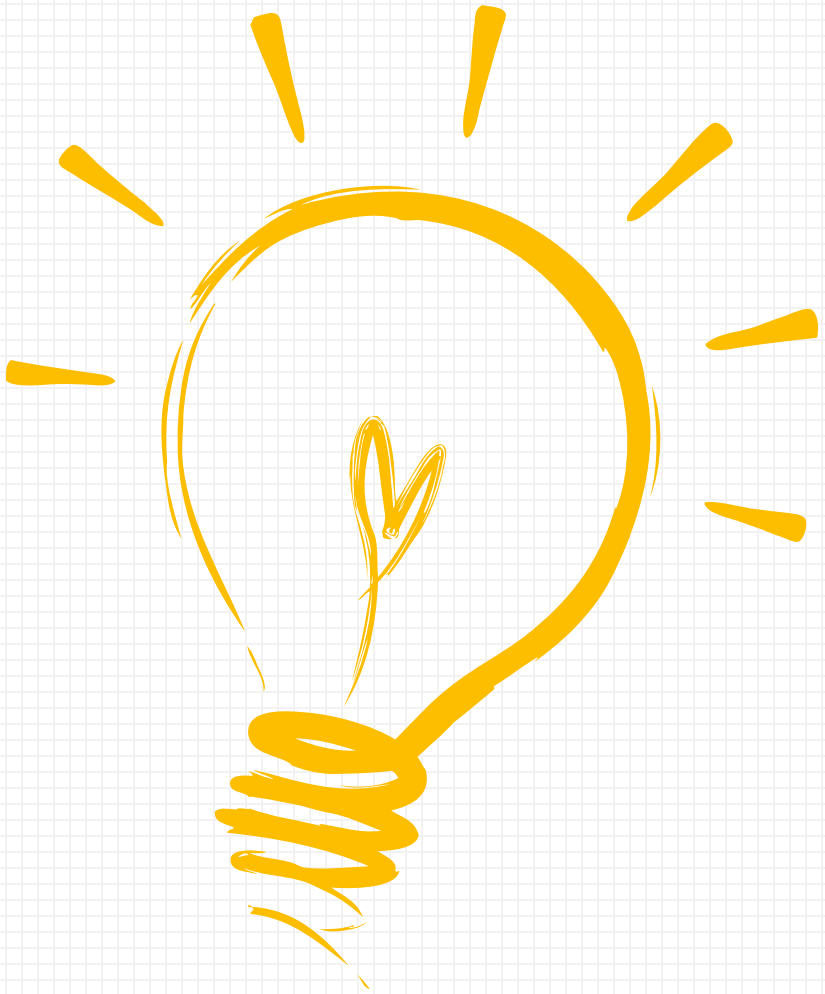
数据挖掘算法

主成分分析(PCA)

CONTENTS



di yi zhang jie

第一章节

算法背景

1.1 随机变量

随机变量: 在实际理解应用中, 那些能够取得多个值的变量我们都可以理解为或者称为一个随机变量.(这里的解释并不是按照准确定义).下表,CRIM,CHAS等都可以叫做随机变量.

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
0.02985	0	2.18	0	0.458	6.43	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9
0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.9	19.15	27.1
0.21124	12.5	7.87	0	0.524	5.631	100	6.0821	5	311	15.2	386.63	29.93	16.5
0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.1	18.9
0.22489	12.5	7.87	0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15
0.11747	12.5	7.87	0	0.524	6.009	82.9	6.2267	5	311	15.2	396.9	13.27	18.9
0.09378	12.5	7.87	0	0.524	5.889	39	5.4509	5	311	15.2	390.5	15.71	21.7
0.62976	0	8.14	0	0.538	5.949	61.8	4.7075	4	307	21	396.9	8.26	20.4

1.2 期望与方差

期望(均值): $\mathbf{E}(\boldsymbol{X})$,在实际理解应用中,样本较多时, 期望用均值计算.

方差: $\mathbf{E}\left([\boldsymbol{X}-\mathbf{E}(\boldsymbol{X})]^{\mathbf{2}}\right)$,在实际理解应用中,样本较多时, S=$\frac{1}{n}\sum(X-\mu)^2$.

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
0.02985	0	2.18	0	0.458	6.43	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9
0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.9	19.15	27.1
0.21124	12.5	7.87	0	0.524	5.631	100	6.0821	5	311	15.2	386.63	29.93	16.5
0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.1	18.9
0.22489	12.5	7.87	0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15
0.11747	12.5	7.87	0	0.524	6.009	82.9	6.2267	5	311	15.2	396.9	13.27	18.9
0.09378	12.5	7.87	0	0.524	5.889	39	5.4509	5	311	15.2	390.5	15.71	21.7
0.62976	0	8.14	0	0.538	5.949	61.8	4.7075	4	307	21	396.9	8.26	20.4

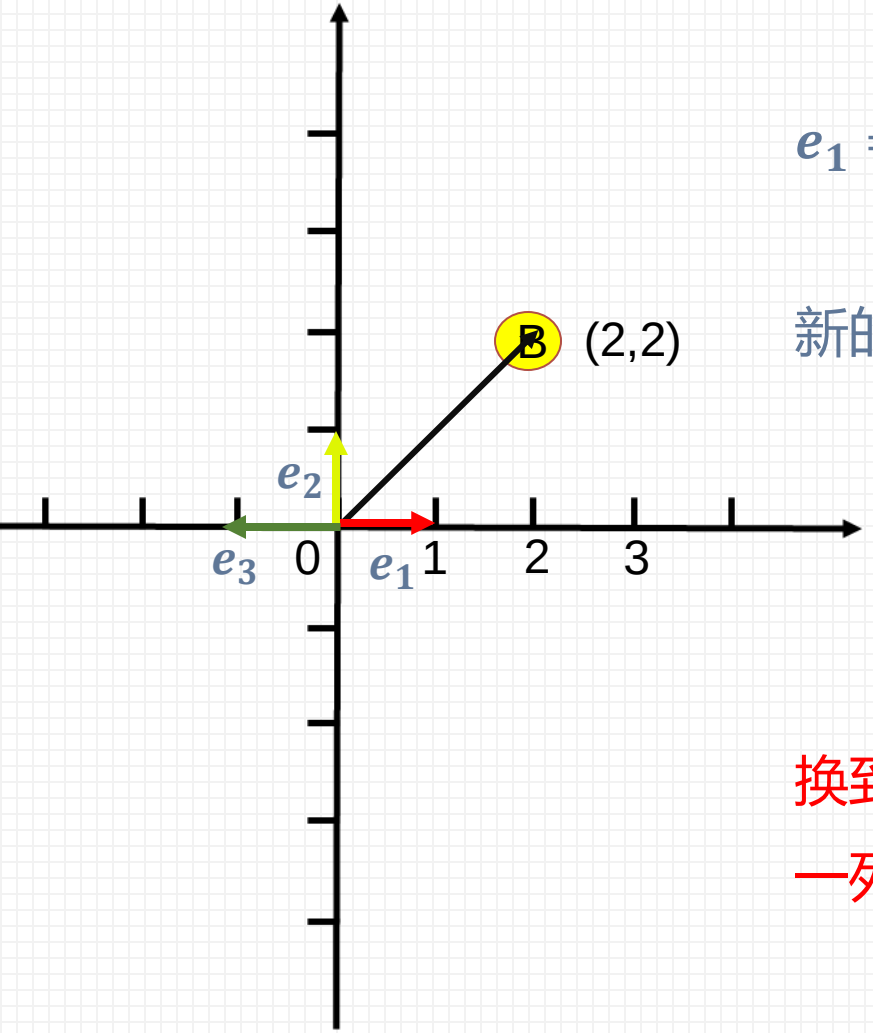
1.3 协方差

协方差: $\mathbf{E}([\boldsymbol{X}-\mathbf{E}(\boldsymbol{X})][\boldsymbol{Y}-\mathbf{E}(\boldsymbol{Y})])$,在实际理解应用中,样本较多时, $S=\frac{1}{n}\sum[(X-\mu)(Y-\nu)]$

就完事了. **描述的是两个随机变量之间的线性关系,相互影响的程度.**

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
0.02985	0	2.18	0	0.458	6.43	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9
0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.9	19.15	27.1
0.21124	12.5	7.87	0	0.524	5.631	100	6.0821	5	311	15.2	386.63	29.93	16.5
0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.1	18.9
0.22489	12.5	7.87	0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15
0.11747	12.5	7.87	0	0.524	6.009	82.9	6.2267	5	311	15.2	396.9	13.27	18.9
0.09378	12.5	7.87	0	0.524	5.889	39	5.4509	5	311	15.2	390.5	15.71	21.7
0.62976	0	8.14	0	0.538	5.949	61.8	4.7075	4	307	21	396.9	8.26	20.4

1.4 矩阵乘法



以二维平面为例,我们常说点B的坐标为(2,2),事实上是指,B向量在基$e_1 = (1, 0)$和基$e_2 = (0, 1)$下的坐标.从而有(2,2) = 2(1,0)+2(0,1).

现在想在二维空间再找另一组基e_2和$e_3 = (-1, 0)$,得到(2,2)和(3,4)在新的一组基下的表示.即找到向量(2,2),(3,4)在e_2和e_3下的坐标.

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ -2 & -3 \end{bmatrix}$$

则(2,2) = 2(0,1)+(-2)(-1,0)　(3,4) = 4(0,1)+(-3) (-1,0)

从这个角度,两个矩阵相乘的意义是将右边矩阵中的每一列列向量变换到左边矩阵中每一行行向量为基所表示的空间去,且矩阵相乘结果的每一列便是原来列向量在新基下的相对应坐标.

注意:上述基向量长度应为1,确切的说左边矩阵的那些行是空间内的标准正交基.否则得到的结果并不是相对应坐标.

di yi zhang jie

第二章节

算法原理

2.1 PCA降维思想

本来某个样本,含有n个变量,即需要n个变量描述.我们想用更少的变量就描述这个样本.

思想:通过将原来的n个变量线性组合得到k个新的,正交的变量,这样一个样本就可以用这k个新的变量表示了.

原来的变量X($x_1, x_2, x_3 \dots\dots x_n$): 用n个n维空间的标准正交基向量表示 -> 一组基

现在的变量: 用k个n维空间的标准正交基向量 -> 另一组基

假设我们已经从n维空间,经过对原来n个变量线性组合得到了新的k个标准正交基为{$p_1, p_2, p_3 \dots\dots p_k$},每个$p_i$是一个行向量.那么原变量如何用新的K个基表示呢?

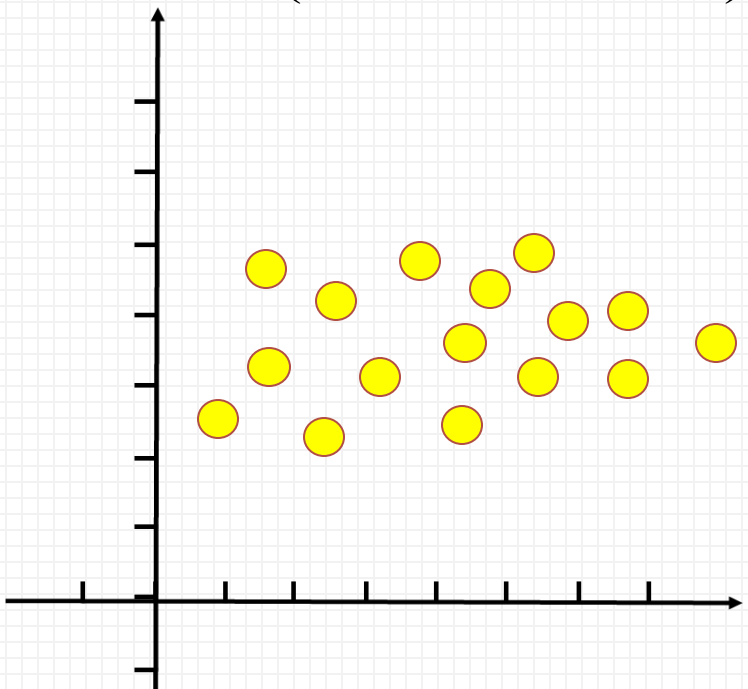
$$\begin{pmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1n} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2n} \\ \vdots & & \vdots & & \vdots \\ p_{k1} & p_{k2} & p_{k3} & \cdots & p_{kn} \end{pmatrix}_{k*n} \begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{pmatrix}_{n*1} = \begin{pmatrix} y_{11} \\ y_{21} \\ \vdots \\ y_{k1} \end{pmatrix}_{k*1}$$

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ -2 & -3 \end{bmatrix}$$

2.2 基的求解

含有n个变量的m个样本,经过转换后变成了含有k个变量的m个样本.(k<n实现降维)

$$\begin{pmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1n} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2n} \\ \vdots & & \vdots & & \vdots \\ p_{k1} & p_{k2} & p_{k3} & \cdots & p_{kn} \end{pmatrix}_{k*n} \begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1m} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2m} \\ \vdots & & \vdots & & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \cdots & x_{nm} \end{pmatrix}_{n*m} = \begin{pmatrix} y_{11} & y_{12} & y_{13} & \cdots & y_{1m} \\ y_{21} & y_{22} & y_{23} & \cdots & y_{2m} \\ \vdots & & \vdots & & \vdots \\ y_{k1} & y_{k2} & y_{k3} & \cdots & y_{km} \end{pmatrix}_{k*m}$$



这些样本现在要压缩到某一维上,直觉来看,你觉得用向横轴投影得到点代表原来的点好,还是用向纵轴轴投影得到点代表原来的点好?如何评价投影结果好还是不好？

2.2 基的求解

将**同一个轴上的坐标做中心化处理**$\widehat{Y_i} = Y_i - \mu_i$,则$E(Y_i)=0$.

目标:(1)投影到每一个新轴上的坐标,应该足够的离散(同轴方差要大)　　$S=\frac{1}{n}\sum(X-\mu)^2$

(2)投影到不同轴上的坐标,应该尽量必要相互影响(不同轴间协方差要小)　　$S=\frac{1}{n}\sum[(X-\mu)(Y-\nu)]$

$$\begin{pmatrix} y_{11} & y_{12} & y_{13} & \cdots & y_{1m} \\ y_{21} & y_{22} & y_{23} & \cdots & y_{2m} \\ \vdots & \vdots & \vdots & & \\ y_{k1} & y_{k2} & y_{k3} & \cdots & y_{km} \end{pmatrix} \rightarrow \begin{pmatrix} \hat{y}_{11} & \hat{y}_{12} & \hat{y}_{13} & \cdots & \hat{y}_{1m} \\ \hat{y}_{21} & \hat{y}_{22} & \hat{y}_{23} & \cdots & \hat{y}_{2m} \\ \vdots & \vdots & \vdots & & \\ \hat{y}_{k1} & \hat{y}_{k2} & \hat{y}_{k3} & \cdots & \hat{y}_{km} \end{pmatrix}$$

2.2 基的求解

$$\hat{y}_{11}*\hat{y}_{11}+\hat{y}_{12}*\hat{y}_{12}+\cdots+\hat{y}_{1m}*\hat{y}_{1m} \qquad \hat{y}_{11}*\hat{y}_{21}+\hat{y}_{12}*\hat{y}_{22}+\cdots+\hat{y}_{1m}*\hat{y}_{2m} \qquad \cdots \qquad \hat{y}_{11}*\hat{y}_{k1}+\hat{y}_{12}*\hat{y}_{k2}+\cdots+\hat{y}_{1m}*\hat{y}_{km}$$

$$\begin{pmatrix} \hat{y}_{11}\hat{y}_{12}\hat{y}_{13}\cdots\hat{y}_{1m} \\ \hat{y}_{21}\hat{y}_{22}\hat{y}_{23}\cdots\hat{y}_{2m} \\ \vdots \quad \vdots \quad \vdots \\ \hat{y}_{k1}\hat{y}_{k2}\hat{y}_{k3}\cdots\hat{y}_{km} \end{pmatrix} \rightarrow$$

$$\hat{y}_{21}*\hat{y}_{21}+\hat{y}_{22}*\hat{y}_{22}+\cdots+\hat{y}_{2m}*\hat{y}_{2m}$$

对　称

…

$$\hat{y}_{k1}*\hat{y}_{k1}+\hat{y}_{k2}*\hat{y}_{k2}+\cdots+\hat{y}_{km}*\hat{y}_{km}$$

目标:(1)投影到每一个新轴上的坐标,应该足够的离散(同轴方差要大)　$S=\frac{1}{n}\sum X^2$

(2)投影到不同轴上的坐标,应该尽量必要相互影响(不同轴间协方差要小)　$S=\frac{1}{n}\sum[XY]$

2.2 基的求解

请你计算:

$$\begin{pmatrix} \hat{y}_{11}\hat{y}_{12}\hat{y}_{13}\cdots\hat{y}_{1m} \\ \hat{y}_{21}\hat{y}_{22}\hat{y}_{23}\cdots\hat{y}_{2m} \\ \vdots \quad \vdots \quad \vdots \\ \hat{y}_{k1}\hat{y}_{k2}\hat{y}_{k3}\cdots\hat{y}_{km} \end{pmatrix} * \begin{pmatrix} \hat{y}_{11}\hat{y}_{12}\hat{y}_{13}\cdots\hat{y}_{1m} \\ \hat{y}_{21}\hat{y}_{22}\hat{y}_{23}\cdots\hat{y}_{2m} \\ \vdots \quad \vdots \quad \vdots \\ \hat{y}_{k1}\hat{y}_{k2}\hat{y}_{k3}\cdots\hat{y}_{km} \end{pmatrix}^{T}$$

结果:

$$\hat{y}_{11}*\hat{y}_{11}+\hat{y}_{12}*\hat{y}_{12}+\cdots+\hat{y}_{1m}*\hat{y}_{1m} \qquad \hat{y}_{11}*\hat{y}_{21}+\hat{y}_{12}*\hat{y}_{22}+\cdots+\hat{y}_{1m}*\hat{y}_{2m} \qquad \cdots \qquad \hat{y}_{11}*\hat{y}_{k1}+\hat{y}_{12}*\hat{y}_{k2}+\cdots+\hat{y}_{1m}*\hat{y}_{km}$$

$$\hat{y}_{21}*\hat{y}_{21}+\hat{y}_{22}*\hat{y}_{22}+\cdots+\hat{y}_{2m}*\hat{y}_{2m}$$

对　　称

…

$$\hat{y}_{k1}*\hat{y}_{k1}+\hat{y}_{k2}*\hat{y}_{k2}+\cdots+\hat{y}_{km}*\hat{y}_{km}$$

2.2 基的求解

发现得到的结果正是之前的**Y的协方差矩阵,**从而目标变为**将Y的协方差矩阵对角化.**

$$Y = PX$$

$$YY^T = (PX)(PX)^T = PXX^TP^T = \Lambda$$

$$Q = XX^T, W = P^T$$

目标:寻找一个矩阵W(列之间是标准正交的),使得

$$W^TQW = \Lambda$$

解:这正是线代中的对称矩阵对角化,对角化后的矩阵斜对角线正是矩阵Q的特征值,且苦苦寻求的矩阵W的每一列便是各个特征值相对应的特征向量组成的.

2.2 基的求解

注意到上述问题我们之前做了两个假定:

(1)要想让新基组成的矩阵乘X之后得到相对应的坐标, 前提是那些矩阵的行**是标准正交基.**

(2)协方差矩阵的结果前提是Y做过中心化处理.

解:

(1)解出 XX^T对应特征值特征向量,将特征向量标准化即可.

(2)将dataset按特征中心化即可.经过矩阵变换的结果即为Y中心化的结果.

$$\begin{pmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} & \cdots & \gamma_{1n} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} & \cdots & \gamma_{2n} \\ \vdots & & \vdots & & \vdots \\ \gamma_{k1} & \gamma_{k2} & \gamma_{k3} & \cdots & \gamma_{kn} \end{pmatrix}_{k*n} \begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1m} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2m} \\ \vdots & & \vdots & & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \cdots & x_{nm} \end{pmatrix}_{n*m} = \begin{pmatrix} y_{11} & y_{12} & y_{13} & \cdots & y_{1m} \\ y_{21} & y_{22} & y_{23} & \cdots & y_{2m} \\ \vdots & & \vdots & & \vdots \\ y_{k1} & y_{k2} & y_{k3} & \cdots & y_{km} \end{pmatrix}_{k*m}$$

RM	AGE	DIS
6.575	65.2	4.09
6.421	78.9	4.9671
7.185	61.1	4.9671
6.998	45.8	6.0622
7.147	54.2	6.0622
6.43	58.7	6.0622
6.012	66.6	5.5605

2.3 算法流程

输入 ： dataset D ,每行是一个样本,每列是一个特征.m行n列.

Step1：处理数据,将数据集D按属性中心化.

Step2：计算XX^T(X = D^T)的各个特征值对应的特征向量,并将得到特征向量标准化.得到矩阵P.

Step3：按P矩阵的每一列按对应特征值降序排列.

Step4：将P矩阵的前K列拿出后按行排列(就是转置),记为W,则X降维后的坐标为

$$Y=WX$$

Y的第L列即为D中第L个样本降维后的结果.

输出 ：1. 对应特征向量(K个),按特征值大小排序依次称为第一主成分,第二主成分等.

2.数据集D降维后的结果.

di san zhang jie

第三章节

算法分析

3.1 算法分析

优点：

1）完全无参数限制.

2）降低算法的计算开销.

缺点：

1）在对数据完全无知的情况下，PCA变换并不能得到较好的保留数据信息.

2）对降维最终得到的数目，也就是潜在的隐变量的数目，不能很好的估计.

3）最终的主成分有时不具有很好的解释性.

4）在将为过程中没有考虑到非线性相关性.

di si zhang jie

第四章节

算法拓展

4.1 算法拓展

降维算法还有很多,比如有LDA, LASSO, MDS, t-SNE,基于深度学习的数据降维等等。

以下是几篇这方面的博客链接，大家感兴趣可以看看。

[1] https://www.cnblogs.com/pinard/p/6244265.html

[2] https://blog.csdn.net/weixin_43374551/article/details/83688913

[3] https://zhuanlan.zhihu.com/p/51441355

[4]http://www.datakit.cn/blog/2017/02/05/t_sne_full.html#11%E5%9F%BA%E6%9C%AC%E5%8E%9F%E7%90%86

[5] https://www.cnblogs.com/huangyc/p/9824202.html

di wu zhang jie

第五章节

案例实操

5.1 案例实操

这里以MATLAB自带PCA函数及其example进行讲解.

Syntax

```
coeff = pca(X)                                          example
coeff = pca(X,Name,Value)                               example
[coeff,score,latent] = pca( __ )                        example
[coeff,score,latent,tsquared] = pca( __ )               example
[coeff,score,latent,tsquared,explained,mu] = pca( __ )  example
```

Description

`coeff = pca(X)` returns the principal component coefficients, also known as loadings, for the *n*-by-*p* data matrix `X`. Rows of `X` correspond to observations and columns correspond to variables. The coefficient matrix is *p*-by-*p*. Each column of `coeff` contains coefficients for one principal component, and the columns are in descending order of component variance. By default, `pca` centers the data and uses the singular value decomposition (SVD) algorithm. example

`coeff = pca(X,Name,Value)` returns any of the output arguments in the previous syntaxes using additional options for computation and handling of special data types, specified by one or more `Name,Value` pair arguments. example

For example, you can specify the number of principal components `pca` returns or an algorithm other than SVD to use.

`[coeff,score,latent] = pca( __ )` also returns the principal component scores in `score` and the principal component variances in `latent`. You can use any of the input arguments in the previous syntaxes. example

Principal component scores are the representations of `X` in the principal component space. Rows of `score` correspond to observations, and columns correspond to components.

The principal component variances are the eigenvalues of the covariance matrix of `X`.

5.1 案例实操

▼ Principal Component Coefficients, Scores, and Variances

Find the coefficients, scores, and variances of the principal components.

Load the sample data set.

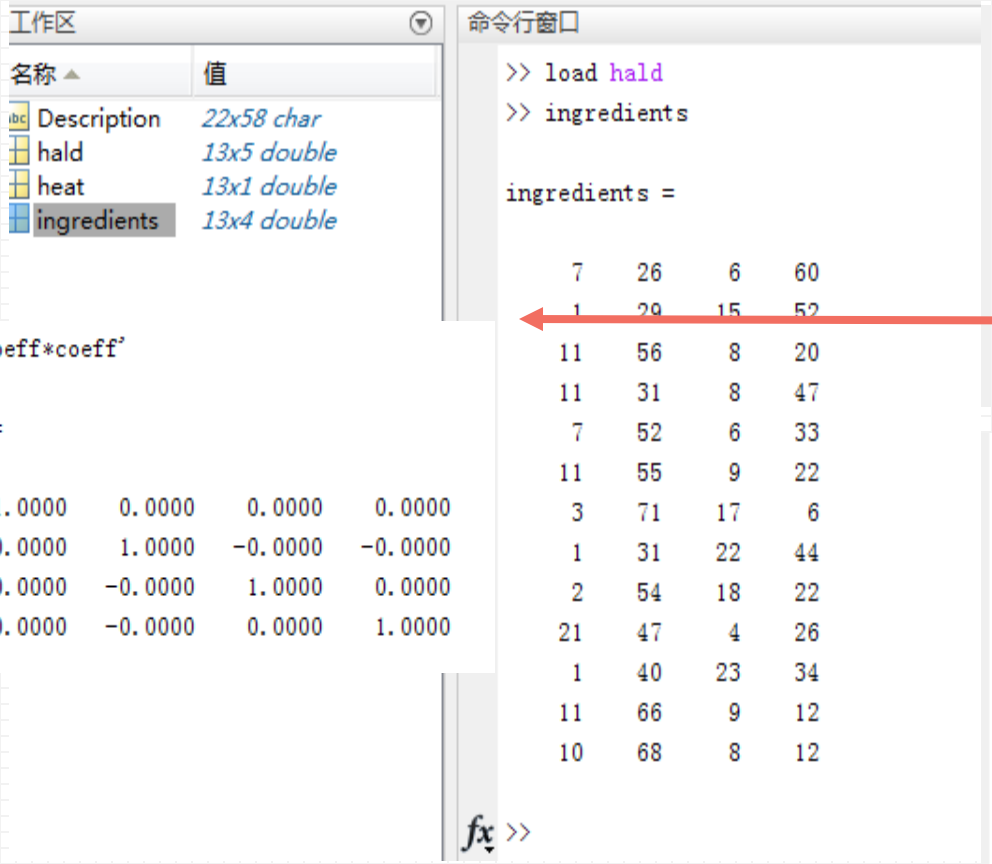
```
load hald
```

The ingredients data has 13 observations for 4 variables.

Find the principal component coefficients, scores, and variances of the components for the ingredients data.

```
[coeff,score,latent] = pca(ingredients)
```

5.1 案例实操




```
>> coeff*coeff'

ans =

    1.0000    0.0000    0.0000    0.0000
    0.0000    1.0000   -0.0000   -0.0000
    0.0000   -0.0000    1.0000    0.0000
    0.0000   -0.0000    0.0000    1.0000
```

```
ingredients (%):
column1: 3Ca0.Al203 (tricalcium aluminate)
column2: 3Ca0.Si02 (tricalcium silicate)
column3: 4Ca0.Al203.Fe203 (tetracalcium aluminoferrite)
column4: 2Ca0.Si02 (beta-dicalcium silicate)
```

```
>> [coeff,score,latent] = pca(ingredients)

coeff =

   -0.0678   -0.6460    0.5673    0.5062
   -0.6785   -0.0200   -0.5440    0.4933
    0.0290    0.7553    0.4036    0.5156
    0.7309   -0.1085   -0.4684    0.4844
```

一组标准正交基

平方和为1,即长度为1

```
score =

   36.8218   -6.8709   -4.5909    0.3967
   29.6073    4.6109   -2.2476   -0.3958
  -12.9818   -4.2049    0.9022   -1.1261
   23.7147   -6.6341    1.8547   -0.3786
   -0.5532   -4.4617   -6.0874    0.1424
  -10.8125   -3.6466    0.9130   -0.1350
  -32.5882    8.9798   -1.6063    0.0818
   22.6064   10.7259    3.2365    0.3243
   -9.2626    8.9854   -0.0169   -0.5437
   -3.2840  -14.1573    7.0465    0.3405
    9.2200   12.3861    3.4283    0.4352
  -25.5849   -2.7817   -0.3867    0.4468
  -26.9032   -2.9310   -2.4455    0.4116
```

```
latent =

  517.7969
   67.4964
   12.4054
    0.2372
```

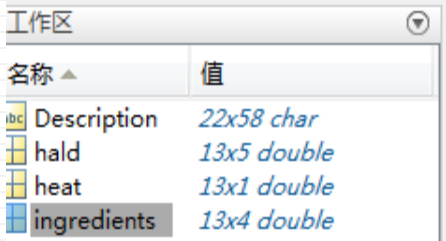
```
>> latent / sum(latent )

ans =

    0.8660
    0.1129
    0.0207
    0.0004
```

Coeff：主成分系数,列按特征值大小降序排序.

Score：原样本在新基下的坐标.

Latent：对应特征值.即各个轴上的方差

5.1 案例实操




```
>> load hald
>> ingredients

ingredients =

     7    26     6    60
     1    29    15    52
    11    56     8    20
    11    31     8    47
     7    52     6    33
    11    55     9    22
     3    71    17     6
     1    31    22    44
     2    54    18    22
    21    47     4    26
     1    40    23    34
    11    66     9    12
    10    68     8    12
```

中心化处理

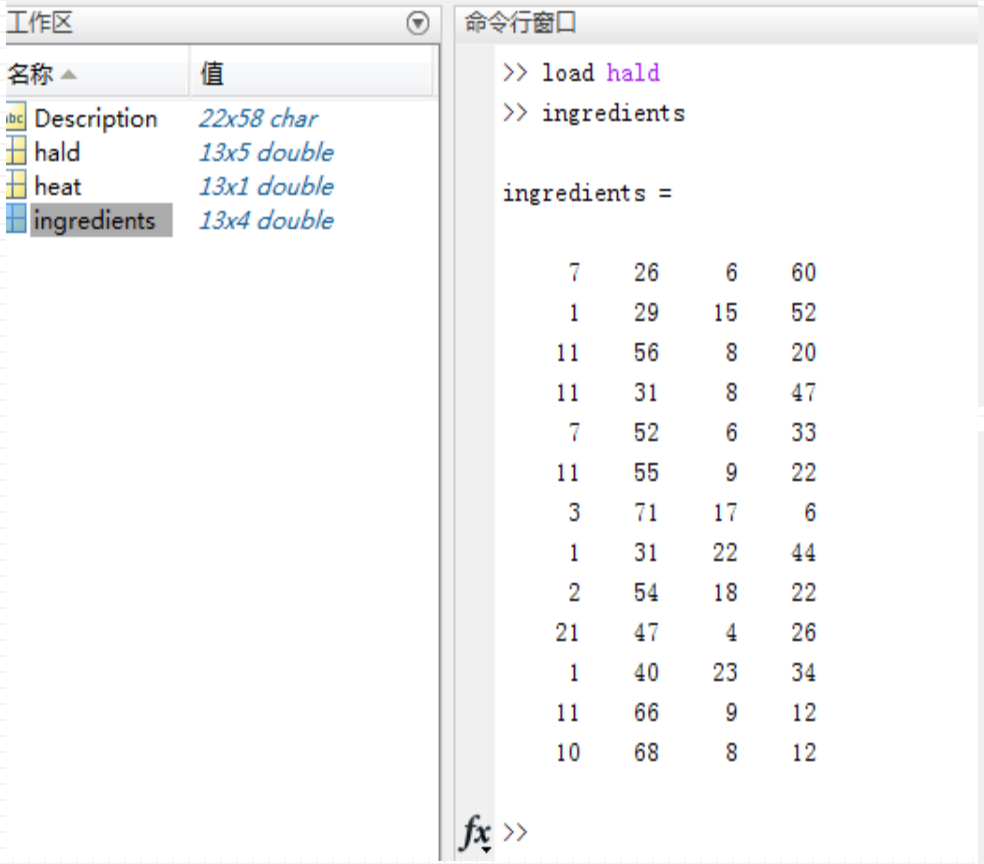
```
>> mean(ingredients)

ans =

    7.4615   48.1538   11.7692   30.0000

>> x1 = ingredients(1,:) - mean(ingredients)

x1 =

   -0.4615  -22.1538   -5.7692   30.0000
```

```
ingredients (%):
column1: 3Ca0.Al203 (tricalcium aluminate)
column2: 3Ca0.Si02 (tricalcium silicate)
column3: 4Ca0.Al203.Fe203 (tetracalcium aluminoferrite)
column4: 2Ca0.Si02 (beta-dicalcium silicate)
```

```
>> [coeff,score,latent] = pca(ingredients)

coeff =

   -0.0678   -0.6460    0.5673    0.5062
   -0.6785   -0.0200   -0.5440    0.4933
    0.0290    0.7553    0.4036    0.5156
    0.7309   -0.1085   -0.4684    0.4844

score =

   36.8218   -6.8709   -4.5909    0.3967
   29.6073   -4.6109   -2.2476   -0.3958
  -12.9818   -4.2049    0.9022   -1.1261
   23.7147   -6.6341    1.8547   -0.3786
   -0.5532   -4.4617   -6.0874    0.1424
  -10.8125   -3.6466    0.9130   -0.1350
  -32.5882    8.9798   -1.6063    0.0818
   22.6064   10.7259    3.2365    0.3243
   -9.2626    8.9854   -0.0169   -0.5437
   -3.2840  -14.1573    7.0465    0.3405
    9.2200   12.3861    3.4283    0.4352
  -25.5849   -2.7817   -0.3867    0.4468
  -26.9032   -2.9310   -2.4455    0.4116
```

```
>> coeff(:,1)'

ans =

   -0.0678   -0.6785    0.0290    0.7309

>> x1'

ans =

   -0.4615
  -22.1538
   -5.7692
   30.0000

>> coeff(:,1)'*x1'

ans =

   36.8218
```

5.1 案例实操




```
ingredients (%):
column1: 3Ca0.Al203 (tricalcium aluminate)
column2: 3Ca0.Si02 (tricalcium silicate)
column3: 4Ca0.Al203.Fe203 (tetracalcium aluminoferrite)
column4: 2Ca0.Si02 (beta-dicalcium silicate)
```

```
>> [coeff,score,latent] = pca(ingredients)

coeff =

   -0.0678   -0.6460    0.5673    0.5062
   -0.6785   -0.0200   -0.5440    0.4933
    0.0290    0.7553    0.4036    0.5156
    0.7309   -0.1085   -0.4684    0.4844

score =

   36.8218   -6.8709   -4.5909    0.3967
   29.6073    4.6109   -2.2476   -0.3958
  -12.9818   -4.2049    0.9022   -1.1261
   23.7147   -6.6341    1.8547   -0.3786
   -0.5532   -4.4617   -6.0874    0.1424
  -10.8125   -3.6466    0.9130   -0.1350
  -32.5882    8.9798   -1.6063    0.0818
   22.6064   10.7259    3.2365    0.3243
   -9.2626    8.9854   -0.0169   -0.5437
   -3.2840  -14.1573    7.0465    0.3405
    9.2200   12.3861    3.4283    0.4352
  -25.5849   -2.7817   -0.3867    0.4468
  -26.9032   -2.9310   -2.4455    0.4116

latent =

  517.7969
   67.4964
   12.4054
    0.2372
```

```
>> latent / sum(latent )

ans =

    0.8660
    0.1129
    0.0207
    0.0004
```

可以看到在前两个轴上的方差之和占比已经约为98%,这说明将原样本投影到这两个轴上之后得到的点已经**可以解释原样本98%的信息**了,所以建议将原样本**降维到这两个新的维度上**.

5.1 案例实操

```
>> [coeff,score,latent] = pca(ingredients)

coeff =

   -0.0678   -0.6460    0.5673    0.5062
   -0.6785   -0.0200   -0.5440    0.4933
    0.0290    0.7553    0.4036    0.5156
    0.7309   -0.1085   -0.4684    0.4844
```

```
ingredients (%):
column1: 3Ca0.Al203 (tricalcium aluminate)
column2: 3Ca0.Si02 (tricalcium silicate)
column3: 4Ca0.Al203.Fe203 (tetracalcium aluminoferrite)
column4: 2Ca0.Si02 (beta-dicalcium silicate)
```

```
ingredients =

     7    26     6    60
     1    29    15    52
    11    56     8    20
    11    31     8    47
     7    52     6    33
    11    55     9    22
     3    71    17     6
     1    31    22    44
     2    54    18    22
    21    47     4    26
     1    40    23    34
    11    66     9    12
    10    68     8    12
```

```
score =

   36.8218   -6.8709   -4.5909    0.3967
   29.6073    4.6109   -2.2476   -0.3958
  -12.9818   -4.2049    0.9022   -1.1261
   23.7147   -6.6341    1.8547   -0.3786
   -0.5532   -4.4617   -6.0874    0.1424
  -10.8125   -3.6466    0.9130   -0.1350
  -32.5882    8.9798   -1.6063    0.0818
   22.6064   10.7259    3.2365    0.3243
   -9.2626    8.9854   -0.0169   -0.5437
   -3.2840  -14.1573    7.0465    0.3405
    9.2200   12.3861    3.4283    0.4352
  -25.5849   -2.7817   -0.3867    0.4468
  -26.9032   -2.9310   -2.4455    0.4116
```

CS = -0.0678*c1-0.6785*c2+0.0290*c3+0.7309*c4
可以看到,这个新的维度中原变量c2和c4占比较大,结合原属性分析,可知构造的CS变量是描述样本内Cao&SiO2属性高低的一个综合型变量.

CA = -0.6460*c1-0.02*c2+0.7553*c3-0.1085*c4
CA变量是描述样本内Cao&Al2O3属性高低的一个综合型变量.

(如果手算结果不对,不要忘记中心化处理,这里的结果是默认中心化处理后的.)

5.2 广告时间

其他课程:

基于Python实现网络爬虫：https://www.bilibili.com/video/BV1WV411U7LQ

通俗易懂关联规则 : https://www.bilibili.com/video/BV13f4y1k7x6

通俗易懂K均值聚类 : https://www.bilibili.com/video/BV16v4y1Z7xJ

通俗易懂ID3分类 : https://www. bilibili.com/video/BV1kg411u7RP

同时如果有一些其他问题，可以联系我

QQ ： 1366420642，Q群：1019030249

欢迎大佬萌新加入

参　考　资　料

【1】http://blog.codinglabs.org/articles/pca-tutorial.html

THANKS

谢谢观看