


第六章 网络流量分析

目录

- 流量分析介绍
- 网络流量的采集方法
- 常用的网络流量分析模型及方法
- 小结

流量分析介绍

1. 网络流量分析
2. 网络流量分析的目的
3. 网络流量分析的现状
4. 网络流量分析的流程

- 随着网络基础设施提升和移动互联网的发展，如何有效的识别和管理网络上的流量变得越来越迫切。
- 网络流量分类(Network Traffic Classification)  热点

网络流量分析

- 需求分析
- 数据采集
- 数据挖掘
- 结果评估

- **网络流量分析**是动态适应，不断调整的处理过程
- **网络流量分类**是基于TCP/IP的互联网络中，按照网络的应用类型将网络通信产生的双向TCP流或UDP流进行分类

常见网络应用类型FTP、DNS、WWW、P2P等

网络流量分类

目前流分类算法

- 基于端口号映射、基于有效载荷分析、基于机器学习等



采用动态端口、协议加密

传统方法达不到满意效果

可靠的流量分类

- 查看数据包的内容（涉及隐私）



网络流量的分类

- **在线和离线分析**

- **在线检测**是在运行着的网络链路上实时采集包追踪数据或IP数据流
- 较小空间复杂性和较低的计算复杂性
- 如何设计在线检测算法以适应高带宽主干网络链路的检测需求 ！
- 基于特征/行为的检测技术 ➡ 入侵检测工具
- 基于采样、哈希、概略、包分类等

网络流量的分类

- 在线和离线分析

- **离线检测**是指对网络流量数据进行离线分析，检测网络异常及对网络进行性能分析和预测。

常基于定时采集的SNMP

- 数据分析和处理的时间周期比较长



网络管理及安全分析和预测方面

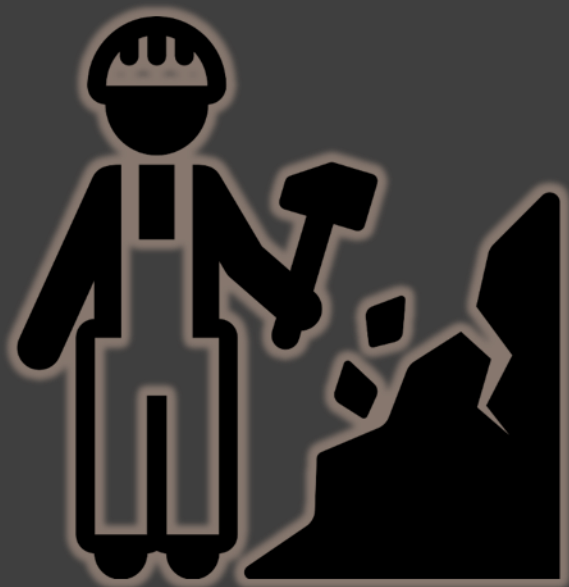
- 统计分析技术及流挖掘技术实现

网络流量的分类

- **基于流的分析方法**
- **基于非流的分析方法**

- **基于流**:输入的数据流进行预处理，每个数据包生成一个五元组，存入数据组，再使用相关算法分析，得出分析结果，以报告或者图表的方法显示出来
- **流(Flow)**: 在同一组特定源地址和目标地址、源端口和目的端口之间传输的，有固定协议类型，有开始和结束时间的数据包的集合

网络流量分析目的



- 帮助运营商了解网络流量的分布，带宽的使用情况，方便进行维护和计费等
- 帮助网络管理员了解网络流量分布，合理规划和升级网络，对应用进行管理
- 识别网络上的安全威胁（异常，恶意行为）和未知应用等

网络流量分析现状



- 国内外的网络流量分类技术主要有：端口分类，特征码分类，BLINC(Blind classification)分类，基于统计特征的机器学习方法，**基于数据挖掘的网络流量分析**等。

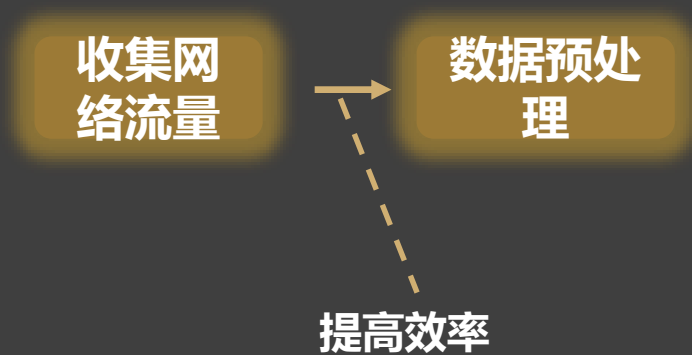


在网络流量分析中引入分类和聚类方法

网络流量分析流程

收集网
络流量

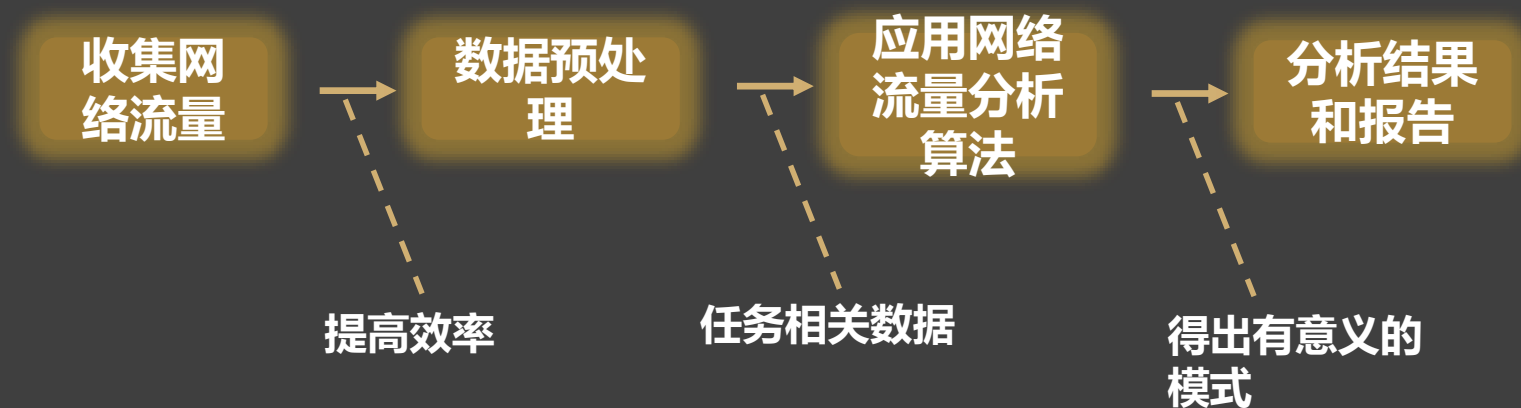
网络流量分析流程



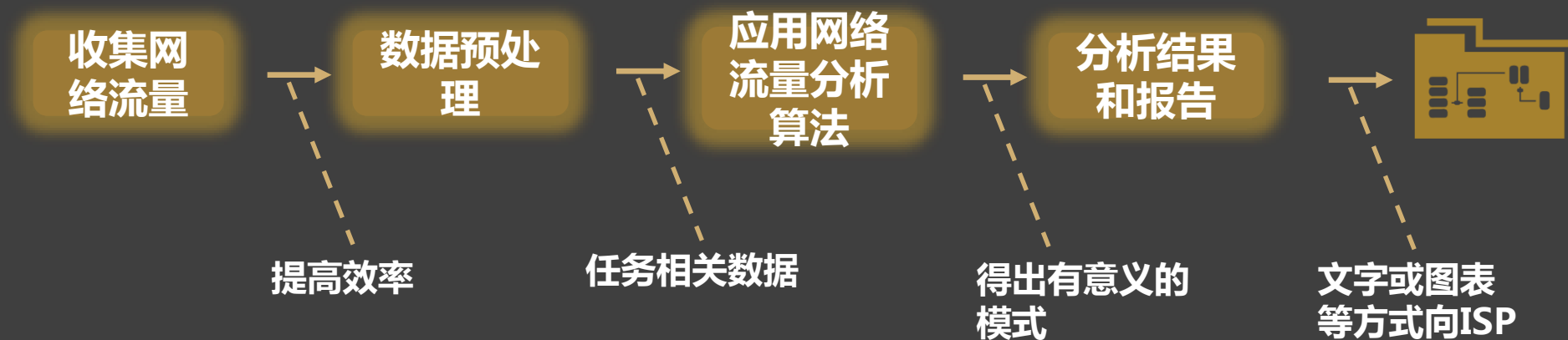
网络流量分析流程



网络流量分析流程



网络流量分析流程



网络流量的采集方法

1. 流量采集概述

2. 流量采集方法

3. 流量采集的问题

4. 网络流量数据集

- 数据包从发送方到接收方需要经过多个网络设备转发
- 合适的网络位置捕获网络流量
- 考虑采集数据的类型以及数据的TCP/IP协议层次

全部or部分采集  软件or硬件采集...

流量采集方法

✓个人用户

利用流量采集软件，Wireshark，Snort，Sniffer...

✓网络管理员和运营商

网络设备

基于端口的镜像

集线器数据的捕获



流量采集方法

• 存在的问题

效率

在高速骨干网上采集网络流量要求相应的网络设备具有更高的处理速度和能力

存储容量

不同的网络环境对存储设备的要求不一样

安全问题

为防范隐私泄露所采用的技术使得流量采集途径被限制

网络流量数据集

✓ <http://www.caida.org/data/>

CAIDA (The Cooperative Association for Internet Data Analysis)

✓ <http://www.ing.unibs.it/ntw/tools/traces/index.php>

Università degli Studi di Brescia

✓ <http://mawi.wide.ad.jp/mawi/>

WIDE (Widely Integrated Distributed Environment)

✓ <http://www.wand.net.nz/wits>

WITS (Waikato Internet Traffic Storage)

常用的网络流量分析模型及方法

1.流量分析模型

2.常用的流量分析方法

3.数据挖掘方法在流量分析中的应用

4.其他的流量分析方法

- 网络流量行为特征的分析还可以在不同测量粒度或者不同的层面上展开
- 比特级(Bit -level)的流量分析
- 分组级(Packet-level)的流量分析
- 流级(Flow-level)的流量分析

流量分析模型

小

粒度

大

- 比特级(Bit-level)的流量分析主要关注网络流量的数据特征，如网络线路的传输速率，吞吐量的变化等等。
- 分组级(Packet-level)的流量分析主要关注的是IP 分组的到达过程、延迟、抖动和丢包率等
- 流级(Flow-level)的流量分析，Flow的划分主要依据地址和应用协议展开，它主要关注流的到达过程、到达间隔及其局部的特征。

小

时间尺度

大

流量分析模型

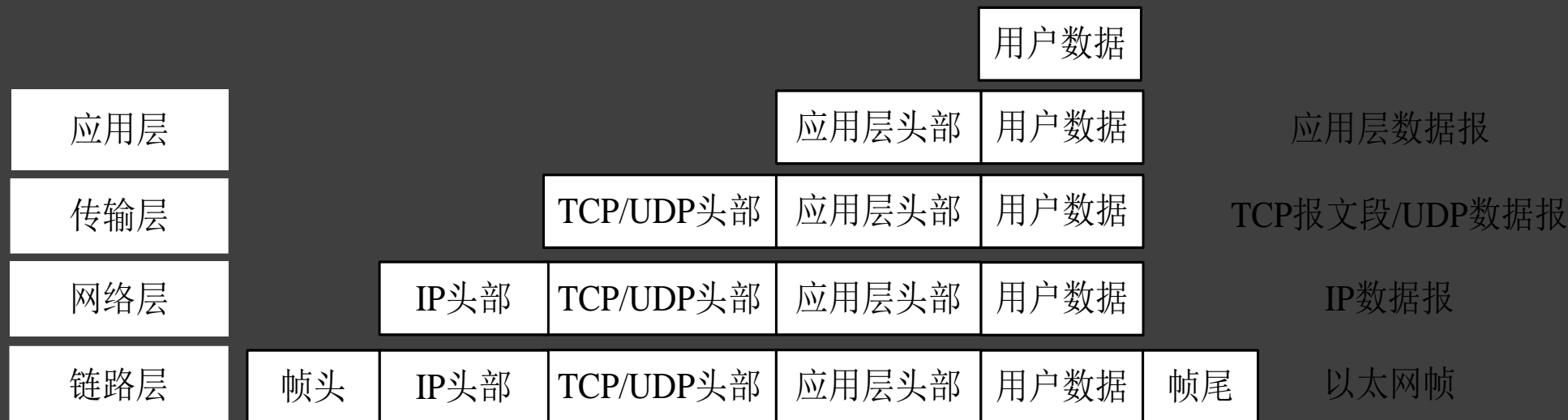
- 毫秒级的细时间粒度的网络流量行为主要受到网络协议的影响；
- 小时以上的粗时间粒度的网络流量行为主要受到外界因素的影响
- 两者之间的秒时间粒度上的网络流量则表现为自相似性。

常用的流量分析方法

- 1.基于端口的方法
- 2.基于特征码的方法
- 3.基于传输层的流量识别技术
- 4.利用统计特征的流量识别技术

基于端口的方法

- 基于端口的流量分类方法实现简单、判定速度快，且适于硬件实现，该方法一般用于高速网络的流量粗选。



TCP/IP数据的封装结构

基于端口的方法



TCP 头部格式



UDP头部格式

基于端口的方法

- IANA最初是按先到先得的原则分配服务名称，规定系统应用的端口号范围为0~1023，用户应用的端口号范围为1024~49151，动态端口号或私有端口号范围为49152~65535。
- 如今，IANA 端口号分配表中注册的一些用户应用端口已被新出现的应用服务所替代或占用，维基百科给出了一个更接近现实网络情况的端口服务映射表

常见端口列表

端口	描述	状态
20/TCP, UDP	文件传输协议 - 默认数据端口	官方
21/TCP, UDP	文件传输协议 - 控制端口	官方
22/TCP, UDP	SSH (Secure Shell) - 远程登录协议, 用于安全登录 文件传输(SCP, SFTP) 及端口重新定向	官方
23/TCP, UDP	Telnet 终端仿真协议 - 未加密文本通信	官方
25/TCP, UDP	SMTP (简单邮件传输协议) - 用于邮件服务器间的 电子邮件 传递	官方
53/TCP, UDP	DNS (域名服务系统)	官方
69/UDP	TFTP (小型文件传输协议)	官方
80/TCP	HTTP (超文本传输协议) - 用于传输网页	官方
81/TCP	HTTP 预备 (超文本传输协议)	官方
110/TCP	POP3 (“邮局协议”, 第3版) - 用于接收 电子邮件	官方
143/TCP, UDP	IMAP4 (Internet Message Access Protocol 4)-used for retrieving E-mails	官方
161/TCP, UDP	SNMP (Simple Network Management Protocol)	官方
162/TCP, UDP	SNMPTRAP	官方
220/TCP, UDP	IMAP , 交互邮件访问协议第3版	
443/TCP	HTTPS - HTTP over TLS/ SSL (加密传输)	官方
993/TCP	IMAP4 over SSL (encrypted transmission)	官方
995/TCP	POP3 over SSL (encrypted transmission)	官方

基于端口的方法遇到的问题

- ✓随着网络应用的发展与普及，大多数的网络应用允许用户手动选择来设置默认的端口号
- ✓许多新出现的网络应用为了躲避流量限制，往往会使用动态的端口来进行数据传输，而不是使用一个公共不变的端口，无法有效的识别网络流量
- ✓端口控制粒度太粗，易出错
- ✓通过端口方式能够识别的协议类型非常有限

基于特征码的方法

- 依据IP数据包中具有协议特征码进行流量识别。
- 特征码的识别方法主要用来识别P2P流量
- 通过分析捕获到的网络数据包，找到每个网络应用的固定特征码，利用这些特征码就能有效的识别不同的网络应用。
- 特征码识别技术是一种基于应用层信息的识别方法

基于特征码的方法

- 对于可以采用特征码识别的业务，必须对不同协议的数据包进行单独分析，因为它们的协议都是自定义的非标准协议。
- 特征码检测法适用于常见的应用，能识别出大部分的业务流量，如 eDonkey、eMule、KAZAA、BitTorrent、Gnutella 等。

基于特征码的方法

- 特点



检测准确率高，不受端口的变化影响



数据包的静态标识特征需要不断的更新和增加



高资源消耗

基于传输层的流量识别技术

BLINC方法基于签名，工作原理是：

- 基于主机的应用行为来分类网络连接，把主机模式分为三个层次：



分析和目标主机通信的主机数量



按照提供的服务分析主机的功能



按照应用的类型生成分类图

利用统计特征的流量识别技术

- 使用NetMate工具根据5元组把数据包划分为不同的流，并计算各种参数，如平均包长，平均间隔时间，流持续时间等。
- 为进一步提高执行速度，还可以对每条流进行采样。
- 之后将流的统计数据以及流的属性模型用于自分类的机器学习算法，无监督的贝叶斯识别技术。
- 机器学习的时间越长，分类的准确性越高，一旦达到一个标准，就可以对后续的输入数据流自动分类

利用统计特征的流量识别技术

- 特点：
- 分析已知业务的流量特征，除了取得流量组成的基本信息之外，将精力集中在统计一种业务的数据包的字节大小分布、数据包间隔分布、流字节大小分布、流间隔分布、流量间的连接特性等上，然后将从中得到的固定规律应用到未知的网络流量上。
- 不需要获取用户数据包的有效载荷，不会涉及到用户隐私问题。
- 有些特征对网络动态变化极其敏感
- 识别过程比较复杂，计算量非常大
- 不能精确的定义出每个业务的名称。

数据挖掘方法在流量分析中的应用

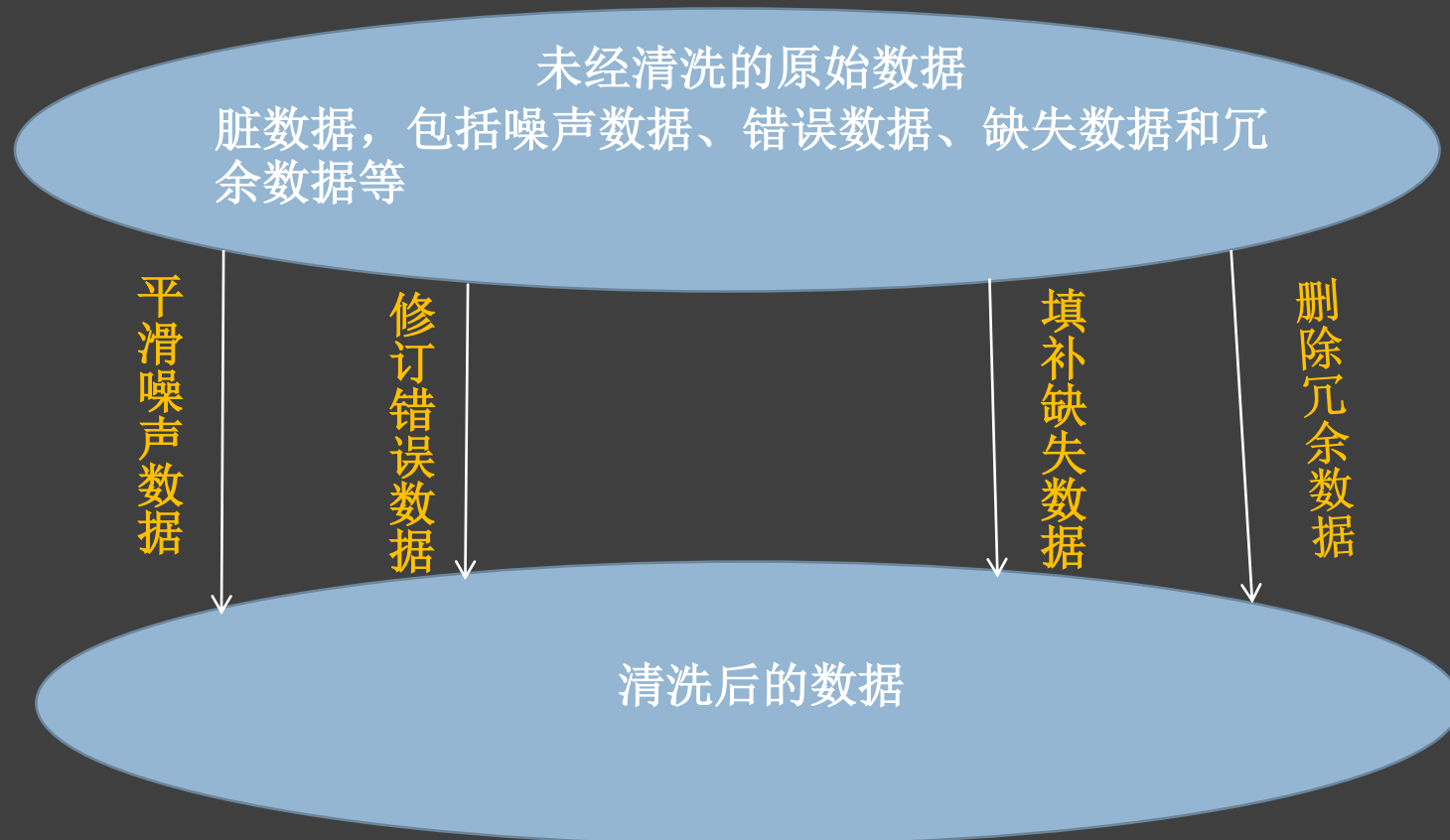
- **1.流量分析中的数据挖掘**
 - **2.数据预处理**
 - **3.数据挖掘技术在流量分析中的应用**
 - **4.其他的流量分析方法**
- 现在已经有多种数据挖掘技术应用于网络流量分析，使用数据挖掘技术可以在流量中找到隐含的、有用的流量特征，然后进行业务流量类型识别，分析网络流量的组成成分及相应的网络行为，发现网络安全威胁，了解网络运行情况，方便网络管理。

数据预处理

- 1.数据清洗
 - 2.数据变换
 - 3.数据规约
- **数据预处理技术**将包含脏数据记录的原始数据集转换成适于分析和挖掘的目标数据集。

数据清洗

- 目的：主要是为了保证数据的一致性，提高数据质量。



数据清洗

- 数据清洗的基本操作包括重复记录清除、异常记录修订、空缺值处理（如补入均值或固定值）等，一般可采用自动方法或人工方法进行。
- 清洗操作的对象可以是网络数据包或网络流，可以采用Libpcap设计专用的数据集清洗程序。
- 清洗检查的内容主要考虑：产生数据包的时间戳是否严格单调递增且在合理的窗口范围内？IP头校验和是否存在错误？包头长度是否位于合理区间？包到达间隔时间是否位于合理区间？流的单向数据包是否完备等。

数据变换

- 目的：为了让数据映射成更便于操作的形式。
-
-
- 常用的数据变换方法
 - 数值型数据的规范化、
 - 层次型概念数据的泛化、
 - 连续数值型数据的离散化，
 - 数值型数据的分桶
 -
-
-

规范化

- 将特征值按比例缩放到特定区间内，目的是方便数据处理及加快程序收敛，常用的区间为 $[-1.0, 1.0]$ 或 $[0.0, 1.0]$ 。
- 最常用的两类方法是最小-最大规范化方法和Z-score 规范化方法。

规范化

- 最小-最大规范化方法又称为离差规范化
- 通过线性变换将原始数据映射到[0.0, 1.0]这个区间中

设特征的最大值和最小值分别为 x_{max} 和 x_{min} ，映射函数定义为 $x' =$

$$\frac{x - x_{min}}{x_{max} - x_{min}}$$

- 保留了原始数据之间的序关系，但需要事先知道特征的最大取值和最小取值，一旦新数据未落在 $[x_{min}, x_{max}]$ 区间内，则会产生越界错误

规范化

- Z-score规范化方法又称为标准差规范化
- 利用特征值的均值和标准差，将原始数据映射到 $[-1.0, 1.0]$ 区间上。
设特征值的均值为 μ ，标准差为 σ ，采用的映射函数为 $x' = \frac{x - \mu}{\sigma}$
- 经过Z-score规范化后的数据，其均值为0，标准差为1，符合标准正态分布，且不必预先确定特征的最大值和最小值。

离散化

- 目的：简化数据结构，一定程度上减少数据规模，适用于特定分析和学习算法，加快处理程序的运行。
- 常用算法包括等宽算法、等频算法和聚类算法，其他方法还包括卡方分裂法、信息增益分裂法等。

离散化

- 等宽方法
- 连续取值区间等分为 k 个子区间，将第 i 个子区间中的原始数据值映射到整数 i 。若连续特征值的最大值和最小值分别为 X_{max} 和 X_{min} ，则每个子区间的宽为 $\frac{X_{max}-X_{min}}{k}$
- 不太适用于取值偏斜严重的情况

离散化

- 等频方法
- 将特征值按相同的频度划分为 k 个子区间，使落在每个子区间内的实例数一致，即若实例总数为 N ，则划分子区间的方法是使每个子区间内刚好包含 N/k 个实例。

离散化

- 聚类算法
- 通常采用 k -均值聚类，首先从训练数据集中挑选 k 个实例作为初始子区间的类心，其次对其他实例，逐个计算它们与 k 个类心之间的距离，将其归入距离最近的那个类心对应的子区间，再次重新计算每子区间的新类心，重复上述步骤直至收敛（如均方差满足设定的阈值）。最后形成的 k 个聚类的类心即作为离散数值点，将属于某个聚类的所有样本均映射到此聚类的类心对应的离散数值点上。

数据归约

- 通过数据立方体、属性选择、维归约、数据压缩、数值归约、离散化和概念分层等方法，从原始数据集中获得一个精简数据集，并使精简数据集尽可能保持原始数据集的有关特性，从而减少要处理的数据量。
- 常用的方法包括聚类、抽样或直方图。

数据归约

- 概念分层和离散化是用高层概念或区间值替换原始数据。
- 概念分层是一种数据泛化方法，通过高层概念代替低层概念以减少数据值个数。
- 离散化一般可通过分箱(Binning)法（又称为分桶法）将连续值离散化（如离散化为对应分箱的均值）。
- 分箱可以用均匀取值法或非均匀取值法。
- 非均匀性取值方法可用固定间隔法或指数间隔法，用指数间隔法对包长和包到达间隔时间的划分间隔经常设计为 $(2^i, 2^{i+1})$

数据挖掘技术在流量分析中的应用

1.关联规则

- 定义1：设 $I = \{i_1, i_2, \dots, i_m\}$ 是全体数据项的集合。数据项集（简称项集 $Itemlist$ ）是由数据项构成的非空集合。项集包含的元素个数称为项集的长度，长度为 k 的项集称为 k 阶项集($k - itemlist$)。事务数据库 $D = \{t_1, t_2, \dots, t_n\}$ 是由一系列具有惟一标识 TID 的事务组成，每个事务 $t_i (i = 1, 2, \dots, n)$ ，都对应 I 上的一个子集。

关联规则

- 定义2：关联规则是描述数据库中数据项之间存在的潜在关系的规则，形式为 $X \Rightarrow Y$ ，其中 $X \subseteq I, Y \subseteq I$ ，且 $X \cap Y = \emptyset$ ， X 称为规则头(antecedent)， Y 称为规则尾(consequent)
- 定义3：项集 X 在事物集合 D 中的支持数是 D 中包含 X 的事务数，记作 $\text{support}(X)$ 。项集 X 在事物集合 D 中支持度就是 X 在 D 中出现的频率，用符号 $P(X)$ 表示， $P(X) = \text{support}(X) / |D|$ ，其中 $|D|$ 是总事务数

关联规则

- 定义4：规则 $X \Rightarrow Y$ 在交易数据库 D 中的支持度是交易集中包含 X 和 Y 的交易数与所有交易数之比，记为 $P(X \Rightarrow Y)$ ，即

$$P(X \Rightarrow Y) = \frac{|\{T : X \cup Y \subseteq T, T \in D\}|}{|D|}$$

- 规则 $X \Rightarrow Y$ 在交易集中的置信度(confidence)是指包含 X 和 Y 的交易数和包含 X 的交易数之比，记为 $\text{conf}(X \Rightarrow Y)$ ，即

$$\text{conf}(X \Rightarrow Y) = \frac{|\{T : X \cup Y \subseteq T, T \in D\}|}{|\{T : X \subseteq T, T \in D\}|}$$

关联规则

- 定义5：事先给定一个最小支持度(minsup)，如果项集的支持度不小于最小支持度，则称 X 为频繁项集或大项集。在频繁项集中挑选出所有不被其他元素包含的频繁项目集称为最大频繁项集或最大项集。
- 给定一个交易集 D ,挖掘关联规则问题就是产生支持度和置信度分别大于用户给定的最小支持度(minsup)和最小置信度(minconf)的关联规则，称为强规则。

关联规则

- 关联规则挖掘的任务就是要挖掘出数据库 D 中所有的强规则。强规则 $X \Rightarrow Y$ 对应的项目集 $(X \sqcup Y)$ 必定是频繁项集，频繁项集 $(X \sqcup Y)$ 导出的关联规则 $X \Rightarrow Y$ 的置信度可以以频繁项集 X 和 $(X \sqcup Y)$ 的支持度计算。因此，可以把关联规则挖掘划分为以下两个子问题：
 - (1) 根据最小支持度找出数据集中的所有频繁项集。
 - (2) 根据频繁项目集和最小置信度产生关联规则。

关联规则

- 使用Apriori算法搜索关联规则。
- 采用逐层搜索迭代的方法，通过 k 项集生成 $k+1$ 项集。
- 算法主要包括两个步骤：
- **连接步**：连接 $k-1$ 频繁项集生成项候选集，可以连接的条件是两个 $k-1$ 项的前 $k-2$ 项相等并且第一个 $k-1$ 项集的第 $k-1$ 项比第二个 $k-1$ 项集的第 $k-1$ 项小。
- **剪枝步**：扫描交易数据库，累加 k 项候选集在交易数据库中出现的次数。对于一条交易记录和一个候选项集，若交易记录包含该候选项集，则该候选项集出现的次数就加1。最后根据给定的最小支持度阈值生成项频繁集。

关联规则

- 算法有如下特点：
- (1) Apriori算法需多次扫描数据库，所以其改进的一个方向是减少数据库扫描的次数；
- (2) Apriori算法产生大量的候选项目集，这些频繁项目集的存储和计数的开销很大。所以，其改进的另一个方向是减少候选项目集的个数。
- (3) Apriori算法主要操作是支持度计数，可采用一些技巧来改进支持度计数。

数据挖掘技术在流量分析中的应用

- 2 . 聚类
- (1) k-means , K均值
- K-means算法是很典型的基于距离的聚类算法，采用距离作为相似性的评价指标，即认为两个对象的距离越近，其相似度就越大。该算法认为簇是由距离靠近的对象组成的，因此把获得紧凑且独立的簇作为最终目标。

聚类

- 目标问题训练样本数据集 D 有 n 个样本 $\{x_1, x_2, \dots, x_n\}$ ，每个样本由 m 个特征 $\{A_1, A_2, \dots, A_m\}$ 决定，K-Means方法将 n 个样本按相似程度划分成 K 个子集 $\{D_1, D_2, \dots, D_k\}$ ， μ_i 为 D_i 的均值，使得

$$\operatorname{argmin}_D \sum_{i=1}^k \sum_{x_j \in D_i} \|x_j - \mu_i\|^2$$

聚类

- K-Means 算法的步骤如下:
- 1) 从训练数据集的 n 个样本中抽样选择 K 个样本作为 K 个聚类的类心 ;
- 2) 对训练数据集中的剩余样本 , 逐一计算它与上述 K 个类心的相似性 , 将其划分到与之最相似的那个类心所代表的聚类 ;
- 3) 对 K 个聚类 , 重新计算其类心 ;
- 4) 重复步骤1) -步骤3) , 直至算法收敛 (如均方差满足设定的阈值) 。

聚类

- 聚类算法的性能
- 聚类特征数用于度量聚类样本的特征数目，维度越少则算法训练和运行时间越短、性能越高。
- 聚类数目用于度量算法最终生成的聚类数目，数目越接近于样本类别数，则越易于理解，越易于避免可能的过适应。
- 聚类凝聚度包括类内凝聚度和类间分离度，类内凝聚度度量同一个聚类内部各样本的相似程度，越凝聚越好；类间分离度度量分属不同聚类的样本的相异程度，越分离越好。

聚类

- (2) K-medoids , k-中心点
- K-medoids聚类算法的基本策略是：首先通过任意为每个聚类找到一个代表对象，确定 n 个数据对象的 k 个聚类。其他对象则根据它们与这些聚类代表的距离分别将它们归属到各相应聚类中。而如果替换一个聚类代表能够改善所获聚类质量的话，那么就可以用一个新对象替换老聚类对象。

聚类

- K-medoids聚类算法具体步骤：
- 输入：聚类个数 k ，以及包含 n 个数据对象的数据库。
- 输出：满足基于各聚类中心对象的方差最小标准的 k 个聚类。
- 1) 从 n 个数据对象任意选择 k 个对象作为初始聚类代表。
- 2) 循环3) 到5) 直到每个聚类不再发生变化为止；
- 3) 依据每个聚类的中心代表对象，以及各对象与这些中心对象间距离，并根据最小距离重新对相应对象进行划分。
- 4) 任意选择一个非中心对象 o_{random} ，计算其与中心对象 o_j 交换的整个成本 S
- 5) 若为 S 负值则交换，以构成新聚类的 k 个中心对象。

聚类

- (3) DBSCAN(Density-Based Spatial Clustering of Applications with Noise) , 基于密度的聚类
- 把目标的密度作为考虑聚类的因素。
- 基于密度聚类算法的代表。它将具有足够高密度的区域划分成簇 , 并可以在带有 “噪声” 的空间数据库中发现任意形状的聚类。它定义簇为密度相连的点的最大集合。

聚类

- 给定对象半径 e 内的区域称为该对象的 e 邻域。如果一个对象的 e 邻域至少包含最小数目 $MinPts$ 个对象，则称该对象为核心对象。
- DBSCAN通过检查数据库中每个点的 e 邻域来寻找聚类。如果一个点的 e 邻域包含多于 $MinPts$ 个点



创建一个以 P 作为核心对象的新簇



DBSCAN反复地寻找从这些核心对象直接密度可达的对象并加入该簇，直到没有新的点可以被添加

聚类

- (4) SNN(Shared Nearest Neighbor) , 共享最近邻算法
- SNN是一种基于共享最近邻的聚类算法，它通过使用数据点间共享最近邻的个数作为相似度来处理变密度簇的问题，从而可以在含有噪音并且高维的数据集中发现大小不同、形状不同、密度不同的空间聚类。
- 当处理大规模和高维的数据集时算法变得代价昂贵

聚类

- (5) CURE(Clustering Using Representatives)
- CURE算法的核心步骤：
 - 1) 从源数据对象中抽取一个随机样本 S
 - 2) 将样本 S 分割为一组划分
 - 3) 对每个划分局部地聚类
 - 4) 通过随机取样剔除孤立点,如果一个簇增长的太慢,就去掉它
 - 5) 对局部的簇进行聚类。落在每个新形成的簇中的代表点根据用户定义的一个收缩因子收缩或向簇中心移动。这些点代表捕捉到了簇的形状
 - 6) 用相应的簇标签来标记数据

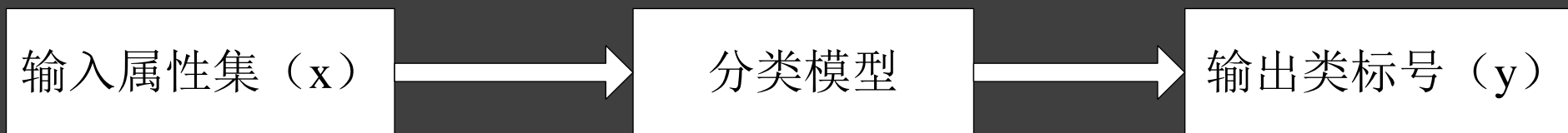
数据挖掘技术在流量分析中的应用

- 3 . 分类

- 分类任务就是通过学习得到一个目标函数 f ，把每个属性集 x 映射到一个预先定义的类标号 y
- 目标函数也称为分类模型(classification Model)。分类模型可以用于以下目的。
- **描述性建模**：分类模型可以作为解释性的工具，用于区分不同类中的对象。
- **预测性建模**：分类模型还可以用于预测未知记录的类标号。

分类

- 分类器的任务：根据输入属性集 x 确定类标号 y



分类

- 分类的技术一般分成两步：
- 1．建立模型，由训练数据建立分类模型；
- 2．用模型进行分类，把模型应用于测试样例。
- 重点介绍决策树和贝叶斯分类

决策树

- **决策树**是一种由结点和有向边组成的层次结构。树中包含3种类型的结点：
- 根节点：没有入边，但有零条或多条出边。
- 内部结点：恰有一条入边和两条或多条出边。
- 叶节点：恰有一条入边，但没有出边。
- 在决策树中，每个叶节点都赋予一个类标号。非终结点包含属性测试条件，用以分开具有不同特征的记录。

决策树

①Gini指标：

衡量决策树划分纯度的一种指标，越大，说明不纯度越高，则包含的信息越大，越低，则包含信息越少，越不可取

$$Gini(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2$$

②Entropy (熵)

- 不纯度的最佳评估方法是平均信息量，也就是信息熵(Entropy)

$$Entropy(t) = - \sum_{i=0}^{c-1} p(i|t) \log_2^{(p(i|t))}$$

决策树

- ③信息增益
- 为了确定测试条件的效果，需要比较父结点（划分前）的不纯程度和子女结点（划分后）的不纯程度，差越大，测试条件的效果就越好。

$$\Delta = I(\text{parent}) - \sum_{j=1}^k \frac{N(\mathbf{v}_j)}{N} I(\mathbf{v}_j)$$

- 当选择熵(entropy) 作为不纯性度量时，熵的差就是信息增益 (information gain)

决策树

- ④信息增益率

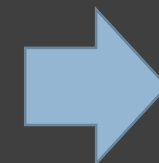
$$\textit{Gain ratio} = \frac{\Delta_{\textit{info}}}{\textit{Split info}}$$

- 其中，划分信息 $\textit{Split Info} = -\sum_{i=1}^k P(v_i) \log_2 P(v_i)$

决策树

- Gini指标和信息熵的计算实例

结点 N_1	计数	
C1	0	$P(C1) = 0/10 = 0$ $P(C2) = 10/10 = 1$
C2	10	$Gini = 1 - P(C1)^2 - P(C2)^2 = 0$ $Entropy(t) = -(0/10)\log_2^{0/10} - (10/10)\log_2^{10/10} = 0$
结点 N_2	计数	
C1	3	$P(C1) = 3/10 = 0.3$ $P(C2) = 7/10 = 0.7$
C2	7	$Gini = 1 - P(C1)^2 - P(C2)^2 = 0.42$ $Entropy(t) = -(3/10)\log_2^{3/10} - (7/10)\log_2^{7/10} = 0.88132$
结点 N_3	计数	
C1	5	$P(C1) = 5/10 = 0.5$ $P(C2) = 5/10 = 0.5$
C2	5	$Gini = 1 - P(C1)^2 - P(C2)^2 = 0.5$ $Entropy(t) = -(5/10)\log_2^{5/10} - (5/10)\log_2^{5/10} = 1$



均衡分布的结点Gini指标最高，有最高的不纯度

决策树

- 决策树算法的学习过程=树构造+树剪枝
- 树构造：决策树采用自顶向下的递归方式进行构造，从根结点开始，在每个结点上按照给定标准选择测试属性，然后按照相应属性的所有可能取值向下建立分支，划分训练样本，直到每个结点上的所有样本都划分到同一类或某一个结点上的样本数量低于给定值为止。
- 树剪枝：许多分支可能是训练数据中的噪声或孤立点。剪枝的过程就是去掉这种分支，剪枝主要有先剪枝和后剪枝或两者相结合。

ID3

- 典型的决策树算法有ID3，C4.5和CART
- ID3算法：核心是用贪心算法来根据“信息熵”分类

信息熵的有效减少量 ➡ 信息增益

- 缺陷在于它偏向于选择具有更多取值的属性作为节点分裂属性，而实际上属性值较多的属性不一定是最优的分类属性。
- 存在分类偏向于取值数量，只能处理离散数据等问题

C4.5算法

- ID3的一个改进
- 通过离散化连续值属性的取值空间，改进了属性只能取离散值的缺点
- 将属性选择标准由信息增益调整为信息增益率，改进了信息增益选择属性时偏向分类取值更多的属性这一不足
- 在模型建立过程中较少依赖样本的分布。

C4.5算法

- 设目标问题训练样本数据集 D 有 n 个样本 $\{x_1, x_2, \dots, x_n\}$ ，每个样本由 m 个属性 $\{A_1, A_2, \dots, A_m\}$ 决定，其中 A_m 为分类类别属性， A_m 有 k 个不同取值。根据 A_m 不同取值，将 D 划分为 k 个子集
- 样本数据集 D 对 k 个分类的信息期望 $I(D)$ 的定义

$$I(D) = -\sum_{i=1}^k p_i \log_2 p_i$$

C4.5算法

- 设属性 A_u 存在 l 个不同取值，利用这 l 个取值，可以将 k 个子集进一步划分为 $l \times k$ 子集 $D(p, q)$ ，其 $1 \leq p \leq k, 1 \leq q \leq l, D_q = \{x_j | x_j \in D \& \& A_u(x_j) = a_q\}$

$$D_{pq} = \{x_j | x_j \in D_p \& \& A_u(x_j) = a_q\}$$

则属性 A_u 对每个取值 a_q 的信息期望的定义为：

$$I(D | A_u = a_q) = -\sum_{j=1}^k p_{pq} \log_2 p_{pq}$$

- A_u 信息熵(Information Entropy) 和信息增益分别用如下公式计算：

$$E(D | A_u) = -\sum_{j=1}^m p_j \cdot I(D | A_u = a_j)$$

$$G(D | A_u) = E(D | A_u) - I(D)$$

C4.5算法

- C4.5算法引入信息增益率，通过公式计算：

$$R_G(D | A_i) = G(D | A_i) / I(D | A_i)$$

- 每次将所有属性 A_i 的信息增益率 $R_G(D|A)$ 进行比较，选择其中最大值作为根结点，下次再对剩余属性运用同样的选择算法建立根结点的分支结点，由此生成完整的决策树。

CART

构造树的阶段：

- 将Gini指数作为选择测试属性的标准，将训练数据划分为不相连的子集。
- 从包括所有训练数据的根节点开始，为求最能减少误差指标的分叉，做一次穷尽搜索。一旦确定最佳分叉，数据集相应地划分成不相连的子集，这些子集用源于根节点的子节点表示。
- 再对子节点实施同样的划分。当与一个节点有关的误差值小于某个值时，或当进一步划分树，误差的减少不超过某个阈值时，这个递归过程终止。

CART

- 递归树的生成：
- 对于一个递归树，节点误差指标常取为拟合节点数据集的局部模型的平方误差或残差：

$$E(t) = \min_{\theta} \sum_{i=1}^{N(t)} (y_i - d_i(x_i, \theta))^2$$

- 把节点 t 分解成 t_r 和 t_l 的任意分叉 s ，误差测度的变化可表示为：

$$\Delta E(s, t) = E(t) - E(t_l) - E(t_r)$$

- 最好的分叉 s^* 为误差测度降低最多的分叉 $\Delta E(s^*, t) = \max \Delta E(s, t)$

CART

- 树剪枝阶段：
- 基于最小代价复杂性或最弱子树收缩原理是CART算法运用的最有效的方法之一
- 第一步是产生一棵充分张开的树 T_{max} ，这棵树拟合训练数据相当好，但规模较大，因此要寻找其中的最弱子树进行剪枝。考虑训练误差测度和终节点数目，即考虑树的复杂性指标，就可以找到最弱子树。

K-最近邻分类

- 核心思想是： n 维特征空间 R^n 中，样本 x 与特征空间中 k 个最相似的已分类样本（又称邻居）中的大多数属于同一类
- 计算两个样本间的距离

K-最近邻分类

- 欧氏空间中常用距离定义如下：
- 欧几里得距离(Euclidean distance)，又称欧氏距离， x 和 y 的欧氏距离为：

$$d(x, y) = \sqrt{\sum_{i=1}^n (x^{(i)} - y^{(i)})^2}$$

- 切比雪夫距离(Chebyshev distance)， x 和 y 的切比雪夫距离为：

$$d(x, y) = \max_i (|x^{(i)} - y^{(i)}|)$$

- 曼哈顿距离(Manhattan distance)，又称绝对值距离， x 和 y 的曼哈顿距离为：

$$d(x, y) = \sum_{i=1}^n |x^{(i)} - y^{(i)}|$$

- 闵可夫斯基距离(Minkowski distance)，又称闵氏距离， x 和 y 的闵氏距离为：

$$d(x, y) = (\sum_{i=1}^n |x^{(i)} - y^{(i)}|^p)^{1/p}$$

K-最近邻分类

- 在向量空间，经常使用余弦相似度来度量两个向量之间的相似程度，余弦相似度已普遍用于文本相似性比较。

$$\cos(x, y) = \frac{x \cdot y}{\|x\| \|y\|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n (x_i)^2} \times \sqrt{\sum_{i=1}^n (y_i)^2}}$$

- K-NN算法的弱点是严重依赖样本库，受 k 的取值大小影响

贝叶斯分类

- 贝叶斯定理：
- 设随机事件A发生的概率为 $P(A)$ 。我们又称 $P(A)$ 为A的先验概率

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

- 贝叶斯公式是在已知3个概率的情况下推出第四个概率。
- 对于 N 个独立事件,每个事件发生的概率为 $P(A_i)$,贝叶斯定理用以下公式描述

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_{j=1}^n P(B|A_j)P(A_j)}$$

贝叶斯分类

- 在网络流量分类问题中,设输入变量 X 为 n 维特征向量 $(X^{(1)}, X^{(2)}, \dots, X^{(n)})$,输出变量 Y 为一维的类标记 y_i ,若存在已标注的训练数据集,则对每个类 C_j ,可以利用训练数据集计算得到 $P(Y = C_j)$ 和条件概率分布 $P(X = x_i | Y = C_j)$,从而可以利用贝叶斯定理得到联合概率分布 $P(X, Y)$

朴素贝叶斯

- 利用贝叶斯定理计算条件概率 $P(X = x_i | Y = C_j)$ 时，对输入变量的特征做出了独立性假设有

$$P(X = x_i | Y = C_j) = \prod_{d=1}^n P(X^{(d)} = X_i^{(d)} | Y = C_j)$$

- 朴素贝叶斯方法在利用训练数据集的实例学习时，对于输入 x ，要让分类到后验概率 $P(Y = C_j | X = x_i)$ 的值最大化的类，此时朴素贝叶斯分类器可以表示为公式

$$f_{NB} = \arg \max_{c_j} P(Y = C_j) \prod_{d=1}^n P(X^{(d)} = X_i^{(d)} | Y = C_j)$$

贝叶斯信念网络

- 针对关联性和不确定性问题
- 一个有向无环图，图中结点表示随机变量，结点之间的连线（边）表示随机变量间的条件依赖，边被赋予一个权值，表示随机变量之间的依赖强度。由于权值表示的是不确定性，因此通常是用概率值度量，这个概率又称为信念(Belief)，贝叶斯网络因此又称为贝叶斯信念网络

贝叶斯信念网络

- 设 $G = (V, E)$ 是一个有向无环图, 对 $v \in V$, 设 $pa(v) \in V$ 是 v 的父结点, $X \subseteq V$, 如果公式成立, 则 X 为贝叶斯网络

$$P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = \prod_{v=1}^n P(X_v = x_v \mid X_{pa(v)} = x_{pa(v)})$$

其他的流量分析方法

- 基于统计的方法
- 基于信息论的方法
- 基于EM的方法
- 数据挖掘技术进行流量分析



难点高维数据如何降维，如何评价特征选择结果...

本章小结

- 本章介绍流量分析的概念，流量分析的目的，方法及现状，介绍了流量采集的方法，目前主流的流量分析模型及方法。传统的流量分析方法包括基于端口的方法、基于特征码的方法，基于传输层的方法，统计特征的流量识别方法等，这些方法都有各自不同的缺点
- 本章主要介绍了使用数据挖掘来进行流量分析的相关方法，包括关联规则，聚类和分类。关联规则方法主要介绍了Apriori算法；聚类算法主要介绍了K-均值、K-中心点、DBSCAN、SNN、CURE等算法；分类算法主要介绍了决策树、KNN、贝叶斯分类等

Thanks!