特征提取

肖雄

维数灾难（Curse of dimensionality)

百度智能云

又称“维度灾难”，随着维数的增加，问题的复杂性呈指数级增长的现象。

维度（数）：特征的数量（或自由度）

问题的复杂性：计算代价

1961年，美国数学家Richard Bellman在研究**动态规划问题**时，首次提出。

维数灾难，是很多问题困难的根本来源，例如物理中的量子多体问题、蛋白质折叠等科学研究领域的问题。

Richard Bellman
（1920-1984）

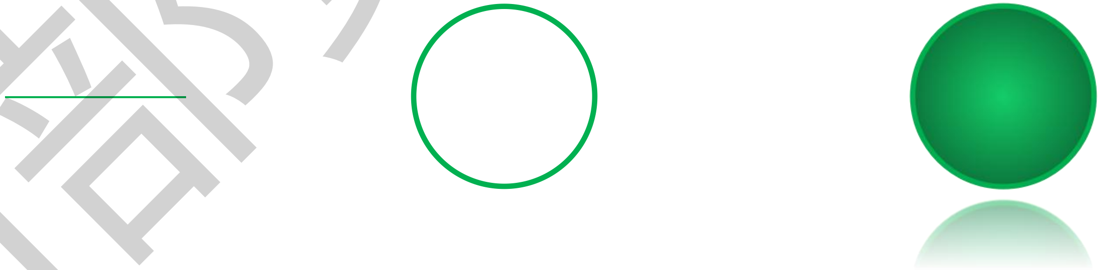
强化学习中的Bellman方程，就是由Bellman发现。

维数灾难（Curse of dimensionality）

在机器学习中，当我们把日常**三维世界**中的认识和规律迁移到**高维世界**中的时候，通常会带来很多意想不到的麻烦，机器学习中将这个现象称之为维度灾难（Curse of Dimensionality）

示例：单位球体积

1维	2维	3维	高维

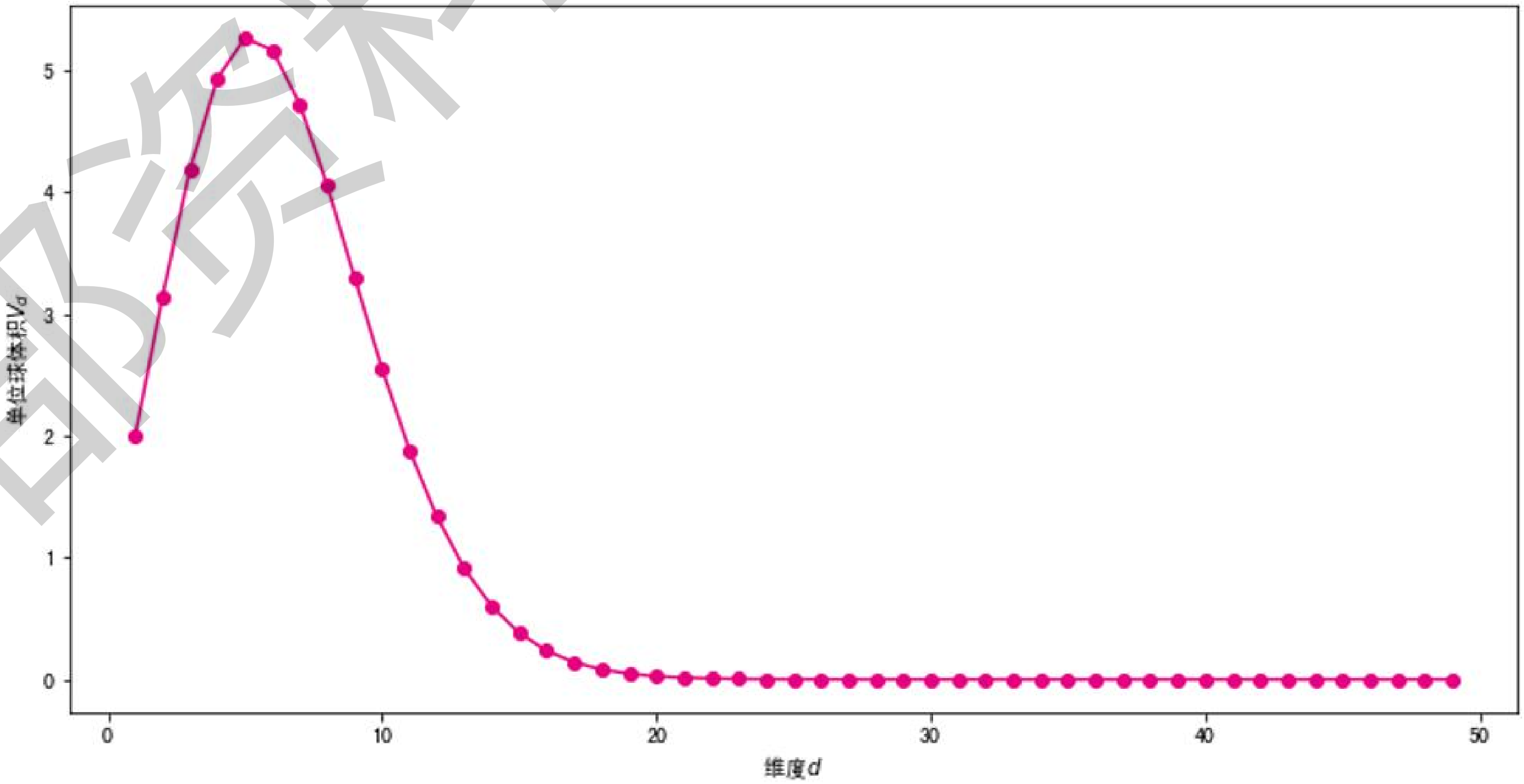


?

2	πr^2	$4/3\pi r^3$	

维数灾难（Curse of dimensionality）

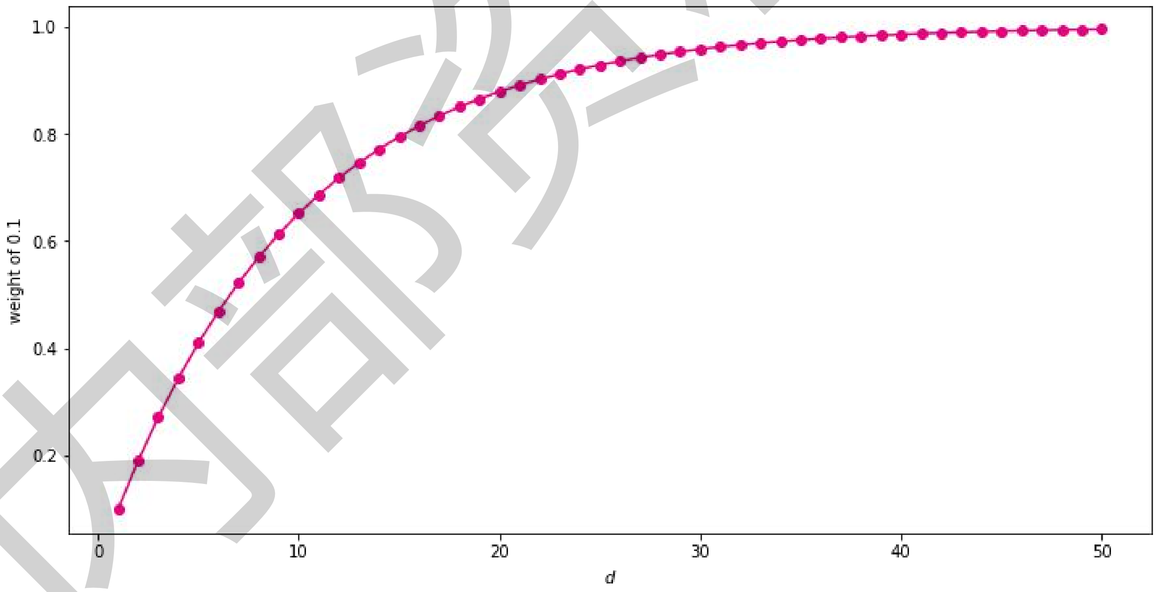
d 维空间半径为 r 的球体体积公式 $$V(d,r)=\frac{\pi^{d/2}}{\Gamma(d/2+1)}r^d$$ **Gamma函数**



高维空间反直觉现象，也就是说我们对高维空间理解有限。

维数灾难（Curse of dimensionality）

高维空间中，球体内部的体积与表面积处的体积相比可以忽略不计



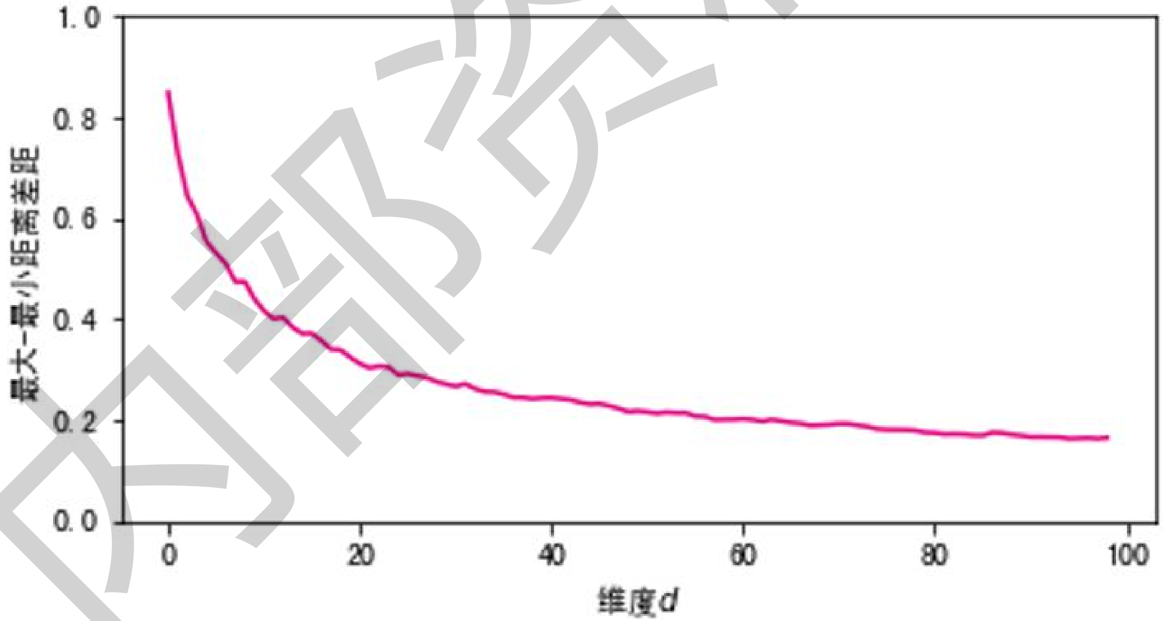
高维空间中，球体的大部分体积分布在球体的边界

维数灾难（Curse of dimensionality）

百度智能云

d 维空间样本 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 的欧式距离为：$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sqrt{\sum_{i=1}^{d}(x_{1i} - x_{2i})^2}$

随着维数的增加，单个维度对距离的影响越来越小，任意样本间的距离趋于相同

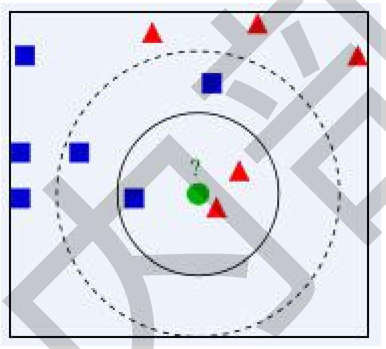


- 随机生成100个维度；
- 每个维度都从0到1均匀分布采样，得到一些点；
- 计算两两之间的最小距离和最大距离；
- 发现：所有样本之间的距离都是差不多的，而且是很远。

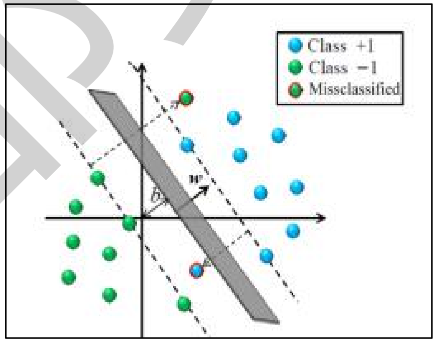
高维空间中，欧氏距离不是那么有效

维数灾难（Curse of dimensionality）

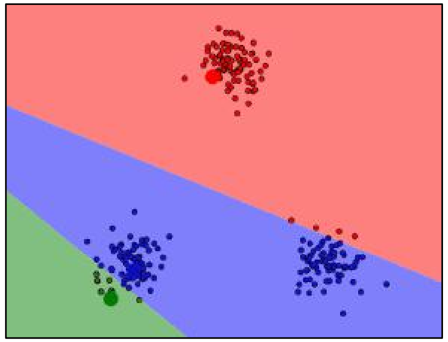
- K近邻：样本间距离
- 支持向量机（SVM）：样本到决策面的距离
- K-means：样本到聚类中心的距离
- 凝聚层次聚类：不同簇之间的距离
- 推荐系统：商品或用户（表示成高维向量）相似度



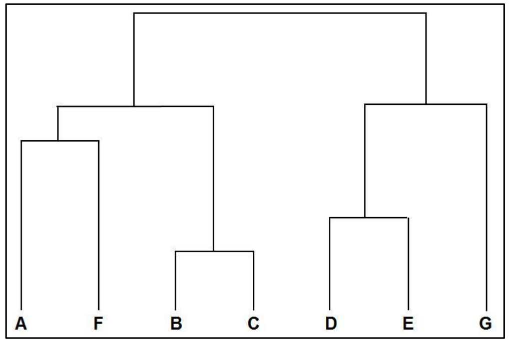
K近邻



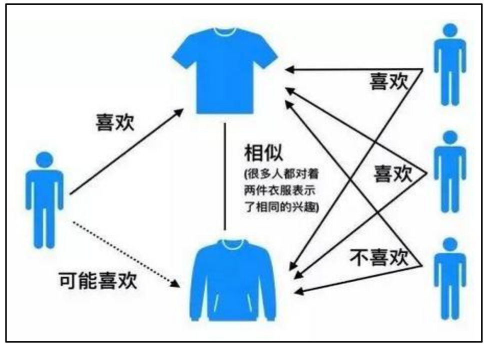
SVM



K-means



凝聚层次

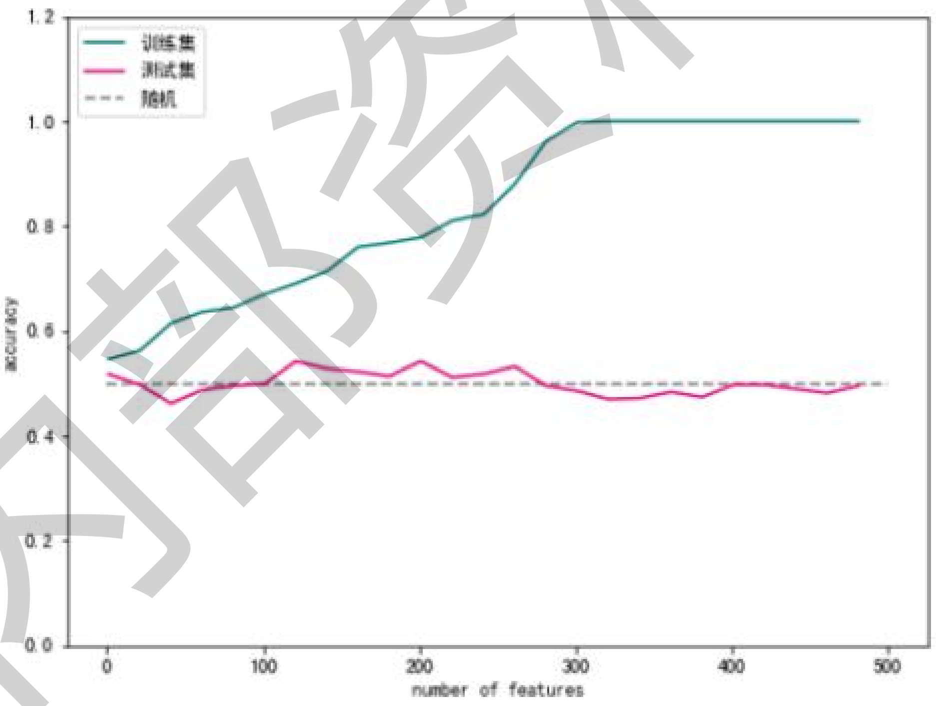


推荐系统

一系列机器学习模型就会出现问题

稀疏性与过度拟合

- 过拟合：模型对已知数据拟合较好，对新的数据拟合较差。
- 高维空间中，样本变得极度稀疏，容易造过度拟合问题。



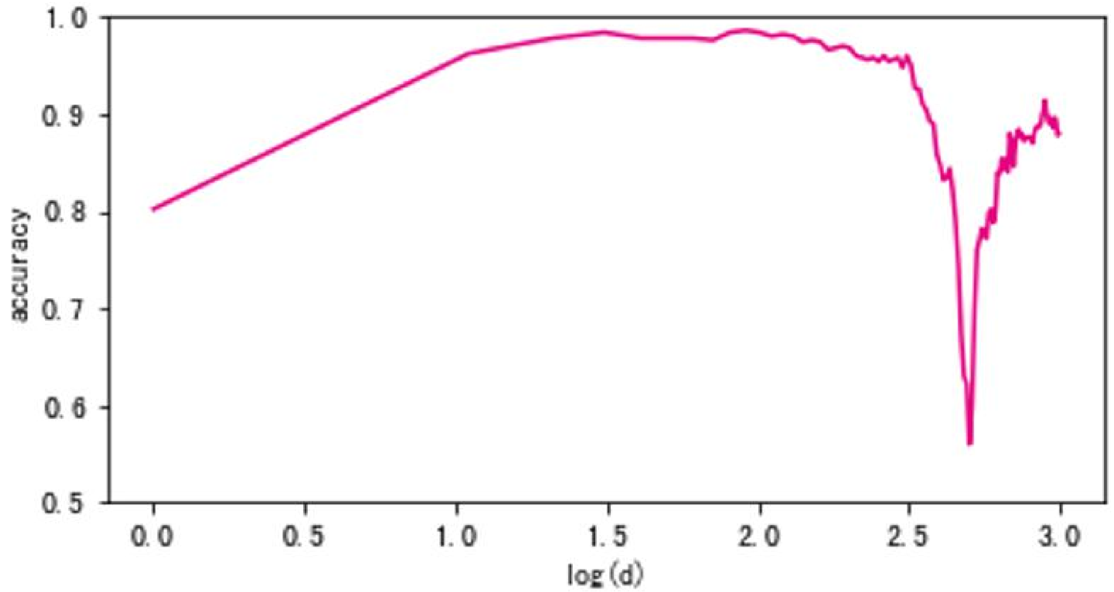
在一个随机的分类数据集（满足高斯分布,标签为0和1，属于2分类）上，随着维度的增大，模型不断拟合误差，训练集准确率不断上升，测试集正确率很低。

测试集上的准确度，只达到0.5，和随机猜测无区别

Hughes现象

1968年，Hughes（休斯）发现：分类问题中，在训练集固定时，随着维度的增大，**分类器**性能不断提升直到达到最佳维度，继续增加维度分类器性能会下降

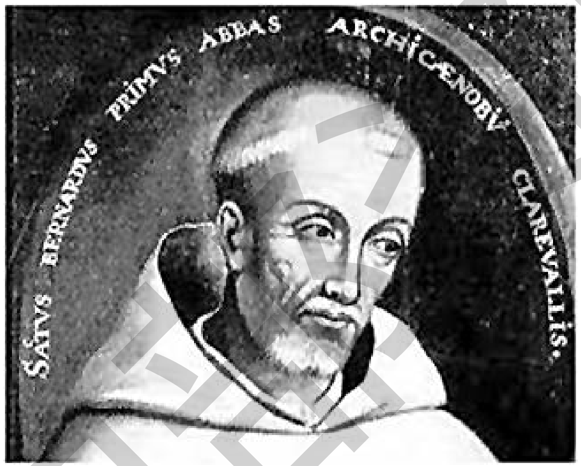
Gordon F. Hughes



维度的增加，不仅会带来信息，也会带来噪音，噪音大于信息时，准确度下降

如何应对维度灾难？

奥卡姆提剃刀：**“如无必要，勿增实体”**。即“简单有效原理”，切勿浪费较多东西去做，用较少的东西，同样可以做好的事情。



William of Occam
欧洲哲学家、神学家
(1285~1349，死于欧洲黑死病)

机器学习的应用：
在能够获得较好拟合效果的前提下，尽量使用较为简单的模型，即不使用那么多的维度。

Entities should not be multiplied unnecessarily

如何应对维度灾难?

- 选择小维度的方法
  - 特征提取
    - 线性降维：PCA，LDA，MDS等
    - 非线性降维
      - 核方法：KPCA，KFDA等
      - 流形学习：Isomap，LLE，LPP等
  - 特征选择

Feature Selection：选取特征子集。

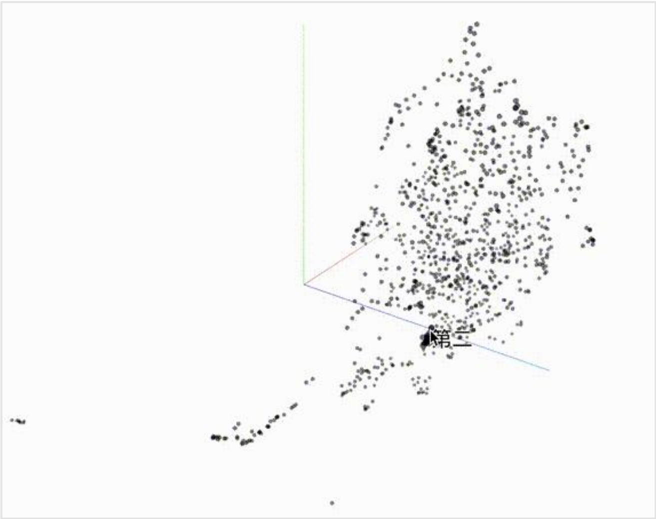
Dimensionality reduction：从原始特征组合线性或非线性关系，将高维数据转换为低维数据。

特征提取举例

三维空间，两个维度即可确定

通信数据
入网时间
套餐价格
每月话费
每月流量
每月通话时长
欠费月数
欠费金额

内在维度
用户忠诚度
消费能力
欠费指数



数据可视化

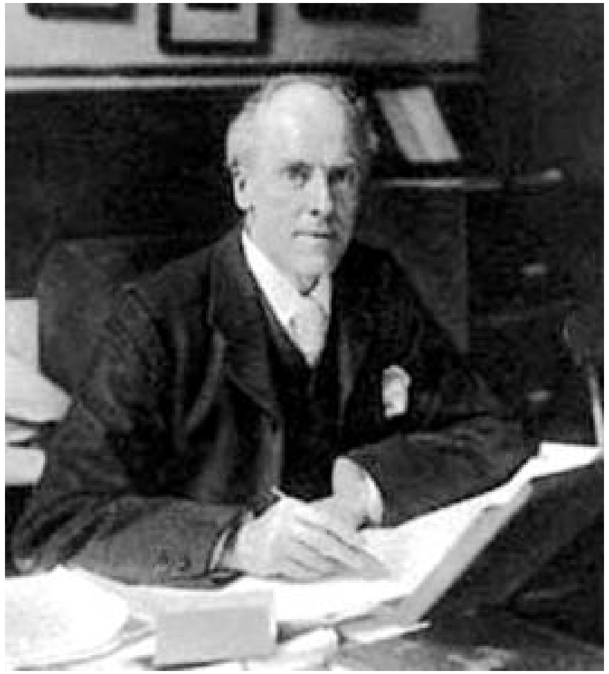
很多高维数据的内在维度，其实较低

特征提取举例

PCA，Principal Component Analys，主成分分析，1901年由Karl Pearson提出。

在人脸识别，图像压缩等领域得到了广泛的应用。

基本思想：构造一系列原始特征的**线性组合**形成的线性无关低维特征，以去除数据的相关性，并使降维后的数据（目的）最大程度的保持原始高维数据的**方差信息**。



Karl Pearson

英国（1857～1936），公认为统计学之父

知识回顾（内积）

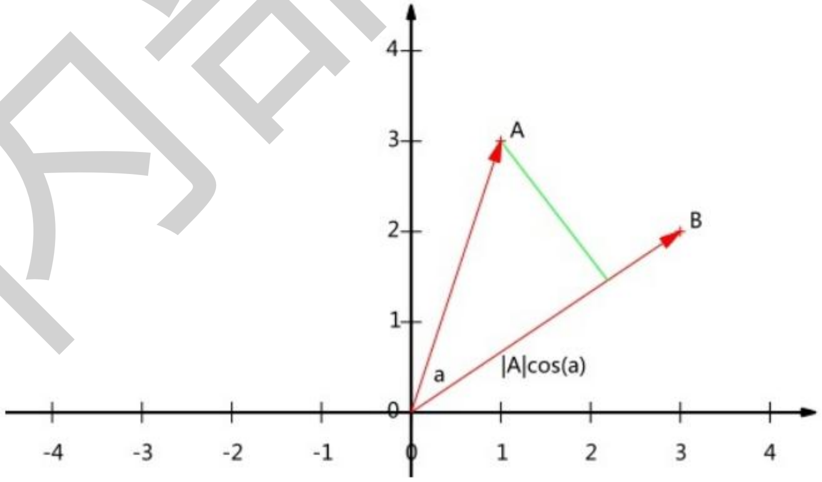
两个向量的 A 和 B 内积我们知道形式是这样的：

$$(a_1, a_2, \cdots, a_n) \cdot (b_1, b_2, \cdots, b_n)^\top = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

将两个向量映射为实数

为了简单起见，我们假设 A 和 B 均为二维向量，从几何角度来分析

$A = (x_1, y_1)$， $B = (x_2, y_2)$ $A \cdot B = |A||B|cos(\alpha)$



A 与 B 的内积等于 A 到 B 的投影长度乘以 B 的模

假设 B 的模为 1 $A \cdot B = |A|cos(a)$

知识回顾（基）

百度智能云

已知：**向量 (3,2)**

隐含前提：**以 x 轴和 y 轴上正方向长度为 1 的向量为标准 (1,0),(0,1)**

向量 (3,2) 实际是说在 x 轴投影为 3 而 y 轴的投影为 2

要准确描述向量：

- 首先要确定一组基（模长为1，**一般为正交基**）
- 然后给出在基所在的各个直线上的**投影值**

知识回顾（基变换的矩阵表示）

$$(1,0),(0,1)$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

$$\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right) \quad \left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$$

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} \frac{5}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$$

知识回顾（基变换的矩阵表示）

推广一下，如果有 m 个二维向量，只要将二维向量按列排成一个两行 m 列矩阵，然后用“基矩阵”乘以这个矩阵就可以得到了所有这些向量在新基下的值。

$$\begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 2/\sqrt{2} & 4/\sqrt{2} & 6/\sqrt{2} \\ 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{pmatrix} \begin{pmatrix} a_1 & a_2 & \cdots & a_M \end{pmatrix} = \begin{pmatrix} p_1a_1 & p_1a_2 & \cdots & p_1a_M \\ p_2a_1 & p_2a_2 & \cdots & p_2a_M \\ \vdots & \vdots & \ddots & \vdots \\ p_Ra_1 & p_Ra_2 & \cdots & p_Ra_M \end{pmatrix}$$

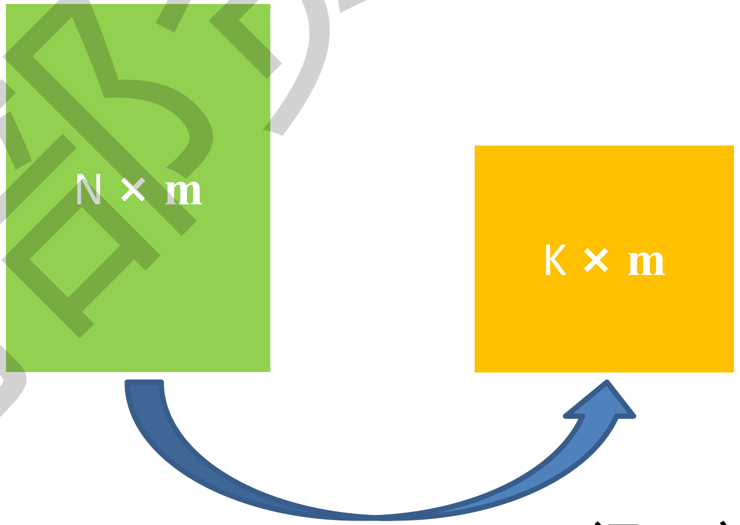
两个矩阵相乘的意义：
将右边矩阵中的每一列向量 a_i 变换到左边矩阵中以每一行行向量为基所表示的空间中去

知识回顾（最大可分性）

选择不同的基可以对同样一组数据给出不同的表示；

以上例子中，我们发现：基的数量==向量的维数；

如果**基的数量少于向量本身的维数**，则可以达到降维的效果。



问：应该如何选择 K 个基？才能最大程度保留原有的信息

答：投影后的投影值尽可能分散，因为重叠就会有样本消失

知识回顾（方差）

数值的分散程度，可以用数学上的方差来表述

$$Var(a) = \frac{1}{m}\sum_{i=1}^{m}(a_i - \mu)^2$$

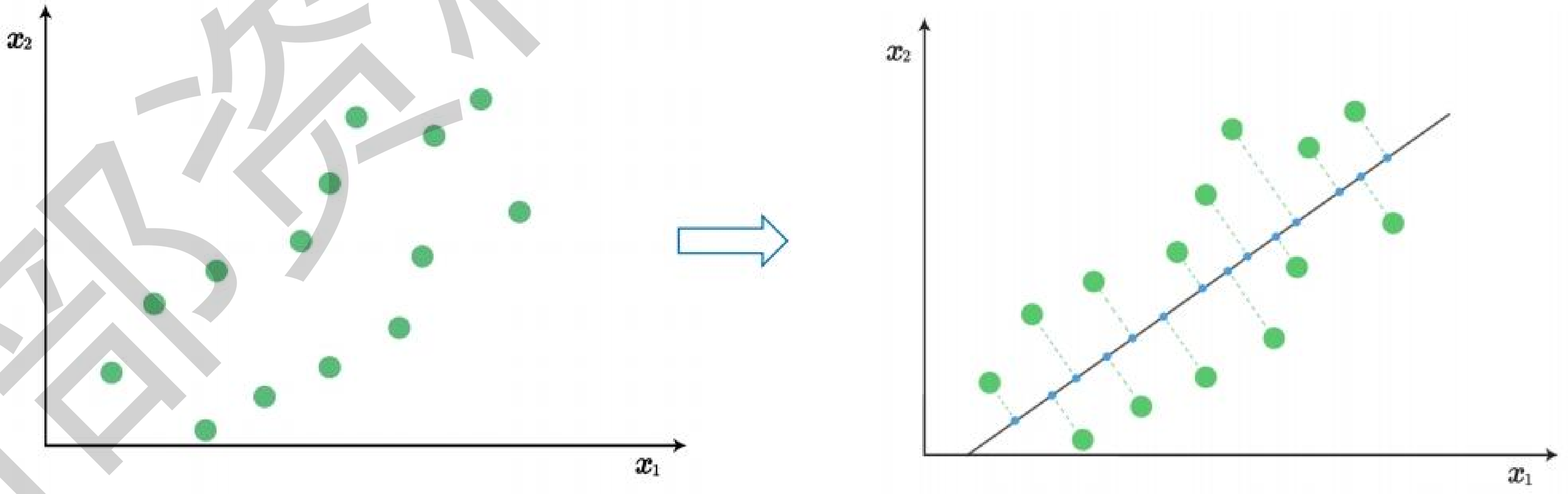
$$Var(a) = \frac{1}{m}\sum_{i=1}^{m}a_i^2$$

去中心化，均值都化为 0

对于2维向量，寻找一个一维基，使得所有数据变换为这个基上的坐标表示后，方差值最大

知识回顾（方差）

如何将左下图中的二维数据投影到一维，使得数据的方差最大化地保留？



协方差

协方差可以表示两个变量的相关性

$$Cov(a,b)=\frac{1}{m-1}\sum_{i=1}^{m}(a_i-\mu_a)(b_i-\mu_b)$$

$$Cov(a,b)=\frac{1}{m}\sum_{i=1}^{m}a_ib_i$$

去中心化，均值都化为 0

当样本数较大时，不必在意 m 还是 m-1

当协方差为 0 时，表示两个变量线性不相关，此时两个基是正交的

将一组 N 维向量降为 K 维，其目标是选择 K 个单位正交基，使得原始数据变换到这组基上后，各变量两两间协方差为 0，而变量方差则尽可能大，即“在正交的约束下，取最大的 K 个方差”

方差与协方差

变量内**方差**及变量间**协方差**

假设我们只有 a 和 b 两个变量，那么我们将它们按行组成矩阵 X：

$$X = \begin{pmatrix} a_1 & a_2 & \cdots & a_m \\ b_1 & b_2 & \cdots & b_m \end{pmatrix}$$

对角线元素：两个变量的方差

其他元素：a 和 b 的协方差

$$\frac{1}{m}XX^{\mathsf{T}} = \begin{pmatrix} \frac{1}{m}\sum_{i=1}^{m} a_i^2 & \frac{1}{m}\sum_{i=1}^{m} a_i b_i \\ \frac{1}{m}\sum_{i=1}^{m} a_i b_i & \frac{1}{m}\sum_{i=1}^{m} b_i^2 \end{pmatrix} = \begin{pmatrix} Cov(a,a) & Cov(a,b) \\ Cov(b,a) & Cov(b,b) \end{pmatrix}$$

设我们有 m 个 n 维数据记录，将其排列成矩阵 $X_{n,\ m}$，设 $C = \frac{1}{m}XX^T$ ，则 C 是一个对称矩阵，其对角线分别对应各个变量的方差，而第 i 行 j 列和 j 行 i 列元素相同，表示 i 和 j 两个变量的协方差。

优化目标

我们需要将：

- **除对角线外的其它元素化为 0**
- **在对角线上将元素按大小从上到下排列（变量方差尽可能大）**

假设：

X 对应的协方差矩阵为 C

P 是一组**基按行**组成的矩阵，Y=PX

设 Y 的协方差矩阵为 D

D 与 C 的关系如右图

$$
\begin{aligned}
D &= \frac{1}{m} Y Y^{T} \\
&= \frac{1}{m}(PX)(PX)^{T} \\
&= \frac{1}{m} P X X^{T} P^{T} \\
&= P\left(\frac{1}{m} X X^{T}\right) P^{T} \\
&= P C P^{T}
\end{aligned}
$$

优化目标

协方差矩阵 C具有如下特征：

$\because$ **是一个是实数对称矩阵**

$\therefore$ **一个 n 行 n 列的实对称矩阵一定可以找到 n 个单位正交特征向量**

假设：

这 n 个特征向量为 $e_1, e_2, \cdots, e_n$

并将其按列组成矩阵 $E = (e_1, e_2, \cdots, e_n)$

$$E^T C E = \Lambda = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}$$

则：**右侧等式成立。**

Λ **为对角矩阵，其对角元素为各特征向量对应的特征值**

此时，已经找到了需要的矩阵 P：$P = E^{\mathsf{T}}$

用 P 的前 K 行组成的矩阵P乘以原始数据矩阵 X，就得到了我们需要的降维后的数据矩阵 Y

PCA总结

总结一下 PCA 的算法步骤：

设有 m 条 n 维数据。

1. 将原始数据按列组成 n 行 m 列矩阵 X；
2. 将 X 的每一行进行零均值化，即减去这一行的均值；
3. 求出协方差矩阵 $C = \frac{1}{m} X X^{\mathsf{T}}$ ；
4. 求出协方差矩阵的特征值及对应的特征向量；
5. 将特征向量按对应特征值大小从上到下按行排列成矩阵，取前 k 行组成矩阵 P；
6. $Y = PX$ 即为降维到 k 维后的数据。

$$K \times N \quad \times \quad N \times m \quad = \quad K \times m$$

PCA总结

① 原始数据：m×n维（满足“一行一样本，一列一特征”）；

② 去中心化处理（$X\text{-}X_{mean}$）；

③ 求原始数据的协方差矩阵；

④ 对协方差矩阵做特征值分解，求出k个特征向量组成的转换矩阵W和对应的特征值；

⑤ 将特征向量按照对应的特征值大小，按行排列称矩阵，取前k行组成的矩阵W；

⑥ Y=X·W就是降到k维之后的数据。

$$m \times N \quad \times \quad N \times K \quad = \quad m \times K$$

知识回顾（特征值分解）

百度智能云

numpy.linalg.eig

`linalg.eig(a)` [source]

Compute the eigenvalues and right eigenvectors of a square array.

特征值　特征向量

```
from numpy import linalg as LA
w, v = LA.eig(np.diag((1, 2, 3)))
```

```
array([1., 2., 3.])

array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

LDA vs PCA

另外一种经典的降维方法线性判别分析（Linear Discriminant Analysis，LDA）

LDA：一种**监督学习**的降维技术

功能：可用于降维、分类

目标：投影后类内方差最小，类间方差最大

PCA：是**无监督学习**的降维技术

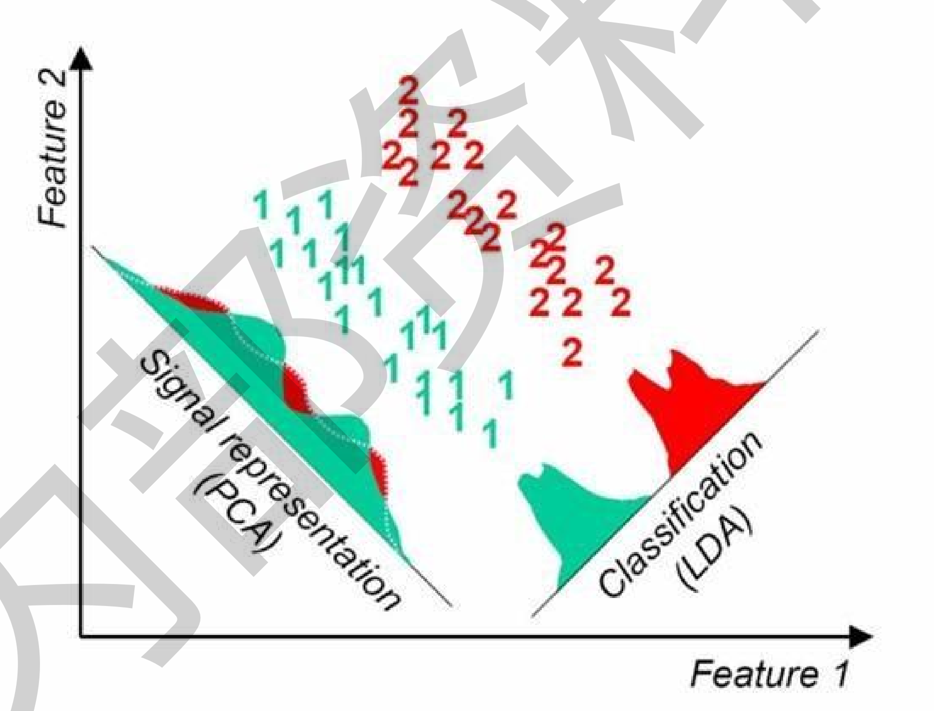
功能：可用于降维

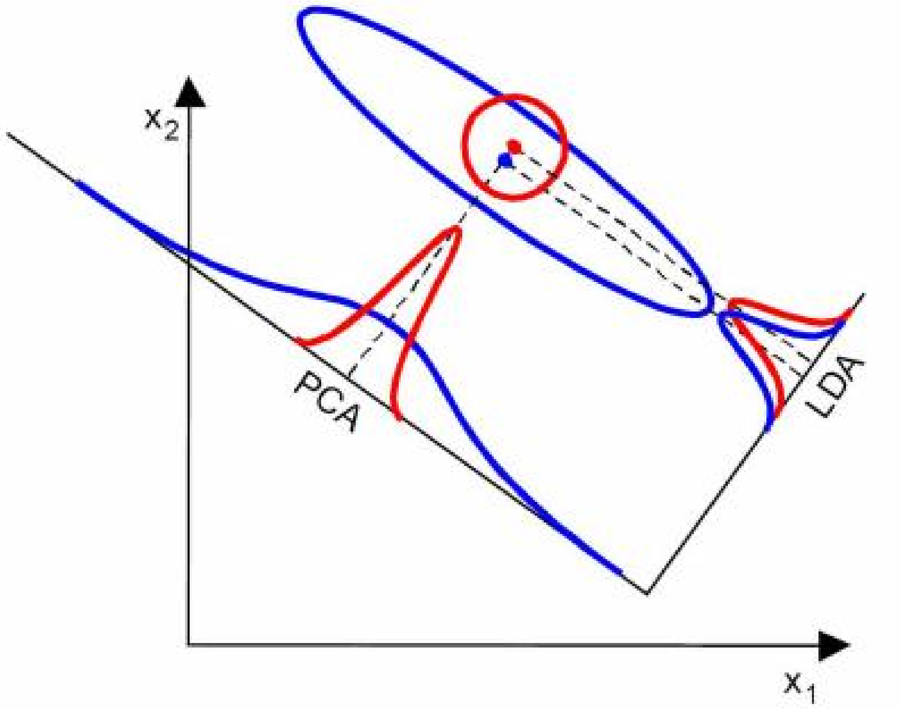
目标：样本点投影具有最大方差的方向

两者都假设数据符合**高斯分布**

两者在降维时均使用了**矩阵特征分解**的思想

LDA vs PCA


Feature 2
Feature 1
Signal representation
(PCA)
Classification
(LDA)


x_2
x_1
PCA
LDA

百度智能云

THANKS