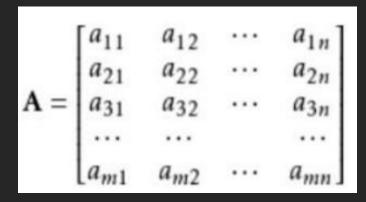
矩阵

定义 代码实现

定义

A = m * n

元素: a (ij) 数字符号或者数学式



运算

1.加减: 行列数相同 A+B A-B

2.乘法: A * B A (m行n列) B (n行m列)

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ a_{31} & a_{32} & \cdots & a_{3n} \\ \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

用途

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ a_{31} & a_{32} & \cdots & a_{3n} \\ \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

代码实现

```
import numpy as np
m, n = map(int, input(). split())
a = np. ones((m, n))
a[1][1] = 9
print(a)

5 6
[[1. 1. 1. 1. 1. 1. ]
[1. 9. 1. 1. 1. 1.]
[1. 1. 1. 1. 1. 1.]
[1. 1. 1. 1. 1.]
[1. 1. 1. 1. 1.]
```

4-6、最大熵模型

最大熵原理是概率模型学习的一个准则,它认为: 学习概率模型时,在所有可能的概率分布中,熵最大的模型是最好的模型。通常用约束条件来确定模型的集合,所以, 最大熵模型原理也可以表述为: 在满足约束条件的模型集合中选取熵最大的模型。

前面我们知道,若随机变量X的概率分布是 $P(x_i)$,则其熵定义如下:

$$H\left(X\right) = -\sum_{i=1}^{n} P\left(x_{i}\right) log P\left(x_{i}\right) = \sum_{i=1}^{n} P\left(x_{i}\right) \frac{1}{log P\left(x_{i}\right)}$$

熵满足下列不等式:

$$0 \leq H\left(X\right) \leq \log |X|$$

式中, |X|是X的取值个数, 当且仅当X的分布是均匀分布时右边的等号成立。也就是 说, 当X服从均匀分布时, 熵最大。

直观地看,最大熵原理认为:要选择概率模型,首先必须满足已有的事实,即约束条件;在没有更多信息的情况下,那些不确定的部分都是"等可能的"。**最大熵原理通过熵的最大化来表示等可能性;"等可能"不易操作,而熵则是一个可优化的指标**。

最大熵模型

