2，在这两个数据子集中每类样本的比例与NIMS基本保持一致。本文中不同算法的训练样本都取自NIMS1，而NIMS2作为测试数据集用以评估不同算法的效果。

表3-2 网络流特征值

|  |  |  |
| --- | --- | --- |
| **编号** | **特征** | **缩写** |
| 1 | Minimum forward packet length | minfpktl |
| 2 | Mean forward packet length | meanfpktl |
| 3 | Maximum forward packet length | maxfpktl |
| 4 | Standard deviation of forward packet length | stdfpktl |
| 5 | Minimum backward packet length | minbpktl |
| 6 | Mean backward packet length | meanbpktl |
| 7 | Maximum backward packet length | maxbpktl |
| 8 | Standard deviation of backward packet length | stdbpktl |
| 9 | Minimum forward inter-arrival time | minfiat |
| 10 | Mean forward inter-arrival time | meanfiat |
| 11 | Maximum forward inter-arrival time | maxfiat |
| 12 | Standard deviation of forward inter-arrival times | stdfiat |
| 13 | Minimum backward inter-arrival time | minbiat |
| 14 | Mean backward inter-arrival time | meanbiat |
| 15 | Maximum backward inter-arrival time | maxbiat |
| 16 | Standard deviation of backward inter-arrival times | stdbiat |
| 17 | Protocol | protocol |
| 18 | Duration of the flow | duration |
| 19 | Number of packets in forward direction | fpackets |
| 20 | Number of bytes in forward direction | fbytes |
| 21 | Number of packets in backward direction | bpackets |
| 22 | Number of bytes in backward direction | bbytes |

分别使用基于相关性（CFS）和信息增益率（Gain Ratio）的方法处理NIMS1数据集，结果如表3-3所示。

表3-3 特征值约简

|  |  |
| --- | --- |
| **处理方法** | **特征子集** |
| CFS Greedy Forward | 1,2,3,4,6,7,8,9 |
| CFS Greedy Backward | 1,2,3,4,6,7,8,9 |
| Gain Ratio | 18,3,1,5,7,4,9,2 |

其中CFS方法贪心向前搜索和贪心向后搜索都得出同样的特征子集，Gain Ratio得到的信息增益率最高的前八项为18，3，1，5，7，4，9，2。综合上述结果，决定选取1,2,3,4,5,6,7,8,9,18为特征子集，如表4-4所示。这10项特征中主要包括双向的数据包大小分布数据和协议。

表4-4 约简后的网络流特征

|  |  |  |
| --- | --- | --- |
| **编号** | **特征** | **缩写** |
| 1 | Minimum forward packet length | minfpktl |
| 2 | Mean forward packet length | meanfpktl |
| 3 | Maximum forward packet length | maxfpktl |
| 4 | Standard deviation of forward packet length | stdfpktl |
| 5 | Minimum backward packet length | minbpktl |
| 6 | Mean backward packet length | meanbpktl |
| 7 | Maximum backward packet length | maxbpktl |
| 8 | Standard deviation of backward packet length | stdbpktl |
| 9 | Minimum forward inter-arrival time | minfiat |
| 10 | Protocol | protocol |

在聚类个数K=10，分类算法为支持向量机，决策算法为簇纯净度时，使用全部22项特征训练集成分类系统，其总体准确率为94.1%，使用约简后的10项特征训练，总体准确率为91%。在约简了一半特征后，总体准确率只下降了3%，表明这10项特征具有很好的代表性，以下的实验都将选取这10项特征。

# 初步研究结果

## 4.1聚类个数K分析

图4-1 总体准确率与聚类个数K的关系

使用聚类（如K-Means）处理流量分类问题时，簇与应用并非是1:1的映射。理想情况是形成的簇的数量与待识别的应用的数量相等，并且每一个应用都在一个簇中占绝大多数。但实际上，簇的数量都大于应用的数量。

分类算法为支持向量机，决策算法为簇纯净度时，如图4-1所示，集成分类系统的总体准确率随着聚类个数K的增加而增加，但准确率提升越来越慢。在聚类个数达到80时，总体准确率达到98%，基本收敛。

集成分类系统的总体准确率可以通过增加聚类个数K来提升，在以下的实验中，我们选取聚类个数K=50。

## 4.2决策算法比较

在聚类个数K=50，分类算法为支持向量机时，分别比较簇纯净度决策算法和基分类器精度决策算法，实验结果如表4-1所示。

从表4-1中可以看出，依据簇纯净度和基分类器精度的决策算法，在大多数应用上都有相同的准确率表现（只有在sftp上簇纯净度略优，在x11上基分类器精度略优），这说明两者在大多数情况下做出了一致的决策。

表4-1 决策算法比较

|  |  |  |
| --- | --- | --- |
| **应用** | **簇纯净度**  **准确率%** | **基分类器精度**  **准确率%** |
| TELNET | 99.8 | 99.8 |
| FTP | 100 | 100 |
| HTTP | 97.6 | 97.6 |
| DNS | 89.6 | 89.6 |
| Lime | 91 | 91 |
| Local | 99.8 | 99.8 |
| Remote | 98.6 | 98.6 |
| Scp | 93.7 | 93.7 |
| Sftp | **99.2** | 98.8 |
| X11 | 96.9 | **97.2** |
| Shell | 100 | 100 |
| Total | 96.93 | 96.92 |

考虑到聚类个数K可能对结果产生不同的影响，在分类算法为SVM时，随着聚类个数K的增加，分别测试不同决策算法的效果，对比结果如图4-2 所示。可以看出，只有在K<10时，两种决策算法的准确率略有出入，随着K的增加，依据簇纯净度和基分类器精度的决策算法效果没有差别。

图4-2 决策算法与聚类个数K的关系

基分类器精度的决策算法需要额外评估训练集上每个基分类器的精度，训练模型开销较大，所以在以下的实验中，我们选取依据簇纯净度的决策算法。

## 4.3集成分类系统与其它分类算法比较

### 4.3.1与支持向量机比较

在聚类个数K=50，分类算法为支持向量机，决策算法为簇纯净度时，集成分类系统与支持向量机的准确率比较如表4-2所示。集成分类系统的各应用识别准确率都高于支持向量机方法（除了DNS），尤其是lime（准确率从56.6%提升到91%）和scp（准确率从0%提升到93.7%）。总体准确率也由83.62%提升到96.93%，效果显著。

表4-2 与SVM比较

|  |  |  |
| --- | --- | --- |
| **应用** | **集成分类系统**  **准确率%** | **SVM**  **准确率%** |
| TELNET | **99.8** | 92.8 |
| FTP | 100 | 100 |
| HTTP | **97.6** | 83.1 |
| DNS | 89.6 | **100** |
| Lime | **91** | 56.6 |
| Local | **99.8** | 96.5 |
| Remote | **98.6** | 97.9 |
| Scp | **93.7** | 0 |
| Sftp | **99.2** | 99.3 |
| X11 | **96.9** | 93.6 |
| Shell | 100 | 100 |
| Total | **96.93** | 83.62 |

在SSH通道上传输的六种协议，Shell login，X11，Local tunneling，Remote tunneling，Scp和Sftp的识别准确率都得到了提高。

### 4.3.2与朴素贝叶斯比较

在聚类个数K=50，分类算法为朴素贝叶斯，决策算法为簇纯净度时，训练并评估集成分类系统，然后在同样的训练集上训练朴素贝叶斯分类器，二者分类准确率比较如表4-3所示。

朴素贝叶斯分类器在scp的识别上完全失败，而集成分类系统能够达到95.1%的识别准确率。集成分类系统的总体准确率也由85.8%提升到95.8%，效果显著。

表4-3 与NaiveBayes比较

|  |  |  |
| --- | --- | --- |
| **应用** | **集成分类系统**  **准确率%** | **NaiveBayes**  **准确率%** |
| TELNET | 95.5 | **97** |
| FTP | 100 | 100 |
| HTTP | **99.4** | 95.4 |
| DNS | 91.1 | **98.5** |
| Lime | **77.9** | 61.1 |
| Local | **99.9** | 96.5 |
| Remote | **98.7** | 97.9 |
| Scp | **95.1** | 0.3 |
| Sftp | 98.7 | **99.1** |
| X11 | 97.6 | **98.2** |
| Shell | 99.6 | **99.9** |
| Total | **95.8** | 85.8 |

### 4.3.3与贝叶斯网络比较

在聚类个数K=50，分类算法为贝叶斯网络，决策算法为簇纯净度时，训练并评估集成分类系统，然后在同样的训练集上训练贝叶斯网络分类器，二者分类准确率比较如表4-4所示。

可以看出，集成分类系统在集成贝叶斯网络后，准确率并没有多少提高，表明集成分类系统并不适合集成贝叶斯网络。

表4-4 与BayesNet比较

|  |  |  |
| --- | --- | --- |
| **应用** | **集成分类系统**  **准确率%** | **BayesNet**  **准确率%** |
| TELNET | 100 | 100 |
| FTP | 100 | 100 |
| HTTP | 99.4 | **99.5** |
| DNS | **98.5** | 94.3 |
| Lime | **95.7** | 94.5 |
| Local | 99.7 | **99.9** |
| Remote | **99.1** | 98.1 |
| Scp | **99.1** | 96.4 |
| Sftp | 98.8 | **99.1** |
| X11 | 98 | **98.7** |
| Shell | 99.6 | **100** |
| Total | **98.9** | 98.2 |