**深度学习**

V. 1.0.1

备注：本文档大量常考《Deep Learning》（[美] Ian Goodfellow ,[加] Yoshua Bengio,[加] Aaron Courvile 著），

《深度学习入门-基于Python的理论与实现》（[日] 斋藤康毅 著），请勿用做商业。

注：如涉及侵权，务必联系 [leep233@foxmail.com](mailto:leep233@foxmail.com) . 本人将立即整改；

目的：本文档只用作记录，分析以及学习，并无任何商业目的

目录

[线性代数 2](#_Toc106870144)

[概念 2](#_Toc106870145)

[导数与偏导数 2](#_Toc106870146)

[积分 7](#_Toc106870147)

[向量与矩阵 8](#_Toc106870148)

[范数 8](#_Toc106870149)

[运算 9](#_Toc106870150)

[特殊向量与矩阵 10](#_Toc106870151)

[特征分解 11](#_Toc106870152)

[奇异值分解 12](#_Toc106870153)

[概率 12](#_Toc106870154)

[随机变量 12](#_Toc106870155)

[随机变量与概率分布 12](#_Toc106870156)

[边缘概率 13](#_Toc106870157)

[条件概率 13](#_Toc106870158)

[独立性与条件独立 14](#_Toc106870159)

[期望(EXPECTATION) 14](#_Toc106870160)

[方差(variance) 14](#_Toc106870161)

[协方差（covariance） 15](#_Toc106870162)

[常用概率分布 17](#_Toc106870163)

[机器学习 19](#_Toc106870164)

# 线性代数

## 概念

**实数（real number）**

**有理数和无理数的总称。用表示实数集。**

**标量（scalar）**

**表示一个具体实数。使用小写斜体来表示（如：标量 ）**

**向量（vector）**

**表示一列数。使用小写加粗来表示（如：向量 ）**

注意: 假设有一个向量 x , 这个向量里面有 *n* 个元素，并且每个元素属于实数，那么我们可以记作 ; **其中向量里面的元素使用标量+下角标的形式表示**

**矩阵（matrix）**

**矩阵是一个二维数组。**

**使用加粗大写字母表示（如：假设矩阵 ，是一个m 行 n 列 的矩阵 我们可以记作 ；如果我们需要表示A 中 第 i 行的 j 列 的元素 我们可以记作 ；）**

注意: 假设矩阵 A ，是一个m 行 n 列 的矩阵 我们可以记作 ；

如果我们需要表示A 中 第 i 行的 j 列 的元素 我们可以记作 ；

表示第i 行的所有列;

表示第 j 列的所有行;

## 导数与偏导数

导数（derivative）

**如下图所示，小明开车去旅游，图中是某段事件内，车速s 和 时间 t 对应的函数关系图。求t1到t2 时间里，小明开车的平均速度？**

**S**

**t**



t1

t2

s1

s2

**我们很简单可以得出小明t1~t2之间的平均速度为 ;**

**如果我们将上图中，s 和 t ，使用函数关系坐标系来表示 s = f(x) , t = x; 那么函数关系图如下：**

**x**



**通过上图我们用来代替轴，那么，之间的**平均变化率, 记作 ;

**通过上面求平均变化率的例子，当 无线趋近 时，即 ) , 那么求出来的就是**导数**，记作**， 或 ;

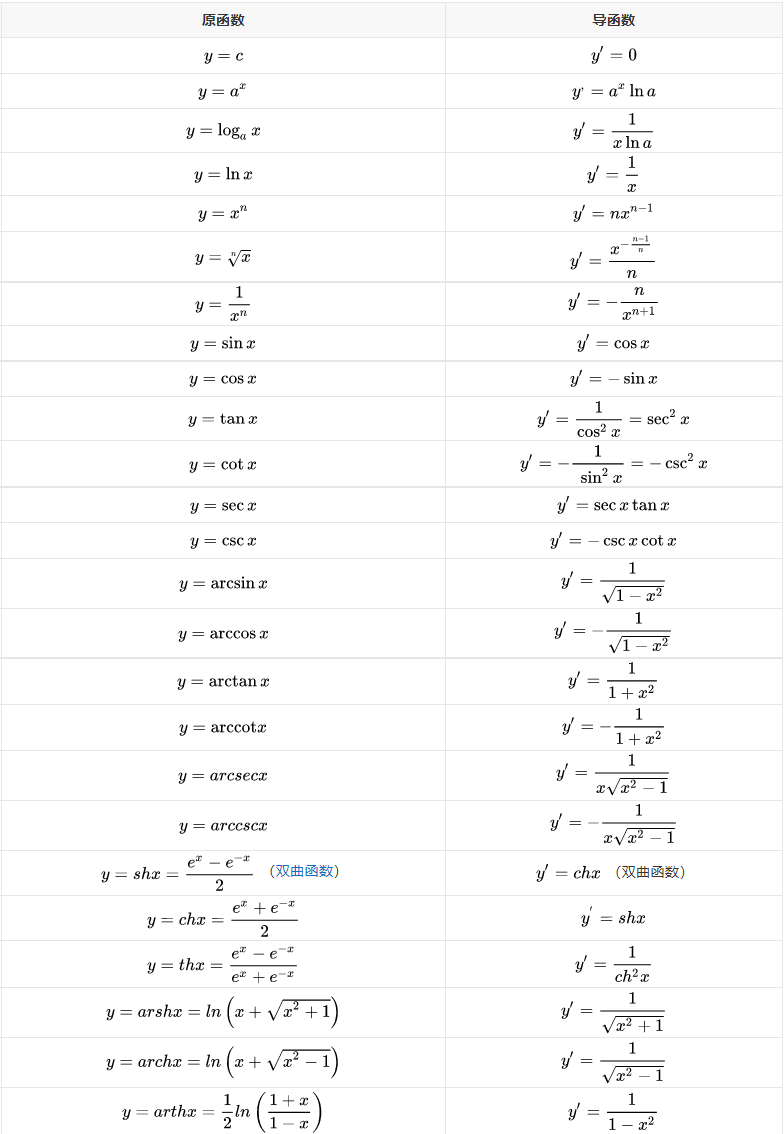
**导数的几何意义:在几何数学中，导数即时该点的斜率，同时我们可以通过点斜式来求得关于过目标点的切线方程。**

**点斜式：已知目标点（ , ）,斜率 *k*, 通过点斜式可以求出目标点的切线方程，即 ;**

**斜率与函数单调性：当斜率k>0时，表示函数单调递增，k<0时，函数单调递减，当k ≈ 0，时表示到达了函数的极值或鞍点**

* 加法:
* 乘法：
* 除法:
* 链式法则:

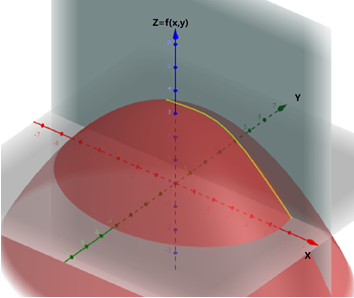
以下是常用的函数的求导公式



偏导数（partial derivative**）**

**当函数的具有多个自变量（输入参数，）求其中一个的导数时，我们称为偏导数**

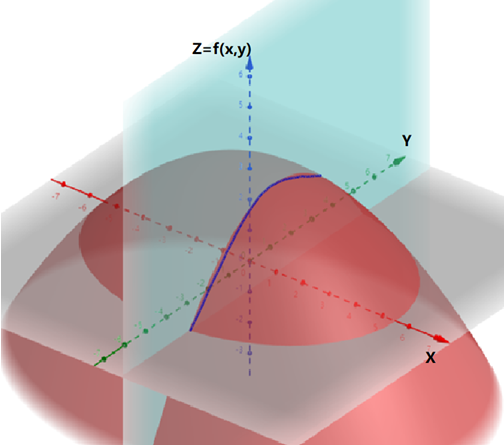
设函数 ,我们求关于函数 的偏导数时，需要将的值固定，的值来求偏导，记作 ,通过坐标系表示如下图



,

由图可以了解到求 关于 的偏导,其实就是求图中 在黄色线中的导数；

同样的如果我们需要求 ，我们只需要固定的值，就可以求出点的导数。如下图

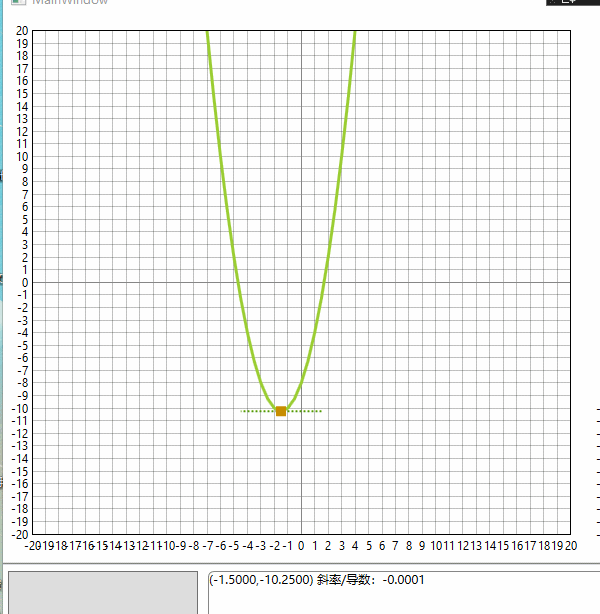


,

梯度下降（Gradient descent）

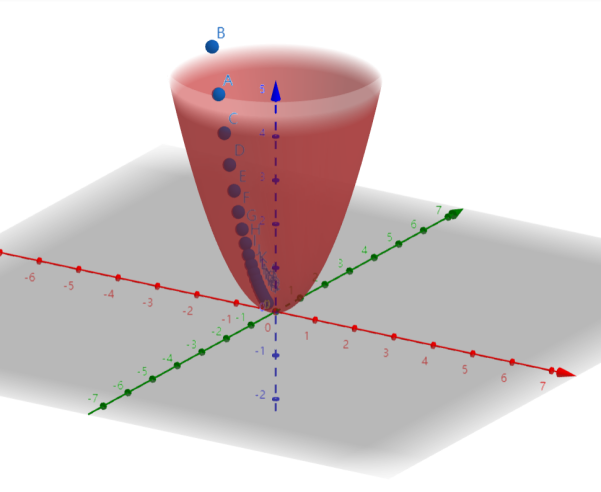
已知函数 ，求该方程的极值。

由上述方程开口向上，有极小值的抛物线，并且导数求方程;

首先我们假设每次我们变动的量，这样的话我们可以求出 处的斜率，通过斜率可以得知当 时，即找到了极值，同时由于斜率可以表示函数单调性的原因,也可以表示 往哪个方向变化才能找到极值，所以我们可以让 , 其中常量 我们也叫做学习率，那么只要我们一直循环n次 之后，当 时，当前点就可以近似认为时函数的极值。我们将这种方法叫做**梯度下降（gradient descent）**；下图为梯度下降的计算的变化图；

以上为2D坐标空间内的梯度下降概念；

假设我们在多维度的空间内，假设函数,同样的我们求出 ，同时我们定义学习率，那么我们可以得到向量,,同样我们循环N次这种操作将会得到一个最低点(极值)；当我们面对的时多维度计算的时候我们判断是否到达极值时,即();下图描述了函数求梯度变化的规律



## 积分

下图时关于函数f(x) 和 x 的函数关系图，求a,b区间内（阴影区域）的面积；



a

b



如图可知，由于我们的线性关系时曲线所以我们没有办法通过寻常求面积的方式求得阴影的面积，那么我们可以使往一个方向移动极小的距离，同时我们可以也可以得出移动 距离后的对应的函数的值，记作，那么我们可以近似这小块的面积等于 ,只要一直将这个区间内所有的小块面积相加就可以得到整个大块阴影的面积，我们记作;

同时如果我们的函数是关于多个自变量求积分则；如,需要求在 区域内的积分记作

## 向量与矩阵

### 范数

在机器学习中，我们使用范数来衡量向量的大小；

**范数:**

**, 其中 ,其实范数是衡量向量x和原点（所有元素都是零的点）之间的距离。我们使用函数来代表范数即，则范数需要满足以下条件**

* **正定性：**
* **三角不等式：**
* **正值齐次性：**

**范数**

**机器学习中，我们需要区分零与非零但是值很小的元素，这种情况下，我们使用各个位置的斜率相同，同时保持简单的数学形式的函数：范数**

**机器学习中，零和非零元素之间的差异非常重要时，通常使用范数。**

**范数（欧几里得范数）**

**当时，称为范数，也叫做欧几里得范数，他表示从原点出发到向量x点的欧几里得距离。由于此范数在机器学习中使用的非常频繁所有我们简写成|| x || , 同时经常用来衡量向量的大小，**

**范数（最大范数）**

**在机器学习中，也经常出现最大范数，它表示向量中具有最大幅值得元素得绝对值；**

**Frobenius范数**

**在机器学习中，当我们需要对矩阵大小进行衡量得时候，我们通常使用Frobenius范数**

### 运算

**转置**

**矩阵的转置即以主对角线（主对角线：从左上角到右下角的对角线）为轴的镜像，即矩阵的行变列，列变行。**

**矩阵的转置为，定义如下：;**

**标量可以看作一个1x1的矩阵，所以标量的转置等于自身 ；**

**向量可以看作是一个列的矩阵，它转置之后变成一个一行的矩阵 ();**

**加减法**

**矩阵与矩阵相加相当于每个对应的元素相加。如;**

**标量与矩阵相加或相乘时，相当于矩阵的每个元素乘以或加上这个标量。**

**如 ;**

**深度都学习中，我们允许矩阵与向量相加，产生一个新的矩阵。如,其中 ; 换言之，将向量b 与 矩阵 A 的每一行相加，这种隐式将向量b 复制到很多位置的方式叫做广播**

**乘积**

**两个想同纬度的向量和的点积（dot product）可以看作。**

**两个矩阵和相乘等到一个新的矩阵。矩阵相乘必须满足左边矩阵的列数和右边矩阵的行数相等。如果矩阵是,矩阵是 ,那么他们相乘等到的矩阵 为 ，即 ，乘法操作定义为：，我们可以看作 的行 和 的列 进行点积。**

* **分配律：**
* **结合律：**
* **交换律（矩阵相乘是不满足交换律的，但是两个向量点积满足交换律）：**

**Hadamard乘积**

**两个矩阵A和B 进行Hadamard乘积时，等于两个矩阵对应行列的元素相乘，等到新矩阵C，记作，操作定义为 ，**

**轨运算**

**迹运算返回矩阵对角线的和，记作 ;**

**迹运算提供了另一种Frobenius的范数方式 : ;**

**迹运算的转置运算下是不变的，即**

**多矩阵的积运算等于将最后一个矩阵移到最前面一个矩阵的位置:或记作**

**标量的迹运算等于标量自己:**

### 特殊向量与矩阵

**单位向量**

**具有单位范数的向量即**

**方阵**

**矩阵的行数和列数相等时，我们称为方阵**

**单位矩阵**

**所有沿主对角线得元素都是1，而且其他位置得元素都是0的方阵叫做单位矩阵。任意向量和单位矩阵相乘都不会进行任何改变。**

**通常我们使用 来表示一个n行n列的单位矩阵。下图为 矩阵：**

**逆矩阵**

**矩阵，是否存在逆矩阵必须满足公式：**

 **假设方程,并且矩阵A时存在逆矩阵则;**

**需要满足上面方程每个都有对应的解，所以要求矩阵*A*必须是方阵：1.方阵一定有逆矩阵;2,必须满足的元素个数和一致**

**对与方阵来说逆矩阵的左乘和右乘时相等的。**

**对角矩阵**

**当一个矩阵主除主对角线上有非零元素，其他位置都是零，称为对角矩阵(diagonal matrix)。我们用表示对角矩阵由向量中元素给定的一个对角方阵。**

**对角矩阵的乘法计算很高效。计算乘法，只需要将向量中的每个元素放大倍数，即; 当且仅当对角方阵对角元素都是非零值时，对角方阵的逆矩阵操作为。**

**如果对角矩阵与矩阵乘积 矩阵为对角矩阵，为三个向量组成的矩阵 如下**

**= ，矩阵左乘对角矩阵详单与对每一行进行对应对角矩阵分量相乘**

**,矩阵右乘对角矩阵详单与对每一列进行对应对角矩阵分量相乘（注意这个特性，特征值分解时将会用到）**

**对称矩阵**

**对称矩阵的转置等于自身:**

**正交矩阵**

**两个向量的乘积为零时，那么向量和向量相互正交，即;如果两个向量都有非零范数说明两个向量之间的夹角90。在**空间中，至多有个范数非零的向量**互相正交**，如果这些向量不但互相正交，而且范数都为1，我们称它们是**标准正交**。

当行向量和列向量是分别标准正交的方阵称为**正交矩阵**，即,同时意味着 ，从上面我们可以得知正交矩阵的求逆计算代价是很小的。



**行列式**

**行列式是将方阵映射到实数的函数。如果行列式是0，那么空间至少沿某一维度完全收缩了，使其失去了所有体积；如果行列式是1，那么这个转换保持空间不变。**

**余子式与代数余子式**

**，我们可以快速的求出矩阵A的行列式 但是当我们的矩阵阶数越高时，我们计算行列式也越麻烦，所以我们如果我们能将高阶数的矩阵转化为低阶进行计算的话就比较简单了，这里我们就引入了余子式和代数余子式的概念；**

**设矩阵 ,那么其中元素 我们将这个元素对应的所有行和列都却除掉,记作 ,其中就是 对应的余子式，那么我们再处的代数余子式则为,用公式记作;**

**这样如果我们通过代数余子式来计算行列式的话，**行列式等于它任意一行(列)的各元素与其对应的代数式余子式乘积之和,即

**伴随矩阵**

**通过方程组的方式对方阵进行求逆的过程时比较复杂的，我们引入伴随矩阵的概念，**

**矩阵A的伴随矩阵记作()为各个元素对应的代数余子式的行列式的值在进行转置**

**即**

**当我们得到方阵A的伴随矩阵，就可以通过公式来求矩阵A的逆矩阵.**

### **qr分解**

一个矩阵， 可以分解成,其中：

* 正交矩阵
* 是上三角矩阵

**Classical Gram-Schmidt**

**对两个线性无关的向量，正交化，如下图所示**

**其中我们需要求出与垂直的**



**我们先求出,**

**上面我们知道如何求得映射的垂直，开始正交化时 我们随机取出矩阵A 中的一个向量**



**Modified Gram-Schmidt**

**CGS**先拿出第一个向量作为基准，接着拿出第二个向量，并通过对照第一个基准找到垂直于第一个基准的第二个基准，并依次类推。因此，在构建第n个基准时，n+1及以后的向量是不可见的。因为在计算的过程中不可避免地引入误差，而后面的向量只有在其被计算的时候才可见，这时误差可能已经积累到一定程度了，所以总的来说其稳定性较差,因此我们对CGS改进一下；

一组线性无关的向量,进行正交化

**householder**

**Householder 核心 公式: 得到矩阵 矩阵；**

**设矩阵 ；其中 使**  ， ，并求出Householder矩阵 ,再取出除去第一行和第一列矩阵 记作 ,重复上一步操作，我们可以得到以下矩阵

**其中**

### 特征分解

方阵 乘以一个向量,另一个向量, 那么一个标量 乘以向量,得到的向量也是**,**即 **;**那么我们称向量是矩阵的**特征向量**， 称为对应这个向量的**特征值**

如果是方阵的特征向量，那么与任何标量相乘，这个结果都是A的特征向量。所以通常我们只考虑单位特征向量。

通过我们知道 特征值和单位特征向量都不是唯一的。

设矩阵A有n个线性无关的特征向量{},同时这些特征向量对应的特征值为{},我们将这些特征向量组合成一个矩阵 ，同样的我们将所有特征值组成一个列向量即其中，这个行为叫做**特征分解**,即将矩阵分解成一组特征向量和特征值。因此矩阵的特征分解记作

注意不是每个矩阵都可以分解成特征值和特征向量。

每个实对称矩阵都可以分解成实特征向量和实特征值**:,**其中是的特征向量组成的正交矩阵，是特征值组成的对角矩阵，通常我们将的元素进行降序排列。

矩阵的特征分解我们知道矩阵式奇异的（列向量线性相关的矩阵），实对称矩阵的特征分解可以用于优化方程**，**其中,当x等于A的某个特征向量时，返回对应的特征值。在限制条件下的最大值时最大特征值，最小值时最小特征值。

* 所有特征值都是正数的矩阵称为**正定**
* 所有特征值都是非负数的矩阵称为**半正定**
* 所有特征值都是负数的矩阵称为**负定**
* 所有特征值都是非正数的矩阵称为**半负定**

### 奇异值分解（singualr value decomposition,svd）

将矩阵分解为**奇异向量**和**奇异值**。

每个实数矩阵都有奇异值分解，但不一定有特征分解。

奇异值分解我们将矩阵A分解称为三个矩阵的乘积:,假设 是 矩阵，那么是矩阵，为,是矩阵,其中每个矩阵都拥有,·和 都是正交矩阵，为对角矩阵。

对角矩阵D对角线上的元素为矩阵A的**奇异值**。矩阵U的列向量为做**左奇异向量**，矩阵V的列向量称为**右奇异向量**。

其中左奇异向量为的特征向量，右奇异向量的特征向量，的非零奇异值是特征值的平方根，也是特征值的平方根。

### Moore-penrose 伪逆

对于非方阵矩阵，其逆矩阵没有定义。

之前我们已经知道如果矩阵有逆矩阵，那么求方程的解，可以分布左乘一个的逆矩阵，得到;但是这个问题在于未必一定就有一个逆矩阵，如果行大于列，可能没有解；如果行小于列可能有无数个解。那么我们能不能找到一个可以代替的矩阵呢（即时），这个问题我们通过Moore-Penrose 伪逆（moore-penrose pseudoinverse）来解决:,但是计算伪逆的实际算法没有基于这个定义，我们使用下面的公式：,其中矩阵是矩阵奇异值分解后得到的矩阵，其中对角矩阵的伪逆是其非零元素倒数之后再转置得到的。

当矩阵A的列数多于行数时，使用伪逆求解线性方程时众多可能解发中的一种。是方程所有可行解欧几里得范数最小的一个;当列小于行数时，可能没有解。这种情况下，通过伪逆得到的使得和的欧几里得最小。

### 主成分分析(Principal components analysis,PCA)

主成分分析的主要作用:

* 主成分分析能降低所研究的[数据空间](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E7%A9%BA%E9%97%B4)的维数
* 有时可通过因子负荷aij的结论，弄清X变量间的某些关系
* 多维数据的一种图形表示方法。我们知道当维数大于3时便不能画出几何图形，[多元统计](https://baike.baidu.com/item/%E5%A4%9A%E5%85%83%E7%BB%9F%E8%AE%A1)研究的问题大都多于3个变量
* 由主成分分析法构造[回归模型](https://baike.baidu.com/item/%E5%9B%9E%E5%BD%92%E6%A8%A1%E5%9E%8B)。即把各主成分作为新自变量代替原来自变量x做[回归分析](https://baike.baidu.com/item/%E5%9B%9E%E5%BD%92%E5%88%86%E6%9E%90" \t "_blank)
* 用主成分分析筛选回归变量

假设我们对空间中有 个点,我们需要对这些数据进行有损压缩(降维)，对于每个点,都有对应的,由于我们需要对数据进行压缩，所以。假设我们定义一个压缩函数,对应的我们需要一个解压函数。

其中解压函数需要将降维的向量映射回向量空间中，我们使用矩阵乘法，定义为由于我们是空间映射回空间中，所以我们可以得知矩阵 ;

目前给到的信息可知 可能有多个解，当时我们需要找到最优解,这样我们可以通过原来对应的求欧几里得范数来得到两个向量的差距(距离)。只要我们求得差距最小的即时对应最优解。

,根据范数公式我们可以求出距离d,我们进行修改一样，对欧几里得范数不进行开根号处理 ;

同样我们将带入公式，即

这里我们使用PCA限制D的列向量彼此正交，所以

这样我们如果需要求

由于向量求导公式 所以

# 概率

概率即用于描述不确定性的声明。

机器学习中通常需要处理不确定量，也可能需要处理随机量

在机器学习中不确定性由3中可能的来源

* 被建模系统内在的随机性
* 不完全观测性
* 不完全建模

### 随机变量

**离散型**

**离散型随机变量拥有有限或者可数无限多的状态。**

**连续型**

**连续型变量拥有不可数实数值**

### **随机变量与概率分布**

**离散型变量和概率质量函数（probability mass function , PMF）**

**对于离散型随机变量的概率分布可以使用概率质量函数来描述，我们通常使用大写字母来表示， 的概率我们使用来表示，当概率=1时，表示事件时必然发生的，概率=0时，事件一定不会发生表示遵循分布。**

**概率质量函数同时可以作用于多个随机变量，我们称为联合概率分布（joint probability distribution）。或,表示 同时发生的概率，也可以写成。**

**如果函数P是随机变量 的PMF，必须满足以下条件**

* **P的定义域必须是的所有可能状态集合**

**连续型变量和概率密度函数（probability density function , PDF）**

**对于连续型随机变量时，我们使用概率密度函数 ，我们时使用p 来表述概率密度函数，必须满足以下条件**

* **的定义域必须是的所有可能状态集合**
* **，其中表示一个极小值**

### **边缘概率**

一组变量的联合概率分布，需要了解其中一个子集的概率分布，这种定义在子集上的概率分布被称为**边缘概率分布（marginal probability distribution），**

已知随机变量和，并且我们知道

* 离散型随机变量边缘概率分布:
* 连续型随机变量边缘概率分布：

### 条件概率

当某个事件在给定其他事件发生时出现的概率我们叫做**条件概率。**

如果事件发生时发生的概率我们记作。此概率的计算公式为：

条件概率的链式法则：

### 独立性与条件独立

两个随机变量x和y，对于的概率分布是完全不相关的我们叫做相互独立事件，如果要计算两个事件同时发生的概率 我们需要 ;

关于x和y的条件概率分布对于z的每一个值都可以写成乘积的形式，我们就称作随机变量x,y 在给定随机变量z是**条件独立**的：

;

### 期望(EXPECTATION)

下表是的分布列

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

当我们 使用 E = 时，我们得出的值E就叫做 关于分布的数学**期望**。

如果我们将x 看作f(x)函数，那么期望值表示为

* 离散型随机变量：
* 连续性随机变量：

假设,那么;即

### 方差(variance)

下面分别为大明，小明进行射击的分布律

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

通过两个分布列可以得知大明的数学期望 为 E() = , 小明为 E() = ，当我们需要对大明和小明的射击水平进行比较我们不能简单的比较期望值的大小，这样我们引入方差的概念，通过比较大明和小明的方差来对比两人的稳定程度，方差越小说明射击水平越稳定，数学公式表示为

* 离散型随机变量：
* 连续性随机变量：

上述为两种随机变量的方差公式，我们进行简化，我们不难发现我们可以把假设 我们使用函数 替代，那么

，

，

我们不难发现我们可以列出另一个分布律

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

同样的 我们使用我们也可以使用期望表示这个分布律,设 ，则上表的分布律可以表示为,所以方差可以表示为 ，我们将带入，我们就得到了简化后的公式

如果随机变量 ,则

在机器学习中，方差一般用来衡量当我们对 依据它的概率分布进行采样时，随机变量的函数值会呈现多大差异。方差越小，的值形成的簇越接近它们的期望值。

方差的平方根也被称为**标准差（standard deviation）**

### 协方差（covariance）

协方差描述两个随机变量线性相关性的强度以及这些变量的尺度；

假设两个随机变量 , 下面是随便列出来的两个随机变量进行随机抽样列表

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1 | 8 | 3 |
| 2 | 3.5 | 1 |
| 3 | 6 | 1.5 |
| 4 | 6.5 | 2.5 |
| 均值 | 6 | 2 |

如果我们需要比较每次出现两个随机变量的相关性我们只需要每次出现的和 的平均值对比，记作 ,那么我们的计算公式则为 ,

通过公式我们可以上表的值为4.75，

目前来看这两个变量是正相关，如果在保持均值不变的情况下再对 进行连次抽样, 分别为（2，4）（10，2），那么我们可以算出 现在的值为-3.25 ，现在又变成负相关了这很明显并不合理，下图是根据上表列数的随机抽样的概率分布律

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 1 | 8 | 3 | 0.12 |
| 2 | 3.5 | 1 | 0.4 |
| 3 | 6 | 1.5 | 0.35 |
| 4 | 6.5 | 2.5 | 0.3 |
| 5 | 2 | 4 | 0.001 |
| 6 | 10 | 2 | 0.02 |
| 均值 | 6 | 2 |  |
| 加权平均 | 5.551637 | 1.72796 |  |

我们将每次随机抽样对应的概率记作,那么我们算出加权平均值()

上面的计算公式可以记作**协方差** , 我们带入上表算出的值为: 1.08573

说明关于两个随机变量的线性关系是正相关的。

同样我们用函数关系来代替随机变量,协方差表示为;

类似的协方差公式，中

那么 ，同时我们将这一部分看作一个整体，有可以使用一个期望公式替换最终得到协方差公式:

注意：当协方差值>0,时表示两个随机变量线性正相关，<0 是负相关，若=0则无关

如果随机向量的**协方差矩阵(covariance matrix)**是一个的矩阵，并且满足,而且协方差矩阵的对角元是方差

### 常用概率分布

**Bernoulli分布**

**Bernoulli分布是单个二值随机变量的分布。单个参数,给出了随机变量等于1的概率。性质如下:**

**Multinoulli分布**

**具有个不同状态的单个离散型随机变量的分布，其中是一个有限值。**

**高斯分布(Gaussian distribution)**

**高斯分布是最常用的分布，同时是也叫做正态分布(normal distribution)。**

**由上图几个正态分布的图像我们可以分析出，正态分布由两个参数控制，。参数给出了中心峰值(对称轴)的坐标，这也是分布的均值 。分布的标准差用表示，方差则是 ；**

**当我们需要对概率密度函数求值时需要对平方并取倒数。当我们经常对不同参数下的概率密度函数求值时，我们使用更高效的参数化分布方式使用参数来控制分布的精度。**

**当我们的时的正态分布称为标准正态分布（standard normal distribution）记作**

**大多数情况下我们默认使用正态分布的原因**

* **很多分布的真实情况是比较接近正态分布的**
* **相同方差的所有可能的概率分布重，正态分布在实数上具有最大不确定性**

**当正态分布被推广到空间中时，我们称为多维正态分布（multivariate normal dstribution）。他的参数是一个正定对称矩阵：**

**指数分布和Laplace分布**

**深度学习中，我们经常需要一个再x=0点处取得边界点的分布，我们可以使用指数分布(exponential distribution):,指数分布用指示函数来使取负值时的概率为零。**

**一个联系紧密的概率分布是Laplace分布(laplace distribution),他允许我们在任意一点处设置概率质量的峰值：**

**Dirac分布和经验分布**

### **常用函数的有用性质**

**Logistic sigmoid函数**

**Softplus 函数**

**常用性质**

# **数值计算**

**Softmax函数**

**下溢:当接近零的数被四舍五入为零的时候发生下溢。**

**上溢:当大量级的数被近似为时发生上溢。**

**这样我们可以使用softmax函数对数值稳定，softmax函数常用于干预预测Multinoulli分布相关联的概率:**

**病变条件**

**基于梯度的优化方式**

**我们把最大化或最小化函数称为目标函数(objective function)或准则(criterion)。当我们对其最小化时，也称它代价函数（cost function）,损失函数(loss function)或误差函数(error function)。**

**通过求导我们知道当 导数将无法提供往哪个方向移动的信息。这个点称为临界点或驻点。**

**Jaconbian矩阵**

**有时候我们需要计算输入和输出都为向量的函数的所有偏导数，包含所有这样的变导数的矩阵被称为Jacobian矩阵。具体说，如果我们有个函数**

**Hesssian矩阵**

**当我们的函数具有多维度输入时，二阶导数也有很多，我们价格这些导数合并成一个矩阵，称作Hessian矩阵。Hessian矩阵定义为。**

**Hessian等价于梯度的Jacobian矩阵。**

**任何二阶偏导数连续点处可交换，也就是互换他们的顺序**

# 机器学习