法律声明

- 本课件包括演示文稿、示例、代码、题库、视频和声音等内容，小象学院和主讲老师拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意及内容，我们保留一切通过法律手段追究违反者的权利。

- 课程详情请咨询
  - 微信公众号：小象
  - 新浪微博：ChinaHadoop

最大熵模型

小象学院
ChinaHadoop.cn

邹博

主要内容

- 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) ≥0
- 掌握最大熵的内涵Maxent
  - Maximum Entropy Models
- 最大熵/MLE在独立成分分析ICA中的应用
- 最大熵模型Maxent和最大似然估计MLE的关系

复习：标量对方阵的导数

- A为n×n的矩阵，|A|为A的行列式，计算 $\frac{\partial |A|}{\partial A}$
- 解：根据等式 $\forall 1 \le i \le n,\ |A| = \sum_{j=1}^{n} a_{ij} \cdot (-1)^{i+j} M_{ij}$
- 有 $\frac{\partial |A|}{\partial a_{ij}} = \frac{\partial \left(\sum_{j=1}^{n} a_{ij} \cdot (-1)^{i+j} M_{ij} \right)}{\partial a_{ij}} = (-1)^{i+j} M_{ij} = A_{ji}^{*}$
- 从而：$\frac{\partial |A|}{\partial A} = \left(A^{*}\right)^{T} = |A| \cdot \left(A^{-1}\right)^{T}$
  - 依据 $A \cdot A^{*} = |A| \cdot I$，第二个等式成立；

随机变量函数的分布

- 给定X的概率密度函数$f_X(x)$，若Y=aX，a是某正实数，求Y的概率密度$f_Y(y)$。
- 记X的累计概率为$F_X(x)$，Y的累计概率为$F_Y(y)$

$$F_Y(y)=P\{Y\le y\}=P\{aX\le y\}=P\{X\le y/a\}=F_X(y/a)$$

$$\Rightarrow f_Y(y)=\frac{dF_X(y/a)}{dy}=f_X(y/a)\cdot\frac{1}{a}$$

即：$f_Y(y)=\frac{1}{a}f_X\left(\frac{1}{a}y\right)$

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：$H\left(\vec{p}\right)=-\sum_{i=1}^{6}p_i\ln p_i$
- 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：

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

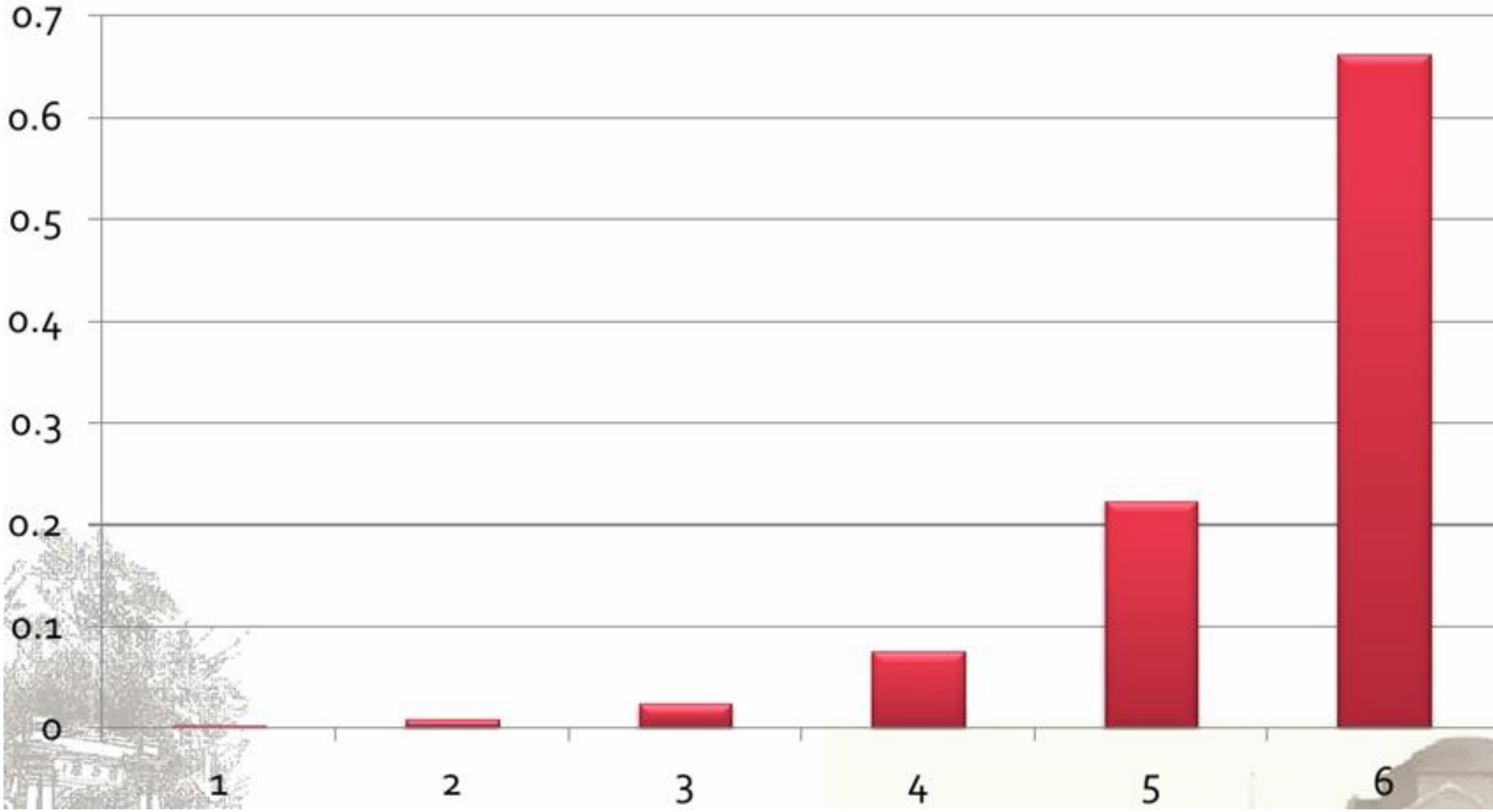
- 求解：

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

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

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

预测结果



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

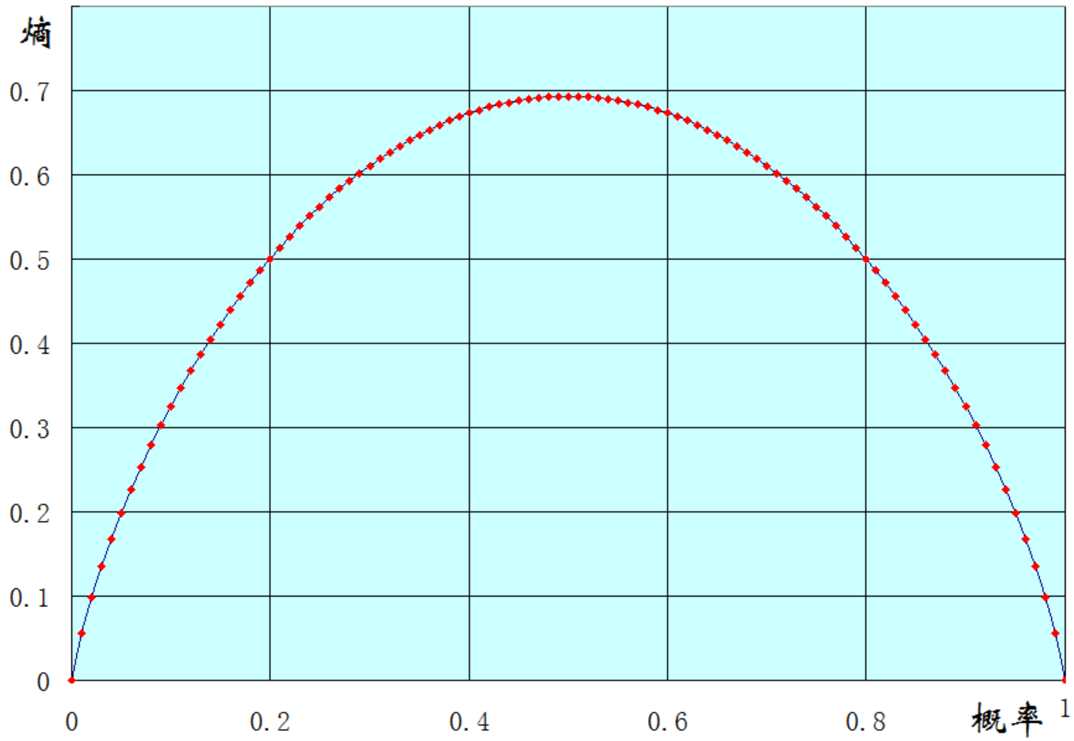
- 对随机事件的信息量求期望，得熵的定义：

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

  - 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)

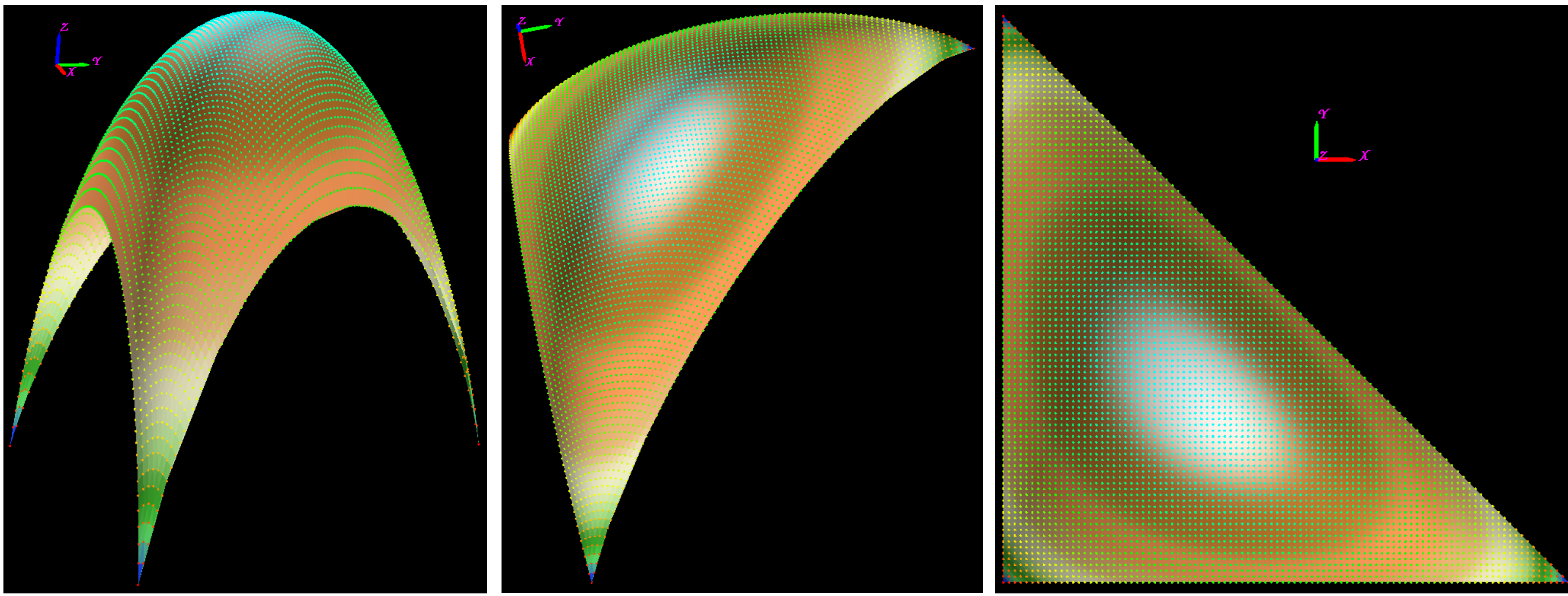
两点分布的熵

- 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？

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



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

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

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

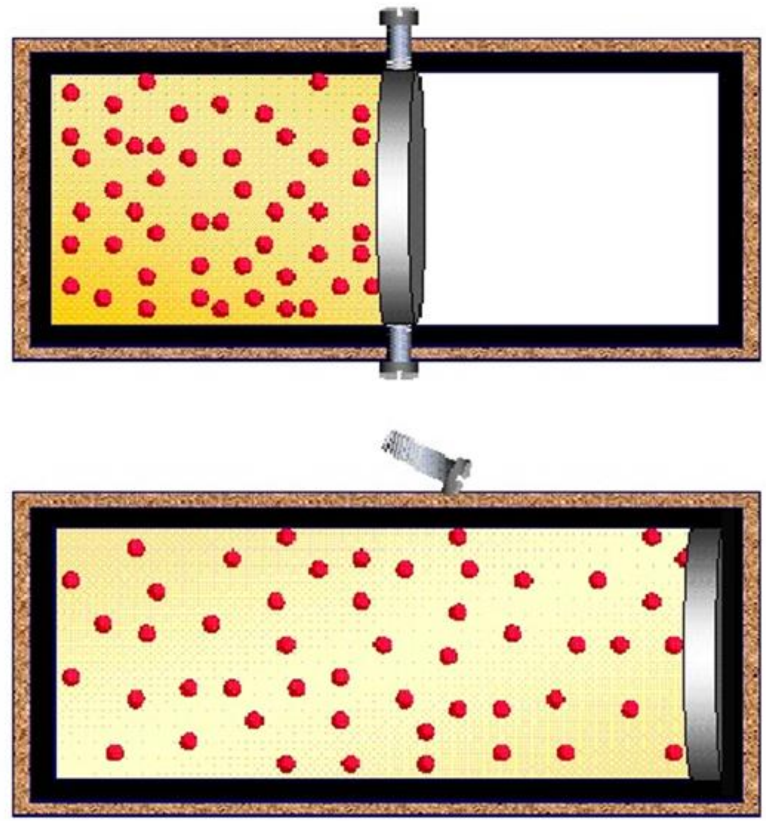
$$= \ln N - \frac{1}{N} \sum_{i=1}^{k} n_i \ln n_i = -\frac{1}{N} \left(\left(\sum_{i=1}^{k} n_i \ln n_i \right) - N \ln N \right)$$

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

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

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

均匀分布的信息熵

- 以离散分布为例：假定某离散分布可取N个值，概率都是1/N，计算该概率分布的熵。
- 解：概率分布律 $p_i=\frac{1}{N},\quad i=1,2,\cdots,N$
- 计算熵：

$$H(p)=-\sum_{i=1}^{N}p_i\ln p_i=-\sum_{i=1}^{N}\frac{1}{N}\ln\frac{1}{N}$$

$$=\sum_{i=1}^{N}\frac{1}{N}\ln N=\ln N$$

- 思考：连续均匀分布的熵如何计算？

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？

引理：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

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

- 对数形式

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

  - 若某连续分布的对数能够写成随机变量一次项和对数项的和，则该分布是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：$E(X)=\frac{\alpha}{\beta}$

给定方差的最大熵分布

- 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

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

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

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

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

- 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

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

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

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

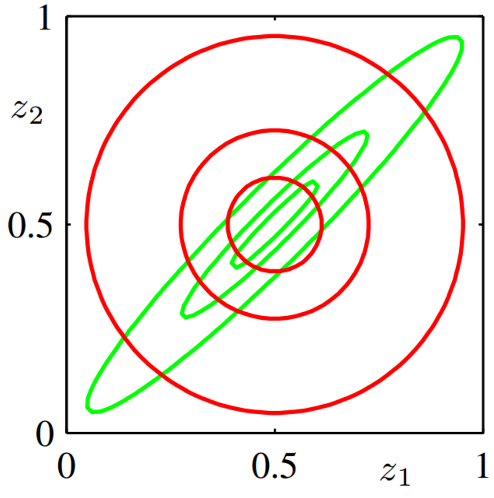
- 说明：
  - 相对熵可以度量两个随机变量的"距离"
    - 在"贝叶斯网络"、"变分推导"等章节会再次遇到
  - 一般的，D(p||q) ≠D(q||p)
  - D(p||q)≥0、 D(q||p) ≥0：凸函数中的Jensen不等式

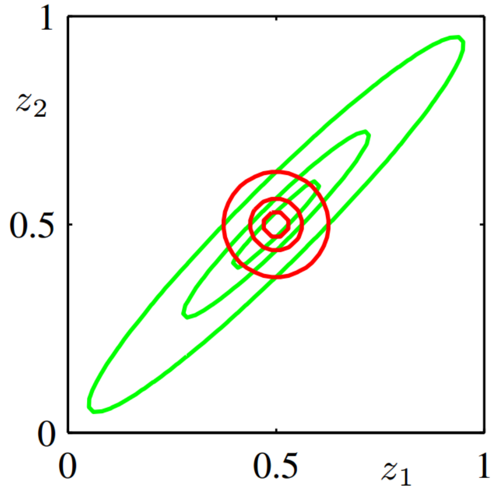
思考

- 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

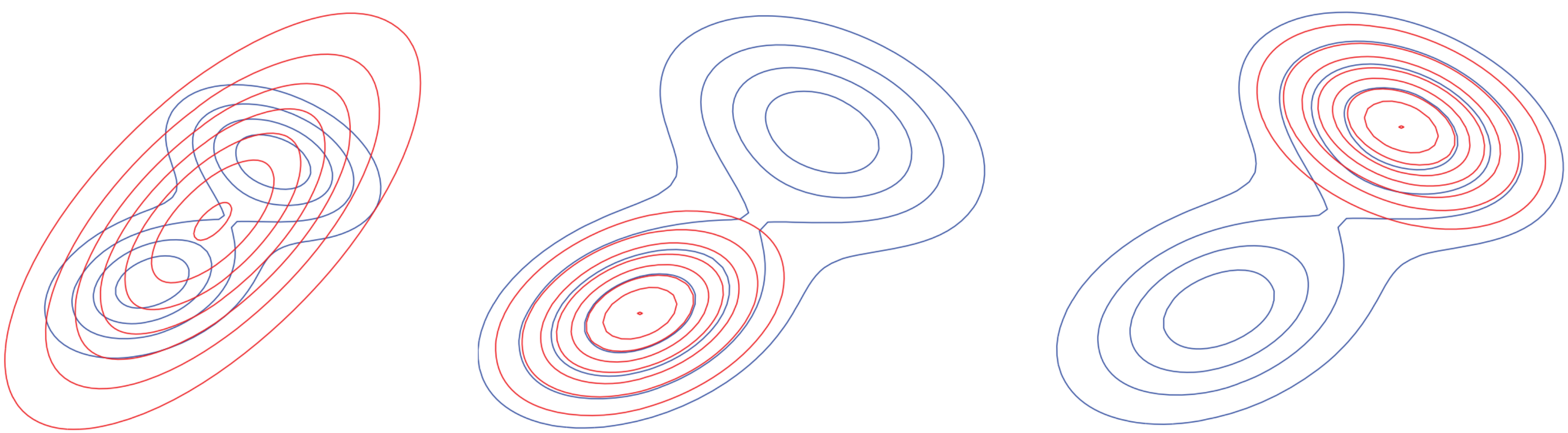
- 绿色曲线是真实分布p的等高线；红色曲线是使用近似$p(z_1,z_2)=p(z_1)p(z_2)$得到的等高线
  - 左：KL(p||q)：zero avoiding
  - 右：KL(q||p)：zero forcing





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)

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

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：$H(X|Y) \le H(X)$， $H(Y|X) \le H(Y)$

互信息：I(X,Y)=H(X)+H(Y)-H(X,Y)

$$I(X,Y) = H(X) + H(Y) - H(X,Y)$$

$$= \left(-\sum_x p(x)\log p(x)\right) + \left(-\sum_y p(y)\log p(y)\right) - \left(-\sum_{x,y} p(x,y)\log p(x,y)\right)$$

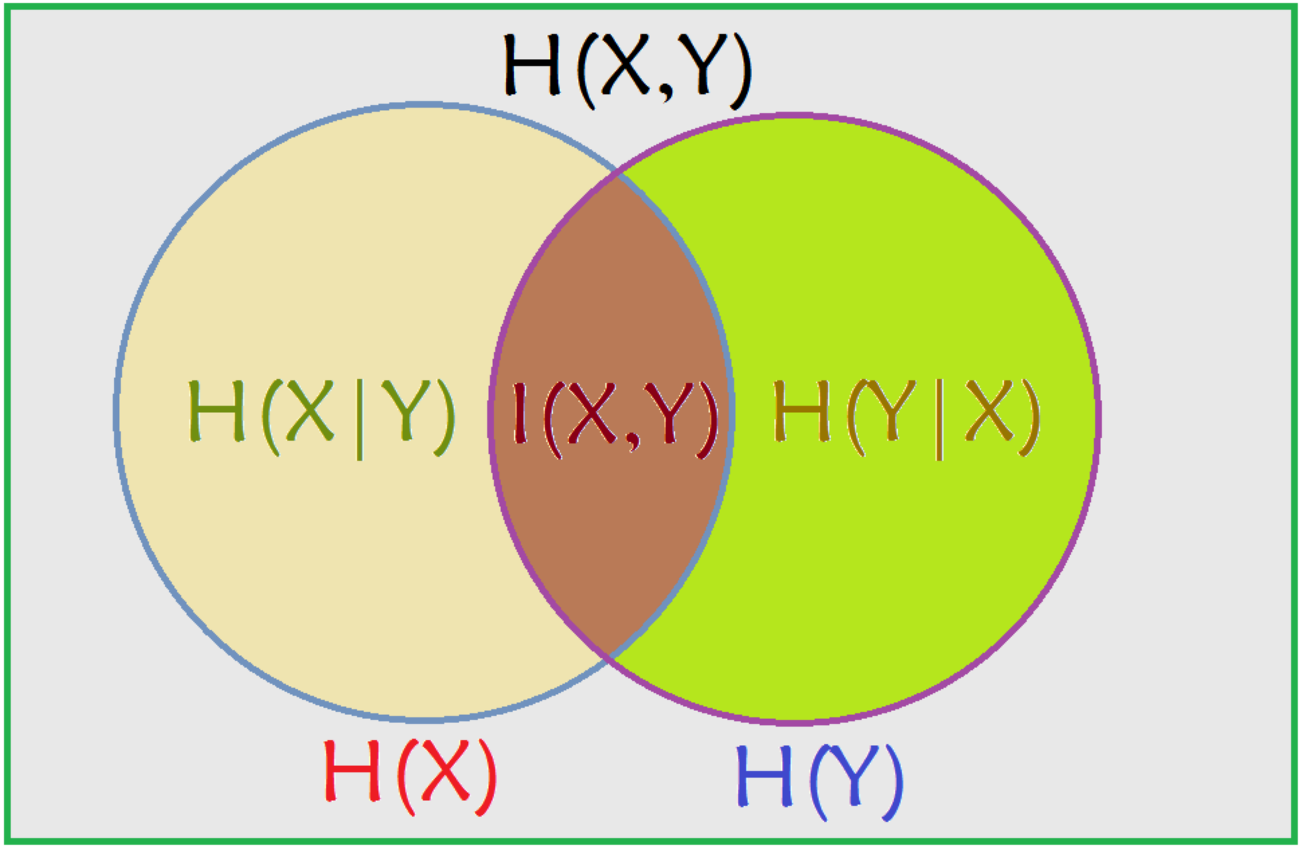
$$= \left(-\sum_x \left(\sum_y p(x,y)\right)\log p(x)\right) + \left(-\sum_y \left(\sum_x p(x,y)\right)\log p(y)\right) + \sum_{x,y} p(x,y)\log p(x,y)$$

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

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

$$= \sum_{x,y} p(x,y)\left(\log \frac{p(x,y)}{p(x)p(y)}\right)$$

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：

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

- P={p | p是X上满足条件的概率分布}

最大熵模型MaxEnt的目标拉格朗日函数L

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

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

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

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

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

λ_0与v_0仅相差常系数，后面的推导将直接以λ_0代替v_0

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

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

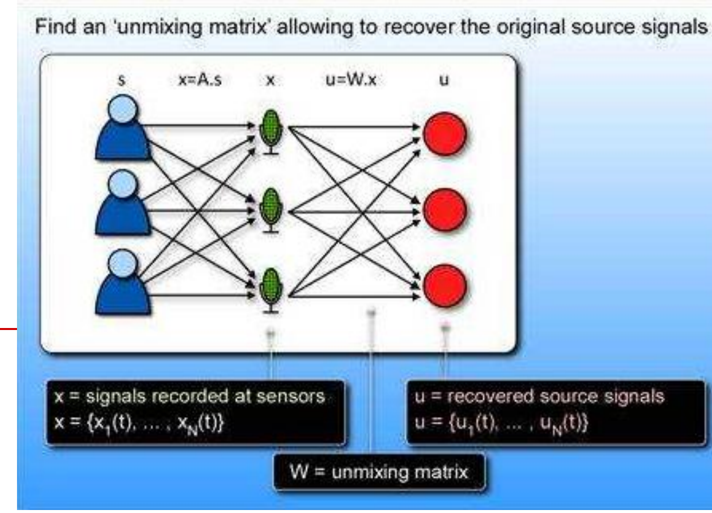
最大熵模型案例：鸡尾酒会问题

- 假设在一个聚会中有n个人同时说话，房间中n个不同位置安装声音接收器记录声音。如何根据n个混合后的声音，还原各自的声音？
  - 将n个人的声音作为n个信号源，假设观测到的声音由n个信号源线性加权得到。
  - 假定每个接收器都记录了m个观测数据。
  - 即盲源分离问题(Blind Source Separation, BSS)，其中，独立成分分析(Independent Component Analysis, ICA)是解决该问题的重要手段。

ICA的应用

- 新闻语料的主题发现
- 图像降噪/人脸识别/遥感图像分类
- 脑电图EEG/脑磁图MEG的处理
  - 感兴趣信号的提取，如眨眼(眼电信号)
- 股市预测
- 甚至移动通讯
  - 凡是有隐变量的问题，都可以尝试ICA算法

盲源分离问题记号



- 源信号

$$s=\begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1,m} \\ s_{21} & s_{22} & \cdots & s_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{n,m} \end{bmatrix} \overset{\Delta}{==} \begin{pmatrix} s^{(1)} \\ s^{(2)} \\ \vdots \\ s^{(n)} \end{pmatrix} \overset{\Delta}{==} \begin{pmatrix} s_1 & s_2 & \cdots & s_m \end{pmatrix}$$

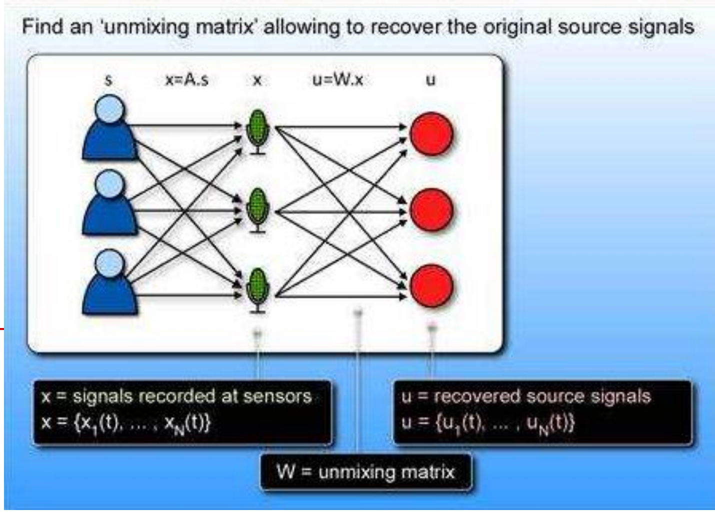
- 观测：

$$x=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1,m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{n,m} \end{bmatrix} \overset{\Delta}{==} \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{pmatrix} \overset{\Delta}{==} \begin{pmatrix} x_1 & x_2 & \cdots & x_m \end{pmatrix}$$

- 线性加权：

$$x_j = A \cdot s_j \qquad x = A \cdot s$$

盲源分离问题目标



- 根据线性模型：

$$x_j = A \cdot s_j \qquad x = A \cdot s$$

- 如果可以根据观测样本x，计算A的逆矩阵W，则： $s = W \cdot x$
- 其核心目标即计算权值矩阵W。
  - 有时把A叫做混合矩阵(mixing matrix)，W叫做解混矩阵(unmixing matrix)

ICA分离源信号的两个假设

- 两点假设：
  - 源信号彼此间统计独立
  - 源信号是非高斯分布。
- 根据中心极限定理，一组有确定方差的独立随机变量的和趋近于高斯分布；则，给定随机变量A和B，则A+B比A或B更接近高斯分布。
- 根据混合信号，如果能够找到了一组独立信号，或者说找到了一组“最不像高斯分布”的信号，则它们极有可能是源信号。

ICA的目标函数

- ICA可以用最大化各个成分的统计独立性作为目标函数，“独立性”判断原则为：
  - 最小化各个成分的互信息
    - Minimization Mutual information, MMI
    - K-L散度、最大熵
  - 最大化各个成分的非高斯性
    - Maximization non-Gaussianity
    - 思考：如何定义某分布与高斯分布的“距离”
    - 峰度近似法、负熵

非高斯性度量公式

- 使用高阶矩近似负熵：$J(x) \approx \frac{1}{12}E^2(x^3) + \frac{1}{48}kurt^2(x)$
- 为了避免估计负熵，可以使用最大熵原则：

$$J(x) \approx \sum_{i=1}^{p} \lambda_i \cdot \left[E(G_i(x)) - E(G_i(v))\right]^2$$

- λ是正常数，ν是标准高斯分布，x是0均值1方差的随机变量，G是某个非二次函数。若$G(x)=x^4$，则与上式等价。
- 实践中可以选择的G函数有：

$$G(x) = \frac{1}{\alpha}\log\cosh(\alpha \cdot x), \quad \alpha \in [1,2] \qquad G(x) = -\exp\left(-x^2/2\right)$$

最大似然估计的ICA推导

- 假定第i个源信号的概率密度函数为$p_i(s)$，第j时刻的n个源信号记做向量s_j，则在j时刻向量s_j的联合密度为：$p(s_j)=\prod_{i=1}^{n} p_i(s_{i,j})$
- 根据 $x_j = A \cdot s_j$，得

$$L(x_j)=|W|\cdot p(W\cdot x_j)=|W|\cdot\prod_{i=1}^{n} p_i(w_i^T\cdot x_j)$$

- 从而得似然函数：

$$L(x)=\prod_{j=1}^{m} p(x_j)=\prod_{j=1}^{m}\left(|W|\cdot\prod_{i=1}^{n} p_i(w_i^T\cdot x_j)\right)$$

建立参数的目标函数

- 根据 $L(x)=\prod_{j=1}^{m}\left(|W|\cdot\prod_{i=1}^{n}p_i\left(w_i^T\cdot x_j\right)\right)$

- 得到对数似然函数：

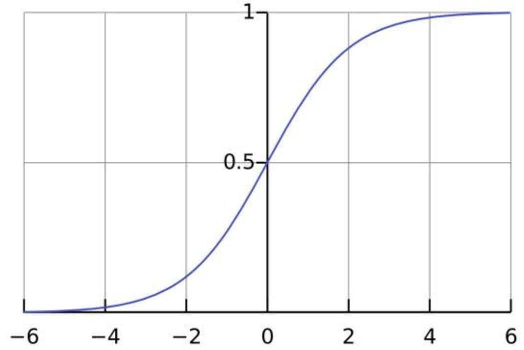
$$l(x)=\ln\prod_{j=1}^{m}\left(|W|\cdot\prod_{i=1}^{n}p_i\left(w_i^T\cdot x_j\right)\right)=\sum_{j=1}^{m}\left(\sum_{i=1}^{n}\ln p_i\left(w_i^T\cdot x_j\right)+\ln|W|\right)$$

- 从而，目标函数为：

$$J(W)=\sum_{j=1}^{m}\left(\sum_{i=1}^{n}\ln p_i\left(w_i^T\cdot x_j\right)+\ln|W|\right)$$

分析目标函数 $J(W)=\sum_{j=1}^{m}\left(\sum_{i=1}^{n}\ln p_i\left(w_i^T\cdot x_j\right)+\ln|W|\right)$

- 该目标函数中，源信号的概率密度函数未知，大胆使用Logistic/Sigmoid函数作为源信号的累积概率函数，从而源信号的概率密度为Logistic函数的导函数。



$$F(x)=\frac{1}{1+e^{-x}}$$

$$f(x)=F'(x)=\left(\frac{1}{1+e^{-x}}\right)'=\frac{e^{-x}}{\left(1+e^{-x}\right)^2}=F(x)\cdot(1-F(x))$$

$$f'(x)=f(x)(1-2\cdot F(x))$$

目标函数的导数 $f'(x)=f(x)(1-2\cdot F(x))$

- 根据假定的源信号分布，得目标函数为：

$$J(W)=\sum_{j=1}^{m}\left(\sum_{i=1}^{n}\ln p_i\left(w_i^T\cdot x_j\right)+\ln|W|\right)=\sum_{j=1}^{m}\left(\sum_{i=1}^{n}\ln f\left(w_i^T\cdot x_j\right)+\ln|W|\right)$$

- 计算目标函数对W_{ij}的导数：

$$\frac{\partial J(W)}{\partial w_{ij}}=\frac{\partial\sum_{t=1}^{m}\left(\sum_{i=1}^{n}\ln f\left(w_i\cdot x_t\right)+\ln|W|\right)}{\partial w_{ij}}$$

$$=\frac{1}{f\left(w_i\cdot x_t\right)}\cdot f\left(w_i\cdot x_t\right)\left(1-2F\left(w_i\cdot x_t\right)\right)\cdot x_{i,t}+\frac{1}{|W|}\cdot|W|\cdot\left(W^{-1}\right)_{ij}^{T}$$

$$=\left(1-2F\left(w_i\cdot x_t\right)\right)\cdot x_{i,t}+\left(W^{-1}\right)_{ij}^{T}$$

参数学习

- 上式写成向量形式：

$$\frac{\partial J(W)}{\partial W}=\begin{bmatrix}1-2F(w_1\cdot x_t)\\1-2F(w_2\cdot x_t)\\\vdots\\1-2F(w_n\cdot x_t)\end{bmatrix}\cdot x_t^T+\left(W^{-1}\right)^T$$

- 梯度下降计算参数：$W=W+\alpha\cdot\left(\begin{bmatrix}1-2F(w_1\cdot x_t)\\1-2F(w_2\cdot x_t)\\\vdots\\1-2F(w_n\cdot x_t)\end{bmatrix}\cdot x_t^T+\left(W^{-1}\right)^T\right)$
  - 固定学习率，如0.01

ICA Code

```
if __name__ == "__main__":
    s1 = [math.sin(float(x)/20) for x in range(0, 1000, 1)]
    s2 = [float(x)/50 for x in range(0, 50, 1)] * 20
    show_data(s1, s2)   # 显示真正的源信号

    A = [[0.6, 0.4], [0.45, 0.55]]  #混合矩阵
    x = mix(A, s1, s2)  #s1/s2线性加权得到输入数据x
    w = ica(x)  # ica分解，计算权值矩阵w
    [ps1, ps2] = decode(w, x)   # 根据w计算独立成分
    show_data(ps1, ps2)   # 显示解码估计的源信号
```

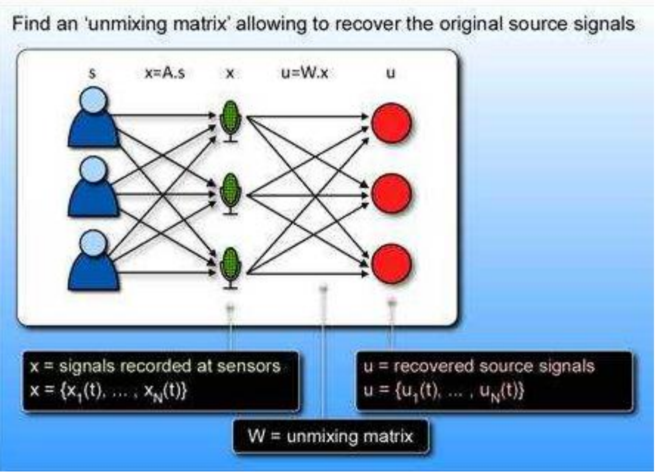
```
def shuffle(x): #将样本x的顺序打乱
    m = len(x)
    for i in range(m-1):
        r = np.random.randint(0, m-i-1)
        [x[r], x[m-i-1]] = [x[m-i-1], x[r]]
```

ICA

```python
def ica(x):
    m = len(x)        # 样本数目
    n = len(x[0])     # mic数目
    w = [[0.0]*n for t in range(n)]
    iw = [[0.0]*n for t in range(n)]
    for i in range(n):
        w[i][i] = 1
    w1 = [[0.0]*n for t in range(n)]
    alpha = 0.001
    # shuffle(x)     # 试验表明：不打乱熟悉仍然可以正常速度收敛
    for time in range(200):
        for i in range(m):
            for j in range(n):
                t = 1 - 2*logistic(dot_product(w[j], x[i]))
                n_multiply(t, x[i], w1[j])  # w1[j] = t*x[i]
            trans_inverse(w, iw)  # iw = w^T^(-1)
            n_multiply2(0.05, iw)  # iw *= alpha
            add(w1, iw)  # w1 += iw
            n_multiply2(alpha, w1)  # w1 *= alpha
            transpose(w1)
            add(w, w1)              # w += w1
        print time, ": \t", w
    return w
```

ICA的不确定性 $x = A \cdot s$

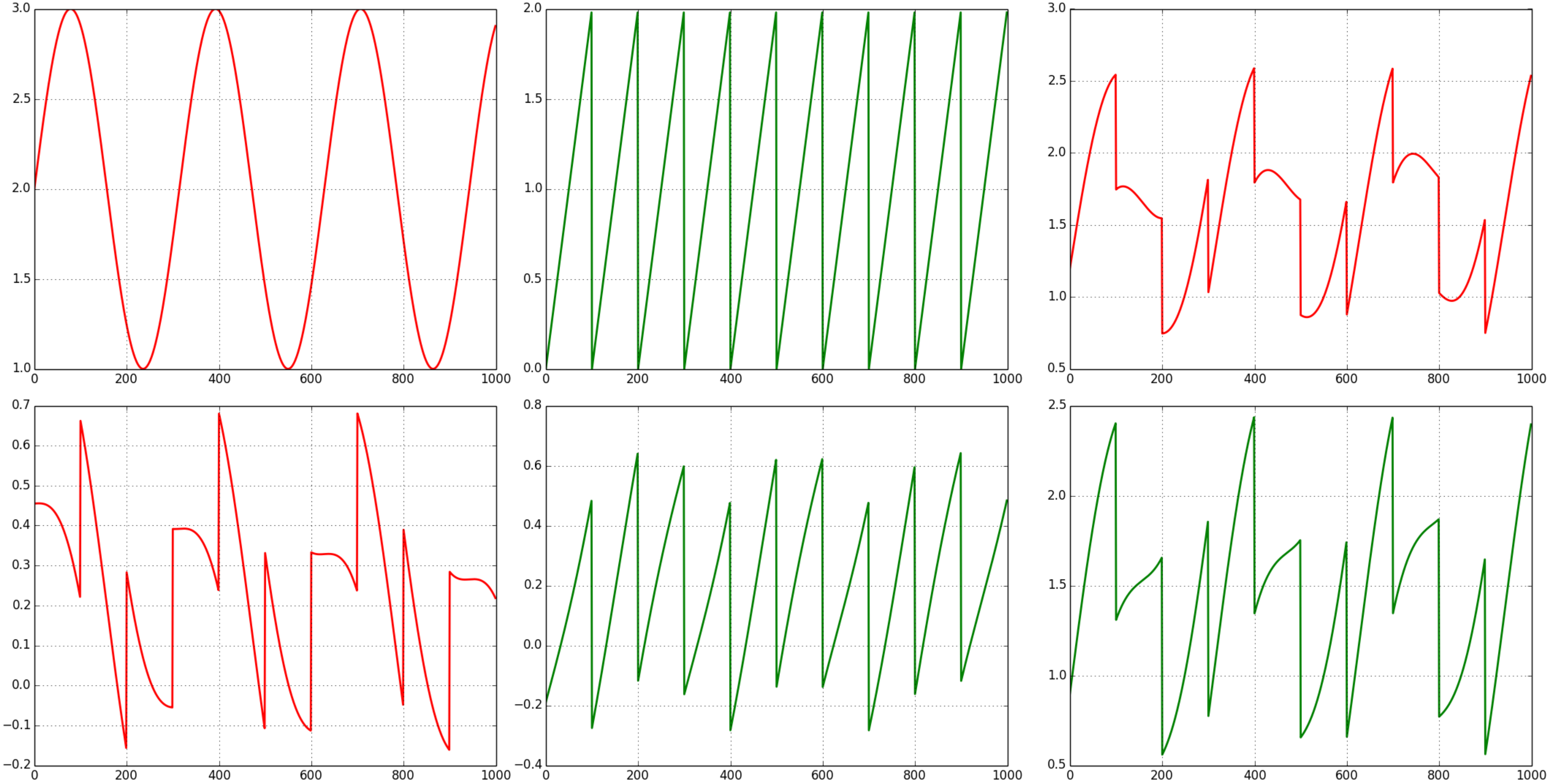
$$x_j = A \cdot s_j$$



- 振幅不确定：
  - 将分离的信号同时数乘n倍，仍然可以保证信号间独立——无法确定源信号的方差。
  - 将分离的信号同时取相反数，信号间保持独立
- 顺序不确定：
  - 将分离后的信号交换位置，仍然独立。
- 解决方案：通过其他信息确定振幅和顺序。

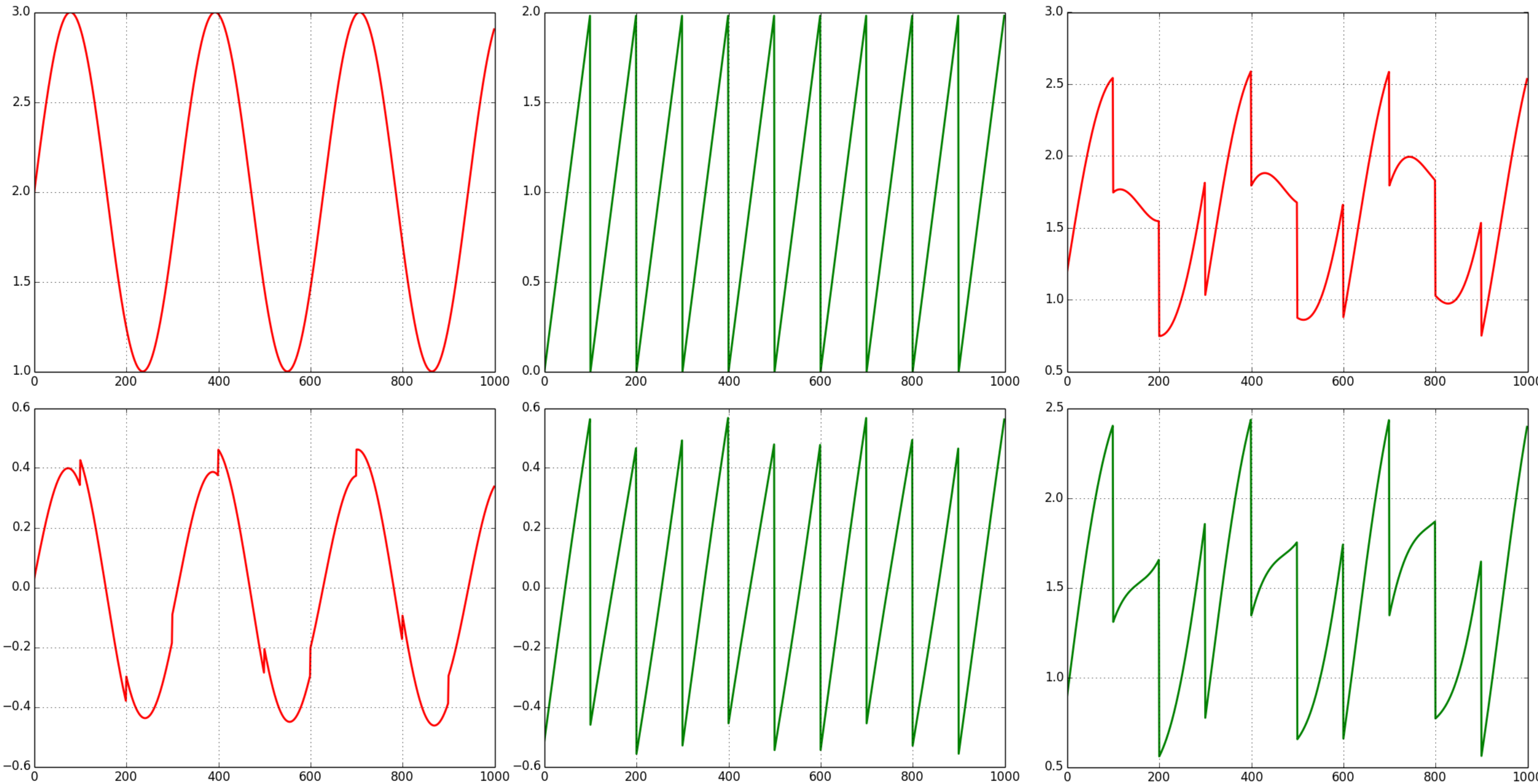
原始ICA分离效果

源信号1	源信号2	混合信号1
独立成分1	独立成分2	混合信号2



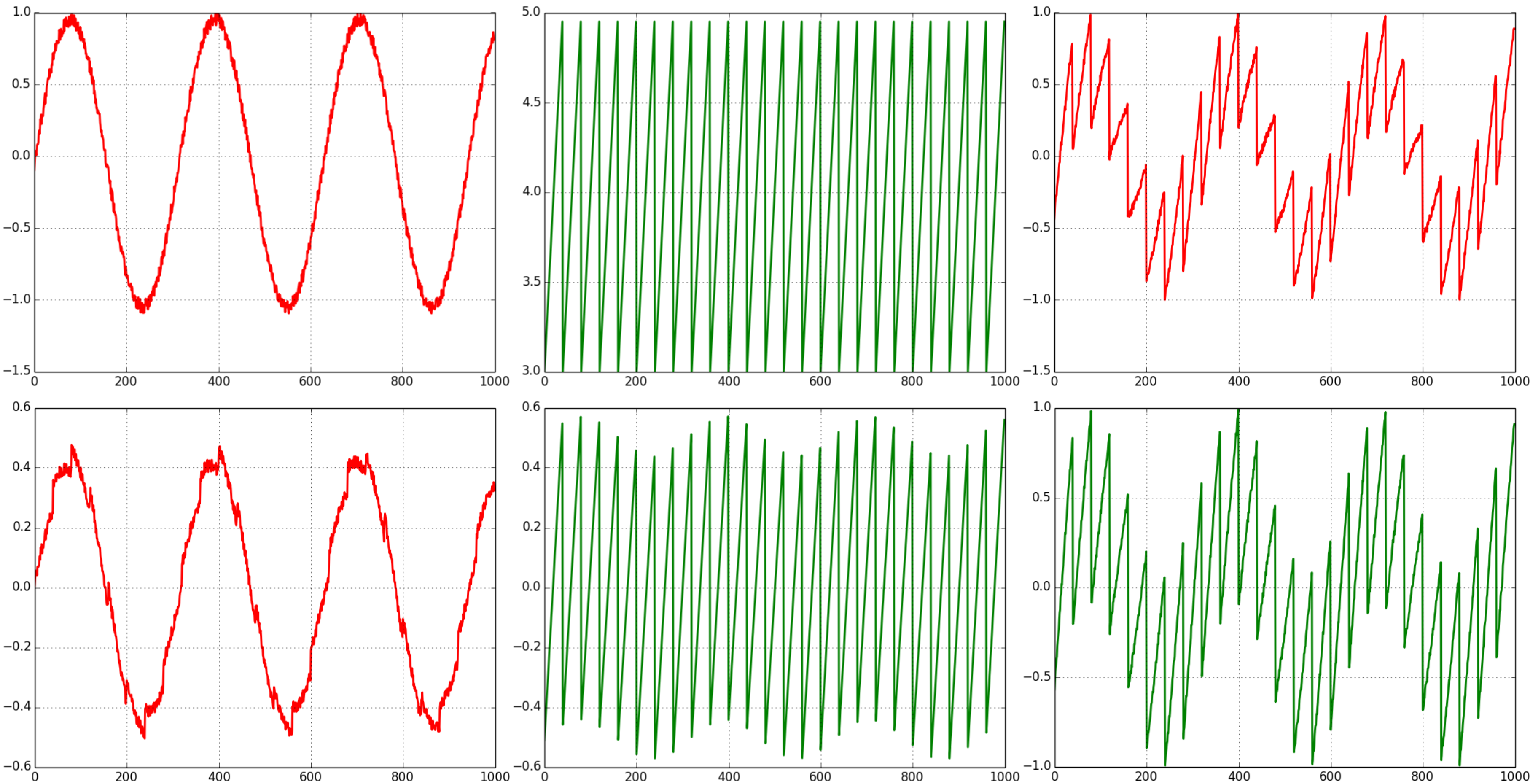
去均值ICA分离

源信号1	源信号2	混合信号1
独立成分1	独立成分2	混合信号2



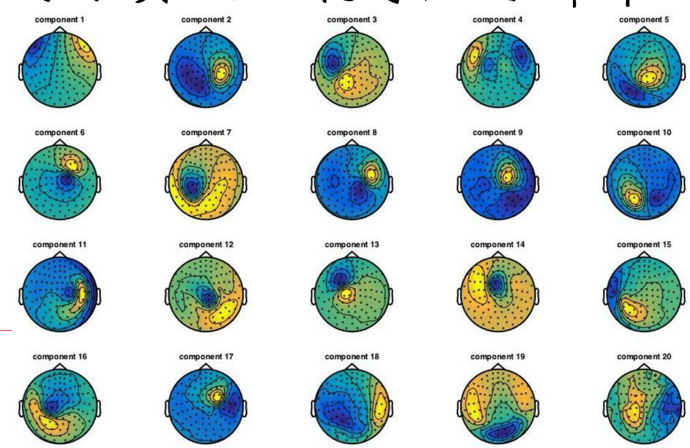
带噪声的信号分离

源信号1	源信号2	混合信号1
独立成分1	独立成分2	混合信号2



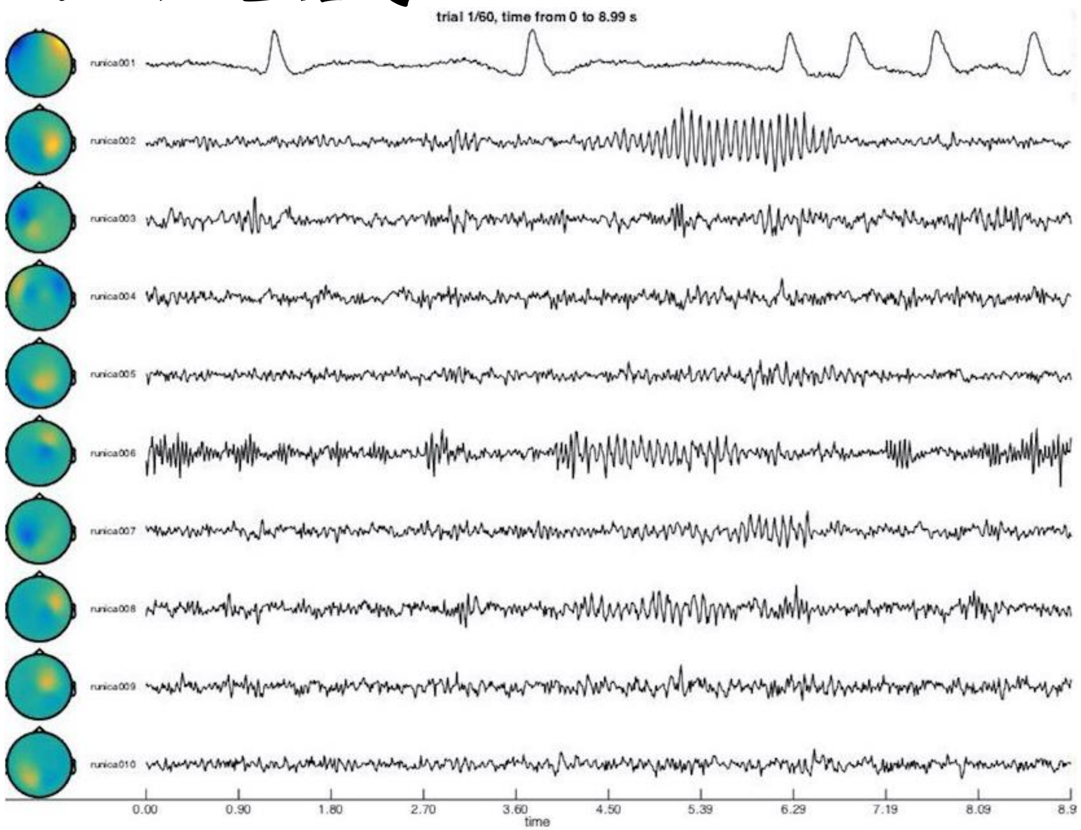
ICA的应用：脑电信号分离

- 将大脑皮层不同位置(图中黑点所示)收集的脑电信号作为采样数据，得到N组数据，每组数据采样M次，得到原始混合数据X。
- 将脑电信号看成眼电信号、神经电信号、心电信号等信号的叠加。从而，使用ICA将眼电信号、神经电信号、心电信号等信号分离出来。
  - 但实际数据中，只有眼电信号的分类效果最好，可解释性最强。
  - 数据由网友清风Laynne提供



脑电信号分离效果

- 组分1：眼电信号



白化/漂白whitening

$$x=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1,m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{n,m} \end{bmatrix} \overset{\Delta}{=\!=} \begin{pmatrix} x_1 & x_2 & \cdots & x_m \end{pmatrix}$$

- 计算观测数据x的n×n的对称阵$x\cdot x^T$的特征值和特征向量，用特征值形成对角阵D，特征向量形成正交阵U，则有：$x\cdot x^T=U^TDU$，令$\tilde{x}=U^TD^{-0.5}U\cdot x$
- 从而：$\tilde{x}\cdot\tilde{x}^T=\left(U^TD^{-0.5}U\cdot x\right)\left(U^TD^{-0.5}U\cdot x\right)^T$

$$=\left(U^TD^{-0.5}U\cdot x\right)\left(x^TU^TD^{-0.5}U\right)$$

$$=U^TD^{-0.5}U\cdot\left(xx^T\right)\cdot U^TD^{-0.5}U$$

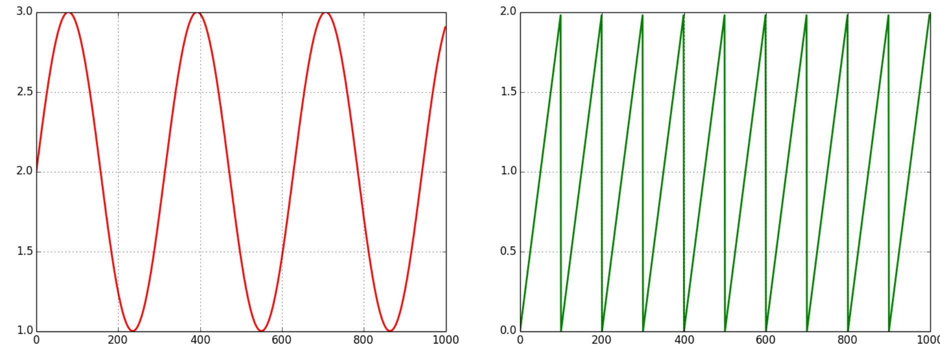
$$=U^TD^{-0.5}U\cdot U^TDU\cdot U^TD^{-0.5}U=I$$

- 白化保证每个初始组分可作为ICA的先验组分，在PCA降维章节中将继续讨论相关技术。

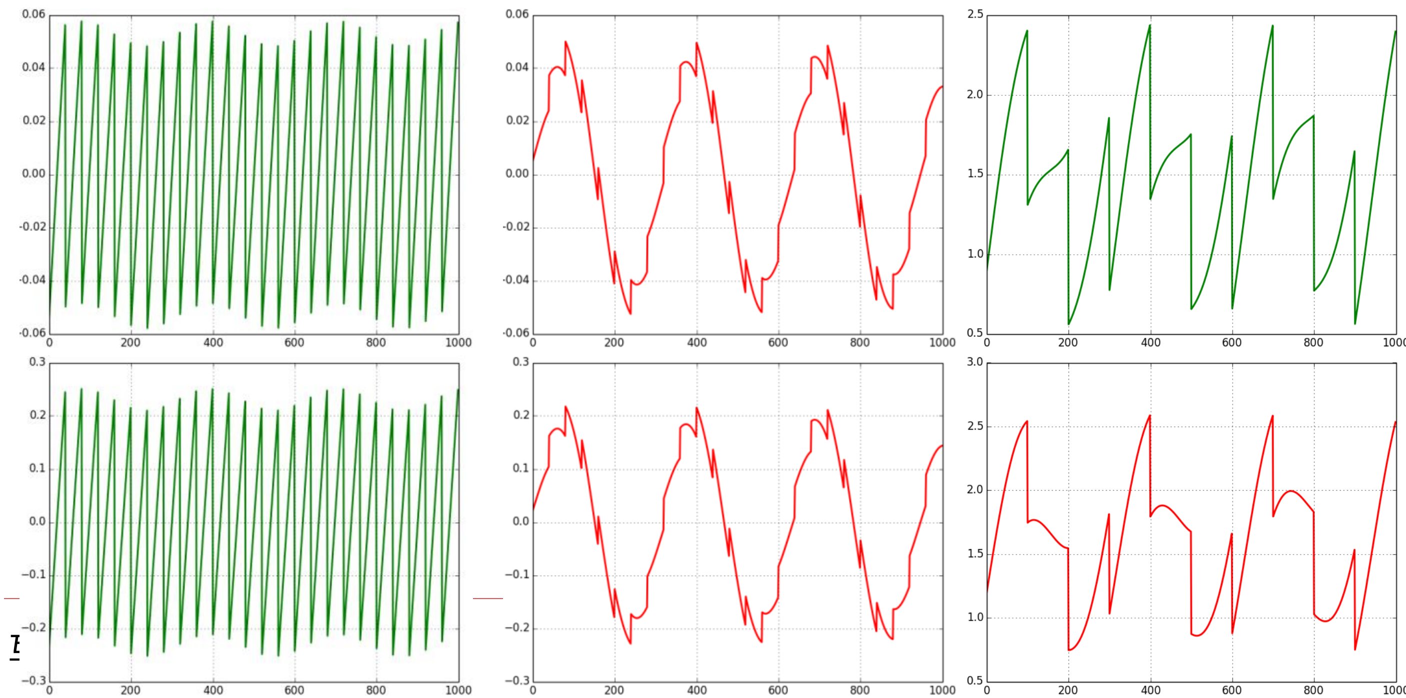
白化Code

```python
def whitening(x):
    m = len(x)
    n = len(x[0])
    # 计算x*x'
    xx = [[0.0]*n for tt in range(n)]
    for i in range(n):
        for j in range(i, n):
            s = 0.0
            for k in range(m):
                s += x[k][i] * x[k][j]
            xx[i][j] = s
            xx[j][i] = s
    # 计算 x*x'的特征值和特征向量
    lamda, egs = np.linalg.eig(xx)
    lamda = [1/math.sqrt(d) for d in lamda]
    # 计算白化矩阵U'D^(-0.5)*U
    t = [[0.0]*n for tt in range(n)]
    for i in range(n):
        for j in range(n):
            t[i][j] = lamda[j] * egs[i][j]
    whiten_matrix = [[0.0]*n for tt in range(n)]
    for i in range(n):
        for j in range(n):
            s = 0.0
            for k in range(n):
                s += t[i][k] * egs[j][k]
            whiten_matrix[i][j] = s
    # 白化x
    wx = [0.0]*n
    for j in range(m):
        for i in range(n):
            s = 0.0
            for k in range(n):
                s += whiten_matrix[i][k] * x[j][k]
            wx[i] = s
        x[j] = wx[:]
```

增加白化后的ICA效果



源信号1	源信号2	
白化信号1	白化信号2	混合信号1
独立成分1	独立成分2	混合信号2



总结与思考

$$W = W + \alpha \cdot \left(\begin{bmatrix} 1-2F(w_1 \cdot x_t) \\ 1-2F(w_2 \cdot x_t) \\ \vdots \\ 1-2F(w_n \cdot x_t) \end{bmatrix} \cdot x_t^T + (W^{-1})^T \right)$$

- 经典的ICA经过中心化、白化、降维作为预处理步骤。根据不同的推导方案和近似公式，ICA实现方案有infomax、FastICA、JADE、KernelICA等。
  - 如：Sigmoid函数替换成tanh函数
- ICA运行前需要预先指定分类数目，由于ICA的不确定性，分离后的组分需要调整顺序和振幅。
- 如果混合信号数目与源信号数目不相等，则是超定或欠定(over/under-determined)问题，如何解决？
  - 矩阵的广义逆

最大似然估计

- 似然函数取对数：

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

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

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

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

- 第二项是常数，可忽略

MLE与条件熵

- 此目标式，与条件件熵具有相同的形式。

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

- 既然函数式相同，极有可能二者殊途同归，得到相同的目标函数。

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

附：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

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

附：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

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

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

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

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

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

总结

- 根据最大似然估计的正确性可以断定：最大熵的解(无偏的对待不确定性)是最符合样本数据分布的解，即最大熵模型的合理性。
- 信息熵可以作为概率分布集散程度的度量，使用熵的近似可以推导出gini系数，在统计问题、决策树等问题中有重要作用。
- 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

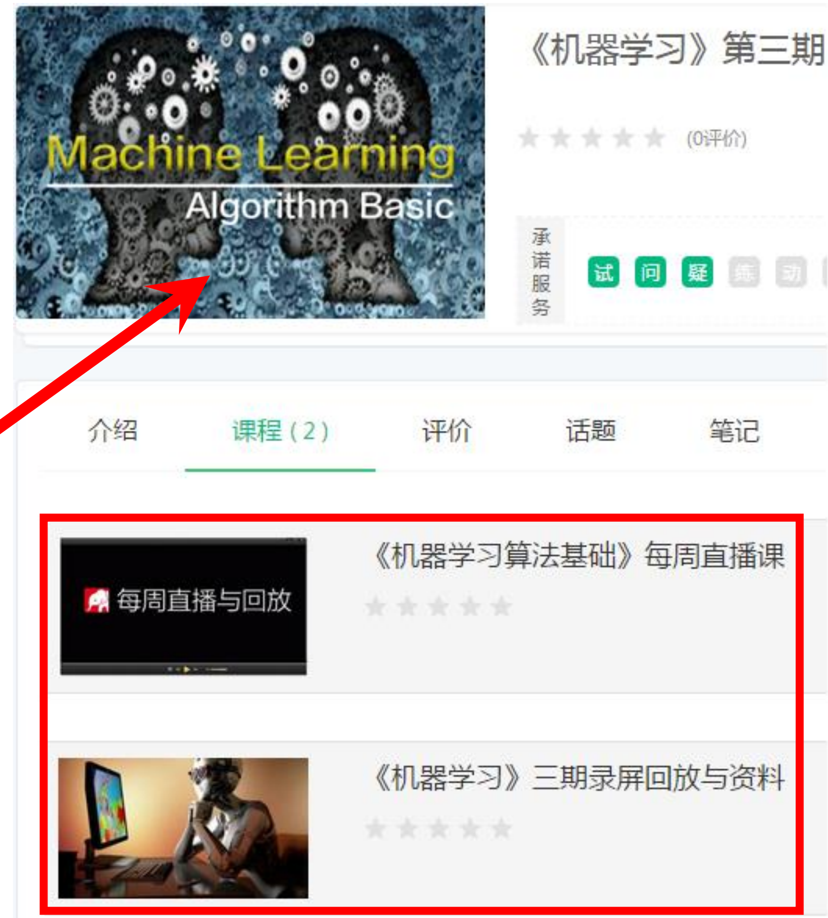
知识＝不确定度的补集

参考文献

- Thomas M. Cover, Joy A. Thomas, *Elements of Information Theory*, 2006
- Aapo Hyvärinen, Erkki Oja, *Independent Component Analysis: Algorithms and Applications*, 2000
- 李航，统计学习方法，清华大学出版社，2012
- https://en.wikipedia.org/wiki/Independent_component_analysis

课程资源

- 直播课的入口
- 录播视频和讲义资料



感谢大家！

恳请大家批评指正！