对数损失函数是如何度量损失的?

对于损失函数，像用平方损失函数 L(Y, f(X)) = (Y - f(X))^2或者绝对值损失函数都十分直观，但是对数损失函数 L(Y, P(Y|X)) = -log P(Y|X)是如何来度量损失的呢?

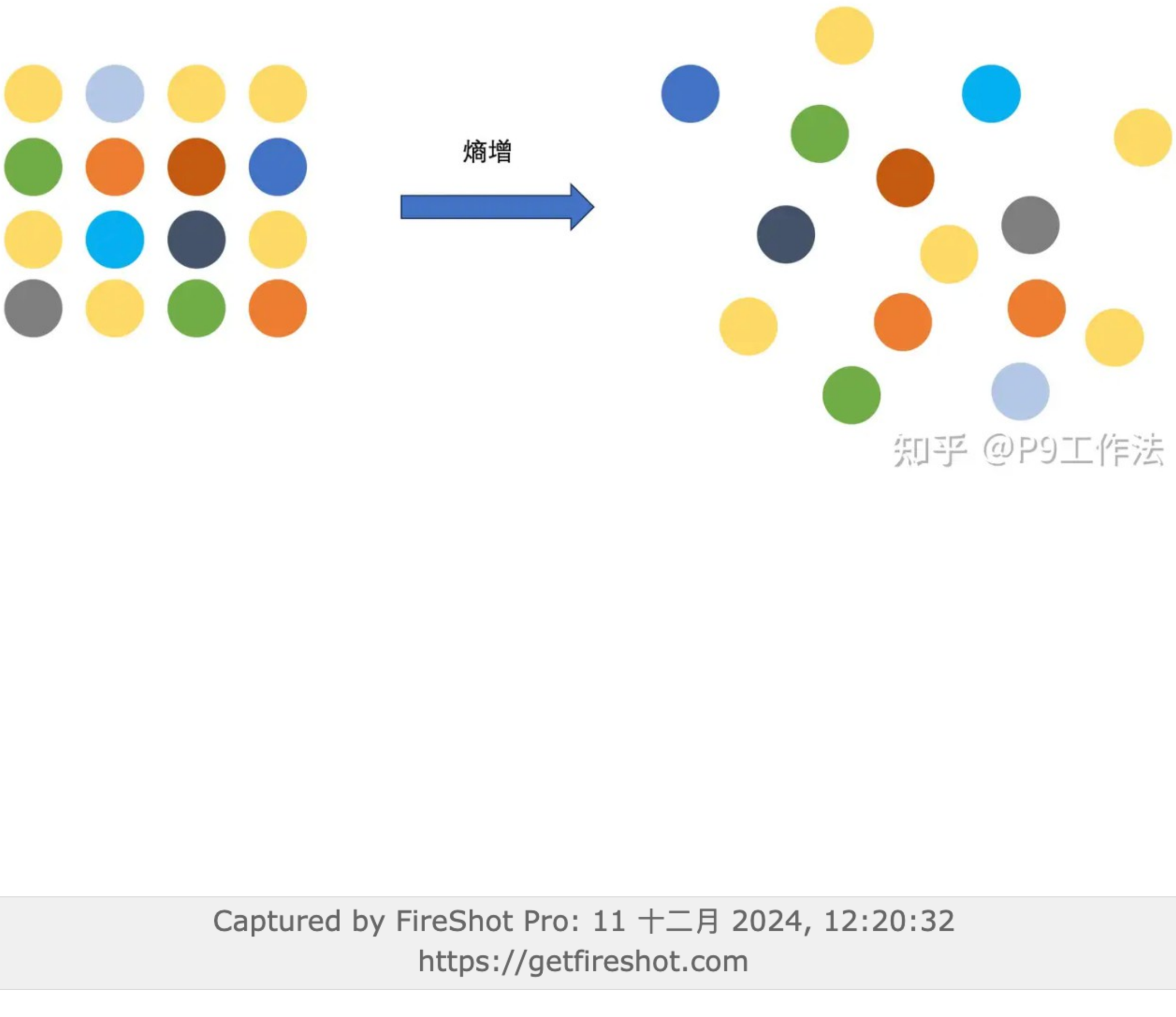
P9工作法
分布式架构、上百人团队管理、全球支付实践与AI技术
已关注

1 人赞同了该回答

交叉熵法

什么是熵

熵（Entropy）原本是一个物理热力学中的概念，代表一个系统无序度的量度。根据热力学第二定律✦，一个孤立系统的总熵不会减少，这意味着能量在转换过程中总会有一部分变得不可利用，即转变为无序的状态。通俗地说，假设一个屋子里面左边热，右边冷，那么没有外界干预的情况下，屋子左右两边的温度会最终一致。



在信息论中，也借用了这个概念，来度量信息的不确定性，由 香农✦ 提出。在一个信息源中，如果都是大量的小概率事件，那么这个系统的熵就越大。直白地理解，如果一个美女选男朋友，3个候选人她的感觉都差不多，这对美女来说是不是最拿不准选谁，是不是不确定性最高。但如果有一个候选人印象非常好，那是不是这个美女的确定性就更强。为此，把这个熵用数学公式来表达，为：

$$H(X) = -\sum_i p(x_i) \log_b p(x_i)$$

p(xi) 是信息源发出第 i 个消息的概率， b 是对数的底数（通常是 2 或 e）， $H(X)$ 就是该信息源的熵。

最优编码长度

提及交叉熵✦之前，必须要先理解一个概念：最优编码长度。即在无损编码✦的情况下，任何有效的编码方案都不可能让平均码长低于信息源的熵。如果存在一种编码，使得其编码后的平均码长等于信源的熵，这种编码被认为是“最优”的，因为它达到了理论上的最小平均码长。这就是最优编码长度。

以如下表的明天天气情况为例：

	阴天	晴天	下雨	下雪
出现概率	1/2	1/4	1/8	1/8
信息量/编码程度	1	2	3	3

按照上面的公式，这个系统的信息熵为：1/2+2*1/4+3*1/8+3*1/8=1.75

比较下信息熵编码与传统编码的区别

	阴天	晴天	下雨	下雪
普通编码	00	01	10	11
信息熵编码	0	10	110	111

那么如果要表达这样一个信息：阴天、晴天、阴天、阴天、下雪、阴天、晴天、下雨。用普通编码那就是：0001000010000110 ，一共16位。用信息熵编码那就是：01000111010110，一共14位。

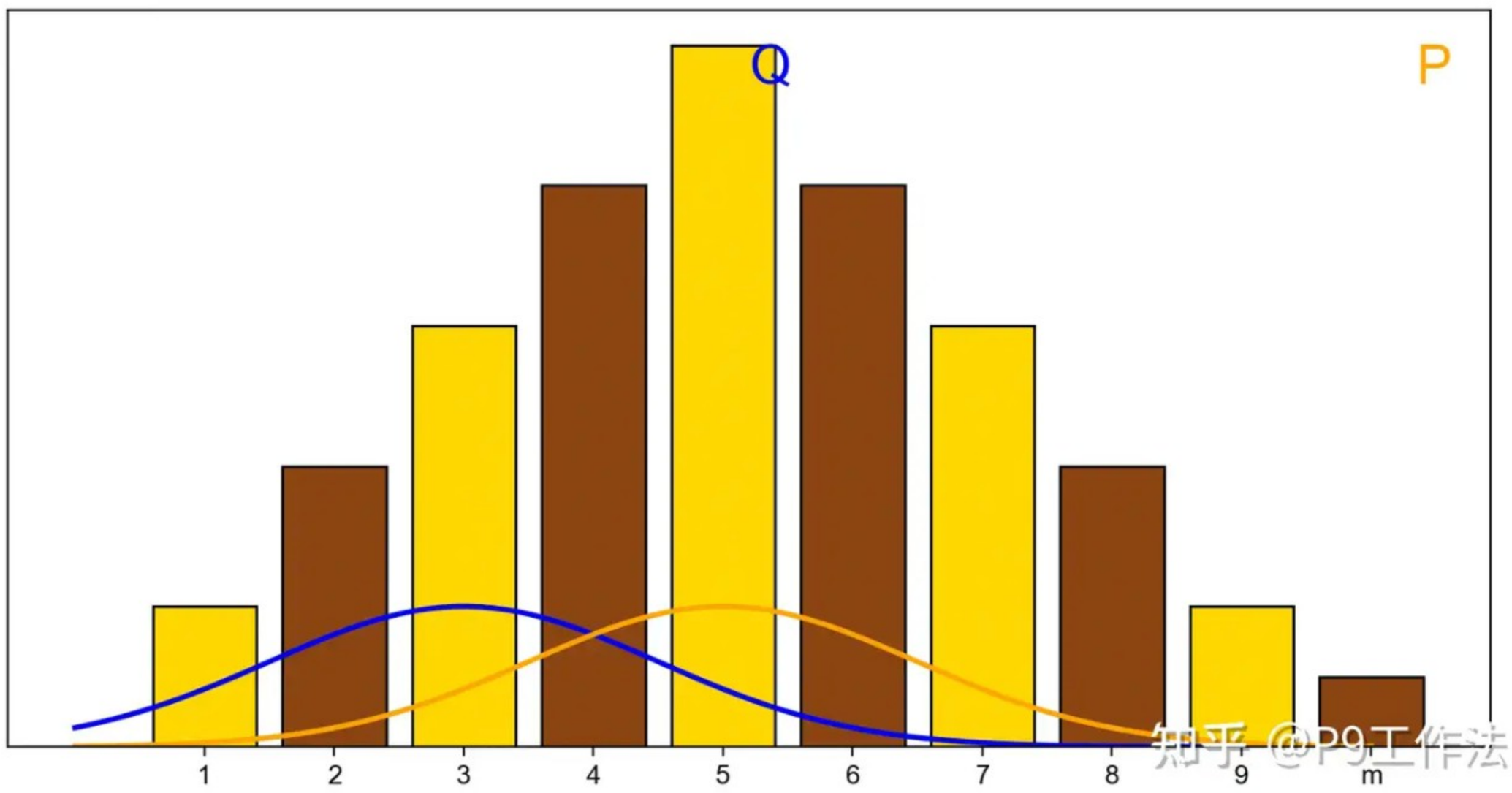
所以明显看到信息熵✦的编码是最低的。

而且可以看到，熵H(X)，其实是给出了在这种最优情况下平均每个符号需要的比特数（阴天1个比特✦，晴天2个比特，下雨3个比特，下雪3个比特）。

什么是交叉熵

那么对于一个概率分布来说，按照上述公式是可以得到熵 $H(x)$ 的，那么如何有这样一个编码方案，它的平均码长 L 满足 L≈H(P)，那么我们可以认为这个编码方案是接近最优的。

那么沿着这条路继续思考，是不是可以用熵来衡量两个概率分布的差异呢。这就是交叉熵。交叉熵 H(P,Q)涉及到了两个概率分布 P 和 Q，其中 P 通常是实际的概率分布，而 Q 则是某种估计或者模型预测的概率分布。交叉熵衡量了使用 Q 编码 P的平均信息量。换句话说，交叉熵告诉我们，如果我们用 Q 分布来编码 P 分布产生的信息，那么平均每个符号需要多少比特。



如果我们希望编码长度尽可能地反映实际分布 P，那么 Q应该尽可能地接近 P。在这种情况下，交叉熵 H(P,Q)将会接近于 P的熵 H(P)。也就是说，如果我们有一个好的 Q（即 Q 接近 P），那么使用 Q来编码 P所产生的信息将是非常高效的，编码长度也将接近最优。

这就是交叉熵，用数学公式表达为： $H(P,Q) = -\sum_x P(x) \log Q(x)$

最小化交叉熵

上面介绍了交叉熵的基础知识，那么在机器学习中是如何运用这个交叉熵理论呢。这就是最小交叉熵损失，这被广泛运用在分类任务中。这实际上是为了让模型预测的概率分布 Q更接近实际分布 P。

最小化交叉熵损失函数公式为：

- 对于多分类问题

对于多分类问题，假设有C个类别，真实标签 y 是一个 one-hot 向量，而模型的预测 $\hat{y}$ 是一个概率向量。交叉熵损失函数✦可以表示为

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

- 对于二分类问题

对于二分类问题，真实标签 y是一个标量（通常是 0 或 1），模型的预测 $\hat{y}$ 是一个介于 0 和 1 之间的概率值。交叉熵损失函数可以简化为：

$$L = -\left[y \log(\hat{y}) + (1-y) \log(1-\hat{y})\right]$$

看到这个函数是不是和极大似然估计推到的一样的，的确是一样的，只是代表的含义不一样。即两个含义得到一个结论。

与极大似然估计的区别

对于最终结果如果是非0即1的值，那么极大似然和交叉熵一样的。极大似然✦看成是一次次抽样，然后反向去求解最优的参数，所以抽样结果就两个，是或者不是。但如果最终结果是一个概率，比如这个图片95%概率像猫，5%概率像狗，那么用交叉熵是最好理解的，因为从概率来理解是最通顺的。

自定义损失函数

损失函数是否可以自定义，这显然是可以的。因为这涉及到你想要达到什么样的目标，比如说预测商品的销量，如果纯粹是只是看销量的准确度显然用MSE就可以搞定。

但实际情况可能是销量预测只是手段，目标是要提升利润。也就是说如果预测值 $\hat{y} - y$ >0，销量预测多了是有成本损失的。如果 $\hat{y} - y$ <0, 那是会减少利润的。如果把这样的信息加入进去，就可能会得到提升利润的损失函数。

可能会是这样的损失函数：

$$f(\hat{y}, y) = \begin{cases} \text{PROFIT} * (\hat{y} - y) & \text{if } y < \hat{y} \\ \text{COST} * (y - \hat{y}) & \text{if } y \geq \hat{y} \end{cases}$$

损失函数小结

损失函数是机器学习非常重要的一环，但只是若非做算法的，不需要理解那么细致，掌握其原理就好了。