最大熵模型

七月算法 邹博

2015年11月1日

本次目标

- 理解并掌握熵Entropy的定义
  - 理解“Huffman编码是所有编码中总编码长度最短的”熵含义
- 理解联合熵H(X,Y)、相对熵D(X||Y)、条件熵H(X|Y)、互信息I(X,Y)的定义和含义，并了解如下公式：
  - $H(X|Y) = H(X,Y) - H(Y)=H(X) - I(X,Y)$
  - $H(Y|X) = H(X,Y) - H(X)=H(Y) – I(X,Y)$
  - $I(X,Y) = H(X) - H(X|Y) = H(X) + H(Y) - H(X,Y) \geqslant 0$
- 掌握最大熵模型Maxent
  - Maximum Entropy Models
- 了解最大熵在自然语言处理NLP中的应用
  - Natural Language Processing
- 与前序知识的联系：最大熵模型和极大似然估计MLE的关系
  - Maximum Likelihood Estimation
- 副产品：了解数据分析、函数作图的一般步骤

预备定理 $N \to \infty \Rightarrow \ln N! \to N(\ln N - 1)$

$$
\begin{aligned}
\ln N! &= \sum_{i=1}^{N} \ln i \approx \int_1^N \ln x dx \\
&= x \ln x \Big|_1^N - \int_1^N x d \ln x \\
&= N \ln N - \int_1^N x \cdot \frac{1}{x} dx \\
&= N \ln N - x \Big|_1^N \\
&= N \ln N - N + 1 \\
&\to N \ln N - N
\end{aligned}
$$

骰子

- 普通的一个骰子的某一次投掷，出现点5的概率是多大？
  - 等概率：各点的概率都是1/6
  - 对于“一无所知”的骰子，假定所有点数等概率出现是“最安全”的做法。
- 对给定的某个骰子，经过N次投掷后发现，点数的均值为5.5，请问：再投一次出现点5的概率有多大？

带约束的优化问题

- 令6个面朝上的概率为(p1,p2…p6)，用向量**p**表示。
- 目标函数：$H(\vec{p})=-\sum_{i=1}^{6} p_i \ln p_i$
- 约束条件： $\sum_{i=1}^{6} p_i = 1 \qquad \sum_{i=1}^{6} i \cdot p_i = 5.5$
- Lagrange函数：

$$L(\vec{p},\lambda_1,\lambda_2)=-\sum_{i=1}^{6} p_i \ln p_i + \lambda_1\left(1-\sum_{i=1}^{6} p_i\right)+\lambda_2\left(5.5-\sum_{i=1}^{6} i \cdot p_i\right)$$

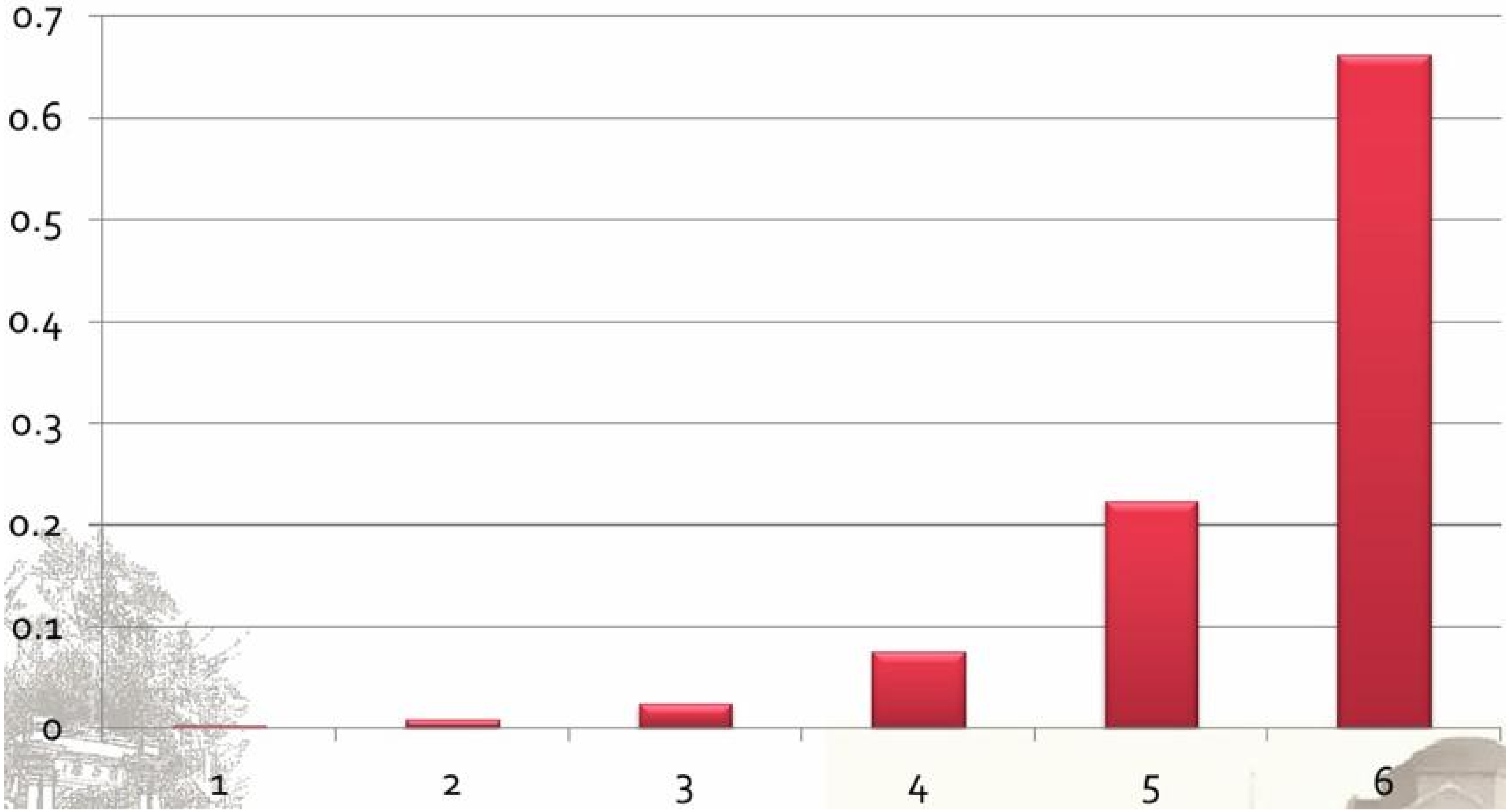
- 求解：

$$\frac{\partial L}{\partial p_i} = \ln p_i + 1 - \lambda_1 - i \cdot \lambda_2 \overset{令}{=\!=} 0$$

$$\Rightarrow p_i = e^{1-\lambda_1 - i\cdot\lambda_2}$$

$$\Rightarrow \lambda_1 = 5.932,\ \lambda_2 = -1.087$$

预测结果

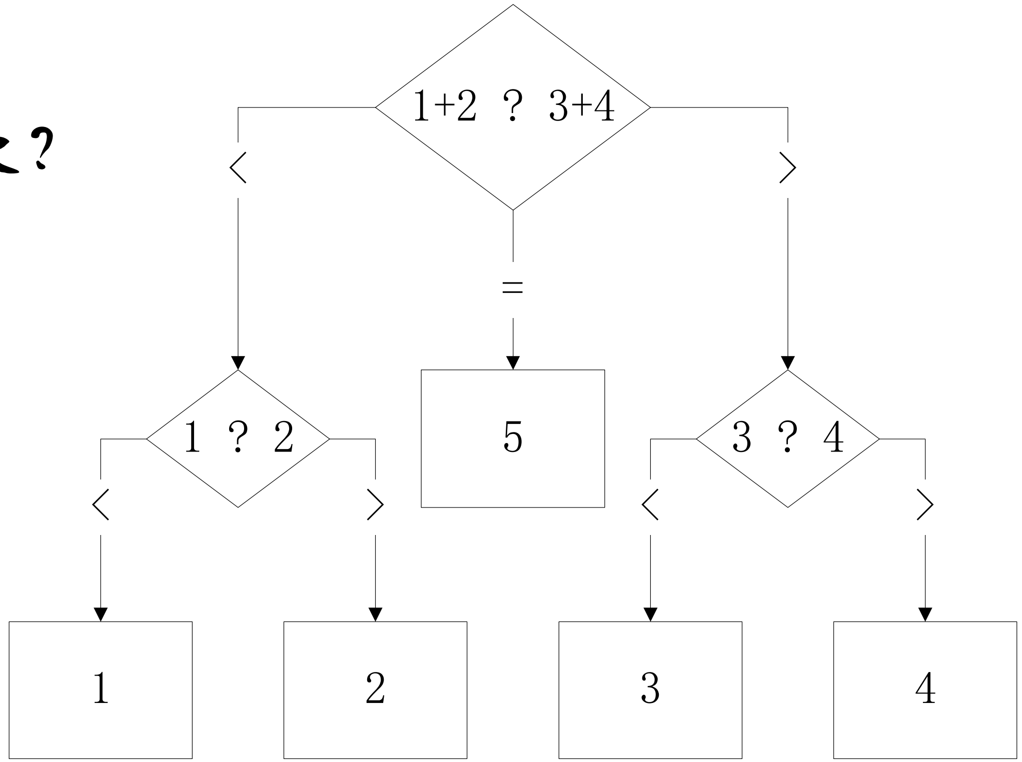


从小学数学开始

- 假设有5个硬币：1,2,3,4,5，其中一个是假的，且比真币轻。有一架没有砝码的天平，天平每次能比较两堆硬币，得出的结果可能是以下三种之一：
  - 左边比右边轻
  - 右边比左边轻
  - 两边同样重
- 问：至少要几次称量才能确保找到假币？

答案

- 一种可能的称量方法如右图所示
- 答案：2次
- 追问：为什么2次？



理论下界

- 令x表示假硬币的序号：$x \in X=\{1,2,3,4,5\}$；
- 令yi表示第i次使用天平得到的结果：$y \in Y=\{1,2,3\}$；
  - 1表示“左轻”，2表示“平衡”，3表示“右轻”
- 用天平称n次，获得的结果是：$y_1 y_2 \ldots y_n$；
  - $y_1 y_2 \ldots y_n$的所有可能组合数目是3^n；
- 根据题意，要求通过$y_1 y_2 \ldots y_n$确定x。即影射$map(y_1 y_2 \ldots y_n)=x$；
- 从而：$y_1 y_2 \ldots y_n$的变化数目大于等于x的变化数目
- 则：$3^n \geqslant 5$——一般意义上：$|Y|^n \ge |X|$

进一步分析

$$|Y|^n \ge |X| \Rightarrow n\log|Y| \ge \log|X| \Rightarrow n \ge \frac{\log|X|}{\log|Y|}$$

- 用$y_1y_2\ldots y_n$表达x。即设计编码x：$y_1y_2\ldots y_n$
- X的“总不确定度”是：$H(X) = \log|X| = \log 5$
- Y的“表达能力”是：$H(Y) = \log|Y| = \log 3$
- 至少要多少个Y才能准确表示X？

$$\frac{H(X)}{H(Y)} = \frac{\log 5}{\log 3} = 1.46$$

题目的变种

- 假设有5个硬币：1,2,3,4,5，其中一个是假的，比其他的硬币轻。已知第1、2个硬币是假硬币的概率都是三分之一；第2、3、4个是假硬币的概率都是九分之一。
- 有一架没有砝码的天平，假设使用天平n次能够找到假币。问n的期望值至少是多少？

解

- 1/3概率的硬币有2个，1/9概率的硬币有3个：

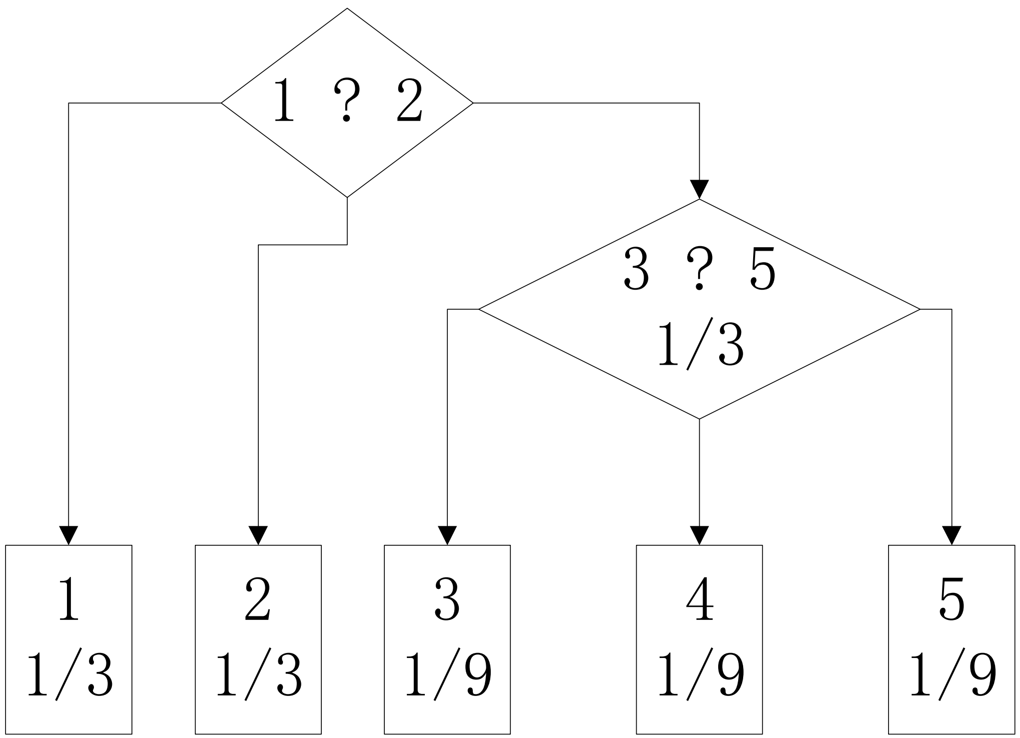
$$\left(\frac{1}{3}+\frac{1}{3}\right)\times\frac{\log 3}{\log 3}+3\frac{1}{9}\times\frac{\log 9}{\log 3}=\frac{4}{3}$$

- 思考：$\log_2 p$是什么？

解释Huffman编码

1	2	3	4	5
1/3	1/3	1/9	1/9	1/9

解释Huffman编码



附：Code

```
void HuffmanCoding(int *pWeight, int N, vector<vector<char> >& code)
{
    if (N <= 0)
        return;
    int m = 2 * N - 1;  //N个结点的Huffman树需要2N-1个结点
    HuffmanNode* pHuffmanTree = new HuffmanNode[m];
    int s1, s2;

    int i;
    //建立叶子结点
    for (i = 0; i < N; i++)
        pHuffmanTree[i].nWeight = pWeight[i];

    //每次选择权值最小的两个结点，建树
    for(i = N; i < m; i++)
    {
        SelectNode(pHuffmanTree, i, s1, s2);
        pHuffmanTree[s1].nParent = pHuffmanTree[s2].nParent = i;
        pHuffmanTree[i].nLeft = s1;
        pHuffmanTree[i].nRight = s2;
        pHuffmanTree[i].nWeight = pHuffmanTree[s1].nWeight + pHuffmanTree[s2].nWeight;
    }

    //根据建好的Huffman树从叶子到根计算每个叶结点的编码
    int node, nParent;
    for(i = 0; i < N; i++)
    {
        vector<char>& cur = code[i];
        node = i;
        nParent = pHuffmanTree[node].nParent;
        while(nParent != 0)
        {
            if(pHuffmanTree[nParent].nLeft == node)
                cur.push_back('0');
            else
                cur.push_back('1');

            node = nParent;
            nParent = pHuffmanTree[node].nParent;
        }
        reverse(cur.begin(), cur.end());
    }
}
```

定义信息量

- 原则：
  - 某事件发生的概率小，则该事件的信息量大。
  - 如果两个事件X和Y独立，即p(xy)=p(x)p(y)，假定X和Y的信息量分别为h(X)和h(Y)，则二者同时发生的信息量应该为h(XY)=h(X)+h(Y)。
- 定义事件X发生的信息量：$h(x) = -\log_2 x$
- 思考：事件X的信息量的期望如何计算呢?

熵

- 对随机事件的信息量求期望，得熵的定义：

$$H(X) = -\sum_{x \in X} p(x) \ln p(x)$$

  - 注：经典熵的定义，底数是2，单位是bit
  - 本例中，为分析方便使用底数e
  - 若底数是e，单位是nat(奈特)

研究函数f(x)=xlnx

- f(x)=xlnx， x∈[0,1]
- f'(x) = lnx + 1
- f''(x) = 1/x>0(凸函数)
- 当f'(x)=0时， x=1/e， 取极小值；
- 由于 $\lim_{x\to 0} f(x)=0$
- 定义f(0)=0

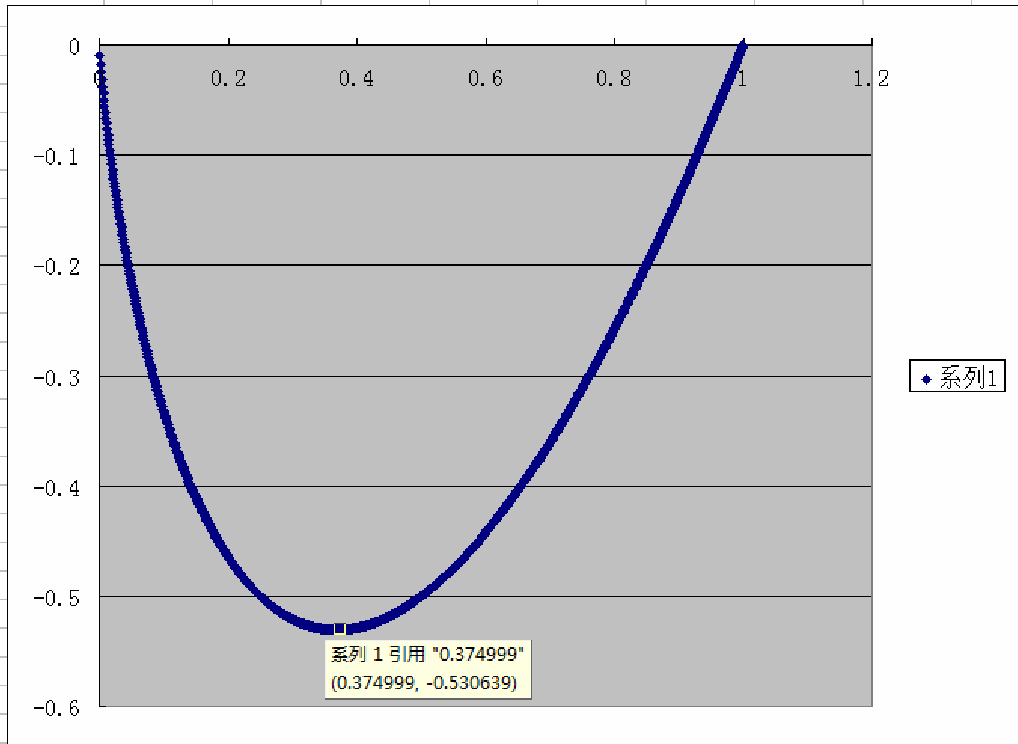
离散采样

```
int _tmain(int argc, _TCHAR* argv[])
{
    float x = 0.001f;
    float y;
    float log2 = log(2.0f);
    ofstream oFile;
    oFile.open(_T("D:\\entropy.txt"));
    while(x < 1)
    {
        y = x * log(x) / log2;
        oFile << x << '\t' << y << '\n';

        x += 0.001f;
    }
    oFile.close();
    return 0;
}
```

绘图

0.001	-0.00997
0.002	-0.01793
0.003	-0.02514
0.004	-0.03186
0.005	-0.03822
0.006	-0.04428
0.007	-0.05011
0.008	-0.05573
0.009	-0.06116
0.01	-0.06644
0.011	-0.07157
0.012	-0.07657
0.013	-0.08145
0.014	-0.08622
0.015	-0.09088
0.016	-0.09545
0.017	-0.09993
0.018	-0.10433
0.019	-0.10864
0.02	-0.11288
0.021	-0.11704
0.022	-0.12114
0.023	-0.12517
0.024	-0.12914
0.025	-0.13305
0.026	-0.1369
0.027	-0.14069
0.028	-0.14444
0.029	-0.14813
0.03	-0.15177
0.031	-0.15536

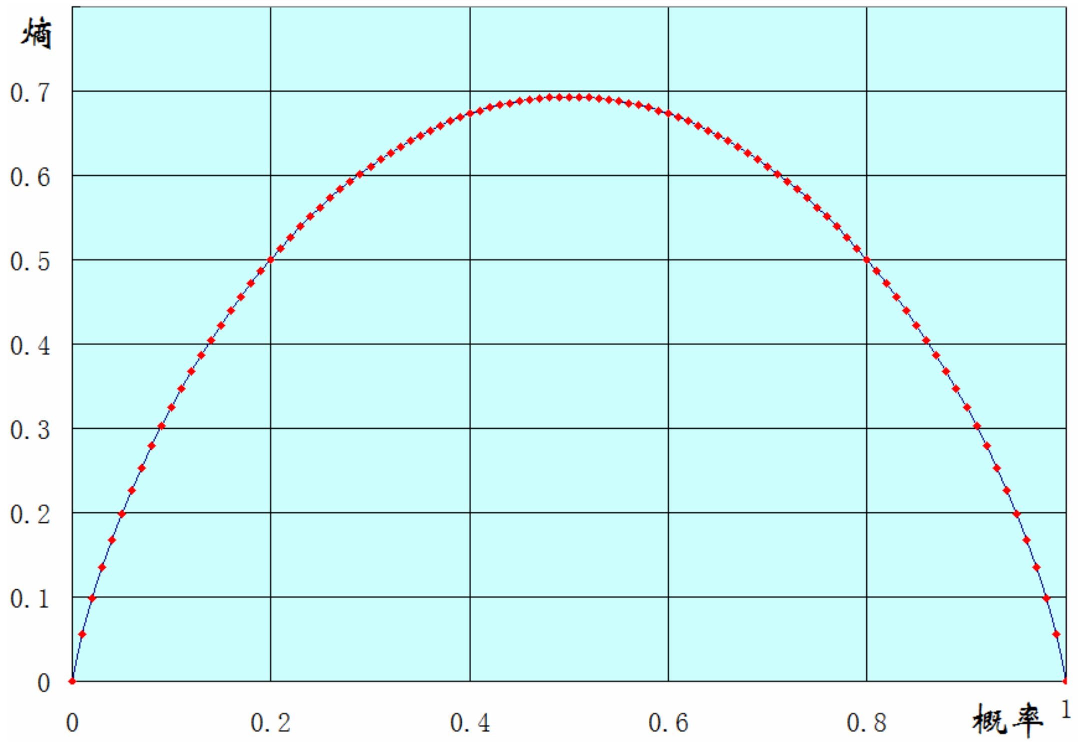


对熵的理解 $0 \le H(X) \le \log|X|$

- 熵是随机变量不确定性的度量，不确定性越大，熵值越大；若随机变量退化成定值，熵为0
  - 该不确定性度量的本质即为信息量的期望。
  - 均匀分布是"最不确定"的分布
- 熵其实定义了一个函数(概率分布函数)到一个值(信息熵)的映射。
  - P(x)→H （函数→数值）
  - 泛函："变分推导"章节

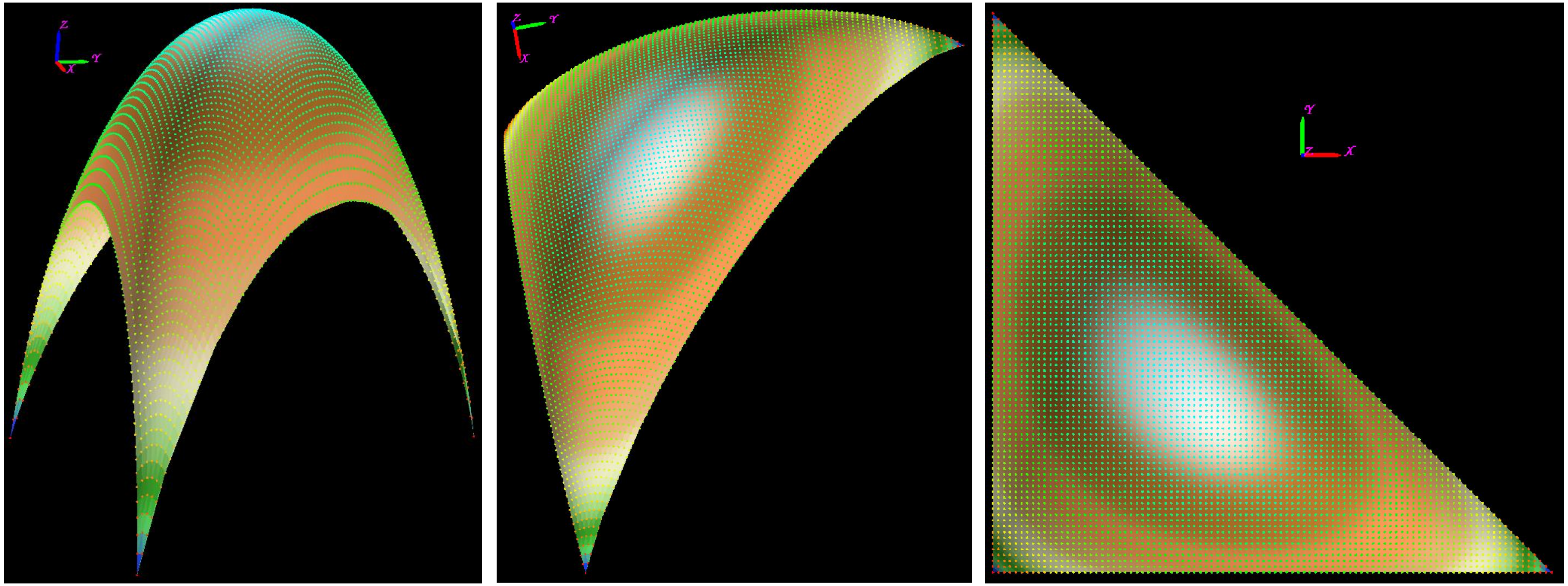
两点分布的熵

- 两点分布的熵 $H(X) = -\sum_{x \in X} p(x) \ln p(x) = -p \ln p - (1-p)\ln(1-p)$



继续思考：三点分布呢？

$$H(X) = -\sum_{x \in X} p(x) \ln p(x) = -p_1 \ln p_1 - p_2 \ln p_2 - (1 - p_1 - p_2) \ln(1 - p_1 - p_2)$$



组合数的关系

- 把N件物品分成k组，使得每组物品的个数分别为$n_1,n_2,\ldots,n_k$，$(n=n_1+n_2+\ldots+n_k)$，则不同的分组方法有$\frac{N!}{n_1!n_2!n_3!\cdots n_k!}$种。
- 记：$W(n_1,n_2,\cdots n_k)=\frac{N!}{n_1!n_2!n_3!\cdots n_k!}$
- 求 $H(n_1,n_2,\cdots n_k)=\frac{1}{N}\ln W=?$

公式推导 $N \to \infty \Rightarrow \ln N! \to N(\ln N - 1)$

$$H = \frac{1}{N}\ln\frac{N!}{\prod_{i=1}^{k} n_i!} = \frac{1}{N}\ln(N!) - \frac{1}{N}\sum_{i=1}^{k}\ln(n_i!)$$

$$\to (\ln N - 1) - \frac{1}{N}\sum_{i=1}^{k} n_i(\ln n_i - 1)$$

$$= (\ln N - 1) - \frac{1}{N}\left(\sum_{i=1}^{k} n_i \ln n_i - \sum_{i=1}^{k} n_i\right)$$

$$= \ln N - \frac{1}{N}\sum_{i=1}^{k} n_i \ln n_i = -\frac{1}{N}\left(\left(\sum_{i=1}^{k} n_i \ln n_i\right) - N \ln N\right)$$

$$= -\frac{1}{N}\sum_{i=1}^{k}(n_i \ln n_i - n_i \ln N) = -\frac{1}{N}\sum_{i=1}^{k}\left(n_i \ln\frac{n_i}{N}\right)$$

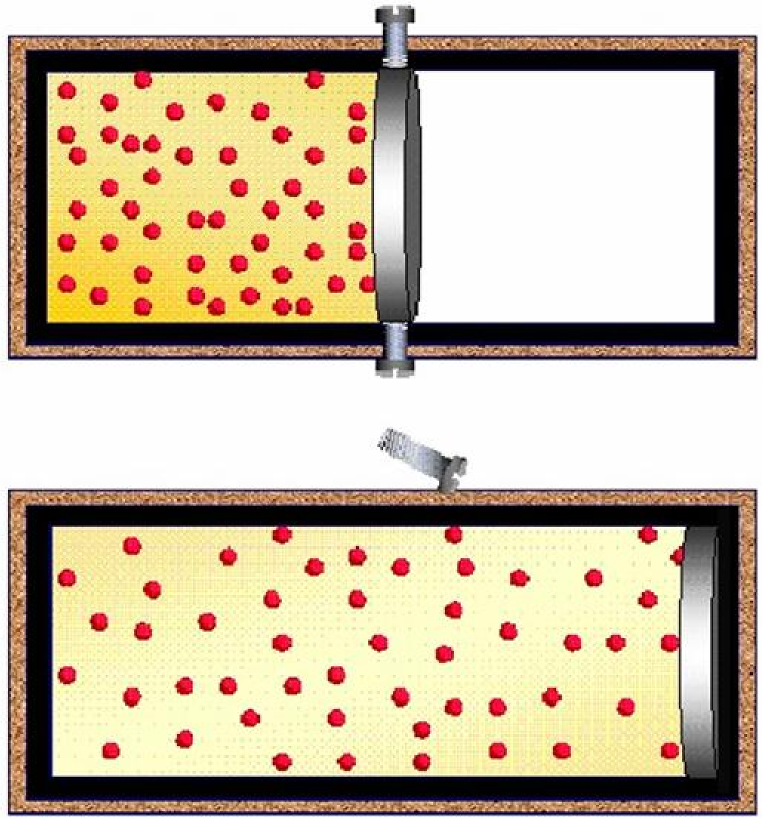
$$= -\sum_{i=1}^{k}\left(\frac{n_i}{N}\ln\frac{n_i}{N}\right) \to -\sum_{i=1}^{k}(p_i \ln p_i)$$

自封闭系统的运动总是倒向均匀分布

- 密封箱子中间放一隔板
- 隔板左边空间注入烟，右边真空

去掉隔板会怎样?

- 左边的烟就会自然(自发)地向右边扩散，最后均匀地占满整个箱体



思考：根据函数形式判断概率分布

- 正态分布的概率密度函数

$$p(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- 对数正态分布

$$\ln p(x)=\ln\frac{1}{\sqrt{2\pi}}-\ln\sigma-\frac{(x-\mu)^2}{2\sigma^2}=\alpha\cdot x^2+\beta\cdot x+\gamma$$

- 该分布的对数是关于随机变量x的二次函数
  - 根据计算过程的可逆性，若某对数分布能够写成随机变量二次形式，则该分布必然是正态分布。

举例

- Gamma分布的定义

$$p(x;\alpha,\beta)=\frac{\beta^{\alpha}x^{\alpha-1}e^{-\beta x}}{\Gamma(\alpha)}\ ,\quad x\geq 0(常系数\alpha,\beta>0)$$

- 对数形式

$$\ln p(x;\alpha,\beta)=\alpha\ln\beta+(\alpha-1)\ln x-\beta x-\ln\Gamma(\alpha)=A\cdot x+B\cdot\ln x+C$$

  - 若某对数分布能够写成随机变量一次项和对数项的和，则该分布必然是Gamma分布。

- 注：
  - Gamma函数：$\Gamma(\alpha)=\int_0^{\infty}t^{\alpha-1}e^{-t}dt \qquad \Gamma(n)=(n-1)!$
  - Gamma分布的期望为：$E(X)=\dfrac{\alpha}{\beta}$

最大熵的理解 $0 \le H(X) \le \log|X|$

- 熵是随机变量不确定性的度量，不确定性越大，熵值越大；
  - 若随机变量退化成定值，熵最小：为0
  - 若随机分布为均匀分布，熵最大。
- 以上是无条件的最大熵分布，若有条件呢？
  - 最大熵模型
- 思考：若只给定期望和方差的前提下，最大熵的分布形式是什么？

思考过程

- 建立目标函数

$$\arg\max_{p(x)} H(X) = -\sum_x p(x)\ln p(x) \qquad s.t.\begin{cases} E(X) = \mu \\ Var(X) = \sigma^2 \end{cases}$$

- 使用方差公式化简约束条件

$$Var(X) = E(X^2) - E^2(X)$$
$$\Rightarrow E(X^2) = E^2(X) + Var(X) = \mu^2 + \sigma^2$$

- 显然，此问题为带约束的极值问题。
  - Lagrange乘子法

建立Lagrange函数，求驻点

$$\arg\max_{p(x)} H(X) = -\sum_x p(x)\ln p(x) \quad s.t. \begin{cases} E(X) = \mu \\ E(X^2) = \mu^2 + \sigma^2 \end{cases}$$

$$L(p) = -\sum_x p(x)\ln p(x) + \lambda_1 (E(X) - \mu) + \lambda_2 (E(X^2) - \mu^2 - \sigma^2)$$

$$= -\sum_x p(x)\ln p(x) + \lambda_1 \left(\sum_x xp(x) - \mu \right) + \lambda_2 \left(\sum_x x^2 p(x) - \mu^2 - \sigma^2 \right)$$

$$\Rightarrow \frac{\partial L}{\partial p} = -\ln p(x) - 1 + \lambda_1 x + \lambda_2 x^2 \stackrel{\Delta}{==} 0 \Rightarrow \ln p(x) = \lambda_2 x^2 + \lambda_1 x - 1$$

- P(x)的对数是关于随机变量x的二次形式，所以，该分布p(x)必然是正态分布！

联合熵和条件熵

- 两个随机变量X，Y的联合分布，可以形成联合熵Joint Entropy，用H(X,Y)表示
- H(X,Y) – H(Y)
  - (X,Y)发生所包含的熵，减去Y单独发生包含的熵：在Y发生的前提下，X发生“新”带来的熵
  - 该式子定义为Y发生前提下，X的熵：
    - 条件熵H(X|Y)

推导条件熵的定义式

$$
\begin{aligned}
& H(X,Y)-H(Y) \\
&= -\sum_{x,y} p(x,y)\log p(x,y) + \sum_{y} p(y)\log p(y) \\
&= -\sum_{x,y} p(x,y)\log p(x,y) + \sum_{y}\left(\sum_{x} p(x,y)\right)\log p(y) \\
&= -\sum_{x,y} p(x,y)\log p(x,y) + \sum_{x,y} p(x,y)\log p(y) \\
&= -\sum_{x,y} p(x,y)\log \frac{p(x,y)}{p(y)} \\
&= -\sum_{x,y} p(x,y)\log p(x\,|\,y)
\end{aligned}
$$

根据条件熵的定义式，可以得到

$$H(X,Y)-H(X)=-\sum_{x,y}p(x,y)\log p(y|x)$$

$$=-\sum_{x}\sum_{y}p(x,y)\log p(y|x)$$

$$=-\sum_{x}\sum_{y}p(x)p(y|x)\log p(y|x)$$

$$=-\sum_{x}p(x)\sum_{y}p(y|x)\log p(y|x)$$

$$=\sum_{x}p(x)\left(-\sum_{y}p(y|x)\log p(y|x)\right)$$

$$=\sum_{x}p(x)H\left(Y|X=x\right)$$

相对熵

- 相对熵，又称互熵，交叉熵，鉴别信息，Kullback熵，Kullback-Leible散度等
- 设p(x)、q(x)是X中取值的两个概率分布，则p对q的相对熵是

$$D(p \| q)=\sum_{x} p(x) \log \frac{p(x)}{q(x)}=E_{p(x)} \log \frac{p(x)}{q(x)}$$

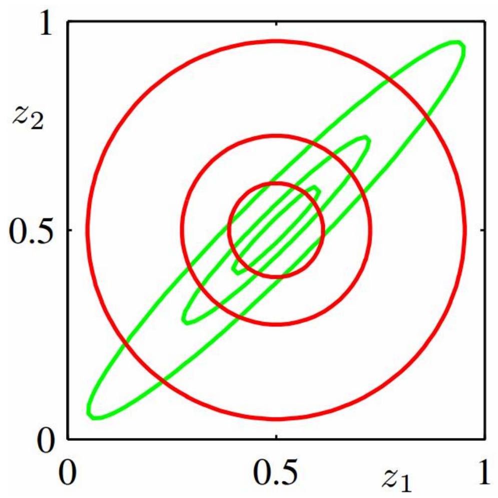
- 说明：
  - 相对熵可以度量两个随机变量的“距离”
    - 在“贝叶斯网络”、“变分推导”等章节会再次遇到
  - 一般的，D(p||q) ≠D(q||p)
  - D(p||q)≥0、 D(q||p) ≥0：凸函数中的Jensen不等式

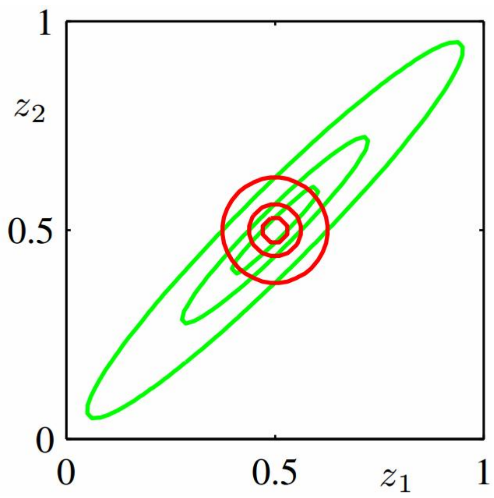
思考

- 假定已知随机变量P，求相对简单的随机变量Q，使得Q尽量接近P
  - 方法：使用P和Q的K-L距离。
  - 难点：K-L距离是非对称的，两个随机变量应该谁在前谁在后呢？
- 假定使用KL(Q||P)，为了让距离最小，则要求在P为0的地方，Q尽量为0。会得到比较“窄”的分布曲线；
- 假定使用KL(P||Q)，为了让距离最小，则要求在P不为0的地方，Q也尽量不为0。会得到比较“宽”的分布曲线；

两个KL散度的区别

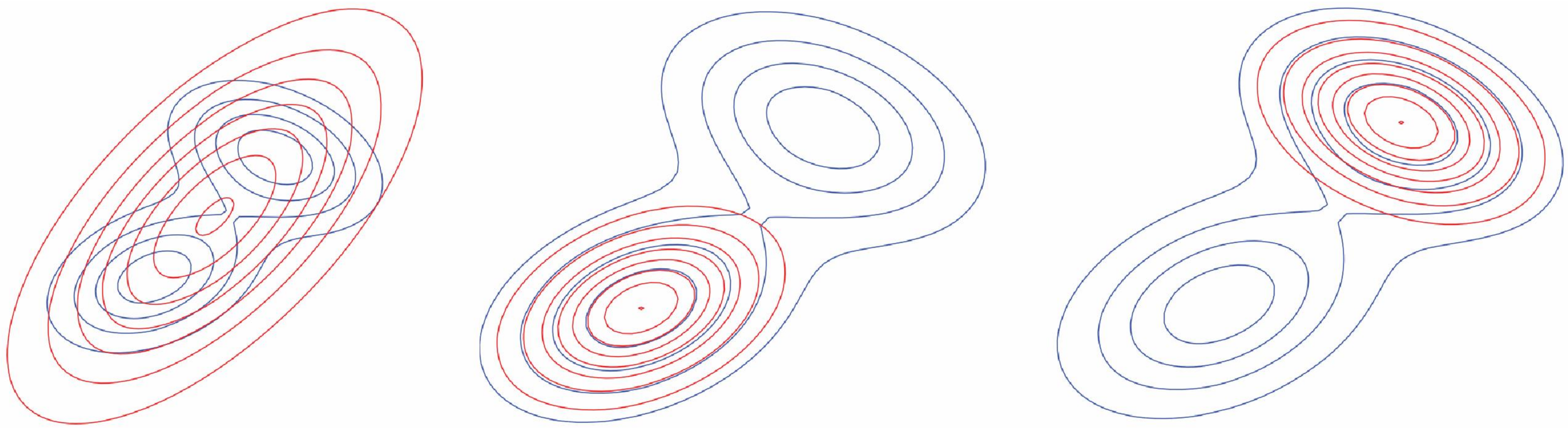
- 绿色曲线是真实分布p的等高线；红色曲线是使用近似$p(z_1,z_2)=p(z_1)p(z_2)$得到的等高线
  - 左：KL(p||q)：zero avoiding
  - 右：KL(q||p)：zero forcing





两个KL散度的区别

- 蓝色曲线是真实分布p的等高线；红色曲线是单模型近似分布q的等高线。
  - 左：KL(p||q)：q趋向于覆盖p
  - 中、右：KL(q||p)：q能够锁定某一个峰值



互信息

- 两个随机变量X，Y的互信息，定义为X，Y的联合分布和独立分布乘积的相对熵。
- I(X,Y)=D(P(X,Y) || P(X)P(Y))

$$I(X,Y)=\sum_{x,y}p(x,y)\log\frac{p(x,y)}{p(x)p(y)}$$

计算条件熵的定义式：H(X)-I(X,Y)

$$
\begin{aligned}
& H(X)-I(X,Y) \\
& =-\sum_{x} p(x)\log p(x)-\sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \\
& =-\sum_{x}\left(\sum_{y} p(x,y)\right)\log p(x)-\sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \\
& =-\sum_{x,y} p(x,y)\log p(x)-\sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \\
& =-\sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(y)} \\
& =-\sum_{x,y} p(x,y)\log p(x\mid y) \\
& =H(X\mid Y)
\end{aligned}
$$

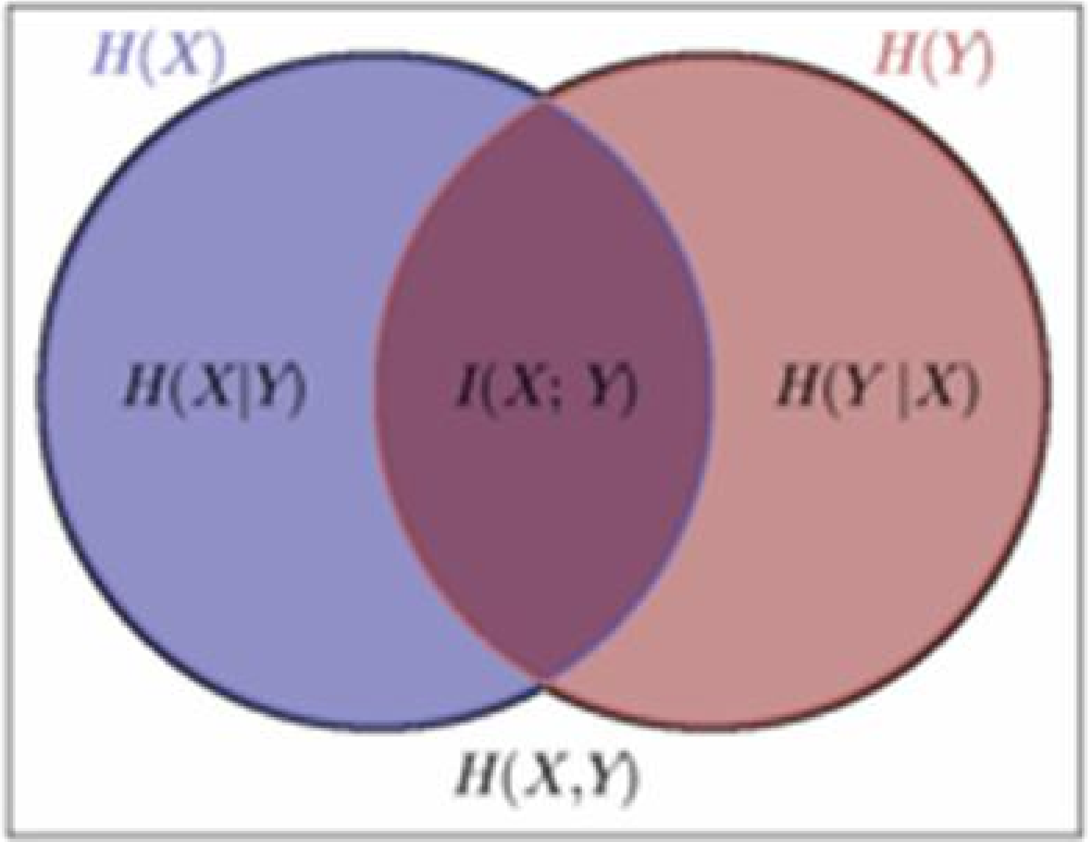
根据条件熵的定义式，可以得到

$$
\begin{aligned}
H(X,Y)-H(X) &= -\sum_{x,y} p(x,y)\log p(y\mid x) \\
&= -\sum_{x}\sum_{y} p(x,y)\log p(y\mid x) \\
&= -\sum_{x}\sum_{y} p(x)p(y\mid x)\log p(y\mid x) \\
&= -\sum_{x} p(x)\sum_{y} p(y\mid x)\log p(y\mid x) \\
&= \sum_{x} p(x)\left(-\sum_{y} p(y\mid x)\log p(y\mid x)\right) \\
&= \sum_{x} p(x)H\left(Y\mid X=x\right)
\end{aligned}
$$

整理得到的等式

- $H(X|Y) = H(X,Y) - H(Y)$
  - 条件熵定义
- $H(X|Y) = H(X) - I(X,Y)$
  - 根据互信息定义展开得到
  - 有些文献将$I(X,Y)=H(Y) – H(Y|X)$作为互信息的定义式
- 对偶式
  - $H(Y|X)= H(X,Y) - H(X)$
  - $H(Y|X)= H(Y) - I(X,Y)$
- $I(X,Y)= H(X) + H(Y) - H(X,Y)$
  - 有些文献将该式作为互信息的定义式
- 试证明：$H(X|Y) \leqslant H(X)$，$H(Y|X) \leqslant H(Y)$

强大的Venn图：帮助记忆



思考题：天平与假币

- 有13枚硬币，其中有1枚是假币，但不知道是重还是轻。现给定一架没有砝码的天平，问**至少**需要多少次称量才能找到这枚假币？
  - 答：3次。
  - 如何称量？如何证明？

最大熵模型的原则

- 承认已知事物(知识)
- 对未知事物不做任何假设，没有任何偏见

例如

- 已知：
  - “学习”可能是动词，也可能是名词。
  - “学习”可以被标为主语、谓语、宾语、定语……
- 令x1表示“学习”被标为名词，x2表示“学习”被标为动词。
- 令y1表示“学习”被标为主语，y2表示被标为谓语，y3表示宾语，y4表示定语。得到下面的表示：

$$p(x_1)+p(x_2)=1 \qquad \sum_{i=1}^{4} p(y_i)=1$$

- 根据无偏原则

$$p(x_1)=p(x_2)=0.5$$

$$p(y_1)=p(y_2)=p(y_3)=p(y_4)=0.25$$

引入新知识

- 若已知："学习"被标为定语的可能性很小，只有0.05 $p(y_4)=0.05$

- 仍然坚持无偏原则：

$$p(x_1)=p(x_2)=0.5$$

$$p(y_1)=p(y_2)=p(y_3)=\frac{0.95}{3}$$

再次引入新知识

- 当“学习”被标作动词的时候，它被标作谓语的概率为0.95

$$p(y_2 \mid x_1) = 0.95$$

- 除此之外，仍然坚持无偏见原则，尽量使概率分布平均。
- 问：怎么样能尽量无偏见的分布？

最大熵模型Maximum Entropy

- 概率平均分布 等价于 熵最大
- 问题转化为：计算X和Y的分布，使H(Y|X)达到最大值，并且满足条件

$$p(x_1)+p(x_2)=1$$

$$\sum_{i=1}^{4} p(y_i)=1$$

$$p(y_4)=0.05$$

$$p(y_2 \mid x_1)=0.95$$

最大熵模型Maxent

$$\max H(Y|X) = -\sum_{\substack{x\in\{x_1,x_2\}\\ y\in\{y_1,y_2,y_3,y_4\}}} p(x,y)\log p(y|x)$$

$$p(x_1) + p(x_2) = 1$$

$$p(y_1) + p(y_2) + p(y_3) + p(y_4) = 1$$

$$p(y_4) = 0.05$$

$$p(y_2|x_1) = 0.95$$

Maxent的一般式

- 一般模型：

$$\max_{p\in P} H(Y\mid X) = -\sum_{(x,y)} p(x,y)\log p(y\mid x)$$

- P={p | p是X上满足条件的概率分布}
  - 注意区分这里的p和P。

特征(Feature)和样本(Sample)

- 特征：(x,y)
  - y:这个特征中需要确定的信息
  - x:这个特征中的上下文信息
- 样本：关于某个特征(x,y)的样本，特征所描述的语法现象在标准集合里的分布：
  - (xi,yi)对
  - yi是y的一个实例
  - xi是yi的上下文
  - (x1,y1) (x2,y2) (x3,y3)……

特征函数

- 关于某个特征(x,y)的样本
  - y:这个特征中需要确定的信息
  - x:这个特征中的上下文信息
- 特征函数：对于一个特征(x0,y0)，定义特征函数：

$$f(x,y)=\begin{cases}1 & x=x_0\text{且}y=y_0\\ 0 & otherwises\end{cases}$$

- 对于一个特征(x0,y0)，在样本中的期望值是：

$$\overline{p}(f)=\sum_{(x_i,y_i)}\overline{p}(x,y)f(x,y)$$

  - $\overline{p}(x,y)$ 是(x,y)在样本中出现的概率

条件Constraints

- 对每一个特征(x,y)，模型所建立的条件概率分布要与训练样本表现出来的分布相同。
- 假设样本的分布是(已知)：

$$\overline{p}(x) = x\text{出现的概率}$$

$$\overline{p}(x,y) = xy\text{出现的概率}$$

$$\overline{p}(f) = \text{特征}f\text{在样本中的期望值}$$

条件Constraints

- 特征f在模型中的期望值：

$$
\begin{aligned}
p(f) &= \sum_{(x_i,y_i)} p(x_i,y_i) f(x_i,y_i) \\
&= \sum_{(x_i,y_i)} p(y_i \mid x_i) p(x_i) f(x_i,y_i) \\
&= \sum_{(x_i,y_i)} p(y_i \mid x_i) \overline{p}(x_i) f(x_i,y_i)
\end{aligned}
$$

$$
p(f) = \overline{p}(f)
$$

最大熵模型：最大条件熵

- NLP模型：

$$p^* = \arg\max_{p\in P} H(Y|X) = -\sum_{(x,y)} p(x,y)\log p(y|x)$$

- 可行域：p是y|x的概率分布并且对训练样本，对任意给定的特征fi：

$$p(f) = \overline{p}(f)$$

最大熵模型在NLP中的完整提法

$$p^* = \arg\max_{p \in P} H(Y \mid X)$$

$$= -\sum_{(x,y)} p(x,y) \log p(y \mid x)$$

$$= -\sum_{(x,y)} p(y \mid x)\bar{p}(x) \log p(y \mid x)$$

$$P = \left\{ p(y \mid x) \middle| \forall f_i : \sum_{(x,y)} p(y \mid x)\bar{p}(x) f_i(x,y) = \sum_{(x,y)} \bar{p}(x,y) f_i(x,y),\ \forall x : \sum_{y} p(y \mid x) = 1 \right\}$$

最大熵模型总结

定义条件熵 $$H(y|x) = -\sum_{(x,y)\in z} p(y,x)\log p(y|x)$$

模型目的 $$p^*(y|x) = \arg\max_{p(y|x)\in P} H(y|x)$$

定义特征函数 $$f_i(x,y)\in\{0,1\} \quad i = 1,2,\cdots,m$$

约束条件 $$\sum_{y\in Y} p(y|x) = 1 \qquad (1)$$

$$E(f_i) = \tilde{E}(f_i) \quad i = 1,2,\cdots,m \qquad (2)$$

$$\tilde{E}(f_i) = \sum_{(x,y)\in z} \tilde{p}(x,y) f_i(x,y) = \frac{1}{N}\sum_{(x,y)\in T} f_i(x,y) \qquad N = |T|$$

$$E(f_i) = \sum_{(x,y)\in z} p(x,y) f_i(x,y) = \sum_{(x,y)\in z} p(x)p(y|x)f_i(x,y)$$

求解Maxent模型

- 该条件约束优化问题的Lagrange函数

$$\Lambda(p,\vec{\lambda}) = H(y|x) + \sum_{i=1}^{m} \lambda_i \left(E(f_i) - \tilde{E}(f_i) \right) + \lambda_{m+1} \left(\sum_{y \in Y} p(y|x) - 1 \right)$$

- 分析：
  - 已知若干条件，要求若干变量的值使到目标函数(熵)最大
- 数学本质：
  - 最优化问题(Optimization Problem)
- 条件：线性、等式
- 目标函数：非线性
- 非线性规划(线性约束)
  - non-linear programming with linear constraints

最大熵模型MaxEnt的目标拉格朗日函数L

$$L=\left(-\sum_{(x,y)}p(y|x)\overline{p}(x)\log p(y|x)\right)$$

$$+\left(\sum_i\lambda_i\sum_{(x,y)}f_i(x,y)\left[p(y|x)\overline{p}(x)-\overline{p}(x,y)\right]\right)+v_0\left[\sum_y p(y|x)-1\right]$$

$$\frac{\partial L}{\partial p(y|x)}=\overline{p}(x)(-\log p(y|x)-1)+\sum_i\lambda_i\overline{p}(x)f_i(x,y)+v_0\overset{\Delta}{=}0$$

$$\Rightarrow\left(\text{令}\lambda_0=\frac{v_0}{\overline{p}(x)}\right)\Rightarrow$$

$$p^*(y|x)=\exp\left(\sum_i\lambda_i f_i(x,y)+\lambda_0-1\right)=\frac{1}{\exp(1-\lambda_0)}\cdot\exp\left(\sum_i\lambda_i f_i(x,y)\right)$$

λ_0与ν_0仅相差常系数，后面的推导将直接以λ_0代替ν_0

“泛函求导”

- 通过条件熵最大，能够得到关于未知概率分布$p(y|x)$的目标函数L，而$p(y|x)$本质是一个函数(随机变量)，从而，目标函数L是关于函数的函数——泛函。
  - 计算L关于p的导数，进而让导数为0，可求得驻点，即得到关于$p(y|x)$的方程F
  - 至于如何根据方程F计算$p(y|x)$，是参数优化问题，目前先解决建立F的问题。
    - 根据方程F求最优的$p(y|x)$，是用的IIS算法。
- 可以通过“朴素”的办法计算泛函的导数。

泛函求导——“类比”

- 根据积分的定义，很容易得知以下两个式子：

$$F(x)=\int_0^x f(x)dx \Rightarrow F'(x)=f(x) \qquad (1)$$

$$F(x)=\int_0^x f(t(x))dx \Rightarrow F'(x)=f(t(x))t'(x) \qquad (2)$$

  - (1)式是函数的不定积分定义
  - (2)式中的t是关于x的函数

- 将其中的积分号变成加和符号，即得到如下式子：

$$F(x)=\sum_x f(x) \Rightarrow F'(x)=f(x) \qquad (3)$$

$$F(x)=\sum_x f(t(x)) \Rightarrow F'(x)=f(t(x))t'(x) \qquad (4)$$

最优解形式Exponential：求偏导，等于0

$$p^*(y|x)=\frac{1}{\exp(1-\lambda_0)}\cdot\exp\left(\sum_i\lambda_i f_i(x,y)\right)$$

- 上式通过直接求偏导所得到的p*是没有归一化的，求归一化因子：

$$p^*(y|x)=\frac{1}{Z_\lambda(x)}\exp\left(\sum_i\lambda_i f_i(x,y)\right)$$

$$\sum_y\frac{1}{Z_\lambda(x)}\exp\left(\sum_i\lambda_i f_i(x,y)\right)=1\Rightarrow Z_\lambda(x)=\sum_y\exp\left(\sum_i\lambda_i f_i(x,y)\right)$$

最大熵模型与Logistic/Softmax回归

- Logistic/Softmax回归的后验概率：

$$\begin{cases} h(c=1\mid x;\theta)=\dfrac{1}{1+e^{-\theta^T x}}=\dfrac{e^{\theta^T x}}{e^{\theta^T x}+1}\propto e^{\theta^T x} \\ h(c=0\mid x;\theta)=\dfrac{e^{-\theta^T x}}{1+e^{-\theta^T x}}=\dfrac{1}{e^{\theta^T x}+1}\propto 1 \end{cases}$$

$$h(c=k\mid x;\theta)=\frac{e^{\theta_k^T x}}{\sum_{j=1}^{K} e^{\theta_j^T x}},\quad k=1,2,\cdots,K$$

- 最大熵模型的后验概率

$$p^*(y\mid x)=\frac{1}{Z_\lambda(x)}\exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

最大似然估计Maximum likelihood estimate

- 找出与样本的分布最接近的概率分布模型。
- 简单的例子
  - 10次抛硬币的结果是：正正反正正正反反正正
- 假设p是每次抛硬币结果为正的概率。则：
- 得到这样的实验结果的概率是：

$$
\begin{aligned}
P &= pp(1-p)ppp(1-p)(1-p)pp \\
&= p^7(1-p)^3
\end{aligned}
$$

极大似然估计MLE

- 目标函数：$\max P = \max_{0\le p\le 1} p^7(1-p)^3$
- 最优解是：p=0.7
  - 思考：如何求解？
- 一般形式：

$$L_{\bar{p}} = \prod_x p(x)^{\bar{p}(x)}$$

$p(x)$模型是估计的概率分布

$\bar{p}(x)$是实验结果的分布

$$\log L(\theta_1,\theta_2,\cdots\theta_k) = \sum_{i=1}^{n} \log f(x_i;\theta_1,\theta_2,\cdots\theta_k)$$

取对数

- 似然函数取对数：

$$L_{\bar{p}} = \log\left(\prod_x p(x)^{\bar{p}(x)}\right) = \sum_x \bar{p}(x) \log p(x)$$

$$L_{\bar{p}}(p) = \sum_{x,y} \bar{p}(x,y) \log p(x,y)$$

$$= \sum_{x,y} \bar{p}(x,y) \log\left[\bar{p}(x) p(y \mid x)\right]$$

$$= \sum_{x,y} \bar{p}(x,y) \log p(y \mid x) + \sum_{x,y} \bar{p}(x,y) \bar{p}(x)$$

- 第二项是常数，可忽略

MLE与条件熵

- 此目标式，与条件熵具有相同的形式。

$$L_{\bar{p}}(p)=\sum_{x,y}\overline{p}(x,y)\log p(y\mid x)$$

- 既然函数式相同，极有可能二者殊途同归，目标函数是相同的。
  - 演示推导

$$L=\left(-\sum_{(x,y)}p(y\mid x)\overline{p}(x)\log p(y\mid x)\right)+\left(\sum_i\lambda_i\sum_{(x,y)}f_i(x,y)\left[p(y\mid x)\overline{p}(x)-\overline{p}(x,y)\right]\right)+v_0\left[\sum_y p(y\mid x)-1\right]$$

求L的对偶函数

- 最优解 $p_\lambda(y|x)=\frac{1}{Z_\lambda(x)}\exp\left(\sum_i \lambda_i f_i(x,y)\right)$ 代入L，得到关于λ的函数

$$
\begin{aligned}
L(\lambda)\\
&=-\sum_{x,y} p(y|x)\overline{p}(x)\log p(y|x)+\sum_{i=1}^{k}\lambda_i\sum_{x,y} f_i(x,y)\left[p(y|x)\overline{p}(x)-\overline{p}(x,y)\right]+v_0\left[\sum_y p(y|x)-1\right]\\
&=-\sum_{x,y} p_\lambda(y|x)\overline{p}(x)\log p_\lambda(y|x)+\sum_{i=1}^{k}\lambda_i\sum_{x,y} f_i(x,y)\left[p_\lambda(y|x)\overline{p}(x)-\overline{p}(x,y)\right]\\
&=-\sum_{x,y}\overline{p}(x)p_\lambda(y|x)\log p_\lambda(y|x)+\sum_{i=1}^{k}\overline{p}(x)p_\lambda(y|x)\lambda_i\sum_{x,y} f_i(x,y)-\sum_{i=1}^{k}\overline{p}(x,y)\lambda_i\sum_{x,y} f_i(x,y)\\
&=-\sum_{x,y}\overline{p}(x)p_\lambda(y|x)\log p_\lambda(y|x)+\sum_{x,y}\overline{p}(x)p_\lambda(y|x)\sum_{i=1}^{k}\lambda_i f_i(x,y)-\sum_{i=1}^{k}\overline{p}(x,y)\lambda_i\sum_{x,y} f_i(x,y)\\
&=\sum_{x,y}\overline{p}(x)p_\lambda(y|x)\log Z_\lambda(x)-\sum_{i=1}^{k}\overline{p}(x,y)\sum_{x,y}\lambda_i f_i(x,y)
\end{aligned}
$$

将最优解 $p_\lambda(y|x)=\frac{1}{Z_\lambda(x)}\exp\left(\sum_i \lambda_i f_i(x,y)\right)$ 带入MLE

$$L_{\bar{p}}(p)=\sum_{x,y}\bar{p}(x,y)\log p(y|x)$$

$$=\sum_{x,y}\bar{p}(x,y)\left(\sum_{i=1}^{n}\lambda_i f_i(x,y)-\log Z_\lambda(x)\right)$$

$$=\sum_{x,y}\bar{p}(x,y)\sum_{i=1}^{n}\lambda_i f_i(x,y)-\sum_{x,y}\bar{p}(x,y)\log Z_\lambda(x)$$

$$=\sum_{x,y}\bar{p}(x,y)\sum_{i=1}^{n}\lambda_i f_i(x,y)-\sum_{x}\bar{p}(x)\log Z_\lambda(x)$$

$$=-\sum_{x,y}\bar{p}(x)p_\lambda(y|x)\log Z_\lambda(x)+\sum_{i=1}^{k}\bar{p}(x,y)\sum_{x,y}\lambda_i f_i(x,y)$$

结论

- 可以看到，二者的右端具有完全相同的目标函数。
- 根据MLE的正确性，可以断定：最大熵的解(无偏的对待不确定性)同时是最符合样本数据分布的解，进一步证明了最大熵模型的合理性。
- 做点思考：
  - 熵：不确定度
  - 似然：与知识的吻合程度
  - 最大熵模型：对不确定度的无偏分配
  - 最大似然估计：对知识的无偏理解

知识 = 不确定度的补集

λ 的求解

- 因为没有显式的解析式，使用IIS计算最大熵模型的数值解
  - IIS是目前最大熵模型的最优化算法，优于梯度下降算法
    - IIS：Improved Iterative Scaling 改进的迭代尺度算法
    - 具体内容在本PPT最后篇末的附录中。
  - 但工业界使用最多的仍然是梯度下降算法。
    - Logistic回归/Softmax回归

Softmax参数求解

- 目标函数：

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\left\{y^{(i)} = j\right\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right]$$

- 梯度：

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[x^{(i)}\left(1\{y^{(i)} = j\} - p(y^{(i)} = j | x^{(i)}; \theta)\right)\right]$$

IIS的思想

- 假设最大熵模型当前的参数向量是λ，希望找到新的参数向量$\lambda+\delta$，使得模型的对数似然函数值L增加。重复这一过程，直至找到对数似然函数的最大值。

再次强调

- 熵是描述不确定度的
- 知识是不确定度的补集
  - 不确定度越小，模型越准确。
- 直观的过程：
  - 什么特征都不限定：熵最大
  - 加一个特征：熵少一点
    - Condition Reduces Entropy (C.R.E.)
  - 加的特征越多，熵越少

总结

- MaxEnt已经是比较成功的一个NLP模型，并获得广泛应用
- 从信息论获得启发(1948-)：自然语言处理也是信息处理的一种。
  - 词性标注也可以看作一种编码的过程?
  - 思考：身边的哪些问题，可以看做或类别编码过程?
- 求极值的技术手段：Lagrange对偶问题
- 最大熵模型，涉及了很多前序的数学知识
  - 事实上，机器学习本身就是多种手段的综合应用。

参考文献

- Elements of Information Theory (Cover & Thomas)
- http://ufldl.stanford.edu/wiki/index.php/Softmax%E5%9B%9E%E5%BD%92
- A maximum entropy approach to natural language processing (Adam Berger)
- A Brief MaxEnt Tutorial (Adam Berger)
- Learning to parse natural language with maximum entropy models (Adwait Ratnaparkhi)
- A simple Introduction to Maximum Entropy Models for Natural Language Processing (Adwait Ratnaparkhi)
- 统计学习方法，李航著，清华大学出版社，2012年

我们在这里

- 七月算法 http://www.julyedu.com/
  - 视频/课程/社区
- 七月题库APP：Android/iOS
  - http://www.julyapp.com/
- 微博
  - @研究者July
  - @七月题库
  - @邹博_机器学习
- 微信公众号
  - julyedu

感谢大家！

恳请大家批评指正！

附：IIS算法公式推导

改进的迭代尺度法IIS

$$p^*(y \mid x) = \frac{1}{Z_\lambda(x)} e^{\sum_i \lambda_i f_i(x,y)}$$

$$Z_\lambda(x) = \sum_y e^{\sum_i \lambda_i f_i(x,y)}$$

$$L_{\overline{p}}(p) = \sum_{x,y} \overline{p}(x,y) \sum_{i=1}^{n} \lambda_i f_i(x,y) - \sum_x \overline{p}(x) \log Z_\lambda(x)$$

$$L_{\bar{p}}(p)=\sum_{x,y}\bar{p}(x,y)\sum_{i=1}^{n}\lambda_i f_i(x,y)-\sum_{x}\bar{p}(x)\log Z_\lambda(x)$$

$$\begin{aligned}
&L(\lambda+\delta)-L(\lambda)\\
&=\sum_{x,y}\bar{p}(x,y)\sum_{i=1}^{n}\delta_i f_i(x,y)-\sum_{x}\bar{p}(x)\log\frac{Z_{\lambda+\delta}(x)}{Z_\lambda(x)}\\
&\ge\sum_{x,y}\bar{p}(x,y)\sum_{i=1}^{n}\delta_i f_i(x,y)+1-\sum_{x}\bar{p}(x)\frac{Z_{\lambda+\delta}(x)}{Z_\lambda(x)}\\
&=\sum_{x,y}\bar{p}(x,y)\sum_{i=1}^{n}\delta_i f_i(x,y)+1-\sum_{x}\bar{p}(x)\frac{Z_{\lambda+\delta}(x)}{Z_\lambda(x)}\\
&=\sum_{x,y}\bar{p}(x,y)\sum_{i=1}^{n}\delta_i f_i(x,y)+1-\sum_{x}\bar{p}(x)\exp\left(\sum_{i=1}^{n}\delta_i f_i(x,y)\right)\\
&=\sum_{x,y}\bar{p}(x,y)\sum_{i=1}^{n}\delta_i f_i(x,y)+1-\sum_{x}\bar{p}(x)\sum_{y}p_\lambda(y\mid x)\exp\left(\sum_{i=1}^{n}\delta_i f_i(x,y)\right)
\end{aligned}$$

注：$-\ln x \geqslant 1-x$, $x>0$

$$Z_\lambda(x)=\sum_{y}e^{\sum_i \lambda_i f_i(x,y)}$$

针对凸函数f(x)=e^x 使用Jensen不等式

$$f^{\#}(x,y)=\sum_i f_i(x,y)$$

$$A(\delta\,|\,\lambda)=\sum_{x,y}\overline{p}(x,y)\sum_{i=1}^{n}\delta_i f_i(x,y)+1-\sum_x \overline{p}(x)\sum_y p_\lambda(y\,|\,x)\exp\left(\sum_{i=1}^{n}\delta_i f_i(x,y)\right)$$

$$=\sum_{x,y}\overline{p}(x,y)\sum_{i=1}^{n}\delta_i f_i(x,y)+1-\sum_x \overline{p}(x)\sum_y p_\lambda(y\,|\,x)\exp\left(f^{\#}(x,y)\sum_{i=1}^{n}\frac{\delta_i f_i(x,y)}{f^{\#}(x,y)}\right)$$

$$\geq\sum_{x,y}\overline{p}(x,y)\sum_{i=1}^{n}\delta_i f_i(x,y)+1-\sum_x \overline{p}(x)\sum_y p_\lambda(y\,|\,x)\sum_{i=1}^{n}\frac{f_i(x,y)}{f^{\#}(x,y)}\exp\left(\delta_i f^{\#}(x,y)\right)$$

对该下界求偏导，令为0，求出 δ

$$B(\delta|\lambda)=\sum_{x,y}\overline{p}(x,y)\sum_{i=1}^{n}\delta_i f_i(x,y)+1-\sum_x\overline{p}(x)\sum_y p_\lambda(y|x)\sum_{i=1}^{n}\frac{f_i(x,y)}{f^{\#}(x,y)}\exp\left(\delta_i f^{\#}(x,y)\right)$$

$$\frac{\partial B(\delta|\lambda)}{\partial\delta_i}=\sum_{x,y}\overline{p}(x,y)f_i(x,y)-\sum_x\overline{p}(x)\sum_y p_\lambda(y|x)f_i(x,y)\exp\left(\delta_i f^{\#}(x,y)\right)$$

$$=\sum_{x,y}\overline{p}(x,y)f_i(x,y)-\sum_{x,y}\overline{p}(x)p_\lambda(y|x)f_i(x,y)\exp\left(\delta_i f^{\#}(x,y)\right)$$

$$=E_{\overline{p}}(f_i)-\sum_{x,y}\overline{p}(x)p_\lambda(y|x)f_i(x,y)\exp\left(\delta_i f^{\#}(x,y)\right)$$

令梯度为0，得到：

$$\sum_{x,y}\overline{p}(x)p_\lambda(y|x)f_i(x,y)\exp\left(\delta_i f^{\#}(x,y)\right)-E_{\overline{p}}(f_i)=0$$

δ 的求法：若 $f^{\#}(x,y)=M$ 为常数

$$\delta_i = \frac{1}{M}\log\frac{E_{\overline{p}}\left(f_i\right)}{E_p\left(f_i\right)}$$

δ的求法：若$f^{\#}(x,y)$不是常数

- 令 $g(\delta_i)=\sum_{x,y}\overline{p}(x)p_\lambda(y|x)f_i(x,y)\exp\left(\delta_i f^{\#}(x,y)\right)-E_{\overline{p}}(f_i)$
  - 转换为求g(δ)=0的根。
- 牛顿法：

$$\delta_i^{(k+1)}=\delta_i^{(k)}-\frac{g(\delta_i^{(k)})}{g'(\delta_i^{(k)})}$$

- 说明：
  - 因为需要计算g(δ)=0的根而不是求g(δ)的极小值，上式是函数值除以一阶导，而不是一阶导除以二阶导；
  - 实践中，可采用拟牛顿BFGS或者L-BFGS的方法。

最终解

- 上述求解过程中得到的权值λ，回代到下式中，即得到最大熵模型的最优估计。

$$p^*(y|x)=\frac{1}{Z_\lambda(x)}e^{\sum_i \lambda_i f_i(x,y)}$$

$$Z_\lambda(x)=\sum_y e^{\sum_i \lambda_i f_i(x,y)}$$