



符号学习简介

命题规则学习（上）

(Press ? for help, n and p for next and previous slide)

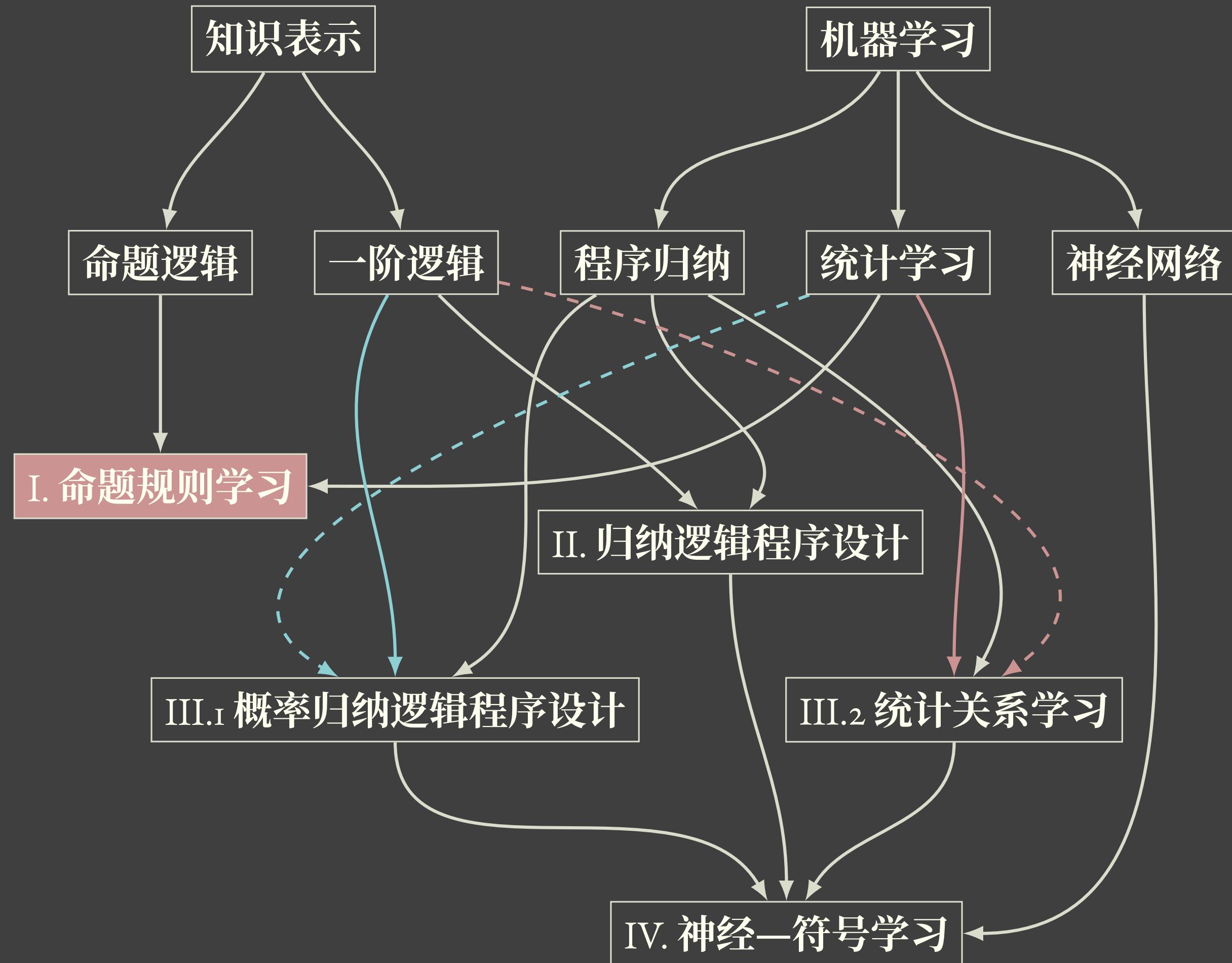
戴望州

南京大学智能科学与技术学院
2025年-秋季

<https://daiwz.net>

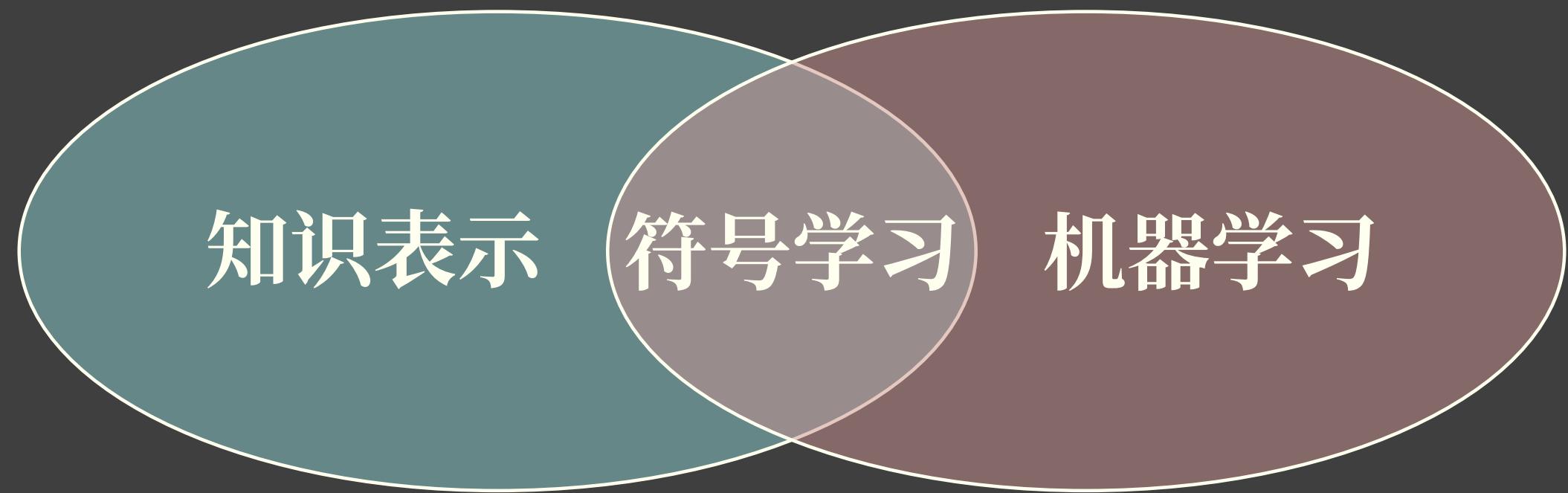


路径图





命题规则学习



- > 选择“词汇”：
 - » 背景知识 B : 原子命题、命题逻辑语言
- > 寻找“解释”：
 - » 假设模型 H : 命题逻辑规则
 - » 寻找过程: 启发式搜索



符号学习简介 · 命题规则（上）

1. 命题（逻辑）规则
2. 命题规则学习
3. 打分函数



命题逻辑语言

命题逻辑语言是用来描述原子命题与它们之间关系的一种**形式语言**。

形式语言（Formal language）与自然语言（Natural Language）不同，是一种人为设计的语言，例如程序语言、数学语言等。具体地，一个语言 L 包含：

- › **词汇表**（alphabet） Σ : 该语言所描述的对象
- › **合式公式**（well-formed formulas） $L \subseteq \Sigma^*$: 该语言中所有可能的表达式，“ $*$ ”表示Kleene闭包

通常，用 Σ 来定义或生成 S 的方法被称为**句法**（Syntax/grammar）。



命题逻辑语言

- > 词汇表：原子命题集合 $\Sigma = \{p, q, r, \dots\}$
 - » 逻辑真值 $\{\perp, \top\}$ (或者 $\{0, 1\}$ 、 $\{F, T\}$)
- > 公式：用**命题连接词**描述原子命题间关系的句子
 - » $\{\neg, \vee\}$ 或 $\{\neg, \wedge, \vee, \rightarrow\}$ ($p \rightarrow q \equiv \neg p \vee q$)
 - » $\{\neg, \wedge\}, \{\rightarrow, \neg\}, \{\neg, \wedge, \vee\}, \{\neg, \wedge, \vee, \rightarrow, \equiv\}, \{| \}$ (并非二者都...) , $\{\downarrow\}$ (既非...也非...)

命题逻辑演算的推理规则: $p, p \rightarrow q \vdash q$

命题逻辑演算公理 (Łukasiewicz) :

1. $p \rightarrow (q \rightarrow p)$
2. $(p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$
3. $(\neg p \rightarrow \neg q) \rightarrow (q \rightarrow p)$



命题逻辑语言

三段论: $p \rightarrow q, q \rightarrow r \vdash p \rightarrow r$

证明:

1. 等价于证明 $(\neg p \vee q) \wedge (\neg q \vee r) \rightarrow (\neg p \vee r)$ 是重言式 (tautology)
2. $\neg((\neg p \vee q) \wedge (\neg q \vee r)) \vee (\neg p \vee r)$
3. $(\neg(\neg p \vee q) \vee \neg(\neg q \vee r)) \vee (\neg p \vee r)$ (*de Morgan law*)
4. $(p \wedge \neg q) \vee (q \wedge \neg r) \vee (\neg p \vee r)$ (*de Morgan law*)
5. 验证真值, 显然。 ■



命题逻辑规则

然而，完整的命题逻辑系统并不实用：

- I. 复杂度高，一般的命题逻辑公式可满足性是NP-完全问题

- » 验证速度慢，学习速度只会更慢
- » 表达能力过强，常见问题用不上

2. 不够直观，我们更习惯if...then... else...形式的推理

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
I	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
II	浅白	硬挺	清脆	模糊	平坦	硬滑	否
I2	浅白	蜷缩	浊响	模糊	平坦	软粘	否
I3	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否
I4	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
I5	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
I6	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
I7	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否



命题逻辑规则

我们想要的规则是这样的：

IF 色泽 = 青绿
AND 根蒂 = 稍蜷
THEN 好瓜 = 是

或者，

IF 色泽 = 浅白
AND 纹理 != 清晰
THEN 好瓜 = 否

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
I	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
II	浅白	硬挺	清脆	模糊	平坦	硬滑	否
I2	浅白	蜷缩	浊响	模糊	平坦	软粘	否
I3	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否
I4	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
I5	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
I6	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
I7	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否



命题逻辑规则

命题逻辑规则是命题逻辑（合式）公式的一个子集，形式如下：

$$H \leftarrow B_1 \wedge B_2 \wedge \dots \wedge B_n.$$

- > $H, B_1, \dots, B_N \in \Sigma \cup \neg\Sigma$ 为原子命题或原子命题的否定，称为**逻辑文字**（ Literal ）
- > H 是规则的**后件**，叫做**规则头**（ head ）
- > $B_1 \wedge B_2 \wedge \dots \wedge B_n$ 是规则的**前件**，叫做**规则体**（ body ）
- > n 是规则的**长度**（ length ）。

它可以等价地写为 $H \vee \neg B_1 \vee \neg B_2 \vee \dots \vee \neg B_n$ ，即只含1个肯定原子的析取式。

其正式名称叫做**命题霍恩子句**（ Propositional Horn clause ）。

找到满足一组命题霍恩子句的有效赋值的问题叫做HORNSAT，它是P-完全问题，因此线性时间内可解。



实际应用中的命题规则

与专家合作的冠心病诊断规则学习[Fürnkranz et al., 2012.]

```
IF    总胆固醇 >= 6.1 mmol/L  
AND 年龄 >= 53 岁  
AND BMI < 30  
THEN 冠心病
```

```
IF    BMI >= 25  
AND 高密度脂蛋白 < 1.25 mmol/L  
AND 尿酸 < 360 mmol/L  
AND 血糖 < 7 mmol/L  
AND 血纤维蛋白原 > 3.7 g/L  
THEN 冠心病
```

专家一般对学得规则提出的建议：

1. 规则长度要小
2. 避免使用昂贵、不可靠的检查



样本覆盖 (Coverage)

规则 \mathcal{R} 在样本集 \mathcal{E} 上的覆盖情况：

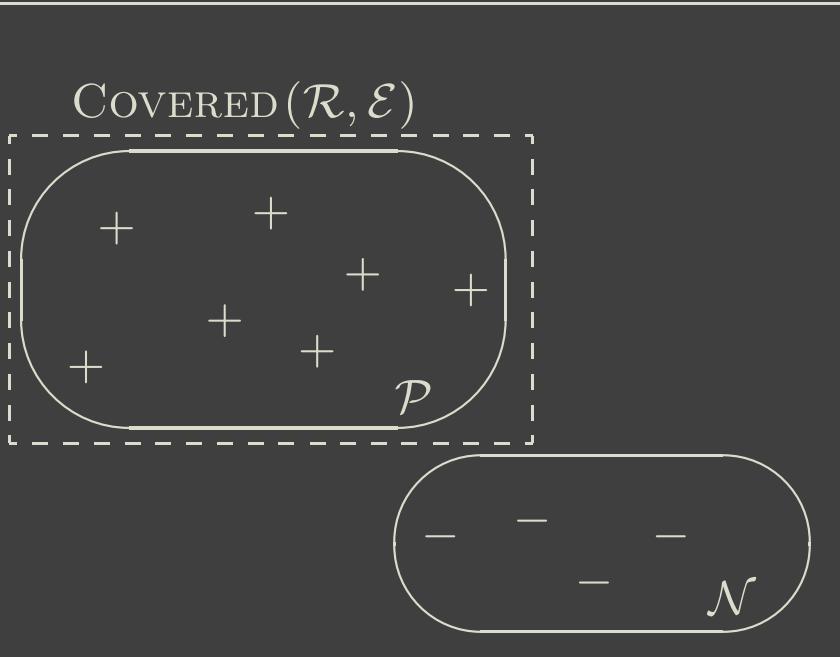
- > \mathcal{P} : 正样例
- > \mathcal{N} : 负样例
- > 完备 (complete) :

$$\forall e^+ (e^+ \in \mathcal{P} \rightarrow \text{Covered}(\mathcal{R}, e^+))$$

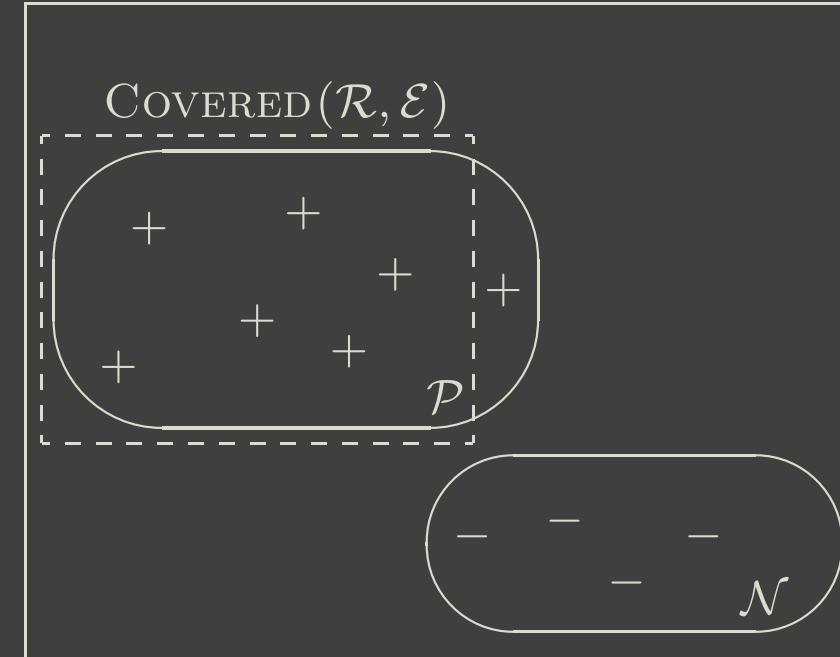
- > 一致 (consistent) :

$$\forall e^- (e^- \in \mathcal{N} \rightarrow \neg \text{Covered}(\mathcal{R}, e^-))$$

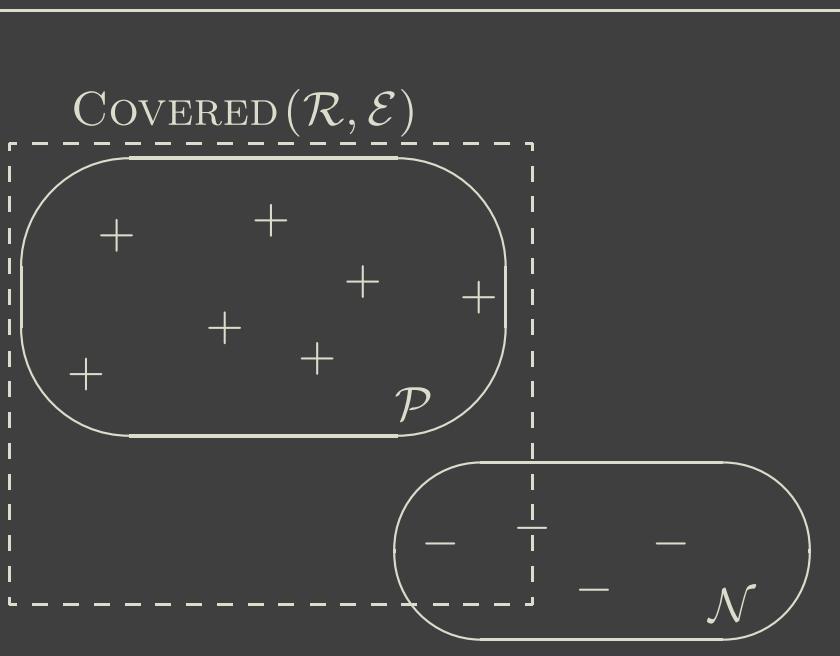
\mathcal{R} : complete, consistent



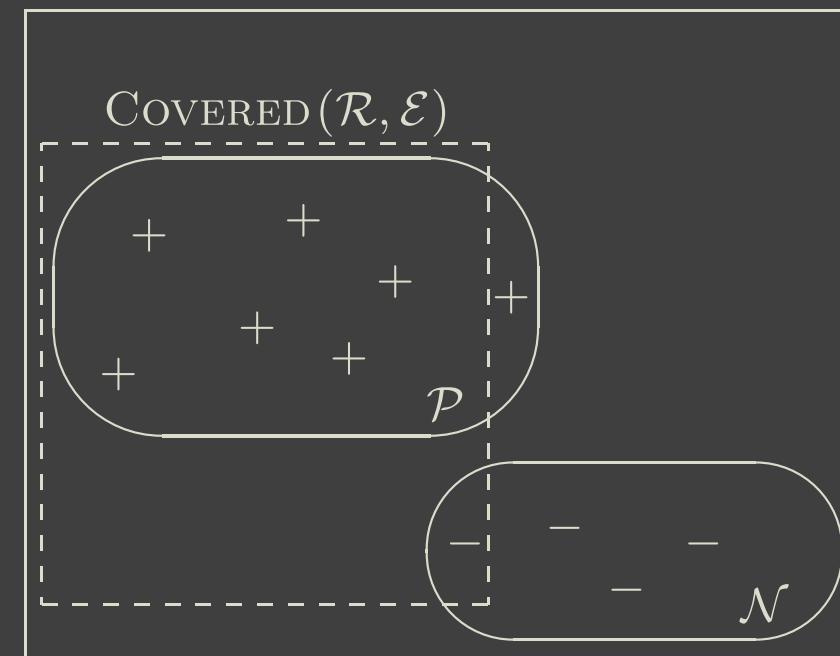
\mathcal{R} : incomplete, consistent



\mathcal{R} : complete, inconsistent



\mathcal{R} : incomplete, inconsistent





符号学习简介 · 命题规则（上）

- 1. 命题（逻辑）规则
- 2. 命题规则学习
 - » 命题特征构建
 - » 命题规则搜索
- 3. 打分函数



命题逻辑规则中的特征

命题规则的**特征**（feature）为原子命题或其否定，即取值为{0, 1}的命题逻辑公式。

常见的数据集一般是**属性-值**（attribute-value）表格，常见的种类有：

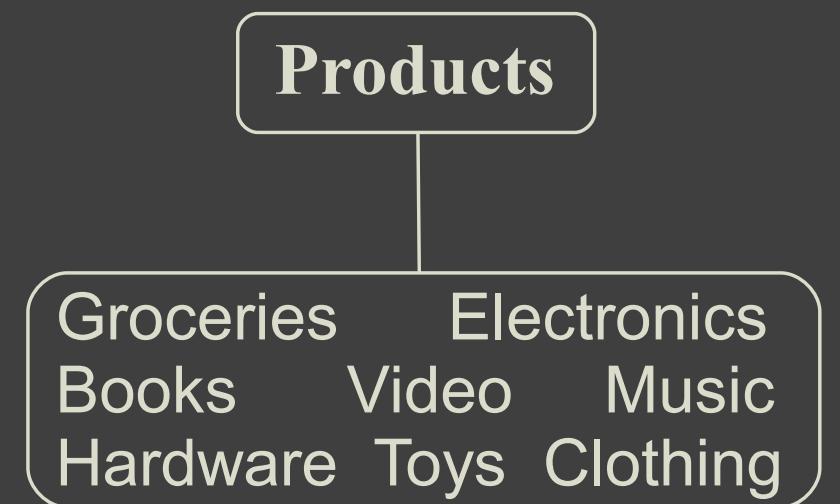
- > 离散值属性
- > 连续值属性

色泽	根蒂	敲击	纹理	脐部	触感	密度	含糖率	好瓜
青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.46	是
乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	0.666	0.091	否
青绿	硬挺	清脆	清晰	平坦	软粘	0.243	0.267	否
浅白	硬挺	清脆	模糊	平坦	硬滑	0.245	0.057	否



离散值属性

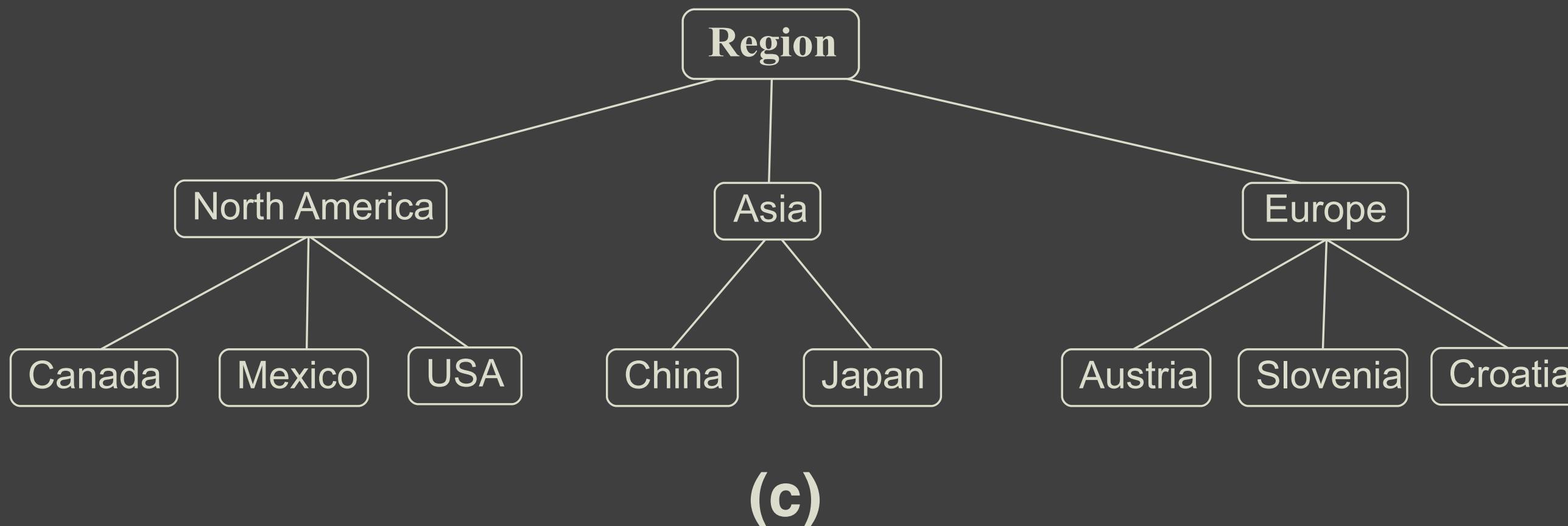
集合值属性、有序值属性、层次值属性



(a)



(b)



(c)



离散值属性特征

离散属性一般通过**选择器**（ selector ）来转化为命题特征。

例如，当属性 \mathbf{A}_i 的取指拥有 $v_{i,j}, j = 1, 2, \dots$ 时，可以构造命题：

- > $\mathbf{A}_i = v_{i,j}$
- > $\mathbf{A}_i \neq v_{i,j}$ (否命题)
- > $\mathbf{A}_i \in \{v_{i,k_1}, v_{i,k_2}, \dots\}$ (属性值析取)
 - » $\mathbf{A}_i = v_{i,k_1} \vee \mathbf{A}_i = v_{i,k_2}, \dots$
- > $\{\mathbf{A}_i, \mathbf{A}_j, \dots\} \in \{v_k, v_l, \dots\}$ (属性合取、属性值析取)
 - » $\mathbf{A}_i \in \{v_k, v_l, \dots\} \wedge \mathbf{A}_j \in \{v_k, v_l, \dots\}, \dots$
- > $\mathbf{A}_i = v_{1,j}/v_{2,k}/\dots/v_{m,l}$ (层次值属性)
 - » $\mathbf{A}_i \in \{v \mid v \in v_{m,l}\}$



连续值、有序值属性特征

连续值和有序值属性可以通过等号大、小于号来转化为命题特征，例如

- > $\mathbf{A}_i > v_{i,j}, \mathbf{A}_i < v_{i,j}$
- > $\mathbf{A}_i \leq v_{i,j}, \mathbf{A}_i \geq v_{i,j}$ (否命题)
- > $v_{i,j} < \mathbf{A}_i \leq v_{i,k}$ (区间)

连续值属性也可用函数建关系型特征 (relational feature) :

- > $f(\mathbf{A}_1, \mathbf{A}_2, \dots) < 0$
- > $\mathbf{A}_i - \mathbf{A}_j < v, \mathbf{A}_i - \mathbf{A}_j \geq v$
- > $\mathbf{A}_i + \mathbf{A}_j < v, \mathbf{A}_i + \mathbf{A}_j \geq v$

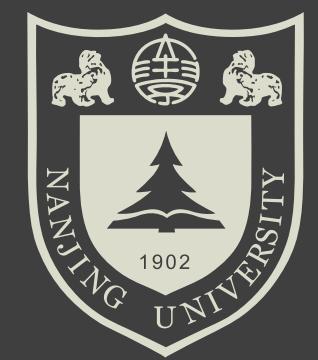


充分特征集

定义2.1 (特征可区分性) 一个特征 $f \in \mathcal{F}$ 可区分一个正负样本对 $\langle p, n \rangle$ 当且仅当 f 能够正确区分正样例 $p \in \mathcal{P}$ 和负样例 $n \in \mathcal{N}$ 。即对 p 有 $f = \text{true}$ 且 n 有 $f = \text{false}$ 。我们记作 $f \sqsupseteq \langle p, n \rangle$ 。

ID	色泽	根蒂	敲击	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
4	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
5	青绿	硬挺	清脆	清晰	平坦	软粘	否
6	浅白	硬挺	清脆	模糊	平坦	硬滑	否

- > “色泽 = 青绿”可以区分样本对 $\langle 1, 4 \rangle$, 但无法区分 $\langle 1, 5 \rangle$



充分特征集

定义2.2（充分特征集）若使用特征集 \mathcal{F} 可以构建出一个一致（不覆盖负样本）且完备（覆盖所有正样本）的命题规则模型，则 \mathcal{F} 被称为一个充分特征集。

ID	色泽	根蒂	敲击	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
4	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
5	青绿	硬挺	清脆	清晰	平坦	软粘	否
6	浅白	硬挺	清脆	模糊	平坦	硬滑	否

- > $\{\text{敲击} = \text{浊响}, \text{纹理} = \text{清晰}\}$ 不是充分特征集
- > $\{\text{根蒂} = \text{蜷缩}\}$ 、 $\{\text{脐部} \neq \text{凹陷}\}$ 都是充分特征集



充分特征集

定理2.3 (充分特征集存在的充要条件) \mathcal{F} 对训练样本 $\mathcal{E} = \mathcal{P} \cup \mathcal{N}$ 是充分的, 当且仅当对于任意正负样例对 $\langle p, n \rangle \in \mathcal{P} \times \mathcal{N}$ 都至少存在一个 $\mathbf{f} \in \mathcal{F}$ 使得 $\mathbf{f} \sqsupseteq \langle p, n \rangle$ 。

证明:

1. 必要性: 若存在 $\langle p, n \rangle$ 使得 \mathcal{F} 中任意特征均无法区分, 无法构建一条一致规则覆盖 p 。
2. 充分性: 轮流对正样例 p_i 进行以下操作

- » 令 $\mathcal{F}_i = \{\mathbf{f}_{i,j} \in \mathcal{F} \mid j \in \{1, \dots, |\mathcal{N}|\} \wedge \mathbf{f}_{i,j} \sqsupseteq \langle p_i, n_j \rangle\}$
- » 利用 \mathcal{F}_i 构建规则 $r_i \equiv (\textit{positive} \leftarrow \mathbf{f}_{i,1} \wedge \dots \wedge \mathbf{f}_{i,|\mathcal{N}|})$
- » 可知, r_i 必然覆盖 p_i 且不覆盖任何负样例, 是一致的
- » 对所有 $p_i \in \mathcal{P}$ 进行以上操作, 可令 $\mathcal{R} = \{r_1, \dots, r_{|\mathcal{P}|}\}$ 完备

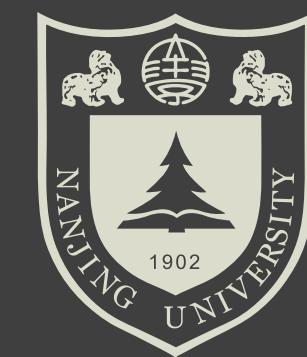


完全无关特征集

定义2.4 (完全无关特征) 一个特征 $f \in \mathcal{F}$ 对与样本集 $\mathcal{E} = \mathcal{P} \cup \mathcal{N}$ 是完全无关的当且仅当不存在 $\langle p, n \rangle \in \mathcal{P} \times \mathcal{N}$ 使得 $f \sqsupset \langle p, n \rangle$ 。

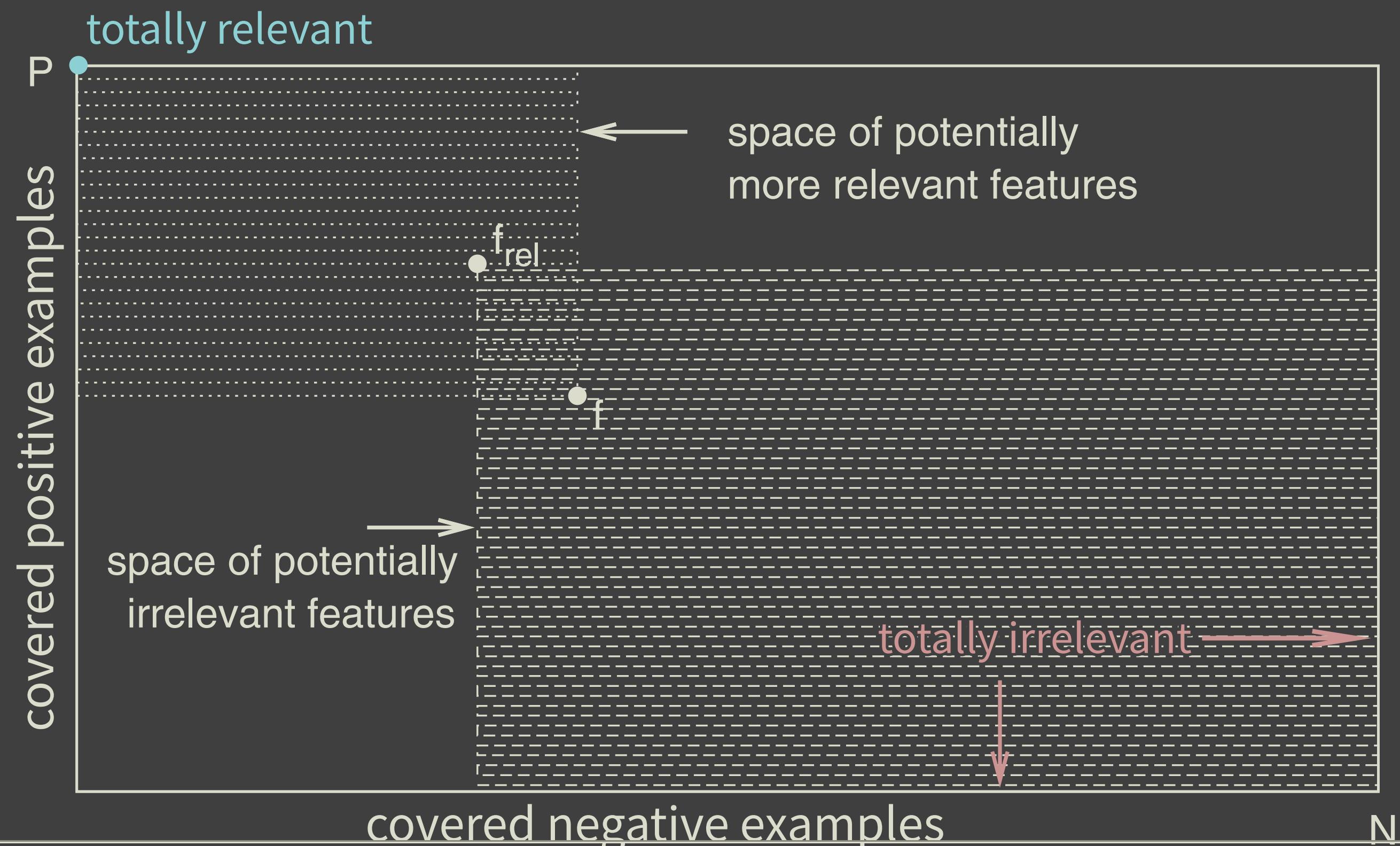
ID	色泽	根蒂	敲击	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
4	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
5	青绿	硬挺	清脆	清晰	平坦	软粘	否
6	浅白	硬挺	清脆	模糊	平坦	硬滑	否

- > “敲击 ≠ 浊响”就是一个完全无关特征
 - » 它覆盖正样例2，但完全无法区分2与其它负样例
- > “脐部 = 平坦”也是一个完全无关特征：它无法覆盖任意一个正样例



相对无关性

定义2.5 (相对无关性) 令 \mathcal{PN}_f 表示特征 f 在样本集 $\mathcal{E} = \mathcal{P} \cup \mathcal{N}$ 上正确区分的正负样例对。若 $\mathcal{PN}_{f_1} \subseteq \mathcal{PN}_{f_2}$, 则称 f_2 相对 f_1 更无关。





离散值生成特征的算法

```
1: # P/N: 正样例/负样例
2: # A: 属性集合
3: function gen_discrete_attribute_feature(P, N, A)
4:     F = []
5:     for attr_i in A # 第i个属性
6:         # 正样例里出现的所有属性值
7:         V_P = [p[attr_i] for p in P]
8:         # 负样例里出现的所有属性值
9:         V_N = [n[attr_i] for n in N]
10:        for v in V_P
11:            push!(F, Expr(attr_i == v))
12:        end
13:        for v in V_N
14:            push!(F, Expr(attr_i != v))
15:        end
16:    end
17:    return F
18: end
```

- > 易证明，该算法不会生成完全无关特征



连续值生成特征的算法

```
1: # P/N: 正样例/负样例
2: # A: 属性集合
3: function gen_continuous_attribute_feature(P, N, A)
4:     F = []
5:     for attr_i in A # 第i个属性
6:         # 正样例里出现的所有属性值
7:         V_P = [p[attr_i] for p in P]
8:         # 负样例里出现的所有属性值
9:         V_N = [n[attr_i] for n in N]
10:        # 将所有样本中属性i的取值排序
11:        V = sort!(values(attr_i))
12:        for i in 1:length(V)-1
13:            # 两个取值的中间值
14:            v = (v[i] + v[i+1])/2
15:            if (v[i] in V_P) && (v[i+1] in V_N) # i正, i+1负
16:                push!(F, Expr(attr_i < v))
17:            end
18:            if (v[i] in V_N) && (v[i+1] in V_P) # i负, i+1正
19:                push!(F, Expr(attr_i >= v))
20:            end
21:        end
```



生成连续特征的方法

ID	密度	含糖率	好瓜	覆盖
1	0.697	0.46	是	f_1, f_3
2	0.774	0.376	是	f_1, f_3
3	0.556	0.215	是	f_1, f_2
4	0.666	0.091	否	f_1
5	0.243	0.267	否	f_2
6	0.245	0.057	否	f_2

以密度为例：

I. 排序：

» 0.243 (n), 0.245 (n), 0.556 (p), 0.666 (n), 0.697 (p),
0.774 (p)

2. 找到分界点，构建特征（覆盖正/负样例数目）：

» $f_1 \equiv \text{密度} \geq (0.245 + 0.556)/2$ (3/1)
» $f_2 \equiv \text{密度} < (0.556 + 0.666)/2$ (1/2)
» $f_3 \equiv \text{密度} \geq (0.666 + 0.697)/2$ (2/0)

3. 容易构建规则集合：

» 好瓜 $\leftarrow (f_1 \wedge f_2)$
» 好瓜 $\leftarrow (f_1 \wedge f_3)$



生成连续特征的方法

没有被该算法所生成的特征，要么完全无关，要么比该算法生成的特征相对更无关。

- > 也就是说，若 v_i 和 v_{i+1} 符号相同，则没必要用特征值切开
- > 不失一般性，假设 $v_i(p)$ 且 $v_{i+1}(p)$ ，并用特征 $f \equiv A \geq \frac{v_i+v_{i+1}}{2}$ 将它们切开
- > 假设有 $\dots, v_n, v_p, \dots, v_i, (\frac{v_i+v_{i+1}}{2}), v_{i+1}$
 - » $v_n(n)$ 是小于 v_i 最大的负样本取值
 - » $v_p(p)$ 是 v_i 和 v_n 间最小的正样本取值
- > 若存在 v_n ，那么 f 比 $(v_n + v_p)/2$ 相对更无关
 - » 它少覆盖一些正样例，却覆盖相同的负样例
- > 若不存在这样的 v_n ，那么 f 覆盖所有负样例，因此完全无关
- > 对于 $v_i(n)$ 且 $v_{i+1}(n)$ 同理可证



属性值缺失时

1. 删除法：

- » 删除有缺失值的属性
- » 删除有缺失值的样例

2. 默认值法：

- » 默认填False, True或一个特殊值作为新的属性值（如“MISSING”）

3. 估计法：

- » 平均值、众数
- » 预测法：用其它特征训练一个预测该属性的模型
- » 分布估计（如高斯）再采样

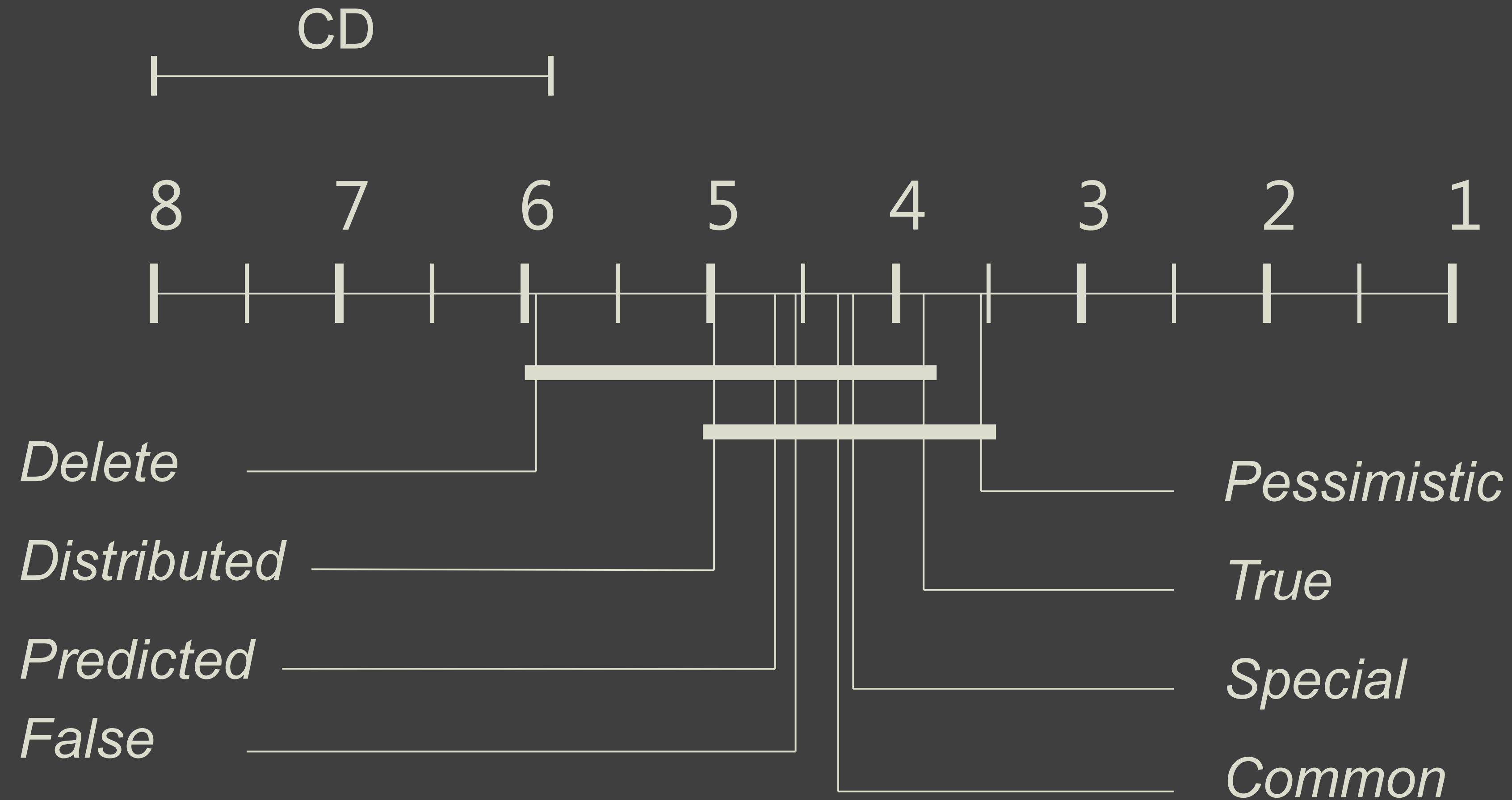
4. “悲观”法：

- » 对正样例填False, 负样例填True
- » 该属性的正原子命题（不含否定符）区分的 $\langle p, n \rangle$ 对减少
- » 尽量减少使用有缺失值的属性特征



处理属性值缺失的方法对比

UCI数据集上的结果[Wohlrab and Fürnkranz, 2011]:





符号学习简介 · 命题规则（上）

1. 命题（逻辑）规则
2. 命题规则学习
 - » 命题特征构建
 - » 命题规则搜索
3. 打分函数



命题规则搜索

目标：学习一条

- I. 尽可能完备、
2. 尽可能一致、
3. 最简单的

命题逻辑规则。



命题规则搜索算法框架

```
1: # E = P + N: 样本集合
2: # F: 特征集合
3: function find_best_rule(E, F)
4:     # 初始化规则
5:     r_best = init_rule(E, F)
6:     # 计算规则打分
7:     h_best = score(r_best, E)
8:     # 初始化备选规则集
9:     R = [ r_best ]
10:    # 开始对每条备选规则进行精化
11:    while !empty(R)
12:        for r in R
13:            # 对r进行精化
14:            rho_r = refine_rule(r, E, F)
15:            for r_refined in rho_r
16:                # 如果符合停止标准(如过长)则跳过
17:                if meet_stop_criterion(r_refined)
18:                    continue
19:                else
20:                    # 计算打分
21:                    h_refined = score(r_refined, F)
```



精化算子

定义2.6（精化算子）若 S 是一个集合，且 \preceq 是 S 上的一个偏序， S 的**精化算子** ρ 是一个 S 到 2^S 的映射，使得对任意 $r \in S$ 有 $r' \in \rho(r)$ 蕴涵 $r' \preceq r$ 。 r' 也被称为 r 的**特化**，而 r 则是 r' 的**泛化**。

- > 对于Horn子句来说，偏序 \preceq 就是蕴含关系

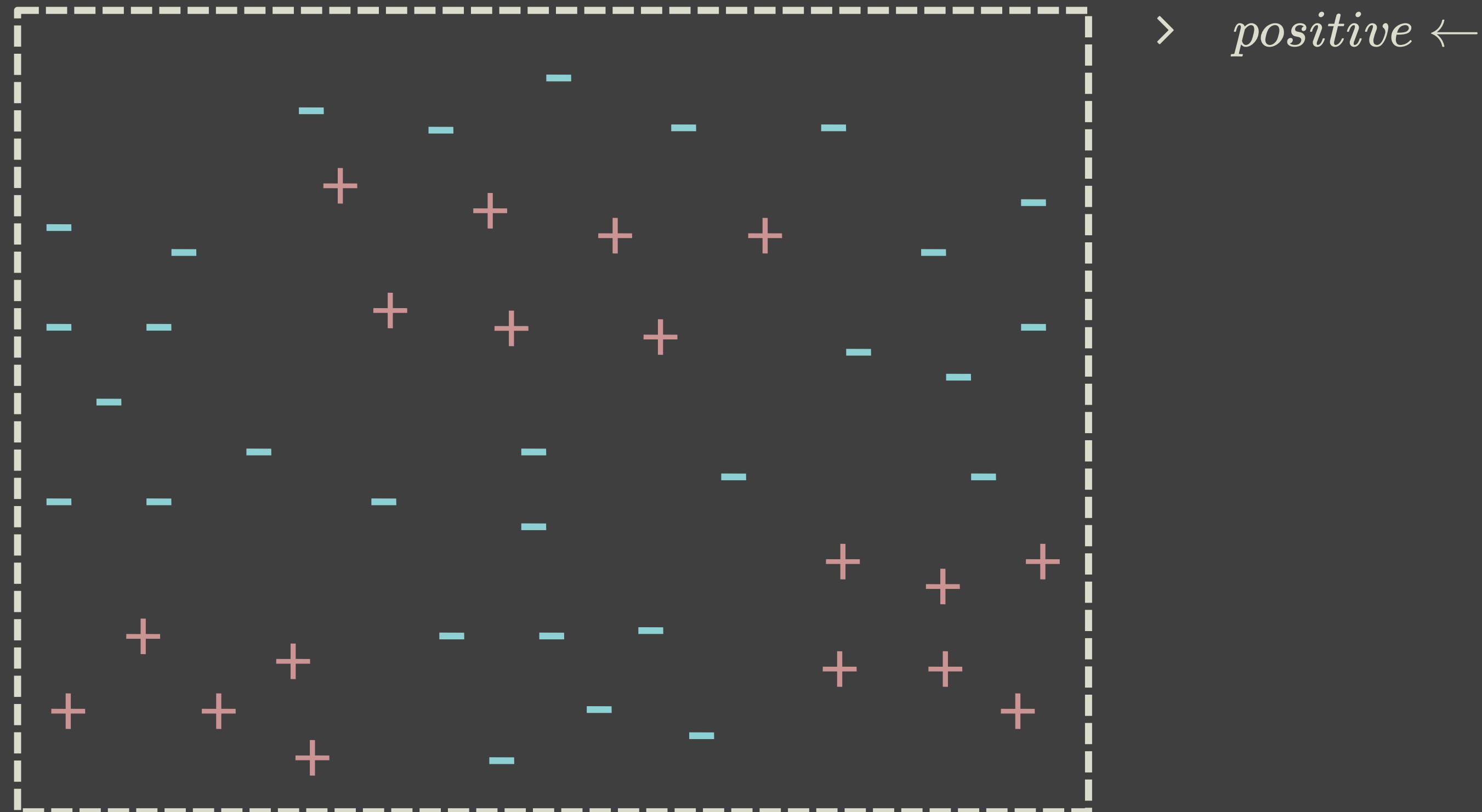
定义2.7（最小精化算子） ρ 是一个**最小精化算子**当且仅当在偏序 \preceq 形成的格上，任意 $r' \in \rho(r)$ 均为 r 的邻居。

- > 比如“删一个文字”就是最小精化算子，“删两个以上的文字”就不是



特化 (SPECIALISATION)

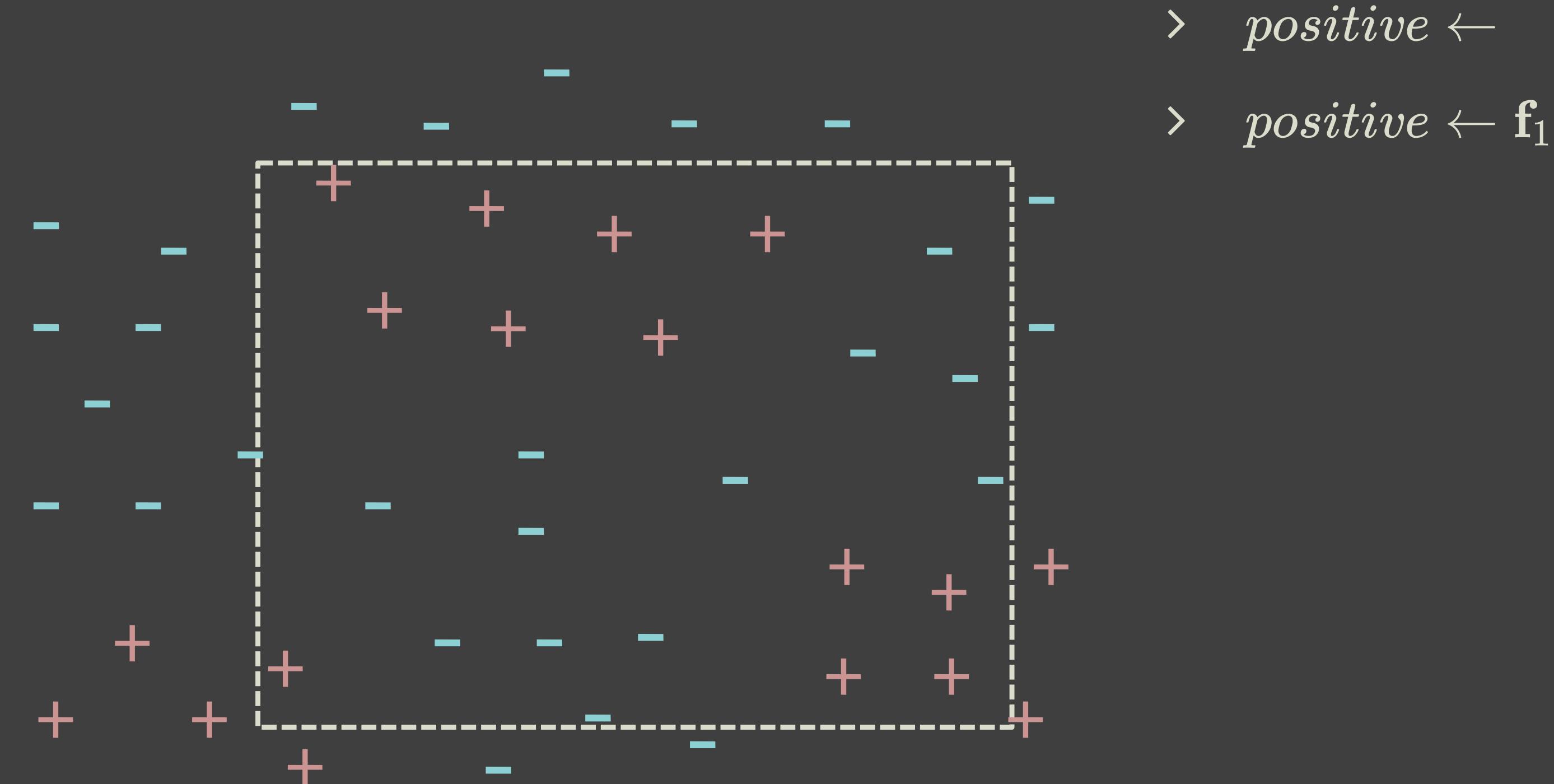
在命题规则学习中，特化就是向规则中增加文字：





特化 (SPECIALISATION)

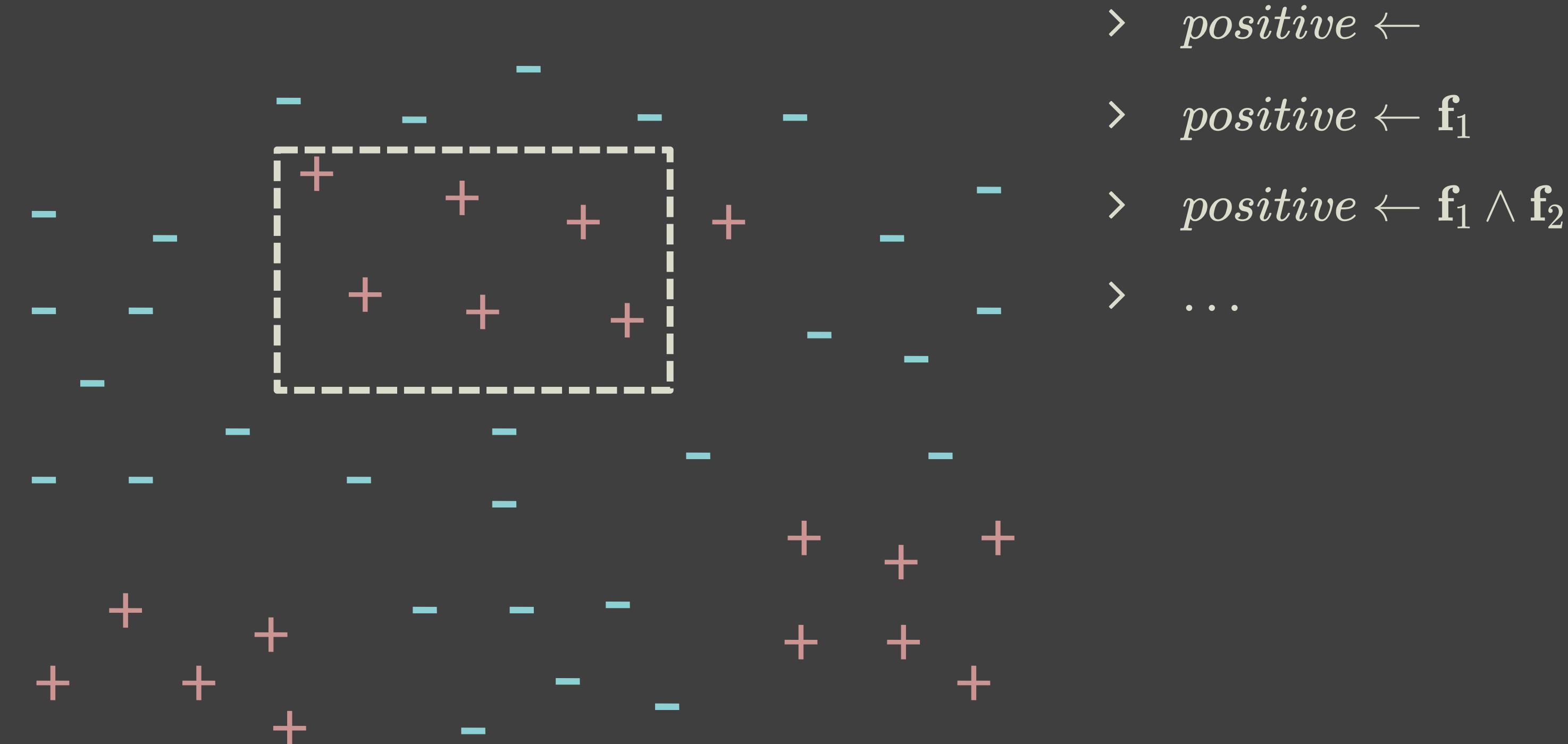
在命题规则学习中，特化就是向规则中增加文字：





特化 (SPECIALISATION)

在命题规则学习中，特化就是向规则中增加文字：

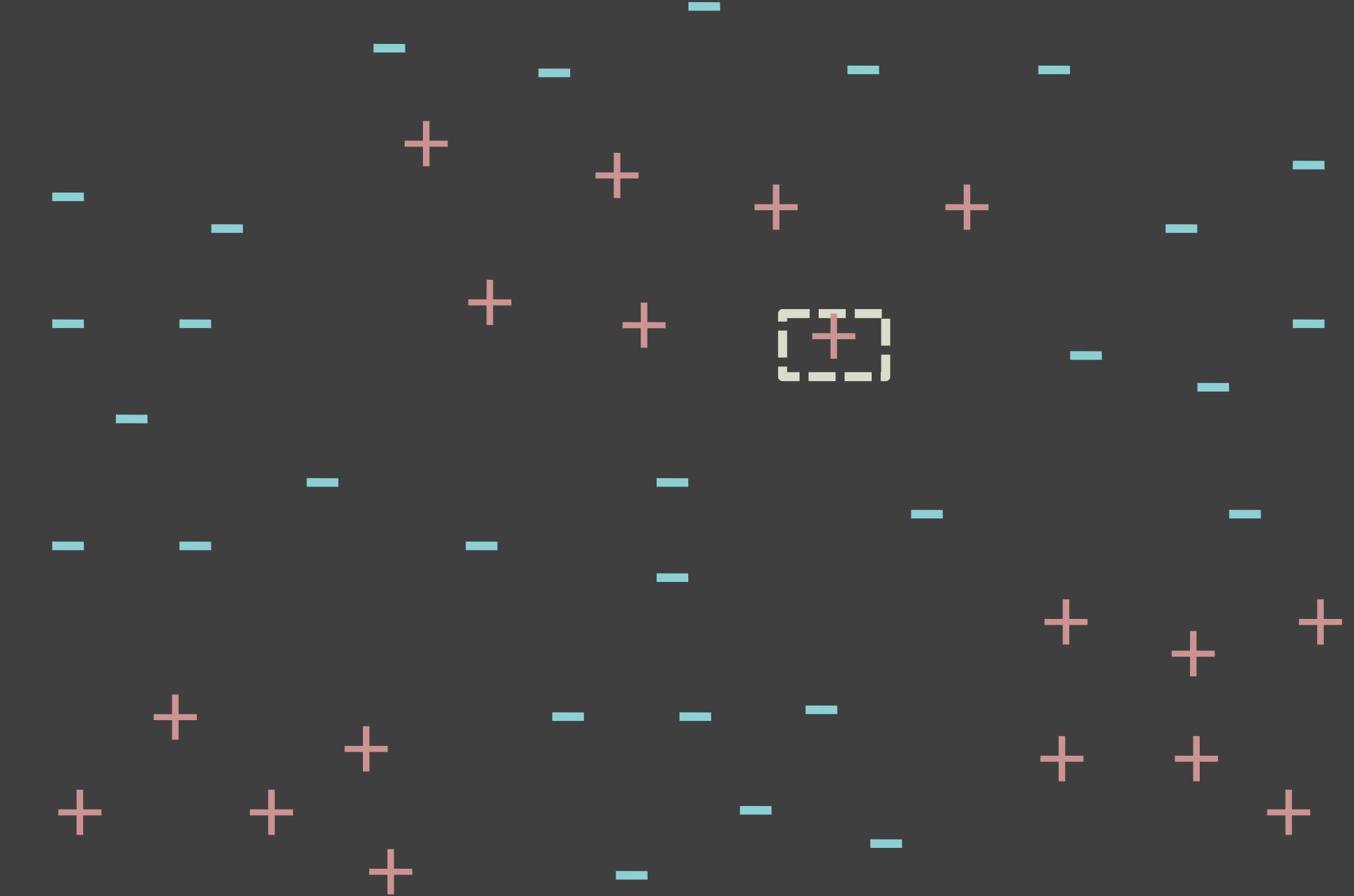




泛化 (GENERALISATION)

在命题规则学习中，特化就是从规则中删除文字：

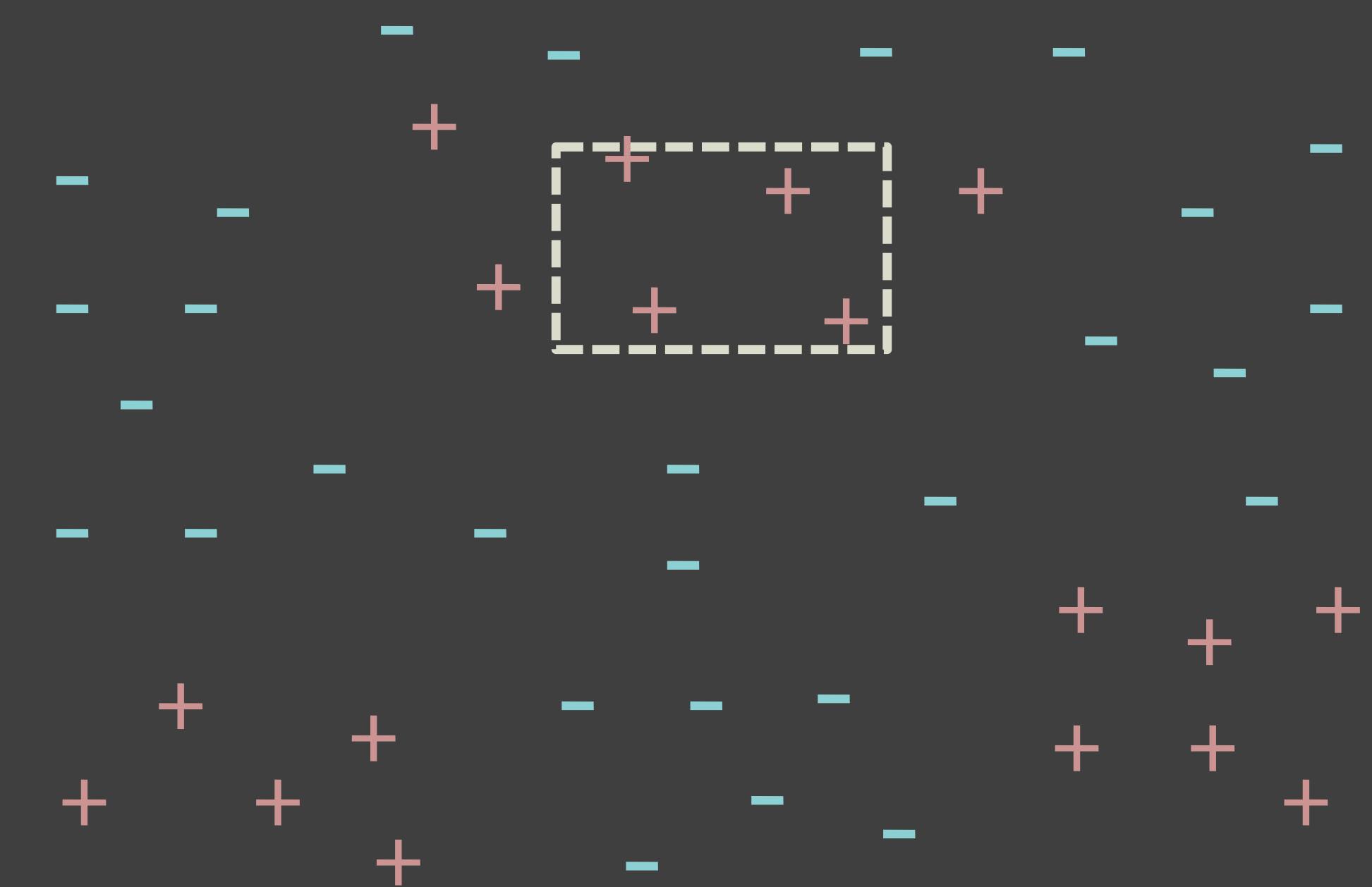
› $positive \leftarrow f_1 \wedge f_2 \wedge f_3 \wedge f_4$





泛化 (GENERALISATION)

在命题规则学习中，特化就是从规则中删除文字：

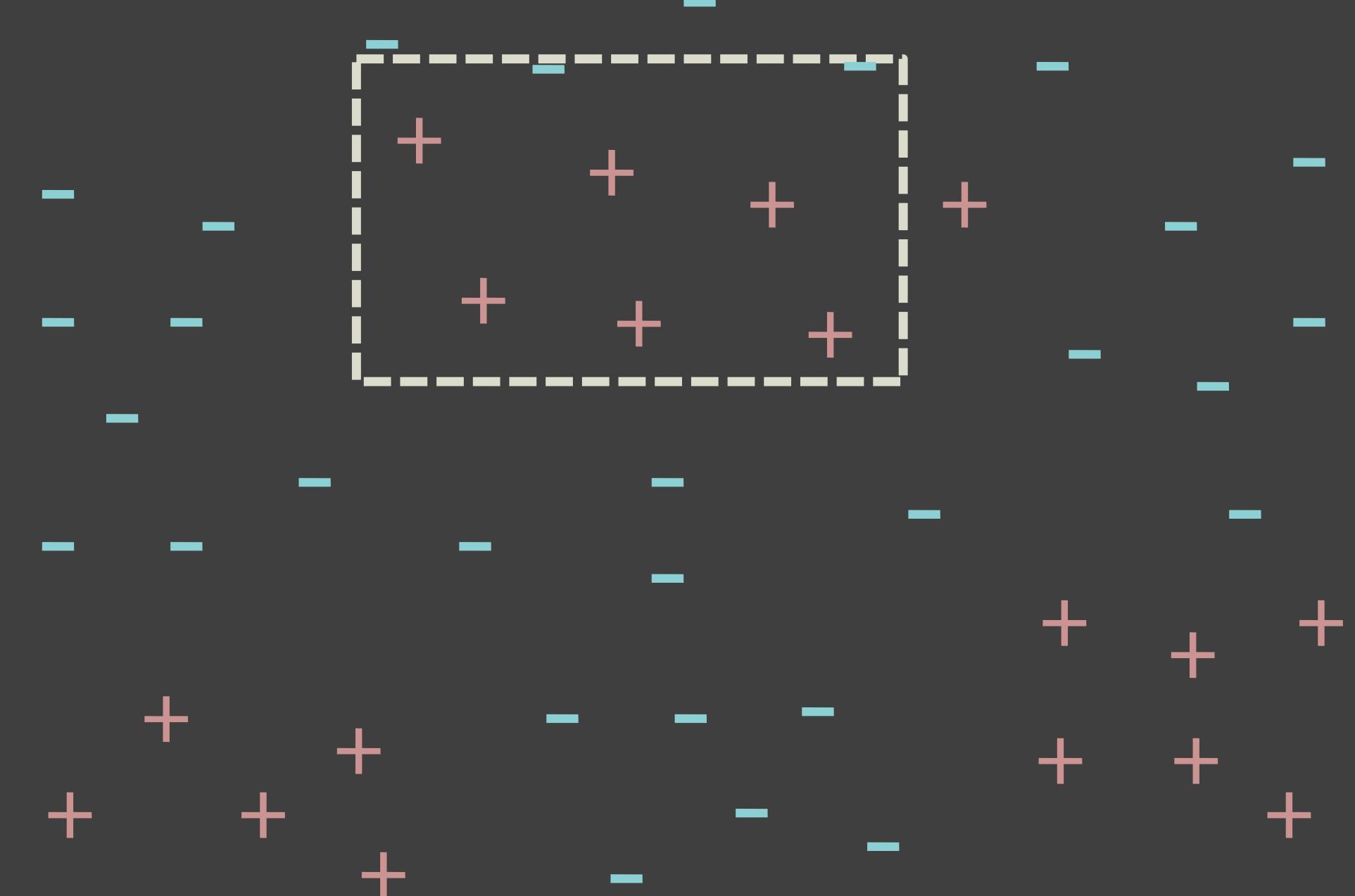


- › $positive \leftarrow f_1 \wedge f_2 \wedge f_3 \wedge f_4$
- › $positive \leftarrow f_1 \wedge f_2 \wedge f_4$



泛化 (GENERALISATION)

在命题规则学习中，特化就是从规则中删除文字：



- > $positive \leftarrow f_1 \wedge f_2 \wedge f_3 \wedge f_4$
- > $positive \leftarrow f_1 \wedge f_2 \wedge f_4$
- > $positive \leftarrow f_2 \wedge f_4$
- > ...



搜索策略：启发式搜索

I. Hill-climbing: 每次只保留最优规则

» 尽可能避免局部最优: Look- n -step-ahead (ATRIS) [Mladenić, 1993]

2. Beam-search: 每次保留 b 条最优规则

» 被大量方法采用: AQ [Michalski, Mozetič, Hong and Lavrač, 1986], CN2 [Clark and Niblett, 1989], mFOIL [Džeroski and Bratko, 1992], BEXA [Theron and Cloete, 1996] 等等

3. A*: 通过设置启发式函数估计未来的精化算子打分



搜索策略：枚举搜索

I. 最佳优先搜索 (Best-first-search)

» 保留所有未精化过的规则, $b = \infty$ 的beam search

2. 层次搜索 (Level-wise search)

» 例如挖掘频繁项集的APRIORI算法

3. 随机搜索：

» 精化算子计算过程中引入随机性（马尔可夫链、模拟退火）

» 演化算法

4. OPUS (Optimized Pruning for Unordered Search) [Webb, 1995]

» 为特征定义一个序: $f_1 > f_2 > \dots > f_F$

» 第 I 轮只考虑长度为 I 的规则: $positive \leftarrow f_i, i \in \{1, \dots, F\}$

» 第 k 轮只考虑长度为 k 的规则, 但精化算子只计算序比当前规则大的特征:

» 例如 $positive \leftarrow f_3 \wedge f_4 \wedge f_6$ 只考虑 f_1 和 f_2

» 搜索过程中预剪枝



符号学习简介 · 命题规则（上）

1. 命题（逻辑）规则
2. 命题规则搜索
3. 打分函数



打分函数

打分函数是一种启发式函数，用于定量地描述规则的质量：

1. 一致性（consistency）：覆盖多少负样例？
2. 完备性（completeness）：覆盖多少正样例？
3. 增益（Gain）：当前规则比其它规则好多少？
4. 简洁性（Simplicity）：规则是否简单、好理解？
5. 偏好（Bias）：规则是否有太强的bias，可能导致过拟合？

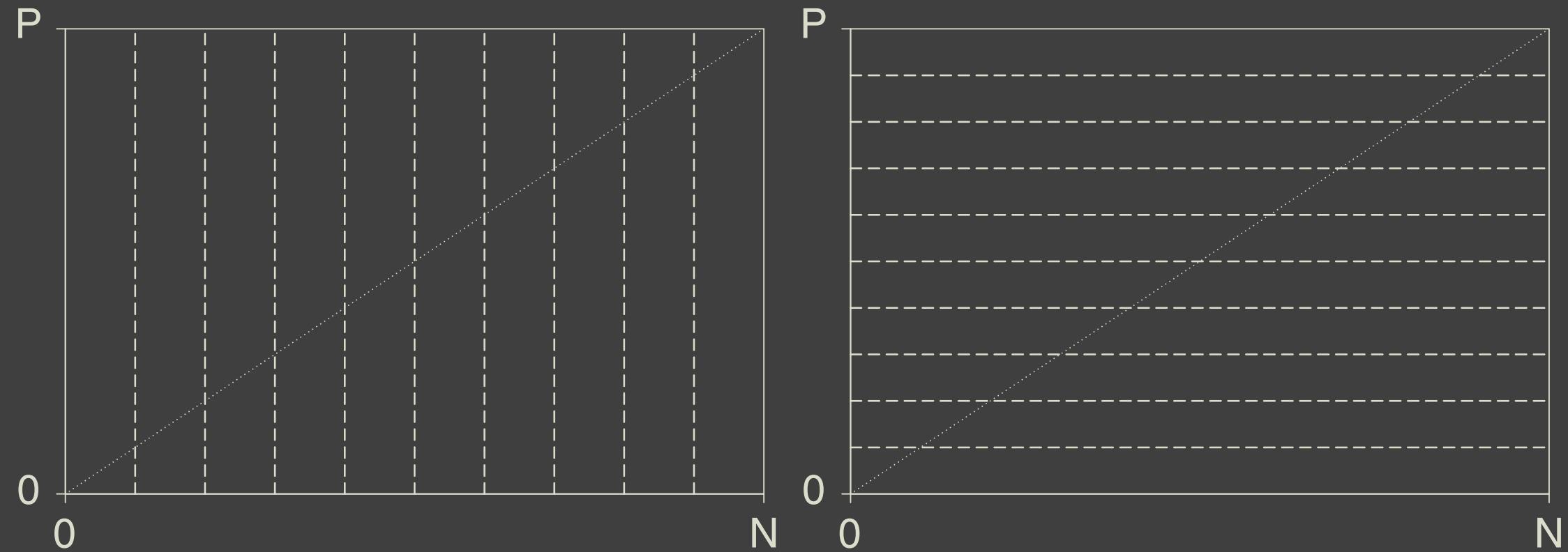


基本评价指标

名字	记号
正样例数	P
负样例数	N
覆盖正样例数 (TP)	\hat{P}
覆盖负样例数 (FP)	\hat{N}
未覆盖正样例数 (FN)	\bar{P}
未覆盖负样例数 (TN)	\bar{N}



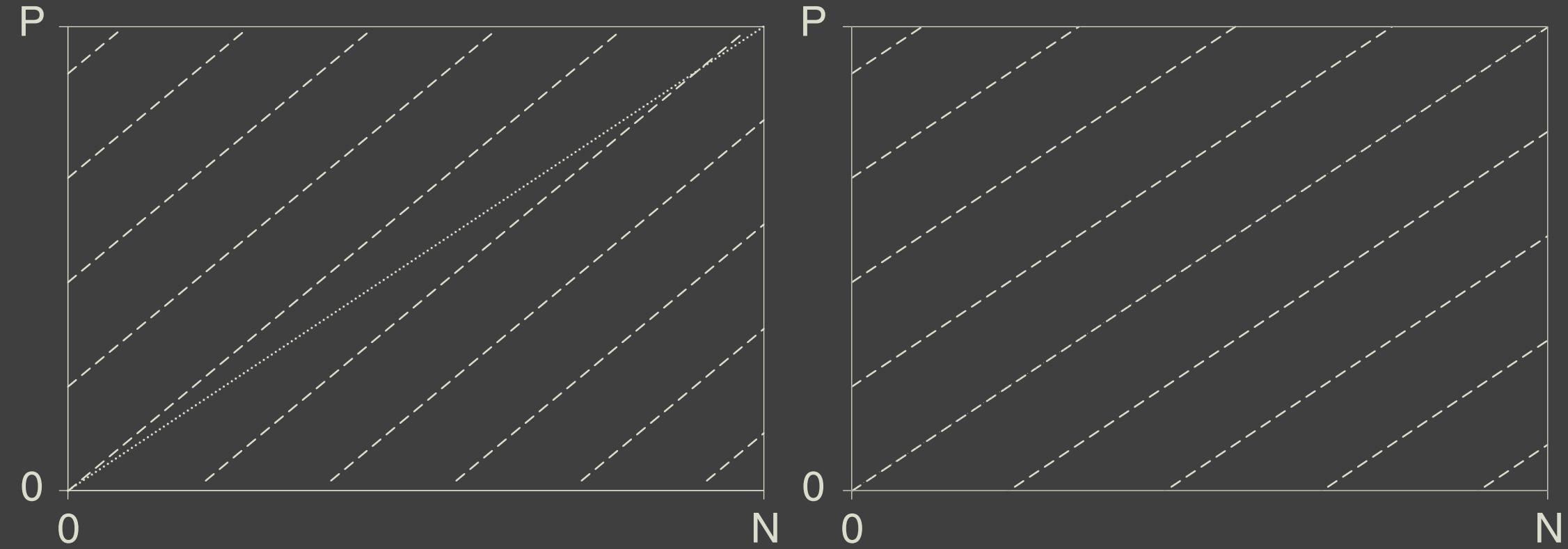
基础打分函数



- > 负样本覆盖率: $\hat{N} \sim \frac{\hat{N}}{N}$
 - » 未覆盖负样本: $\bar{N} = N - \hat{N} \sim -\hat{N}$
- > 正样本覆盖率 (召回): $\hat{P} \sim \frac{\hat{P}}{P}$



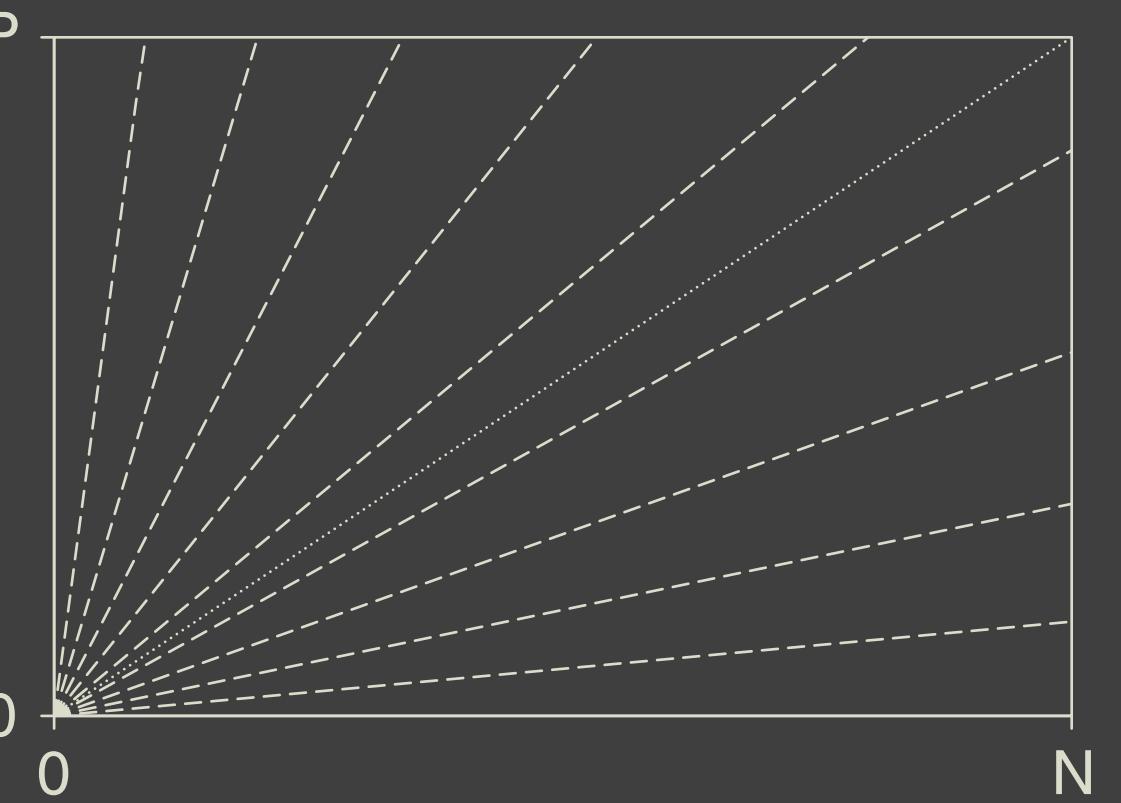
精度



- > 覆盖差 (coverage difference) : $\hat{P} - \hat{N}$
 - » 精度 (accuracy) : $\frac{\hat{P} + \bar{N}}{\hat{P} + N}$
- > 覆盖率差 (coverage rate difference) : $\frac{\hat{P}}{P} - \frac{\hat{N}}{N}$
 - » 带权相对精度 (weighted relative accuracy) : $\frac{\hat{P} + \hat{N}}{\hat{P} + N} \cdot \left(\frac{\hat{P}}{\hat{P} + \hat{N}} - \frac{P}{P + N} \right)$
- > 一般形式 (线性代价) : $(1 - c) \cdot \hat{P} - c \cdot \hat{N}$



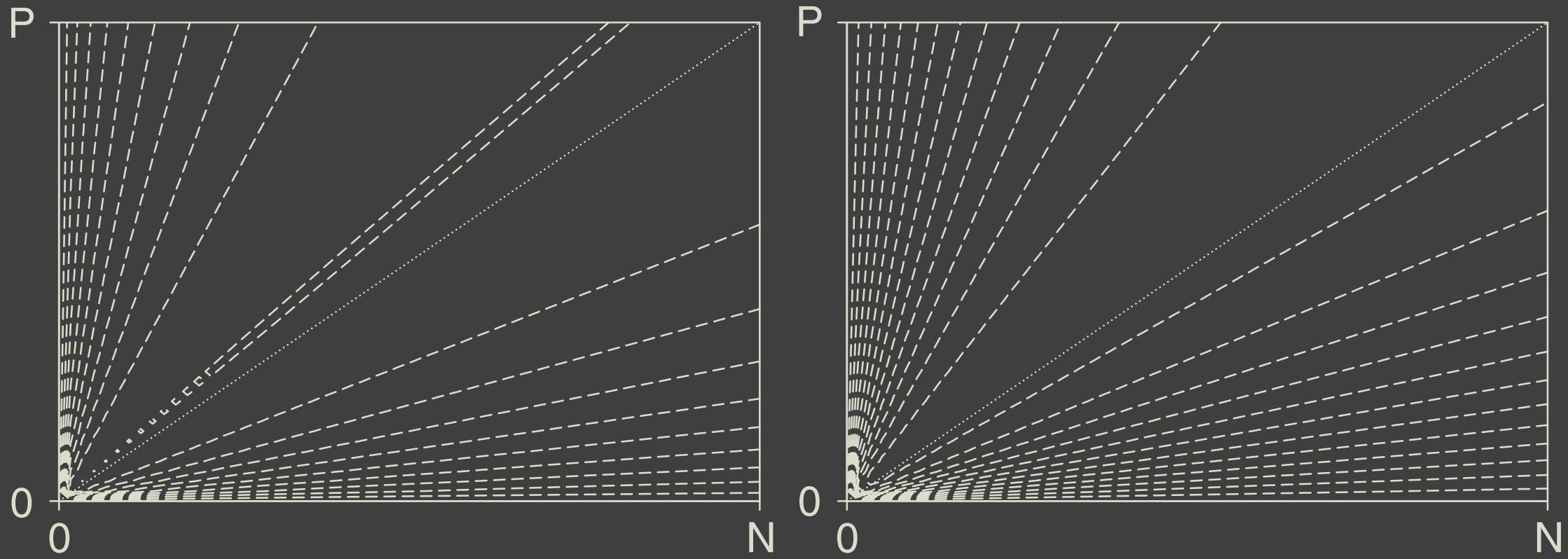
准确率 (查准率)



- > 准确率 (precision) : $\frac{\hat{P}}{\hat{P} + \hat{N}}$
- » RIPPER剪枝打分函数: $\frac{\hat{P} - \hat{N}}{\hat{P} + \hat{N}}$
- » 覆盖比率 (covering ratio) : $\frac{\hat{P}}{\hat{N}}$



信息熵

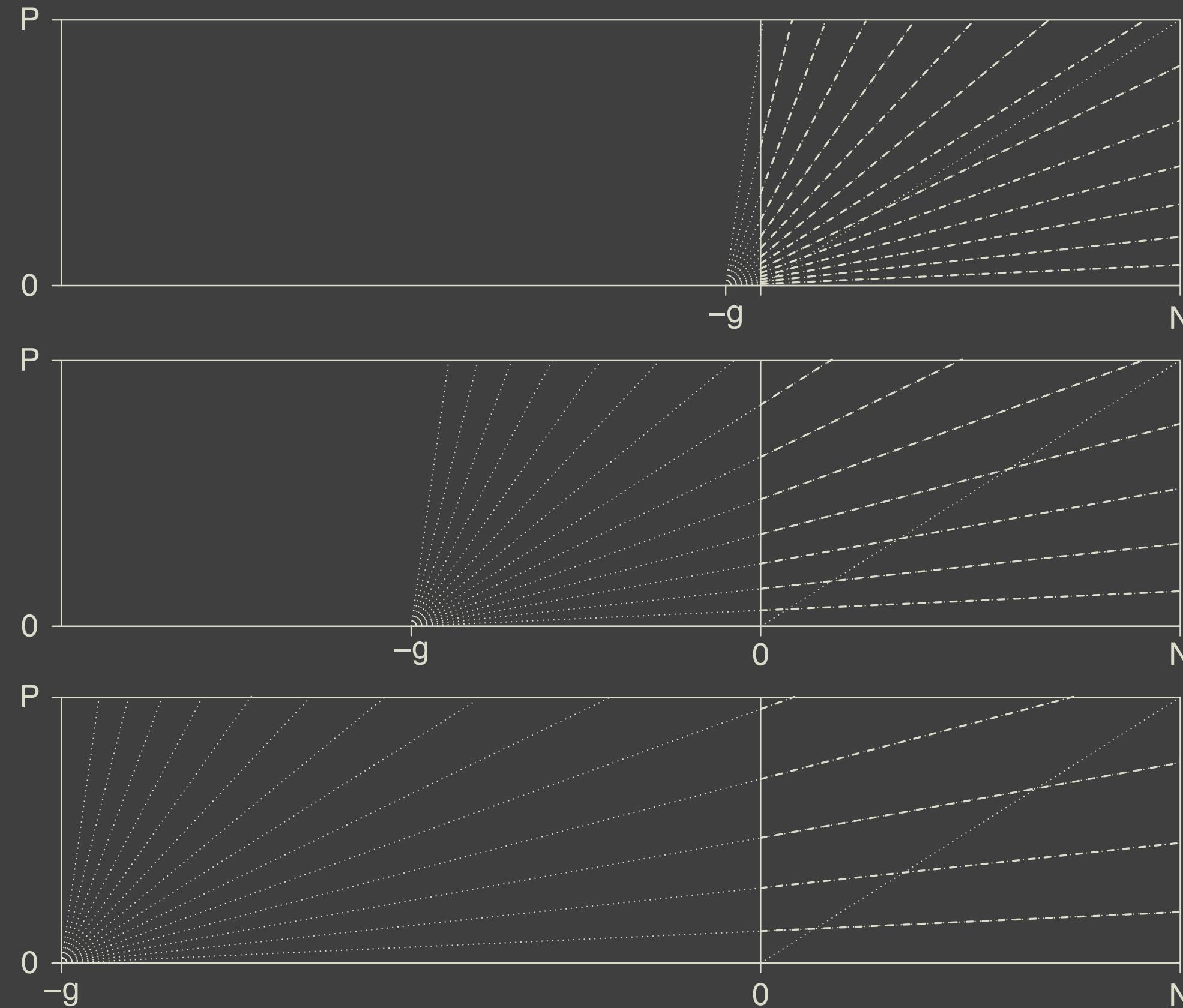


- > 信息熵 (entropy) : $-\left(\frac{\hat{P}}{\hat{P}+\hat{N}} \cdot \log_2 \frac{\hat{P}}{\hat{P}+\hat{N}} + \frac{\hat{N}}{\hat{P}+\hat{N}} \log_2 \frac{\hat{N}}{\hat{P}+\hat{N}} \right)$
- > 基尼系数 (Gini index) : $-\left(\frac{\hat{P}}{\hat{P}+\hat{N}} \right)^2 - \left(\frac{\hat{N}}{\hat{P}+\hat{N}} \right)^2$



F-度量

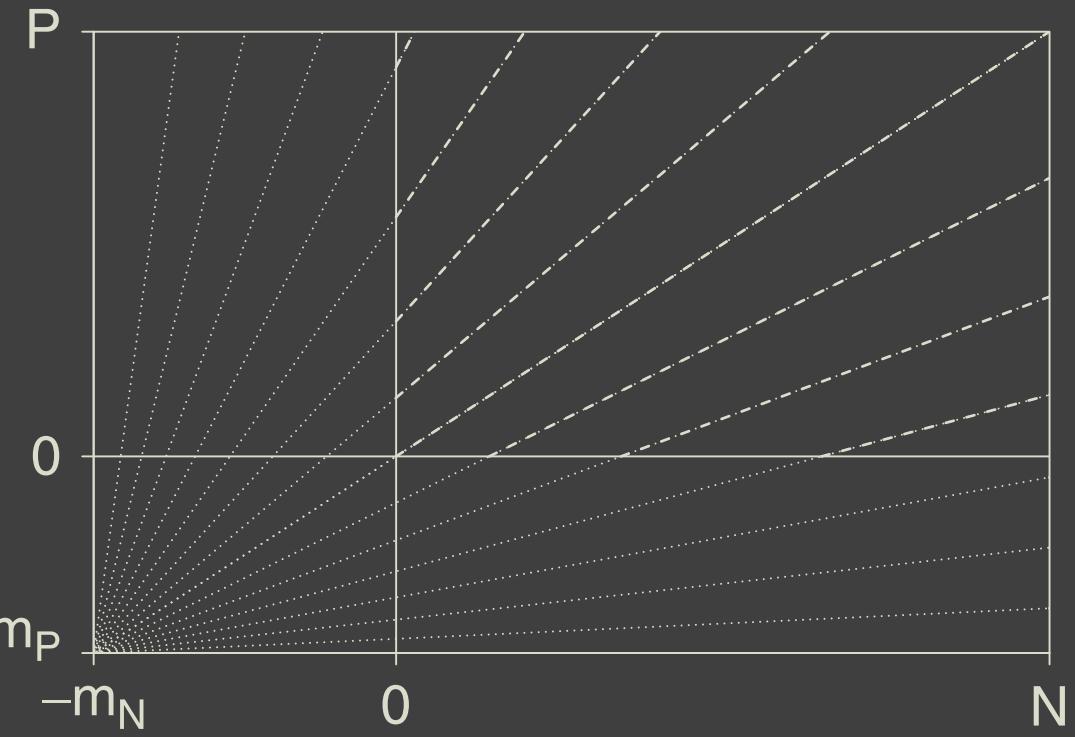
在查准率和查全率中折衷



$$\begin{aligned} & \frac{(\beta^2 + 1) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \\ = & \frac{(\beta^2 + 1) \cdot \frac{\hat{P}}{\hat{P} + \hat{N}} \cdot \frac{\hat{P}}{P}}{\beta^2 \cdot \frac{\hat{P}}{\hat{P} + \hat{N}} + \frac{\hat{P}}{P}} \\ = & \frac{(\beta^2 + 1) \cdot \hat{P}}{\hat{P} + \hat{N} + \beta^2 \cdot P} \\ \sim & \frac{\hat{P}}{\hat{P} + \hat{N} + g} \end{aligned}$$



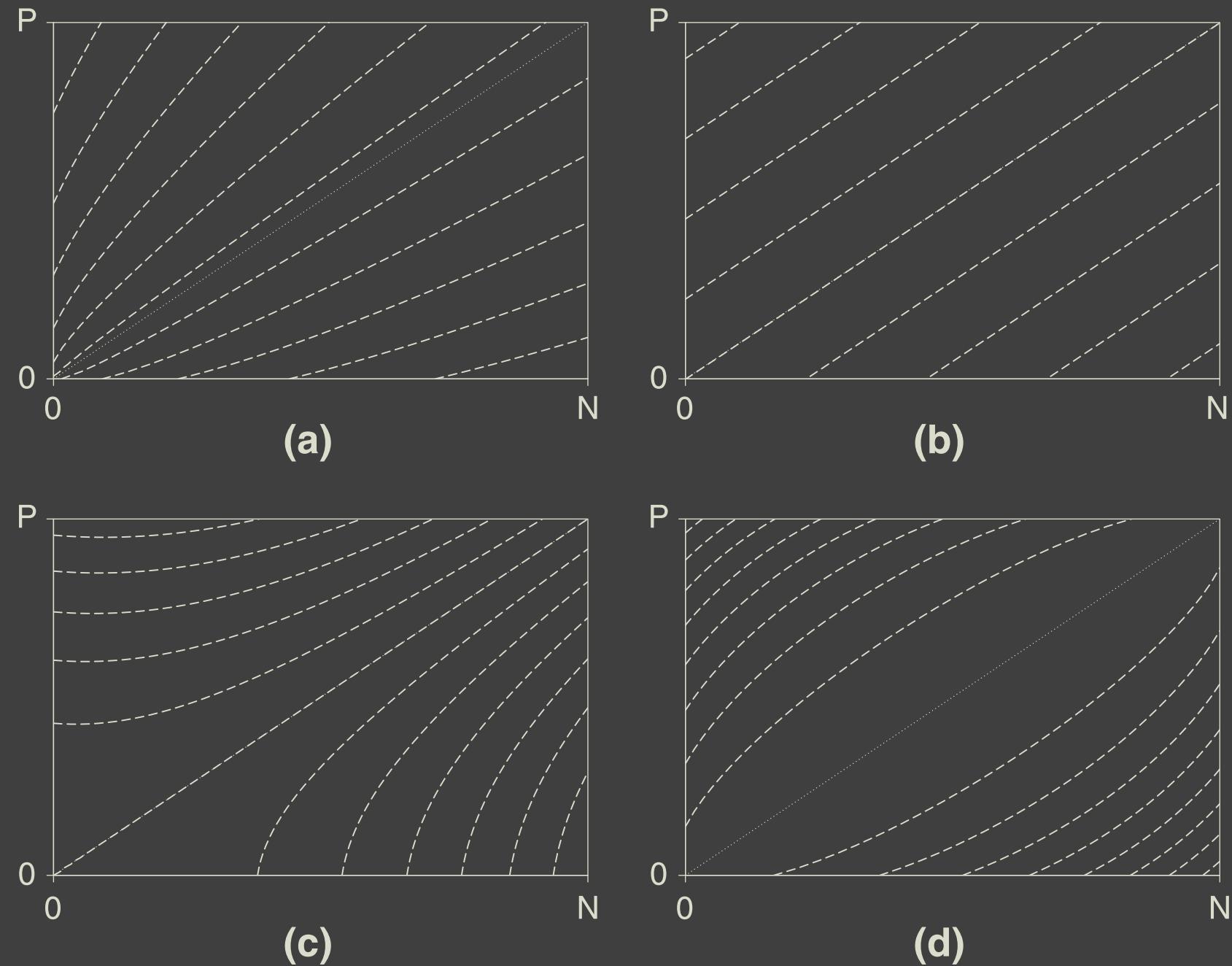
M-估计



- > Laplace估计 (Laplace estimate) : $\frac{\hat{P}+1}{\hat{P}+\hat{N}+2}$
 - » 规则准确率先验为 $\frac{1}{2}$
 - » 也就是估计：不覆盖任何训练样本的规则 ($\hat{P} = \hat{N} = 0$)，在样本空间的准确率是 $\frac{1}{2}$
- > m-估计 (m-estimate) : $\frac{\hat{P}+m \cdot \frac{P}{P+N}}{\hat{P}+\hat{N}+m}$
 - » 准确率先验为 $\frac{P}{P+N}$, 权重为 m
 - » $m_P = m \cdot \frac{P}{P+N}, m_N = m \cdot \frac{N}{P+N}$



准确率增益、KLÖSGEN度量

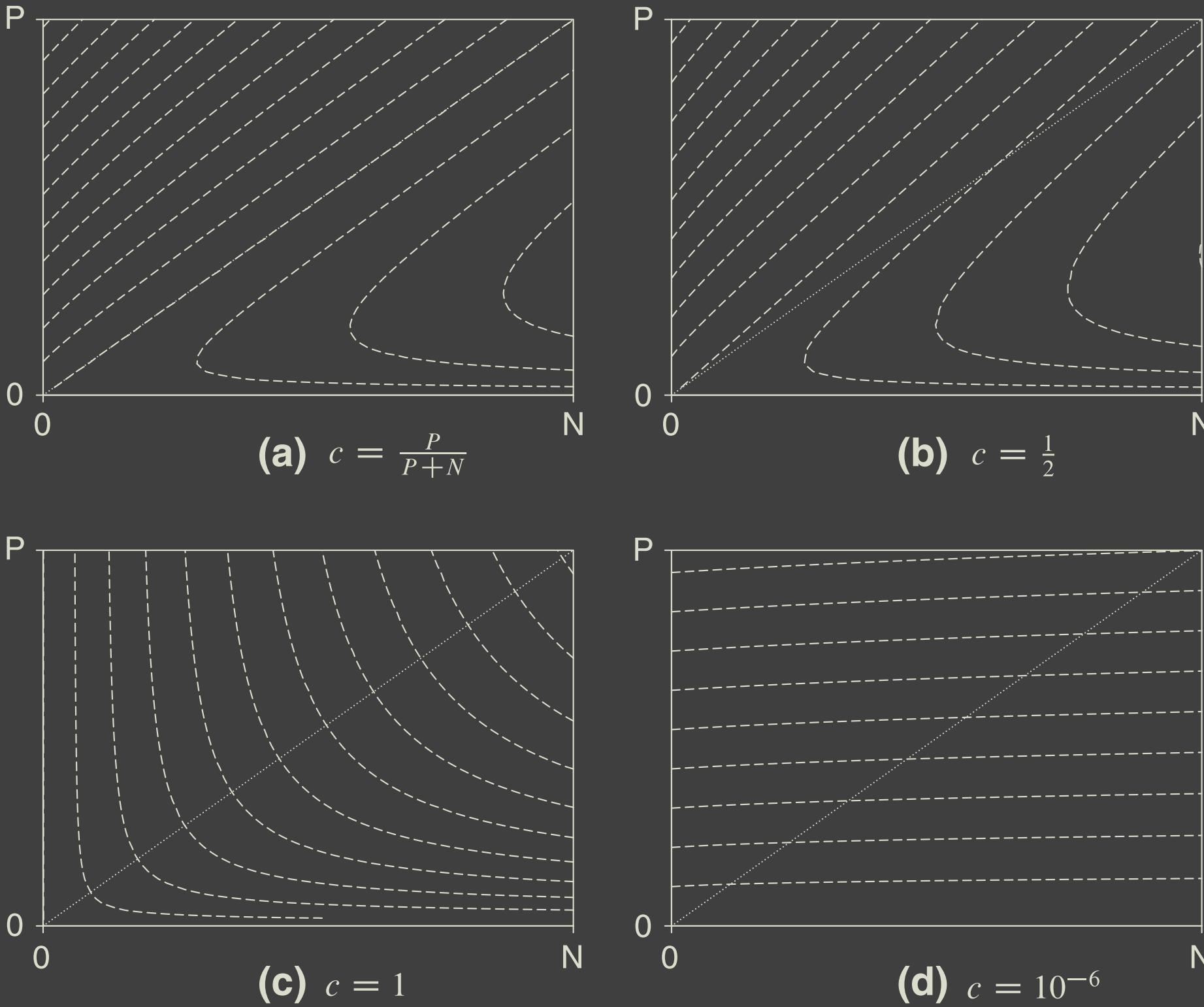


- > (Gain) $\frac{\hat{P}}{\hat{P}+\hat{N}} - \frac{P}{P+N}$
- > (Lift) $\frac{\frac{\hat{P}}{\hat{P}+\hat{N}}}{\frac{P}{P+N}}$
- > (Leverage) $\frac{\hat{P}}{P+N} - \frac{P}{P+N} \cdot \frac{\hat{P}+\hat{N}}{P+N}$

显然，这一类方法都和（带权相对）准确率 $\frac{\hat{P}}{\hat{P}+\hat{N}}$ 等价，
因此可以与样本覆盖率 $\frac{\hat{P}+\hat{N}}{P+N}$ 作各种复合：

1. $\sqrt{\text{覆盖率}} \cdot \text{准确率增益}$
2. $\text{覆盖率} \cdot \text{准确率增益}$
3. $\text{覆盖率}^2 \cdot \text{准确率增益}$
4. $\frac{\text{覆盖率}}{1-\text{覆盖率}} \cdot \text{准确率增益}$

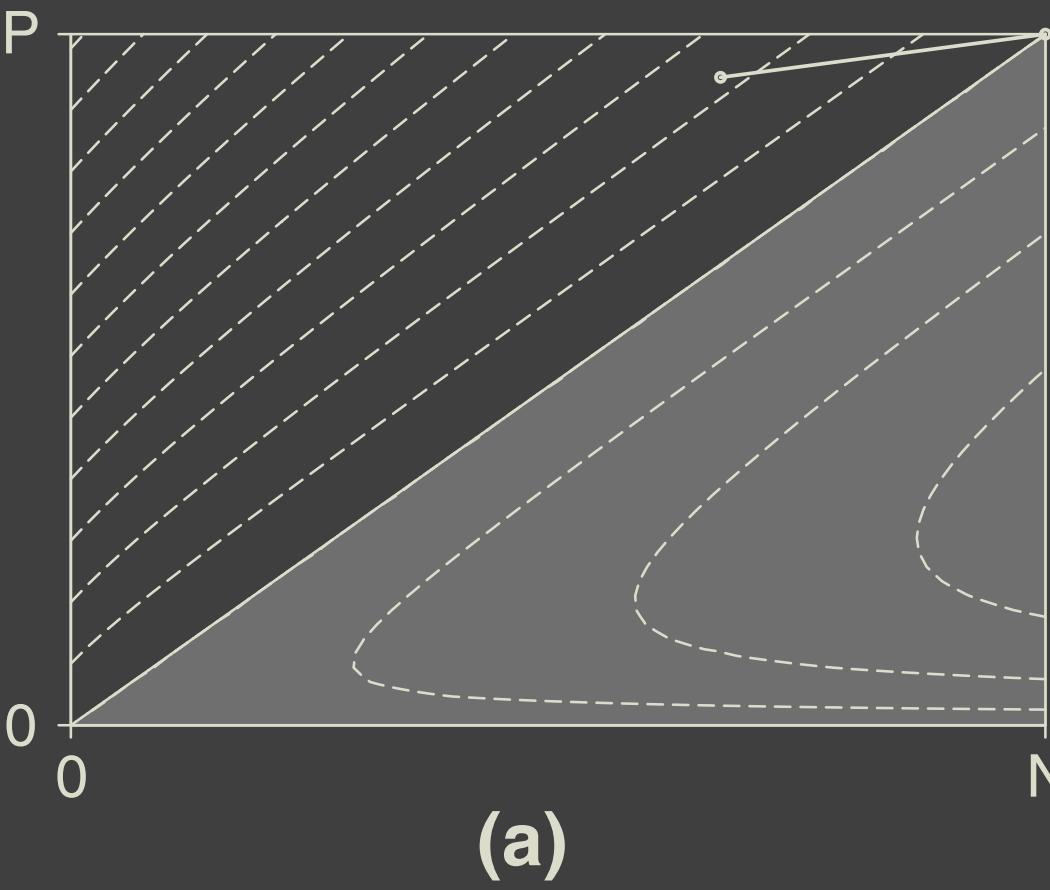
FOIL增益



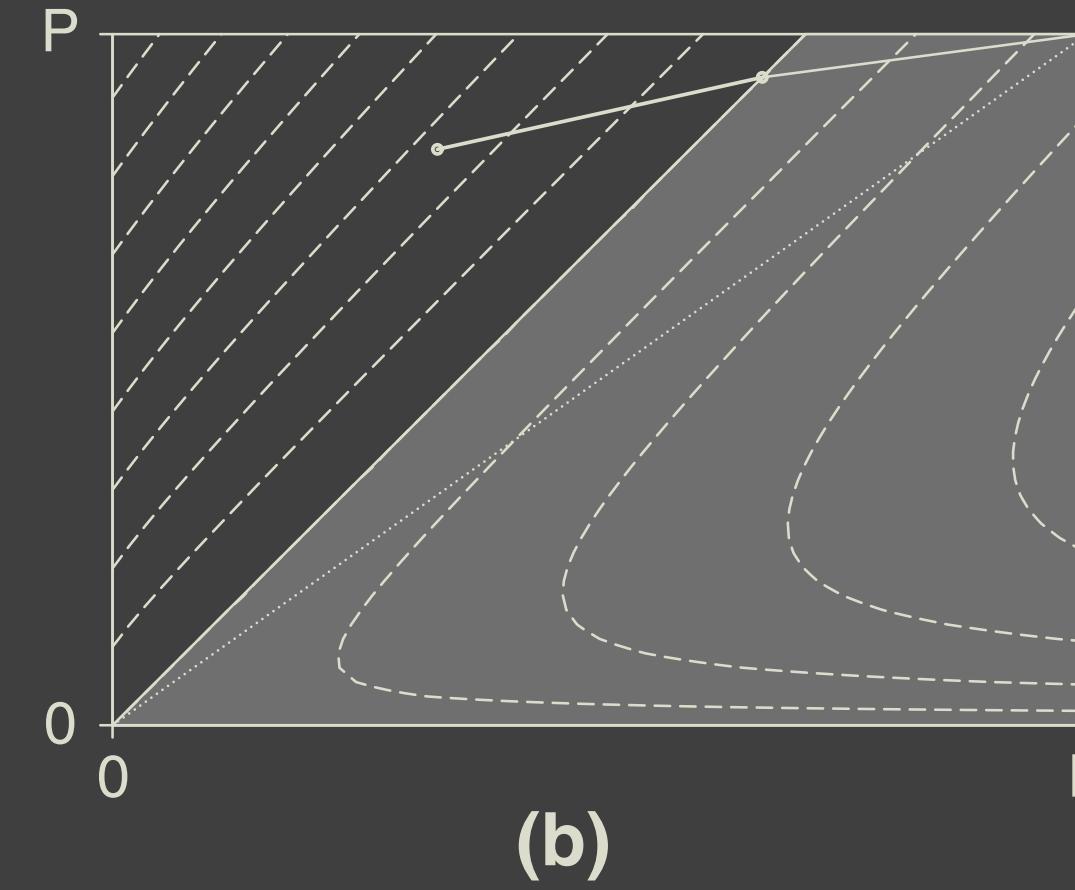
$$\text{正样本覆盖率} \cdot (\text{信息量} - \text{精化前信息量}) = \hat{P} \cdot \left(\log_2 \frac{\hat{P}}{\hat{P} + \hat{N}} - \log_2 c \right)$$



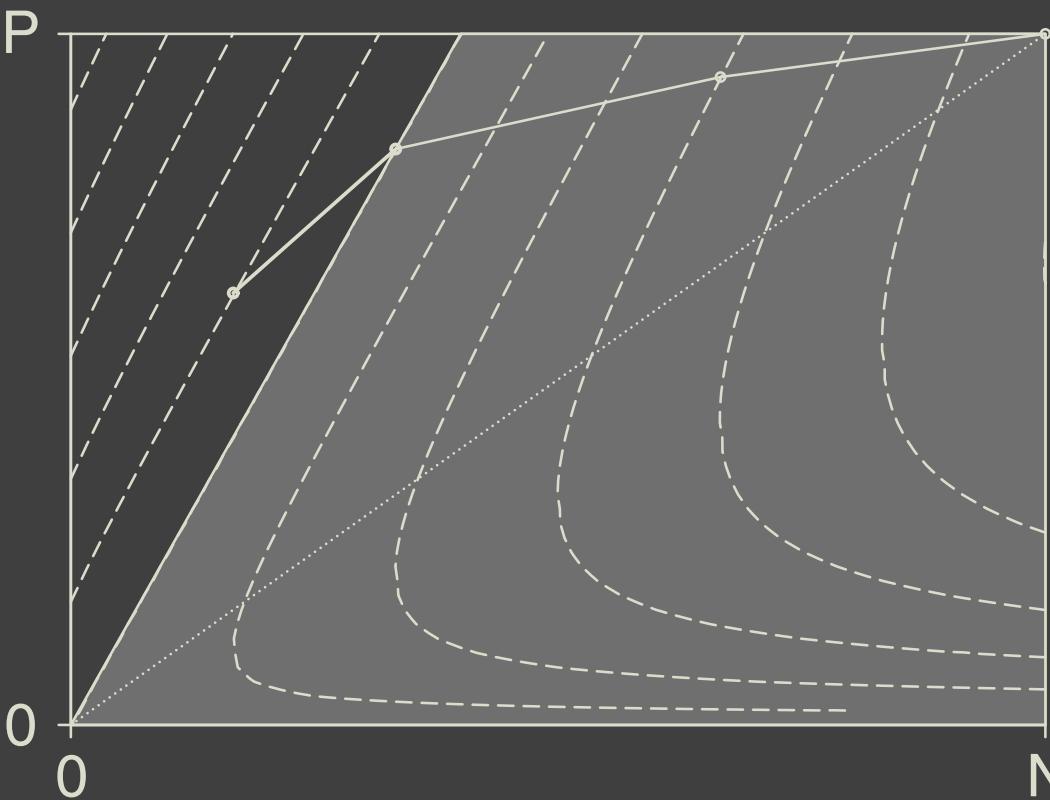
使用FOIL增益的规则精化



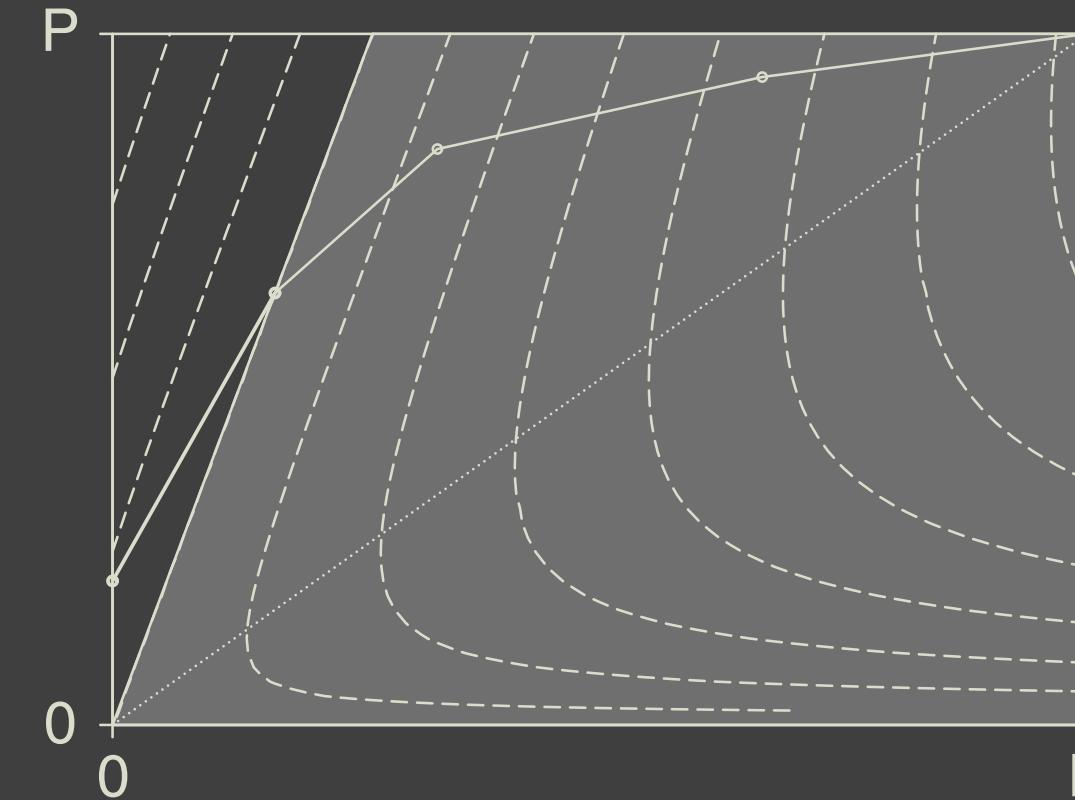
(a)



(b)



(c)



(d)



其它打分函数

I. 非线性打分函数，例如

- » J-Measure
- » Correlation
- » Odds ratio

2. 基于规则复杂度的度量：

- » 规则长度： $Length(\mathbf{r})$
- » 最小信息长度（Minimum Message Length）
- » 最小描述长度（Minimum Description Length）

3. 各种度量的线性组合

4. 各种度量的有序组合

- » 当打分函数 $score_i$ 无法区分时，使用打分函数 $score_{i+1}$



小结

符号学习

命题规则学习（上）

<https://daiwz.net>



命题规则（上）小结

- I. 命题规则是一种简化的命题逻辑规则
 - » 命题Horn子句
2. 构成命题规则的“词汇”（背景知识）：
 - » 属性、特征、命题逻辑文字
 - » 特征的相关性、如何生成充分特征集
3. 命题规则的搜索
 - » 精化算子、偏序、格（lattice）
 - » 启发式搜索、枚举搜索
4. 打分函数
 - » 打分函数之间的等价性
 - » 如何分析打分函数：PN图、等高线