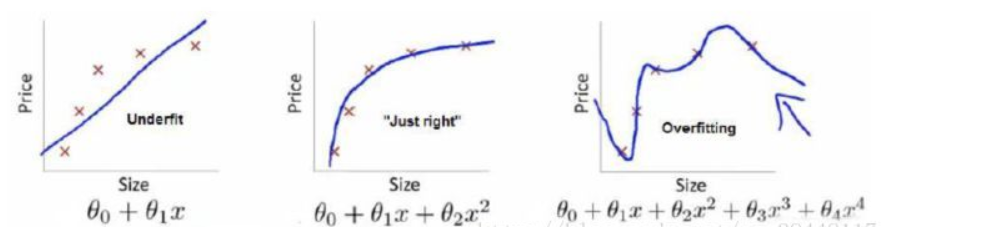
# 目标函数与损失函数的联系

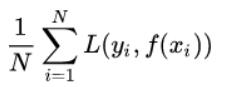
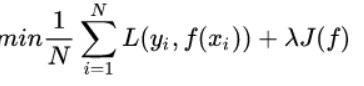
损失函数：是描述预测样本值与真实样本值误差的函数

目标函数是一个评估范围更广的概念，举例说明：



上面三个图的曲线函数依次为f1(x),f2(x),f3(x)，我们想用这三个函数分别来拟合真实值Y。我们给定x，这三个函数都会输出一个f(X)，这个输出的f(X)与真实值Y可能是相同的，也可能是不同的，为了表示我们拟合的好坏，我们就用一个函数来度量拟合的程度。这个函数就称为损失函数。

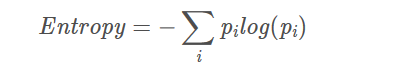
损失函数越小，就代表模型拟合的越好。那是不是我们的目标就只是让loss function越小越好呢？还不是。这个时候还有一个概念叫风险函数。风险函数是损失函数的期望，这是由于我们输入输出的(X,Y)遵循一个联合分布，但是这个联合分布是未知的，所以无法计算。但是我们是有历史数据的，就是我们的训练集，f(X)关于训练集的平均损失称作经验风险(empirical risk)，所以我们的目标就是最小化经验风险。那我们看上面的图，那肯定是最右面的f3(x)的经验风险函数最小了，因为它对历史的数据拟合的最好。但是我们从图上来看它肯定不是最好的，因为它过度学习历史数据，导致它在真正预测时效果会很不好，这种情况称为过拟合。为什么会造成这种结果？大白话说就是它的函数太复杂了，都有四次方了，这就引出了下面的概念，我们不仅要让经验风险最小化，还要让结构风险最小化。这个时候就定义了一个函数J(f)，这个函数专门用来度量模型的复杂度，在机器学习中也叫正则化。常用的有L1， L2范数。到这一步我们就可以说我们最终的优化函数，观察如下加入正则化的变化。

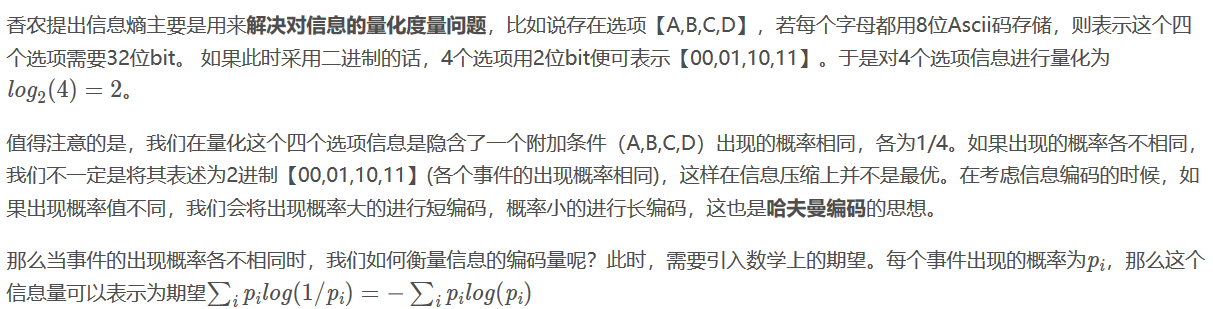
**即最优化经验风险和结构风险，而这个函数就被称为目标函数**

# **信息熵、条件熵、相对熵、交叉熵**

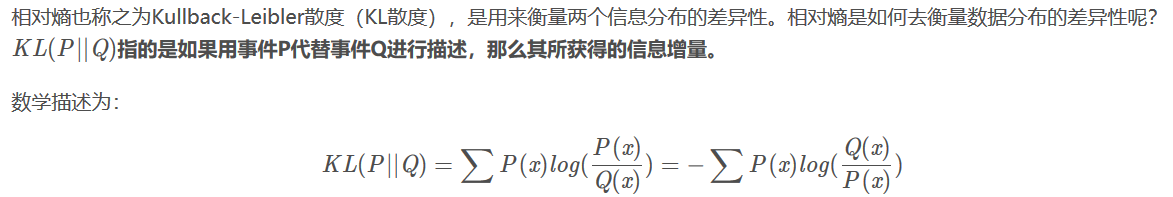
所谓信息熵：是对每个可能性编码所需要的期望值，延伸一下，也是对信息的量化指标。香农则将信息熵用来**描述信息源的不确定度**， 信息熵越大，混乱度越大。一般均匀分布的信息熵是最大的等于一，其表达式为：

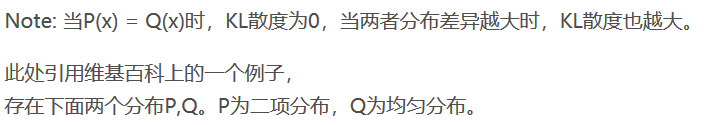


**P为某一事件发生的概率**



**相对熵**





**比如同一批样本的两个相似的特征，这两个特征的相对熵就非常低，KL散度就非常小**



**交叉熵**

