基于子空间聚类算法的流量分类方法研究

许学研[1] 王苏南[1,2] 吴春明[1]
(浙江大学计算机科学与技术学院 杭州 310027)[1]
(深圳职业技术学院电子与通信工程学院 深圳 518005)[2]

摘 要 目前网络流量业务类型具有不断变化和业务特征不断更新两大特点,但是,现有的流量分类器由于存在业务特征库更新代价大、误判率高等缺点,而无法满足正常的业务分类需求。因此需要设计一种子空间聚类算法来实现业务分类精细化,保障分类精确率、召回率以及效率等特性。实验验证表明,子空间聚类算法的业务分类精细化程度高,分类精确率平均超过95%,训练数据需求量低,并且这类方法对于改进DPI分类器对网络环境的适应能力有重大意义。

关键词 深度包检测,机器学习,流量分类,子空间聚类
中图法分类号 TP393 **文献标识码** A

Network Traffic Classification Method Research Based on Subspace Clustering Algorithm

XU Xue-yan[1] WANG Su-nan[1,2] WU Chun-ming[1]
(Computer Science College, Zhejiang University, Hangzhou 310027, China)[1]
(School of electronic and Communication Engineering, Shenzhen Polytechnic, Shenzhen 518005, China)[2]

Abstract Currently, service types, features of network traffic are changing constantly, but existing classification methods aren't able to satisfy such network traffic environment, because they lack capability to update features library efficiently, and have high misjudgement rate. So a subspace clustering algorithm was designed to test classification properties. Experemnts show that it can classify lots of business types, its classification precision rate exceeds 95%, and quantity demand of training samples is low. It is recommended to help DPI classifier adapt to changing network environment.

Keywords Deep packet inspection, Machine learing, Traffic classification, Subspace clustering

1 引言

互联网(Internet)的设计初衷是实现主机之间的互联互通,而互联网自身的发展现状却已经远远超出了其最初的设计理念。随着互联网的各类业务的种类和流量的快速增长,传统的尽力而为的传输网络已经不能满足互联网流量可管理、可控制、可扩展的需求。为此人们提出了不少创新型的网络体系结构[1-3],而在对网络资源进行有效管理和分配之前,都必须对网络流量的业务类型和特征进行分析,因此许多关于网络流量业务的分类方法应运而生[4]。

流量分类方法依据其分析的数据粒度的不同可以划分为两类,一类是基于网络包的分析方法(Packet-Based Methods),典型方法是DPI;另一类是基于网络流的分析方法(Flow-Based Methods),典型方法是机器学习(Machine Learing)。它们的基本流程如图1所示,分为离线特征提取与在线特征匹配两部分。离线特征提取的目的是保障在线分类特征匹配的准确有效,离线特征提取工作主要通过人工方式给样本流量打上业务标签,然后进行业务特征计算和学习。已有的流量分类器普遍都缺乏对离线特征提取效率的关注,导致了分类器无法快速适应互联网发生的业务类型变化[6]。提高离线特征提取的关键工作在于提升流量的离线分类能力,包括提高离线分类的业务类别识别能力、分类效率、精确率等特性,这项工作是提取流量业务特征的基础工作,也是整个流量分类器实现适应流量特征不断变化的网络环境的关键环节。

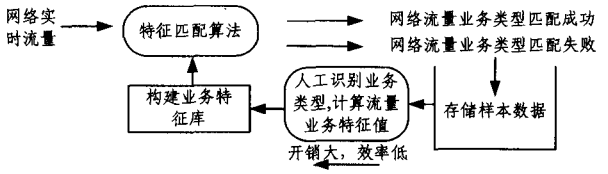

图1 流量分类器分类流程

本文第2节介绍离线分类模型的构建工作;第3节给出基于子空间聚类方法所构造的离线分类器;第4节是仿真实验及性能分析;最后是结束语。

本文受国家重点基础研究发展计划(973计划)资金项目(2012CB315903),国家自然科学基金项目(61103200,61379118),浙江省重点科技创新团队(2011R50010)资助。

许学研(1985—),男,硕士生,主要研究方向为网络流量测量、网络流量业务分类;**王苏南**(1983—),男,博士,主要研究方向为计算机网络、高性能路由、流量测量;**吴春明**(1967—),男,教授,博士生导师,主要研究方向为网络服务质量、可重构网络、网络虚拟化。

2 构造离线分类模型

通过图 1 可知，在线特征匹配的准确性以及对网络环境中流量业务类型更新变化的适应能力，是与实现特征库准确、高效的更新工作密切相关的。而这项工作又取决于流量离线分类工作的效率和准确程度。

对于离线分类工作而言，人工参与又是一个不可忽视的环节，因为对于未知类型的流量数据，只有通过人工方式确定样本数据的业务类型之后，才能够进行业务特征的计算、学习等工作。文献[9,14-16]已经对如何自动获取业务特征进行了介绍，其中包括了生物信息学中的 DNA 比对算法等。但是这些方法局限于提升获取业务特征的准确率，并没有解决降低人工参与程度的问题。本文经过对现有技术的深入分析，决定使用子空间聚类算法模型来提升离线分类的效率，同时验证分类特征的精确率等特性，具体分析参考 3.1 节的算法研究。离线分类模型的架构如图 2 所示。

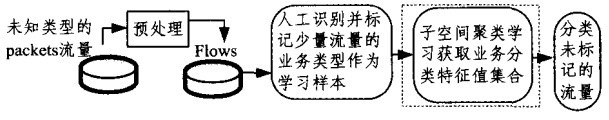

图 2 离线分类模型

图 2 中的离线分类模型的目标是：对网络中无法识别业务类型的流量，采用人工方式标记少量的样本数据，然后通过子空间聚类算法进行业务特征学习，最后采用各类业务特征对未标记的样本数据进行分类。在后续的实验中，我们验证了这类算法在进行离线流量分类时具有精确率高、召回率良好、分类业务精细程度高等特点。需要展开讨论的是，能够以一种比较少的人工方式提取业务特征，为什么不把子空间聚类算法获得的特征作为在线匹配计算的特征库？我们认为子空间聚类方法提取业务特征是网络流量的数值型的业务特征（包括包长度、时间间隔等），这类业务特征容易随着网络环境的不同而不同，这就意味着子空间聚类算法所提取的业务特征可扩展性不足，仅适合用于当前相应的网络环境中进行离线分类，我们建议将这类方法加入到 DPI 分类器模型中。因为 DPI 分类器采用的特征库是业务类型的公共字段特征，子空间聚类算法能够有效提高离线分类的效率并保证提取公共字段特征的准确率，可以辅助 DPI 分类器更准确、快速地实现特征库更新，因此对于增强 DPI 分类器的网络环境适应能力有重大意义。

3 使用子空间聚类方法构造离线分类器

本章将首先分析选择使用子空间聚类算法进行离线分类的缘由。然后我们将涉及一种子空间聚类算法-FPCLIQUE（Fast Pruning based CLUstering In QUEst）。主要设计子空间聚类参数的模型，目标是提升子空间聚类算法的效率，在第 4 节中，我们将通过实验验证子空间聚类算法的性能。

3.1 子空间聚类算法研究

机器学习方法可以分为有监督、无监督和半监督 3 种类型。

对于有监督类型的机器学习方法，运用到网络流量分类特征挖掘上的有很多。文献[10,11,18]介绍了 Naïve Bayes，C5.0，Bayesian Network 和 Naïve Bayes Tree 等算法。但是我们发现它们都有缺陷：1）所有采集到的样本数据必须标记业务类型；2）每一类应用类型的样本流量需要尽可能的丰富，以保证包含业务的关键特征。这就需要采集大量的样本数据和消耗大量的计算资源，而且大量样本带来的噪音数据会导致有监督机器学习算法过度学习，形成错误的业务特征集合。因此，我们认为有监督机器学习算法并不适合用于辅助构建离线分类模型。

另一类算法是无监督算法。文献[12,13]是聚类算法在网络流量分类中的运用，从实验分析结果可以看出，虽然无监督算法不需要对样本数据进行业务标签标记就可以计算出业务特征，但从文献[13]中给出的业务特征精确率实验结果可以看出，无监督聚类算法得到的业务特征存在精确率低的缺点。我们分析聚类算法要获取高精确率特性的流量特征，有两个必须的要素：首先是样本数据自身必须包含将不同业务准确区分出来的属性特征集合，文献[13]的算法显然没有考虑计算出这些特征，它们将全部属性进行聚类计算，这就使业务特征受到噪音属性的影响，无法进行精确分类；其次，聚类算法都有相关的参数需要设置，而文献[13]的参数设置本身并没有成熟的模型，比如 K-means 算法中的中心点个数设置、DBSCAN 算法中的密度门限值、半径值的初始参数设定等，都是依靠人的经验判断来设置，缺乏可理解性，容易对聚类结果造成不利影响，使得在线流量分类的熵值很大，甚至完全无法分类。所以我们必须对无监督算法做改进。

半监督方法就是对一部分由人工标记好业务类型的样本数据，使用聚类算法提取潜在的、划分业务类别效果最好的属性集合。然后用这些相关程度好、聚类效果明显的业务特征对其余的大量没有类别标记的数据进行分类。这类方法可以看作是对无监督算法的改进，它计算当前业务流量区别于其他业务流量的特征属性集合，消除了噪音属性值带来的不利影响，保证了分类的精确性和可靠性。这类方法可以更好地帮助在线分类器适应不断发生演进和变化的网络环境，更高效、准确地实现业务特征库的更新。其中最典型的就是子空间聚类方法。

综上分析，在本文中，我们将采用子空间聚类方法来学习样本数据的业务特征，测试业务特征的精确率等性能，验证这类方法在离线分类中的有效性。

3.2 子空间聚类算法介绍

子空聚类算法分为两类，一类是自底向上的方法（Bottom-Up），它是从低维度子空间向高维度子空间计算的过程，相关算法包括 CLIQUE，ENCLUS，MAFIA，CBF，CLTree，DOC 等；另一类是自顶向下的方法（Top-Down），它是从高维度中去除无关的属性的计算方式，相关算法包括 PROCLUS，ORCLUS，FINDIT，COSA 等。上述所有算法可以参考文献[19]的具体介绍内容。

一条 flow 已经可以总结出近 250 种数值属性信息[5]，而已有的流分类算法研究中，可用于业务分类的属性却很小，一般不超过 10 个。因此自底向上的子空间聚类算法更合适网络流量的子空间特征计算。算法流程如下：

（1）将整个空间进行网格化（grid-based），划分出微小的

凸形或者球形单元；

(2)在低维度空间中分析单元中正例样本数据的密度，选取稠密单元(dense unit)，用于计算高维度子空间的新单元；

(3)算法在当前维度空间集合中的单元无法形成任何高维度的单元时停止；

(4)分析最大子空间中的数据分布，计算聚类规则。

本文利用已有的 CLIQUE 子空间聚类算法[21]的聚类思想，设计了一种新的、适合用于网络流量离线分类的算法 FPCLQUE(Fast Pruning based CLUstering In QUEst)。

已有 CLIQUE 算法不适合用于流量分类，原因在于：一方面，在输入参数的设定上它采取人工确定的方式，这使得参数缺乏可理解性，同时也导致算法计算后的聚类结果精确率不高；另一方面，该算法计算子空间的过程中并没有限定子空间应该达到的覆盖率最低值，而是一直计算直到已有子空间无法聚合出新的子空间为止，这会导致算法丢失大量的有效正例数据特征。这些缺陷使得算法不能够直接应用到网络流量的离线分类中，所以本文对于算法计算过程中所需的关键参数进行了模型设计，对算法的执行条件进行了限定。在本文的第 4 节，我们将验证改进后的算法 FPCLIQUE 计算得到的流量业务特征具有高精确率的特点，同时保障了参数的可理解性、子空间计算的高效性和准确性。

3.3 FPCLIQUE 算法实现原理

FPCLIQUE 子空间聚类算法对样本数据空间、聚类结果的相关定义，文献[21]进行了详细的描述，本文不做赘述。下面将介绍算法实现中所需关键参数的计算原理和聚类结果的计算原理。

(1)确定等长区间的个数

本文从 flow 中提取了 33 种数值属性，具体参考 4.1 节的说明。这些属性的组合构成了样本空间，每个属性的取值范围按照样本数据在该维度上的最小值和最大值进行设定。算法的第一步是将每个维度按照取值范围划设出个数相同的等长区间，文献[23]提出了一种自适应的区间划分方法，它首先需要一个统一的等分区间个数，然后在相邻两个单元(units)之间的点个数相差不超过一定比例的前提下进行自适应的单元合并重组，此时产生了另外一个参数，即相差的比例，所以我们不考虑采用文献[23]中提及的网格单元个数计算方法，使用统一划分区间个数的方式更为简单可控。在本文中，针对提取的属性集合和测试结果，算法将每个维度按照取值范围划分出 10^4 个等长区间、k 个不同属性的等长区间组合成对应 k 维子空间的一个单元。

(2)确定单元的密度门限值

在每个 k 维子空间中，样本覆盖率高的单元才有更大的可能性用于高维度子空间单元的计算，因此需要设定密度门限值。文献[21,23]采用人工方式设定密度门限值，这种做法是不可取的；文献[22]则通过统计学规律认为每个网格中的平均的样本点至少为 35 个，对于高维度空间而言，为了达到这个均值，必须存储大量的数据样本，这不符合本文的要求。我们采用熵计算模型[23]，将一维子空间集合中样本覆盖率为 0 的单元删除，将其余的单元按照密度从大到小排序；通过表 1 定义相关的变量，我们利用式(1)，初始假定所有单元都是稠密单元，计算出初始熵值，其中 α 代表的是稠密单元的平均样本覆盖率，β 代表的是非稠密单元的平均样本覆盖率。接着逐步将低覆盖率的单元划分到非稠密网格单元集合中重新计算熵值。上述计算过程将一直持续到 α 值小于 β 值时结束，此时取得最小熵值的单元密度值就是密度门限值。

表 1 一维子空间网格单元相关定义

n	1 维子空间集合所有包含样本点的网格单元数量
k	稠密单元的数量
$p_1, p_2, \cdots, p_k$	每个 p 值代表的是稠密单元的样本覆盖程度(样本点的个数)
$p_{k+1}, \cdots, p_n$	每个 p 值代表的是非稠密单元的样本覆盖程度(样本点的个数)

$$H=-\left(\sum_{i=1}^{k} p_i \log p_i+\sum_{j=k+1}^{n} p_j \log p_j\right)$$
$$=-[k\alpha\log\alpha+(n-k)\beta\log\beta] \quad (1)$$

(3)确定子空间的样本覆盖率门限值

FPCLIQUE 算法从 k 维子空间集合中，去除每个 k 维子空间中的非稠密单元，对于不包含任何稠密单元的 k 维空间也会被删除。为了更有效地保障子空间能够在聚类时得到较大的样本覆盖率，提升子空间计算的效率，本文引入另一个数学模型，它对所有 k 维子空间按照样本覆盖率从大到小排序，按照图 3 和式(2)－式(4)计算覆盖率分割点。算法将把子空间划分成为两个集合，一个是高覆盖率的集合，定义为可选集合 S(selected subspaces set)，另一个是低覆盖率的集合，定义为剪枝集合 P(prune subspaces set)，分别计算两个集合的平均覆盖率，取整数上界作为均值结果，如式(2)和式(3)所示，接着算法会计算对应集合中每个子空间覆盖率与均值之间的方差，并计算出均值，最后将它们全部相加就得到 CP 值(cut point 值，相当于子空间覆盖率门限值)。初始时，算法将所有的 k 维子空间设置成可选集合并计算出初始的 CP 值，然后逐步剔除低覆盖率的子空间，重新计算 CP 值直到取得最小 CP 值，于是就可以在排好序的数组中得到相应的子空间覆盖率门限值。

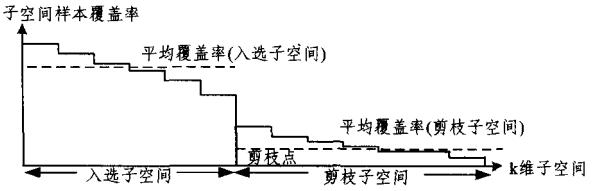

图 3 子空间覆盖率门限值计算

$$AVE_S(i)=\lceil\left(\sum_{1\leqslant j\leqslant i} x_j\right)/i\rceil \quad (2)$$

$$AVE_P(i)=\lceil\left(\sum_{i+1\leqslant j\leqslant m} x_j\right)/(m-i)\rceil \quad (3)$$

$$CP(i)=AVE_S(i)+\frac{1}{i}\sum_{1\leqslant j\leqslant i}(x_j-AVE_S(i))^2+$$
$$AVE_P(i)+\frac{1}{m-i}\sum_{i+1\leqslant j\leqslant m}(x_j-AVE_P(i))^2 \quad (4)$$

(4)子空间聚类结果的计算

如果出现以下 4 种情况之一时，算法结束子空间计算：

- 当前维度为 k 的子空间集合无法合成维度数目为 $k+1$ 的子空间；
- 合成之后的新高维度子空间没有稠密单元；
- 当前子空间的维度数目已经是最大值；
- 对于当前维度为 k 的子空间集合，是否有必要继续进行高维度子空间的计算，还必须给予一定的限定条件，这个条

件就是样本覆盖率。依据文献[21]中的引理，本文限定当 k 维子空间集合中没有子空间的样本覆盖率大于等于 75% 的时候，不再继续进行高维度子空间计算。

此时，对拥有最大样本覆盖率的 k 维子空间需要进行聚类结果的计算，计算过程是循环搜索一个 k 维子空间中具有邻接关系的稠密单元集合，集合之间如果存在共面的稠密单元对，则进行合并，当所有稠密单元都处理过之后，循环结束。

(5)FPCLIQUE 算法聚类结果表达式

在得到聚类结果之后，需要对每个 cluster 进行准确的描述。以图 4 作为例子，图中阴影部分的单元集合就是一个聚类结果。首先，从一个聚类结果中任意抽取一个未经任何区域包含的稠密单元，比如图 4 中选择 u 单元，然后从这个稠密单元第一个维度的左方和右方进行扩展，形成区域 A，接着以 A 区域作为起点，在第二个维度上扩展区域 A，得到了区域 B，它就是包含单元 u 的最大区域(maximal region)，接着继续搜索未被覆盖的其他稠密单元的最大覆盖区域，整个过程持续到 cluster 中的所有稠密单元都处理过为止。每个最大区域可以这样表示：用一个在子空间对应各个维度上取值最小的单元作为起点，最大覆盖区域中各个维度的最长跨度信息作为边进行表示。$\langle(ID_{a1}\ u_{a1}\ len_{a1}),\cdots,(ID_{ak}\ u_{ak}\ len_{ak})\rangle$，圆括号中的前两个信息是最大覆盖区域的起始单元相关信息，ID_{ai} 代表的是起点稠密单元对应的维度编号，u_{ai} 代表的是稠密单元在该维度上所属的区间编号，len_{ai} 代表的是最大覆盖区域在该维度上的连续区间总个数，$1\leqslant i\leqslant k$。

一个 cluster 能够用多个最大覆盖区域表示，最大覆盖区域之间在计算时会出现重叠的情况，所以需要把所有最大覆盖区域按照覆盖率进行排序，然后去除冗余区域得到聚类结果的最小化描述。k 维子空间中所有的聚类结果，可以由所有最大覆盖区域通过析取的方式进行表示，作为对应的业务流量在 k 维子空间中的业务聚类特征，析取范式(DNF)的描述方式如下：

$$\langle(ID_{a1}\ u_{a1}\ len_{a1}),\cdots,(ID_{ak}\ u_{ak}\ len_{ak})\rangle V\langle(ID_{b1}\ u_{b1}\ len_{b1}),\cdots,(ID_{bk}\ u_{bk}\ len_{bk})\rangle V\cdots$$

其中，$1\leqslant i\leqslant k$。

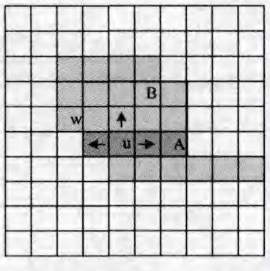

图 4　搜索每个稠密单元的最大覆盖区域

3.4　FPCLIQUE 算法效率评估

由于采取了剪枝计算，使得子空间聚类算法在计算稠密单元时聚合成高维度子空间的算法复杂度。

首先，对于 k 维子空间而言，每个 k 维子空间集合中，在计算 k+1 维子空间集合过程中进行了稠密单元的划定，因此算法整个的复杂度由原来的 $O(k*g(k)*f(k)^2)$ 降低为了 $O(k*g(k)*a^2)$，其中 $g(k)$ 代表能够聚合的子空间个数值，它随着维度的增加呈现出先增加、后递减的函数关系，假设 n 为总的维度个数，属于常量，k 维子空间集合中能够用于子空间聚合计算的子空间总个数的计算公式则是 $0.5\times k\times(n-k)\times(n-k+1)$，$f(k)$ 为划分当前 k 维子空间的单元的总个数，它是以等长区间个数 m 这个常数为底数、维度个数为指数的指数级函数 m^k，经过对非稠密单元的剪枝优化之后，需要比对的单元大量减少，a 是经过熵模型计算之后得到的稠密单元的数量级，为常数；

其次，在进行 k+1 维子空间构造的过程中，又采用了删除与业务类型关联性比较低的即对学习样本覆盖率比较低的子空间，使得子空间的个数 $0.5\times k\times(n-k)\times(n-k+1)$ 值进一步缩小为常数级别 b。

最后，我们对于子空间聚合又添加了相应的停止条件，使得 k 值进一步减小，在实验中聚合出的子空间维度平均约为整个维度集合的 20%～30%。

综合分析可知，整个 FPCLIQUE 算法的计算效率已经提升为 $O(c)$，c 为常数级别，算法的效率高，而且在对不同业务类别特征集合的计算时，FPCLIQUE 算法能够并行执行。

4　聚类特征性能评估

本章将测试和评估子空间聚类算法，得到业务特征的性能。4.1 节主要介绍样本数据的获得；4.2 节介绍性能评估标准；4.3 节将对 FPCLIQUE 算法得到的业务特征规则与 K-means、DBSCAN 聚类方法得到的业务特征规则做对比和分析，验证子空间聚类算法在流量离线分类中的精确率和分类业务的精细化程度等特性。

4.1 数据集合

样本数据的相关制作工作包括获取 packets，将 packets 重组为 flows，去除无关噪音数据，利用 flows 中 packets 的类型为 flow 打上业务标签，最后提取 flow 中潜在的可以作为分类依据的数值属性[5,12]形成流记录。由于本文的重点是测试 FPCLIQUE 算法的性能，我们并没有涉及人工如何标记样本数据的工作。对于样本数据的业务类型，我们是通过在校园网的汇聚端口处进行数据包采集并利用采集设备进行应用标签的标记获得的，同时采集设备去除掉了应用层的数据内容，减少了存储压力。需要强调的是如果按照图 2 进行采集时，对未知类型的数据包必须进行全包采集，并且还需要对部分数据包的业务类型进行人工标记。

本文提出的 FPCLIQUE 离线分类算法的目的是尽可能降低人工参与程度。样本数据从采集设备获取后按照一定格式保存到 pcap 文件中，然后我们按照五元组定义对数据包进行了流的重组，去除无关的噪音信息，选取包数量至少为 10 个的流作为样本流，这些样本流既有 tcp 流，也有 udp 流，每条流都是单向的，我们通过检测 flow 中带应用标签的 packets 的分布情况来确定一条 flow 所属的业务类型。我们对样本数据进行了抽样，表 2 列举了抽样出的样本数据的相关统计信息，表 3 则统计从表 2 的数据中随机抽取出的业务流量的分布特征，这些业务流量涵盖了目前网络中较为流行的应用类型，比如传统的 Web 服务、网络多线程下载、P2P 服务、在线视频流量等。对于表 3 中有些业务的数据既有 tcp 流，也有 udp 流，它们之间的业务特征是否相同，以及对于流数量和字节占比较小的业务流量是否能够得到对应的业务属性特征集合，加密数据是否也能够得到独特的业务数值属性特征等问题，我们将在下节的实验中进行测试和验证。

文献[5]概括总结了 200 多种 flow 的数值属性特征，我

们从样本数据流中总结了 33 个数值属性、涉及包大小分布、时间间隔分布、间隔时间分布等内容，这些特征具有普遍性，具备作为分类特征的潜在可能性[10,11,13]。我们针对每种应用业务制作了正例学习样本和测试样本，如图 5 所示。在试验中，对于每种业务类型数据，我们随机抽取 1000 到 3000 条流数量作为对应业务流量分类器的学习样本，其余的数据流作为对应的测试样本。学习样本的大小可以由用户进行设定，以此测试是否采用较少的样本数量就能得到精确率高的业务特征。

表 2　总体样本数据统计

流数量统计	TCP 流数量统计	UDP 流数量统计
	746067	726946
1473013	TCP 流数量比重	UDP 流数量比重
	50.65%	49.35%
总字节数(GB)	TCP 流总字节数(GB)	UDP 流总字节数(GB)
	117.3631	131.8344
249.1975	TCP 流总字节数比重	UDP 流总字节数比重
	47.10%	52.90%

表 3　典型应用样本数据抽样统计

TCP 应用类型	流数量	占比	总字节(GB)	占比
http	279315	37.44%	8.5651	7.30%
https	15761	2.11%	0.27	0.23%
BT 下载	18123	2.43%	16.5064	14.06%
非 P2P 下载	30114	4.04%	5.4514	4.64%
utorrent tcp	25505	3.42%	11.0733	9.44%
统计	368818	49.43%	41.8662	35.67%
UDP 应用类型	流数量	占比	总字节(GB)	占比
加密 P2P 业务	63921	8.79%	51.236	38.86%
sina video	25190	3.47%	3.3749	2.56%
tvkoo	13571	1.87%	3.7172	2.82%
utorrent udp	27367	3.76%	28.8031	21.85%
统计	130049	17.89%	87.1312	66.09%

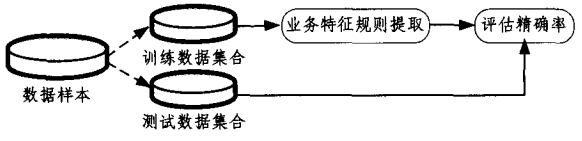

图 5　样本数据的划分与实验流程

4.2　评估标准

对于每种分类器而言，它都有如表 4 所示的评估结果，这 4 种判断情况对应的定义是：

• True Positive 指的是原数据属于某一类别，分类器判断该数据也属于该类别；

• False Negative 指的是原数据属于某一类别，分类器没有识别出来(漏判)；

• False Positive 指的是原数据不属于某一类别，分类器判断属于该类别(误判)；

• True Negative 指的是原数据不属于某一类别，分类器判断也不属于该类别。

表 4　样本类别判断真值表

布尔值	判别 1	判别 0	合计
实际 1	True Positive(TP)	False Negative(FN)	Actual Positive(TP+FN)
实际 0	False Positive(FP)	True Negative(TN)	Actual Negative(FP+TN)
合计	Predicted Positive (TP+FP)	Predicted Negative (FN+TN)	TP+FP+FN+TN= total flows

针对分类器得到的判别结果，我们有如下的指标可以用于评估算法计算得到的流量分类器性能：

• 失误率：是指误判和漏判的流量占整体测试样本的比例，$Error\ rate=(FP+FN)/Total\ flows$；

• 准确率：是指被正确识别判断的流量占整体测试样本的比例，$Accuracy=(TP+TN)/Total\ flows$；

• 精确率：是指实际确实为正例的样本占预测为正例的样本的比例，$Precision=TP/(TP+FP)$；

• 召回率：是指判断正确的正例样本占所有被判断为正例的样本的比例，$Recall=TP/(TP+FN)$；

• 特异性：指判断正确的反例样本占全部被判断为反例的样本的比例，$Specificity=TN/(FP+TN)$。

本文测试关注的重点是精确率和召回率。接近 100%的精确率，以及保持较高的召回率才能保证离线分类的可靠性。

4.3　实验测试结果分析与对比

所有业务经过多次测试之后得到的平均精确率和召回率最后的结果如图 6 所示。我们采用相同的学习样本集合和测试样本集合，以及相同的属性集合，用 K-means 算法和 DBSCAN 算法进行了测试对比，结果如图 7 和图 8 所示。通过对比，我们发现在精确率方面，两个无监督分类算法的精确率都很低，这充分说明了两种算法在应对高维度空间聚类时的缺陷，因为它们都是将所有属性进行了距离计算，而这些属性并不一定都是适合用于划分业务类别的，此时，这些无关业务类别的特征属性值就会变成噪音数据，对分类结果产生不利的影响。

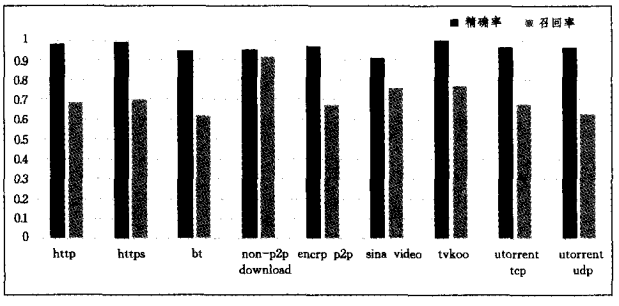

图 6　FPCLIQUE 子空间聚类算法测试结果

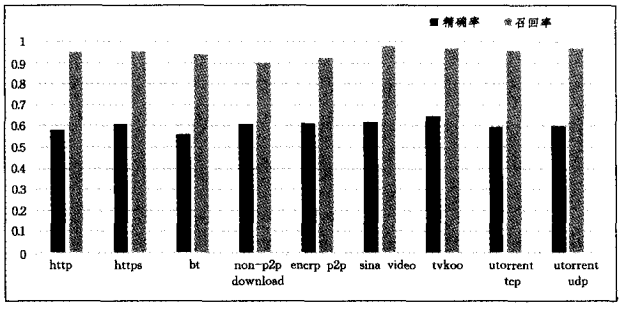

图 7　k-means 算法测试结果

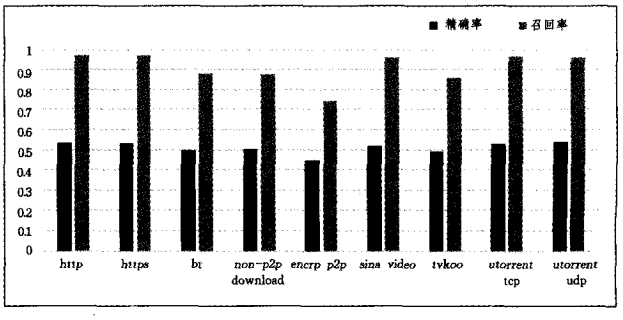

图 8　DBSCAN 算法测试结果

K-means 和 DBSCAN 算法都需要相应的初始化参数，对于 K-means 算法而言，需要输入 k 的个数；DBSCAN 算法则需要输入样本点成为核心对象(core object)的两个条件，即样本点的邻域半径 eps 值和邻域内至少应该包含的点数 MinPts。在文献[13]中，这些参数值都已经根据最优的测试结果进行了设定，但是引用文章中最优的 K 值 150 和最优的 eps 值 0.03 以及 MinPts 值 3 时，我们发现这些参数值并非让聚类算法获得有效的分类规则，这些分类规则不能够达到和文献[13]一样的分类精确率。这是因为相应输入参数值对应的样本集合发生了变化，比如样本数量和样本属性空间维度的变化，所以所有参数都需要重新调整，但是这些参数的调整本身又缺乏可靠的数学模型计算，所以我们只能通过多次实验确定参数取值，然而图 7 和图 8 始终显示的是精确率低的实验结果。这一方面是高维度空间噪音属性数据值的影响造成的，另一方面是参数的设定凭借经验，缺乏依据，导致提取的业务特征不准确造成的。

而使用 FPCLIQUE 算法进行聚类获得业务特征规则之后，其对测试样本的分类结果如图 6 所示。

第一，对于每种业务所需的学习样本数量，我们仅随机抽取 1000 到 3000 条流数据就可以获得图 6 的实验结果，这说明算法本身可以有效地降低人工参与程度，而且算法本身所需的参数的智能化程度好，有着良好的运算模型，这是一个非常大的优点，子空间聚类算法能够对流量进行高效的业务特征聚类计算。

第二，我们看到子空间聚类算法在提取业务特征中进行业务分类时，召回率偏低。这是因为 FPCLIQUE 算法在进行子空间聚类的过程中进行了大量的剪枝计算，而剪枝在去除噪音的同时，不可避免地会忽略部分正例样本及其对应的特征。我们建议将这类算法用于离线分类，并辅助 DPI 分类器提高特征库更新效率，这也是考虑到了子空间聚类算法计算出的业务特征在分类流量上召回率相对偏低的问题，将其应用在 DPI 模型中，可以规避这种不足，因为从理论上说从 100%的业务样本数据和 60%的业务样本数据能够抽取到的业务公共字段集合应该是相同的；另一方面，从图 6 的实验结果分析可知，FPCLIQUE 算法获得的业务特征在分类数据时的精确率很高(平均达到 95%以上)，而为了保证公共字段抽取的准确性，分类特征必须在分类时能够保持高精确率，这样才能达到降低噪音数据对正例数据的影响，提高 DPI 公共业务特征抽取的准确性。FPCLIQUE 算法抽取的业务特征又确实满足这种要求。所以我们建议将这类算法用于辅助提升 DPI 分类器特征库的更新效率上。

第三，我们还对 FPCLIQUE 算法的召回率与精确率之间的关系进行了多组测试，其中每组测试由所有应用的精确率和召回率的测试结果平均值组成。如图 9 所示，我们针对每种应用业务的流量，通过逐步增加学习样本来提高正例样本的覆盖率。我们发现 FPCLIQUE 算法得出的业务特征在样本平均召回率超过 60%，并且平均精确率始终保持着接近 100%的水平。这就说明子空间聚类算法能够通过较少的学习样本学习得到可靠、稳定的业务特征规则，每种业务特征规则能够对流量进行准确的分类。

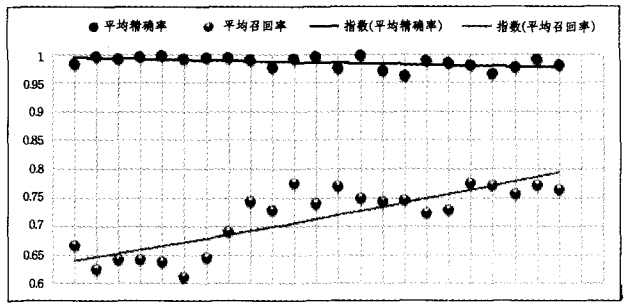

图 9　平均召回率与平均精确率之间的关系

最后，我们与文献[24]的子空间聚类算法进行了分析对比。第一，在业务分类种类上，我们验证了 FPCLIQUE 算法确实能够做到更为精细化的业务分类目的，而文献[24]中仅仅是针对 4 种典型业务数据进行了分类；第二，FPCLIDQUE 算法需要输入的参数仅有一个，而且该参数易于调整，文献[24]中的 subflow 子空间聚类算法采用 DBSCAN 思想计算聚类特征，算法需要手动控制输入的参数相对较多，不利于部署调整。最后，我们也发现，subflow 算法本身的样本召回率也比较低，实际上也并不适合用于构建在线分类器。

结束语　本文通过分析已有的分类器模型存在特征库更新效率低下的缺陷，引入了子空间聚类方法，我们的工作成果包括：第一，测试并验证了网络流量在业务上确实存在可以进行聚类计算的业务特征，并且这些业务特征具有接近 100%的业务分类精确率；第二，我们设计的 FPCLIQUE 算法可以采用较少的数据样本来进行聚类计算，这使得离线分类的效率能够得到更好的提升；第三，实现了算法输入参数的智能化程度，相比已有的 3 种聚类算法，它具有更好的网络环境适应能力。

我们希望将 FPCLIQUE 算法引入到 DPI 分类模型中，以提升特征库的更新效率。当然本文还存在不足，需要在后续的工作中继续完善。首先，在样本数据上，我们仅仅通过设备从校园网获得了有精细业务分类的数据，数据集较为单一，我们还需要获取更多有精细业务分类的数据对算法的提取业务特征性能做进一步深入的验证。其次，DPI 模型本身也并不是完美的，黑客的一些看似合法的"非法"数据容易通过 DPI 匹配混入网络中。最后，我们对于如何进行人工标记数据，尤其是加密数据本身也存在疑问。任何技术都存在相对意义上的不足，在后续的工作中，我们需要进一步完善上述工作，使 DPI 分类器真正能够适应业务特征不断变化的网络环境，提供安全有保障的流量分类结果。

参 考 文 献

[1] Chandrashekar J, Zhang Z L, Zhenhai D, et al. Towards a service oriented internet[J]. IEICE transactions on communications, 2006, 89(9): 2292-2299

[2] Srinivasan S R, Lee J W, Liu E, et al. Netserv: Dynamically deploying in-network services[C]// Proceedings of the 2009 workshop on Re-architecting the internet. ACM, 2009: 37-42

(下转第 319 页)

[6] McCallum A, Nigam K, Ungar L H. Efficient clustering of high-dimensional data sets with application to reference matching[C]// Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2000: 169-178

[7] 李应安. 基于 MapReduce 的聚类算法的并行化研究[D]. 广州: 中山大学, 2010

[8] Ruspini E H. Numerical methods for fuzzy clustering[J]. Information Sciences, 1970, 2(3): 319-350

[9] 赵洪昌. 云环境下的关联分析和模糊聚类研究[D]. 南京: 南京信息工程大学, 2013

[10] 陈爱平. 基于 Hadoop 的聚类算法并行化分析及应用研究[D]. 成都: 电子科技大学, 2012

[11] Ohmann T, Rahal I. Efficient clustering-based source code plagiarism detection using PIY[J]. Knowledge and Information Systems, 2014, 3: 1-28

[12] 余丹. 关于查全率和查准率的新认识[J]. 西南民族大学学报, 2009(2): 283-285

(上接第 306 页)

[3] Femminella M, Francescangeli R, Reali G, et al. An enabling platform for autonomic management of the future internet[J]. Network, IEEE, 2011, 25(6): 24-32

[4] Arthur C, Carlos K, Stênio F, et al. A Survey on Internet Traffic Identification and Classification [J]. Communications Surveys and Tutorials, IEEE, 2009, 11(3): 37-52

[5] Moore A, Zuev D, Crogan M. Discriminators for use in flow-based classification[M]. Queen Mary and Westfield College, Department of Computer Science, 2005

[6] Szabó G, Szabó I, Orincsay D. Accurate traffic classification [C]// IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2007 (WoWMoM 2007). IEEE, 2007: 1-8

[7] Kumar S, Dharmapurikar S, Yu F, et al. Algorithms to accelerate multiple regular expressions matching for deep packet inspection [J]. ACM SIGCOMM Computer Communication Review, 2006, 36(4): 339-350

[8] Smith R, Estan C, Jha S, et al. Deflating the big bang: fast and scalable deep packet inspection with extended finite automata [J]. ACM SIGCOMM Computer Communication Review, 2008, 38(4): 207-218

[9] 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. ACM, 2005: 197-202

[10] Moore A W, Zuev D. Internet traffic classification using bayesian analysis techniques[J]. ACM SIGMETRICS Performance Evaluation Review, 2005, 33(1): 50-60

[11] Williams N, Zander S, Armitage G. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification[J]. ACM SIGCOMM Computer Communication Review, 2006, 36(5): 5-16

[12] Bernaille L, Teixeira R, Akodkenou I, et al. Traffic classification on the fly[J]. ACM SIGCOMM Computer Communication Review, 2006, 36(2): 23-26

[13] Erman J, Arlitt M, Mahanti A. Traffic classification using clustering algorithms [C] // Proceedings of the 2006 SIGCOMM workshop on Mining network data. ACM, 2006: 281-286

[14] Park B C, Won Y J, Kim M S, et al. Towards automated application signature generation for traffic identification [C] // IEEE Network Operations and Management Symposium, 2008 (NOMS 2008). IEEE, 2008: 160-167

[15] Ye M, Xu K, Wu J, et al. Autosig-automatically generating signatures for applications[C] // Ninth IEEE International Conference on Computer and Information Technology, 2009 (CIT' 09). IEEE, 2009, 2: 104-109

[16] Szabó G, Turányi Z, Toka L, et al. Automatic protocol signature generation framework for deep packet inspection[C] // Proceedings of the 5th International ICST Conference on Performance Evaluation Methodologies and Tools. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011: 291-299

[17] Karagiannis T, Papagiannaki K, Faloutsos M. BLINC: multilevel traffic classification in the dark[J]. ACM SIGCOMM Computer Communication Review, 2005, 35(4): 229-240

[18] Bujlow T, Riaz T, Pedersen J M. A method for classification of network traffic based on C5. 0 Machine Learning Algorithm[C]// 2012 International Conference on Computing, Networking and Communications (ICNC). IEEE, 2012: 237-241

[19] Parsons L, Haque E, Liu H. Subspace clustering for high dimensional data: a review[J]. ACM SIGKDD Explorations Newsletter, 2004, 6(1): 90-105

[20] Müller E, Günnemann S, Assent I, et al. Evaluating clustering in subspace projections of high dimensional data[J]. Proceedings of the VLDB Endowment, 2009, 2(1): 1270-1281

[21] Agrawal R, Gehrke J E, Gunopulos D, et al. Automatic subspace clustering of high dimensional data for data mining applications: U. S. Patent 6, 003, 029[P]. 1999-12-14

[22] Cheng C H, Fu A W, Zhang Y. Entropy-based subspace clustering for mining numerical data[C] // Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 1999: 84-93

[23] Goil S, Nagesh H, Choudhary A. MAFIA: Efficient and scalable subspace clustering for very large data sets[C]// Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 1999: 443-452

[24] Xie G, Iliofotou M, Keralapura R, et al. SubFlow: towards practical flow-level traffic classification[C] // INFOCOM, 2012 Proceedings IEEE. IEEE, 2012: 2541-2545