《机器学习》课程——

机器学习中的数学基础

主讲人： 徐义田教授

学　校：中国农业大学

目 录

- 代数学基础
- 优化理论
- 概率论与数理统计

机器学习(Machine Learning, ML)是一门多领域交叉学科，涉及概率论、统计学、逼近论、凸分析、算法复杂度理论等多门学科。

数学是基石，算法是利器，编程是工具。三者对于机器学习都很重要。机器学习中大量的问题最终都可以归结为一个优化问题，而微积分、概率、线性代数和矩阵是优化的基础。

为什么机器学习中的数学很重要?

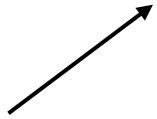
1. 选择合适的算法，要考虑的包括算法准确性、训练时间、模型复杂度等。
2. 选择参数设置和验证策略。
3. 理解偏差与方差的权衡以确定欠拟合和过拟合。
4. 预估正确的置信区间和不确定性。

➢代数学基础

一、基本概念

1. 向量

一列有序排列的数。索引单个元素的位置，可以确定每个数。



物理学

$$\overrightarrow{x}$$

数学

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

计算机

向量的常用运算主要是加法运算、数乘运算、内积运算等。

$$[x_1\ x_2\ x_3] \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = x_1y_1+x_2y_2+x_3y_3$$

内积

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_1+y_1 \\ x_2+y_2 \\ x_3+y_3 \end{bmatrix}$$

加法

$$k \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} kx_1 \\ kx_2 \\ kx_3 \end{bmatrix}$$

数乘

2. 矩阵

矩阵是二维数组，其中的每一个元素被两个索引而非一个所确定。如果一个实数矩阵高度为m，宽度为n，那么我们记作：$A \in R^{m \times n}$

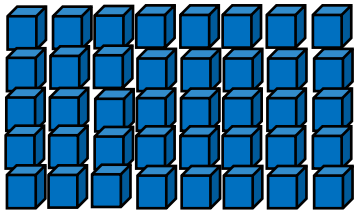
$$A = [a_{ij}]_{mn} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$$

矩阵在机器学习中非常重要！实际上，如果我们现在有m个用户的数据，每条数据含有n个特征，那其实它对应的就是一个$m \times n$的矩阵；再比如，一张图由16×16的像素点组成，那这就是一个16×16的矩阵。类似的，矩阵也有相应的加法、乘法（变换）运算等。

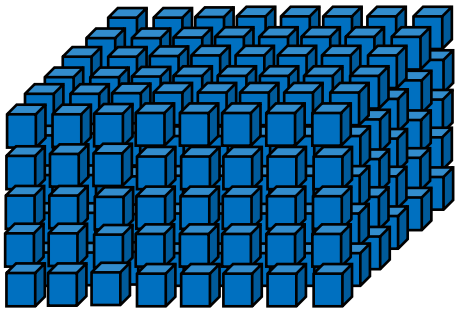
3. 张量

是向量和矩阵的推广，可理解为向量是1阶张量，矩阵为2阶张量。

一阶张量（向量）



二阶张量（矩阵）



三阶张量

一、基本概念

3. 张量

张量在深度学习中是一个很重要的概念，后续的所有运算和优化算法几乎都基于张量进行。例如，可以将任意一张彩色图片表示成一个三阶张量，三个维度分别是图片的高度、宽度和色彩数据。

	0	1	2	3	4	5	6	7	8	9	...	310	311	312	313	314	315	316	317	318	319
0	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	...	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]
1	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	...	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]
2	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	...	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]
3	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	...	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]
4	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	...	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]

二、范数

1. 范数

范数(norm)是数学中的一种基本概念。在泛函分析中，它定义在赋范线性空间中，并满足一定的条件，即：非负性、齐次性、三角不等式。通过范数可以诱导出距离。设$x = (x_1, x_2, \cdots x_n)$，则常用的向量范数表示如下：

$$LP\text{范数：}\ L_p = \sqrt[p]{\sum_{i=1}^{n} {x_i}^p}$$

$p = \infty$　　$p = 2$　　$p = 1$　　$0 < p < 1$　　$p = 0$

二、范数

1. 范数

$\|x\|_0$: 向量中非零元素的个数;

$\|x\|_1 = |x_1| + |x_2| + \cdots + |x_n|$;

$\|x\|_2 = \sqrt{(x_1{}^2 + x_2{}^2 + \cdots + x_n{}^2)}$;

稀疏化(特征选择，可解释)

防止过拟合

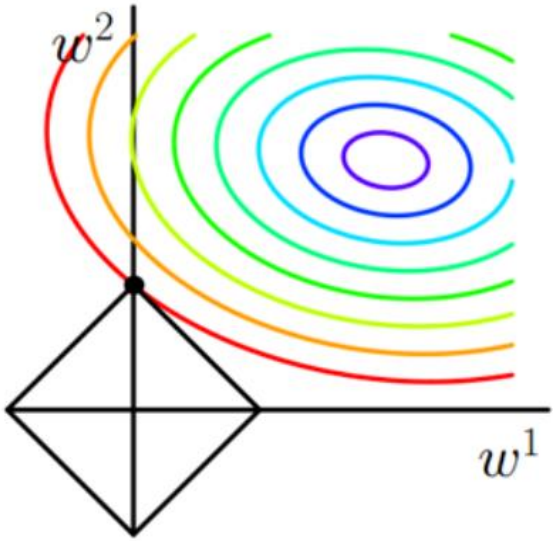
$\|x\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{\frac{1}{p}}$;

$\|x\|_\infty = max(|x_1|, |x_2|, \cdots, |x_n|)$.

二、范数

- Lasso

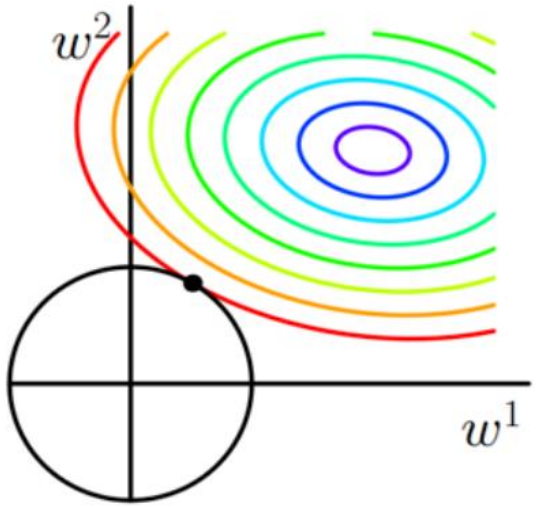
$$\min_{W} \ \|Y - Xw\|^2$$
$$s.t. \ \|w\|_1 \leq C$$



1范数球与二次函数相遇

- Ridge

$$\min_{W} \|Y - Xw\|^2$$
$$s.t. \|w\|_2 \leq C$$



2范数球与二次函数相遇

1范数会趋向于让许多特征都是0，而2范数会保留更多的特征。所以，Lasso在特征选择时候非常有用，而Ridge就只是体现了正则化。

2. 矩阵范数

设$A=[a_{ij}]_{mn}=\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$，则常见的矩阵范数如下：

$$\|A\|_1=max\left\{\sum_{i=1}^{m}|a_{i1}|,\sum_{i=1}^{m}|a_{i2}|,\cdots,\sum_{i=1}^{m}|a_{in}|\right\}$$

$\|A\|_2=A$的最大奇异值

$$\|A\|_F=\left(\sum_{i=1}^{m}\sum_{j=1}^{n}\left(a_{ij}{}^2\right)\right)^{\frac{1}{2}}$$

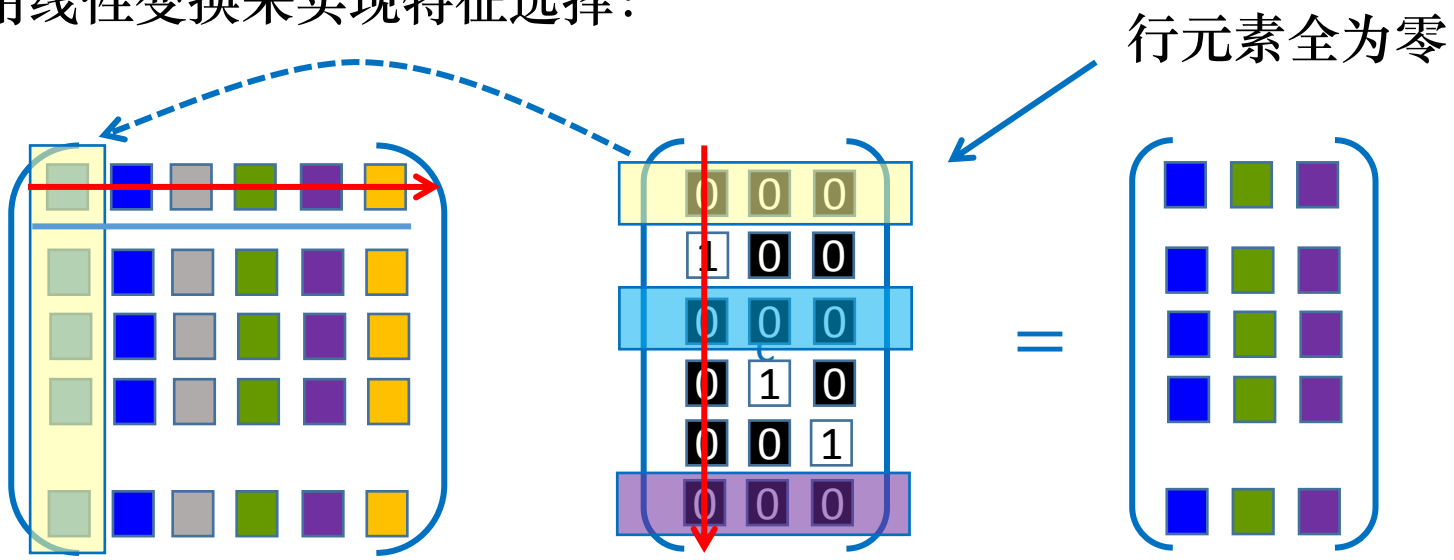
$$\|A\|_\infty=max\left\{\sum_{j=1}^{n}|a_{1j}|,\sum_{j=1}^{n}|a_{2j}|,\cdots,\sum_{j=1}^{n}|a_{mj}|\right\}$$

$$\|A\|_{2,1}=\sum_{i=1}^{m}\left(\sum_{j=1}^{n}\left(a_{ij}\right)^2\right)^{\frac{1}{2}}$$

结构化稀疏

二、范数

采用线性变换来实现特征选择：



$X \in R^{n\times d}$

$W \in R^{d\times m}$

$X_S \in R^{n\times m}$

一行一个数据点

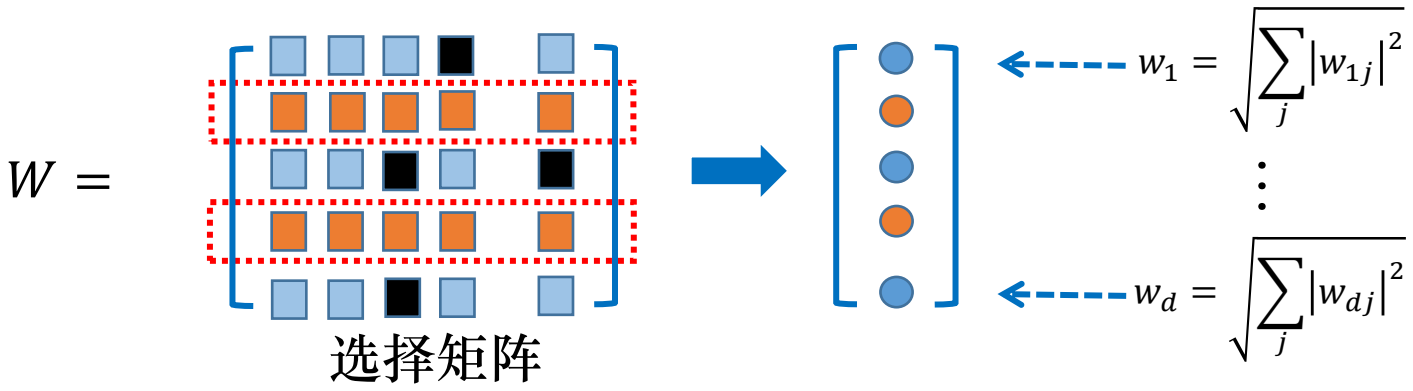
0-1列选择矩阵

$XW = X_S$

若列选择矩阵第一行全为零，则第一个特征分量不起作用！

二、范数

矩阵行稀疏性度量：结构化稀疏



由于选择矩阵只含0、1，因此矩阵的某一行的0范数直接等价于矩阵的1范数

要求W的某行为零，只需要该行元素平方和为零。因此可以将行平方和开根号收集为一个向量，再考虑其零范数

零范数为NP难问题，所以将其转化为1范数，可得：

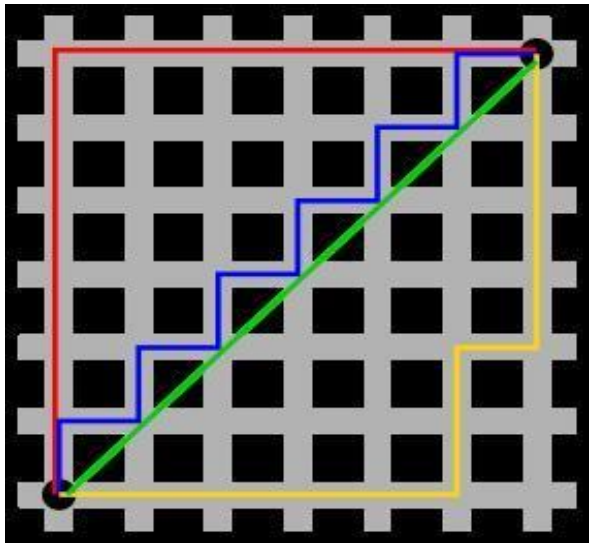
$$\|W\|_{2,1} = \|w\|_1 = \sum_{i=1}^{d} \sqrt{\sum_{j=1}^{n} |W_{i,j}|^2}$$

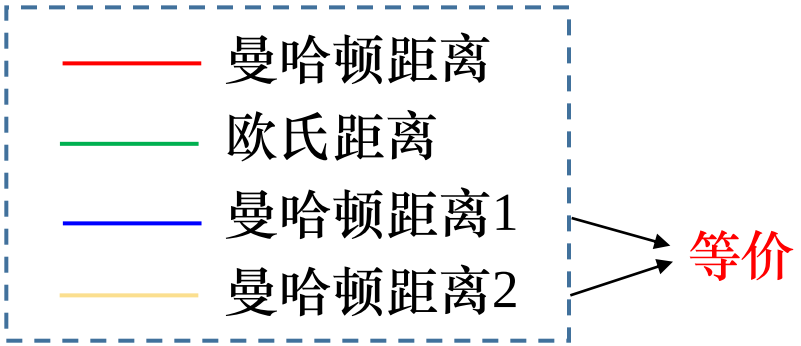
三、距离

在机器学习里运算一般都是基于向量的，一条用户具有100个特征，那么他对应的就是一个100维的向量，通过计算两个用户对应向量之间的距离值大小，有时候能反映出这两个用户的相似程度。这在后面的KNN算法和K-means算法中很明显。设有两个n维变量：$x=(x_1,x_2,\cdots,x_n)$和$y=(y_1,y_2,\cdots,y_n)$，则常见的距离公式如下：

闵可夫斯基距离：$d_{xy}=\sqrt[p]{\sum_{i=1}^{n}(x_i-y_i)^p}$

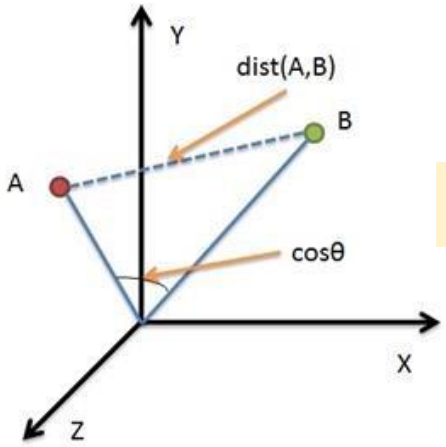
曼哈顿距离：$d_{xy}=\sum_{i=1}|x_i-y_i|$





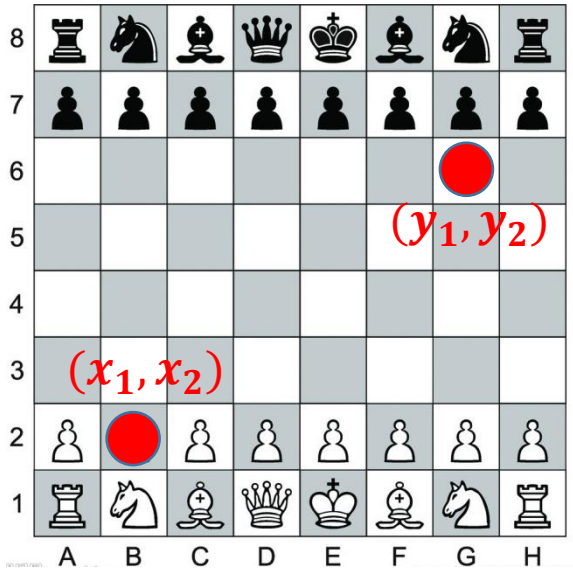
三、距离

欧氏距离：$d_{xy}=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$



余弦距离

切比雪夫距离：$d_{xy}=max(|x_1-y_1|,|x_2-y_2|,\cdots,|x_n-y_n|)$



国王走一步可以移动到相邻8个方格中的任意一个。从格子(x_1,x_2)走到格子(y_1,y_2)最少需要多少步这就是切比雪夫距离。

三、距离

但是闵氏距离、曼哈顿距离、欧氏距离和切比雪夫距离都存在明显的缺点。

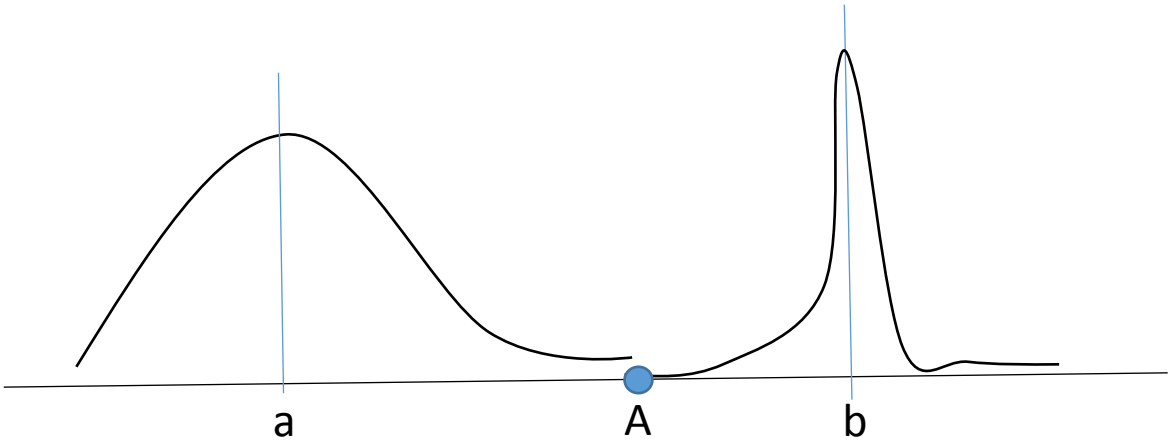
例如： 二维样本(身高[单位:cm],体重[单位:kg]),现有三个样本：a(180,50)，b(190,50)，c(180,60)。那么a与b的闵氏距离（或者曼哈顿距离、欧氏距离、切比雪夫距离）等于a与c的闵氏距离（相应距离）。但实际上身高的10cm并不能和体重的10kg划等号。

闵氏距离的缺点:

- 将各个分量的量纲(scale)，也就是“单位”相同的看待了;
- 未考虑各个分量的分布（期望，方差等）可能是不同的。

三、距离

1. 马氏距离：



直观解释：上图有两个正态分布的总体，它们的均值分别为 a 和 b，但方差不一样，则图中的A点离哪个总体更近？或者说A有更大的概率属于谁？

显然，A离左边的更近，A属于左边总体的概率更大，但是A与a的欧式距离却远一些。这就是马氏距离的直观解释。

三、距离

2. 定义：

有m个样本向量$X=(x_1,x_2,\cdots,x_m)$，协方差矩阵为S，均值记为向量μ，则其中样本向量X到μ的马氏距离表示为：

$$D(X)=\sqrt{(X-\mu)^T S^{-1}(X-\mu)}$$

向量x_i与x_j之间的马氏距离定义为：

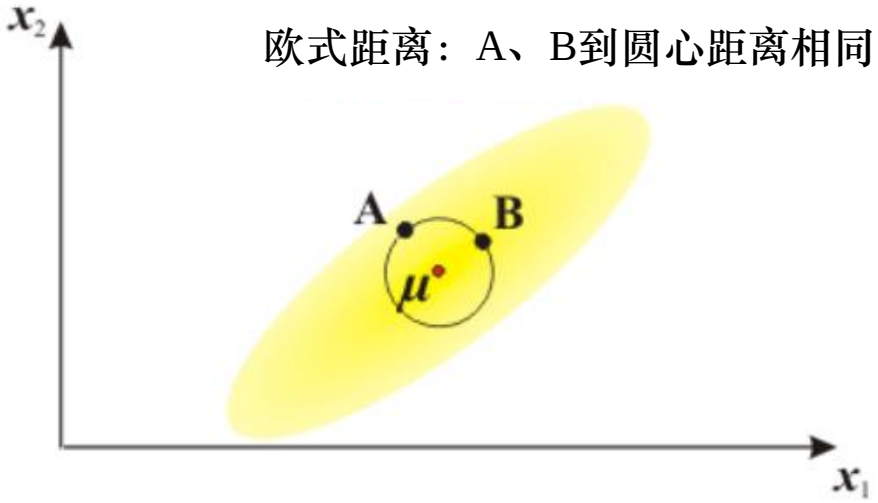
$$D(x_i,x_j)=\sqrt{(x_i-x_j)^T S^{-1}(x_i-x_j)}$$

特点：

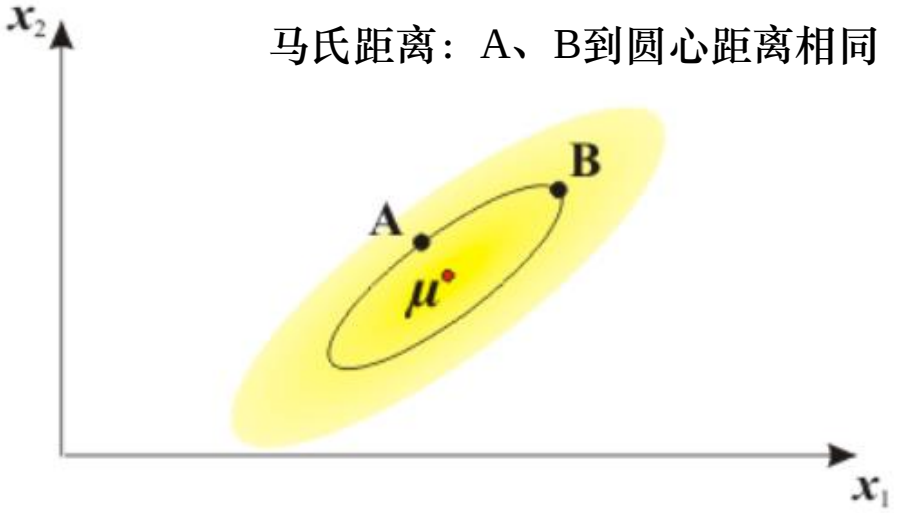
- 量纲无关，排除变量之间相关性的干扰；
- 以总体样本为基础，同样的两个样本在不同的总体中，计算出的马氏距离通常是不相同的；
- 适用于样本数大于维数，否则S不可逆。

三、距离

欧式距离：A、B到圆心距离相同



马氏距离：A、B到圆心距离相同



四、特征值分解与奇异值分解

在机器学习领域，有相当多的应用与特征值或者奇异值有关，比如做feature reduction的PCA，做数据压缩（以图像压缩为代表）的算法，还有做搜索引擎语义层次检索的LSI等。

- 特征值分解和奇异值分解在机器学习领域都是常用的数学方法；
- 特征值分解和奇异值分解有着很紧密的关系；
- 特征值分解和奇异值分解的目的都是为了提取出一个矩阵最重要的特征。

四、特征值分解与奇异值分解

1. 特征值分解：

方阵$A \in R^{n\times n}$的特征ξ满足$A\xi = \lambda\xi$，λ称为特征向量ξ对应的特征值。一个矩阵的一组特征向量是一组正交向量。设Q为矩阵A的特征向量组成的矩阵，$\sum$为特征值构成的对角阵，则$AQ = Q\sum$。矩阵A的特征值分解就是将一个矩阵分解成$A = Q\sum Q^{-1}$的形式。

具体的：

$$\sum = \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{pmatrix}$$

其中$\lambda_1, \cdots, \lambda_n$为特征值，由大到小排列。

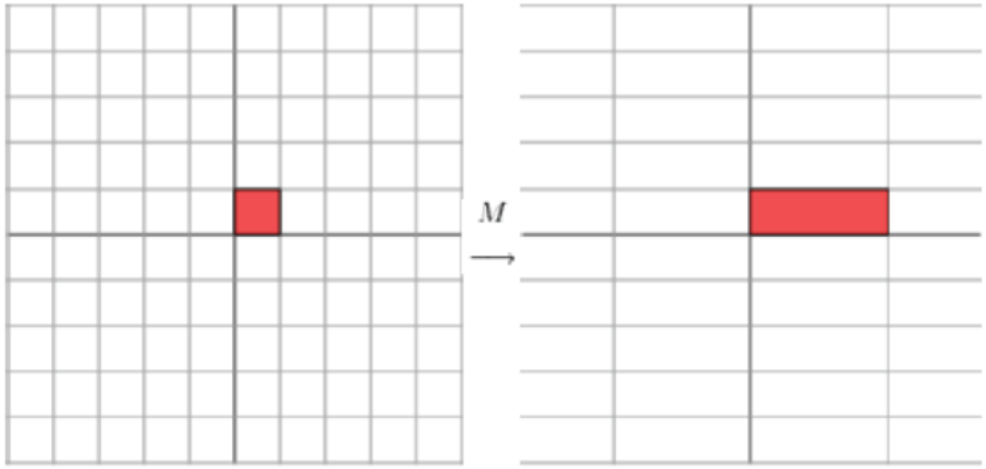
四、特征值分解与奇异值分解

特征值与特征向量的几何意义：

一个矩阵对应着一个变换，也就是说一个矩阵与某个向量相乘时，相当于将该向量变成另外一个方向或长度都不同的新向量。在这个变换的过程中，原向量主要发生旋转、伸缩的变化。如果矩阵对某一个向量或某些向量只发生伸缩变换，不对这些向量产生旋转的效果，那么这些向量就称为这个矩阵的特征向量，伸缩的比例就是特征值。

四、特征值分解与奇异值分解

$M=\begin{bmatrix}3 & 0\\ 0 & 1\end{bmatrix}$ 该矩阵对应的线性变换是下面的形式：

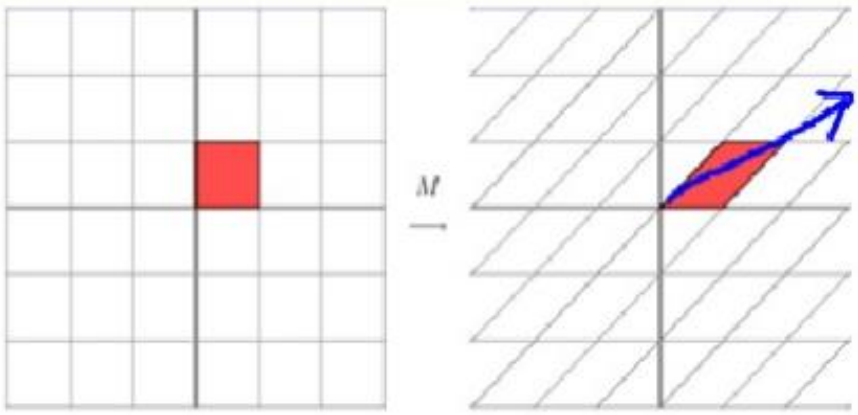


因为这个矩阵M乘以向量(x,y)的结果是：$\begin{bmatrix}3 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}3x\\ y\end{bmatrix}$

上面的矩阵是对称的，所以这个变换是一个对x,y轴的方向的一个拉伸变换（对角线上的每一个元素分别对每一个维度进行拉伸，当值>1时，是拉长，当值<1时是缩短。当矩阵不对称时会怎样？

四、特征值分解与奇异值分解

$M=\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}$ 它所描述的变换如下图所示：



这个矩阵M乘以向量(x,y)的结果是：$\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}x+y\\ y\end{bmatrix}$

这其实是在平面上对一个轴进行的拉伸变换（如蓝色箭头所示），在图中，蓝色的箭头是一个最主要的变化方向（变化方向可能有不止一个）。如果我们想要描述好一个变换，我们就描述好主要的变化方向就好了。

四、特征值分解与奇异值分解

特征值分解得到的结果恰好描述了变化的方向。我们知道，分解得到的矩阵∑是一个对角阵，对角元素为特征值从大到小排列，这些特征值对应的特征向量就是来描述矩阵的变化方向的（从主要方向到次要方向排列）。利用这N个变化方向，就可以近似这个矩阵（变换）。

总之，特征值表示的是这个特征到底有多重要，而特征向量表示这个特征是什么。但是特征值分解的局限性在于要求矩阵必须是方阵，那么不是方阵的情况呢?

四、特征值分解与奇异值分解

现实世界中，我们看到的大部分矩阵都不是方阵，比如说有N个学生，每个学生有M科成绩，这样形成的一个$N \times M$的矩阵通常不是方阵，我们怎样才能描述这样普通的矩阵的重要特征呢？奇异值分解（SVD）就是用来做这件事情的一种矩阵分解方法：

四、特征值分解与奇异值分解

分解形式： $A = U \sum V^T$

假设A是一个$m \times n$的矩阵，那么得到的U是一个$m \times m$的方阵(称为左奇异向量)，$\sum$是一个$m \times n$的对角矩阵(对角线上的元素为奇异值，除了对角线元素都是0)，V^T是一个$n \times n$的矩阵（称为右奇异向量）：

$$\left[A \right]_{m \times n} = \left[U \right]_{m \times m} \times \left[\sum \right]_{m \times n} \times \left[V^T \right]_{n \times n}$$

四、特征值分解与奇异值分解

- 奇异值与特征值关系

我们要对矩阵A进行奇异值分解，利用 A^TA 将会得到下面这样一个方程：

$$(A^TA)v_i = \lambda_i v_i$$

我们利用这个方阵求特征值λ_i以及特征向量组成的矩阵V，V中的每一个元素v就是我们前面提到的右奇异向量。

此外还可以根据下面的公式求得：

$$\sigma_i = \sqrt{\lambda_i}$$

$$u_i = \frac{1}{\sigma_i} A v_i$$

这里的σ就是奇异值，u就是左奇异向量。

四、特征值分解与奇异值分解

奇异值σ跟特征值类似，在矩阵$\sum$中也是从大到小排列，而且σ的值减少的特别的快，在很多情况下，前10%甚至1%的奇异值的和就占了全部的奇异值之和的99%以上了。我们也可以用前r（r远小于m,n）个奇异值来近似描述矩阵，即部分奇异值分解：

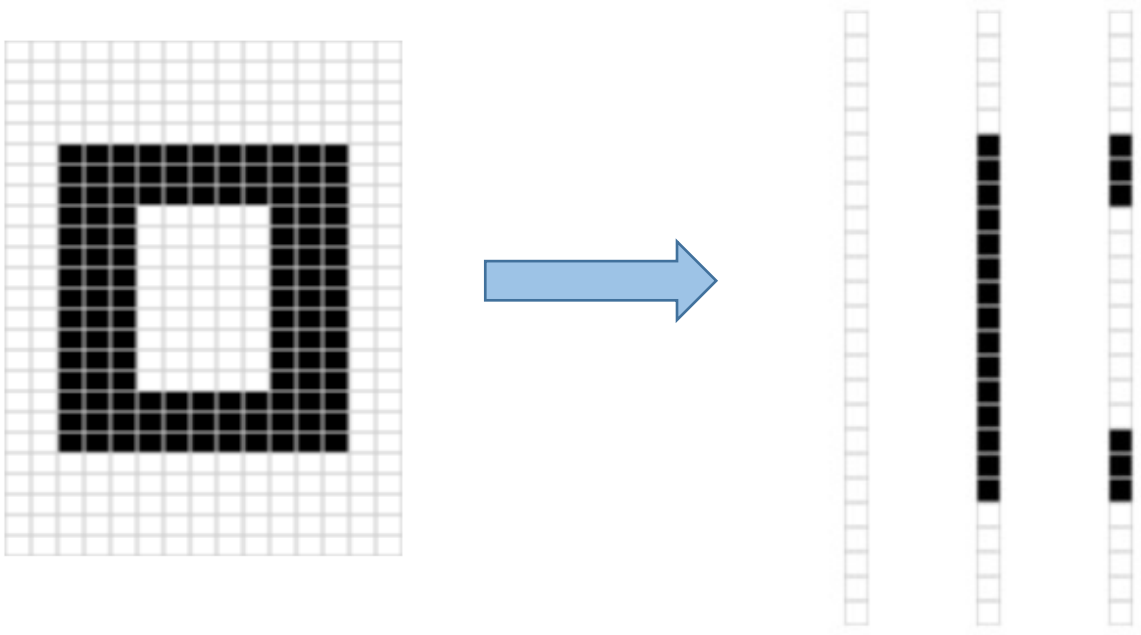
$$\underset{m\times n}{\left[A\right]}=\underset{m\times r}{\left[U\right]}\times\underset{r\times r}{\left[\sum\right]}\times\underset{r\times n}{\left[V^T\right]}$$

右边的结果是一个接近A的矩阵，r越接近m或n，相乘的结果越接近于A。

四、特征值分解与奇异值分解

- 奇异值分解在数据表达中的应用

假设我们有一张15*25的图像数据，该图像主要由三部分组成：



我们将图像表示成 15*25 的矩阵，矩阵的元素对应着图像的不同像素，如果像素是白色的话，就取 1，黑色的就取 0。我们得到了一个具有375个元素的矩阵。

四、特征值分解与奇异值分解

$$M=\begin{bmatrix}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1
\end{bmatrix}$$

对矩阵M进行奇异值分解，得到：

$$\sigma_1 = 14.72, \sigma_2 = 5.22, \sigma_3 = 3.31$$

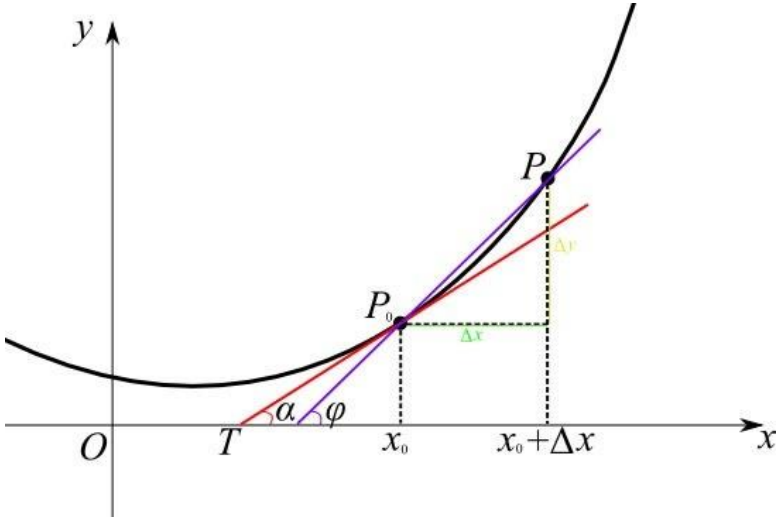
$$M = u_1\sigma_1{v_1}^T + u_2\sigma_2{v_2}^T + u_3\sigma_3{v_3}^T$$

每个v_i具有15个元素，每个u_i具有25个元素，σ_i对应不同的奇异值。这样就可以用123个元素来表示具有375个元素的图像。

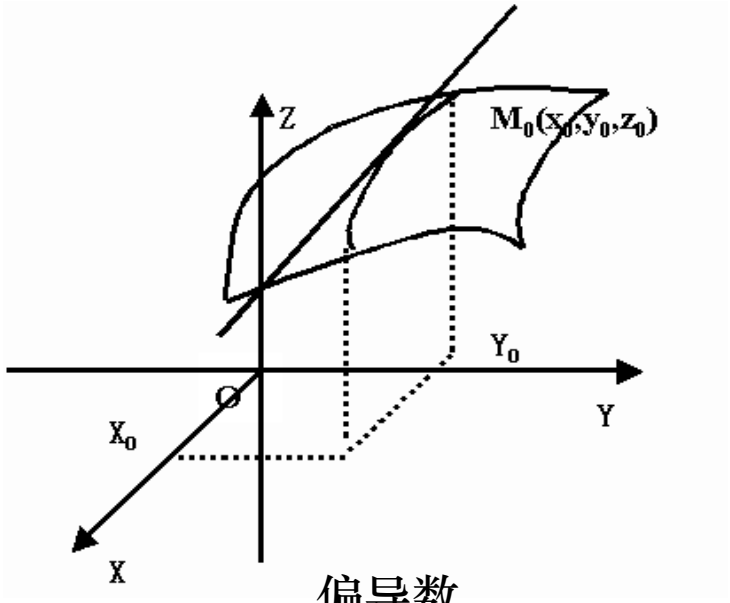
对应：
u_i三个，每个长度25；
v_i三个，每个长度15；
3个长度为1的奇异值

五、其他概念

- 导数：$f'(x_0)=\lim\limits_{\Delta x\to 0}\dfrac{\Delta y}{\Delta x}=\lim\limits_{\Delta x\to 0}\dfrac{f(x_0+\Delta x)-f(x_0)}{\Delta x}$



导数



偏导数

- 偏导数：

二元函数$u=f(x,y)$的偏导数就是函数沿两个坐标轴方向的导数：u'_x，u'_y

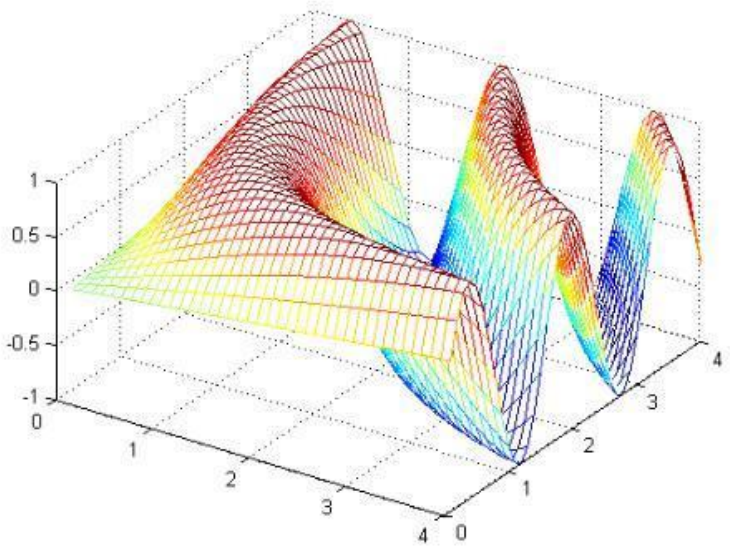
五、其他概念

- **方向导数：**

若函数$u=f(x,y)$在点(x_0,y_0)处可微，则函数f在点(x_0,y_0)处沿任一方向 $\vec{l^0}=(\cos\alpha,\cos\beta)$的方向导数存在为：

$$\frac{\partial u}{\partial l}=\frac{\partial u}{\partial x}\cos\alpha+\frac{\partial u}{\partial l}cos\beta$$

- **梯度：** $\nabla u=(u'_x(x,y),u'_y(x,y))$



梯度方向是函数增长最快的方向

- **泰勒展开：**

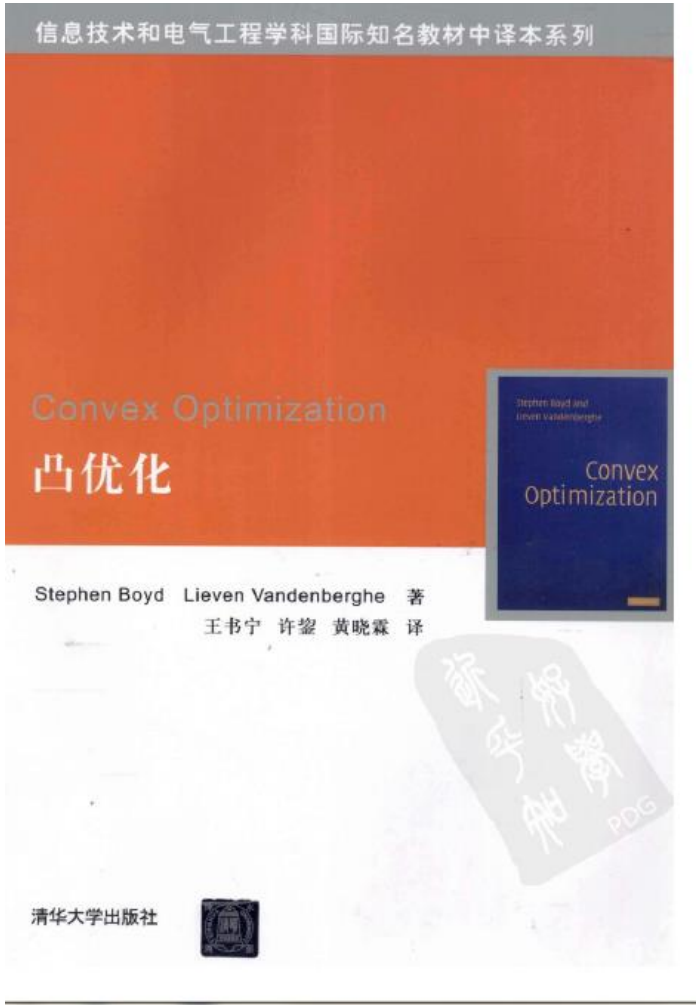
$$f(x)=\sum_{i=0}^{n}\frac{f^{(i)}(x_0)}{i!}(x-x_0)^i+o(x-x_0)^n$$

凸优化理论基础

最优化方法是一种数学方法，它是研究在给定约束之下如何寻求某些因素(的量)，以使某一(或某些)指标达到最优的一些学科的总称。

学习和工作中遇到的大多问题都可以建模成一种最优化模型进行求解，同时大部分的机器学习算法的本质都是建立优化模型，通过最优化方法对目标函数（或损失函数）进行优化，从而训练出最好的模型。

书籍介绍：



理论部分由4章构成，涵盖了凸优化的所有基本概念和主要结果，还详细介绍了几类基本的凸优化问题以及将特殊的优化问题表述为凸优化问题的变换方法。

应用部分由3章构成，分别介绍凸优化在解决逼近与拟合、统计估计和几何关系分析这三类实际问题中的应用。算法部分也由3章构成，依次介绍求解无约束凸优化模型、等式约束凸优化模型以及包含不等式约束的凸优化模型的经典数值方法等。

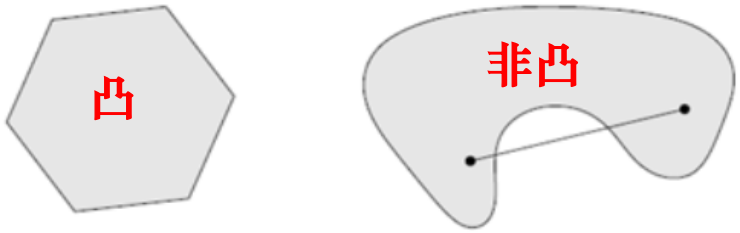
一、凸集

- 定义：

给定一个集合C，满足下列条件则称为凸集：

$$x, y \in C \Rightarrow tx + (1-t)y \in C, \forall t \in [0,1]$$

一句话来概括凸集就是:集合内任意两点间连线仍然在集合内。



- 凸集例子：

空集、点、线都是凸集合；

范数球：半径为 r 的范数球为：$\{x: \|x\| \leq r\}$；

超平面：给定任意a, b $\{x: a^T x = b\}$;

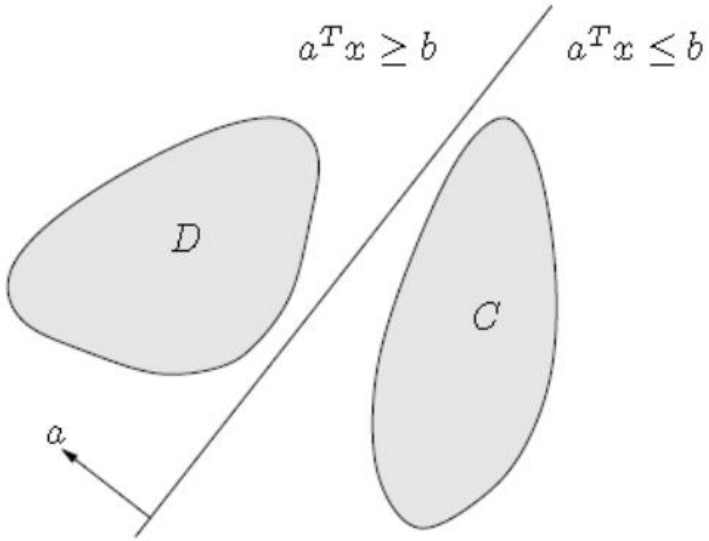
半空间：$\{x: a^T x \leq b\}$;

多面体：$\{x: A^T x \leq b\}$.

一、凸集

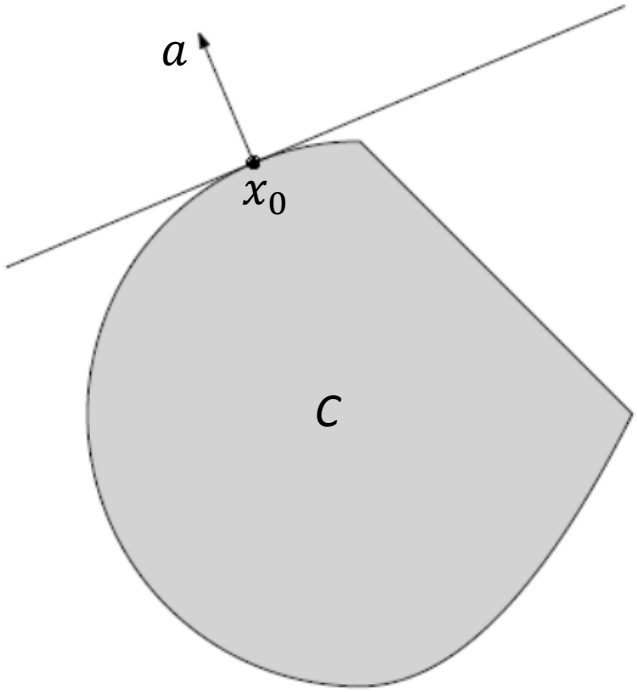
- **凸集特性：**

可分离超平面定理：两个不相交的凸集总存在一个超平面能将两者分离，如果$C \cap D = \emptyset$，那么总存在着a, b使得有：C包含于$\{x: a^T x \leq b\}$，D包含于$\{x: a^T x \geq b\}$如下图所示：



一、凸集

支撑超平面理论：凸集边界上的一点必然存在一个支撑超平面穿过该点，即如果 C 是非空凸集，x_0 为其边界点，那么必然存在一个超平面，使得：C包含于$\{x: a^T x \leq a^T x_0\}$，如下图所示：



二、凸函数

1. 定义：

给定映射$f: R^n \to R$并且定义域为凸集，如果：

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$

对于任意的$0 \leq t \leq 1$都成立，则称f为凸函数。如下图所示：

两点连线总在凸函数图像上方

严格凸：任给$x \neq y$，定义中的不等号严格成立；

强凸：对于参数$m > 0$: $f - \frac{m}{2}\|x\|_2^2$依旧是一个凸函数；

强凸 $\Rightarrow$ 严格凸 $\Rightarrow$ 凸

二、凸函数

2. 凸函数例子：

抛物线：$f(x) = x^2, \forall x \in R$;

线性函数：$f(x) = ax + b$即是凸函数又是非凸函数；

二次函数：$\frac{1}{2}x^T Qx + b^T x + c$，当$Q$为半正定矩阵时为凸函数；

最小平方损失函数：$\|y - Ax\|_2{}^2$总是凸的；

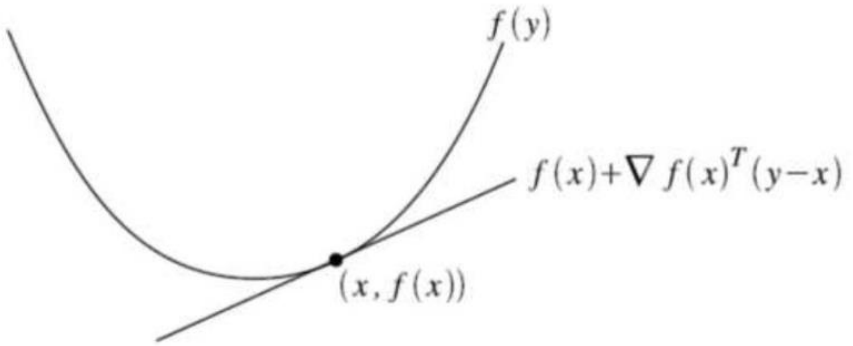
最大值函数：$f(x) = max(x_1, \cdots, x_n)$

二、凸函数

3. 凸函数特性：

一阶特性：假设 f 为处处可微的凸函数，那么对于定义域中的任意两点 x 和 y，有：

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$



切线总在凸函数图像下方

二阶特性：如果函数f二阶可微，则f为凸函数当且仅当对于定义域中的任意一点x，有$\nabla^2 f(x) \geq 0$。

三、最优化问题

$$\begin{aligned} \min \quad & f(x) \\ s.t. \quad & g_i(x) \le 0, i = 1, \cdots, m, \\ & h_i(x) = 0, i = 1, \cdots, p, \end{aligned}$$

目标函数

不等式约束

等式约束

① 当问题中的$m = p = 0$时，即它不含任何限定条件时，称该问题为无约束问题。当$m + p > 0$时，即包含限定条件时，称为约束问题；

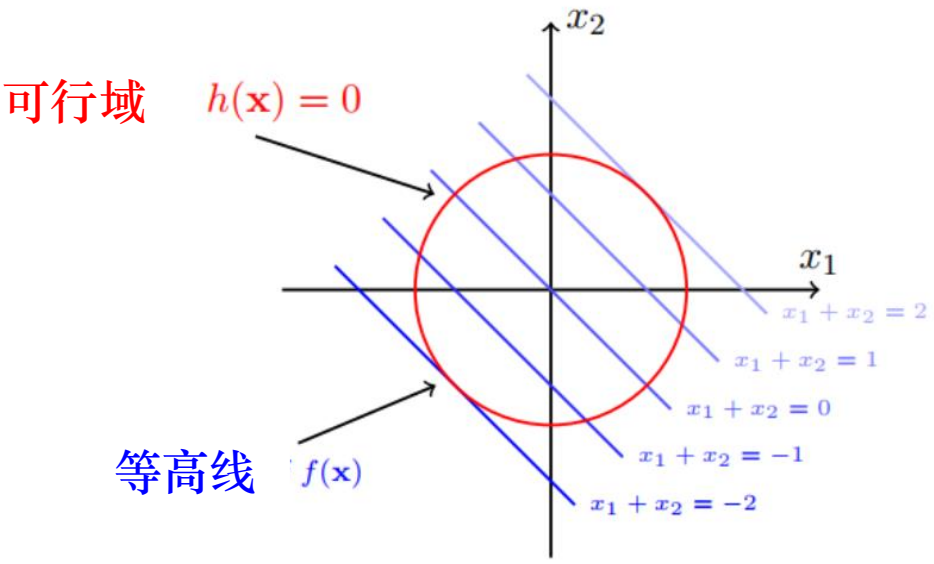
② 称满足所有约束条件的点为可行点，全体可行点组成的集合称为可行域；问题的最优值是指目标函数在可行域上取得的最小值；

③ 最优解是指取得最优值时对应的可行点，分为全局最优解和局部最优解；若$f(x)$为凸函数，$g(x)$为凸函数，$h(x)$为线性函数，则该问题称为凸优化.

四、拉格朗日乘子法与KKT条件

1.等式约束情形：

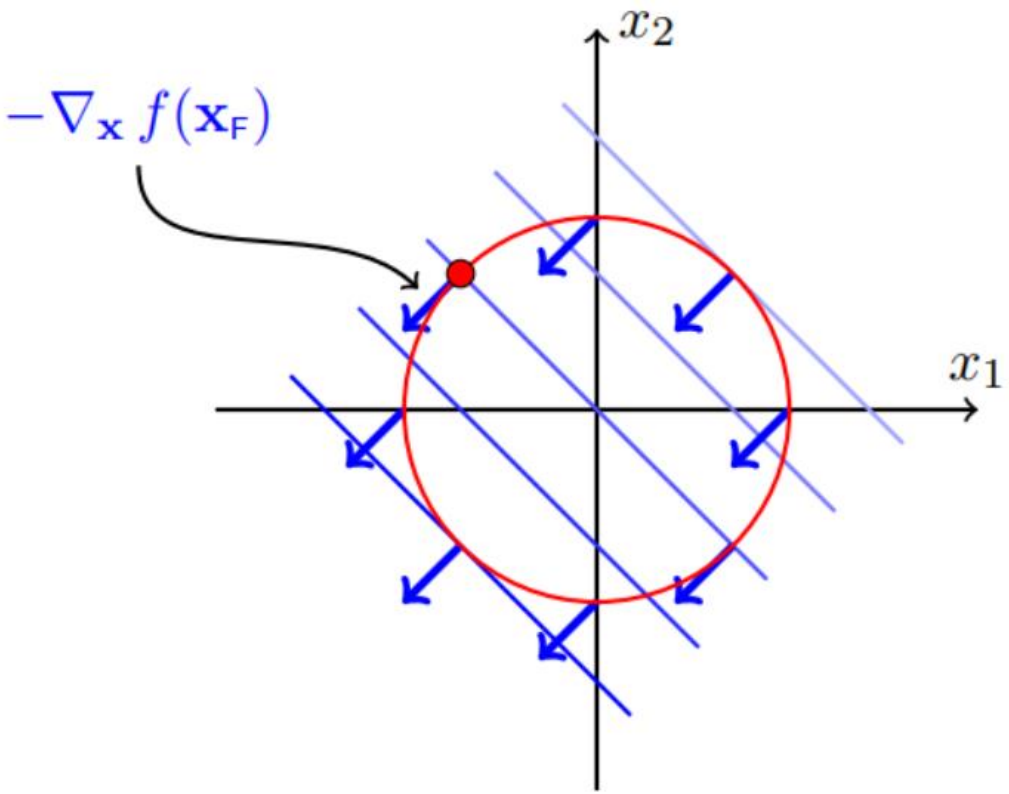
考虑$f(x)=x_1+x_2$，等式约束$h(x)=x_1{}^2+x_2{}^2-2$，求解极小值点。$f(x)$在二维平面上画出等高线就是一条条斜率相同的直线，$h(x)=0$的等高线就是一个圆。在圆圈$h(x)$限制下，直线$f(x)$的最小值为-2，在左下角的交点上。



$$h(x)=x_1{}^2+x_2{}^2-2$$

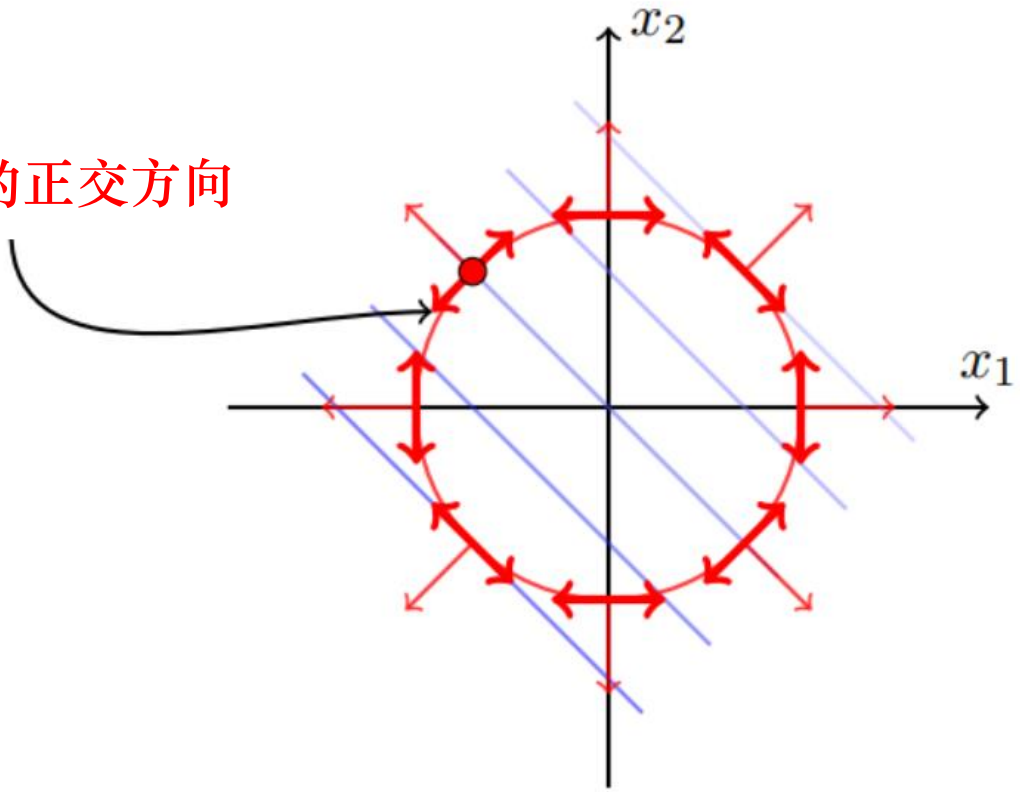
四、拉格朗日乘子法与KKT条件

不考虑圆$h(x)$的限制时，$f\ (x)$要得到极小值，需要往$f\ (x)$的负梯度(下降最快的方向)方向走，如下图蓝色箭头。



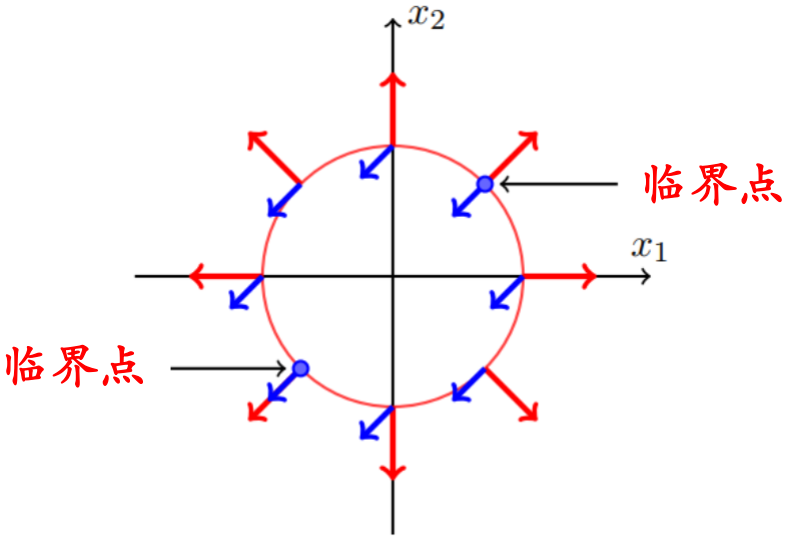
四、拉格朗日乘子法与KKT条件

如果考虑圆$h(x)$的限制，要得到极小值，需要沿着圆的切线方向走，如下图红色**粗**箭头。注意这里的方向不是$h(x)$的梯度，而是正交于$h(x)$的梯度，$h(x)$梯度如下图的红色细箭头。在极小值点，$f(x)$和$h(x)$的等高线是相切的。



四、拉格朗日乘子法与KKT条件

容易发现，在关键的极小值点处，$f(x)$的负梯度和$h(x)$的梯度在同一直线上，如下图左下方critical point的蓝色和红色箭头所示。



在极小值点，$h(x)$和$f\ (x)$的梯度在同一直线上，有：

$$\nabla_x f(x^*) = \mu \nabla_x h(x^*)$$

所以，对于$f(x)$和$h(x)$而言，只要满足上面这个式子，同时使得$\boldsymbol{h(x) = 0}$，解得的x就是我们要求的极小值点。要做到这一点，可以构造一个**拉格朗日函数**。

四、拉格朗日乘子法与KKT条件

对拉格朗日函数求偏导等于0求解，恰好等价于“满足前面的式子，同时使得 $h(x)=0$”，原问题转化为对拉格朗日函数求极值问题，这就是拉格朗日乘子法：

$$\min_{x \in R^2} \quad f(x)$$
$$s.t. \quad h(x)=0$$

定义拉格朗日函数为：

$$L(x,\mu)=f(x)+\mu h(x)$$

则x^*是一个局部极小值等价于存在唯一的一个μ^*使得：

$$\nabla_x L(x^*,\mu^*)=0$$

$$\nabla_\mu L(x^*,\mu^*)=0$$

2.不等式约束情形：

等式约束 $h(x)=0$可以在平面上画出一条等高线，而$g(x)\leq 0$是一个由很多个等高线堆叠成的区域，我们把这块区域称为可行域。

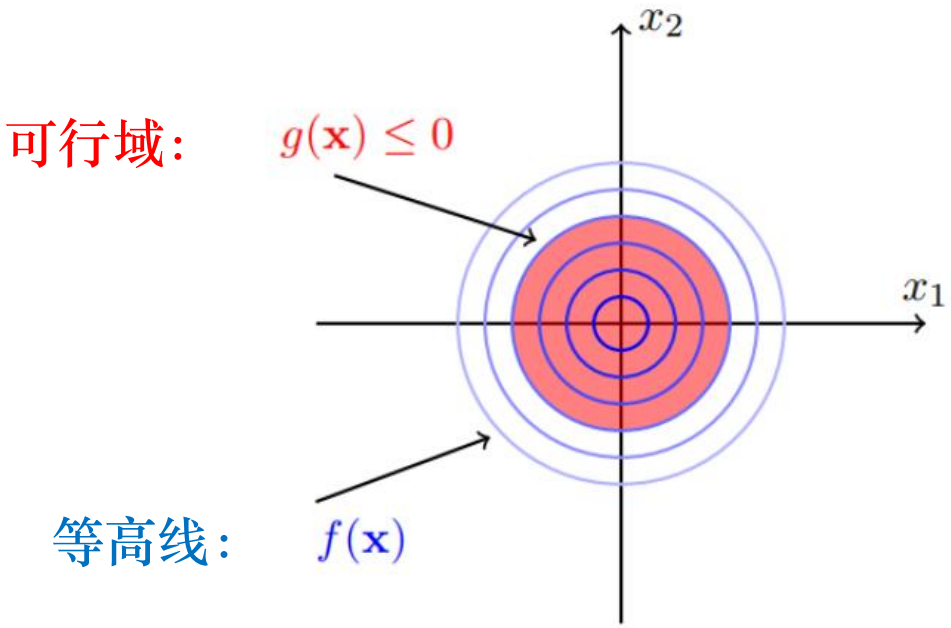
不等式约束分两种情况来讨论，第一种是（不考虑可行域限制时的）极小值点落在可行域内（不包含边界），第二种是（不考虑可行域限制时的）极小值点落在可行域外（包含边界）。

下面举两个例子来解释这两种情况。

四、拉格朗日乘子法与KKT条件

- 极小值点落在可行域内（不包含边界）

考虑目标函数$f(x) = {x_1}^2 + {x_2}^2$，不等式约束$g(x) = {x_1}^2 + {x_2}^2 - 1 \le 0$，显然 $f\ (x)$ 的极小值为原点 (0,0) 落在可行域内。可行域以原点为圆心，半径为1。这种情况约束不起作用，考虑极小值点x^*，这个时候，$g(x^*) < 0, f(x^*)$的梯度等于0。



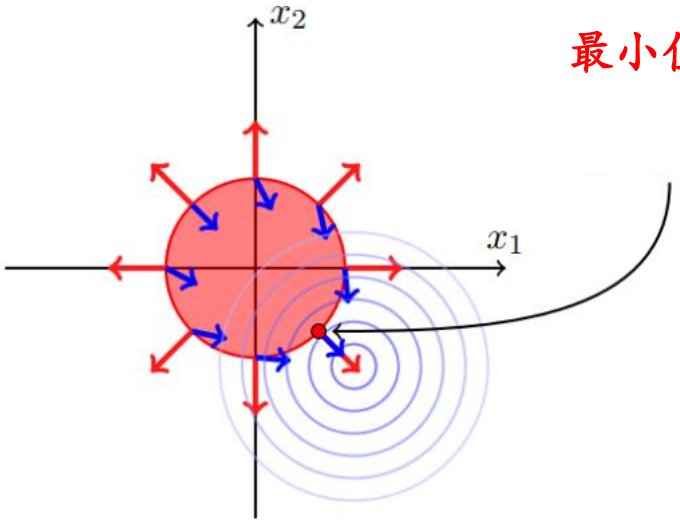
$$g(x) = {x_1}^2 + {x_2}^2 - 1$$

四、拉格朗日乘子法与KKT条件

- 极小值点落在可行域外（包含边界）

考虑$f(x)=(x_1-1.1)^2+(x_2+1.1)^2$，不等式约束$g(x)={x_1}^2+{x_2}^2-1\le 0$，这种情况约束起作用，要考虑$f(x)$在可行域内的极小值点。

沿着$f(x)$的负梯度方向走，才能走到极小值点，如下图的蓝色箭头。此时$g(x)$的梯度往区域外发散，如下图红色箭头。走到极小值点时，$g(x)$的梯度和$f(x)$的负梯度同向。因为极小值点在边界上，这个时候$g(x)=0$。



$-\nabla_x f(x)=\lambda\nabla_x g(x)$ 和 $\lambda>0$

四、拉格朗日乘子法与KKT条件

总结：

情形 1：无约束局部最小值出现在可行域内。

$$g(x^*) < 0$$

$$\nabla_x f(x^*) = 0$$

情形 2：无约束局部最小值出现在可行域外部。

$$g(x^*) = 0$$

$$-\nabla_x f(x^*) = \lambda \nabla_x g(x^*) \text{其中} \lambda > 0$$

以上两种情况均可构造拉格朗日函数来转换求解问题。对于不等式约束的优化，需要满足三个条件，满足这三个条件的解 x^*就是极小值点。

这三个条件就是著名的***KKT*** **(*karush-Kuhn-Tucker*) 条件**，它整合了上面两种情况的条件：

四、拉格朗日乘子法与KKT条件

KKT：

给定优化问题：

$$\min_{x \in R^2} \quad f(x)$$
$$s.t. \quad g(x) \leq 0$$

定义拉格朗日函数：$L(x, \lambda) = f(x) + \lambda g(x)$

则x^*是一个局部极小值等价于存在唯一的一个λ^*满足下列KKT条件：

$$\nabla_x L(x^*, \lambda^*) = 0$$
$$\lambda_j{}^* \geq 0$$
$$\lambda^* g(x^*) = 0$$
$$g_j(x^*) \leq 0$$

四、拉格朗日乘子法与KKT条件

总结：

给定有约束的优化问题：

$$\begin{aligned} \min_{x \in R^n} \quad & f(x) \\ s.t. \quad & h_i(x) = 0, i = 1, \cdots, l \\ & g_j(x) \le 0, j = 1, \cdots, m \end{aligned}$$

定义拉格朗日函数：$L(x, \mu, \lambda) = f(x) + \mu^t h(x) + \lambda^t g(x)$

则x^*是一个局部极小值等价于存在唯一的一个λ^*满足下列KKT条件：

$$\nabla_x L(x^*, \mu^*, \lambda^*) = 0$$

$$\lambda_j{}^* \ge 0, j = 1, \cdots, m$$

$$\lambda_j{}^* g_j(x^*) = 0, j = 1, \cdots, m$$

$$g_j(x^*) \le 0, j = 1, \cdots, m$$

$$h(x^*) = 0$$

五、对偶理论

在约束优化问题中，常常用拉格朗日方法来将原始问题转为对偶问题，通过对偶问题的解得到原始问题的解。

对偶问题最优值不一定等于原问题的最优值（弱对偶），但有两点性质：

① 满足某些条件时，对偶问题和原问题最优值相等（强对偶）；

② 无论原始问题是否是凸的，对偶问题都是凸优化问题；

在某些情况下，原问题求解较为复杂，求解对偶问题可以间接得到原问题的解，而且对偶问题是凸优化，易于求解。所以在许多机器学习方法中利用对偶来求解非常有效。

五、对偶理论

考虑原问题：

$$\min f(x)$$
$$s.t.\ g_i(x) \le 0, i = 1, \cdots, m$$
$$h_i(x) = 0, i = 1, \cdots, p$$

引入Lagrange函数：

$$L(x, \lambda, v) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{i=1}^{p} v_i h_i(x)$$

拉格朗日乘子

显然有：$L(x, \lambda, v) \le f(x)$，从而 $\inf_{x \in R^n} L(x, \lambda, v) \le \inf_{x \in D} L(x, \lambda, v) \le \inf_{x \in D} f(x) = p^*$，则任给$\lambda \ge 0$，不等号左侧都是$p^*$的一个下界，我们要寻找这些下界中最好的下界，从而得到下述对偶问题：

$$\max \inf_{x \in R^n} L(x, \lambda, v)$$
$$s.t. \lambda \ge 0.$$

五、对偶理论

- **对偶间隙：**

设原问题最优值为p^*，对偶问题最优值为d^*，则称原问题与对偶问题的最优值之差$p^* - d^*$为对偶间隙；

- **弱对偶定理：**

凸优化的对偶间隙永远是非负值；

- **强对偶定理：**

对凸优化问题，如果存在可行点x使得全部不等式约束严格成立，即：$h_i(x) = 0, i = 1,2,\cdots,p;\ \ g_i(x) < 0, i = 1,2,\cdots,m,$ 则对偶间隙为零。

六、求解优化算法

1. 梯度下降法（Gradient Descent）：

梯度下降法是最早最常用的优化方法。它通过迭代的方式，使得下一个点能比上一个点有更小的函数值，由于负梯度方向是最快下降方向，所以选择负梯度方向为搜索方向：

$$p_k = -\nabla f(x_k)$$

然后沿着搜索方向进行搜索可得下述迭代公式：

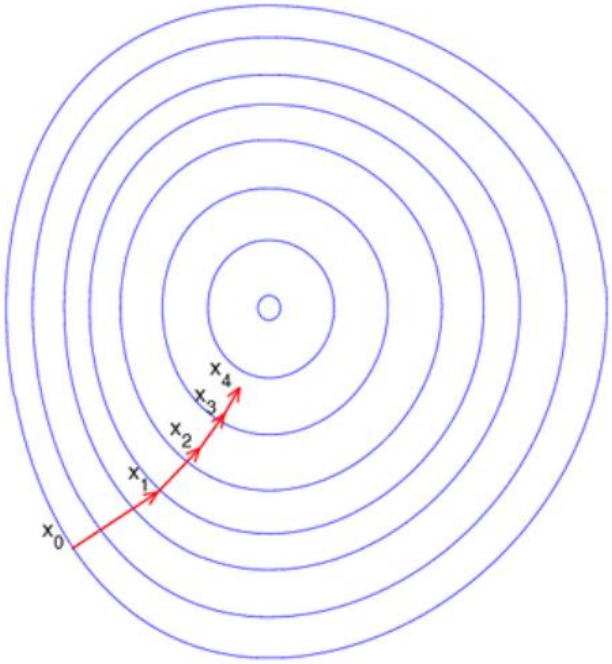
$$x_{k+1} = x_k - t_k \nabla f(x_k)$$

其中t_k为沿负梯度方向的搜索步长；它满足：

$$f(x_k - t_k \nabla f(x_k)) = \min f(x_k - t \nabla f(x_k))$$

即，我们希望x_{k+1}为搜索方向上函数值“最小”的点。

六、求解优化算法



评价：①梯度下降法实现简单，当目标函数是凸函数时，梯度下降法的解是全局解。一般情况下，其解不保证是全局最优解；

②梯度下降法在接近最优解的区域，步长变小，收敛速度明显变慢，利用梯度下降法求解需要很多次的迭代。

六、求解优化算法

在机器学习中，基于基本的梯度下降法发展了几种梯度下降方法，分别为随机梯度下降法、批量梯度下降法、小批量梯度下降法等。

批量梯度下降法(Batch Gradient Descent, BGD)

是梯度下降法的最原始的形式,其特点就是每一次训练迭代都需要利用所有的训练样本。这样会导致训练速度随着训练样本的增大极大地减慢,不适合于大规模训练样本的场景,但是,其解是全局最优解,精度高。

六、求解优化算法

随机梯度下降法(Stochastic Gradient Descent, SGD)

是梯度下降法的改进形式之一，每次参数更新过程中,只需要选取训练集中的一个训练样本,因此训练速度快,但是因为其只是利用了训练集的一部分知识,因此解为局部最优解,精度较低。

小批量梯度下降法(Mini-Batch Gradient Descent, MBGD)

训练速度快，同时精度较高，每一次训练迭代在训练集中随机采样batch_size个样本，这个改进版本在深度学习的网络训练中有着广泛地应用，因为其既有较高的精度，也有较快的训练速度。

2.牛顿法(Newton's method)

假设$f(x)$具有二阶连续偏导数，假设第k次迭代值为x^k的值，那么可将$f(x)$在x^k附近进行二阶泰勒展开得到：

$$f(x) \approx T(x) = f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k)(x - x^k)$$

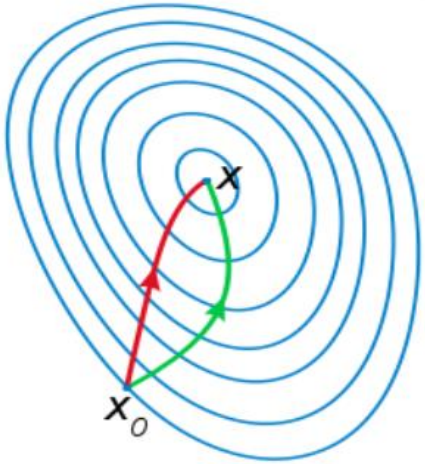
我们对上述公式求导可得：

$$\nabla f(x^k) + \nabla^2 f(x^k)(x - x^k) = 0$$

假设其中$\nabla^2 f(x^k)$可逆，我们就可以得到牛顿法的迭代公式为：

$$x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k)$$

六、求解优化算法



注：红色的为牛顿法路径，绿色的为梯度下降法路径

牛顿法是二阶收敛，梯度下降是一阶收敛，所以牛顿法更快。更通俗地讲，如果你想找一条最短的路径走到一个盆地的最底部，梯度下降法每次只从你当前所处位置选一个坡度最大的方向走一步，牛顿法在选择方向时，不仅会考虑坡度是否够大，还会考虑你走了一步之后，坡度是否会变得更大。所以，牛顿法具有全局思想，但是牛顿法求逆矩阵运算复杂。

六、求解优化算法

3.其他方法

共轭梯度法：介于最速下降法与牛顿法之间的一个方法，仅利用一阶导数信息但克服了最速下降法收敛慢的缺点，又避免了牛顿法需要存储Hesse矩阵并求逆的缺点。 其优点是存储量小，而且不需要任何外来参数。

信赖域方法：在迭代中给出一个信赖域，这个信赖域一般是当前迭代点x_k的一个小邻域。然后在这个邻域内求解一个子问题，得到试探步长s_k，如果试探步较好，在下一步信赖域扩大或保持不变，否则减小信赖域。

其他方法：对偶坐标下降算法(DCDM)、序列最小化算法SMO(Sequential minimal optimization)等等。

➢概率统计基础

一、机器学习为什么要使用概率?

1. 我们需要借助概率论来解释分析机器学习为什么是这样的，有什么依据，同时反过来借助概率论来推导出更多机器学习算法。

2. 是因为机器学习通常必须处理不确定性问题，例如：

- 被建模系统内在的随机性、不完全观测：假如你要预测抽的纸牌点数，虽然牌的数量和所有牌有什么是确定的，但是若我们随机抽一张，这个牌是什么是随机的。这个时候就要使用概率去建模了。

二、基本概念

1.随机变量

定义：表示随机现象（在一定条件下，并不总是出现相同结果的现象称为随机现象）各种结果的变量。例如某一时间内公共汽车站等车乘客人数，医院一分钟出生的婴儿个数等等。

随机变量可以是离散的或者连续的。比如：

① 离散：一次掷20个硬币，k个硬币正面朝上，k是随机变量，k的取值只能是自然数0，1，2，…，20，而不能取小数3.5。

② 连续：公共汽车每15分钟一班，某人在站台等车时间x是个随机变量，x的取值范围是[0,15)，在这个区间内可取任一实数。

2. 概率分布

给定随机变量的取值范围，概率分布就是导致该随机事件出现的可能性。而从机器学习的角度，概率分布就是符合随机变量取值范围的某个对象，属于某个类别或服从某种趋势的可能性。

3. 联合概率

联合概率为两个事件同时发生的概率。记为：P(A and B)或直接P(AB)。

4. 条件概率

定义：其记号为$P(A|B)=\frac{P(AB)}{P(B)}$，表示在给定条件B下A事件发生的概率。

链式法则：$P(x^1,\cdots,x^n)=p(x^1)\prod_{i=2}^{n}p\left(x^1,\cdots,x^{i-1}\right)$

二、基本概念

条件概率举例：

P(第二次投硬币是正面|第一次投硬币是正面)：就是在“第一次投硬币是正面”时“第二次投硬币是正面”的概率。P(第二次投硬币是正面|第一次投硬币是正面)的结果是1/4？错，答案是1/2。

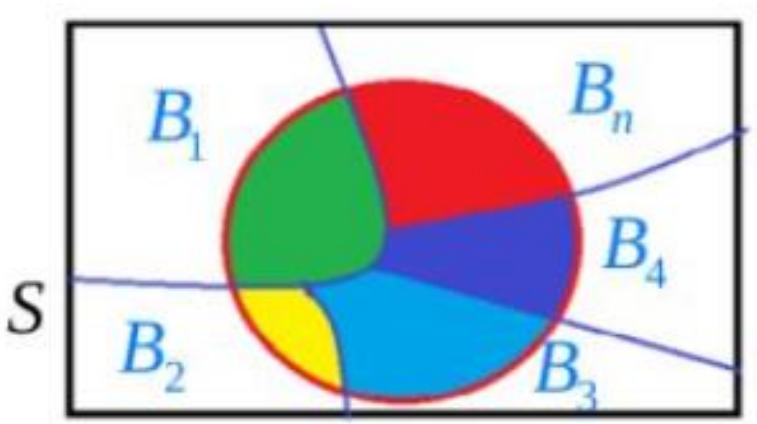
条件概率的两种情况：

① B事件的结果不会影响到A事件的发生。如上例，A事件发生的概率=A事件单独发生的概率。记为：P(A|B) =P(A)；

② B事件的结果会影响A事件的发生。如：若头天下雨，则第二天下雨的可能性会增大。即：A事件在B事件之后发生的概率> A事件单独发生的概率。记为：P(A|B)> P(A)。

三、全概率公式

若事件$B_1, B_2, \cdots, B_n$构成一个完备事件A，且都有正概率，则对事件A我们可以由各个事件B_i的概率导出其概率。如果我们把B_i看做原因，A事件看做结果，则全概率公式就是“由因导果”，是一种正向思维。公式表述如下：



定理：设$B_1, B_2, \cdots, B_n$为S的一个划分且$P(B_i) > 0$则有全概率公式：

$$P(A) = \sum_{j=1}^{n} P(B_j) \cdot P(A|B_j)$$

三、全概率公式

举例：

某地盗窃风气盛行，我们根据过往的案件记录，推断B_1今晚作案的概率是0.8，B_2今晚作案的概率是0.1，B_3今晚作案的概率是0.5，除此之外，还推断出B_1的得手率是0.1，B_2的得手率是1.0，B_3的得手率是0.5。现在的情况是该村确定有东西被偷，我们想知道是B_1偷的概率？

解：已知：

$$P(B_1) = 0.8, P(B_2) = 0.1, P(B_3) = 0.5;$$

将“村里有东西被偷”记为A，根据得手率可得到：

$$P(A|B_1) = 0.1, P(A|B_2) = 1.0, P(A|B_3) = 0.5$$

从而得：

$$P(A) = P(B_1)P(A|B_1) + P(B_2)P(A|B_2) + P(B_3)P(A|B_3) = 0.43$$

四、贝叶斯公式

贝叶斯公式：贝叶斯公式是建立在条件概率的基础上寻找事件发生的原因，“由果寻因”（即A已经发生的条件下，小事件B_i的概率），是一种逆向思维，设$B_1, B_2, \cdots, B_n$ 是样本空间Ω的一个划分，则对任一事件$A(P(A) > 0)$,有：

$$P(B_i|A) = \frac{P(B_i)P(A|B_i)}{\sum_{j=1}^{n} P(B_j)P(A|B_j)}$$

公式描述：公式中，事件B_i发生的概率为$P(B_i)$，事件B_i已经发生的条件下事件 A 发生的概率为$P(A|B_i)$，事件A发生的条件下事件B_i发生的概率为$P(B_i|A)$.

四、贝叶斯公式

举例：

某地盗窃风气盛行，我们根据过往的案件记录，推断B_1今晚作案的概率是0.8，B_2今晚作案的概率是0.1，B_3今晚作案的概率是0.5，除此之外，还推断出B_1的得手率是0.1，B_2的得手率是1.0，B_3的得手率是0.5。现在的情况是该村确定有东西被偷，我们想知道是B_1偷的概率？

解：　已知：$P(B_1)=0.8, P(B_2)=0.1, P(B_3)=0.5;$

将“村里有东西被偷”记为A，根据得手率可得到：

$$P(A|B_1)=0.1, P(A|B_2)=1.0, P(A|B_3)=0.5$$

从而得：

$$P(B_1|A)=\frac{P(B_1)P(A|B_1)}{\sum_{j=1} P(B_j)P(A|B_j)}=\frac{0.8*0.1}{0.8*0.1+0.1*1+0.5*0.5}=0.19$$

四、贝叶斯公式

贝叶斯理论及应用

数学领域	▪ 贝叶斯分类算法 (应用:统计分析、测绘学) ▪ 贝叶斯公式 (应用:概率空间) ▪ 贝叶斯区间估计 (应用:数学中的区间估计) ▪ 贝叶斯序贯决策函数 (应用:统计决策论)	▪ 贝叶斯风险 (应用:统计决策论) ▪ 贝叶斯估计 (应用:参数估计) ▪ 贝叶斯统计 (应用:统计决策论) ▪ 经验贝叶斯方法 (应用:统计决策论)
工程领域	▪ 贝叶斯定理 (应用:人工智能、心理学、遗传学) ▪ 贝叶斯分析 (应用:计算机科学) ▪ 贝叶斯逻辑 (应用:人工智能) ▪ 贝叶斯网络 (应用:人工智能)	▪ 贝叶斯分类器 (应用:模式识别、人工智能) ▪ 贝叶斯决策 (应用:人工智能) ▪ 贝叶斯推理 (应用:数量地理学、人工智能) ▪ 贝叶斯学习 (应用:模式识别)
其他领域	▪ 贝叶斯主义 (应用:自然辩证法)	▪ 有信息的贝叶斯决策方法 (应用:生态系统生态学)

五、期望、方差、协方差和相关系数

1. 数学期望:

在17世纪，有一个赌徒向法国著名数学家帕斯卡挑战，给他出了一道题目：甲乙两个人赌博，他们两人获胜的机率相等，比赛规则是先胜三局者为赢家，一共进行五局，赢家可以获得100法郎的奖励。当比赛进行到第四局的时候，甲胜了两局，乙胜了一局，这时由于某些原因中止了比赛，那么如何分配这100法郎才比较公平?

五、期望、方差、协方差和相关系数

1. 数学期望：

由概率论知甲获胜的可能性大。因为甲输掉后两局的可能性只有(1/2)×(1/2）=1/4，也就是说甲赢得后两局的概率为1－(1/4)=3/4，甲有75%的期望获得100法郎；而乙期望赢得100法郎就得在后两局均击败甲，乙连续赢得后两局的概率为1/4，即乙有25%的期望获得100法郎奖金。

依据上述可能性推断，甲乙双方最终胜利的客观期望分别为75%和25%，因此甲应分得75(法郎)，乙应分得25(法郎)。这个故事里出现了“期望”这个词，数学期望由此而来。

数学期望是每次可能结果的概率乘以其结果的和，反映随机变量平均值大小。

五、期望、方差、协方差和相关系数

- **离散随机变量：**

假设x是一个离散随机变量，其可能的取值有：$\{x_1,\cdots,x_n\}$，各个取值对应的概率为$p(x_k),k=1,\cdots,m$则其数学期望被定义为：$E(x)=\sum_{k=1}^{n}x_kP(x_k)$。

- **连续型随机变量：**

设x是一个连续型随机变量，其概率密度函数为$p(x)$，则其数学期望为：

$$E(X)=\int_{-\infty}^{+\infty}xp(x)dx$$

- **性质：**

$$E(C)=C,C\text{是常数};$$

$$E(CX)=CE(X);$$

$$E(X+Y)=E(X)+E(Y);$$

当X和Y相互独立时，有$E(XY)=E(X)E(Y)$.

2. 方差:

概率中，方差(Variance)用来衡量随机变量与其数学期望之间的偏离程度；数学表达式如下:

$$Var(x) = E([x - E(x)]^2) = E(x^2) - [E(x)]^2$$

方差越大，数据的波动越大；方差越小，数据的波动就越小。

统计中，样本中各数据与样本平均数的差的平方和的平均数叫做样本方差；样本方差的算术平方根叫做样本标准差。它们都是衡量一个样本集合波动大小的量，样本方差越大，样本数据的波动就越大。

3. 协方差：

二维随机变量 (X,Y)，X与Y之间的协方差定义为：

$$Cov(X,Y)=E\{[X-E(Y)]\cdot[Y-E(Y)]\}$$

其中：$E(X)$为分量X的期望，$E(Y)$为分量Y的期望。

协方差用来描述随机变量相互关联程度，它是X的偏差 $[X-E(X)]$ 与Y的偏差 $[Y-E(Y)]$ 的乘积的数学期望。由于偏差可正可负，因此协方差也可正可负。

1. 当协方差$Cov(X,Y)>0$时，称X与Y正相关；
2. 当协方差$Cov(X,Y)<0$时，称X与Y负相关；
3. 当协方差$Cov(X,Y)=0$时，称X与Y不相关。

4. 协方差矩阵：

设$X=(X_1,\cdots,X_n)^T$为n维随机变量，称矩阵：

$$C=(c_{ij})_{n\times n}=\begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} \end{pmatrix}$$

为n维随机变量X的协方差矩阵(covariance matrix)，也记为$D(X)$，

其中：$c_{ij}=Cov(X_i,Y_j),\ i,j=1,\cdots,n$为$X$的分量$X_i$和$X_j$的协方差。

五、期望、方差、协方差和相关系数

例1：

	身高X(cm)	体重Y(500g)	X-E(Y)	Y-E(Y)	[X-E(X)]*[Y-E(Y)]
1	152	92	-19.4	-39.7	770.18
2	185	162	13.6	30.3	412.08
3	169	125	-2.4	-6.7	16.08
4	172	118	0.6	-13.7	-8.22
5	174	122	2.6	-9.7	-25.22
6	168	135	-3.4	3.3	-11.22
7	180	168	8.6	36.3	312.18
	E(x)=171.4	E(Y)=131.7			E{[X-E(X)][Y-E(Y)]}=209.4

正相关

根据计算结果，身高和体重是有正相关性的，身高较高的体重一般会比较大，同样体重大的身高一般也比较高。

五、期望、方差、协方差和相关系数

例2：

	游戏时间X(h/天)	学习成绩Y	X-E(Y)	Y-E(Y)	[X-E(X)]*[Y-E(Y)]
1	0	95	-1.36	20.7	-28.152
2	1	65	-0.36	-9.3	3.348
3	3	70	1.64	-4.3	-7.052
4	2	55	0.64	-19.3	-12.352
5	2.5	65	1.14	-9.3	-10.602
6	0.5	80	-0.86	5.7	-4.902
7	0.5	90	-0.86	15.7	-13.502
	E(x)=1.36	E(Y)=74.3			E{[X-E(X)][Y-E(Y)]}=-10.5

负相关

计算结果表明，小朋友玩游戏时间越长，学习成绩越差的可能性越大。

3. 相关系数：

协方差仅能进行定性的分析，并不能进行定量的分析，因此我们引出相关系数的定义：

$$Corr(X,Y)=\frac{Cov(X,Y)}{\sqrt{Var(X)}\cdot\sqrt{Var(Y)}}=\frac{Cov(X,Y)}{\sigma_x\sigma_y}\in[-1,1]$$

其中：$Var(X)$为X的方差，$Var(Y)$为Y的方差。

① $Corr(X,Y)=1$，两个随机变量完全正相关，即$Y=ax+b,a>0$;

② $Corr(X,Y)=-1$，两个随机变量完全负相关，即$Y=-ax+b,a>0$；

③ $0<|Corr(X,Y)|<1$，连个随机变量具有一定程度的线性关系。

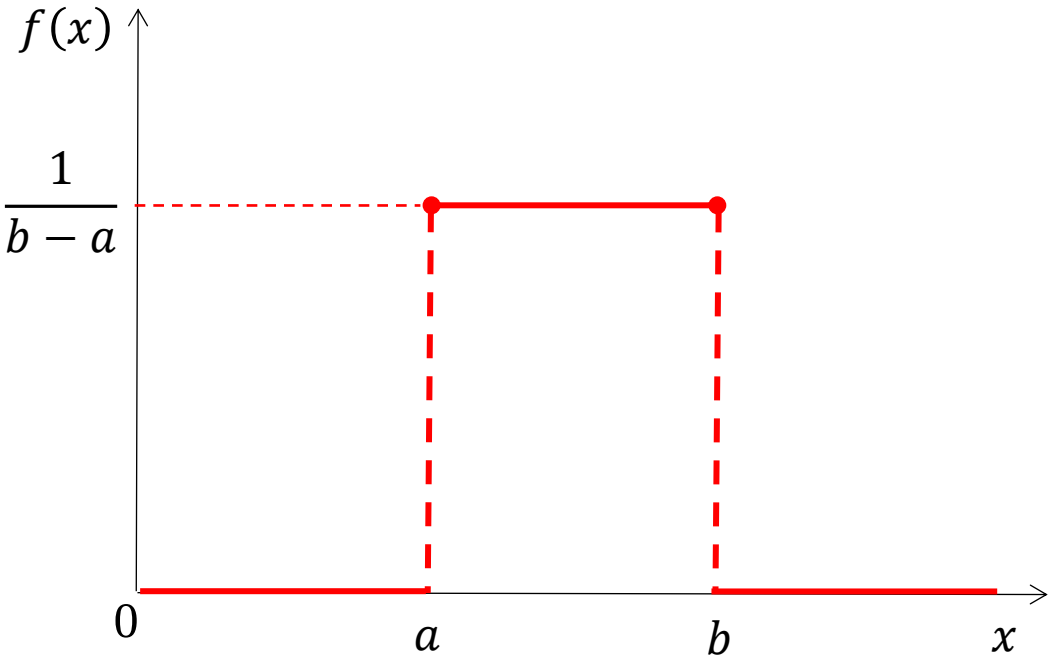
注意，仅可判断线性相关，无法得出非线性相关关系。

1. 0-1分布

0-1分布是单个二值型离散随机变量的分布，其概率分布函数为：

$$f(x)=\frac{1}{b-a}, \qquad a<x<b$$

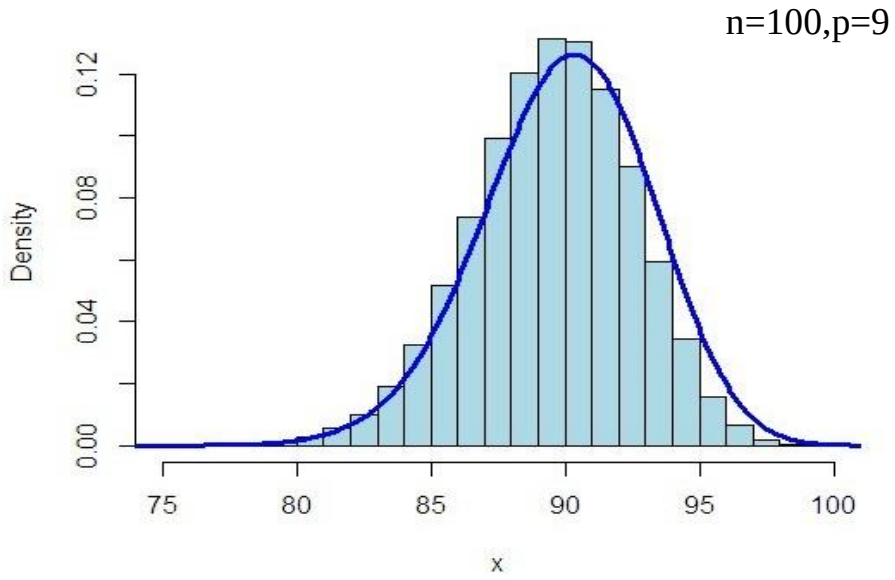
$$f(x)=0, \qquad else$$



2. 二项分布

即重复n次伯努利试验，各次试验相互独立且只有两种可能的结果，且两种结果发生与否相互对立。每次试验时，事件发生的概率为p，不发生的概率为$1-p$，则n次重复独立试验中发生k次的概率为：

$$P(X=k)=c_n^k p^k (1-p)^{n-k}$$



六、常见分布函数、中心极限定理

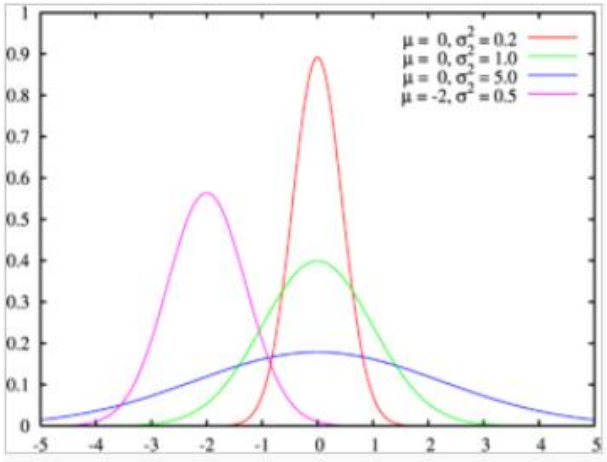
3. 几何分布

几何分布是离散型概率分布，其定义为：在n次伯努利试验中，试验k次才得到第一次成功的机率。即： $P(X=k)=p(1-p)^{k-1}$

4. 高斯分布

若随机变量X服从一个数学期望为μ，方差为σ^2的正态分布，则我们将其记为$N(\mu,\sigma^2)$，高斯分布又叫正态分布，其概率分布函数为：

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



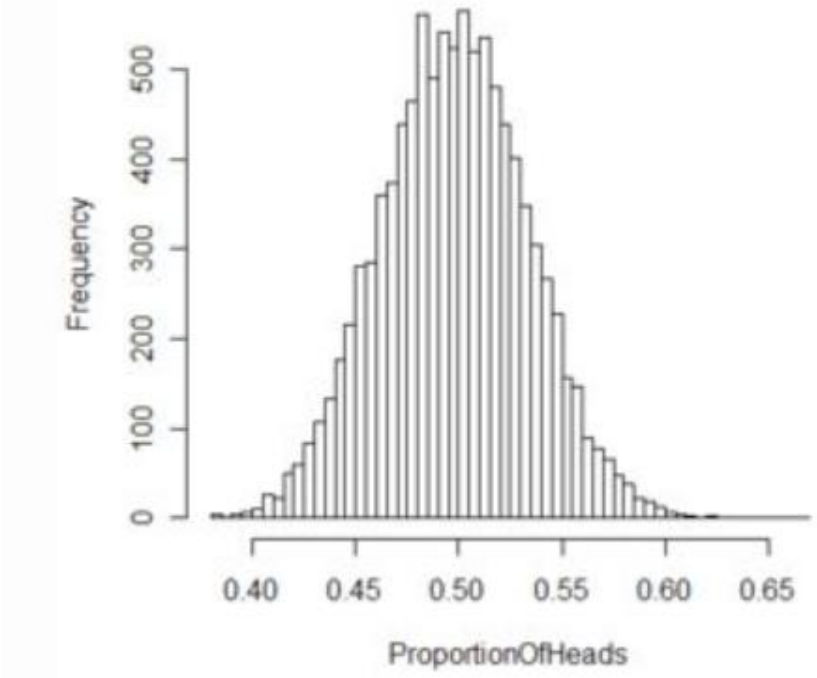
σ 大，数据分布分散，曲线平滑；

σ 小，数据分布集中，曲线陡峭。

六、常见分布函数、中心极限定理

5. 中心极限定理：

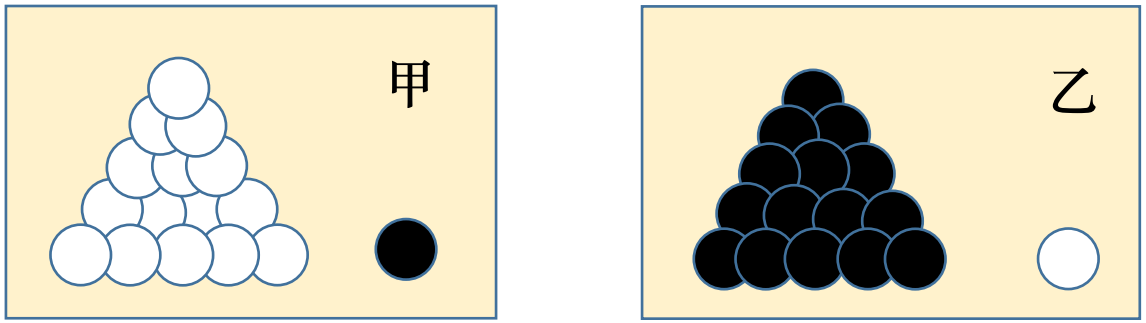
中心极限定理：大量相互独立的随机变量，其均值的分布以正态分布为极限。这组定理是数理统计学和误差分析的理论基础，指出了大量随机变量之和近似服从正态分布的条件。



图中描绘了多次抛掷硬币实验中出现正面的平均比率，每次实验均抛掷了大量硬币。我们就可以发现是符合高斯分布的。

七、极大似然估计（MLE）

极大似然估计的原理，用一张图片来说明，如下图所示：



例：有两个外形完全相同的箱子，甲箱中有99只白球，1只黑球；乙箱中有99只黑球，1只白球。一次试验取出一球，结果取出的是黑球。

问：黑球从哪个箱子中取出？

人们的第一印象就是：“此黑球最像是从乙箱中取出的”，这个推断符合人们的经验事实。“最像”就是“极大似然估计”之意。

七、极大似然估计（MLE）

极大似然估计：建立在极大似然原理的基础上的一个统计方法。极大似然估计提供了一种给定观察数据来评估模型参数的方法，即：“模型已定，参数未知”。

原理：利用若干次试验结果得到某个参数值能够使样本出现的概率最大，则称为极大似然估计。

由于样本集中的样本都是独立同分布的，可以只考虑一类样本集D，来估计参数向量θ，记已知的样本集为：

$$D = \{x_1, \cdots, x_N\}$$

似然函数(likelihood function):联合概率密度函数$p(D|\theta)$称为相对于$\{x_1, \cdots, x_N\}$的θ似然函数：

$$l(\theta) = p(D|\theta) = p(x_1, \cdots, x_N|\theta) = \prod_{i=1}^{N} p(x_i|\theta)$$

七、极大似然估计（MLE）

如果$\hat{\theta}$是参数空间中能使似然函数$l(\theta)$最大的θ值，则$\hat{\theta}$应该是“最可能”的参数值，那么$\hat{\theta}$就是极大似然估计量。它是样本集的函数，记作：

$$\hat{\theta} = d(x_1, \cdots, x_N) = d(D)$$

$\hat{\theta}(x_1, \cdots, x_N)$称作极大似然函数估计值。

求解极大似然函数：

ML估计：求使得出现该组样本的概率最大的θ值。

$$H(\theta) = \ln l(\theta)$$

$$\hat{\theta} = \arg\max_{\theta} l(\theta) = \arg\max_{\theta} \prod_{i=1}^{N} p(x_i|\theta)$$

实际中为了便于分析，定义了对数似然函数：

$$\hat{\theta} = \arg\max_{\theta} H(\theta) = \arg\max_{\theta} \ln l(\theta) = \arg\max_{\theta} \sum_{i=1}^{N} \ln p(x_i|\theta)$$

t 检验和 F检验的由来:

一般而言，为了确定从样本(sample)统计结果推论至总体时所犯错的概率，我们会利用统计学家所开发的一些统计方法，进行统计检验。

通过把所得到的统计检验值，与统计学家建立的一些随机变量的概率分布(probability distribution)进行比较，我们可以知道在百分之多少的机会下会得到目前的结果。

t检验和F检验的由来:

若经比较后发现，出现这结果的机率很少，那我们便可以有信心的说，这不是巧合，是具有统计学上的意义的(用统计学的话讲，就是能够拒绝虚无假设null hypothesis)。

F 值和 t 值就是这些统计检验值，与它们相对应的概率分布，就是F分布和t分布。统计显著性（sig）就是出现目前样本所呈现结果的机率。

1. t检验

当总体呈正态分布，如果总体标准差未知，那么这时一切可能的样本平均数与总体平均数的离差统计量呈t分布。

t 检验用来比较两个平均数的差异是否显著。单总体t检验是检验一个样本平均数与一已知的总体平均数的差异是否显著。当样本容量$n < 30$,检验统计量为：

$$t = \frac{\bar{X} - \mu}{\frac{\sigma_X}{\sqrt{n-1}}}$$

当样本容量n>30，检验统计量为：

$$t = \frac{\bar{X} - \mu}{\frac{\sigma_X}{\sqrt{n}}}$$

t为样本平均数与总体平均数的离差统计量：$\bar{X}$为样本平均数；σ_X为样本标准差；n为样本容量。

例：某校二年级学生期中英语考试成绩，其平均分数为73分，标准差为17分，期末考试后，随机抽取 20人的英语成绩，其平均分数为79.2分。问二年级学生的英语成绩是否有显著性进步?

检验步骤如下:

第一步　建立原假设 $H_0: \mu = 73$

第二步　计算t值

$$t = \frac{\bar{X} - \mu}{\frac{\sigma_X}{\sqrt{n-1}}} = \frac{79.2 - 73}{\frac{17}{\sqrt{19}}} = 1.63$$

第三步　判断

以0.05为显著性水平，df $= n - 1 = 19$，查t值可得$t(19)_{0.05} =$ 2.093而样本离差的t=1.63小与临界值2.093。所以接受原假设。

2. F检验

F检验法是英国统计学家Fisher提出的，主要通过比较两组数据的方差，以确定他们的精密度是否有显著性差异。

- 假设一系列服从正态分布的母体，都有相同的标准差。这是最典型的F检验，该检验在方差分析（ANOVA）中也非常重要。
- 假设一个回归模型很好地符合其数据集要求。

$$S^2 = \sum \frac{(x - \bar{x})^2}{(n-1)}$$

两组数据可以得到两个S^2的值，从而得到：$F = \frac{{S_1}^2}{{S_2}^2}$

将F值与查表得到的F值相比较，若 $F > F_{表}$ ，则两组数据存在显著性差异。

八、机器学习中的统计检验

置信度95%时F值（单边）

f大 f小	2	3	4	5	6	7	8	9	10	∞
2	19.0	19.16	19.25	19.30	19.33	19.36	19.37	19.38	19.39	19.5
3	9.55	9.28	9.12	9.01	8.94	8.88	8.84	8.81	8.78	8.53
4	6.94	6.59	6.39	6.26	6.16	6.09	6.04	6.00	5.96	5.63
5	5.79	5.41	5.19	5.05	4.95	4.88	4.82	4.78	4.74	4.36
6	5.14	4.76	4.53	4.39	4.28	4.21	4.51	4.10	4.06	3.67
7	4.74	4.35	4.12	3.97	3.87	3.79	3.73	3.68	3.63	3.23
8	4.46	4.07	3.84	3.69	3.58	3.50	3.44	3.39	3.34	2.93
9	4.26	3.86	3.63	3.48	3.37	3.29	3.23	3.18	3.13	2.71
10	4.10	3.71	3.48	3.33	3.22	3.14	3.07	3.02	2.97	2.54
∞	3.00	3.60	2.37	3.21	2.10	2.01	1.94	1.88	1.83	1.00

横向为大方差数据的自由度；纵向为小方差数据的自由度。

3. Friedman 检验

该检验在机器学习中被广泛采用。

其应用背景为：作者提出的算法在一部分数据集上表现好，另外一部分数据集上表现不好，那么总体来看究竟新提出的算法与已有算法是否存在显著性差异?

八、机器学习中的统计检验

①. 为每种算法打分：

Average rank on classification accuracy of five algorithms.

Dataset	RLLSVM	THSVM	SSLM	MMTSVM	RM^{3}TSM
Ionosphere	2	4	5	3	1
Spectf heart	3	4	5	2	1
Breast cancer	1	5	3	2	4
Ecoli-0-1-3-7-vs-2-6	1	3.5	5	3.5	2
Ecoli(0-1-4-6-vs-5)	3	4	2	1	5
Winequality-white-9-vs-4	1	4	5	3	2
Liver Disorder	2	3	5	4	1
Pima	3	2	4	5	1
Cleveland-0-vs-4	2	4	5	3	1
Glass4	2	4	1	3	5
Glass	2	3	4	5	1
Fertility	4	5	1	3	2
DBworld	3	1	5	4	2
LSVT	3	2	5	4	1
Average rank	2.2857	3.4643	3.9286	3.25	2.0714

根据精度结果为每种算法排序，共5个算法，在每一个数据集上，表现最好的算法得分为1，表现最差的算法得分为5。最后求每种算法在所有数据集上的平均得分；

②. 原假设：H_0:文中所提算法与已有算法无显著性差异。

③. 计算：

$$\chi_F{}^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j{}^2 - \frac{k(k+1)^2}{4}\right] = 17.09$$

其中，N是数据集的个数，k是算法的个数；$R_j = \frac{1}{N}\sum_i r_i{}^j$，$r_i{}^j$就是表中第$j$个算法在第$i$个数据集上的得分，进一步可得：

$$F_F = \frac{(N-1)\chi_F{}^2}{N(k-1) - \chi_F{}^2} = 5.71$$

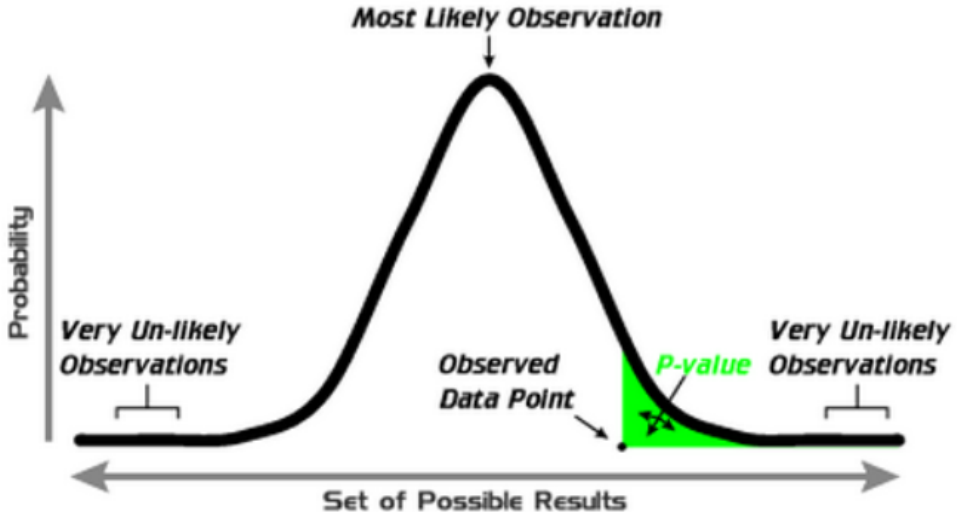
④. 比较得出结论：

与自由度为$k-1$和$(k-1)(N-1)$的F分布的表格值比较。得到的F_F如果大于表中的值，则提出的算法较原有的算法具有显著性差异，拒绝原假设。

$$F_F = 5.71 > F(4.52) = 2.55$$

4. p-value

p值是一种概率：在原假设为真的前提下，出现该结果或比该结果更极端的结果的概率之和。



- 如果 p值<0.01，说明是较强的判定结果，拒绝假定的参数取值。
- 如果 0.01<p值<0.05，说明是较弱的判定结果，拒绝假定的参数取值。
- 如果 p值>0.05，说明结果更倾向于接受假定的参数取值。

4. p-value

以抛一枚质地均匀的硬币为例：

在一次实验中，共扔硬币20次，其中出现了14次正面，我们是否能从这个结果中推断出出现正面的概率为 $\frac{1}{2}$ 呢?

① 原假设：H_0:出现正面的概率是$\frac{1}{2}$

② 计算 p 值：$p = \frac{1}{2^{20}}\left[\binom{20}{14} + \binom{20}{15} + \cdots + \binom{20}{20}\right] = \frac{60460}{1048576} \approx 0.058$

③ 得出结论：因为 $p > 0.05$，所以接受原假设，出现正面的概率是 $\frac{1}{2}$.

中国农业大学
China Agricultural University

谢　谢！