机器学习：信息论(Information Theory)与决策树(Decision Tree)

~~Copyright: Jingmin Wei, Automation - Pattern Recognition and Intelligent System, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology~~

Copyright: Jingmin Wei, Computer Science - Artificial Intelligence, Department of Computer Science, Viterbi School of Engineering, University of Southern California

---

---

1. 信息论基础

信息论主要研究的是对一个信号能够提供信息的多少进行量化，最初用于研究在一个含有噪声的信道上用离散的字母表来发送消息，指导最优的通信编码等。

关于信息的一个基本想法：一个不太可能的事情竟然发生了，要比一个非常可能的时间的发生能提供更多的信息，也就是说导致那些"异常"事件发生的背后拥有着更多我们更想知道的东西。即概率越小的事情发生时，包含的信息量更大。

2. 自信息：一个时间所包含的信息量

如果一个事件发生的概率小，则其包含的信息量大。表示信息量是个减函数。

事件 x, y 满足相互独立($P(x,y)=P(x)P(y)$)，且 (x,y) 联合的信息量应该是 x, y 信息量之和($I(x,y)=I(x)+I(y)$)。所以对概率取对数，信息量可以定义为：

$$I(x)=-\log P(x)$$

在通信领域中，$\log$ 的底数通常以 2 为底(单位：比特)；在机器学习中，$\log$ 的底数通常以 e 为底(单位：奈特)。因此这里和后面的熵的定义都以 e 为底，即：

$$I(x)=-\ln P(x)$$

3. 信息熵($Information\ Entropy$)

熵用来表示当随机变量取多个不同值时，信息量的总体期望。

信息熵用来对概率分布的随机性程度进行度量，反映了一组数据所包含的信息量大小。

随机变量或各系统的熵越大，随机变量或系统的不确定性就越大，反之就越小。即描述的是有关事件 X 的所有可能结果的自信息期望值。

对于离散型随机变量：

$$H(p)=E_p[-\ln p(x)]=-\sum_{i=1}^{n} p_i \ln p_i$$

其中 n 代表事件 X 的所有 n 种可能的取值，p_i 代表了事件 X 为 i 时的概率且满足 $\sum_{i=1}^{n} p(x_i)=1$。

信息熵的定义：熵的作用计算损失用于调整梯度递减的步长，如果本次熵损失比上次上损失大，则说明步长太大了。用于决策树的熵越大，说明特征的划分数据能力越强。

3.1. 性质

当 X 服从均匀分布($x_i = \frac{1}{n}$)，熵有极大值 $\ln n$ ，当 X 中某一个变量取值的概率为 1 ，其他为 0 ，熵有极小值 0 ，即 $0 \leqslant H(p) \leqslant \ln n$ 。

这是一个带约束的优化问题，可通过拉格朗日乘子法，以及黑塞矩阵负定(凹函数)证明。

4. 微分熵(连续变量的信息熵，$Differential\ Entropy$)

对于连续型随机变量的信息熵可以定义为：

$$H(p) = -\int_{-\infty}^{+\infty} p(x) \ln p(x) \mathrm{d}x$$

此时的熵是一个泛函。

4.1. 性质

当随机变量 x 服从正态分布 $N(\mu, \sigma^2)$ 时，熵有极大值，即正态分布的熵： $\ln(\sqrt{2\pi}\sigma) + \frac{1}{2}$ 。

这是个带约束的泛函极值问题，可通过拉格朗日乘子法和欧拉-拉格朗日方程证明：

$$F[y] = \int_a^b L(x, y(x), y'(x)) \mathrm{d}x$$
$$\text{极值点满足：} \frac{\partial L}{\partial y} - \frac{\mathrm{d}}{\mathrm{d}x}(\frac{\partial L}{\partial y'}) = 0$$

5. 联合熵($Joint\ Entropy$)

联合熵用来度量二维 / 多维随机变量的不确定性。

对于随机变量 (X, Y) ，其联合分布为 $P(x_i, y_i)$ ，则联合熵为：

$$H(X, Y) = -\sum_i \sum_j P(x_i, y_j) \ln P(x_i, y_j)$$

性质：联合熵是非负的；如果二者相互独立，则满足 $H(X, Y) = H(X) + H(Y)$ 。

推广到多维：

$$H(X_1, \cdots, X_n) = -\sum_{x_1} \cdots \sum_{x_n} P(x_1, \cdots, x_n) \ln P(x_1, \cdots, x_n)$$

对于连续型随机变量：

$$H(X, Y) = -\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(x, y) \ln p(x, y) \mathrm{d}x\mathrm{d}y$$
$$H(x) = -\int_{-\mathbb{R}^n} p(x) \ln p(x) \mathrm{d}x$$

多维正态分布 $N(\mu, \Sigma)$ 的联合熵为： $\frac{n}{2} \ln 2\pi + \frac{1}{2}(|\Sigma|) + \frac{n}{2}$ ，这个式子表明其只与协方差有关。

6. 交叉熵($Cross\ Entropy$)

交叉熵的定义和熵是类似的，但是定义在两个概率分布上，主要用于衡量两个分布的相似度。

对于离散型概率分布 $p(x),q(x)$ ，定义交叉熵为：

$$H(p,q)=E_p[-\ln Q(X)]=-\sum_x P(X)\ln Q(X)$$

其值越大，两个概率分布的差异也就越大，反之越小。Lesson 3.5 参数估计(MLE, MAP, Bayes, KNN, Parzen, GMM, EM算法)我们学习到，如果 LR (逻辑回归)和 $Softmax$ 从最大似然的角度解释，即当给出了标签和训练集 (y_{label},X) ，求使得 $p(y|X,\theta)$ 最大的参数 θ 。表示既然这组样本出现了，那么它们出现的概率理应是最大化的。如果 LR 和 $Softmax$ 从交叉熵的角度解释，就是最小化 y_{pred} 和 y_{label} 之间的分布差异。最大似然和最小化交叉熵，其实在 LR 算法中，是一个意思。

对于连续型随机变量：

$$H(p,q)=E_p[-\ln Q(x)]=-\int_{-\infty}^{+\infty}p(x)\ln q(x)\mathrm{d}x$$

如果两个概率分布相同，则交叉熵退化为熵，即 $H(p,q)=H(p)=H(q)$ 。

6.1. 性质与应用

交叉熵不具有对称性，不是距离度量，也不满足三角不等式。

当两个的概率分布相等时，交叉熵有极小值。可通过拉格朗日乘子和黑塞矩阵正定(凸函数)来证明。

题外话： $Logistic,Softmax$ 回归的目标函数，就是求这个损失的极小值，即让 y_{pred},y_{label} 之间的差异尽可能小。

交叉熵被应用于 $Logistic$ 回归和 $Softmax$ 回归问题(Lesson 5 监督学习之分类(Logistic, Bayes, MAP))。

7. 相对熵 $Relative\ Entropy$ ($Kullback-Leibler$ 散度)

相对熵的定义由熵和交叉熵共同决定，它和交叉熵类似，主要也用来衡量两个分布的相似度。

相对熵又称为 KL 散度，信息散度，信息增益，常用在对抗神经网络里。

在很多算法中，假设连续随机变量 x ，其概率分布为 $p(x)$ ，模型得到的近似分布为 $q(x)$ 。二者的相对熵越大，两个概率分布的差异也就越大。我们的目标就是最小化这个相对熵。

对于离散型概率分布 p,q ，其 KL 散度为：

$$\begin{aligned}D_{KL}(p||q)&=-\sum_{i=1}^n p(x_i)\log q(x_i)-(-\sum_{i=1}^n p(x_i)\log p(x_i))\\&=\sum_{i=1}^n p(x_i)\log\frac{p(x_i)}{q(x_i)}\end{aligned}$$

两个伯努利分布的 $D_{KL}(p||q)=p_1\ln\frac{p_1}{p_2}+(1-p_1)\ln\frac{1-p_1}{1-p_2}$ 。

对于连续型概率分布 p,q ，其 KL 散度为：

$$D_{KL}(p||q)=-\int_{-\infty}^{+\infty}p(x)\ln\frac{p(x)}{q(x)}\mathrm{d}x$$

两个正态分布的 $D_{KL}(p||q)=\ln\frac{\sigma_1}{\sigma_2}\int_{-\infty}^{+\infty}p_1(x)\mathrm{d}x+\int_{-\infty}^{+\infty}\frac{(x-\mu_2)^2}{2\sigma_2^2}p_1(x)\mathrm{d}x-\int_{-\infty}^{+\infty}\frac{(x-\mu_1)^2}{2\sigma_1^2}p_1(x)\mathrm{d}x$ 。

根据正态分布方差和数学期望计算公式，可以简化为： $D_{KL}(p||q)=\frac{1}{2}(\ln\frac{\sigma_2^2}{\sigma_1^2}+\frac{\sigma_1^2}{\sigma_2^2}+\frac{\mu_1-\mu_2}{\sigma_2^2}-1)$ 。

如果第一个正态分布各变量独立(协方差矩阵为对角阵)，第二个正态分布是标准正态 $N(0,I)$ ，则二者的 KL 散度为： $\frac{1}{2}\sum_{i=1}^{d}(\sigma_i^2+\mu_i^2-\ln\sigma_i^2-1)$ 。

对于多维正态分布的 $D_{KL}(p||q)=\frac{1}{2}(\ln\frac{|\Sigma_2|}{|\Sigma_1|}-d+tr(\Sigma_2^{-1}\Sigma_1)+(\mu_2-\mu_1)^T\Sigma_2^{-1}(\mu_2-\mu_1))$ 。

7.1. 性质(和熵的相互关系)

满足 $Gibbs$ 不等式，即 $D_{KL}(p||q)\geqslant 0$ ，当且仅当 $p(x)=q(x)$ ， KL 散度取最小值 0 。

KL 散度不具有对称性，不是距离度量，也不满足三角不等式。

KL 散度和交叉熵一样，也反映了两个概率分布之间的差异程度。其定义公式可推导为交叉熵和熵的差：

$$D_{KL}(p||q)=H(p,q)-H(p)$$

如果某机器学习算法需要以 $p(x)$ 为目标，以 $q(x)$ 作为拟合函数，此时 $H(p)$ 是不变的，只需要计算 $H(p,q)$ 即可，这也从 KL 散度角度说明了逻辑回归(交叉熵)本身的正确性。

8. $Jensen-Shannon$ 散度

JS 散度根据 KL 散度来构造，也用来衡量两个分布的相似度。不同的是，它具有对称性。

$$D_{JS}(p||q)=\frac{1}{2}D_{KL}(p||m)+\frac{1}{2}D_{KL}(q||m)$$

其中概率分布 m 为 p,q 的平均值：

$$m(x)=\frac{1}{2}(p(x)+q(x))$$

8.1. 性质与应用

JS 散度其实是根据 KL 散度的均值构造，$D_{JS}(q||p)\geqslant 0$，且具有对称性：

$$D_{JS}(q||p)=D_{JS}(p||q)$$

当且仅当 $m(x)=p(x)=q(x)$ 时，有最小值 $D_{JS}(p||q)=0$ 。和 KL 散度一样，JS 散度越大，两个概率分布之间的差异也就越大。

应用：KL,JS 散度常被用于流型学习(Lesson 8 无监督学习(聚类, 信号分解, 流形降维))，变分推断(Lesson 3.5 参数估计(MLE, MAP, Bayes, KNN, Parzen, GMM, EM算法))，以及生成对抗网络。

9. 互信息($Mutual\ Information$)

互信息用来衡量两个相同的一维分布变量之间的独立性，或者说相关性和依赖程度。

$I(X,Y)$ 是衡量联合分布 $p(x,y)$ 和 $p(x)p(y)$ 分布之间的关系，即他们之间的相关系数。两个随机变量的依赖程度越高，则互信息值越大，反之越小。

设两个随机变量 (X,Y) 的联合分布为 $p(x,y)$，边际分布分别为 $p(x),p(y)$ ，则互信息定义为：

$$
\begin{aligned}
I(X,Y) &= \sum_{x\in X}\sum_{y\in Y} p(x,y)\ln\frac{p(x,y)}{p(x)p(y)} \\
&= D_{KL}(p(x,y)||p(x)p(y))
\end{aligned}
$$

9.1. 性质(和熵的相互关系)

$I(X,Y)\geqslant 0$ ，当且仅当两个事件独立时，取最小值 0 ；如果两个变量之间相互独立 $p(x,y)=p(x)p(y)$，则 $I(X,Y)=0$ 。

相互关系和结果推导：

$$
\begin{aligned}
I(X,Y) &= H(X)-H(X|Y) \\
&= H(X)+H(Y)-H(X,Y) \\
&= \sum_x p(x)\ln\frac{1}{p(x)}+\sum_y p(y)\ln\frac{1}{p(y)}-\sum_{x,y} p(x,y)\ln\frac{1}{p(x,y)} \\
&= \sum_{x,y} p(x,y)\ln\frac{p(x,y)}{p(x)p(y)}
\end{aligned}
$$

即两个变量的联合熵等于它们各自的熵之和减去互信息：

$$
H(X,Y)=H(X)+H(Y)-I(X,Y)
$$

这一结论类比于两个集合的并集。根据互信息定义，也可以推导出 $I(X,Y)\leqslant H(X)$ ， $I(X,Y)\leqslant H(Y)$ 。

当两个变量相互独立时，有：

$$
H(X,Y)=H(X)+H(Y)
$$

9.2. 应用：特征选择

互信息常用语特征选择，如果 Y 为标签，X 为数据，则二者的互信息反映了类别和标签的相关程度。在做分类前，可以先进行特征选择，选取一部分互信息最大的特征列，排除其他列，最终形成最后用于训练的特征向量。

特征选择的合理使用，可以极大地提高模型的泛化能力。

10. 条件熵($Conditional\ Entropy$)

条件熵用于衡量，已知一个随机变量的取值条件下，另一个随机变量的信息量。

或者表示为 X 给定条件下，Y 的条件概率分布的熵对 X 的数学期望(平均不确定性)。

对于随机变量 Y ，在 X 的条件下其条件熵为：

$$\begin{aligned} H(Y|X) &= \sum_{i=1}^{n} P(X=x_i) H(Y|X=x_i) \\ &= -\sum_i \sum_j P(x_i, y_j) \ln P(y_j|x_i) \\ &= -\sum_i \sum_j P(x_i, y_j) \ln \frac{P(x_i, y_j)}{P(x_i)} \end{aligned}$$

相比于联合熵，条件熵只多了分母一项。

10.1. 条件微分熵

即连续变量的条件熵：

$$H(Y|X) = -\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p(x,y) \ln \frac{p(x,y)}{p(x)} \mathrm{d}x \mathrm{d}y$$

10.2. 性质与应用(和熵的相互关系)

$H(Y|X) \geqslant 0$ ，当且仅当 Y 完全由 X 确定时，值为 0 。

当且仅当这两个随机变量相互独立时，$H(Y|X) = H(Y)$ 。

相互关系和结果推导：

$$\begin{aligned} H(X,Y) &= H(X) + H(Y|X) = H(Y) + H(X|Y) \\ H(X,Y) &= -\sum_i \sum_j P(x_i, y_j) \ln P(x_i, y_j) \\ &= -\sum_i \sum_j P(x_i, y_j) \ln P(y_j|x_i) + (-\sum_i \big(\sum_j P(x_i, y_j)\big) \ln P(x_i)) \\ &= H(Y|X) + H(X) \end{aligned}$$

即：

$$H(Y|X) = H(X,Y) - H(X)$$

X 对 Y 的条件熵是它们的联合熵和 $H(X)$ 的差值。可推导 $H(X,Y) \geqslant \max(H(X), H(Y))$ 。

又 $\because H(X,Y) = H(X) + H(Y) - I(X,Y)$ ，可以得到：

$$I(X,Y) = H(X) - H(X|Y)$$

即互信息等于熵和条件熵的差。并且能推导 $H(X) \geqslant H(X|Y)$ 。

10.3. 应用：信息增益($Information\ Gain$)

假设系统原有的熵为 $H(Y)$ ，后来引入了特征 T ，在固定特征 T 的情况下，系统的混乱度减小，熵减小为 $H(Y|T)$ ，那么特征 T 给系统带来的信息增益为：

$$IG(T) = H(Y) - H(Y|T)$$

其意义可以看成，决策树左右子集划分后，信息熵的下降值。信息增益为决策树算法的构造基础。

11. 决策树概述

决策树从父节点往子节点挨个分类。

决策树在分类问题中，表示基于特征对实例空间进行划分的方法，可以视为 $if-then$ 规则的集合，也可以认为是定义在特征空间和类空间上的条件概率分布。

步骤：

- 特征选择
- 决策树生成
- 决策树剪枝(防止过拟合)

12. 决策树特征选择

作用：决定选取哪些特征来划分特征空间。

在进一步讨论特征选择之前，首先需要根据之前信息论的知识，定义一个概念：信息增益。

$$\begin{aligned} \text{信息增益} &= \text{信息熵} - \text{条件熵} \\ &= H(D) - H(D|A) \end{aligned}$$

信息熵表示随机变量的不确定性。

条件熵表示给定一个特征属性的情况下，随机变量的不确定性。

随机事件： P 。

信息量： $\log \frac{1}{P} = -\log P$ 。

12.1. 熵和决策时分类原则

信息量对于 $P(X)$ 的期望。熵用来对概率分布的随机性程度进行度量，反映了一组数据所包含的信息量大小。

在决策树的生成中，各类样本出现的概率服从一个概率分布，如果熵越小，说明内部的样本越纯(即树的子集都为其中的某一类或者某几类样本)。换句话说，在信息熵中我们知道，当所有的样本都只属于某一类时，熵有极小值；当样本均匀地分布在所有类别中，熵有极大值。

这也是 $ID3$ 决策树的分类规则。

$$H(X) = E_{P(X)}[\log P]$$

复习信息论的内容：

设 X 是一个有限的离散随机变量，其概率分布如下：

$$P(X=x_i)=p_i, \quad i=1,2,\cdots,n$$

熵的离散型：

$$\begin{gathered}H(x)=-\sum_x P(x)\ln(x)\\ D=\{(x_i,y_i)\}_{i=1}^m\\ H(D)\end{gathered}$$

用积分来表示连续型：

$$\int -P(x)\ln(P(x))\mathrm{d}x$$

交叉熵：当随机变量只取两个值，例如 $1,0$ 时：

$$\begin{aligned}&P(X=1)=p\\ &P(X=0)=1-p\\ &\text{熵：}H(X)=-p\ln p-(1-p)\ln(1-p)\end{aligned}$$

条件熵：设有随机变量 (X,Y) ，其联合概率分布为:

$$P(X=x_i,Y=y_i)=p_{ij}, \quad i=1,2,\cdots,n;\ j=1,2,\cdots,m$$

定义条件熵 $H(Y|X)$ ，其表示为 X 已知的情况下 Y 的不确定性：

$$\begin{gathered}H(Y|X)=\sum_{i=1}^n P(X=x_i)H(y|X=x_i)\\ H(y|X=x_i)\text{表示在}X=x_i\text{的情况下}Y\text{的条件熵}\end{gathered}$$

定义信息增益：特征 A 对训练数据集 D 的信息增益定义为：(其表示为在 A 已知的情况下 D 的不确定性的减少量)

$$g(D|A)=H(D)-H(D|A)$$

12.2. 特征选择算法(决策树训练)

利用信息增益进行特征选择：选取信息增益最大的特征来作为决策树的父节点，也就是说，有无该特征对数据集的影响最大。

$$\arg\max_{A_i} g(D|A_i)=H(D)-H(D|A_i)$$

训练数据集 $|D|$ 表示样本容量，设有 K 个类 $C_k,k=1,\cdots,K$ ，特征 A_i 设有 n_i 个不同取值(其为离散的特征变量)，则根据 A_i 的不同将 D 划分为 n_i 个子集，并记 $|D_{ik}|=|D_i\cap C_k|$ 。

特征的信息增益算法流程如下：

1. 计算数据集 D 的熵 $H(D)$ ：

$$H(D)=-\sum_{k=1}^{K}\frac{|C_k|}{|D|}\log\frac{|C_k|}{|D|}$$

2. 计算特征 A_i 对于数据集 D 的条件熵 $H(D|A_i)$：

$$H(D|A_i)=\sum_{i=1}^{n_i}\frac{|D_i|}{|D|}H(D_i)=-\sum_{i=1}^{n_i}\frac{|D_i|}{|D|}\sum_{k=1}^{K}\frac{|D_{ik}|}{|D|}\log\frac{|D_{ik}|}{|D|}$$

3. 计算 A_i 的信息增益：

$$g(D|A_i)=H(D)-H(D|A_i)$$

即根据 $1,2$ 两个公式，计算信息增益，并选择信息增益最大的特征。

$$\arg\max_{A_i}=g(D|A_i)=H(D)-H(D|A_i)$$

13. 决策树生成

基于前文的特征选择算法，可以构造两种经典树：$ID3,\ C4.5$ 。

13.1. $ID3$

主要思想是：对于每个分裂规则，用 $ID3$ 分为左右子集 D_L,D_R ，并计算熵 $H(D_L),H(D_R)$ 。如果能找到一个判定规则，让二者的熵最小化，则它就能使分裂后的左右子集的纯度最大化。

这个判定规则就是前文所说的信息增益：

$$g(D|A_i)=H(D)-H(D|A_i)$$

信息增益的意义可以看成，决策树左右划分后熵的下降值。信息增益越大，表明熵下降的最多，也就表明划分之后的子集更纯。因此 $ID3$ 主要就是根据熵计算信息增益，并极大化这个增益。

基于信息增益特征选择的 $ID3$ 流程如下：

- 从根节点的全量数据开始，计算各特征的的信息增益。
- 选取特征信息增益最大的构建分支，以特征类型将数据分割为各子数据集，并去除其使用的特征。
- 在分割的子数据集和子节点上，重复调用前两步，直到信息增益小于给定阈值或者数据无特征为止，将其标签的众数作为类标签。

13.2. $C4.5$

$C4.5$ 算法即为将 $ID3$ 中的特征选择方式由信息增益替换为信息增益比。

特征的信息增益比算法：

1. 计算数据集 D 关于 A_i 的熵 $H_{A_i}(D)$：

$$H_{A_i}(D)=-\sum_{i=1}^{n_i}\frac{|D_i|}{|D|}\log\frac{|D_i|}{|D|}$$

2. 计算 A_i 的信息增益比(规避离散化太强的特征作为主结点的可能)：

$$g_R(D|A_i) = \frac{g(D|A_i)}{H_{A_i}(D)} = \frac{\text{信息增益}}{\text{切分信息}}$$

14. 决策树剪枝(预防过拟合)

树的规模越大，在模型训练中的拟合效果虽然会更好，但模型的泛化能力会下降。

实现方式：极小化决策树整体的损失函数或者代价函数。

函数定义：设树 T 的叶子节点数为 $|T|$ ，t 是数 T 的叶子节点，记该叶子节点有 N_t 个样本点，其中 k 类的样本点有 N_tk 个，则定义损失函数为：

$$C_\alpha(T) = \sum_{i=1}^{|T|} N_t H_t(T) + \alpha|T|,\ \text{其中} H_t(T) = -\sum_k \frac{N_{ik}}{N_t} \log \frac{N_{tk}}{N_t}$$

$$\alpha|T| \text{类似于回归的惩罚项}$$

剪枝流程：

1. 递归的从树的叶子节点回溯。
2. 计算并比较其损失函数，若 $C_\alpha(T)$ 在剪枝后更小则剪枝。
3. 返回第一步，直到根节点。

15. 分类与回归树 - $CART$

$CART$ 的假设条件：假设决策树是二叉树形式(即一次的特征只能将数据集分为两个类别)

相对于 $ID3$、$C4.5$ ， $CART$ 可以对于连续型变量进行分类和回归，但每个特征只能对数据集进行二分类。

15.1. 回归树原理

$$D = (x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)$$

$$\text{其中例如：} x_1 = (x_1^{(1)}, x_1^{(2)}, \cdots, x_1^{(N)})$$

假设已经将输入空间划分为 M 个单元 $R_1, R_2, \cdots, R_M$ ，并在 R_i 上固定一个输出值 c_i ：

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

对于回归树，误差函数可以形象化定义为：

$$\sum_{x_i \in R_m} (y_i - f(x_i))^2$$

由此产生的几个问题：

- 怎样对于空间划分是最好的?
- 划分空间 R_i 上固定的 c_i 值为多少最好?

15.2. 划分空间 R_i 上固定的 c_i 值为多少最好?

由误差函数可知：

$$\hat{c}_m = averge(y_i|x_i \in R_m)$$

时是最佳的。(类似于总长一定，正方形面积最小，而这个问题可以用概率期望解释)

15.3. 怎样对于空间划分是最好的?

在思考这个问题之前要思考另一个问题：假设我们用第 j 个特征进行切分，我们怎样选取切分点(明确特征为连续型特征，eg: 用一条鱼的长度分大鱼和小鱼)，不可用特征对于结果进行直接划分。对此我们需要找到一个切分点构建一个映射关系。(对于上例来说就是找到一个标准长度，大于这个长度将其视为大鱼，小于等于这个长度将其视为小鱼)

概念化说明：

假设选择第 j 个变量 $x^{(j)}$ 作为切分变量，s 是其切分点，则输入空间可以划分为：

$$R_1(j,s) = \{x|x^{(j)} \leqslant s\} \quad 和 \quad R_2(j,s) = \{x|x^{(j)} > s\}$$

而目标函数可定义为：

$$\min_{j,s}\Big[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2\Big]$$

对于固定的 j ：

$$\hat{c}_1 = ave\{y_i|x_i \in R_1(j,s)\} \quad 和 \quad \hat{c}_2 = ave\{y_i|x_i \in R_2(j,s)\}$$

遍历所有的输入变量，找到最优的切分变量 j 。

15.4. 回归树算法流程

1. 求解：

$$\min_{j,s}\Big[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2\Big]$$

2. 对于选定的 (j,s) 划分区域和决定其输出：

$$R_1(j,s) = \{x|x^{(j)} \leqslant s\} \quad 和 \quad R_2(j,s) = \{x|x^{(j)} > s\}$$
$$\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m(j,s)} y_i$$

3. 对于划分的子区域重返第一步。

求解出生成的回归决策树：

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

16. 决策树优缺点

16.1. 优点

- 不需要任何领域知识或者参数假设。
- 适合高维数据。
- 简单易于理解。
- 短时间内处理大量数据，得到可行且效果较好的结果

16.2. 缺点

- 对于各类别样本数量不一致的数据，信息增益偏向于那些具有更多数值的特征。
- 易于过拟合，特别是特征多的情况下，容易引入噪声特征。
- 忽略属性之间的相关性。
- 不支持在线学习。