从信息熵到自动编码机：（1）理解信息熵


人工智能狂想曲 简书作者

1.311 2019-06-02 16:11 IP属地: 上海 打开App

（使用 app 无法显示公式，请用网页浏览）

关键词：信息熵，自动编码机，Shannon Entropy， information theory，霍夫曼编码

写这一篇目的是形象化理解“信息熵”，避开“熵”的词语迷惑，信息熵是一个平均数。同时借由抛硬币的例子和霍夫曼编码展现信息熵理论的意义。

1.信息熵公式

信息熵公式为：

$$H(X) = -\sum_{x} P(x) \log P(x) \tag{1}$$

或者写为：

$$H(X) = \sum_{x} P(x) \log \frac{1}{P(x)} \tag{2}$$

(公式1，2)

H 为熵，X 为一个随机变量，p(x) 是 x 事件发生的概率。

2.信息熵来由

信息熵是香农在信息论里面提出的（1948年）。香农需要一个公式（量）来描述信息的属性如信息多少编码长度等，所以参考热力学熵及公式形式发明的信息熵。统计热力学熵公式是波尔兹曼发明的（1877年），他用统计学的方法把气体宏观温度和气体微观粒子动能联系起来，这个联系系数就是波尔兹曼常量。香农完全可以用其他名称而不是“熵 entropy”来命名这个公式，有些文章说香农也听从了朋友建议最终还是用“熵”来命名这个公式，原因是：“没有人能说清熵到底是什么，起名信息熵，这样在辩论中可以保持优势“。

这篇文章不图理解”熵“的完整哲学，尤其是热力学方面的含义，而仅仅从具体的“信息熵”概念和例子出发试图形象的记住“信息熵”的含义。

3.信息熵的定义

> Information entropy is the average rate at which information is produced by a stochastic source of data. The measure of information entropy associated with each possible data value is the negative logarithm of the probability mass function for the value。

来自 wikipedia 的定义。第二句是直接复述了信息熵公式（公式1）的内容。第一句是什么意思呢，大体说，信息熵是个平均数。

4.什么是信息（information）

我们已经知道信息熵是某某某的平均值，我们会猜这里的某某某是“信息”这个词。那么什么是信息?

> “Information: the negative reciprocal value of probability.”—Claude Shannon
>
> 香农直接用公式中的一部分定义什么是信息。

看 wikipedia 上对信息的定义和描述：

> The information content(also called the surprisal) of an event is an decreasing function of the reciprocal of the probability p(E) of the event, precisely:

$$I(E) = -log2(p(E)) = log(1/p(E)) \tag{3}$$

这里的信息概念有点太抽象，能用纯粹的数学公式定义还是很有趣的，知道信息的定义也有利于形象看待公式1的结构。

一个小例子来看什么是信息。几个同事下班一块坐地铁，到了“上海游泳馆”地铁站，A 问：“不知道这里能不能游泳”。另外一个同事 B 回答，“搞不懂上海游泳馆是干啥的，不过，游泳的话肯定得花钱”。

“肯定得花钱” 是一句搞笑的废话，听的大家哈哈笑。在这个情景下，A 是要咨询一定的信息，B 的回答是句废话没有提供任何有用的信息。因为“肯定得花钱”是 100% 发生，说了跟没说一样，信息量是零。

由信息公式可见，传递一个概率为 1 的事件，这个事件所携带的信息量为 0。从公式看概率为 0 的事件也没意思，那么在概率在 0-1 之间的事件所携带的信息量有大有小，概率越小的事件传递的信息量越大。这也符合我们日常描述，当某某稀缺消息发生时，我们说信息量爆炸了，需要冷静处理一下。

这里会有疑问信息千种万样，怎么度量信息的量差别呢？信息熵就是这样的一种标准化和

抽象化工具。公式和上面的定义其实一步走到了量化信息本身，其实我们可以研究一下中间过程：如果有种完美的编码方式，可以编码任何信息，那么一条信息的量大小可以用编码长度来衡量。另外一个重要的角度是引入随机变量 X 和传递事件这一过程（通信）来观察信息量大本质。结合公式和例子会更容易理解这两点。

5.再看公式1

$$H(X) = \sum_{x} P(x) \log \frac{1}{P(x)} \tag{1}$$

H 为熵，X 为一个随机变量，p(x) 是 x 事件发生的概率。 由前面讲述 log（1/p(x)) 衡量的是 x 事件携带的信息量，那么为什么会有一个 log?

信息量在实际使用中还得转变为具体信息载体如 bit 或者其他编码载体，所以 log 就是用这种具体载体的消耗量来表征信息量的抽象。当信息用二进制的 bit 为载体的时候，log 就是 2 为底的对数。假设一个事件 x 发生的概率是 1/4，且这个事件从发生地点 C（如客户端/信号源/信号发生器） 传递到 S（如服务器/信号接收器）所携带的信息量是 log(4)，当用 bit 表征信息量大小时候，这个信息量是 log2(4) = 2 bit。 也就是公式中的 log 可以是任何底的对数，一般是 2，也可以是 10，如果是 2 时候正好可以用二进制的 bit 来做信息载体。

再看公式，信息熵是信息量的平均值，普通的 log 是 2 为底的对数，这时候信息量的单位可以是 bit，信息熵的单位也是 bit。

信息量的平均值不是凭空产生的，而是有具体的环境，就是上面多次提到的随机事件。

6.随机事件和消息传递

这里的信息传递其实抽象成了随机事件的传递，所谓一条信息其实是一个事件发生且传递出去。事件是符合一定分布和规律的，熵代表了信息量（传递一个事件发生）的平均大小。

举个例子：俩个完美硬币，被抛的时候，符合正反面 1/2 的发生概率。这两个完美硬币只在 A 地有，B 地没有。B 地用户需要这类抛硬币随机数的时候，只能雇佣 A 地人员实地操作并且把结果传递给回来。具体的可能 4 种事件，每种概率如下：（H - 头正面，T - 尾反面）

$$
\begin{aligned}
&HH\text{：}1/4,\\
&HT\text{：}1/4,\\
&TH\text{：}1/4,\\
&TT\text{：}1/4,
\end{aligned}
$$

具体传递时候，双方可以事先约定好利用 bit 编码：

$$
\begin{aligned}
&00\text{就是（}HH\text{），}\\
&01\text{就是（}HT\text{），}\\
&10\text{就是（}TH\text{），}\\
&11\text{就是（}TT\text{），}
\end{aligned}
\qquad (\text{encode } 1)
$$

（编码1）

我们用公式计算一下信息熵：

随机事件 X 是抛两个完美硬币，

$$
\begin{aligned}
&HH\text{发生概率：}\ p(x=HT)=1/4,\ p(x=HT)\text{下面记做}p(x1)\\
&HT\text{发生概率：}\ p(x=HT)=1/4,\ p(x=HT)\text{下面记做}p(x2)\\
&TH\text{发生概率：}\ p(x=TH)=1/4,\ p(x=TH)\text{下面记做}p(x3)\\
&TT\text{发生概率：}\ p(x=TT)=1/4,\ p(x=TT)\text{下面记做}p(x4)
\end{aligned}
$$

我们用 bit 传递消息（事件），log 是 2 为底的对数。计算信息熵 H(X):

$$H(X)=p(x1)*log(1/p(x1))+p(x2)*log(1/p(x2))+p(x3)*log(1/p(x3))+p(x4)*log(2/p(x4))$$

代入具体数：

$$H(X) = (1/4) * 2 + (1/4) * 2 + (1/4) * 2 + (1/4) * 2 = 2$$

单位是 bit。

也就是传递抛两个完美硬币事件结果的信息，其熵是 2 bit。

信息熵是平均值，传递 X（抛两枚硬币的随机结果），信息的平均值是 2 bit。

也就是多次重复实验的时候，每次消息传递所消耗的是 2 bit。

这个结果并没人任何意外，因为我们的编码就是每个事件用相同 2 bit 编码的，不偏不倚。所以我们需要看一个有概率偏差的例子：

假设两个硬币不是完美硬币，抛掷时候两个硬币概率如下：

$$HH: 10/16$$
$$HT: 3/16$$
$$TH: 2/16$$
$$TT: 1/16$$

抛不完美 2 枚硬币信息熵为：

$$H(X) = (10/16) * 0.678 + (3/16) * 2.415 + (2/16) * 3 + (1/16) * 4 = 1.5bit$$

传递这 X 的熵是 1.5 bit。比上一个无偏差的 2 bit 要小。

这里可以看到越是无序熵越大。第一个 X 事件每个发生概率相等，无任何规律就和噪音一样平铺在平面上，第二个 X 有些规律 HH 发生概率很大，多次重复采样话，图片在这一点会很亮

然会很完。

因为事件概率分布不同，两次计算的熵不一样，一个 2 bit 一个 1.5 bit。根据熵定义传递第二组信息的时候，一条有效消息所占编码长度的平均值是 1.5 bit。

但是实际情况呢？我们已经约定消息传递的编码（编码1），每条消息都是占用 2 个bit，所以在这种编码的情况下，多次重复实验，即传多次传递这样的消息之后，每条消息的平均长度肯定还是 2 bit，那么信息熵计算的 1.5 bit 的平均值意义何在？

7.霍夫曼编码

当我们约定使用编码1传递第二组信息的时候，我们实每条信息的平均长度是 2 bit，那么比信息熵 1.5 bit多出的 0.5bit是什么？是冗余，我们在传递信息本身时候还有空余能力没被使用或者被噪音占据。

我们的编码方式（编码1）在传递第二组信息的时候，不是最高效的。我们很容想出更不高效的，比如：我们协议拍照片传递事件结果，那么个消息的平局值能到几 M 比特。

是否存在更高效的编码呢？使得实际平均编码长度更短接近信息熵。霍夫曼编码就是这种更高效编码方式之一。

霍夫曼编码就是把消息按照概率大小排序，再进行树状编码。

以传递第二组信息为例子（抛掷有偏差的两个硬币）

$$
\begin{aligned}
&HH\text{：}10/16\text{序号：}0\\
&HT\text{：}3/16\text{序号：}1\\
&TH\text{：}2/16\text{序号：}2\\
&TT\text{：}1/16\text{序号：}3
\end{aligned}
$$

我们把事件按照概率排序，发明一种编码：让序号的二进制就是事件的编码：

$$
\text{编码0就是事件}HH \qquad (\text{encode2 - not work})
$$

编码0就是事件HH，　　(encode2，not work)
编码1就是事件HT，
编码10就是事件TH，
编码11就是事件TT，

（编码2）

但是在实际操作中，以上编码有问题，如当 B 收到一个bit 1 的时候，会有歧义，这个 1bit 是指的事件 HT 还是需要继续等待后面一个比特来区分 TH 和 TT。

所以，以上编码是无效的，我们为了高效把固定长度编码变为变长度，但是变长度编码需要区分消息完整性或者消息的分界点（面向实际使用的流式编码）。

霍夫曼编码是这样解决这个问题的：

bit0就是事件HH　　(encode3，Huffman Coding)
bit10就是事件HT
bit110就是事件TH
bit111就是事件TT

（编码3）

霍夫曼编码是能工作的，当 B 收到 0 时候确认事件 HH，当收到 1 时候需要等待下一个 bit ，下一 bit 是 0 确认事件 HT，下一比特是 1 的时候需要等待第二 bit 确认 TH 或者 TT。

霍夫曼编码把概率最高的事件用最短编码标记，可以想象如果事件 HH 概率非常高，那么这种编码比固定长度编码会节省很多。实际应用如 HTTP2 协议 HPACK 头部压缩协议就是用霍夫曼编码传递 HTTP 头部如方法名 “GET” “POST”，客户端和服务器约定霍夫曼编码表，这个编码表中概率最高的 “GET” 字符串用较短的 bit 数标记，来增加信息压缩效率。

我们分别计算编码 2 和编码 3 的平均消息长度

编码2

$$E(X) = (10/16) * 1 + (3/16) * 1 + (2/16) * 2 + (1/16) * 3 = 1.25bit;$$

编码3(霍夫曼编码)

$$E(X) = (10/16) * 1 + (3/16) * 2 + (2/16) * 3 + (1/16) * 3 = 1.5625bit;$$

可见编码2虽然小于信息熵 1.5 bit，但是编码2是不工作的。在此概率分布下，霍夫曼编码工作且高效 1.5625 bit 比 1.5 bit 理论极限只有少量冗余。

霍夫曼编码冗余源于哪里？下一篇写。

7.总结

信息熵是个平均数，是信息长度平均数，当用 bit 作为信息载体的时候，这个平均数的单位是 bit。

为了传递一组消息（随机变量 X），会多次重复的传递其中一条消息（事件 x 发生），每条消息平均占用的 bit 长度就是信息熵。（基于一种理想的编码方式）

ps. 信息、消息、事件、information、message、sample 这些词汇分开讲述，提前约定会有利于交流，否则引起混乱

（下一篇：熵编码的冗余）

最后编辑于：2022-07-25 12:44

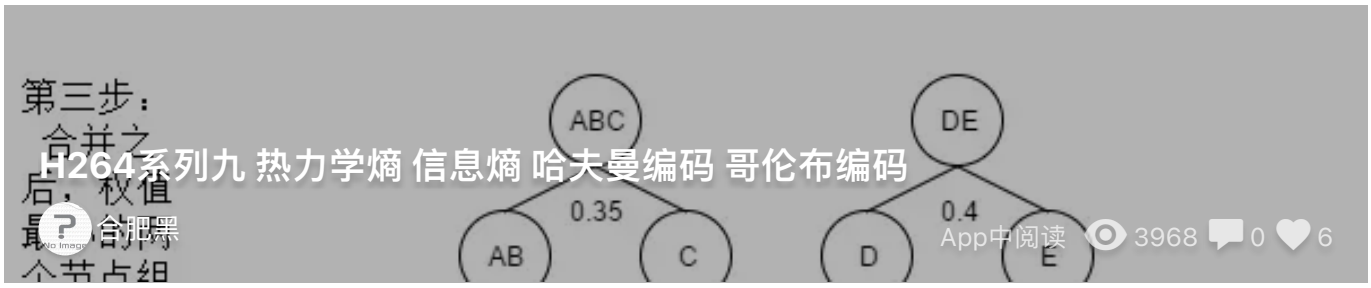
暂无评论

写评论

智慧如你，不想 发表一点想法 咩~

推荐阅读

更多精彩内容 >



H264系列九 热力学熵 信息熵 哈夫曼编码 哥伦布编码

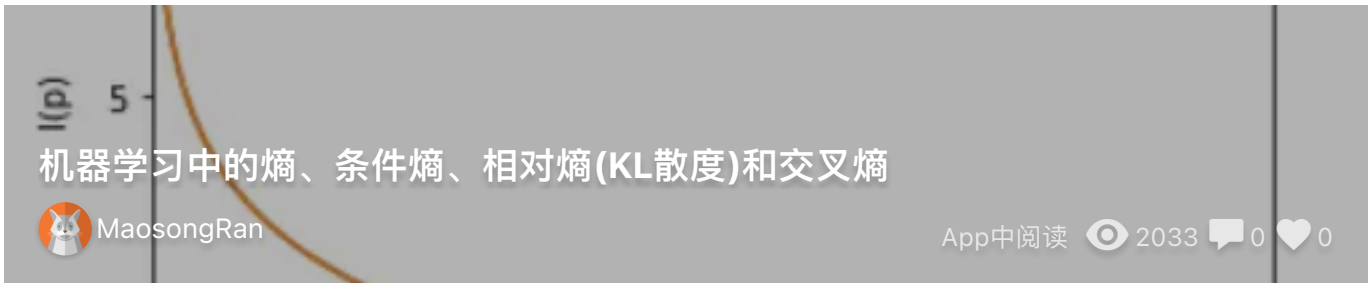
合肥黑

App中阅读 3968 0 6

对熵的理解

小碧小琳

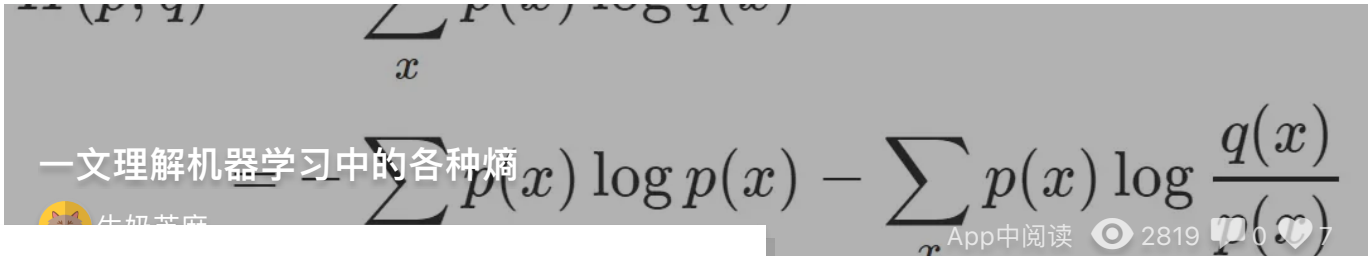
App中阅读 4776 0 5



机器学习中的熵、条件熵、相对熵(KL散度)和交叉熵

MaosongRan

App中阅读 2033 0 0



一文理解机器学习中的各种熵

App中阅读 2819 0 7