

网络数据层  
两大关键技术

路由查找  
数据包分类

(地址查找) / (名称查找)  
路由查找: 根据数据包携带的信息在路由表中  
按照最长前缀匹配原则 (LPM), 找到转  
发出口。

数据包分类: 根据数据包携带的信息 (源, 目的地址,  
源, 目的端口号和协议) 在规则集中  
按照LPM原则, 找到数据包待执行的任务。

现状

硬件上加速: PPA, 三态内容寻址存储器 芯片实现  
→ 成本高

软件: 无成本, 灵活性好。



路由查找  
数据结构不  
同

字典树  
决策搜索树  
哈希表

包分类

维度降解  
空间划分

# 报文分类算法.

报文  $\longrightarrow$  头部相关域数值  $\xrightarrow{\text{查询匹配}}$  规则集  $\rightarrow$  处理

定义:

① 分类器  $C = \{R_1, R_2, \dots, R_N\}$

②  $N$  条规则  $R_i$

一条规则  $R_i$  } 范围表达式  $R[i]$ ,  $1 \leq i \leq d$ , 每条规则  $d$  个域  
优先级。报文匹配的条规则, 确定最佳匹配规则。  
规则决策。处理动作。

收到 报文头部  $P = \{p_1, p_2, \dots, p_d\}$  ( $d$  个域)

$p_i$  为第  $i$  个域的数值, 在  $C$  中查找与之相匹配且值优先级最高的规则  $R_m$ 。

一个分类器  $C$  包含  $N$  条规则, 一条规则包括  $d$  个域  
一个报文头部也包含  $d$  个域

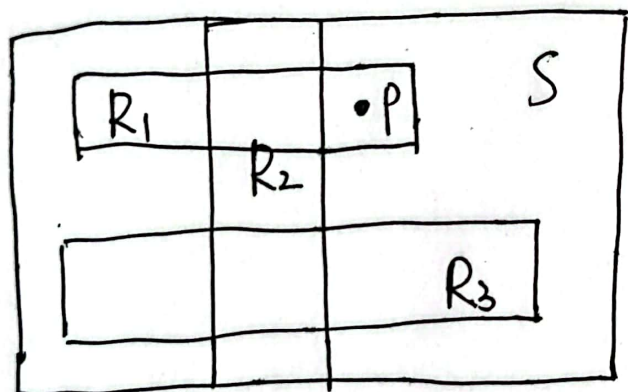
- 当且仅当报文头部  $P$  的每一个域  $p_i$  和规则  $R_m$  中相应域  $R_m[i]$  匹配时, 称  $P$  和  $R_m$  匹配

# 数学模型

报文分类  $\xrightarrow{\text{抽象}}$  计算几何中的维空间点定位

报文头部提取到的关键字, 可视为空间中的点  $p$ .

报文分类过程  $\longrightarrow$  查询覆盖点  $p$  的优先级最高的矩形.

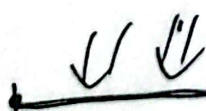


当前节点规则集  $n$  条规则, 所有规则在第  $i$  个域的投影点 构成集合,  $\longrightarrow m$  个

$m$  个点 存储在  $\xrightarrow{\text{从小到大}}$   $pt[i]$



$(m-1)$  个投影区间  $\longrightarrow sg[i]$



规则数  $sr[k]$ : 投影区间数组



# Hypersplit

输入: 搜索空间  $S$ , 域  $f$ , 一系列规则  $R$ .  
应用基于规则的分割策略.

① 投影 规则  $R \rightarrow f$  域  
产生  $m$  ( $2 \leq m \leq 2N+1$ ) 个终点

② 对  $m$  个点, 从小到大排序, 存储在  
临时数组  $pt[i]$  中 ( $1 \leq i \leq m$ )  
产生  $\downarrow$  ( $m-1$ ) 个段 (区间)  $\longrightarrow Sg[j]$   
 $1 \leq j \leq m-1$

③ 一个正交于  $f$  域的超平面 通过  $m$  个点中的一个点 将搜索空间分成 2 个子空间.

如何选 end-point 来分割?

① 选  $pt[\lfloor \frac{m}{2} \rfloor]$ ,  $\frac{m}{2}$  向下取整,  
表明, 分段平衡的分割, 沿  $f$  域  
每个子空间有相同数量的区间 (segment)

② 选择  $pt[m]$ , 使得 ~~重叠~~ 这个区间  
 $[pt[1], pt[m]]$  ~~这个区间的长度~~

的规则的数量为  $\lfloor \frac{|R|}{2} \rfloor$

③ 假设在  $R$  中有  $sr[i]$  规则和区间  $sg[i]$   
重叠, 选择  $pt[m]$ ,  $m$  是满足以下式子的  
最小值:

$$\sum_{j=1}^m sr[i] > \frac{1}{2} \sum_{j=1}^m sr[i]$$

$sr[i]$  可以视为  $sg[i]$  的权重.

1. 域(字段), 规则, 空间 这些含义?

规则 包括 源IP地址, 目的IP.....

规则	源IP	目的IP	源port	目的port	协议
R <sub>1</sub>			80	*	Tcp
R <sub>2</sub>					

10.1.190.67/32

153.64.80.20/32

通配符  
(任意值都可以)

一个规则 如上图所示

**字段**: 源IP、目的IP 等都是 一个字段

① 上图规则 R<sub>1</sub> 中有 5 个字段

② 每个字段的值可以是一个定值, 也可以是一个范围.

**空间**: 通过决策树, 规则 → 空间

规则中每一个 ~~字段~~ 字段 (域) 对应着一个维度.

每个规则对应着一个空间, R<sub>1</sub> 有 5 个字段, 对应 5 维空间, 各个维度上的大小 对应 空间大小.



IP 为: 10.1.190.67/32 表示一个值.

IP 为: 12.68.6.0/24 表示一个范围值.

因此, 规则 对应到空间中是 多维空间超长方体  
数据包  $\longrightarrow$  多维空间中一个点

因为数据包的源 IP、目的 IP 等都是一个值,  
在每个维度都是一个定值, 也就对应着一个点 P

报文分类  $\longrightarrow$  寻找覆盖点 P 的超长方体.

以 2 维空间 举例

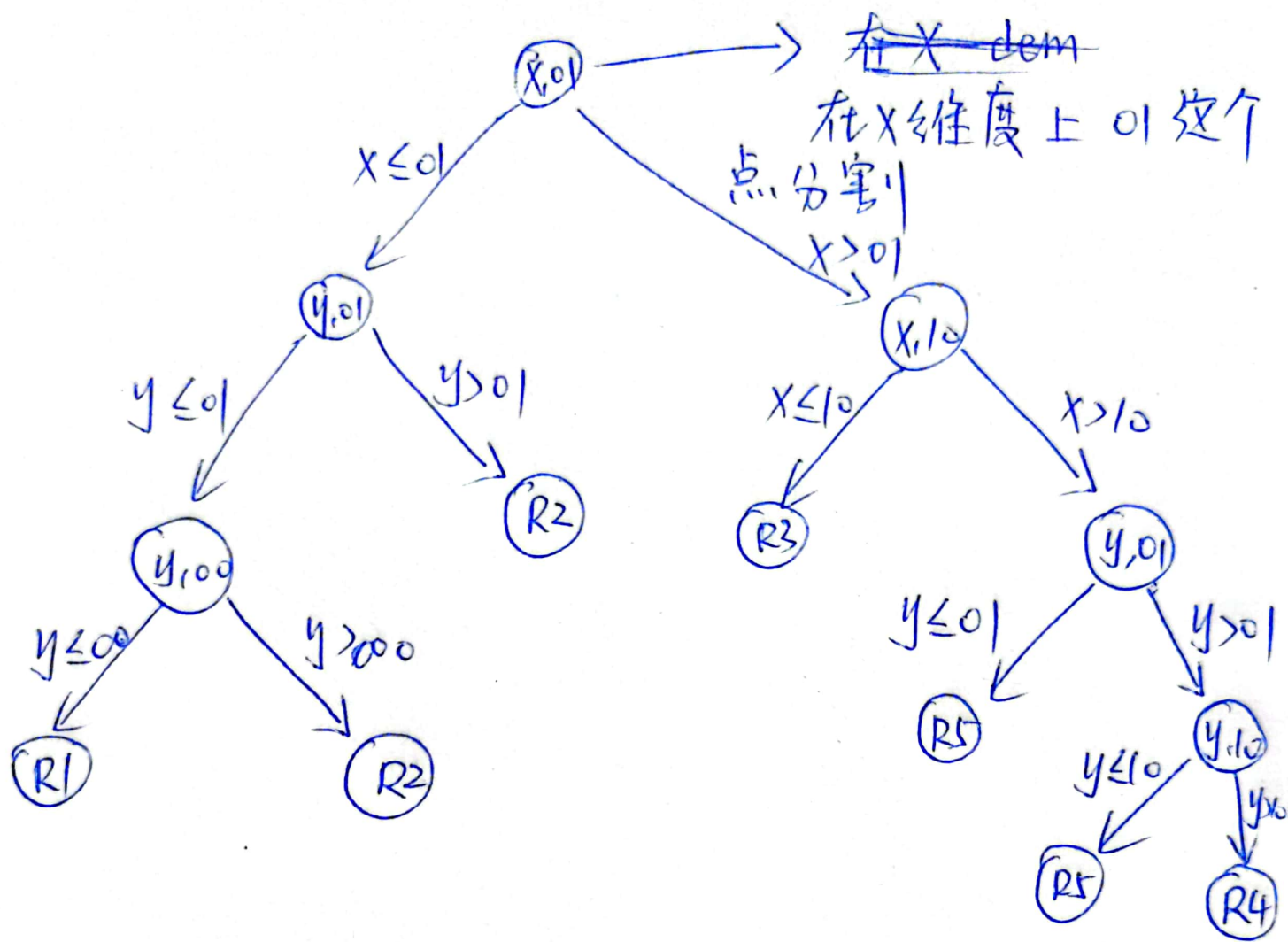
图 1: 展示规则, X, Y 就是字段, X, Y 的值  
有的是一个范围, 有的是一个点

图 4: 规则 对应到空间, 所有规则组成的  
空间就是整个搜索空间 S, 对空间进行  
分割, 便于更快查找.

## 2. Hints

{ 均匀分割

{ 不断重复, 直到叶子节点数量小于一定阈值

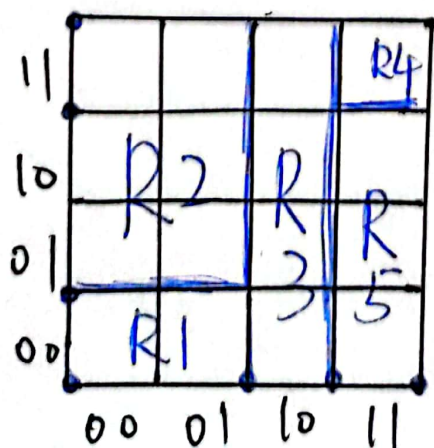


选哪个维度切割?  
选最优的进行切割.

### 3. Hyper split.

尽可能地在规则的分界处进行切割, 尽可能地减少 ~~把~~ 把同一个规则一分为二的情况.

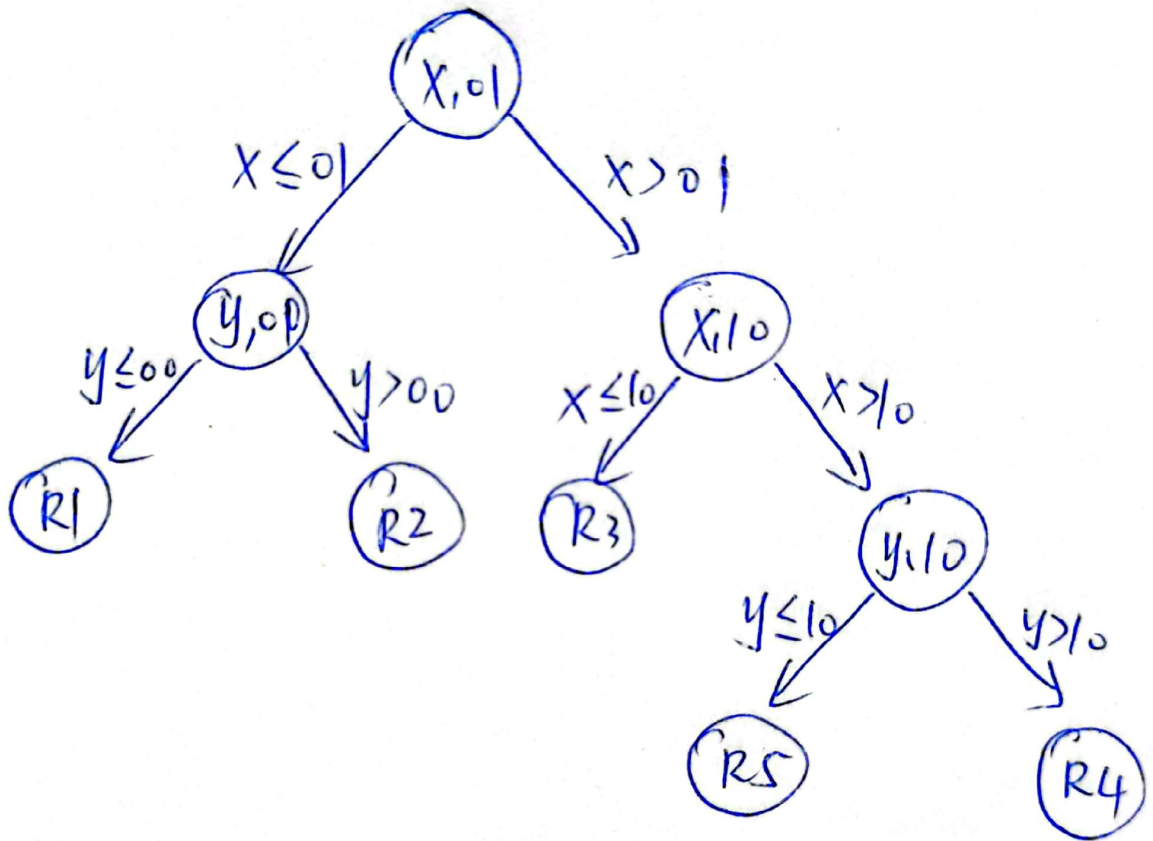
①



规则的分界向  
轴上投影



- ② 投影之后, 每次选点进行分割  
使左右段数一样  
左右规则数一样



Hypersplit 构建出的决策树比 Hicuts  
更平衡, 查找更快.

