知乎用户

853 人赞同了该回答

熵的本质是香农信息量($log\frac{1}{p}$)的期望。

现有关于样本集的2个概率分布p和q，其中p为真实分布，q非真实分布。按照真实分布p来衡量识别一个样本的所需要的编码长度的期望(即平均编码长度)为：H(p)= $\sum_i p(i) * log\frac{1}{p(i)}$。如果使用错误分布q来表示来自真实分布p的平均编码长度，则应该是：H(p,q)= $\sum_i p(i) * log\frac{1}{q(i)}$。因为用q来编码的样本来自分布p，所以期望H(p,q)中概率是p(i)。H(p,q)我们称之为“交叉熵”。

比如含有4个字母(A,B,C,D)的数据集中，真实分布p=(1/2, 1/2, 0, 0)，即A和B出现的概率均为1/2，C和D出现的概率都为0。计算H(p)为1，即只需要1位编码即可识别A和B。如果使用分布Q=(1/4, 1/4, 1/4, 1/4)来编码则得到H(p,q)=2，即需要2位编码来识别A和B(当然还有C和D，尽管C和D并不会出现，因为真实分布p中C和D出现的概率为0，这里就钦定概率为0的事件不会发生啦)。

可以看到上例中根据非真实分布q得到的平均编码长度H(p,q)大于根据真实分布p得到的平均编码长度H(p)。事实上，根据Gibbs' inequality可知，H(p,q)>=H(p)恒成立，当q为真实分布p时取等号。我们将由q得到的平均编码长度比由p得到的平均编码长度多出的bit数称为“相对熵”：D(p||q)=H(p,q)-H(p)= $\sum_i p(i) * log\frac{p(i)}{q(i)}$，其又被称为KL散度(Kullback–Leibler divergence，KLD) Kullback–Leibler divergence。它表示2个函数或概率分布的差异性：差异越大则相对熵越大，差异越小则相对熵越小，特别地，若2者相同则熵为0。注意，KL散度的非对称性。

比如TD-IDF算法就可以理解为相对熵的应用：词频在整个语料库的分布与词频在具体文档中分布之间的差异性。

交叉熵可在神经网络(机器学习)中作为损失函数，p表示真实标记的分布，q则为训练后的模型的预测标记分布，交叉熵损失函数可以衡量p与q的相似性。交叉熵作为损失函数还有一个好处是使用sigmoid函数在梯度下降时能避免均方误差损失函数学习速率降低的问题，因为学习速率可以被输出的误差所控制。

PS：通常“相对熵”也可称为“交叉熵”，因为真实分布p是固定的，D(p||q)由H(p,q)决定。当然也有特殊情况，彼时2者须区别对待。

编辑于 2016-07-01

CyberRep

856 人赞同了该回答

讨论这个问题需要从香农的信息熵开始。

小明在学校玩王者荣耀被发现了，爸爸被叫去开家长会，心里悲屈的很，就想法子惩罚小明。到家后，爸爸跟小明说：既然你犯错了，就要接受惩罚，但惩罚的程度就看你聪不聪明了。这样吧，我们俩玩猜球游戏，我拿一个球，你猜球的颜色，你每猜一次，不管对错，你就一个星期不能玩王者荣耀，当然，猜对，游戏停止，否则继续猜。**当然，当答案只剩下两种选择时，此次猜测结束后，无论猜对猜错都能100%确定答案，无需再猜一次，此时游戏停止（因为好多人对策略 1 的结果有疑问，所以请注意这个条件）。**

题目1：爸爸拿来一个箱子，跟小明说：里面有橙、紫、蓝及青四种颜色的小球任意个，各颜色小球的占比不清楚，现在我从中拿出一个小球，你猜我手中的小球是什么颜色？

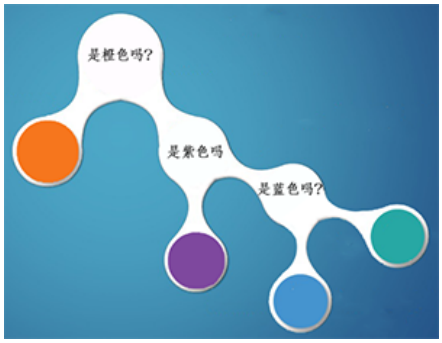
为了使被罚时间最短，小明发挥出最强王者的智商，瞬间就想到了以最小的代价猜出答案，简称策略1，小明的想法是这样的。

在这种情况下，小明什么信息都不知道，只能认为四种颜色的小球出现的概率是一样的。所以，根据策略1，1/4概率是橙色球，小明需要猜两次，1/4是紫色球，小明需要猜两次，其余的小球类似，所以小明预期的猜球次数为：

H = 1/4 * 2 + 1/4 * 2 + 1/4 * 2 + 1/4 * 2 = 2

题目2：爸爸还是拿来一个箱子，跟小明说：箱子里面有小球任意个，但其中1/2是橙色球，1/4是紫色球，1/8是蓝色球及1/8是青色球。我从中拿出一个球，你猜我手中的球是什么颜色的?

小明毕竟是最强王者，仍然很快得想到了答案，简称策略2，他的答案是这样的。



在这种情况下，小明知道了每种颜色小球的比例，比如橙色占比二分之一，如果我猜橙色，很有可能第一次就猜中了。所以，根据策略2，1/2的概率是橙色球，小明需要猜一次，1/4的概率是紫色球，小明需要猜两次，1/8的概率是蓝色球，小明需要猜三次，1/8的概率是青色球，小明需要猜三次，所以小明预期的猜题次数为：

H = 1/2 * 1 + 1/4 * 2 + 1/8 * 3 + 1/8 * 3= 1.75

题目3：其实，爸爸只想让小明意识到自己的错误，并不是真的想罚他，所以拿来一个箱子，跟小明说：里面的球都是橙色，现在我从中拿出一个，你猜我手中的球是什么颜色?

最强王者怎么可能不知道，肯定是橙色，小明需要猜0次。

上面三个题目表现出这样一种现象：针对特定概率为p的小球，需要猜球的次数 = $\log_2 \frac{1}{p}$，例如题目2中，1/4是紫色球，$\log_2 4$ = 2次，1/8是蓝色球，$\log_2 8$ = 3次。那么，针对整个整体，预期的猜题次数为：$\sum_{k=1}^{N} p_k \log_2 \frac{1}{p_k}$，这就是**信息熵**，上面三个题目的预期猜球次数都是由这个公式计算而来，第一题的信息熵为2，第二题的信息熵为1.75，最三题的信息熵为1 * $\log 1$ = 0 。那么信息熵代表着什么含义呢?

信息熵代表的是随机变量或整个系统的不确定性，熵越大，随机变量或系统的不确定性就越大。上面题目1的熵 > 题目2的熵 > 题目3的熵。在题目1中，小明对整个系统一无所知，只能假设所有的情况出现的概率都是均等的，此时的熵是最大的。题目2中，小明知道了橙色小球出现的概率是1/2及其他小球各自出现的概率，说明小明对这个系统有一定的了解，所以系统的不确定性自然会降低，所以熵小于2。题目3中，小明已经知道箱子中肯定是橙色球，爸爸手中的球肯定是橙色的，因而整个系统的不确定性为0，也就是熵为0。所以，在什么都不知道的情况下，熵会最大，针对上面的题目1~~题目3，这个最大值是2，除此之外，其余的任何一种情况，熵都会比2小。

所以，每一个系统都会有一个真实的概率分布，也叫真实分布，题目1的真实分布为（1/4，1/4，1/4，1/4），题目2的真实分布为（1/2，1/4，1/8，1/8），而**根据真实分布，我们能够找到一个最优策略，以最小的代价消除系统的不确定性，而这个代价大小就是信息熵，记住，信息熵衡量了系统的不确定性，而我们要消除这个不确定性，所要付出的【最小努力】（猜题次数、编码长度等）的大小就是信息熵。**具体来讲，题目1只需要猜两次就能确定任何一个小球的颜色，题目2只需要猜测1.75次就能确定任何一个小球的颜色。

现在回到题目2，假设小明只是钻石段位而已，智商没王者那么高，他使用了策略1，即

爸爸已经告诉小明这些小球的真实分布是（1/2，1/4, 1/8，1/8），但小明所选择的策略却认为所有的小球出现的概率相同，相当于忽略了爸爸告诉小明关于箱子中各小球的真实分布，而仍旧认为所有小球出现的概率是一样的，认为小球的分布为（1/4，1/4，1/4，1/4），这个分布就是**非真实分布**。此时，小明猜中任何一种颜色的小球都需要猜两次，即1/2 * 2 + 1/4 * 2 + 1/8 * 2 + 1/8 * 2 = 2。

很明显，针对题目2，使用策略1是一个坏的选择，因为需要猜题的次数增加了，从1.75变成了2，小明少玩了1.75的王者荣耀呢。因此，当我们知道根据系统的真实分布制定最优策略去消除系统的不确定性时，我们所付出的努力是最小的，但并不是每个人都和最强王者一样聪明，我们也许会使用其他的策略（非真实分布）去消除系统的不确定性，就好比如我将策略1用于题目2（原来这就是我在白银的原因），那么，当我们使用非最优策略消除系统的不确定性，所需要付出的努力的大小我们该如何去衡量呢？

这就需要引入**交叉熵，其用来衡量在给定的真实分布下，使用非真实分布所指定的策略消除系统的不确定性所需要付出的努力的大小。**

正式的讲，交叉熵的公式为：$\sum_{k=1}^{N} p_k \log_2 \frac{1}{q_k}$，其中 p_k 表示真实分布，q_k 表示非真实分布。例如上面所讲的将策略1用于题目2，真实分布 $p_k = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$，非真实分布 $q_k = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$，交叉熵为 $\frac{1}{2} * \log_2 4 + \frac{1}{4} * \log_2 4 + \frac{1}{8} * \log_2 4 + \frac{1}{8} * \log_2 4 = 2$，比最优策略的1.75来得大。

因此，交叉熵越低，这个策略就越好，最低的交叉熵也就是使用了真实分布所计算出来的信息熵，因为此时 $p_k = q_k$，交叉熵 = 信息熵。这也是为什么在机器学习中的分类算法中，我们总是最小化交叉熵，因为交叉熵越低，就证明由算法所产生的策略最接近最优策略，也间接证明我们算法所算出的非真实分布越接近真实分布。

最后，我们如何去衡量不同策略之间的差异呢？这就需要用到**相对熵，其用来衡量两个取值为正的函数或概率分布之间的差异**，即：

KL(f(x) || g(x)) = $\sum_{x \in X} f(x) * \log_2 \frac{f(x)}{g(x)}$

现在，假设我们想知道某个策略和最优策略之间的差异，我们就可以用相对熵来衡量这两者之间的差异。即，相对熵 = 某个策略的交叉熵 - 信息熵（根据系统真实分布计算而得的信息熵，为最优策略），公式如下：

KL（p || q） = H（p，q） - H（p） = $\sum_{k=1}^{N} p_k \log_2 \frac{1}{q_k} - \sum_{k=1}^{N} p_k \log_2 \frac{1}{p_k} = \sum_{k=1}^{N} p_k \log_2 \frac{p_k}{q_k}$

所以将策略1用于题目2，所产生的相对熵为2 - 1.75 = 0.25.

参考：

《数学之美》吴军

Information entropy

还有，小明同学，我帮你分析得这么清楚，快带我上王者。

Agenter
图像视觉算法和美剧爱好者

仅从机器学习的角度讨论这个问题。

相对熵（relative entropy）就是KL散度（Kullback–Leibler divergence），用于衡量两个概率分布之间的差异。

对于两个概率分布$p(x)$和$q(x)$，其相对熵的计算公式为：

$$\mathrm{KL}(p \parallel q) = -\int p(x) \ln q(x) dx - (-\int p(x) \ln p(x) dx)$$

注意：由于$p(x)$和$q(x)$在公式中的地位不是相等的，所以$\mathrm{KL}(p \parallel q) \neq \mathrm{KL}(q \parallel p)$.

相对熵的特点，是只有$p(x)=q(x)$时，其值为0。若$p(x)$和$q(x)$略有差异，其值就会大于0。其证明利用了负对数函数（$-\ln x$）是严格凸函数（strictly convex function）的性质。具体可以参考*PRML* 1.6.1 Relative entropy and mutual information.

相对熵公式的前半部分$-\int p(x)\ln q(x)dx$就是**交叉熵（cross entropy）**。

若$p(x)$是数据的真实概率分布，$q(x)$是由数据计算得到的概率分布。机器学习的目的就是希望$q(x)$尽可能地逼近甚至等于$p(x)$，从而使得相对熵接近最小值0. 由于真实的概率分布是固定的，相对熵公式的后半部分$(-\int p(x)\ln p(x)dx)$就成了一个常数。那么相对熵达到最小值的时候，也意味着交叉熵达到了最小值。对$q(x)$的优化就等效于求交叉熵的最小值。另外，对交叉熵求最小值，也等效于求最大似然估计（maximum likelihood estimation）。具体可以参考*Deep Learning* 5.5 Maximum Likelihood Estimation.

张一山
FPGA / IC for AI

正在学习 DL Book 第6章，查资料看到了这个问题。
受第一个答案启发，自己写了一些笔记。
分享出来供大家参考。有问题欢迎讨论。

先给结论：
1）信息熵：编码方案完美时，最短平均编码长度的是多少。
2）交叉熵：编码方案不一定完美时（由于对概率分布的估计不一定正确），平均编码长度的是多少。
平均编码长度 = 最短平均编码长度 + 一个增量
3）相对熵：编码方案不一定完美时，平均编码长度相对于最小值的增加值。（即上面那个增量）

零、信息熵
1、熵的本质是**香农信息量 log(1/p) 的期望**；（参考了第一个答案）
H(p) = E[log(1/p)] = ∑ p_i * log(1/p_i)，是一个期望的计算，也是记录随机事件结果的平均编码长度；
为什么信息量 是 log(1/p) 呢?
因为：一个事件结果的出现概率越低，对其编码的bit长度就越长。
以期在整个随机事件的无数次重复试验中，用最少的 bit 去记录整个实验历史。
即无法压缩的表达，代表了真正的信息量。

2、熵的本质的另一种解释：**最短平均编码长度**；
【本质含义：编码方案完美时，最短平均编码长度的是多少】

3、**交叉熵**，则可以这样理解：**使用了“估算”的编码后，得到的平均编码长度**（可能不是最短的）
p是真实概率分布，q是你以为的概率分布（可能不一致）；
你以 q 去编码，编码方案 log(1/q_i)可能不是最优的；
于是，平均编码长度 = ∑ p_i *log(1/q_i)，就是交叉熵；
只有在估算的分布 q 完全正确时，平均编码长度才是最短的，交叉熵 = 熵

一、交叉熵
1、定义
【本质含义：编码方案不一定完美时，平均编码长度的是多少】
连续函数：

$$H(p,q)=\mathrm{E}_p[-\log q]=H(p)+D_{\mathrm{KL}}(p\|q)$$

两项中 H(p)是 p的信息熵，后者是相对熵；
离散函数：

$$H(p,q)=-\sum_x p(x)\,\log q(x)$$

=Entropy(P) + D_KL(P||Q)

2、在 ML 中等效于相对熵
【作用：用来评估，当前训练得到的概率分布，与真实分布有多么大的差异】
因为与相对熵只差一个 分布 P 的信息熵，
若 P 是固定的分布，与训练无关；
Q 是估计的分布，应尽量等于 P。
二者一致时，交叉熵就等于 P 的熵

二、相对熵
【本质含义：由于编码方案不一定完美，导致的平均编码长度的增大值】
离散：

$$D_{\mathrm{KL}}(P\|Q) = \sum_i P(i)\,\log\frac{P(i)}{Q(i)}$$

【发现：D_KL(P||Q) = ∑P(i) *logP(i) - ∑P(i) *logQ(i)
= - Entropy(P) + 交叉熵 H(p,q) 】
连续：

$$D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x)\,\log\frac{p(x)}{q(x)}\,\mathrm{d}x$$

1）用来衡量2个取值为正数的函数的相似性
2）2个完全相同的函数，相对熵为0；差异越大，相对熵越大；
3）概率分布函数，或 概率密度函数，若函数值均大于0，相对熵可以度量两个随机分布的差异性；
4）相对熵不对称，没有交换律

参考：
《数学之美》吴军
wiki
第一个高票答案
编辑于 2017-01-15
▲赞同 106 ▼ 9 条评论
分享
收藏 感谢 收起

Louis
机器学习算法工程师
20 人赞同了该回答

交叉熵H(p,q)=-sum(p*logq)，表示用分布q模拟真实分布p所需的信息，可作为一种loss。

相对熵又称KL距离，KL(p||q)=H(p,q)-H(p,p)，表示用分布q模拟真实分布p相比用p模拟p，所需的**额外**信息。

KL>=0，当p==q时，取等。最小化KL距离，等价于最小化交叉熵，对应机器学习的一种目标。

发布于 2017-02-21
▲赞同 20 ▼ 2 条评论
分享
收藏 感谢

爱上层楼
关注计算机视觉和深度学习，喜欢登山和古典乐
7 人赞同了该回答

在机器学习中，比如分类问题，如果把结果当作是概率分布来看，标签表示的就是数据真实的概率分布，由softmax函数产生的结果其实是对于数据的预测分布，预测分布和真实分布差值叫做KL散度或者是相对熵。我们希望的是预测值尽量靠近真实分布，也就是希望相对熵可以越来越小。相对熵又等于交叉熵减去数据真实分布的熵，后者是确定的，所以最小化相对熵就等价于最小化交叉熵。这就是交叉熵损失函数的由来，衡量的是预测值和真实标签之间的差异性，训练的目的是不断减少Loss，也就是让预测值不断靠近真实值。大家可以结合前几个高票答案的公式一起理解。

编辑于 2018-04-28
▲赞同 7 ▼ 添加评论
分享

★收藏 ♥感谢

匿名用户

23 人赞同了该回答

信息熵完美编码，
交叉熵不完美编码，
相对熵是两者的差值，交叉熵减去信息熵。差值即差异，也即KL散度。

编辑于 2017-08-18

▲赞同 23 ▼ 2 条评论

分享

★收藏 ♥感谢

溪Mio

人工智障

18 人赞同了该回答

在深度学习，分类器里的loss函数也就是交叉熵的本质其实是logistic函数的最大似然估计然后取log，

$$
\begin{aligned}
L(\hat{p},\beta) &= \prod_{y_i=1} P(Y=y_i|X_i)\cdot\prod_{y_i=0} P(Y=y_i|X_i) \\
&= \prod_{i=1} P(Y=y_i=1|X_i)^{y_i}\cdot P(Y=y_i=0|X_i)^{1-y_i} \\
&= \prod_{i=1} P(Y=y_i=1|X_i)^{y_i}\cdot (1-P(Y=y_i=1|X_i))^{1-y_i} \\
log(L(\hat{p},\beta)) &= \sum_{i=1}(y\log\hat{y}+(1-y)\log(1-\hat{y}))
\end{aligned}
$$

至于你们信息论那一套，对不起我没学过

编辑于 2017-12-03

▲赞同 18 ▼ 37 条评论

分享

★收藏 ♥感谢

俞子酱

有深度，还有高度

3 人赞同了该回答

整理、总结了

@Agenter
的回答，仅供参考。个人感觉Agenter回答得比较好。

相对熵和交叉熵

1 相对熵

- 相对熵（relative entropy）就是KL散度（Kullback–Leibler divergence），用于**衡量两个概率分布之间的差异**。
- 对于两个概率分布p(x)和q(x)，其相对熵的计算公式为：$\mathrm{KL}(p \parallel q) = -\int p(x) \ln q(x) dx - (-\int p(x) \ln p(x) dx)$ 注意：由于p(x)和q(x)在公式中的地位不是相等的，所以$\mathrm{KL}(p \parallel q) \neq \mathrm{KL}(q \parallel p)$
- 相对熵的特点，当p(x)=q(x)时，其值为0。若p(x)和q(x)略有差异，其值就会大于0。其证明利用了负对数函数（-ln x）是严格凸函数（strictly convex function）的性质。

2 交叉熵

- 交叉熵（cross entropy）用于**衡量计算出的概率分布与真实的概率分布之间的差异**。
- 相对熵公式的前半部分$-\int p(x) \ln q(x) dx$就是交叉熵。
- 若p(x)是数据的真实概率分布，q(x)是由数据计算得到的概率分布。机器学习的目的就是希望q(x)尽可能地逼近甚至等于p(x)，从而使得相对熵接近最小值0. 由于真实的概率分布是固定的，相对熵公式的后半部分$(-\int p(x) \ln p(x) dx)$就成了一个常数。那么相对熵达到最小值的时候，也意味着交叉熵达到了最小值。对q(x)的优化就等效于求交叉熵的最小值。
- 对交叉熵求最小值，也等效于求最大似然估计（maximum likelihood estimation）。

发布于 2018-01-15

▲赞同 3 ▼ 1 条评论

分享

★收藏 ♥感谢

chenwi

这里有tensorflow实现的各种交叉熵函数：

tensorflow中四种不同交叉熵函数tf.nn.softmax_cross_entropy_with_logits() - CSDN博客

发布于 2018-05-01

▲赞同 ▼ 添加评论

分享

★收藏 ♥感谢