人工智能工程基础

数学基础

课程网址： mooc.nju.edu.cn

彭成磊

SEE NJU

2022年3月

目录 I

目录

标量、向量和矩阵

- 标量(scalar)：只有大小(数值)，没有方向的量。小写变量名。
  - n t, α
- 向量(vector)：矢量，一列数，粗体小写x。向量元素用普通小写加下标x_1。
  - $\mathbf{x}_1^{(i)}, \mathbf{x}^{(i)}$和$\mathbf{x}^T$分别表示啥?
  - $[1, 2, 3, 4]^T$
- 矩阵(matrix)：二维数组，粗体大写变量名，比如A。

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

numpy中的向量

```
import numpy as np
a = np.array([1, 2, 3])
print(a)
print(a.shape)
b = a.reshape(1, -1)
print(b)
print(b.shape)
c = a.reshape(-1, 1)
print(c)
print(c.shape)
```

张量

- 张量(tensor)：张量是基于向量和矩阵的推广，n维数组。
  - 标量：零维张量
  - 向量：一维张量
  - 矩阵：二维张量
- 张量A中坐标为(i, j, k)的元素记做$A_{i,j,k}$

```
d = a.reshape(2, 1, -1)
print(d)
print(d.shape)
```

数组、矩阵乘法

- matrix product: $C = AB$, $C_{i,j} = \sum_k A_{i,k} B_{k,j}$

$$C = AB = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} = \begin{bmatrix} 14 & 32 \\ 32 & 77 \end{bmatrix}$$

```
A = np.arange(1,5).reshape(2,2)
B = np.arange(0,4).reshape(2,2)
np.multiply(A,B)
np.multiply(np.mat(A),np.mat(B))
np.sum(np.multiply(np.mat(A),np.mat(B)))
np.dot(A,B)
C = np.arange(1,4)
D = np.arange(0,3)
np.dot(C,D)
np.dot(C.reshape(3,-1).T, D.reshape(3,-1))
np.dot(np.mat(A),np.mat(B))
np.mat(A)*np.mat(B)
```

单位矩阵、逆矩阵、行列式和伴随矩阵

- 单位矩阵$I_n \in R^n, \forall x \in R^n, I_n x = x$

$$I_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

- $A^{-1}A = I_n$
- $Ax = b \rightarrow x = A^{-1}b$

```
np.eye(5)
A_inv = np.linalg.inv(A)
A_det = np.linalg.det(A)
A_ad = A_inv * A_det
```

方差、标准差和协方差

- 方差(variance)： 衡量随机变量或一组数据时离散程度的度量

$$\delta^2 = \frac{\sum(x_i - \mu)^2}{n}$$

- 标准差(standard deviation)： 方差的平方根

$$\delta = \sqrt{\frac{\sum(x_i - \mu)^2}{n}}$$

- 协方差(covariance)，主要用来衡量多维数据集关系
- 协方差矩阵，计算的是两个不同维度之间的协方差，不是样本间。
  - 拿到一组样本数据，首先要明确一行是一个样本还是一个维度。

方差、标准差和协方差

- 协方差计算

$$cov(x,y)=\frac{\sum(x_i-\overline{x})(y_i-\overline{y})}{n-1}$$

```
x = [1, 2, 3, 4]
y = [3, 1, 5, 8]
np.var(x)
np.cov(x)
z = np.vstack((x, y))
np.cov(z)
np.cov(x, y)

array([[1.66666667, 3.16666667],
       [3.16666667, 8.91666667]])
```

范数

- 范数(norm)：衡量一个向量大小的量。
  - L0范数：向量中非零元素个数
  - L1范数：向量中各元素绝对值之和
  - L2范数：向量中各元素平方和的平方根(欧几里得范数，Euclidean norm)

$$\|\mathrm{x}\|_p = (\sum_i |\mathrm{x}_i|^p)^{\frac{1}{p}}$$

```python
x = np.array([1, 2, 3, 4])
x1 = x.reshape(-1,1)
n1 = np.linalg.norm(x)
n2 = np.sqrt(np.dot(x1.T, x1))
print(n1, n2)
```

特征分解与奇异值分解

- 矩阵分解意义：发现矩阵表示成数组元素时不明显的函数性质
- 特征分解(eigen decomposition)：将矩阵分解成一组特征向量和特征值
- 奇异值分解(SVD, singular value decomposition)，从方阵到矩阵

$$Ax = \lambda x$$

$$A = Q\Lambda Q^{-1}$$

$$A_{m\times n} = U_{m\times m}S_{m\times n}V_{n\times n}^T$$

```
A = np.array([[2,1,0],[1,3,1],[0,1,2]])
a, b = np.linalg.eig(A)
aa = np.diag(a)
print(np.dot(np.dot(b,aa),np.linalg.inv(b)))
u,s,v = np.linalg.svd(A)
```

目录

夹逼定理与极限

- 夹逼定理(Squeeze Theorem)
- 单调有界数列必有极限

$$\forall x \in U(x_0, r), g(x) \leq f(x) \leq h(x),$$

$$suppose \lim_{x \to x_0} g(x) = A, \lim_{x \to x_0} h(x) = A$$

$$then \lim_{x \to x_0} f(x) = A$$

$$x_0 = 2, \quad g(x) = -\frac{1}{3}x^3 + x^2 - \frac{7}{3}, \quad h(x) = \cos\left(\frac{\pi}{2}x\right)$$

$$求: \lim_{x \to 2} f(x)$$

导数

- 导数(derivative)：一个函数在某一点的导数描述了这个函数在这一点附近的变化率。
- 链式法则

$$C' = 0 \qquad (x^n)' = nx^{n-1}$$

$$(\sin x)' = \cos x \qquad (\cos x)' = -\sin x$$

$$(a^x)' = a^x \ln a \qquad (e^x)' = e^x$$

$$(\log_a x)' = \frac{1}{x}\log_a e \qquad (\ln x)' = \frac{1}{x}$$

$$(u+v)' = u' + v' \qquad (uv)' = u'v + v'u$$

泰勒定理

- 泰勒定理(Taylor’s theorem)：用导数系数构建多项式来近似表示函数在某点周围的情况

Let $f:\mathbb{R}\to\mathbb{R}$ be a function that has $k+1$ continuous derivatives in some neighborhood U of $x=a$. Then for any $x\in U$

$$f(x)=f(a)+f'(a)(x-a)+\frac{f''(a)}{2!}(x-a)^2+\cdots\frac{f^{(k)}(a)}{k!}(x-a)^k+R_n(x).$$

- 麦克劳林公式(Maclaurin formula)：a=0

$$f(x)=f(0)+f'(0)(x)+\frac{f''(0)}{2!}(x)^2+\cdots\frac{f^{(k)}(0)}{k!}(x)^k+o[(x)^n].$$

偏导数、方向导数和梯度

- 偏导数(Partial derivative)：关于其中一个变量的导数，而保持其他变量恒定，记做 f'_x 或$\frac{\partial f}{\partial x}$
- 方向导数(Directional derivative)

$$\frac{\partial f}{\partial l} = \frac{\partial f}{\partial x}\cos\varphi + \frac{\partial f}{\partial y}\sin\varphi$$

- 梯度(Gradient)：向量，方向导数最大的方向

$$\nabla f = \mathrm{grad} f = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y})^T$$

求函数$f(x, y, z) = x^3 - xy^2 - z$在点$p(1, 1, 0)$处的梯度

numpy中的求导和梯度运算

```
from sympy import *
import numpy as np

x, y = symbols('x y')
f = 1 / (1 + x ** 2 + y ** 2)
diff(f, x)

a = np.array([ 2,  8, 15, 13, 15])
np.gradient(a)

b = np.array([[1, 2, 6], [3, 4, 5]])
np.gradient(b)
```

梯度应用于图像处理

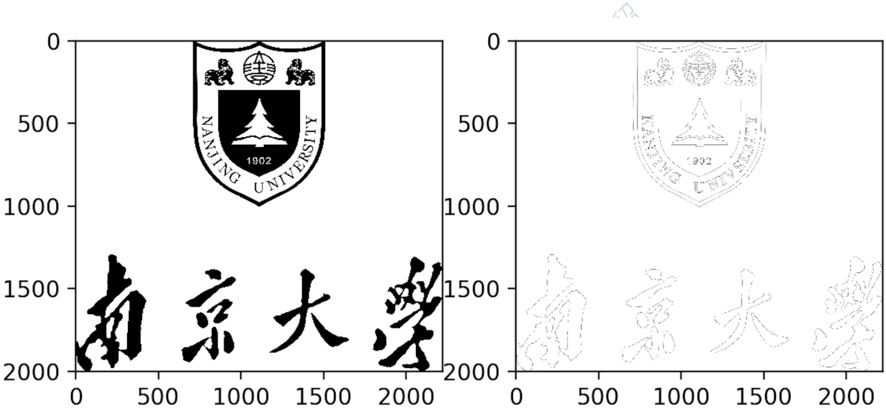
```python
import numpy as np
import matplotlib.pyplot as plt
from skimage import color, util
from scipy import signal

def norm(x, axis=0):
    return np.sqrt(np.sum(np.square(x), axis=axis))
```

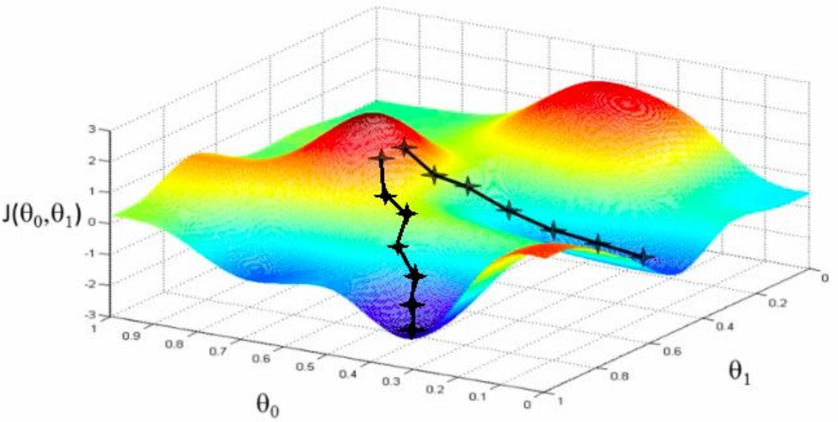
梯度应用于图像处理

```
if __name__ == '__main__':
    img = plt.imread('./logo3.png')
    img = color.rgb2gray(img)
    util.random_noise(img, mode='gaussian')
    img = img - np.mean(img)
    img_grad = np.gradient(img)
    img_grad = norm(img_grad)
    img_grad = util.invert(img_grad)
    fig = plt.figure()
    ax = fig.add_subplot(1, 2, 1)
    ax.imshow(img, 'gray')
    ax = fig.add_subplot(122)
    ax.imshow(img_grad, 'gray')
    plt.show()
```

梯度应用于图像处理


0
500
1000
1500
2000
0
500
1000
1500
2000
NANJING UNIVERSITY
1902
0
500
1000
1500
2000
0
500
1000
1500
2000

梯度示意

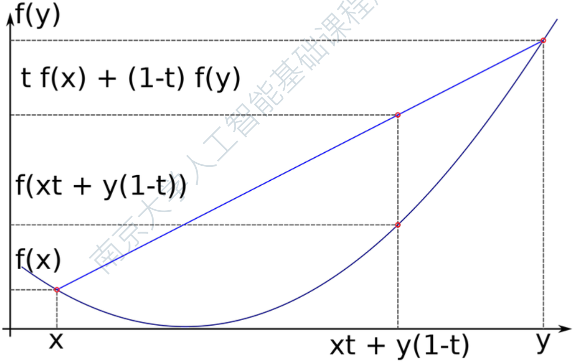


$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1

凸函数

$$\forall x_1, x_2 \in X, \forall t \in [0,1]$$

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$



目录

为什么要使用概率

- 机器学习通常必须处理不确定量，有时可能需要处理随机量
- 不确定性来源：
  - 被建模系统内在的随机性
  - 不完全观测
  - 不完全建模

随机变量

- 随机变量(random variable)是可以随机地取不同值的变量。通常用无格式字体中的小写字母表示，如x。其可能的取值表示为x_1, x_2
- 对于向量值变量，一般写成x，取值为x
- 随机变量可能是离散的，也可能是连续的。

概率分布(probability distribution)

- 离散型变量和概率质量函数(probability mass function, PMF)
  - 大写的P来表示，如P(x)，P(y)。x $\sim P$(x)
  - x=x的概率用$P(x)$来表示。
  - $\forall x \in \mathrm{x}, 0 \leq P(x) \leq 1$
  - $\sum_{x \in \mathrm{x}} P(x) = 1$
- 假设x是均匀分布(uniform distribution)的，则 $P(\mathrm{x}=x_i)=\frac{1}{k}$
- 伯努利分布 (Bernoulli distribution)、二项分布 (Binomial distribution)、泊松分布 (Poisson distribution) 和几何分布 (geometric distribution) 等。

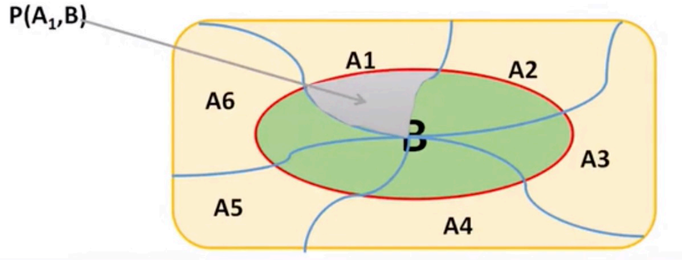
概率分布(probability distribution)

- 连续型变量和概率密度函数(probability density function, PDF), p
  - p的定义域必须是x所有可能状态的集合
  - $\forall x \in \mathrm{x}, p(x) \geq 0$
  - $\int p(x)dx = 1$, x落在区间$[a, b]$的概率是 $\int_{[a,b]} p(x)dx$
- 均匀分布概率密度函数: $u(x; a, b)$, $\mathrm{x} \sim U(a, b)$
  - $\forall x \notin [a, b], u(x; a, b) = 0.$
  - $\forall x \in [a, b], u(x; a, b) = \frac{1}{b-a}.$
- 正态分布 (normal distribution)、指数分布 (exponential distribution) 和β分布 (beta distribution)。

联合概率(joint probability)

- 表示两个或多个事件共同发生的概率，记作

$$P(x,y), P(A,B), P(AB)$$



边缘概率(marginal pd)

- 当我们知道一组变量的联合概率分布时，若我们想知道一个子集的概率分布。那么定义在子集上的概率分布就被我们称为边缘概率分布。

$$\forall \mathrm{x} \in \mathrm{x}, P(\mathrm{x}=x)=\sum_{y} P(\mathrm{x}=x, \mathrm{y}=y)$$

$$p(x)=\int p(x, y) d y$$

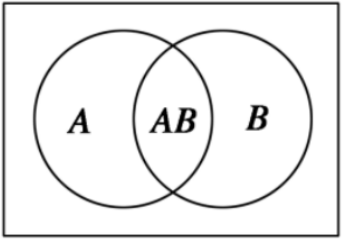
条件概率(conditional probability)

- 事件A在另外一个事件B已经发生的条件下的发生概率

$$P(A|B) = \frac{P(AB)}{P(B)} \qquad P(\mathrm{y} = y|\mathrm{x} = x) = \frac{P(\mathrm{y} = y, \mathrm{x} = x)}{P(\mathrm{x} = x)}$$

- 条件概率链式法则

$$P(\mathrm{x}^{(1)}, \mathrm{x}^{(2)} \cdots \mathrm{x}^{(n)}) = P(\mathrm{x}^{(1)}) \prod_{i=2}^{n} P(\mathrm{x}^{(i)}|\mathrm{x}^{(1)}, \mathrm{x}^{(2)} \cdots \mathrm{x}^{(i-1)})$$



条件概率举例

- 问题：从1,2,……，15中甲，乙依次任取一数（不放回）已知甲取到的数是5的倍数，则甲数大于乙数的概率？
- 假设事件A为“甲取到的数是5的倍数”，B为“甲数大于乙数”，则实际上是求条件概率

$$P(B|A)$$

$$P(B|A) = \frac{P(AB)}{P(A)} = \frac{9}{14}$$

条件概率举例

- 问题：某家庭有3个小孩，已知至少有1个是女孩，求该家庭至少有1个男孩的概率？
- 假设事件A为“3个小孩至少有1个女孩”，B为“3个小孩至少有1个男孩”，则求条件概率

$$P(B|A)$$

$$P(A) = 1 - P(\bar{A}) = 1 - \frac{1}{8} = \frac{7}{8}$$

$$P(AB) = 1 - \frac{1}{8} - \frac{1}{8} = \frac{6}{8}$$

$$P(B|A) = \frac{P(AB)}{P(A)} = \frac{6}{7}$$

条件概率举例

- 问题：袋中有1个白球和1个黑球，现每次从袋中取出1个球，若取出白球，则把白球放回并再加进1个白球，直至取出黑球为止，求取了n次都未取出黑球的概率？
- 假设事件B为“取了n次都未取出黑球”，A_i为“第i次取出白球”，则有

$$B = A_1 A_2 \cdots A_n$$

- 可以使用条件概率链式法则求解

$$\begin{aligned} P(B) &= P(A_1 A_2 \cdots A_n) \\ &= P(A_1)P(A_2|A_1)P(A_3|A_1A_2)\cdots P(A_n|A_1A_2\cdots A_{n-1}) \\ &= \frac{1}{2}\cdot\frac{2}{3}\cdot\frac{3}{4}\cdot\cdots\cdot\frac{n}{n+1} \\ &= \frac{1}{n} \end{aligned}$$

全概率公式

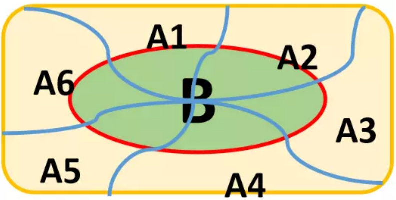
- 样本空间 Ω 有一组事件$A_1, A_2, \cdots, A_n$

$$\forall i \neq j \in \{1, 2, \cdots, n\}, A_i \cap A_j = \phi$$

$$A_1 \cup A_2 \cup \cdots \cup A_n = \Omega$$

- 对于任意事件B的全概率公式为：

$$P(B) = \sum_{i=1}^{n} P(A_i)P(B|A_i)$$



全概率举例

- 某射击小组共有20名射手，其中一级射手4人，二级射手8人，三级射手7人，四级射手1人．一、二、三、四级射手能通过选拔进入决赛的概率分别是0.9，0.7，0.5，0.2，求在小组内任选一名射手，该射手能通过选拔进入决赛的概率?

$$P(A_1)=\frac{4}{20}, P(A_2)=\frac{8}{20}, P(A_3)=\frac{7}{20}, P(A_4)=\frac{1}{20}$$

$$P(S|A_1)=0.9, P(S|A_2)=0.7, P(S|A_3)=0.5, P(S|A_3)=0.2$$

$$P(S)=P(A_1)P(S|A_1)+P(A_2)P(S|A_2)+P(A_3)P(S|A_3)+P(A_4)P(S|A_4)=0.645$$

独立性

- 两个随机变量 x和y，如果它们的概率分布可以表示成两个因子的乘积形式，并且一个因子只包含x另一个因子只包含y，我们就称这两个随机变量是相互独立的。x⊥y

$$\forall x \in \mathrm{x}, y \in \mathrm{y}, p(\mathrm{x}=x, \mathrm{y}=y) = p(\mathrm{x}=x)p(\mathrm{y}=y)$$

- 如果x和y的条件概率分布对于z的每一个值都可以写乘积的形式，那么x和y在给定随机变量z时是条件独立的。x⊥y|z

$$\forall x \in \mathrm{x}, y \in \mathrm{y}, z \in \mathrm{z}, p(\mathrm{x}=x, \mathrm{y}=y \mid \mathrm{z}=z) = p(\mathrm{x}=x \mid \mathrm{z}=z)p(\mathrm{y}=y \mid \mathrm{z} =$$

期望(expectation)

- 函数$f(x)$关于某分布$P(x)$的期望是指：当x由P产生，f作用于x时，$f(x)$的平均值。可以简化写成$\mathbb{E}_x[f(x)]$或$\mathbb{E}[f(x)]$
  - 对于离散型随机变量，可以通过求和得到

$$\mathbb{E}_{x\sim P}[f(x)] = \sum_x P(x)f(x)$$

  - 对于连续型随机变量，可以通过积分得到

$$\mathbb{E}_{x\sim p}[f(x)] = \int p(x)f(x)dx$$

- 期望是线性的，例如

$$\mathbb{E}_x[\alpha f(x) + \beta g(x)] = \alpha \mathbb{E}_x[f(x)] + \beta \mathbb{E}_x[g(x)]$$

用期望表示的方差和协方差

- 我们对x依据它的概率分布进行采样时，随机变量x的函数值会呈现多大的差异(variance)

$$Var(f(x)) = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2]$$

- 方差的平方根即为标准差(standard deviation)
- 协方差(covariance)给出了两个变量线性相关性的强度以及这些变量的尺度

$$Cov(f(x), g(y)) = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])(g(y) - \mathbb{E}[g(y)])]$$

伯努利分布(bernouli)

- 0-1分布，二点分布，单个而只随机变量的分布，$\phi \in [0,1]$为随机变量值为1的概率。

$$P(\mathrm{x}=1)=\phi$$

$$P(\mathrm{x}=0)=1-\phi$$

$$P(\mathrm{x}=x)=\phi^{x}(1-\phi)^{1-x}$$

$$\mathbb{E}_{\mathrm{x}}[\mathrm{x}]=\phi$$

$$Var_{\mathrm{x}}(\mathrm{x})=\phi(1-\phi)$$

二项分布(Binomial)

- n次独立的伯努利分布(n=1)

$$p(k; n, \phi) = P(x = k) = C(n, k)\phi^k(1-\phi)^{n-k}$$

$$\mathbb{E}_x[x] = n\phi$$

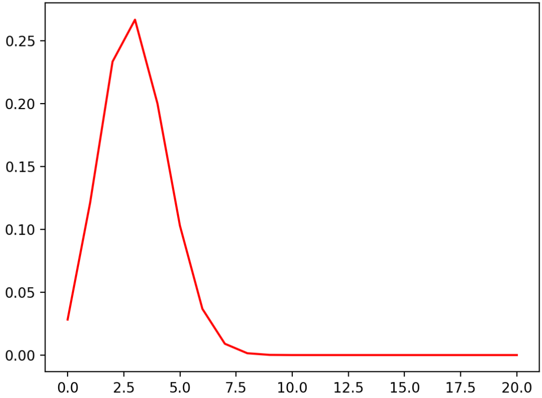
$$Var_x(x) = n\phi(1-\phi)$$

- 例如：掷骰子10次，掷得4的次数服从n=10，ϕ=1/6的二项分布
- 计算扔硬币10次，2次为正面的概率

```
from scipy.stats import binom

k = np.arange(0, 21)
binomial = binom.pmf(k, n, phi)
plt.plot(k, binomial, 'r-')
plt.show()
```

二项分布



泊松分布(Poisson Distribution)

- λ称为比率参数(rate parameter)，单位时间内发生该事件的次数

$$P(x=k)=\frac{\lambda^k e^{-\lambda}}{k!}$$

$$\mathbb{E}_{(}x)=\lambda$$

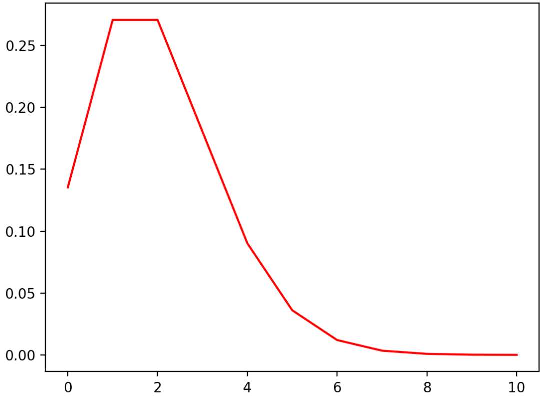
$$Var_x(x)=\lambda$$

- 已知某路口发生事故的比率是每天2次，那么在此处一天内发生4次事故的概率是多少?

```
from scipy.stats import poisson

lambd = 2
n = np.arange(0, 11)
poisson = poisson.pmf(n, lambd)
plt.plot(n, poisson, 'r-')
plt.show()
```

泊松分布



高斯分布(Gaussian/normal)

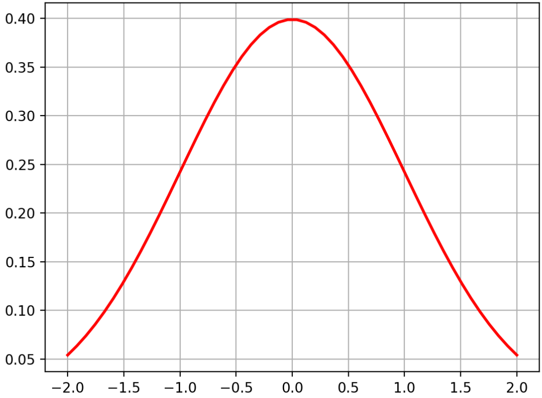
- 又称为正态分布

$$\mathcal{N}(x;\mu,\sigma^2)=\sqrt{\left(\frac{1}{2\pi\sigma^2}\right)}exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

$$\mu\in\mathbb{R},\sigma\in(0,\infty)$$

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
u = 0
sig = np.sqrt(1)
x = np.linspace(u - 2*sig, u + 2*sig, 50)
y = mlab.normpdf(x, 0, 1)
plt.plot(x, y, "r-", linewidth=2)
plt.grid(True)
plt.show()
```

高斯分布(Gaussian/normal)



指数分布(exponential distribution)

- λ成为率参数(rate parameter)，单位时间内发生该事件的次数

$$p(x;\lambda) = \lambda \mathbf{1}_{x\geq 0} exp(-\lambda x)$$

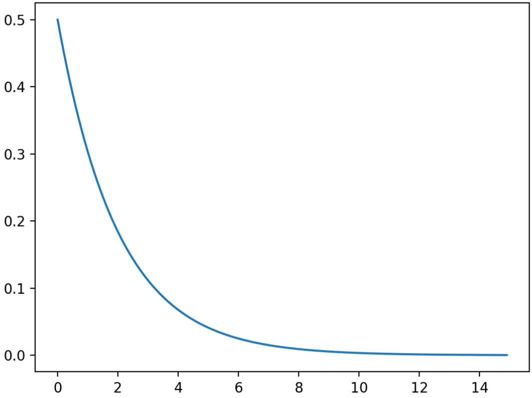
$$\mathbb{E}_{(}\mathrm{x}) = \frac{1}{\lambda}$$

$$Var_{\mathrm{x}}(\mathrm{x}) = \frac{1}{\lambda^2}$$

- 如果你平均每个小时接到2次电话，那么你预期等待每一次电话的时间是半个小时。

```
lambd = 0.5
x = np.arange(0, 15, 0.1)
y = lambd * np.exp(-lambd * x)
plt.plot(x, y)
plt.show()
```

指数分布



β分布

- λ称为率参数(rate parameter)，单位时间内发生该事件的次数

$$p(x;\alpha,\beta)=\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1}$$

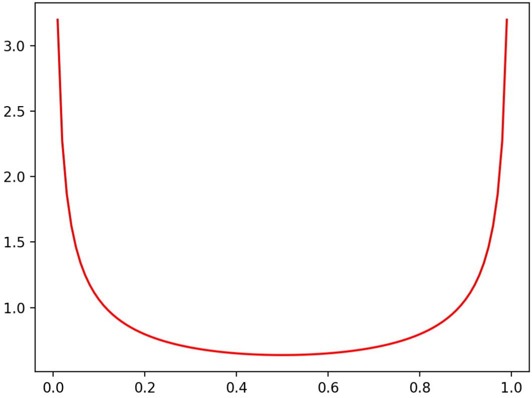
$$\mathbb{E}_{(}x)=\frac{\alpha}{\alpha+\beta}$$

$$Var_x(x)=\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

- 当α和β都为1时，即为均匀分布(uniform distribution)

```
from scipy.stats import beta
a = 0.5
b = 0.5
x = np.arange(0.01, 1, 0.01)
y = beta.pdf(x, a, b)
plt.plot(x, y, 'r-')
plt.show()
```

β分布



贝叶斯规则

- 已知$P(\mathrm{y}|\mathrm{x})$时计算$P(\mathrm{x}|\mathrm{y})$

$$P(\mathrm{x}|\mathrm{y}) = \frac{P(\mathrm{x})P(\mathrm{y}|\mathrm{x})}{P(\mathrm{y})}$$

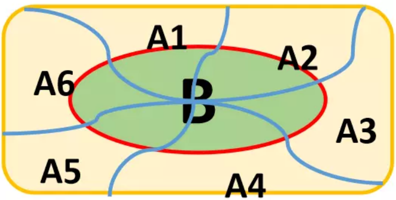
$$P(\mathrm{y}) = \sum_{x} P(\mathrm{y}|\mathrm{x})P(\mathrm{x})$$

- 以Thomas Bayes名字命名
- $P(\mathrm{x})$： x的先验概率(prior)，或标准化常量，边缘概率，全概率
- $P(\mathrm{x}|\mathrm{y})$： x的条件概率或后验概率(posterior)
- $P(\mathrm{y}|\mathrm{x})$： x的似然度(likelihood)，y的条件概率或后验概率
- $P(\mathrm{y})$： y的先验概率，或标准化常量，边缘概率，全概率

贝叶斯规则

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$$P(A_i|B) = \frac{P(B, A_i)}{P(B)} = \frac{P(A_i) \cdot P(B|A_i)}{\sum_{j=1}^{n} P(A_j) \cdot P(B|A_j)}$$



贝叶斯规则应用

- 碗1中有30个香草曲奇饼干和10个巧克力饼干，碗2中有上述饼干各20个。闭上眼随机拿一块，从碗1中拿到香草曲奇的概率是多少?
- 碗1： x， 香草： y
  - $P(x)$： 先验概率(prior)，1/2
  - $P(x|y)$： 后验概率(posterior)，待求
  - $P(y|x)$： 似然度(likelihood)，3/4
  - $P(y)$： 标准化常量，5/8

贝叶斯规则应用

- 一座别墅在过去的20年里一共发生过2次被盗，别墅的主人有一条狗，狗平均每周晚上叫3次，在盗贼入侵时狗叫的概率被估计为0.9，问题是：在狗叫的时候发生入侵的概率是多少?
- 狗在晚上叫： x，盗贼入侵： y
  - $P(x)$： x的先验概率(prior)，3/7
  - $P(x|y)$： x的后验概率(posterior)，0.9
  - $P(y|x)$： y的后验概率，待求
  - $P(y)$： y的先验概率，2/(20*365)

贝叶斯规则应用

- 用某种机器学习方法检查判定肺癌，设A为“用此方法判断被检查者患有肺癌”，B为“被检查者确实患有肺癌”，已知：P(A|B) = 0.95，P($\overline{A}|\overline{B}$) = 0.90，P(B) = 0.0004，如果某人通过该方法被检查患有肺癌，求此人真正换肺癌的概率？

$$
\begin{aligned}
P(B|A) &= \frac{P(A|B)P(B)}{P(A)} \\
&= \frac{P(A|B)P(B)}{P(A|B)P(B) + P(A|\overline{B})P(\overline{B})} \\
&= \frac{0.0004 * 0.95}{0.0004 * 0.95 + 0.9996 * 0.10} \\
&= 0.0038
\end{aligned}
$$

目录

信息与熵

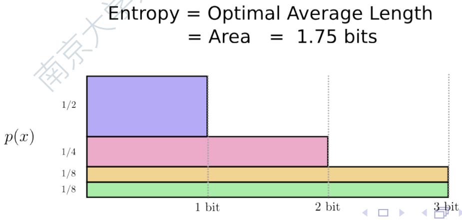
- 自信息量(self-information)，一个事件不确定性的度量，事件发生的概率越大，信息量越少。

$$I(x) = -\log(P(x))$$

- 熵(entropy)，自信息量的期望，越随机，熵越大，系统越混乱，不确定性越大。

$$H(X) = E[-\log P_i] = -\sum P_i \log P_i$$

$$H(X) = -\int p(x)\log p(x)\,dx$$



联合熵、条件熵

- 联合熵(joint entropy)

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

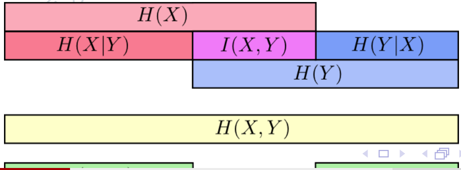
$$H(X)+H(Y) \geq H(X,Y) \geq max[H(X),H(Y)]$$

- 条件熵(conditional entropy)，自信息量的期望，越随机，熵越大，系统越混乱，不确定性越大。

$$H(X|Y) = H(X,Y) - H(Y)$$

- 互信息(mutual information)

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



交叉熵、KL散度

- 交叉熵(cross entropy)，用非真实分布Q来表示来自真实分布P的平均编码长度

$$H(P,Q) = -\sum_{x} P(x)\log(Q(x,y))$$

- 交叉熵在机器学习中经常用作损失函数，用来衡量真实标记分布于模型预测分布的相似度。
- 交叉熵作为损失函数的好处是使用sigmoid函数在梯度下降时能避免均方误差损失函数学习速率降低的问题，因为学习速率可以被输出的误差所控制。
- KL散度(KL divergence)，相对熵

$$KL(P||Q) = D(P||Q) = \sum_{x}(P(x)\log\frac{P(x)}{Q(x)}) = H(P,Q) - H(P)$$

交叉熵、KL散度

- 机器学习的目标是希望在训练数据上学到的模型分布(Q)和真实数据分布(P_{real})越接近越好

$$Q \simeq P_{real}$$

- 真实数据分布未知，因此假设训练数据是从真实数据中独立同分布采样而来

$$P_{traing} \simeq P_{real}$$

- 退而求其次，我们希望学到的模型分布至少和训练数据的分布一致

$$Q \simeq P_{traing}$$

- 因此目标是最小化 KL散度

$$KL(P_{traing}||Q) = H(P_{traing}, Q) - H(P_{traing})$$

- 当$H(P_{traing})$固定不变，那么就相当于最小化交叉熵 $H(P_{traing}, Q)$

Thank you

Thank You!