

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



Logistic回归



小象学院
ChinaHadoop.cn

邹博

主要内容

□ Logistic 回归

- 分类问题的首选算法

□ 多分类：Softmax 回归

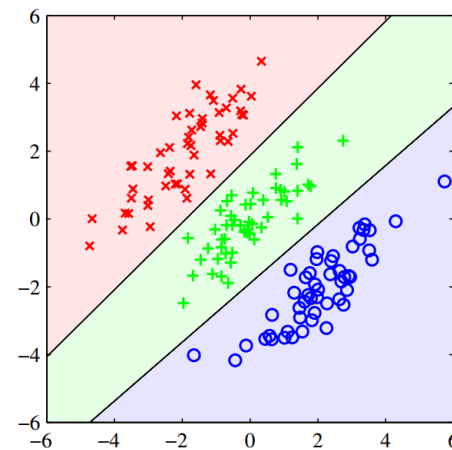
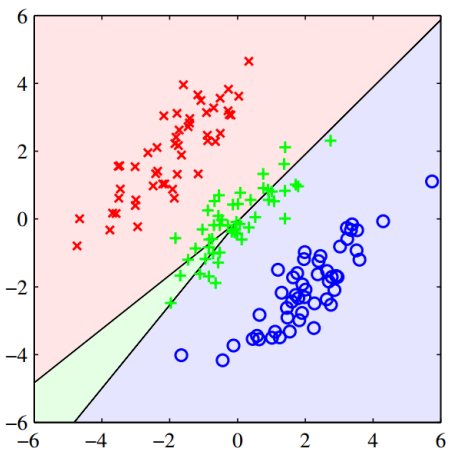
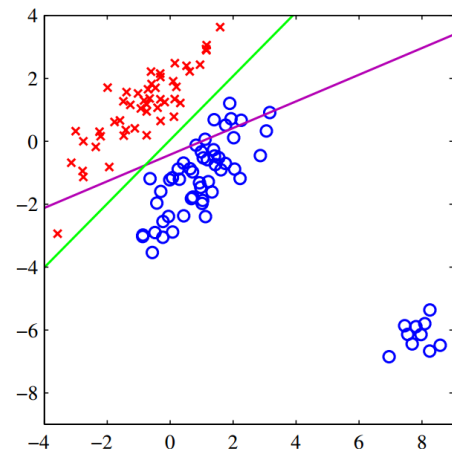
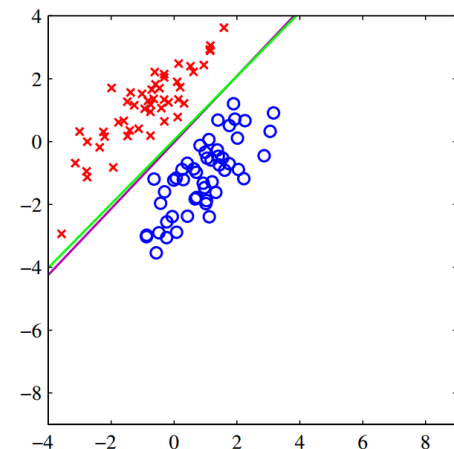
- 目标函数

□ 信息熵

- 熵
- 联合熵、条件熵、相对熵
- 互信息

线性回归-Logistic回归

- ☐ 紫色:
 - ☒ 线性回归
- ☐ 绿色:
 - ☒ Logistic回归
- ☐ 左侧:
 - ☒ 线性回归
- ☐ 右侧:
 - ☒ Softmax回归

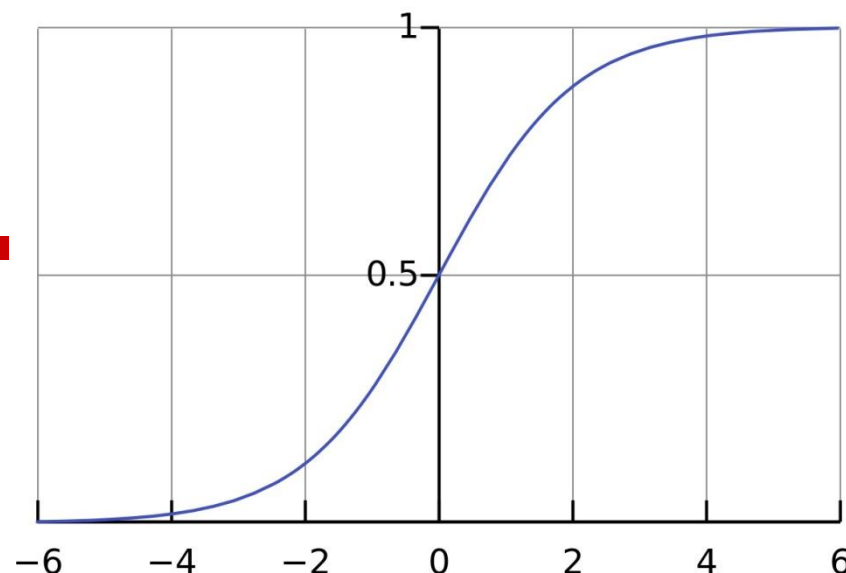


Logistic回归

□ Logistic/sigmoid函数

$$h_{\theta}(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$\begin{aligned} g'(x) &= \left(\frac{1}{1 + e^{-x}} \right)' = \frac{e^{-x}}{(1 + e^{-x})^2} \\ &= \frac{1}{1 + e^{-x}} \cdot \frac{e^{-x}}{1 + e^{-x}} = \frac{1}{1 + e^{-x}} \cdot \left(1 - \frac{1}{1 + e^{-x}} \right) \\ &= g(x) \cdot (1 - g(x)) \end{aligned}$$



$$g(z) = \frac{1}{1 + e^{-z}}$$

Logistic回归参数估计

□ 假定: $P(y = 1 \mid x; \theta) = h_{\theta}(x)$

$$P(y = 0 \mid x; \theta) = 1 - h_{\theta}(x)$$

$$p(y \mid x; \theta) = (h_{\theta}(x))^y (1 - h_{\theta}(x))^{1-y}$$

$$L(\theta) = p(\vec{y} \mid X; \theta)$$

$$= \prod_{i=1}^m p(y^{(i)} \mid x^{(i)}; \theta)$$

$$= \prod_{i=1}^m (h_{\theta}(x^{(i)}))^{y^{(i)}} (1 - h_{\theta}(x^{(i)}))^{1-y^{(i)}}$$

$$l(\theta) = \log L(\theta) = \sum_{i=1}^m y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))$$

对数似然函数

$$\begin{aligned} \frac{\partial l(\theta)}{\partial \theta_j} &= \sum_{i=1}^m \left(\frac{y^{(i)}}{h(x^{(i)})} - \frac{1 - y^{(i)}}{1 - h(x^{(i)})} \right) \cdot \frac{\partial h(x^{(i)})}{\partial \theta_j} \\ &= \sum_{i=1}^m \left(\frac{y^{(i)}}{g(\theta^T x^{(i)})} - \frac{1 - y^{(i)}}{1 - g(\theta^T x^{(i)})} \right) \cdot \frac{\partial g(\theta^T x^{(i)})}{\partial \theta_j} \\ &= \sum_{i=1}^m \left(\frac{y^{(i)}}{g(\theta^T x^{(i)})} - \frac{1 - y^{(i)}}{1 - g(\theta^T x^{(i)})} \right) \cdot g(\theta^T x^{(i)}) \cdot (1 - g(\theta^T x^{(i)})) \cdot \frac{\partial \theta^T x^{(i)}}{\partial \theta_j} \\ &= \sum_{i=1}^m (y^{(i)}(1 - g(\theta^T x^{(i)})) - (1 - y^{(i)})g(\theta^T x^{(i)})) \cdot x_j^{(i)} \\ &= \sum_{i=1}^m (y^{(i)} - g(\theta^T x^{(i)})) \cdot x_j^{(i)} \end{aligned}$$

参数的迭代

□ Logistic回归参数的学习规则：

$$\theta_j := \theta_j + \alpha (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)}$$

□ 比较上面的结果和线性回归的结论的差别：

■ 它们具有相同的形式！

Repeat until convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^m (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)}$$

}

Loop {

for i=1 to m, {

$$\theta_j := \theta_j + \alpha (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)}$$

}

}

对数线性模型

- 一个事件的几率odds，是指该事件发生的概率与该事件不发生的概率的比值。
- 对数几率：logit函数

$$P(y = 1 \mid x; \theta) = h_{\theta}(x)$$

$$P(y = 0 \mid x; \theta) = 1 - h_{\theta}(x)$$

$$\log it(p) = \log \frac{p}{1-p} = \log \frac{h_{\theta}(x)}{1-h_{\theta}(x)} = \log \left(\frac{\frac{1}{1+e^{-\theta^T x}}}{\frac{e^{-\theta^T x}}{1+e^{-\theta^T x}}} \right) = \theta^T x$$

Logistic回归的损失函数 $y_i \in \{0,1\}$

$$L(\theta) = \prod_{i=1}^m p_i^{y_i} (1 - p_i)^{1-y_i}$$

$$\hat{y}_i = \begin{cases} p_i & y_i = 1 \\ 1 - p_i & y_i = 0 \end{cases}$$

$$\Rightarrow l(\theta) = \sum_{i=1}^m \ln \left[p_i^{y_i} (1 - p_i)^{1-y_i} \right]$$

$$\xrightarrow{p_i = \frac{1}{1+e^{-f_i}}} l(\theta) = \sum_{i=1}^m \ln \left[\left(\frac{1}{1+e^{-f_i}} \right)^{y_i} \left(\frac{1}{1+e^{f_i}} \right)^{1-y_i} \right]$$

$$\therefore \text{loss}(y_i, \hat{y}_i) = -l(\theta)$$

$$= \sum_{i=1}^m \left[y_i \ln(1 + e^{-f_i}) + (1 - y_i) \ln(1 + e^{f_i}) \right]$$

Logistic回归的损失: $y_i \in \{-1, 1\}$

$$L(\theta) = \prod_{i=1}^m p_i^{(y_i+1)/2} (1-p_i)^{-(y_i-1)/2} \Rightarrow l(\theta) = \sum_{i=1}^m \ln \left[p_i^{(y_i+1)/2} (1-p_i)^{-(y_i-1)/2} \right]$$

$$\xrightarrow{p_i = \frac{1}{1+e^{-f_i}}} l(\theta) = \sum_{i=1}^m \ln \left[\left(\frac{1}{1+e^{-f_i}} \right)^{(y_i+1)/2} \left(\frac{1}{1+e^{f_i}} \right)^{-(y_i-1)/2} \right]$$

$$\therefore \text{loss}(y_i, \hat{y}_i) = -l(\theta)$$

$$= \sum_{i=1}^m \left[\frac{1}{2} (y_i + 1) \ln(1 + e^{-f_i}) - \frac{1}{2} (y_i - 1) \ln(1 + e^{f_i}) \right]$$

$$= \begin{cases} \sum_{i=1}^m [\ln(1 + e^{-f_i})] & y_i = 1 \\ \sum_{i=1}^m [\ln(1 + e^{f_i})] & y_i = -1 \end{cases} \Rightarrow \text{loss}(y_i, \hat{y}_i) = \sum_{i=1}^m [\ln(1 + e^{-y_i \cdot f_i})]$$

$$\begin{aligned} y_i &= \{-1, 1\} \\ \hat{y}_i &= \begin{cases} p_i & y_i = 1 \\ 1 - p_i & y_i = -1 \end{cases} \end{aligned}$$

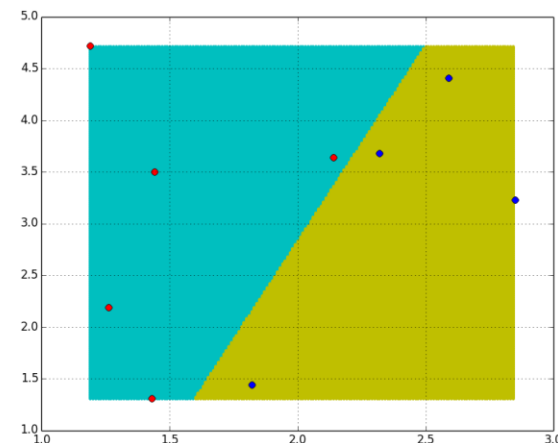
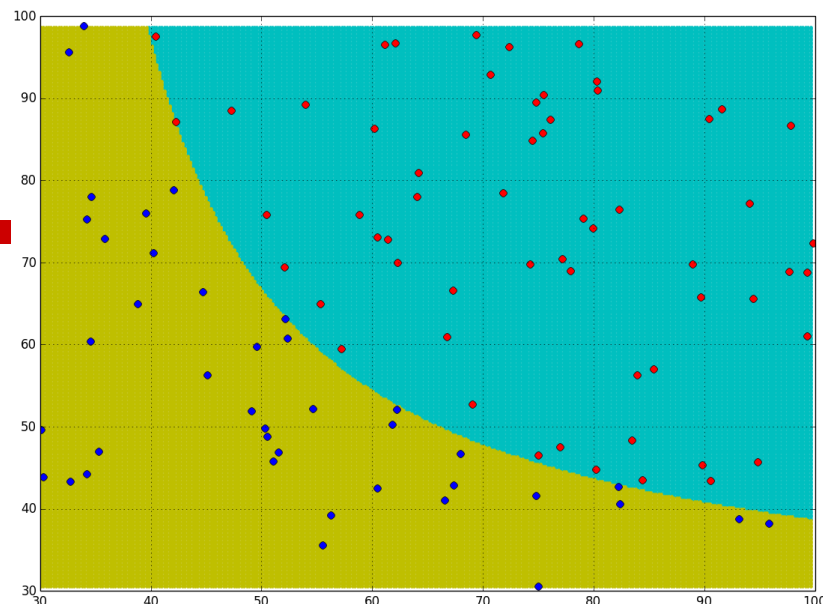
分类：Logistic回归

□ 沿似然函数正梯度上升

□ 维度提升

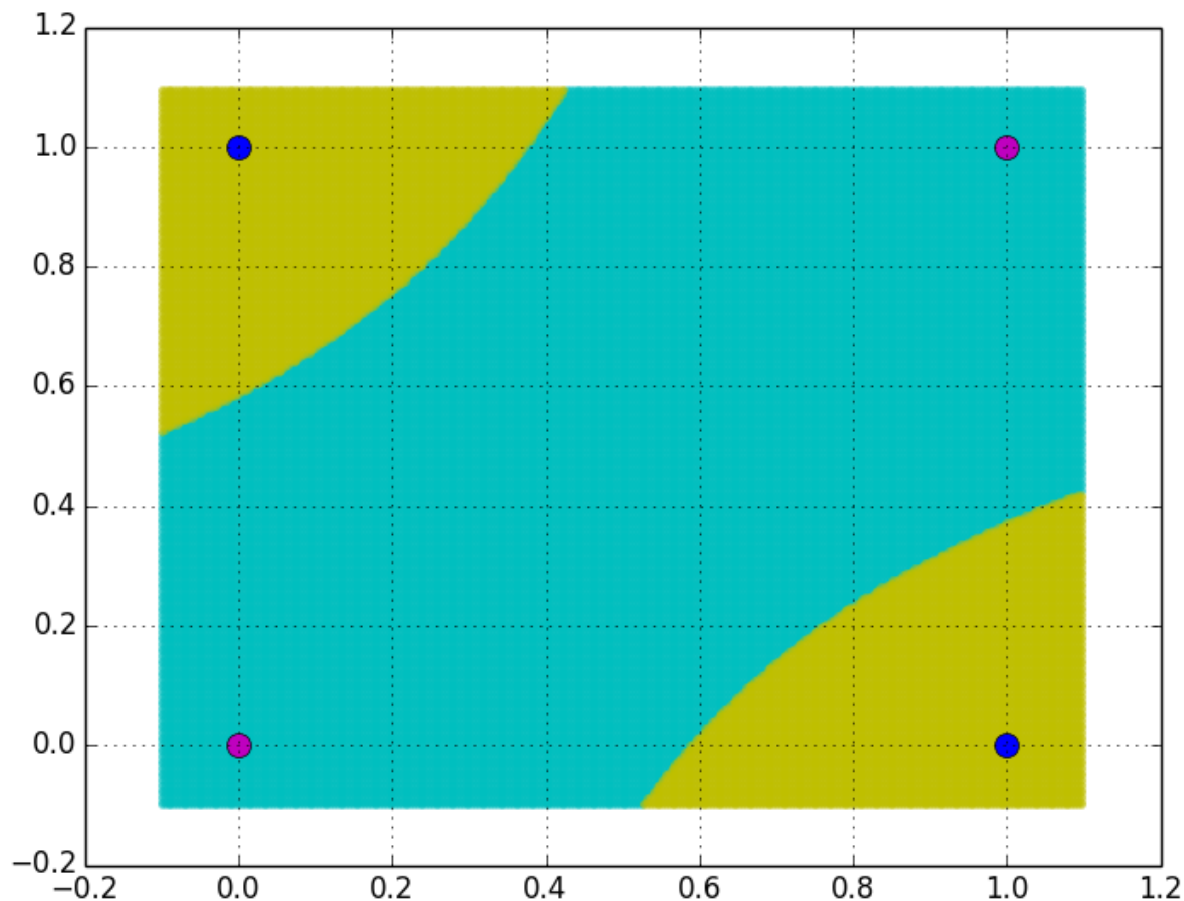
```
def logistic_regression(data, alpha, lamda):  
    n = len(data[0]) - 1  
    w = [0 for x in range(n)]  
    w2 = [0 for x in range(n)]  
    for times in range(10000):  
        for d in data:  
            for i in range(n):  
                w2[i] = w[i] + alpha * (d[n] - h(w,d))*d[i] + lamda * w[i]  
            for i in range(n):  
                w[i] = w2[i]  
            print w  
    return w
```

```
def feature_whole(x1, x2, e):  
    d = [1]  
    for n in range(1,e+1):  
        for i in range(n+1):  
            d.append(pow(x1, n-i) * pow(x2, i))  
    return d
```



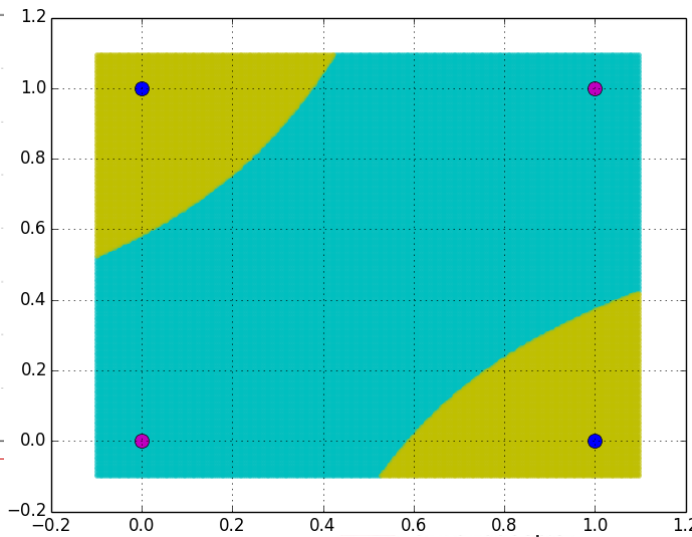
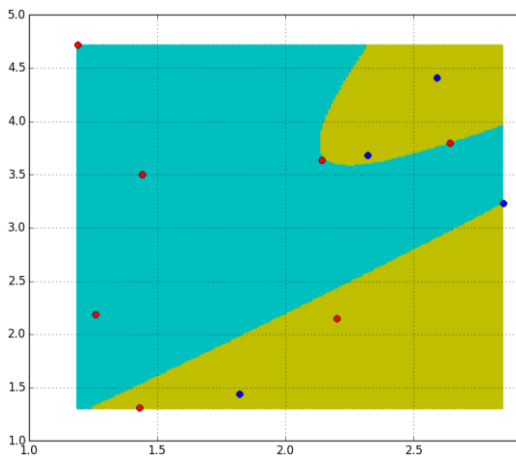
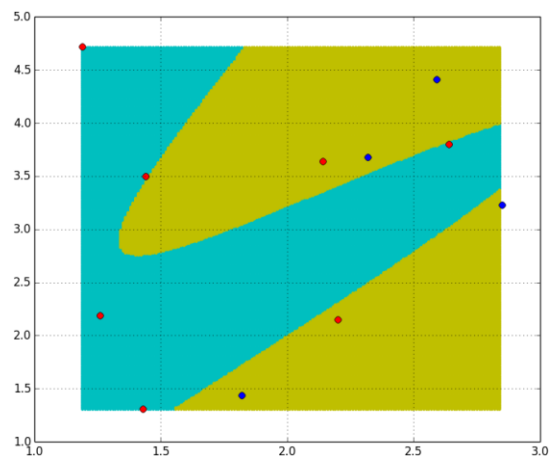
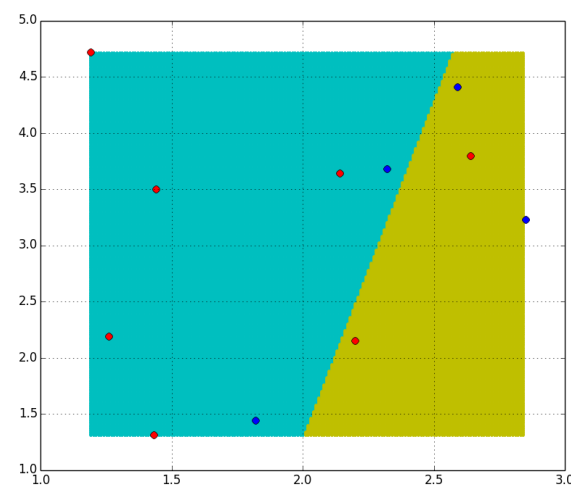
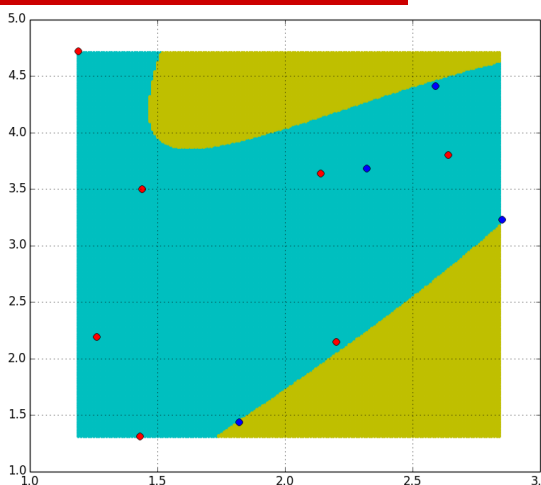
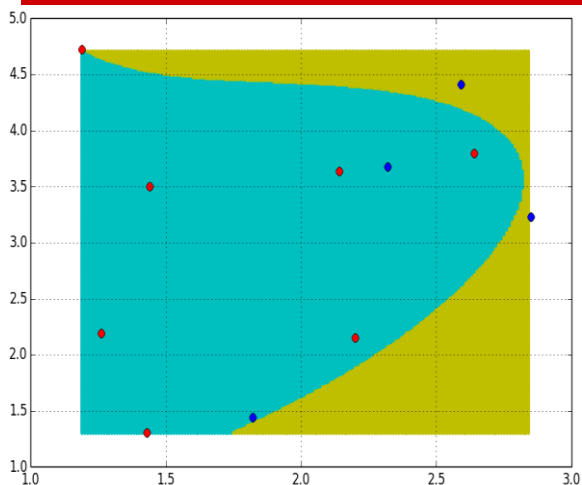
异或

x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	0



数据升维：“选取特征”

3	5	1
9	20	



广义线性模型Generalized Linear Model

□ y 不再只是正态分布，而是扩大为指数族中的任一分布；

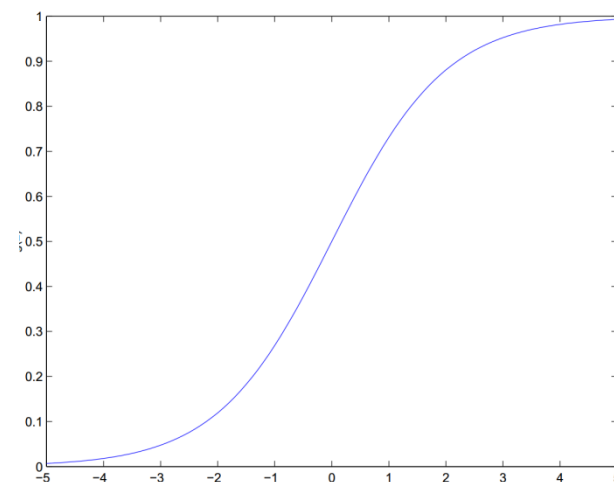
□ 变量 $x \rightarrow g(x) \rightarrow y$

■ 连接函数 g

□ 连接函数 g 单调可导

■ 如Logistic回归中的 $g(z) = \frac{1}{1 + e^{-z}}$

■ 拉伸变换：
$$g(z) = \frac{1}{1 + e^{-\lambda z}}$$



广义线性模型

Common distributions with typical uses and canonical link functions

Distribution	Support of distribution	Typical uses	Link name	Link function	Mean function
Normal	real: $(-\infty, +\infty)$	Linear-response data	Identity	$\mathbf{X}\beta = \mu$	$\mu = \mathbf{X}\beta$
Exponential	real: $(0, +\infty)$	Exponential-response data, scale parameters	Inverse	$\mathbf{X}\beta = \mu^{-1}$	$\mu = (\mathbf{X}\beta)^{-1}$
Gamma					
Inverse Gaussian	real: $(0, +\infty)$		Inverse squared	$\mathbf{X}\beta = \mu^{-2}$	$\mu = (\mathbf{X}\beta)^{-1/2}$
Poisson	integer: $0, 1, 2, \dots$	count of occurrences in fixed amount of time/space	Log	$\mathbf{X}\beta = \ln(\mu)$	$\mu = \exp(\mathbf{X}\beta)$
Bernoulli	integer: $\{0, 1\}$	outcome of single yes/no occurrence	Logit	$\mathbf{X}\beta = \ln\left(\frac{\mu}{1-\mu}\right)$	$\mu = \frac{\exp(\mathbf{X}\beta)}{1 + \exp(\mathbf{X}\beta)} = \frac{1}{1 + \exp(-\mathbf{X}\beta)}$
Binomial	integer: $0, 1, \dots, N$	count of # of "yes" occurrences out of N yes/no occurrences			
Categorical	integer: $[0, K)$	outcome of single K-way occurrence			
	K-vector of integer: $[0, 1]$, where exactly one element in the vector has the value 1				
Multinomial	K-vector of integer: $[0, N]$	count of occurrences of different types (1 .. K) out of N total K-way occurrences			

Softmax回归

□ K分类，第k类的参数为 $\vec{\theta}_k$ ，组成二维矩阵 $\theta_{k \times n}$

□ 概率： $p(c = k | x; \theta) = \frac{\exp(\theta_k^T x)}{\sum_{l=1}^K \exp(\theta_l^T x)}$, $k = 1, 2, \dots, K$

□ 似然函数： $L(\theta) = \prod_{i=1}^m \prod_{k=1}^K p(c = k | x^{(i)}; \theta)^{y_k^{(i)}} = \prod_{i=1}^m \prod_{k=1}^K \left(\frac{\exp(\theta_k^T x^{(i)})}{\sum_{l=1}^K \exp(\theta_l^T x^{(i)})} \right)^{y_k^{(i)}}$

□ 对数似然： $J_m(\theta) = \ln L(\theta) = \sum_{i=1}^m \sum_{k=1}^K y_k^{(i)} \cdot \left(\theta_k^T x^{(i)} - \ln \sum_{l=1}^K \exp(\theta_l^T x^{(i)}) \right)$

□ 随机梯度： $J(\theta) = \sum_{k=1}^K y_k \cdot \left(\theta_k^T x - \ln \sum_{l=1}^K \exp(\theta_l^T x) \right)$

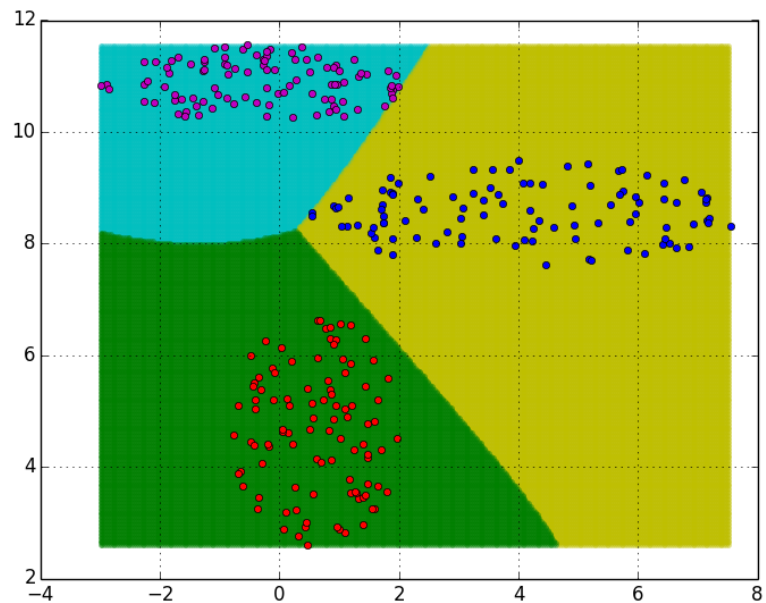
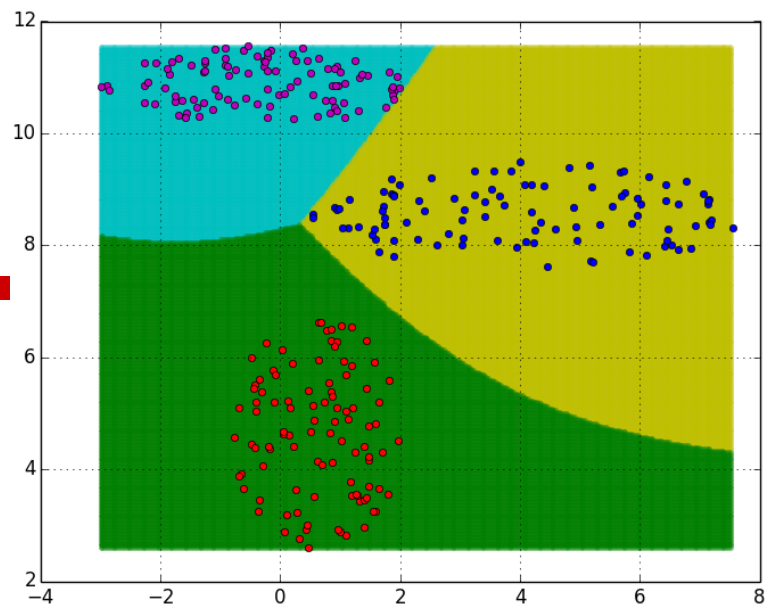
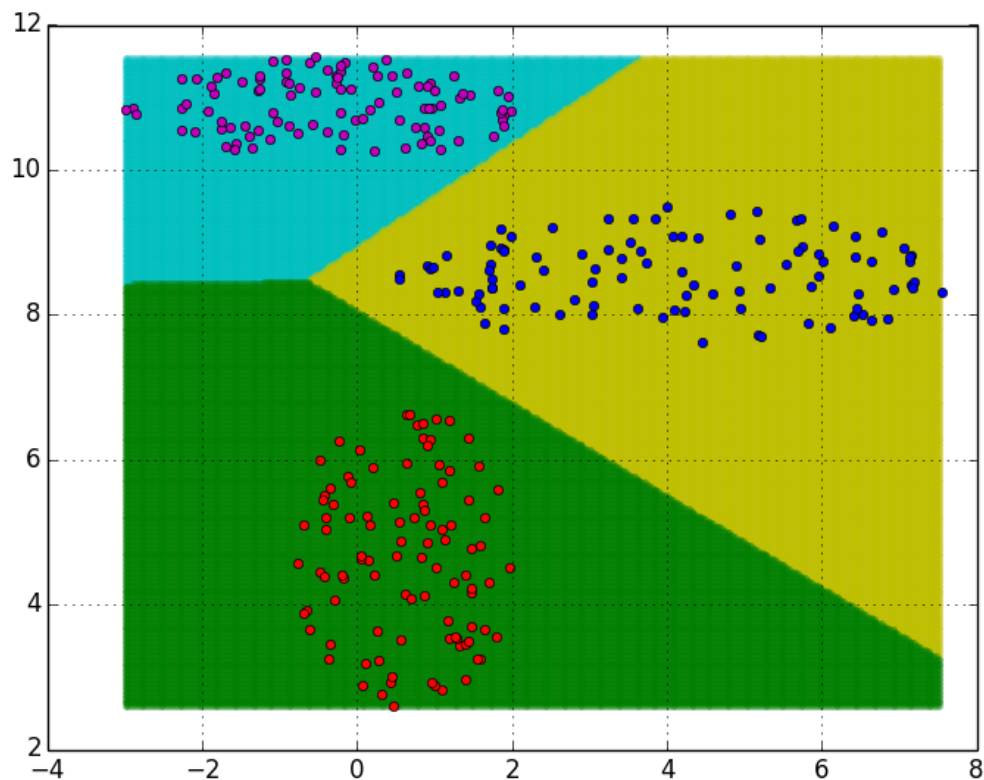
$$\frac{\partial J(\theta)}{\partial \theta_k} = (y_k - p(y_k | x; \theta)) \cdot x$$

Code

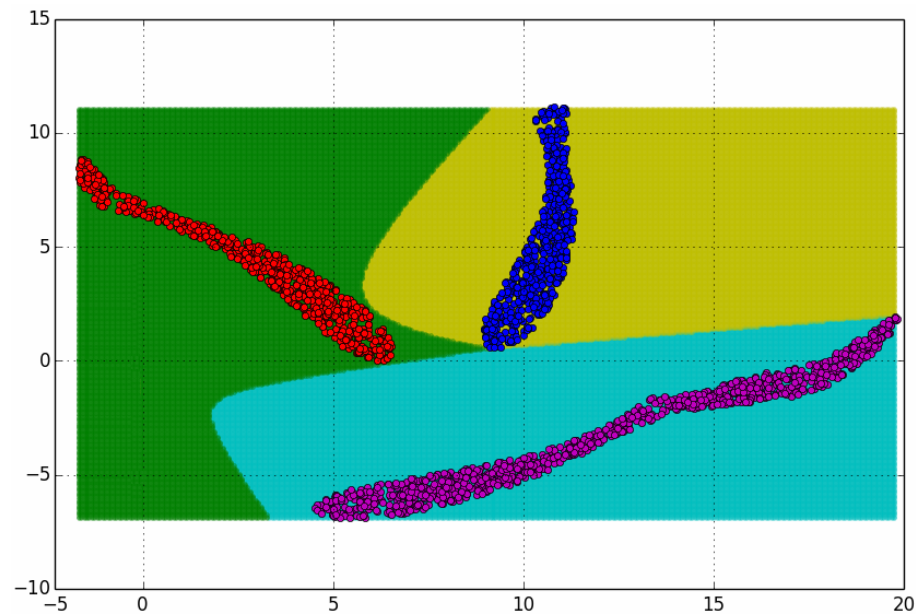
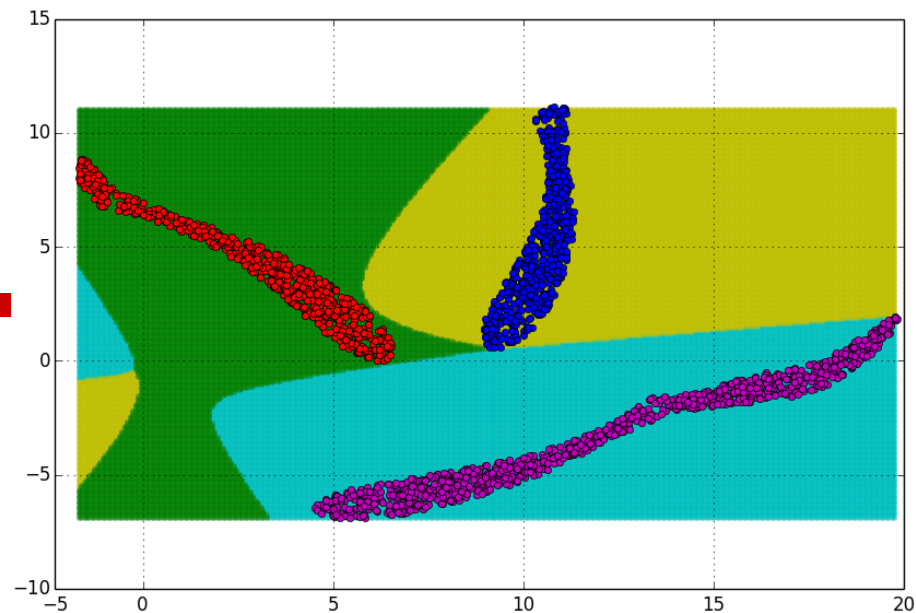
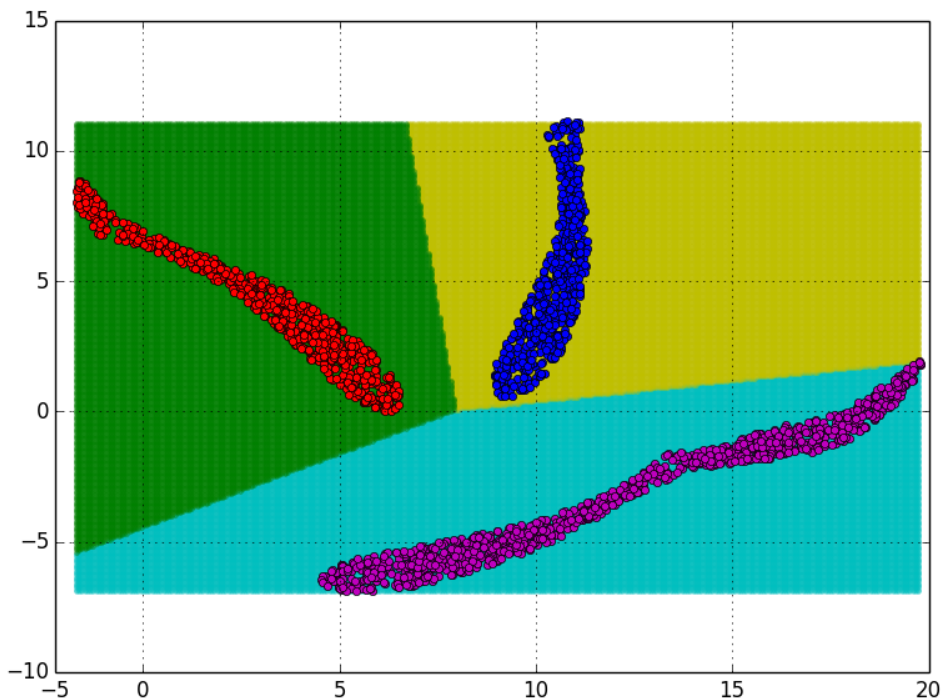
```
def soft_max(data, K, alpha, lamda):  
    n = len(data[0]) - 1    # 样本维度  
    w = np.zeros((K, n))    # 当前权值: 每个类别有各自的权值  
    wNew = np.zeros((K, n))    # 临时权值: 迭代过程中的权值  
    for times in range(1000):  
        for d in data:  
            x = d[:-1]        # 输入向量  
            for k in range(K):    # K: 类别数目  
                y = 0            # 期望输出(标量)  
                if int(d[-1] + 0.5) == k:  
                    y = 1  
                p = predict(w, x, k)  
                g = (y - p) * x    # 梯度(n维列向量)  
                wNew[k] = w[k] + (alpha * g + lamda * w[k])  
            w = wNew.copy()  
    return w
```

Softmax分类

1 | $\frac{2}{3}$



特征选择



骰子问题

- 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2.71828，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 $(p_1, p_2 \dots p_6)$ ，用向量 \mathbf{p} 表示。

□ 目标函数： $H(\vec{p}) = -\sum_{i=1}^6 p_i \ln p_i$

□ 约束条件： $\sum_{i=1}^6 p_i = 1 \quad \sum_{i=1}^6 i \cdot p_i = 2.71828$

□ Lagrange函数：

$$L(\vec{p}, \lambda_1, \lambda_2) = -\sum_{i=1}^6 p_i \ln p_i + \lambda_1 \left(1 - \sum_{i=1}^6 p_i \right) + \lambda_2 \left(2.71828 - \sum_{i=1}^6 i \cdot p_i \right)$$

□ 求解：

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

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

$$\Rightarrow \lambda_1 = -0.0787, \lambda_2 = 0.2808$$

使用梯度下降计算Lagrange乘子

□ 根据 p_i 的解： $p_i = e^{-1-\lambda_1-i\cdot\lambda_2}$, $i=1,2,\dots,5,6$

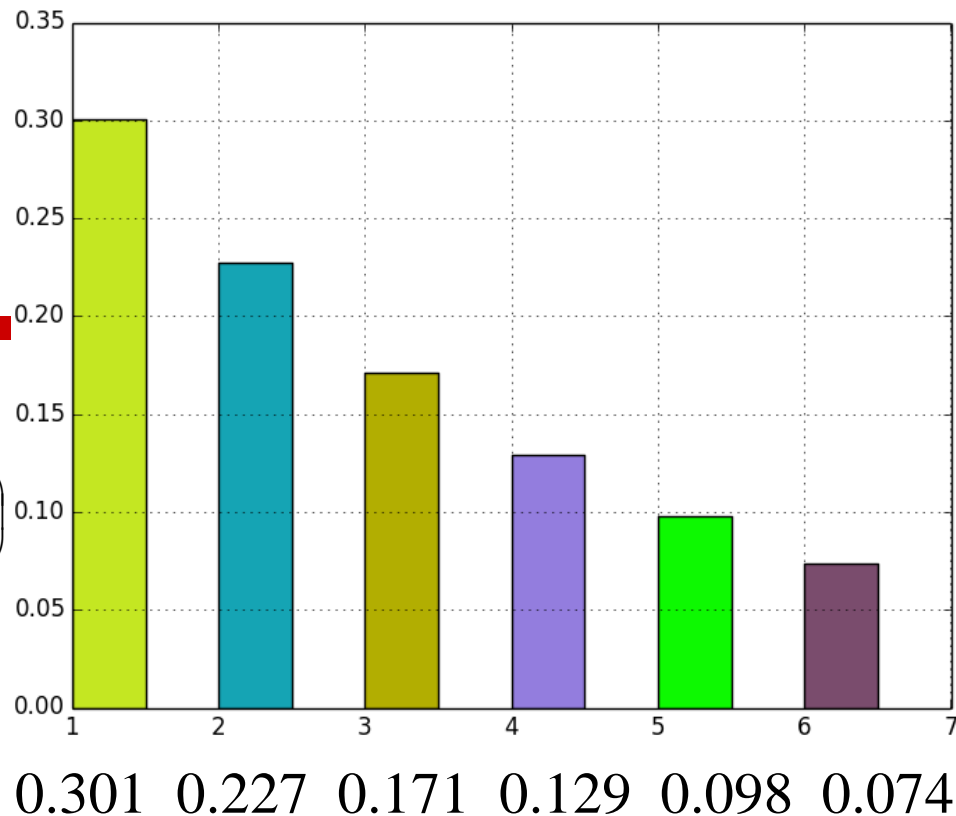
□ 构造目标函数并计算梯度：

$$J(\lambda) = \left(\sum_{i=1}^6 p_i - 1 \right)^2 + \left(\sum_{i=1}^6 i \cdot p_i - E \right)^2$$
$$\Rightarrow \begin{cases} \frac{\partial J(\lambda)}{\partial \lambda_1} = -2 \left(\sum_{i=1}^6 p_i - 1 \right) \cdot \left(\sum_{i=1}^6 p_i \right) - 2 \left(\sum_{i=1}^6 i \cdot p_i - E \right) \cdot \left(\sum_{i=1}^6 i \cdot p_i \right) \\ \frac{\partial J(\lambda)}{\partial \lambda_2} = -2 \left(\sum_{i=1}^6 p_i - 1 \right) \cdot \left(\sum_{i=1}^6 i \cdot p_i \right) - 2 \left(\sum_{i=1}^6 i \cdot p_i - E \right) \cdot \left(\sum_{i=1}^6 i^2 \cdot p_i \right) \end{cases}$$

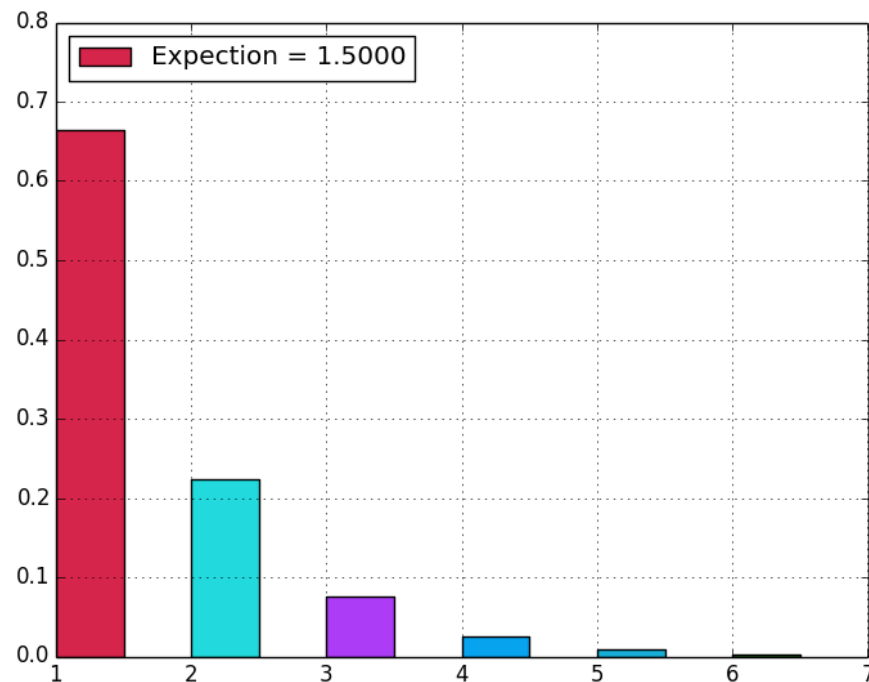
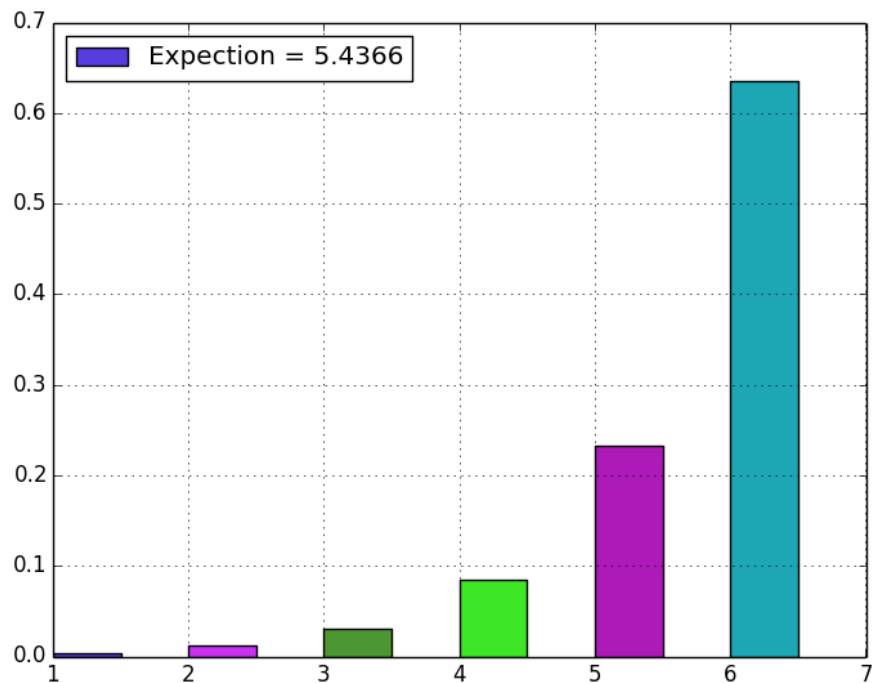
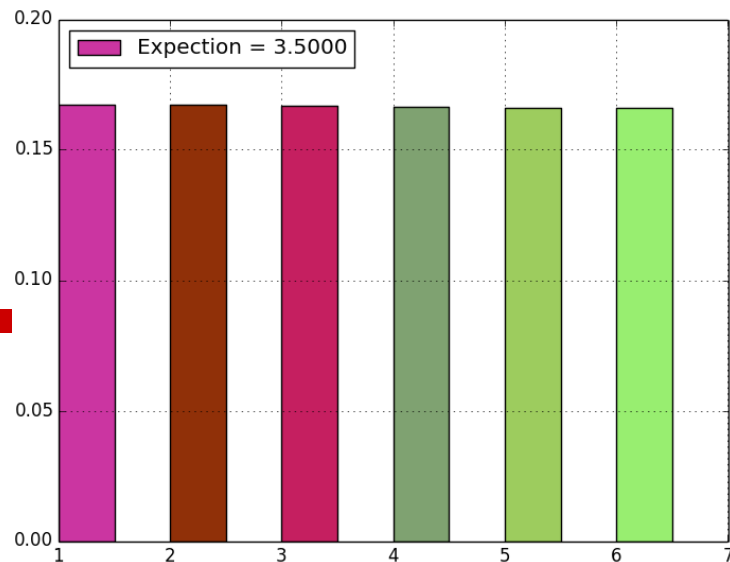
预测结果

$$\begin{cases} \frac{\partial J(\lambda)}{\partial \lambda_1} = -2 \left(\sum_{i=1}^6 p_i - 1 \right) \cdot \left(\sum_{i=1}^6 p_i \right) - 2 \left(\sum_{i=1}^6 i \cdot p_i - E \right) \cdot \left(\sum_{i=1}^6 i \cdot p_i \right) \\ \frac{\partial J(\lambda)}{\partial \lambda_2} = -2 \left(\sum_{i=1}^6 p_i - 1 \right) \cdot \left(\sum_{i=1}^6 i \cdot p_i \right) - 2 \left(\sum_{i=1}^6 i \cdot p_i - E \right) \cdot \left(\sum_{i=1}^6 i^2 \cdot p_i \right) \end{cases}$$

```
if __name__ == "__main__":
    theta1, theta2 = 0, 0
    i = np.arange(1, 7)
    i2 = i ** 2
    alpha = 0.01
    beta = 0.1
    a = np.zeros(6)
    for times in range(5000):
        a = -np.arange(1, 7) * theta2 - theta1 - 1
        a = np.exp(a)
        f1 = np.dot(a, i)
        f2 = np.sum(a)
        theta1 += alpha * (f1 - np.e) * f1 + beta * (f2 - 1) * f2
        theta2 += alpha * (f1 - np.e) * np.dot(a, i2) + beta * (f2 - 1) * f1
    show(a/a.sum())
```



目标函数的有效性



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 $h(XY)=h(X)+h(Y)$ 。

□ 定义随机变量X的概率分布为 $p(x)$ ，从而定义X信息量： $h(x)=-\log_2 p(x)$

□ 思考：事件X的信息量的期望如何计算呢？

熵

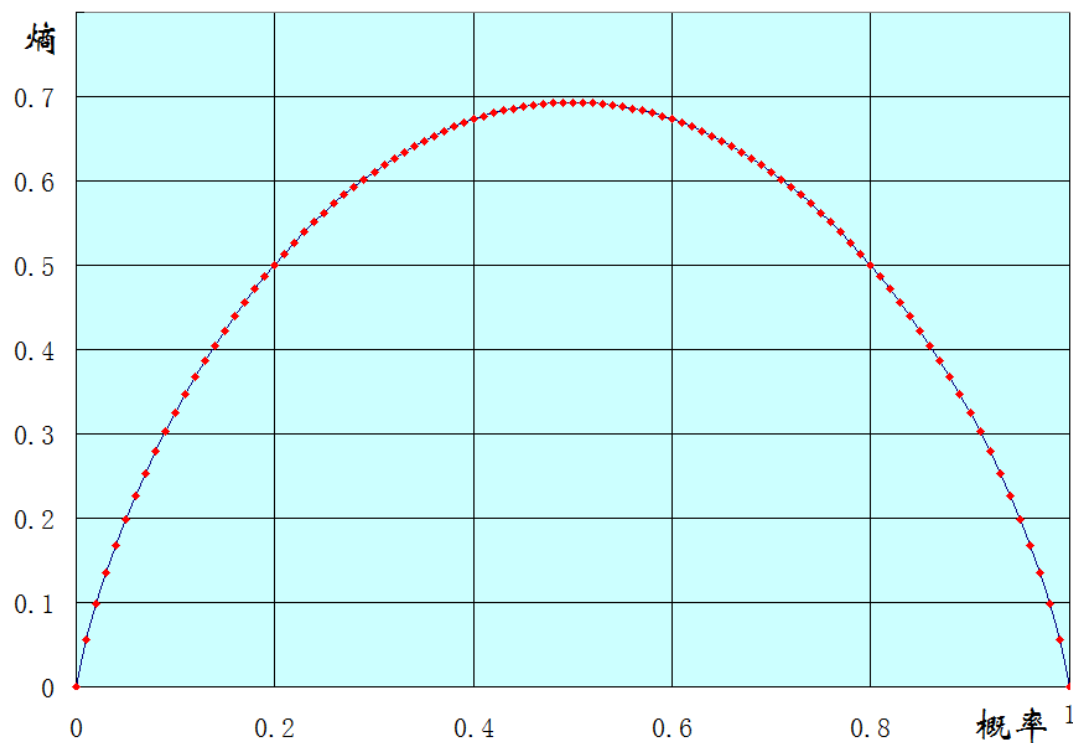
□ 对随机事件的信息量求期望，得熵的定义：

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

- 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)

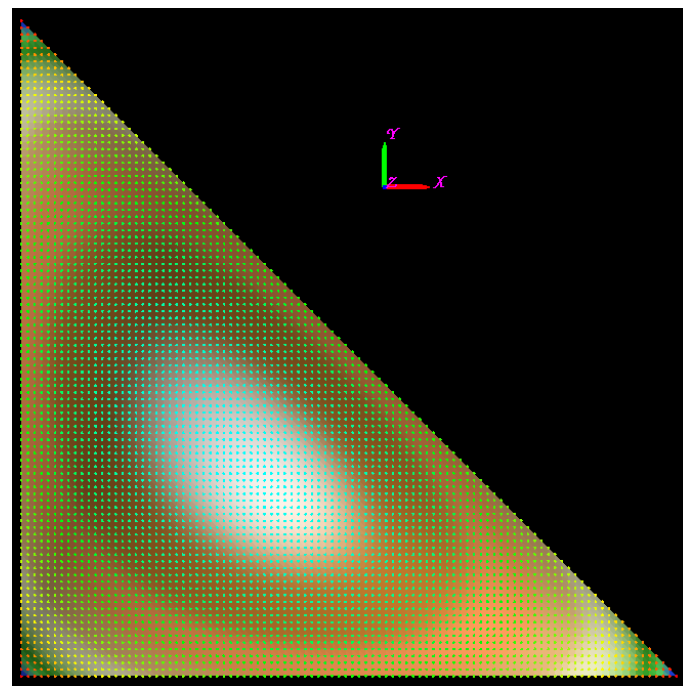
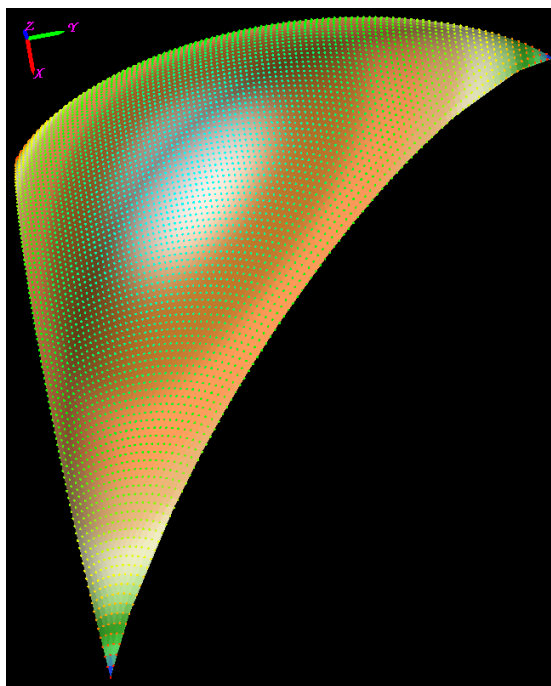
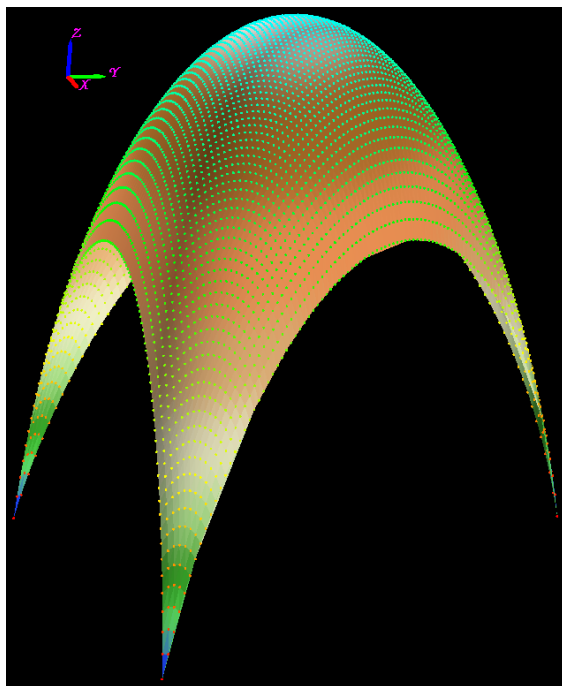
两点分布的熵

□ 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？

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

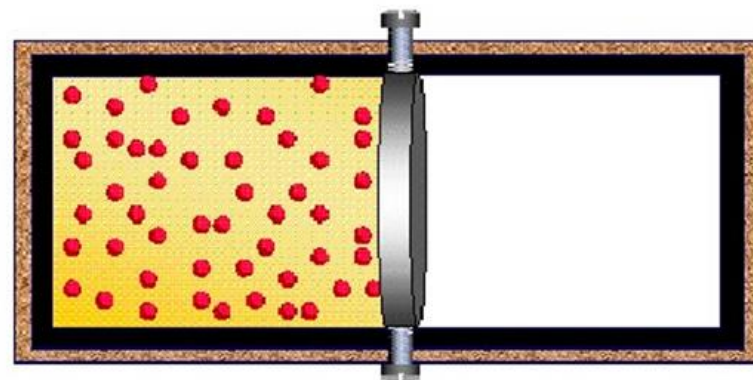


公式推导 $N \rightarrow \infty \Rightarrow \ln N! \rightarrow N(\ln N - 1)$

$$\begin{aligned} H &= \frac{1}{N} \ln \frac{N!}{\prod_{i=1}^k n_i!} = \frac{1}{N} \ln(N!) - \frac{1}{N} \sum_{i=1}^k \ln(n_i!) \\ &\rightarrow (\ln N - 1) - \frac{1}{N} \sum_{i=1}^k n_i (\ln n_i - 1) \\ &= \ln N - \frac{1}{N} \sum_{i=1}^k n_i \ln n_i = -\frac{1}{N} \left(\left(\sum_{i=1}^k n_i \ln n_i \right) - N \ln N \right) \\ &= -\frac{1}{N} \sum_{i=1}^k (n_i \ln n_i - n_i \ln N) = -\frac{1}{N} \sum_{i=1}^k \left(n_i \ln \frac{n_i}{N} \right) \\ &= -\sum_{i=1}^k \left(\frac{n_i}{N} \ln \frac{n_i}{N} \right) \rightarrow -\sum_{i=1}^k (p_i \ln p_i) \end{aligned}$$

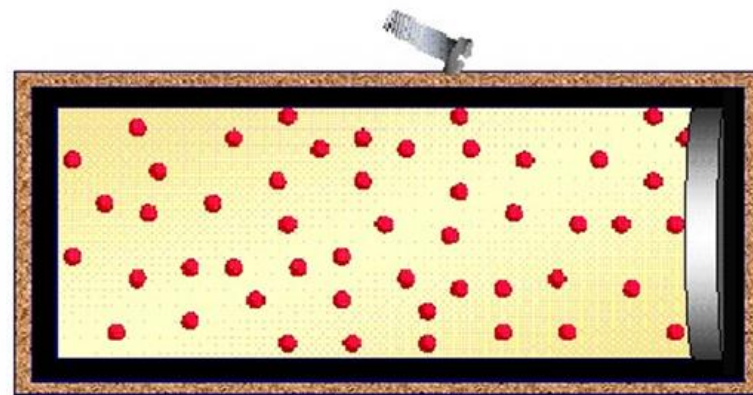
自封闭系统的运动总是倒向均匀分布

- 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 $p_i = \frac{1}{N}$, $i = 1, 2, \dots, N$

□ 计算熵：

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

□ 思考：连续均匀分布的熵如何计算？

最大熵的理解 $0 \leq H(X) \leq \log|X|$

- 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

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

□ 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： $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt$ $\Gamma(n) = (n-1)!$

- 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

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

□ 使用方差公式化简约束条件

$$\begin{aligned} Var(X) &= E(X^2) - E^2(X) \\ \Rightarrow E(X^2) &= E^2(X) + Var(X) = \mu^2 + \sigma^2 \end{aligned}$$

□ 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

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

$$\begin{aligned} L(p) &= -\sum_x p(x) \ln p(x) + \lambda_1 (E(X) - \mu) + \lambda_2 (E(X^2) - \mu^2 - \sigma^2) \\ &= -\sum_x p(x) \ln p(x) + \lambda_1 \left(\sum_x x p(x) - \mu \right) + \lambda_2 \left(\sum_x x^2 p(x) - \mu^2 - \sigma^2 \right) \\ &\Rightarrow \frac{\partial L}{\partial p} = -\ln p(x) - 1 + \lambda_1 x + \lambda_2 x^2 \stackrel{\Delta}{=} 0 \Rightarrow \ln p(x) = \lambda_2 x^2 + \lambda_1 x - 1 \end{aligned}$$

□ $P(x)$ 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
- $H(X,Y) - H(X)$
 - (X,Y) 发生所包含的熵, 减去 X 单独发生包含的熵: 在 X 发生的前提下, Y 发生“新”带来的熵
 - 该式子定义为 X 发生前提下, Y 的熵:
 - 条件熵 $H(Y|X)$

推导条件熵的定义式

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

根据条件熵的定义式，可以得到

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

相对熵

- 相对熵，又称互熵，交叉熵，鉴别信息，Kullback熵，Kullback-Leibler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

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

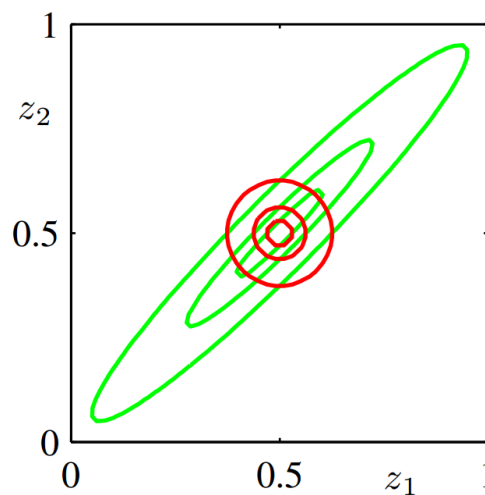
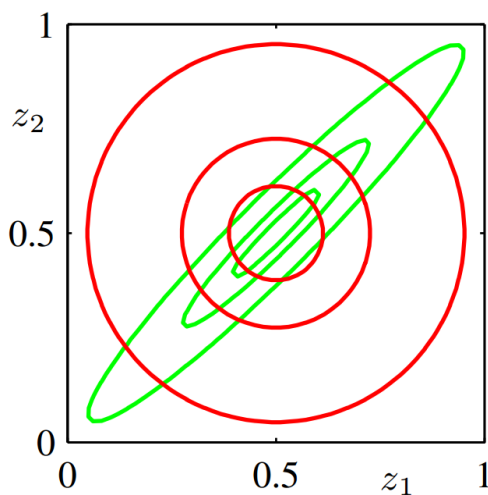
- 说明：
 - 相对熵可以度量两个随机变量的“距离”
 - 在“贝叶斯网络”、“变分推导”等章节会再次遇到
 - 一般的， $D(p \parallel q) \neq D(q \parallel p)$
 - $D(p \parallel q) \geq 0$ 、 $D(q \parallel p) \geq 0$ ：凸函数中的Jensen不等式

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；

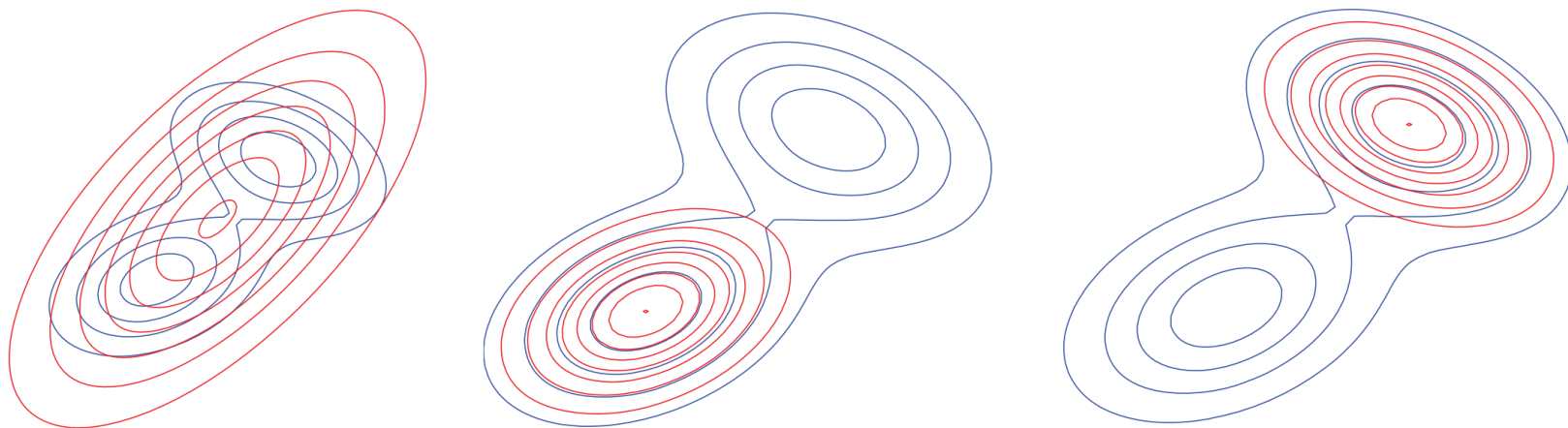
两个KL散度的区别

- 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。

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

计算条件熵的定义式： $H(Y)-I(X,Y)$

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

整理得到的等式

□ $H(Y|X) = H(X,Y) - H(X)$

■ 条件熵定义

□ $H(Y|X) = H(Y) - I(X,Y)$

■ 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

■ $H(X|Y) = H(X,Y) - H(Y)$

■ $H(X|Y) = H(X) - I(X,Y)$

□ $I(X,Y) = H(X) + H(Y) - H(X,Y)$

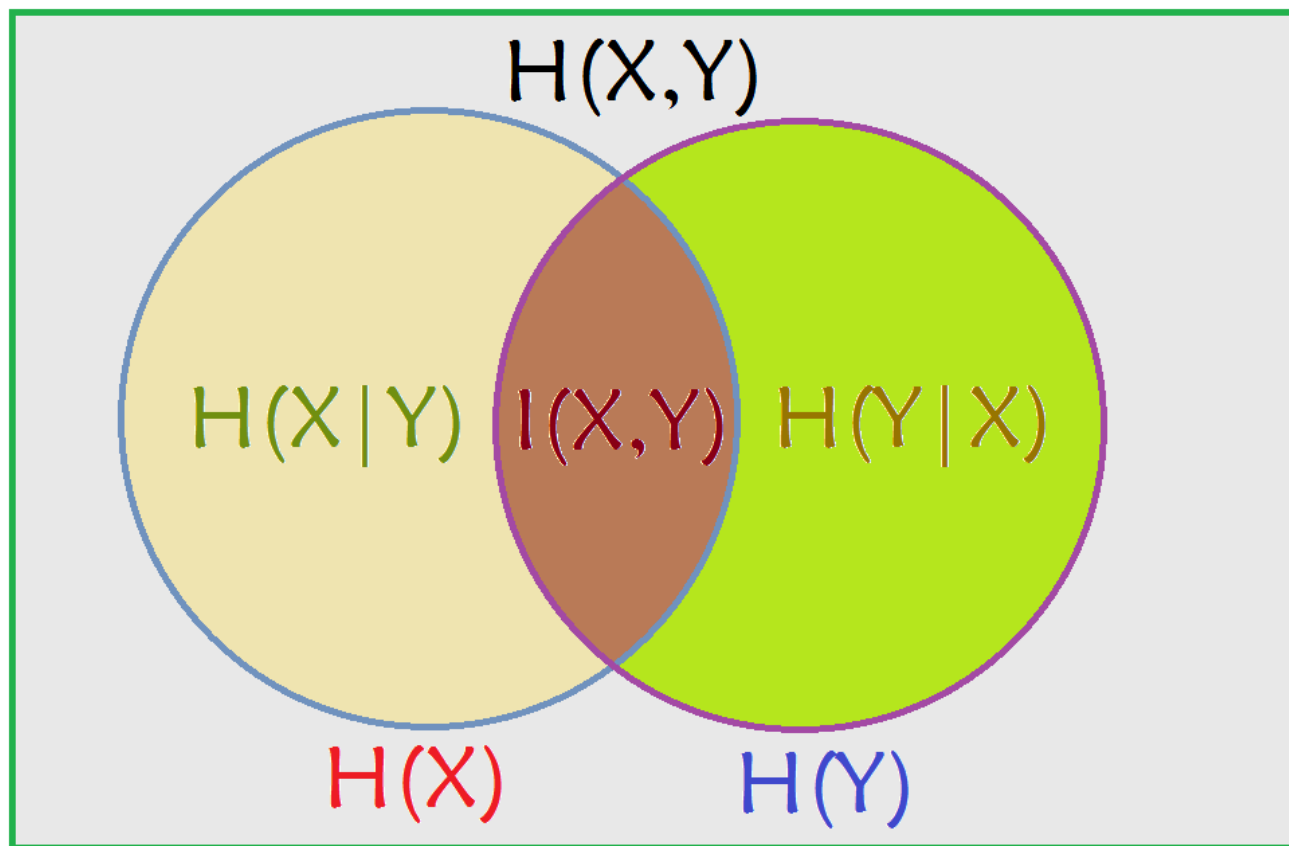
■ 有些文献将该式作为互信息的定义式

□ 试证明： $H(X|Y) \leq H(X)$, $H(Y|X) \leq H(Y)$

互信息： $I(X,Y)=H(X)+H(Y)-H(X,Y)$

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

强大的Venn图：帮助记忆



思考题：天平与假币

□ 有12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？

总结和思考

□ Logistic/Softmax 回归是实践中解决分类问题的最重要方法。

■ 方法简单、容易实现、效果良好、易于解释

■ 不止是分类：推荐系统

□ 思考

■ Logistic 回归的目标函数，可否用相对熵解释？

■ 信息熵有什么用？

作业

- ☐ 推导Logistic/Softmax回归的梯度。
- ☐ 思考广义线性回归中link function的作用。
- ☐ 自己实现一个Softmax回归，并用于鸢尾花数据的分类。

参考文献

- Prof. Andrew Ng. *Machine Learning*. Stanford University
- 李航, 统计学习方法, 清华大学出版社, 2012

我们在这里

□ <http://wenda.ChinaHadoop.cn>

■ 视频/课程/社区

□ 微博

■ @ChinaHadoop

■ @邹博_机器学习

□ 微信公众号

■ 小象学院

■ 大数据分析挖掘

互联网新技术在线教育领航者

小象问答 搜索标题、用户 全站内容搜索 提问 首页 动态 发现 话题 通知

全部 招聘求职 机器学习 大数据平台技术 DCon 大数据行业应用 NoSQL数据库 数据科学 江湖救急

发现 最新 推荐 热门 等待回复

graphviz has no attribute 'write' 贡献
邹博 回复了问题 • 2 人关注 • 1 个回复 • 3 次浏览 • 2017-04-09 15:48

sklearn中如何理解Pipeline机制 贡献
数据分析与数据挖掘 邹博 回复了问题 • 2 人关注 • 1 个回复 • 28 次浏览 • 2017-04-09 15:39

关于9.Logistic回归的ppt中第9页的对数线性函数 贡献
机器学习 邹博 回复了问题 • 3 人关注 • 3 个回复 • 39 次浏览 • 2017-04-09 15:35

关于“贝叶斯估计中，最大后验概率估计就是结构化风险最小化的例子：当模型是条件概率分布，损失函数为对数损失函数，模型的复杂度由模型的先验概率表示，结构化风险最小化就等价于最大后验概率估计” 贡献
机器学习 邹博 回复了问题 • 2 人关注 • 1 个回复 • 26 次浏览 • 2017-04-09 15:27

关于连续值的预测 贡献
咨询 邹博 回复了问题 • 2 人关注 • 1 个回复 • 31 次浏览 • 2017-04-09 15:24

拉格朗日对偶函数为什么一定是凸函数 贡献
数据科学 邹博 回复了问题 • 2 人关注 • 2 个回复 • 26 次浏览 • 2017-04-09 15:20

梯度下降公式中的斯堪J 是 贡献
机器学习 邹博 回复了问题 • 2 人关注 • 1 个回复 • 29 次浏览 • 2017-04-09 15:17

深度学习适合做预测吗？ 贡献
深度学习 邹博 回复了问题 • 2 人关注 • 1 个回复 • 27 次浏览 • 2017-04-09 15:15

关于6.4PCA_FeatureSelection.py中plt.legend的参数疑问 贡献
机器学习 邹博 回复了问题 • 2 人关注 • 1 个回复 • 28 次浏览 • 2017-04-09 15:04

@邹博 有哪些可以下载数据源的网站？ 贡献
数据分析与数据挖掘 邹博 回复了问题 • 4 人关注 • 1 个回复 • 31 次浏览 • 2017-04-09 14:53

LDA主题模型 贡献
机器学习 邹博 回复了问题 • 2 人关注 • 1 个回复 • 29 次浏览 • 2017-04-09 14:45

代码10.6bagging_ridged老师提到了采样率设为0.2能够使峰值部分的数据被体现出来。这是为什么呢？ 贡献
机器学习 邹博 回复了问题 • 2 人关注 • 1 个回复 • 22 次浏览 • 2017-04-09 14:26

GraphViz's executables not found 贡献
机器学习 邹博 回复了问题 • 3 人关注 • 2 个回复 • 23 次浏览 • 2017-04-09 13:47

决策树中关于feature_importances代码的问题 贡献
机器学习 邹博 回复了问题 • 2 人关注 • 1 个回复 • 6 次浏览 • 2017-04-09 13:11

专题
招聘求职
大数据行业应用
数据科学
系统与编程
云计算技术

热门话题 更多 >
机器学习 907 个问题, 230 人关注
spark 387 个问题, 172 人关注
hadoop 1059 个问题, 155 人关注
python数据分析 171 个问题, 28 人关注
数据分析与数据挖掘 54 个问题, 111 人关注

热门用户 更多 >
小心巴 14 个问题, 0 次赞同
叉叉V 45 个问题, 22 次赞同
铁甲无声 10 个问题, 0 次赞同
带刀锦衣卫 13 个问题, 0 次赞同

感谢大家！

恳请大家批评指正！