人工智能学习笔记

西瓜书学习笔记

烂石

2025 年 6 月 14 日

1 绪论

1.1 引言

略

1.2 基本术语

(i) 样本/示例-sample/instance

(ii) 训练集-training data

(iii) 测试集-testing data

(iv) 标记-label

(v) 样例-example: 拥有 label 的 instance

(vi) 泛化-generalization

1.3 假设空间

1.3.1 科学推理

- 归纳–induction 从具体的事实归结出一般性规律
- 演绎–deduction 从基础原理推演出具体状况

1.3.2 归纳学习–inductive learning

- 广义归纳学习
- 狭义归纳学习–概念学习 eg: 布尔概念学习

1.3.3 版本空间–version space

即存在着一个与训练集一致的" 假设集合"

1.4 归纳偏好

有多个与训练集一致的假设, 但测试新样本时有不同的输出结果, 那么采用哪种模型(假设)?

1.4.1 ” 奥卡姆剃刀” 原则

若有多个假设与观察一致, 则选最简单的那个. 利用什么原则, 取决于算法能否获得更好的性能, 泛化能力是否更强

1.4.2 NFL(No Free Lunch Theorem) 定理–” 没有免费的午餐” 定理

定理 1.1 (No Free Lunch 定理)**.** 对于所有学习算法 $\mathcal{L}_a$ 和 $\mathcal{L}_b$，在均匀分布的目标函数空间下，它们的训练外误差满足：

$$\sum_f E_{ote}(\mathcal{L}_a|X,f) = \sum_f E_{ote}(\mathcal{L}_b|X,f)$$

证明.

步骤 1.1.1 (定义与假设)**.** 假设样本空间 $\mathcal{X}$ 和假设空间 $\mathcal{H}$ 是离散的。定义：

$$E_{ote}(\mathcal{L}_a|X,f) = \sum_{h\in\mathcal{H}} \sum_{x\in\mathcal{X}-X} P(x)\cdot \mathbb{I}(h(x)\neq f(x))\cdot P(h|X,\mathcal{L}_a)$$

其中 $\mathbb{I}(\cdot)$ 为指示函数。

步骤 1.1.2 (总误差求和)**.** 对所有目标函数求和：

$$\sum_f E_{ote}(\mathcal{L}_a|X,f) = \sum_f \sum_h \sum_{x\in\mathcal{X}-X} P(x)\mathbb{I}(h(x)\neq f(x))P(h|X,\mathcal{L}_a) \tag{1}$$

步骤 1.1.3 (交换求和顺序)**.** 将 $\sum_f$ 移至内部：

$$\begin{aligned} &= \sum_{x\in\mathcal{X}-X} P(x) \sum_h P(h|X,\mathcal{L}_a) \\ &\quad \times \underbrace{\sum_f \mathbb{I}(h(x)\neq f(x))}_{\boxed{\text{关键项}}} \end{aligned} \tag{2}$$

步骤 1.1.4 (计算关键项)**.** 对于二分类问题，每个 x 处的 $f(x)$ 有等概率取 *0* 或 *1*：

$$\sum_f \mathbb{I}(h(x)\neq f(x)) = \frac{1}{2}\cdot 2^{|\mathcal{X}|} = 2^{|\mathcal{X}|-1}$$

步骤 1.1.5 (最终化简)**.** 代入关键项并利用 $\sum_h P(h|X,\mathcal{L}_a) = 1$：

$$\text{原式} = 2^{|\mathcal{X}|-1} \sum_{x\in\mathcal{X}-X} P(x)$$

该结果与算法 $\mathcal{L}_a$ 无关，故对任意 $\mathcal{L}_a, \mathcal{L}_b$：

$$\sum_f E_{ote}(\mathcal{L}_a|X,f) = \sum_f E_{ote}(\mathcal{L}_b|X,f)$$

□

由1.1可知, 脱离具体问题, 空谈” 什么学习算法最好” 是毫无意义的

1.5 发展历程

推理期:1950s-1970s–符号知识, 演绎推理
知识期:1970s 中期–符号知识, 领域知识
学习期:1980s–机器学习, 归纳逻辑程序设计 (Inductive Logic Programming)
统计学习:1990s 中期–向量机 (Support Vector Machine), 核方法 (Kernel Methods)
深度学习:2000s–神经网络

1.6 应用现状

信息科学, 自然科学...

1.7 阅读材料

1.7.1 机器学习

国际会议:ICML,NIPS，COLT,ECML(Europe),ACML(Asia)
国际期刊:JMLR,ML
国内会议:CCML,MLA

1.7.2 人工智能

国际会议:IJCAI,AAAI
国际期刊:AI,JAIR

1.7.3 数据挖掘

国际会议:KDD,ICDM
国际期刊:ACM-TKDD,DMKD

1.7.4 计算机视觉

CVPR(会议),IEEE-TPAMI(期刊)

1.7.5 神经网络

期刊:NC,IEEE-TNNLS

1.7.6 统计学

期刊:AS

1.8 习题

1.8.1 表 1.1 中若只包含编号为 1 和 4 的两个样例，试给出相应的版本空间.

表 1.1 西瓜数据集

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否

图 1: 表 1.1 西瓜数据集

解:

表 1: 版本空间表

色泽	根蒂	敲声	逻辑表达式
青绿	蜷缩	浊响	(色泽 = 青绿) ∧ (根蒂 = 蜷缩) ∧ (敲声 = 浊响)
青绿	蜷缩	*	(色泽 = 青绿) ∧ (根蒂 = 蜷缩)
青绿	*	浊响	(色泽 = 青绿) ∧ (敲声 = 浊响)
*	蜷缩	浊响	(根蒂 = 蜷缩) ∧ (敲声 = 浊响)
青绿	*	*	(色泽 = 青绿)

1.8.2 估算共有多少种可能的假设

与使用单个合取式来进行假设表示相比，使用“析合范式”将使得假设空间具有更强的表示能力。例如

好瓜 ⇔ (色泽 = ⋆) ∧ (根蒂 = 蜷缩 ∧ 敲声 = ⋆)

∨ (色泽 = 乌黑 ∧ 根蒂 = ⋆)∧ 敲声 = 沉闷)，

会把“(色泽 = 青绿) (根蒂 = 蜷缩) (敲声 = 清脆)”以及“(色泽 = 乌黑) (根蒂 = 硬挺) (敲声 = 沉闷)”都分类为“好瓜”。若使用最多包含 k 个合取式的析合范式来表达表 1.1 西瓜分类问题的假设空间，试估算共有多少种可能的假设。

解: 色泽包含两种情况 (青绿, 乌黑), 三种选择 (青绿, 乌黑,*);
根蒂 (蜷缩, 稍蜷, 硬挺) 共 4 种选择;
敲声 (浊响/沉闷/清脆) 共 4 种选择;

除了不能存在 (*,*,*) 的组合, 共包含 $3 \times 4 \times 4 - 1 = 47$ 种组合.
题目求 k 个合取式的所有可能的组合之和, 则有

$$\boxed{\sum_{i}^{k} \binom{47}{i}} \tag{3}$$

1.8.3 若数据包含噪声，则假设空间中有可能不存在与所有训练样本都一致的假设。在此情形下，试设计一种归纳偏好用于假设选择。

解: 刚入门, 可能无法正确回答此问题, 但存在的方法应该有权重噪声注入/梯度稳定性惩罚/稀疏性偏好/集成鲁棒性.

1.8.4 试证明“没有免费的午餐定理”仍成立。

本章 1.4 节在论述“没有免费的午餐”定理时，默认使用了“分类错误率”作为性能度量来对分类器进行评估。若换用其他性能度量 ℓ，则式 (1.1) 将改为

$$E_{ote}(\mathcal{L}_a|X,f) = \sum_h \sum_{x\in\mathcal{X}-X} P(x)\ell(h(x),f(x))P(h|X,\mathcal{L}_a)$$

试证明“没有免费的午餐定理”仍成立。

证明.

步骤 **1.1.6.** 性能度量虽发生了改变, 目标函数仍然均匀分布, 总误差表达式为*:*

$$\begin{aligned} E_{ote}(\mathcal{L}_a|X,f) &= \sum_h \sum_{x\in\mathcal{X}-X} P(x)\ell(h(x),f(x))P(h|X,\mathcal{L}_a) \\ &= \sum_{x\in\mathcal{X}-X} P(x) \sum_h P(h|X,\mathcal{L}_a) \sum_f \ell(h(x),f(x)) \end{aligned}$$

步骤 **1.1.7.** 由*1.1*证明过程可知*:*$\sum_f f(x) = 2^{|\mathcal{X}|-1}$

$$\therefore \sum_f \ell(h(x),f(x)) = 2^{|\mathcal{X}|-1}\left[\ell(h(x),0)+\ell(h(x),1)\right]. \tag{4}$$

步骤 **1.1.8.**

$$\because P(h(x)=0|X,\mathcal{L}_a) = P(h(x)=1|X,\mathcal{L}_a) = \frac{1}{2}. \therefore \sum_h P(h|X,\mathcal{L}_a)\left[\ell(h(x),0)+\ell(h(x),1)\right] = \frac{1}{2}\left[\ell(0,0)+\ell \tag{5}$$

此结果与算法 $\mathcal{L}_a$ 无关。

$$\sum_{f} E_{\text{ote}}(\mathcal{L}_a|X, f) = 2^{|\mathcal{X}|-1} \cdot \frac{1}{2} \sum_{x \in \mathcal{X}-X} P(x) \left[\ell(0,0) + \ell(0,1) + \ell(1,0) + \ell(1,1)\right]. \tag{6}$$

□

无论选择何种性能度量 ℓ，只要真实目标函数 f 在所有可能的函数上均匀分布，无免费午餐定理仍然成立。算法的平均性能仅由数据分布和度量 ℓ 的对称性决定，而与算法本身的设计无关。

1.8.5 试述机器学习能在互联网搜索的哪些环节起什么作用。

机器学习贯穿搜索的全流程，从理解用户意图到动态优化结果，其核心价值在于通过数据驱动的方式提升搜索效率、准确性和个性化程度。知识库中提到的超参数优化（如随机搜索）、分布式表示（如协同过滤）等技术均为此提供了方法论支持。机器学习贯穿搜索的全流程，从理解用户意图到动态优化结果，其核心价值在于通过数据驱动的方式提升搜索效率、准确性和个性化程度。知识库中提到的超参数优化（如随机搜索）、分布式表示（如协同过滤）等技术均为此提供了方法论支持。

2 模型评估与选择

2.1 经验误差与拟合

1. 错误率:m 个样本中有 a 个样本分类错误, 则错误率 $E = a/m$.

2. 精度:$1 - a/m$

3. 误差: 学习器的实际预测输出与样本的真实输出之间的差异; 训练集上的叫训练误差/经验误差, 新样本上的叫泛化误差;

4. 过拟合 (更容易遇到的情况): 将个例的特殊性错误地视为样本的普遍性; 欠拟合: 没有学好.

过拟合无法避免 (N=NP 问题尚未证明), 只能缓解或减小其风险

5. 模型选择: 在多种学习算法中, 选择合适的算法中的合适的参数配置.

2.2 评估方法

1. 测试集测试学习器对新样本的判断力, 将测试误差近似于泛化误差. 测试集尽可能的不出现在训
师更希望学生有举一反三的能力)

2. 如果只有一个数据集, 既要训练又要测试, 如何解决?

2.2.1 留出法

直接将数据集 D 划分为两个互斥的集合, 其中一个集合作为训练集 S, 另一个作为测试集 T. 在 S 上训练出模型后用 T 来评估其测试误差.

1. 训练/测试集的划分要尽可能保持数据分布的一致性 (分类任务时, 至少要保持样本的类别比例相似) 采样角度上看, 则保留类别比例的采样方式通常称为分层采样

例如: 通过对 D 进行分层采样而获得含 70% 样本的训练集 S 和含 30% 样本的测试集 T, 若 D 包含 500 个正例、500 个反例，则分层采样得到的 S 应包含 350 个正例、350 个反例，而 T 则包含 150 个正例和 150 个反例；若 S、T 中样本类别比例差别很大，则误差估计将由于训练/测试数据分布的差异而产生偏差.

2. 单次使用留出法得到的估计结果往往不够稳定可靠，在使用留出法时，一般要采用若干次随机划分、重复进行实验评估后取平均值作为留出法的评估结果.

例如进行 100 次随机划分，每次产生一个训练/测试集用于实验评估，100 次后就得到 100 个结果，而留出法返回的则是这 100 个结果的平均.

3. 局限性: 若令训练集 S 包含绝大多数样本，则训练出的模型可能更接近于用 D 训练出的模型，但由于 T 比较小，评估结果可能不够稳定准确；若令测试集 T 多包含一些样本，则训练集 S 与 D 差别更大了，被评估的模型与用 D 训练出的模型相比可能有较大差别，从而降低了评估结果的保真性 (fidelity). 这个问题没有完美的解决方案，常见做法是将大约 2/3～4/5 的样本用于训练，剩余样本用于测试.

2.2.2 交叉验证法

将数据集分为 k 个相似的子集, 每一折交叉验证时,k-1 为训练集, 剩余的一个为测试集, 共 k 次不同的测试集, 为 k 折交叉验证法. 交叉验证法同样需要进行随机划分, 例如 10 次 10 折交叉验证法和 100 次留出法都需要进行 100 次训练/测试.

留一法 (LOO), 数据集共 m 个, 当 k=m 时, 为留一法. 训练出来的模型会更准确, 但计算开销较大.

2.2.3 自助法 (Bootstrapping)

有放回的重复随机抽样, 执行 m 次得到 m 个样本的数据集 D'. 其中, 某些样本始终不会被抽中的概率为 $\lim_{m->\infty}(1-\frac{1}{m})^m \mapsto \frac{1}{e} \approx 0.368$ 自助法适用于小样本.

2.2.4 小结

方法	数据划分方式	数据利用率	计算成本	适用场景
留出法	单次划分训练集/测试集	低	低	大数据快速验证
交叉验证	多次划分（k 个子集轮流验证）	高	中	中等数据模型调优
留一法	每次留 1 个样本验证	最高	高	小样本高精度评估
自助法	有放回重采样生成多样本集	灵活	中	小样本统计推断、不确定性估

选择建议

- 数据量大且需快速验证：留出法。
- 中等数据调参：k-折交叉验证（如 5 折或 10 折）。
- 小样本高精度需求：留一法（但需权衡计算成本）。
- 统计量分布估计或小样本：自助法。

2.2.5 调参与最终模型

调参对最终模型有关键性影响，通过验证集来评估模型的好坏。

2.3 性能度量

使用不同的性能度量会导致不同的评判结果。比如，回归任务最常用的性能度量是"均方误差"：

$$E(f;D) = \frac{1}{m}\sum_{i=1} m(f(x_i) - y_i)^2 \tag{7}$$

更一般的，对于数据分布 D 和概率密度函数 $p(\cdot)$, 均方误差可表示为：

$$E(f;D) = \int_{x\sim D} (f(x) - y)^2 p(x)dx \tag{8}$$

2.3.1 错误率与精度

对于样例集 D，分类错误率定义为：

$$E(f;D) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}(f(x_i) \neq y_i) \tag{9}$$

精度定义为：

$$\begin{aligned} \mathrm{acc}(f;D) &= 1 - E(f;D) \\ &= \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}(f(x_i) = y_i) \end{aligned} \tag{10}$$

更一般的，对于数据分布 D 和概率密度函数 $p(\cdot)$, 错误率可表示为：

$$E(f;D) = \int_{x\sim D} \mathbb{I}(f(x) \neq y)p(x)dx \tag{11}$$

精度可表示为：

$$\begin{aligned} \mathrm{acc}(f;D) &= 1 - E(f;D) \\ &= \int_{x\sim D} \mathbb{I}(f(x) = y)p(x)dx \end{aligned} \tag{12}$$

2.3.2 查准率、查全率与 F1 值

查准率（Precision）和查全率（Recall）是分类任务中常用的性能度量，尤其在处理不平衡数据集时非常重要。查准率亦称准确率，表示预测为正例的样本中实际为正例的比例：

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{13}$$

其中,TP 表示真正例（True Positives），即被正确预测为正例的样本数；FP 表示假正例（False Positives），即被错误预测为正例的样本数。查全率亦称召回率，表示实际为正例的样本中被正确预测为正例的比例：

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{14}$$

其中,TP 表示真正例，FN 表示假负例（False Negatives），即被错误预测为负例的正例样本数。查准率和查全率之间通常存在权衡关系：提高查准率可能会降低查全率，反之亦然。P-R 曲线（Precision-Recall Curve）可以帮助可视化这种权衡关系。如图2所示，P-R 曲线展示了不同阈值下查准率和查全率的变化情况。

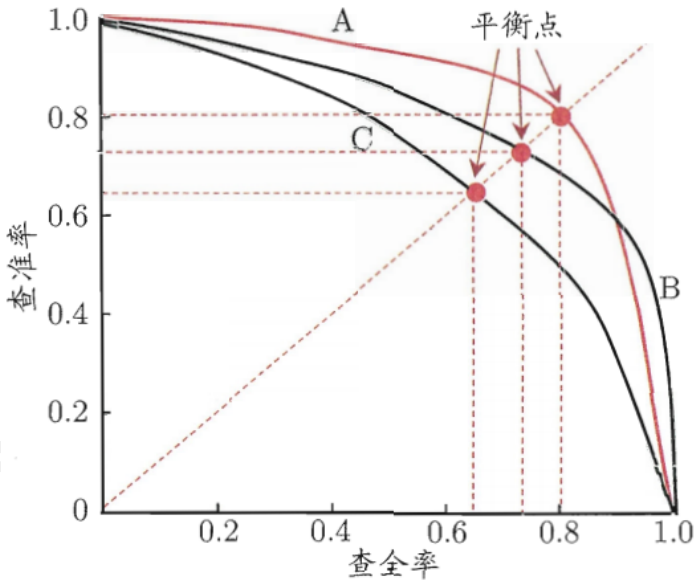

图 2: P-R 曲线示例

虽然 BEP 是查准率 = 查全率时的点，但在实际应用中，查准率和查全率通常不会相等，因此需要综合考虑两者的平衡。F1 值是查准率和查全率的调和平均数，常用来综合评估分类器的性能：

$$\begin{aligned} F1 &= \frac{2 \times P \times R}{P + R} \\ &= \frac{2 \times TP}{\text{样例总数} + TP - TN} \end{aligned} \tag{15}$$

但在实际应用中, 查准率和查全率的权衡关系通常需要根据具体任务和数据集来调整。所以 F1 的一般形式— F_β, 加权调和平均数能够根据任务需求调整查准率和查全率的权重：

$$F_\beta = \frac{(1 + \beta^2) \times P \times R}{\beta^2 \times P + R} \tag{16}$$

与算术平均数和几何平均数相比，调和平均数更重视较小值

对于有多个二分类混淆矩阵的多分类任务，可以使用宏平均（Macro-Averaging）和微平均（Micro-Averaging）来计算整体的查准率、查全率和 F1 值。宏平均是对每个类别

单独计算查准率和查全率，然后取平均值：

$$P_{macro} = \frac{1}{n}\sum_{i=1}^{n} P_i$$
$$R_{macro} = \frac{1}{n}\sum_{i=1}^{n} R_i$$
$$F1_{macro} = \frac{2 \times P_{macro} \times R_{macro}}{P_{macro} + R_{macro}} \tag{17}$$

微平均则是将所有类别的 TP、FP 和 FN 加总后计算查准率和查全率：

$$P_{micro} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i}$$
$$R_{micro} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FN_i}$$
$$F1_{micro} = \frac{2 \times P_{micro} \times R_{micro}}{P_{micro} + R_{micro}} \tag{18}$$

2.3.3 ROC 曲线与 AUC

ROC 曲线（Receiver Operating Characteristic Curve）是评估二分类模型性能的常用工具，通过绘制真正率（TPR）与假正率（FPR）的关系来展示分类器在不同阈值下的表现。真正率（TPR）也称为查全率（Recall），表示实际正例中被正确预测为正例的比例：

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{19}$$

假正率（FPR）表示实际负例中被错误预测为正例的比例：

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{20}$$

ROC 曲线通过改变分类阈值，计算不同阈值下的 TPR 和 FPR，从而绘制出一条曲线。理想情况下，ROC 曲线应尽可能接近左上角（TPR=1, FPR=0），表示模型能够正确识别所有正例且不误判负例。

AUC（Area Under the Curve）是 ROC 曲线下的面积，表示模型的整体性能。AUC 值介于 0 和 1 之间，值越大表示模型性能越好。AUC=0.5 表示模型没有区分能力，相当于随机猜测；AUC=1 表示完美分类器。

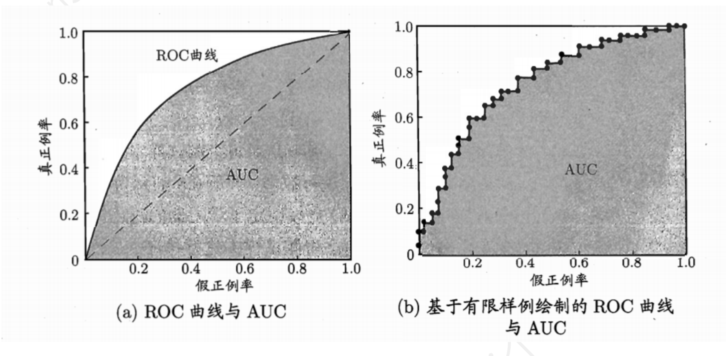

图 3: ROC 曲线示例

AUC 可以通过计算 ROC 曲线下的积分来获得，常用的计算方法包括梯形法则和蒙特卡洛积分等。估算公式为：

$$AUC = \frac{1}{2}\sum_{i=1}^{n-1}(FPR_{i+1} - FPR_i)(TPR_{i+1} + TPR_i) \tag{21}$$

形式上看,AUC 考虑的是样本预测的排序质量, 因此它与排序误差紧密相连, 给定 m^+ 个正例和 m^- 个反例, 令 D^+ 和 D^- 分别为正例和反例的样本集

则 loss 可以表示为:

$$\ell_{rank} = \frac{1}{m^+m^-}\sum_{x^+\in D^+}\sum_{x^-\in D^-}\left(\mathbb{I}(f(x^+) < f(x^-)) + \frac{1}{2}\mathbb{I}(f(x^+) = f(x^-))\right) \tag{22}$$

容易看出,ℓ_{rank} 对应的是 ROC 曲线上的面积, 故 AUC 可以表示为：

$$AUC = 1 - \ell_{rank} \tag{23}$$

2.3.4 代价敏感错误率与代价曲线

非均等代价: 在某些应用中, 不同类型的错误可能具有不同的代价, 例如在医疗诊断中, 漏诊 (假阴性) 可能比误诊 (假阳性) 更严重. 代价敏感错误率: 在这种情况下, 需要引入代价敏感错误率来衡量模型的性能, 定义为:

$$E_{f;D;cost} = \frac{1}{m}\Big(\sum_{x_i\in D^+}\mathbb{I}(f(x_i) \neq y_i)\cdot cost_{FP} + \sum_{x_i\in D^-}\mathbb{I}(f(x_i) \neq y_i)\cdot cost_{FN}\Big) \tag{24}$$

其中, $cost_{FP}$ 和 $cost_{FN}$ 分别表示假正例和假负例的代价. 代价曲线: 代价曲线是代价敏感错误率随分类阈值变化的图形表示, 可以帮助选择最优的分类阈值以最小化总代价. 其中, 横轴是取值为 [0,1] 的正例概率代价:

$$P(+)cost = \frac{p\cdot cost_{FP}}{p\cdot cost_{FP} + (1-p)\cdot cost_{FN}} \tag{25}$$

其中,p 为正例的概率, 纵轴是取值为 [0,1] 的归一化代价:

$$cost_{norm} = \frac{FNR\cdot p\cdot cost_{FP} + FPR\cdot(1-p)\cdot cost_{FN}}{p\cdot cost_{FP} + (1-p)\cdot cost_{FN}} \tag{26}$$

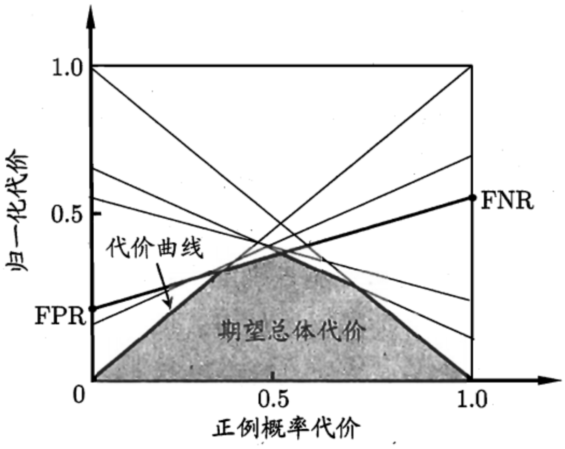

图 4: 代价曲线示例