“自然语言处理导论”课程讲义

自然语言处理的数学基础

孙栩
信息科学技术学院
xusun@pku.edu.cn
http://klcl.pku.edu.cn/member/sunxu/index.htm

内容回顾

- 回顾：自然语言处理的2大类基本方法

- 1，理性主义
  - 基于语言学知识、规则的自然语言处理
  - 理论基础：chomsky文法理论、各种语言学理论、规则，等

- 2，经验主义
  - 基于概率统计、机器学习的自然语言处理
  - 理论基础：概率论、信息论、机器学习等

理性主义 vs 经验主义

- **理性主义**
  - 形式语言
  - 语法理论
  - 推理方法
  - 逻辑运算
  - 自动机
  - …
- **经验主义**
  - 信息论
  - Ngram语言模型
  - 搜索算法
  - HMM模型
  - Perceptron模型
  - CRF模型
  - 句法分析算法
  - …

概率论基础

概率

- 概率是一个在0到1之间的实数，是对随机事件发生之可能性的度量。
- 如果用P(a) 作为事件a的概率，A是实验的样本空间，则概率函数必须满足如下公理：

公理1：非负性

$$\forall a \in A, P(a) \geq 0$$

公理2：规范性

$$\sum_{a \in A} P(a) = 1$$

公理3：可数可加性　对于任意的互斥事件集合$\{a_i\}$有：

$$P(\cup_{i=1}^{\infty} a_i) = \sum_{i=1}^{\infty} P(a_i)$$

- 任意一个满足上述条件的函数P都可以作为A空间的概率函数
- 称函数值P(a)为A空间中事件a的概率

条件概率

- 如果a 和b 是样本空间A 上的两个事件，P(b)>0，那么在给定b 时a 的条件概率P(a|b) 为：

$$P(a|b) = \frac{P(a \cap b)}{P(b)}$$

- 条件概率P(a|b) 给出了在已知事件b 发生的情况下，事件a 发生的概率

贝叶斯定理(Bayes' theorem)

- 贝叶斯定理是关于随机事件A和B的条件概率的一则定理

$$P(a|b) = \frac{P(b|a)P(a)}{P(b)}$$

- P(a)是a的先验概率，之所以称为"先验"是因为它不考虑任何b方面的因素
- P(a|b)是已知b发生后a的条件概率，称作a的后验概率
- P(b|a)是已知a发生后b的条件概率，称作b的后验概率
- P(b)是b的先验概率

- **贝叶斯定理(Bayes' theorem)**
  - 推导过程：利用条件概率的定义

$$P(a|b) = \frac{P(a \cap b)}{P(b)} = \frac{P(b|a)P(a)}{P(b)}$$

  - 贝叶斯决策理论在自然语言处理，包括文本分类、语音分析等问题的研究中具有重要用途

概率论基础

- **例1**
  - 给定语音信号G(siGnal)，找出对应的语句S(Sentence)，使得P(S|G)最大，有

$$S^* = \underset{S}{\operatorname{argmax}}\, P(S|G)$$

  - 根据贝叶斯定理，有

$$S^* = \underset{S}{\operatorname{argmax}} \frac{P(G|S)P(S)}{P(G)}$$

  - 因为P(G)是先验分布，在G给定时是归一化常数，有

$$S^* = \underset{S}{\operatorname{argmax}}\, P(G|S)P(S)$$

概率论基础

- **例1**
  - 给定语音信号G(siGnal)，找出对应的语句S(Sentence)，使得P(S|G)最大，有

$$S^* = \arg\max_{S} \boxed{P(S|G)}$$

难以直接计算

  - 根据贝叶斯定理，有

$$S^* = \arg\max_{S} \frac{P(G|S)P(S)}{P(G)}$$

  - 因为P(G)是先验分布，在G给定时是归一化常数，有

$$S^* = \arg\max_{S} \boxed{P(G|S)}\boxed{P(S)}$$

可以通过声学模型直接计算

可以通过ngram语言模型直接计算

例2

- 对于文本分类问题，假设某文本类A很少出现，平均每10,000个文本中才出现一次。
- 某系统用于文本分类，如果该文本确实为A类型，系统判断为A的概率为0.9。
- 如果该文本不为A，系统判断结果为A的概率为0.1。
- 问题：当该系统判断结果为A时，正确的概率？

概率论基础

- **用贝叶斯理论解决此问题**
  - 令O(Oracle)表示事件”该文本确实类型为A”
  - 令U(oUtput)表示事件“系统预测类型为A”

$$P(O) = \frac{1}{10000} = 0.0001 \qquad P(\overline{O}) = \frac{10000-1}{10000} = 0.9999$$

$$P(U|O) = 0.9 \qquad P(U|\overline{O}) = 0.1$$

$$\begin{aligned} P(O|U) &= \frac{P(U|O)P(O)}{P(U)} \\ &= \frac{P(U|O)P(O)}{P(U|O)P(O) + P(U|\overline{O})P(\overline{O})} \\ &= \frac{0.9 \times 0.0001}{0.9 \times 0.0001 + 0.1 \times 0.9999} \\ &\boxed{\approx 0.001} \end{aligned}$$

二项分布(binomial distribution)

- 当重复一个只有两种输出（假定为a）的实验（伯努利实验），a在一次实验中发生的概率为p，现把实验独立地重复n 次。用k 表示a 在这 n 次实验中发生的次数
- a可以出现在 n 个位置中的任何一个位置，所以，结果序列有种C(n,k)可能。由此，可以得出：

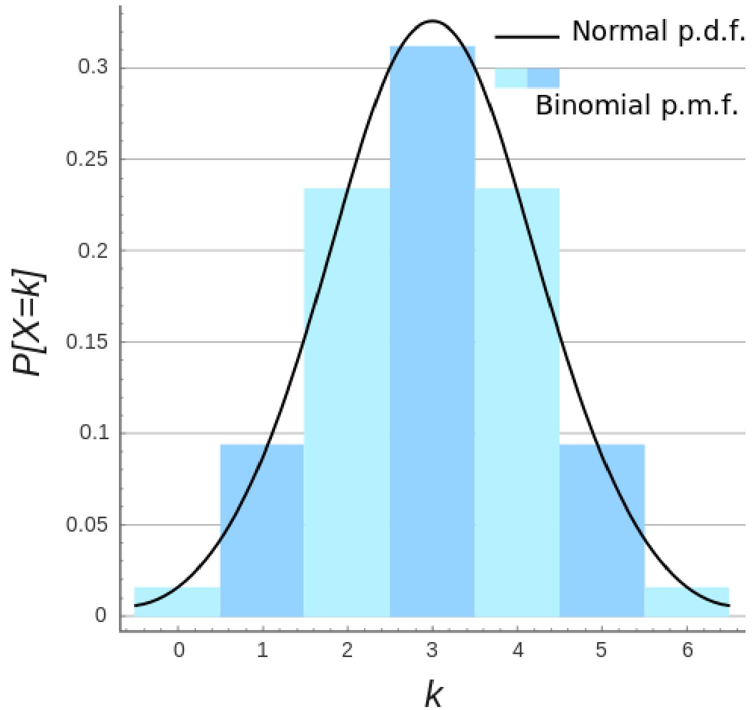
$$P_k = C_n^k p^k (1-p)^{n-k}$$

$$C_n^k = \frac{n!}{(n-k)!k!}$$

- 此概率分布称为二项式分布，并记为 B(n, p)

概率论基础

- **二项分布(binomial distribution)**
  - 当n = 1时，二项分布就是伯努利分布。
  - 自然语言处理的研究常常要检验提出方法和现有方法之间的差异的显著性，二项分布是显著性差异检验的基础。



期望(expectation)

- 期望值是一个随机变量所取值的概率平均
- 设X 为一随机变量，其期望为

$$E(X_n) = \sum_{k=1}^{n} x_k p_k$$

方差(variance)

- 一个随机变量的方差描述的是该随机变量的值偏离其期望值的程度
- 设X 为一随机变量，其方差为：

$$
\begin{aligned}
Var(X) &= E\left[\left(X - E(X)\right)^2\right] \\
&= E(X^2) - E^2(X)
\end{aligned}
$$

标准差(Deviation)

- 是方差的平方根

最大似然估计

- 英文：maximum-likelihood estimation (MLE)
- 是用来估计一个概率模型的参数的一种方法
- 步骤
  - 给定一个问题空间 A，如何学习其概率分布 D？
  - 首先，从 A 随机抽取 n 个样本 X={x_1, x_2, …, x_n}
  - 定义一个以 w 为参数的概率模型，具有如下**似然函数**

$$P(x_1, x_2, \cdots, x_n) = f(x_1, x_2, \cdots, x_n|w)$$

  - **估计**最优的参数 w* ，使得似然函数 f **最大化**

似然函数 + 估计 + 最大化 → 最大似然估计

梯度

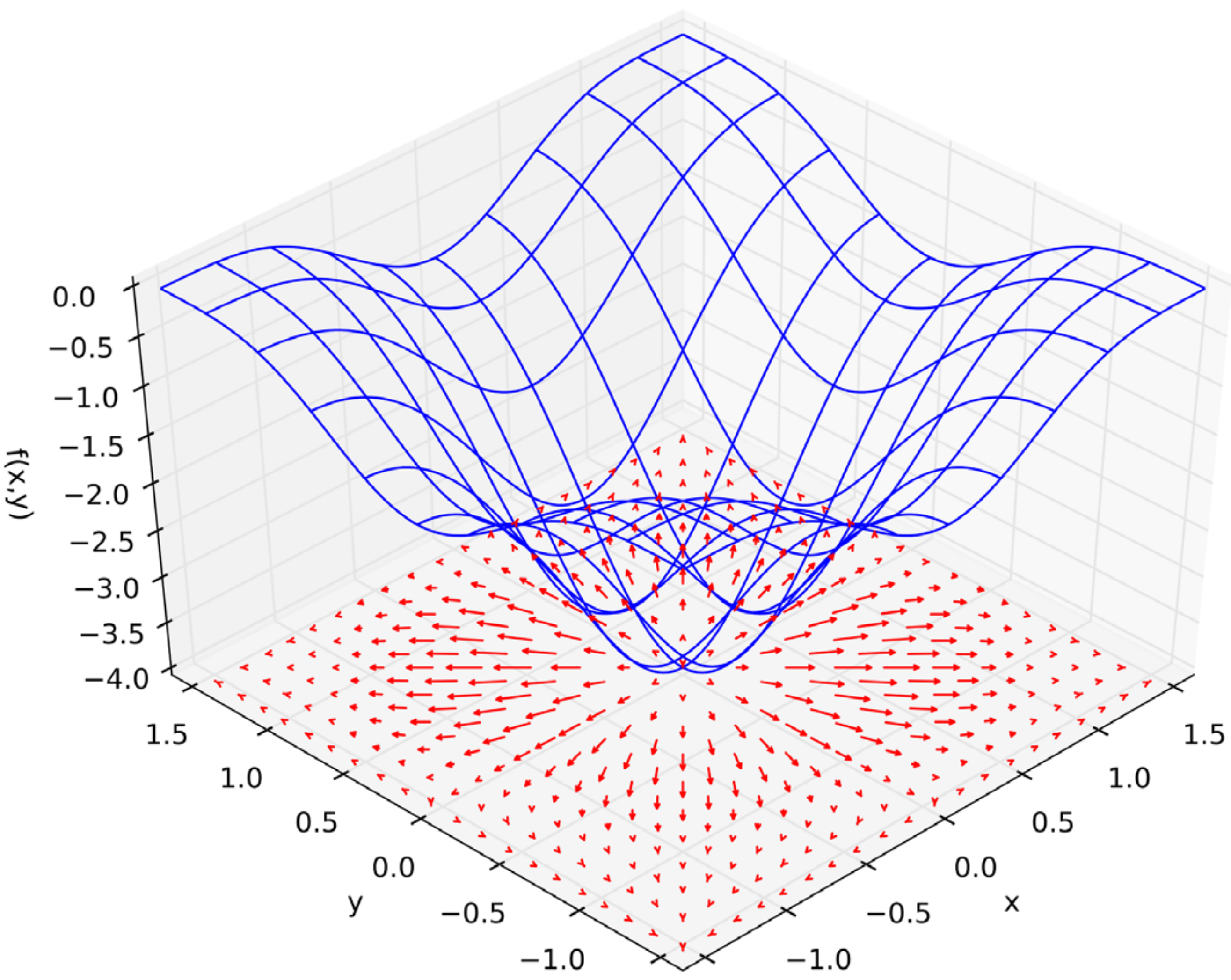
- 标量场中某一点上的梯度指向标量场增长最快的方向
  - 梯度的长度是这个最大的变化率
- 在单变量的实值函数的情况，梯度只是导数
  - 对于一个线性函数，也就是线的斜率

- 数学计算

$$\nabla f(w_1, w_2, ..., w_i) = \left(\frac{\partial f}{\partial w_1}, \frac{\partial f}{\partial w_2}, ..., \frac{\partial f}{\partial w_i}\right)$$

偏导数

梯度



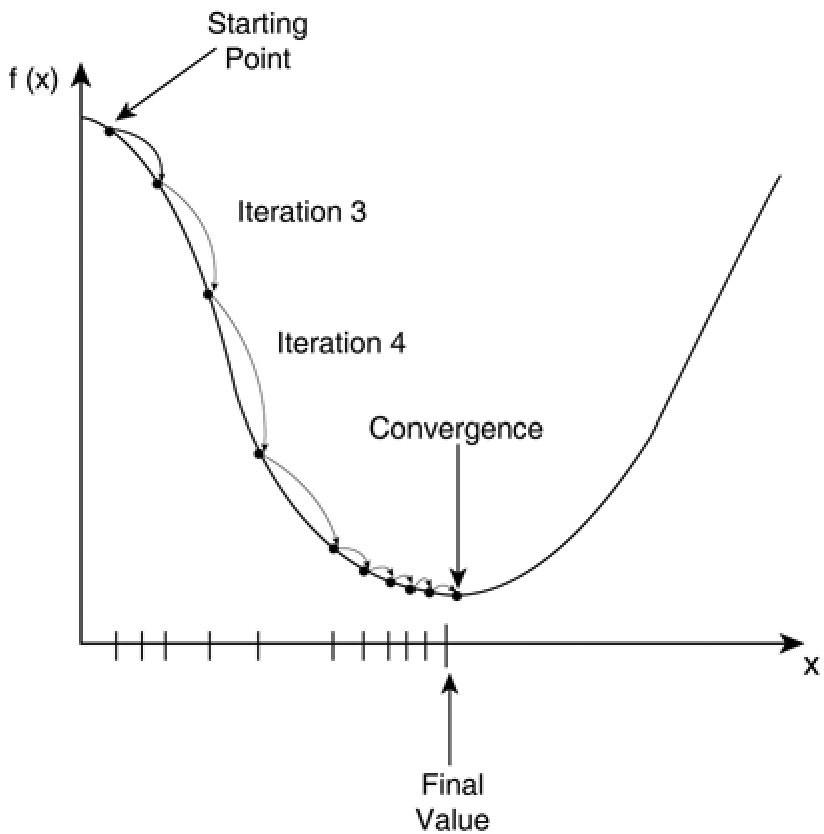
数学基础

- **参数学习**
  - 对于最大似然估算，假设似然函数已经给定
  - 那么具体怎么估算最优的参数 w* ？
  - 这个问题叫做**参数估算**，或者叫**参数学习**

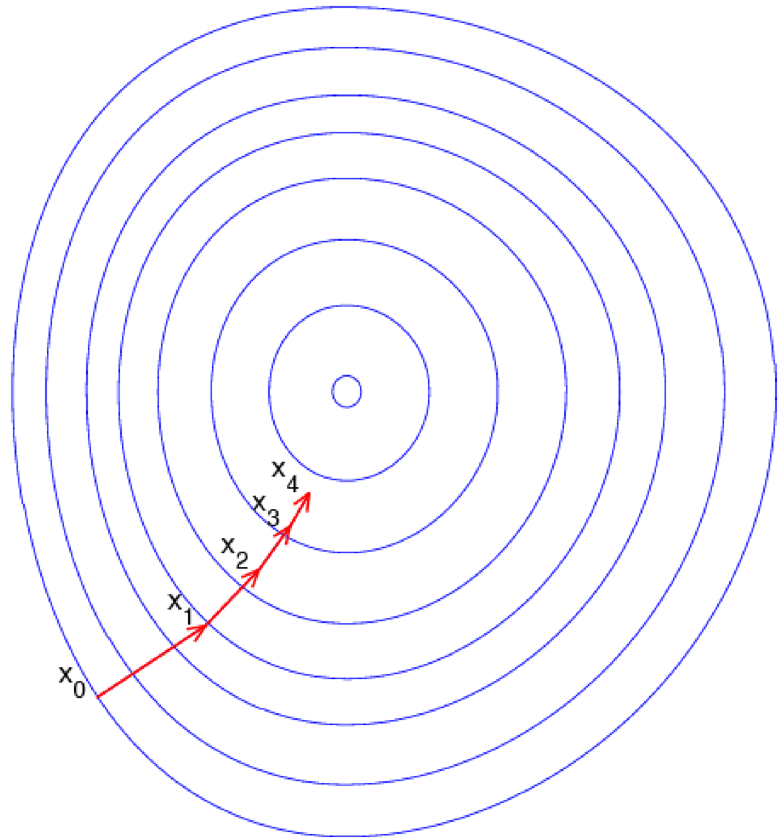
主要思想：
既然梯度是函数增长最快的方向，为什么我们不顺着梯度方向寻找似然函数的最大值？

  - 典型的解决办法
    - **梯度下降方法**
    - 英文：Gradient Descent
    - 其实这个名字是针对求最小值问题
    - 对于求最大值问题，如最大似然估算，实际上应该叫做**梯度上升方法**
    - 但是由于最大值问题和最小值问题本质上是一样的，所以学界一般都统一称为梯度下降方法

梯度下降方法



一维函数情况



二维函数情况

最大似然估计的完整实现

- **最大似然估计：通过一个实例来理解！**
  - **似然函数**是怎么定义的？
  - 怎么搜索**最优参数**，从而获得自然函数的最大值？
    - **梯度**是怎么计算的？
    - 最优参数是怎么通过**梯度下降算法**获得的？

- **一个实例：抛硬币问题**

  - 问题描述：给定一个质量分布**不均匀**的硬币，如何确定其正面、反面的出现概率？

最大似然估计的完整实现

一个实例：抛硬币问题

- 问题描述：给定一个质量分布**不均匀**的硬币，如何确定其正面、反面的出现概率？
- 牛顿力学分析、空气动力学？
  - **No…太复杂 ☹**
- 抛 n 次，如果正面出现 q 次，则正面出现概率为q/n?
  - **这的确是对的！但是该方法太简单，只能用于此特定问题，不是一个通用的方法… 遇到自然语言处理问题怎么办？☹**
- 看看怎么用**最大似然估计**解决这个问题
  - **通用的方法，可以用于其它更复杂的现实问题，包括自然语言处理！☺**

最大似然估计的完整实现

- **一个实例：抛硬币问题**
  - 问题描述：给定一个质量分布**不均匀**的硬币，如何确定其正面、反面的出现概率？

- **看看怎么用最大似然估计解决这个问题**

最大似然估计的完整实现

一个实例：抛硬币问题

- 问题描述：给定一个质量分布**不均匀**的硬币，如何确定其正面、反面的出现概率？

看看怎么用**最大似然估计**解决这个问题

- 参数设定
  - 问题空间是A = { 0, 1 } ，1代表正面，0代表反面
  - 概率分布D={P(1)， P(0)}未知
  - 因为P(0)=1-P(1)，只需要计算P(1)即可获知D
  - 因此，本概率模型只需要一个参数w=P(1)，代表正面出现的概率即可

最大似然估计的完整实现

- **一个实例：抛硬币问题**
  - 问题描述：给定一个质量分布**不均匀**的硬币，如何确定其正面、反面的出现概率？
- **看看怎么用最大似然估计解决这个问题**
  - 参数设定
    - **只需要一个参数w=P(1)，代表正面出现的概率**

最大似然估计的完整实现

一个实例：抛硬币问题

- 问题描述：给定一个质量分布**不均匀**的硬币，如何确定其正面、反面的出现概率？

看看怎么用**最大似然估计**解决这个问题

- 参数设定
  - **只需要一个参数w=P(1)，代表正面出现的概率**
- 抽取样本
  - 从 A 随机抽取 n 个样本 X={x_1, x_2, …, x_n}，假设我们实际上抽取了 6 个样本X={0 ，1 ，1 ，1 ，1 ，0}

最大似然估计的完整实现

- **一个实例：抛硬币问题**
  - 问题描述：给定一个质量分布**不均匀**的硬币，如何确定其正面、反面的出现概率？
- **看看怎么用最大似然估计解决这个问题**
  - 参数设定
    - **只需要一个参数w=P(1)，代表正面出现的概率**
  - 抽取样本
    - **抽取了6个样本　X={0，1，1，1，1，0}**

最大似然估计的完整实现

- 一个实例：抛硬币问题
  - 问题描述：给定一个质量分布**不均匀**的硬币，如何确定其正面、反面的出现概率？

- 看看怎么用**最大似然估计**解决这个问题
  - 参数设定
    - **只需要一个参数w=P(1)，代表正面出现的概率**
  - 抽取样本
    - **抽取了6个样本　X={0，1，1，1，1，0}**
  - 似然函数定义

$$P(X|w) = w^n(1-w)^m$$

$$P(X|w) = w^4(1-w)^2 = f(w)$$

最大似然估计的完整实现

- **一个实例：抛硬币问题**
  - 问题描述：给定一个质量分布**不均匀**的硬币，如何确定其正面、反面的出现概率？

- **看看怎么用最大似然估计解决这个问题**
  - 参数设定
    - **只需要一个参数w=P(1)，代表正面出现的概率**
  - 抽取样本
    - **抽取了6个样本　X={0，1，1，1，1，0}**
  - 似然函数定义

$$P(X|w) = w^4(1-w)^2 = f(w)$$

最大似然估计的完整实现

- **一个实例：抛硬币问题**
  - 问题描述：给定一个质量分布**不均匀**的硬币，如何确定其正面、反面的出现概率？

- **看看怎么用最大似然估计解决这个问题**
  - 参数设定
    - **只需要一个参数w=P(1)，代表正面出现的概率**
  - 抽取样本
    - **抽取了6个样本　X={0，1，1，1，1，0}**
  - 似然函数定义

$$P(X|w) = w^4(1-w)^2 = f(w)$$

  - 估计最优参数

最大似然估计的完整实现

估计最优参数

- (1) 梯度计算步骤

$$\nabla f(w) = 4w^3(w-1)^2 + 2w^4(w-1)$$

- (2) 梯度下降步骤（其实是沿着梯度上升）

$$w \leftarrow w + \nabla f(w)$$

- (3)重复步骤1和步骤2直到收敛状态

最大似然估计的完整实现

估计最优参数

- (1) 梯度计算步骤　w随机初始化为 w1 = 0.5

$$\nabla f(w) = 4 \times 0.5^3 \times (0.5 - 1)^2 + 2 \times 0.5^4 \times (0.5 - 1) = 0.0625$$

- (2) 梯度下降步骤（其实是沿着梯度上升）

  w2 = 0.5 + 0.0625 = 0.5625

- (3)重复步骤1和步骤2直到收敛状态

最大似然估计的完整实现

估计最优参数

- (1) 梯度计算步骤

$$
\begin{aligned}
\nabla f(w) &= 4 \times 0.5625^3 \times (0.5625 - 1)^2 + 2 \times 0.5625^4 \times (0.5625 - 1) \\
&= 0.0487
\end{aligned}
$$

- (2) 梯度下降步骤（其实是沿着梯度上升）

w3 = 0.5625 + 0.0487 = 0.6112

- (3)重复步骤1和步骤2直到收敛状态

最大似然估计的完整实现

- **重复步骤1和步骤2直到收敛状态**

- w4 = 0.6112 + 0.0296 = 0.6407
- w5 = 0.6407 + 0.0147 = 0.6554
- w6 = 0.6554 + 0.0065 = 0.6620
- w7 = 0.6620 + 0.0028 = 0.6647
- w8 = 0.6647 + 0.0011 = 0.6659
- w9 = 0.6659 + 0.0004679 = 0.6663
- w10 = 0.6663 + 0.0001911 = 0.6665
- w11 = 0.6665 + 0.00007793 = 0.6666
- w12 = 0.6666 + 0.00003176 = 0.6666

收敛状态，最大似然估计完成

最终参数结果：w = 0.6666

最大似然估计的完整实现

一个实例：抛硬币问题

- 问题描述：给定一个质量分布**不均匀**的硬币，如何确定其正面、反面的出现概率？

回顾一下，进行结果检验

q/n= 4/6 = 0.6666

- 牛顿力学分析、空气动力学？
  - **No...太复杂 ☹**

- 抛 n 次，如果正面出现 q 次，则正面出现概率为q/n?
  - **这的确是对的！但是该方法太简单，只能用于此特定问题，不是一个通用的方法... 遇到自然语言处理问题怎么办？☹**
- 看看怎么用**最大似然估计**解决这个问题
  - **通用的方法，可以用于其它更复杂的现实问题，包括自然语言处理！☺**

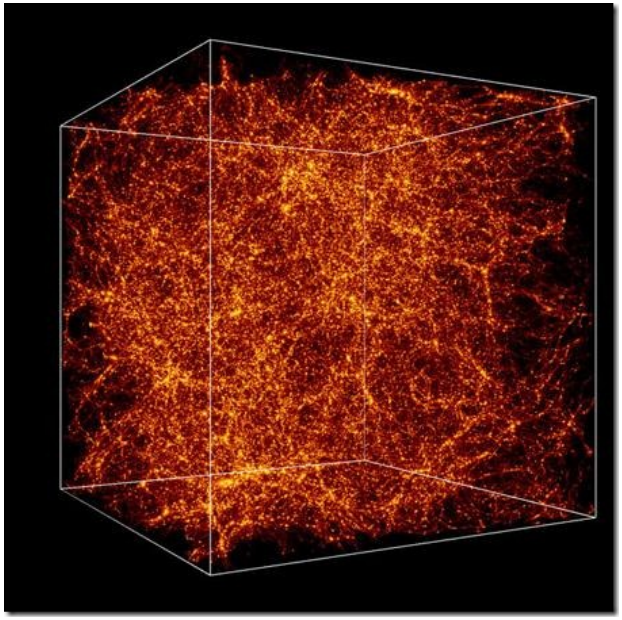
最大似然估计的完整实现

- **最大似然估计广泛应用于自然语言处理**

- 最简单的例子：
  - 比如把抛硬币问题变成文本分类问题
  - 正面 → 文本类A
  - 反面 → 文本类B
  - 当然，实际文本类可以多于2个，而且使用更复杂的概率函数。但最大似然估计的步骤和抛硬币问题是一样的！
    - **针对具体自然语言处理问题，主要的区别就是重新定义了概率函数(即目标函数)**

信息论基础

- **熵(entropy)**
  - 熵是信息论中重要的基本概念
  - 1948年6月和10月，由贝尔实验室出版的《贝尔系统技术》杂志连载了香农博士（Claude Elwood Shannon）的文章《通讯的数学原理》，该文奠定了香农信息论的基础。



信息论基础

熵(entropy)

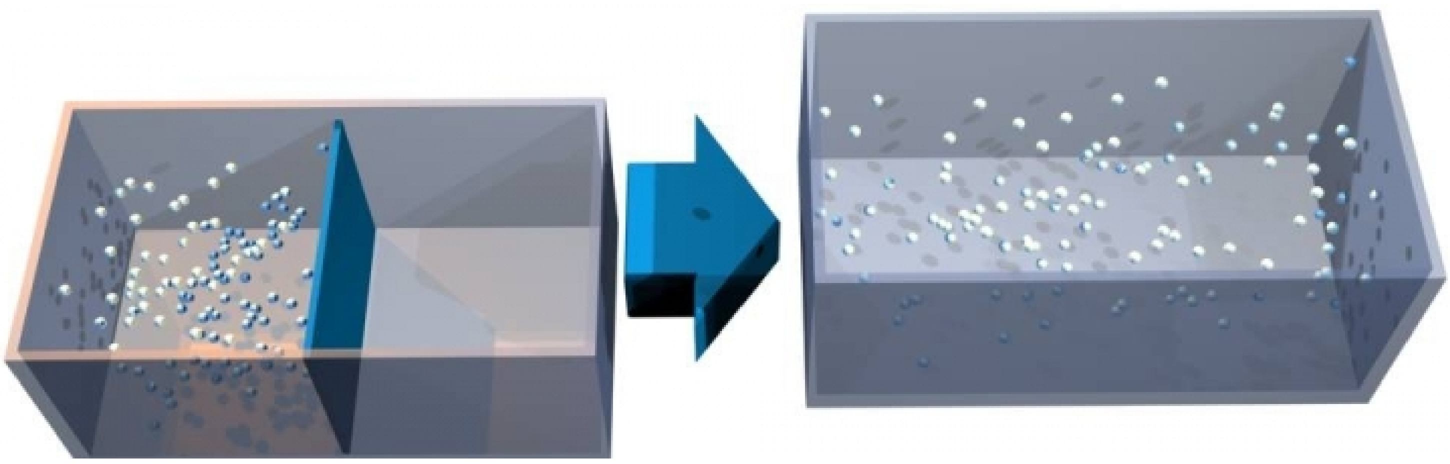
- 如果 X 是一个离散型随机变量，对于一个事件x，其概率分布为p(x)，则该随机变量X的熵为:

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

- 其中，约定0log 0 = 0
- 通常熵的单位为二进制位比特（bit）

熵(entropy)

- 熵又称为自信息（self-information），被视为描述一个随机变量的不确定性的数量。一个随机变量的熵越大，它的不确定性越大。那么，正确估计其值的可能性就越小。越不确定的随机变量越需要大的信息量用以确定其值。



- 熵(entropy)
  - 例子：假设空间A能发生两个事件a和b
  - （1）假设a和b等概率发生，A的熵是多少？

$$
\begin{aligned}
H(A) &= -\sum_{x\in X} p(x)\log_2 p(x) \\
&= -\{0.5\times\log_2 0.5 + 0.5\times\log_2 0.5\} \\
&= -(-0.5-0.5) = \boxed{1}
\end{aligned}
$$

  - （2）假设P(a)=0.9，P(b)=0.1 ，A的熵是多少？

$$
\begin{aligned}
H(A) &= -\sum_{x\in X} p(x)\log_2 p(x) \\
&= -\{0.9\times\log_2 0.9 + 0.1\times\log_2 0.1\} \\
&= -(-0.1368-0.3322) = \boxed{0.469}
\end{aligned}
$$

联合熵(joint entropy)

- 如果X, Y 是一对离散型随机变量X, Y ~ p(x, y)，X, Y 的联合熵H(X, Y) 为：

$$H(X,Y) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2 p(x,y)$$

- 联合熵实际上就是描述一对随机变量平均所需要的信息量。

条件熵(conditional entropy)

- 给定随机变量X 的情况下，随机变量Y 的条件熵定义为：

$$H(Y \mid X) = \sum_{x \in X} p(x) H(Y \mid X = x)$$

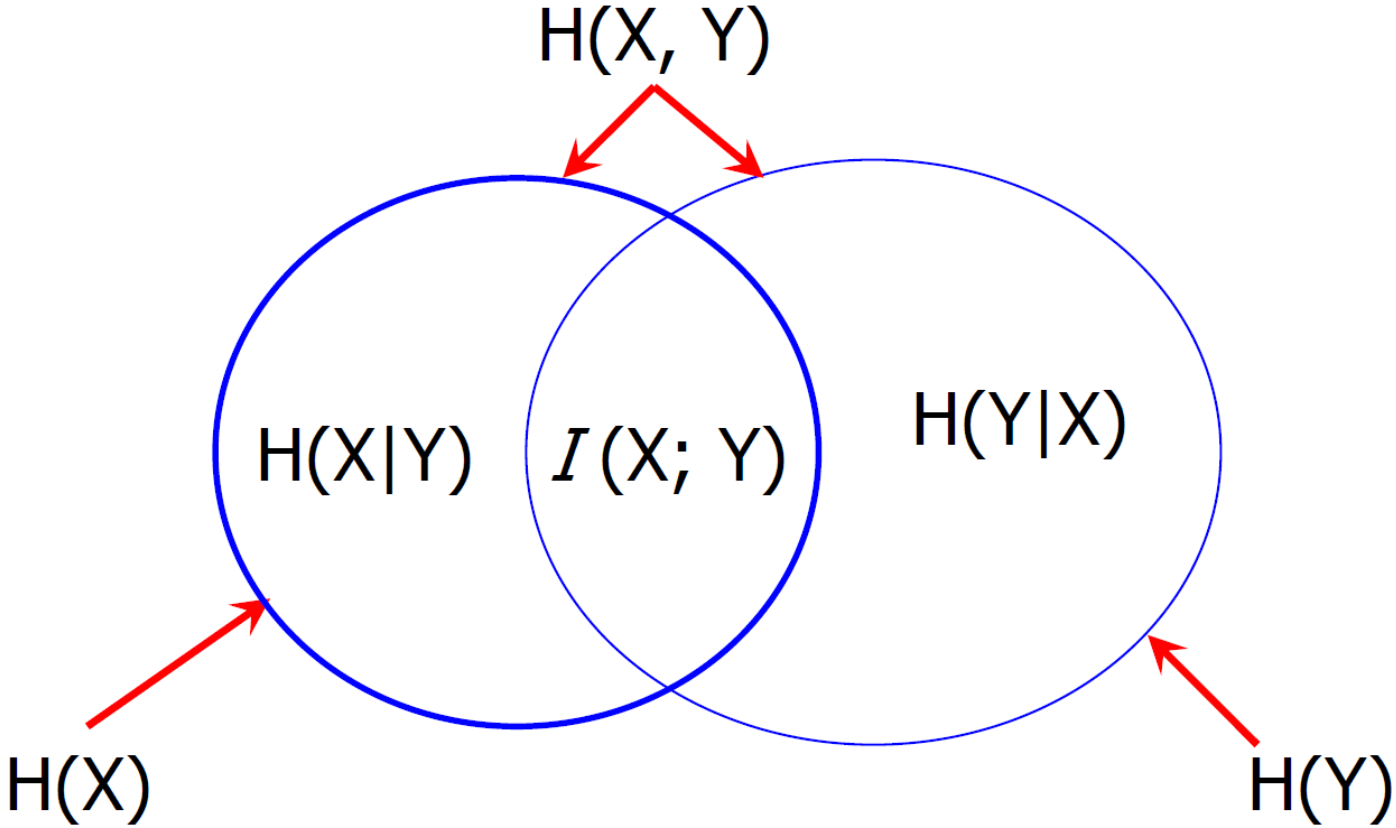
$$= \sum_{x \in X} p(x) [-\sum_{y \in Y} p(y \mid x) \log p(y \mid x)]$$

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

- **互信息(mutual information)**
  - 如果(X, Y) ~ p(x, y)，X, Y 之间的互信息I(X; Y)为：

$$I(X;Y) = H(X) - H(X \mid Y)$$

  - 互信息I (X; Y) 是在知道了Y 的值后X 的不确定性的减少量。即，Y 的值透露了多少关于X 的信息量。



熵、联合熵、条件熵、互信息之间的大致关系

谢 谢

QUESTION？