第六章 函数逼近

殷东生

dyin@math.tsinghua.edu.cn

清华大学数学科学系

2018年秋季学期

设 X是区间$[a,b]$某类函数组成的线性空间，M为X的子集。函数逼近问题可以叙述如下：

定义 (函数逼近)

设$f\in X$，求 $p\in M$使得

$$\|p-f\|=\min_{s\in M}\|f-s\|$$

通常取 $C[a,b]$，M为便于计算的函数集合。一般取M为代数多项式、三角多项式或有理多项式组。

- 代数多项式：$M=\mathcal{P}_n([a,b])$；
- 三角函数：e^{ikx}, $sinkx$, $coskx$, $k=0,1,\cdots$；
- 有理多项式：

$$\frac{p_m(x)}{q_n(x)}, p_m\in\mathcal{P}_m([a,b]),\ p_n\in\mathcal{P}_n([a,b]).$$

常用的度量有：

- 极大范数逼近为 一致逼近–uniform approximation：

$$\|f-p\|_\infty = \max_{a\le x\le b} |f(x)-p(x)|$$

- 平方范数逼近 称为平方逼近–least-squares approximation：

$$\|f-p\|_2 = \left[\int_a^b |f(x)-p(x)|^2 dx\right]^{\frac{1}{2}}$$

- 1-范数逼近

$$\|f-p\|_1 = \int_a^b |f(x)-p(x)|dx.$$

设ρ是 $[a,b]$上的权函数，定义

$$L_\rho^2[a,b]=\left\{f:\|f\|_2=\sqrt{\int_a^b \rho(x)[f(x)]^2dx}\right\}$$

$L_\rho^2[a,b]$是$\mathbb{R}$上的线性空间，若$\rho(x)\equiv 1$，则为$L^2[a,b]$。

定义 (函数系的线性无关)

设$\varphi_0,\varphi_1,\cdots,\varphi_n\in L_\rho^2[a,b]$，若没有$n+1$个不全为零的数$c_j,j=0,1,\cdots,n$ 使得

$$\|c_0\varphi_0+c_1\varphi_1+\cdots+c_n\varphi_n\|_2=0,$$

则称函数系$\{\varphi_j,j=0,1\cdots,n\}$在$L_\rho^2[a,b]$上线性无关。

设 $\varphi_0, \varphi_1, \cdots, \varphi_n$是$L^2_\rho[a,b]$上线性无关的函数， 则

$$\Phi = \text{span}\{\varphi_0, \varphi_1, \cdots, \varphi_n\}$$

是这些函数张成的线性空间。$\forall s \in \Phi$可以写成

$$s(x) = \sum_{j=0}^{n} a_j \varphi_j(x).$$

定义 (最佳平方逼近)

设$f \in L^2_\rho[a,b]$,若$\exists s^* \in \Phi$使得

$$\|f - s^*\|_2 = \min_{s\in\Phi} \|f - s\|_2,$$

则称 s^*为 f在Φ中的最佳平方逼近函数。

求最佳平方逼近s^*等价于求下面的多元函数

$$F(a_0, a_1, \cdots, a_n) = \int_a^b \rho(x)[f(x) - \sum_{j=0}^{n} a_j\varphi_j(x)]^2 dx$$

的极小值，则由多元函数函数取极值的必要条件有

$$\begin{aligned} & \frac{\partial F(a_0, a_1, \cdots, a_n)}{\partial a_k} \\ = \ & 2\int_a^b \rho(x)[f(x) - \sum_{j=0}^{n} a_j\varphi_j(x)]\varphi_k(x)dx = 0, \quad 0 \le k \le n \end{aligned}$$

由此可得

$$\sum_{j=0}^{n} (\varphi_j, \varphi_k)a_j = (f, \varphi_k), k = 0, 1, \cdots, n. \tag{0.1}$$

(0.1)称为法方程–normal equation。

写成矩阵形式有：

$$\begin{bmatrix} (\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & \cdots & (\varphi_0,\varphi_n) \\ (\varphi_1,\varphi_0) & (\varphi_1,\varphi_1) & \cdots & (\varphi_1,\varphi_n) \\ \vdots & \ddots & \ddots & \vdots \\ (\varphi_n,\varphi_0) & (\varphi_n,\varphi_1) & \cdots & (\varphi_n,\varphi_n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \end{bmatrix}$$

其中内积定义为

$$\begin{aligned} (\varphi_k,\varphi_j) &= \int_a^b \rho(x)\varphi_k(x)\varphi_j(x)dx, \\ (f,\varphi_k) &= \int_a^b \rho(x)f(x)\varphi_k(x)dx = d_k. \end{aligned}$$

定理

设$\mathbb{X}$是内积空间，$(\cdot,\cdot)$是其上的内积，对于$x_1,x_2,\cdots,x_n\in\mathbb{X}$，若

$$G_n=\Big(G_{ij}=(x_i,x_j)\Big)_{n\times n},$$

则$\det G_n\neq 0 \Longleftrightarrow x_1,x_2,\cdots,x_n$线性无关。

法方程 (0.1)存在唯一解

$$a_k = a_k^*,\ k = 0, 1, \cdots, n \Longrightarrow s^* = \sum_{j=0}^{n} a_j^* \varphi_k(x).$$

定理

$s^*(x)$是$f(x)$在Φ中的最佳平方逼近，即

$$\|f - s^*\|_2 \leq \|f - s\|_2,\ \forall s \in \Phi.$$

定理

$s^*(x)$是$f(x)$在Φ中的最佳平方逼近当且仅当$s^*(x)$是$f(x)$在Φ中的正交投影，即

$$(f(x) - s^*(x), s(x)) = 0, \quad \forall s(x) \in \Phi.$$

令

$$\delta(x) = f(x) - s^*(x),$$

则 $\|\delta\|_2$称为最佳平方逼近的误差，

$$\|\delta\|_2^2 = \|f\|_2^2 - \sum_{j=0}^{n} a_j^*(\varphi_j, f).$$

有限元方法

考虑一维Poisson方程：

$$
\begin{aligned}
&-u''(x) = f(x), \quad 0 < x < 1, \\
&u(0) = u(1) = 0.
\end{aligned}
$$

设$v(x)$是定义在$[0,1]$上满足$v(0) = v(1) = 0$的函数，在Poisson方程两边同乘于$v(x)$并分部积分：

$$
\begin{aligned}
-\int_0^1 u''v dx &= -\int_0^1 v du' = -u'v\Big|_0^1 + \int_0^1 u'v' dx \\
&= \int_0^1 u'v' dx = \int_0^1 fv dx
\end{aligned}
$$

令

$$
a(u,v) = \int_0^1 u'v' dx, \quad (f,v) = \int_0^1 fv dx
$$

定义空间

$$V = \left\{ v \in L^2([0,1]), \quad a(v,v) < +\infty, \quad v(0) = v(1) = 0 \right\},$$

则Poisson方程等价的转化为如下的变分问题：

$$\begin{aligned} &\text{求} \quad u(x) \in V, \\ &\text{使得} \quad a(u,v) = (f,v), \quad \forall v(x) \in V \end{aligned}$$

此处V是无穷维空间，无法直接求解，必须考虑其有限维子空间V_h中的近似问题：

$$\begin{aligned} &\text{求} \quad u_h(x) \in V_h, \\ &\text{使得} \quad a(u_h,v_h) = (f,v_h), \quad \forall v_h(x) \in V_h \end{aligned} \tag{0.2}$$

有Galerkin正交关系

$$a(u - u_h, v_h) = 0, \quad \forall v_h \in V_h.$$

有限元空间

有限元方法就是选取V_h为有限元空间，考虑$[0,1]$的一个剖分

$$0 = x_0 < x_1 < x_2 < \cdots < x_{i-1} < x_i < \cdots < x_n = 1$$

其中 x_i称为网格节点，$h_i = x_i - x_{i-1}, 1 \le i \le n$，而 $h = \max_{1\le i\le n} h_i$。定义线性有限元空间

$$V_h = \left\{ v \in C([0,1]), v(0) = v(1) = 0,\ \ v\big|_{[x_{i-1},x_i]} \in \mathcal{P}_1 \right\}$$

此空间其实是分片线性函数空间，其基函数即为分片线性插值的节点基函数$\phi_i, 1 \le i \le n-1$。

$$\phi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{h_i}, & x \in [x_{i-1}, x_i], \\ \dfrac{x_{i+1} - x}{h_{i+1}}, & x \in [x_i, x_{i+1}], \end{cases} \qquad \phi_i(x) = 0,\ x \notin [x_{i-1}, x_{i+1}]$$

对$\forall v_h \in V_h$，令

$$v_i = v_h(x_i), \quad 1 \le i \le n-1,$$

则

$$v_h(x) = v_1\phi_1(x) + v_2\phi_2(x) + \cdots + v_{n-1}\phi_{n-1}(x)$$

同理有

$$u_h(x) = \sum_{i=1}^{n-1} u_i\phi_i(x), \qquad \text{其中} \quad u_i = u_h(x_i)$$

(0.2)对任意的$v_h \in V_h$成立，只需对V_h的基底$\phi_i(x), 1 \le i \le n-1$成立：

$$a(u_h, \phi_i) = (f, \phi_i), \quad 1 \le i \le n-1.$$

亦即

$$a(\phi_1, \phi_i)u_1 + a(\phi_2, \phi_i)u_2 + \cdots + a(\phi_{n-1}, \phi_i)u_{n-1} = (f, \phi_i),$$
$$1 \le i \le n-1.$$

令

$$
\begin{aligned}
k_{ij} &= a(\phi_j, \phi_i) = \int_0^1 \phi_j' \phi_i' dx, \\
f_i &= (f, \phi_i) = \int_0^1 f \phi_i dx
\end{aligned}
$$

和

$$
K = (k_{ij}) \in \mathbb{R}^{(n-1)\times(n-1)}, \quad F = (f_i) \in \mathbb{R}^{n-1}, \qquad U = (u_i) \in \mathbb{R}^{n-1}
$$

可得法方程

$$
KU = F \tag{0.3}
$$

此处K称为刚度矩阵，F为荷载向量。
当x_i和x_j不相邻时

$$
a(\phi_i, \phi_j) = 0,
$$

所以K是稀疏矩阵。

刚度矩阵的计算

经过简单的计算有

$$
\begin{aligned}
\int_{x_{i-1}}^{x_i} \phi_i' \phi_i' dx &= \int_{x_{i-1}}^{x_i} \frac{1}{h_i^2} dx = \frac{1}{h_i}, \\
\int_{x_{i-1}}^{x_i} \phi_{i-1}' \phi_{i-1}' dx &= \int_{x_{i-1}}^{x_i} \frac{1}{h_i^2} dx = \frac{1}{h_i}, \\
\int_{x_{i-1}}^{x_i} \phi_i' \phi_{i-1}' dx &= \int_{x_{i-1}}^{x_i} -\frac{1}{h_i^2} dx = -\frac{1}{h_i}.
\end{aligned}
$$

则

$$
\begin{aligned}
\int_0^1 \phi_i' \phi_i' dx &= \int_{x_{i-1}}^{x_i} \phi_i' \phi_i' dx + \int_{x_i}^{x_i+1} \phi_i' \phi_i' dx = \frac{1}{h_i} + \frac{1}{h_{i+1}}, \\
\int_0^1 \phi_i' \phi_{i-1}' dx &= \int_{x_{i-1}}^{x_i} \phi_i' \phi_{i-1}' dx = -\frac{1}{h_i}.
\end{aligned}
$$

$$
\begin{aligned}
a(\phi_i,\phi_i) &= \int_0^1 \phi_i'\phi_i' dx = \frac{1}{h_i}+\frac{1}{h_{i+1}},\\
a(\phi_i,\phi_{i-1}) &= \int_0^1 \phi_i'\phi_{i-1}' dx = -\frac{1}{h_i}.
\end{aligned}
$$

刚度矩阵

$$
K=\begin{pmatrix}
\frac{1}{h_1}+\frac{1}{h_2} & -\frac{1}{h_2} & & & & \\
-\frac{1}{h_1} & \frac{1}{h_2}+\frac{1}{h_3} & -\frac{1}{h_3} & & & \\
 & \ddots & \ddots & \ddots & & \\
 & & -\frac{1}{h_i} & \frac{1}{h_i}+\frac{1}{h_{i+1}} & -\frac{1}{h_{i+1}} & \\
 & & & \ddots & \ddots & \ddots \\
 & & & & -\frac{1}{h_{n-2}} & \frac{1}{h_{n-2}}+\frac{1}{h_{n-1}}
\end{pmatrix}
$$

例

考虑$[0,1]$上连续函数$f(x)$的最佳平方逼近，取$\rho(x)=1$，

$$\varphi_k(x)=x^k, k=0,1\cdots,n,$$

求f在$\mathcal{P}_n[0,1]$中的最佳平方逼近多项式。

计算法方程的系数矩阵和右端，有

$$\begin{aligned}(\varphi_j,\varphi_k) &= \int_0^1 x^{j+k}dx=\frac{1}{j+k+1}, \ k,j=0,1\cdots,n,\\ (f,\varphi_k) &= \int_0^1 f(x)x^k dx=d_k, k=0,1,\cdots,n.\end{aligned}$$

则法方程的系数矩阵为

$$H_{n+1}(0,1)=\begin{bmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n+1} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n+1} \end{bmatrix}$$

H_{n+1}被称为Hilbert矩阵。
求解法方程可得最佳平方逼近函数：

$$P_n^*(x)=a_0^*+a_1^*x+\cdots+a_n^*x^n.$$

Hilbert矩阵是病态的，因此直接从法方程求

$$a_j^*,\quad j=0,1\cdots,n$$

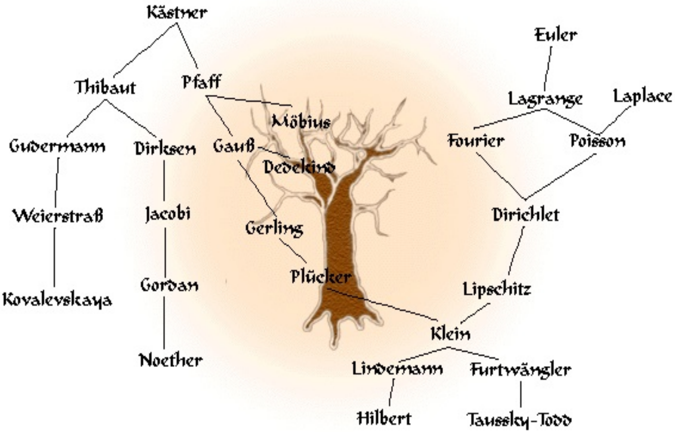
是相当困难的。

数学家简介 (Hilbert)

希尔伯特（*1862–1943*），德国数学家，於代数不变量、代数数论、几何基础、变分法、*Hilbert* 空间等方面都有了不起的贡献，堪称他那时代最伟大的数学家。他提倡数学公理化，还有提出「*Hilbert 23*问题」，对於二十世纪的数学发展影响甚大。希尔伯特去世时，德国《自然》杂志发表过这样的观点：现在世界上难得有一位数学家的工作不是以某种途径导源于希尔伯特的工作。他像是数学世界的亚历山大，在整个数学版图上，留下了他那显赫的名字。*1900*年，希尔伯特在巴黎数学家大会上提出了*23*个最重要的问题供二十世纪的数学家们去研究，这就是著名的"希尔伯特*23*个问题"。*1976*年，在美国数学家评选的自*1940*年以来美国数学的十大成就中，有三项就是希尔伯特第*1*、第*5*、第*10*问题的解决。由此可见，能解决希尔伯特问题，是当代数学家的无上光荣。

他的学生有：Hermann Weyl, Richard Courant等。

Mathematics Genealogy



设

$$f(x) \in L_\rho^2[a,b], \Phi = \mathrm{span}\{\varphi_0(x), \varphi_1(x), \cdots, \varphi_n(x)\},$$

其中 $\varphi_0(x), \varphi_1(x), \cdots, \varphi_n(x)$是以$\rho(x)$为权的正交多项式族。

$$(\varphi_i, \varphi_j) = \int_a^b \varphi_1(x)\varphi_j(x)\rho(x)dx = \begin{cases} 0 & i \neq j \\ A_j > 0 & i = j \end{cases}$$

所以法方程(0.1)的系数矩阵为

$$G_n = \begin{bmatrix} (\varphi_0, \varphi_0) & 0 & \cdots & 0 \\ 0 & (\varphi_1, \varphi_1) & \cdots & 0 \\ \cdots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & (\varphi_n, \varphi_n) \end{bmatrix}$$

直接求解可得

$$a_k^* = \frac{(f, \varphi_k)}{(\varphi_k, \varphi_k)}, k = 0, 1, \cdots, n.$$

因此f在 Φ中的最佳平方逼近函数为

$$s_n^*(x) = \sum_{j=0}^{n} \frac{(f, \varphi_j)}{\|\varphi_j\|_2^2} \varphi_j(x). \tag{0.4}$$

例

在$[0, 1]$上用权$\rho(x) = 1$的正交多项式对$f(x) = \sin \pi x, x \in [0, 1]$做二次最佳逼近。

解：用Gram-Schmidt正交化构造正交多项式$\varphi_0, \varphi_1, \varphi_2$,

$$\begin{aligned} \varphi_0(x) &= 1, \\ (\varphi_0, \varphi_0) &= 1, (x, \varphi_0) = \frac{1}{2}, \varphi_1(x) = x - \frac{1}{2} \end{aligned}$$

$$
\begin{aligned}
(\varphi_1,\varphi_1) &= \frac{1}{12}, \\
(x^2,\varphi_1) &= \frac{1}{12}, \\
(x^2,\varphi_0) &= \frac{1}{3}, \\
\varphi_2(x) &= = x^2 - \frac{(x^2,\varphi_1)}{(\varphi_1,\varphi_1)}\varphi_1 - \frac{(x^2,\varphi_0)}{(\varphi_0,\varphi_0)}\varphi_0 = x^2 - x + \frac{1}{6} \\
(f,\varphi_0) &= \int_0^1 \sin\pi x dx = \frac{2}{\pi}, \\
(f,\varphi_1) &= \int_0^1 (x - \frac{1}{2}\sin\pi x) dx = 0, \\
(f,\varphi_2) &= \int_0^1 (x^2 - x + \frac{1}{6})\sin\pi x dx = -\frac{4}{\pi^3} + \frac{1}{3\pi}, \\
(\varphi_2,\varphi_2) &= \int_0^1 (x^2 - x + \frac{1}{6})^2 dx = \frac{1}{180}
\end{aligned}
$$

所以

$$
\begin{aligned}
a_0^* &= \frac{(f,\varphi_0)}{\|\varphi_0\|_2^2} = \frac{2}{\pi} \\
a_1^* &= \frac{(f,\varphi_1)}{\|\varphi_1\|_2^2} = 0, \\
a_2^* &= \frac{(f,\varphi_2)}{(\varphi_2,\varphi_2)} = 180\Big(\frac{1}{3\pi} - \frac{4}{\pi^3}\Big) \\
s_2^* &= a_0^*\varphi_0(x) + a_1^*\varphi_1(x) + a_2^*\varphi_2(x) \\
&= \frac{2}{\pi} + 180\Big(\frac{1}{3\pi} - \frac{4}{\pi^3}\Big)\Big(x^2 - x + \frac{1}{6}\Big) \\
&= 180\Big(\frac{1}{3\pi} - \frac{4}{\pi^3}\Big)x^2 - 180\Big(\frac{1}{3\pi} - \frac{4}{\pi^3}\Big)x \\
&\quad + \frac{2}{\pi} + 30\Big(\frac{1}{3\pi} - \frac{4}{\pi^3}\Big)
\end{aligned}
$$

定义

设$f \in L_\rho^2[a,b], \varphi_j \in L_\rho^2[a,b], j = 0,1,\cdots$是正交函数组，称

$$a_j = (f,\varphi_j) = \int_a^b \rho(x) f(x) \varphi_j(x) dx, j = 0,1,\cdots,$$

为 f的广义Fourier系数，相应的级数$\sum_{j=0}^{\infty} a_j \varphi_j(x)$为$f$的广义Fourier级数。

由(0.4)知f的最佳平方逼近s_n^*是f的广义Fourier展开的部分和。而

$$\|f - s_n^*\|_2^2 = \|f\|_2^2 - \sum_{j=0}^{n} \left(\frac{(f,\varphi_j)}{\|\varphi_j\|_2} \right)^2.$$

定理

设$f \in C[a,b]$, s_n^*是f的最佳平方逼近多项式，而$\{\varphi_j\}_{j\geq 0}$是正交多项式组，则有

$$\lim_{n\to\infty} \|f - s_n^*\|_2 = 0.$$

对于 $f \in C[a,b]$ 由Parseval等式

$$\|f\|_2^2 = \sum_{j=0}^{\infty} \left(\frac{(f,\varphi_j)}{\|\varphi_j\|_2} \right)^2.$$

对于首项系数为1 的Legendre多项式 $\tilde{P}_n$有

定理

在所有首项系数为1的n次多项式中，Legendre多项式在$[-1,1]$上与零的平方误差最小。

例

用Legendre多项式展开求$f(x) = e^x$在$[-1, 1]$上的最佳平方逼近多项式(取$n = 1, 3$)。

解:

$$
\begin{aligned}
P_0(x) &= 1, \ \ P_1(x) = x, \\
P_2(x) &= \frac{3}{2}x^2 - \frac{1}{2}, \ \ P_3(x) = \frac{5}{2}x^3 - \frac{3}{2}x.
\end{aligned}
$$

则

$$
\begin{aligned}
(f, P_0) &= \int_{-1}^{1} e^x dx \approx 2.3504, \ (f, P_1) = \int_{-1}^{1} xe^x dx \approx 0.7358, \\
(f, P_2) &= \int_{-1}^{1} (\frac{3}{2}x^2 - \frac{1}{2})e^x dx \approx 0.1432 \\
(f, P_3) &= \int_{-1}^{1} (\frac{5}{2}x^3 - \frac{3}{2}x)e^x dx \approx 0.02013.
\end{aligned}
$$

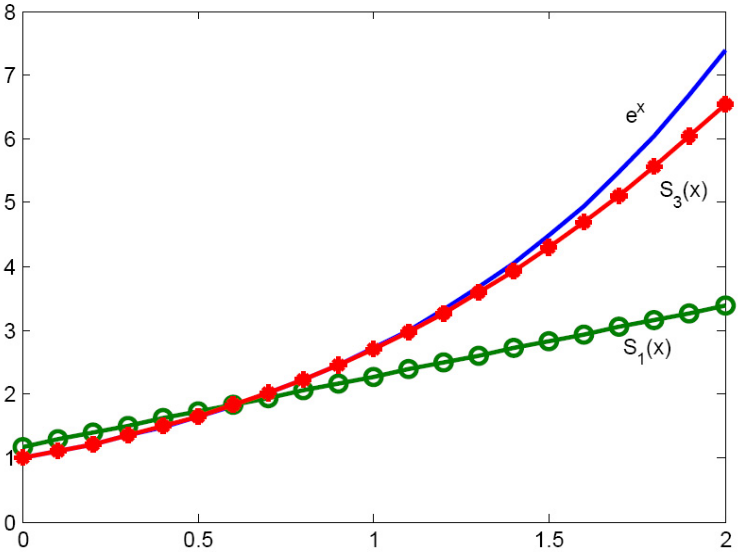
$$\int_{-1}^{1}[P_j(x)]^2dx = \frac{2}{2j+1}, \; j = 0, 1, 2, \cdots,$$

则

$$\begin{aligned} a_0^* &= 1.1752, \;\; a_1^* = 1.1036, \\ a_2^* &= 0.3578, \;\; a_3^* = 0.07046. \end{aligned}$$

所以

$$\begin{aligned} S_1^*(x) &= 1.1752 + 1.1036x, \\ S_3^*(x) &= 1.1752 + 1.1036x + 0.3578(\frac{3}{2}x^2 - \frac{1}{2}) \\ &\quad +0.07046(\frac{5}{2}x^3 - \frac{3}{2}x) \\ &\approx 0.9963 + 0.9979x + 0.5367x^2 + 0.1761x^3 \end{aligned}$$



8
7
6
5
4
3
2
1
0
0
0.5
1
1.5
2
e^x
S_3(x)
S_1(x)

最小二乘法

已知函数$f(x)$在一些离散点上的值

$$(x_k, f(x_k)),\quad k = 0, 1, \cdots, n,$$

来构造$f(x)$的逼近多项式除了前面的插值法外还有最小二乘多项式拟合法。这种方法对 具有下列特点的数据非常有效：

1. 数据本身就有误差(如物理试验中的观测数据就不可避免的会有误差)。
2. 数据量很大。
3. 数据的采样分布能基本反映函数的变化趋势。

对于这样的数据，插值法是不合适的，亦即放弃让所构造的逼近函数$P(x)$满足插值条件 $P(x_i) = f(x_i)$的要求。

设$\varphi_0,\varphi_1,\cdots,\varphi_n$是$L^2_\rho[a,b]$上的线性无关函数组。令

$$\Phi=\operatorname{span}\{\varphi_0,\varphi_1,\cdots,\varphi_n\},$$

而f为在$m+1$个节点$a=x_0<x_1<\cdots<x_m=b$给定的离散函数，亦即f有函数表

$$(x_k,f(x_k)),\ \ k=0,1,\cdots,m$$

定义 (最小二乘法–least square method)

设f为在$m+1$个节点上给定的离散函数，最小二乘法为求$s^*\in\Phi$使得

$$\sum_{j=0}^{m}\rho(x_j)[f(x_j)-s^*(x_j)]^2=\min_{s\in\Phi}\sum_{j=0}^{m}\rho(x_j)[f(x_j)-s(x_j)]^2$$

s^*称为f在$m+1$个节点上的最小二乘解，也称为最小二乘曲线拟合。

上述定义中

$$s(x)=\sum_{k=0}^{n}a_k\varphi_k(x),x\in[a,b],$$

是待求的$a_0,a_1,\cdot,a_n$的线性函数，因而亦称上述问题为线性最小二乘问题。

$s\in\Phi$，所以$s(x)=\sum\limits_{i=0}^{n}a_i\varphi_i(x)$,相应的最小二乘问题为

$$\min_{s\in\Phi}\sum_{j=0}^{m}\rho(x_j)[f(x_j)-s(x_j)]^2=\min_{s\in\Phi}\sum_{j=0}^{m}\rho(x_j)[f(x_j)-\sum_{i=0}^{n}a_i\varphi_i(x_j)]^2$$

求上式的极小值相当于求下面的多元函数的极小值

$$F(a_0,a_1,\cdots,a_n)=\sum_{j=0}^{m}\rho(x_j)[f(x_j)-\sum_{i=0}^{n}a_i\varphi_i(x_j)]^2$$

定义离散函数f,g的内积和范数为

$$(f,g)=\sum_{j=0}^{m}\rho(x_j)f(x_j)g(x_j),\ \|f\|_2=\sqrt{\sum_{j=0}^{m}\rho(x_j)[f(x_j)]^2}.$$

$a_0,a_1,\cdots,a_n$满足的法方程为

$$\sum_{i=0}^{n}(\varphi_k,\varphi_i)a_i=d_k,\quad k=0,1,\cdots,n. \tag{0.5}$$

其中$d_k=(f,\varphi_k)=\sum\limits_{j=0}^{m}\rho(x_j)f(x_j)\varphi_k(x_j)$，具体写成方程组的形式为

$$\begin{bmatrix}(\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & \cdots & (\varphi_0,\varphi_n)\\ (\varphi_1,\varphi_0) & (\varphi_1,\varphi_1) & \cdots & (\varphi_1,\varphi_n)\\ \vdots & \ddots & \ddots & \vdots\\ (\varphi_n,\varphi_0) & (\varphi_n,\varphi_1) & \cdots & (\varphi_n,\varphi_n)\end{bmatrix}\begin{bmatrix}a_0\\ a_1\\ \vdots\\ a_n\end{bmatrix}=\begin{bmatrix}d_0\\ d_1\\ \vdots\\ d_n\end{bmatrix}$$

这个方程组为法方程。

若法方程中的系数矩阵非奇异，则可得唯一的解

$$a^* = (a_0^*, a_1^*, \cdots, a_n^*)^T \in \mathbb{R}^{n+1},$$

相应的最小二乘解为

$$s^*(x) = \sum_{i=0}^{n} a_i^* \varphi_i(x).$$

类似于正交平方逼近可得

$$\|s^* - f\|_2 \leq \|f - s\|_2, \ \forall x \in \Phi.$$

相应的最小二乘拟合的误差为

$$\|\delta^*\|_2^2 = \|s^* - f\|_2 = \|f\|_2^2 - (f, s^*) = \|f\|_2^2 - \sum_{i=0}^{n} a_i^* (f, \varphi_i).$$

取$\Phi = \mathrm{span}(1, x, x^2, \cdots, x^n)$，此时称为多项式曲线拟合.

定理

多项式拟合中，法方程有唯一解。

例

假设系统有一输入参数V和输出参数U，通过试验得到相关数据，可以判断U和V之间存在某种 线性关系，用最小二乘法得到

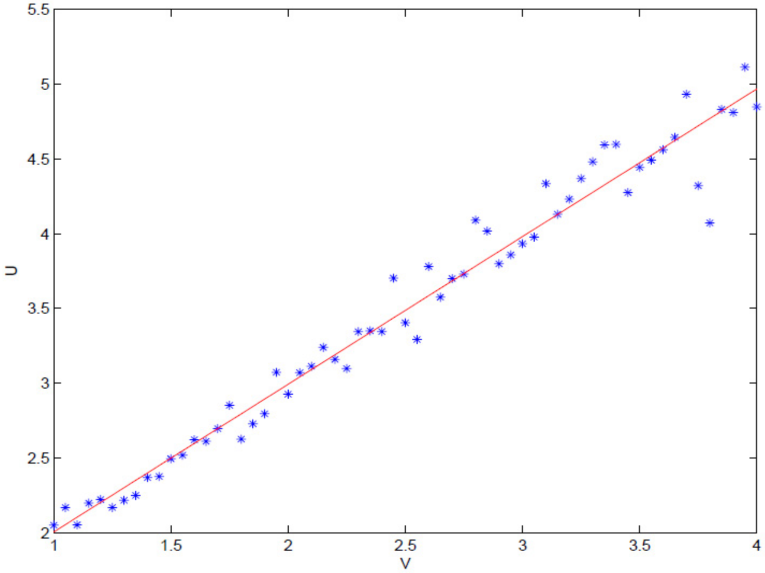
$$U = 0.9875V + 1.018$$

如果令 $\rho = \frac{1}{V^2}$可得

$$U = 0.9915V + 1.008$$

实际数据产生为

$$U = V + 1 + \mathrm{randn}(0, 0.005)V$$


5.5
5
4.5
4
3.5
3
2.5
2
U
1
1.5
2
2.5
3
3.5
4
V

例

由下列数据，试用一次、二次、三次和四次多项式进行曲线拟合，并给出相应的误差

j	0	1	2	3	4
x_j	0.00	0.25	0.50	0.75	1.00
$f(x_j)$	1.0000	1.2840	1.6487	2.1170	2.7183

先考虑二次多项式逼近，此时$n=2, m=4$，法方程为

$$\begin{bmatrix} 5 & 2.5 & 1.875 \\ 2.5 & 1.875 & 1.562 \\ 1.875 & 1.562 & 1.382 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 8.7680 \\ 5.4514 \\ 4.4015 \end{bmatrix}$$

解得$a_0 = 1.0051, a_1 = 0.86468, a_2 = 0.84316$。则二次拟合多项式为

$$s^* = P_2^* = 1.0051 + 0.86468x + 0.84316x^2$$

其误差为

$$\delta_2^* = \sqrt{\sum_{j=0}^{4} [f(x_j) - P_2^*(x_j)]^2} = 1.66 \times 10^{-2}$$

用一次多项式进行拟合，$n=1, m=4$，可得

$$P_1^*(x) = 0.8997 + 1.7078x$$

类似的$n=3, m=4$有

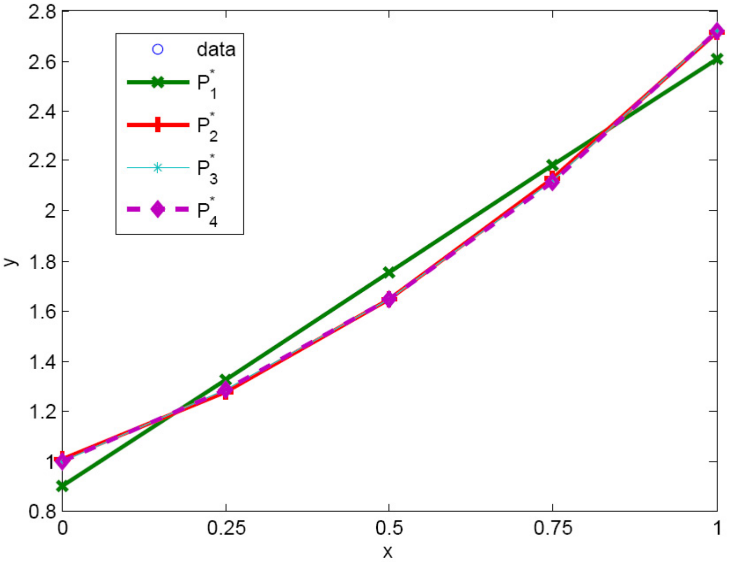
$$P_3^*(x) = 1.0000 + 1.0141x + 0.4253x^2 + 0.2789x^3$$

同样地$n=4, m=4$有

$$P_4^* = 1.0000 + 0.99868x + 0.5101x^2 + 0.1403x^3 + 0.0693x^4$$

相应的误差为

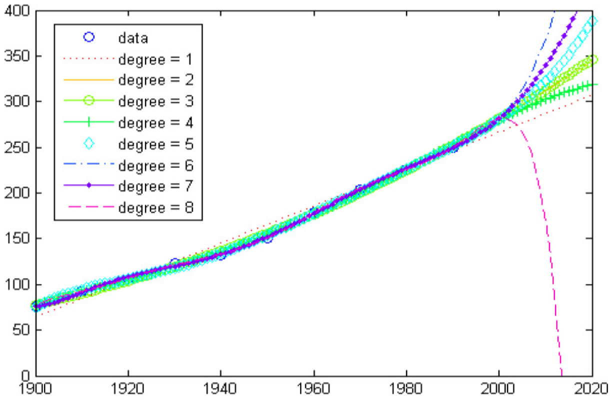
$$\begin{aligned} \delta_1^* &= = 0.916 \\ \delta_3^* &= 7.769 \times 10^{-3}. \end{aligned}$$


data
P_1^*
P_2^*
P_3^*
P_4^*
y
x
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0
0.25
0.5
0.75
1

模型与参数的选取

1. 线性最小二乘拟合的成功与否，与所选取的模型有很大的关系。
   - 模型所含的参数越多，平方误差会越小。
   - 若参数个数等于数据点个数，平方误差为零，但这并不意味模型很准，因为数据有噪音。
2. 完全吻合数据的模型亦代表此模型受噪音的影响最大，预测的准确度也会很差。
3. 「模型复杂度」（即可变参数的个数）和「预测准确度」是相互抗衡的两个因素。

美国人口预测



从上图可以看出，当多项式的次数越来越高时，「外插」(预测)常会出现不可信的结果。这说明选用的模型参数太多，虽然误差的平方和变小了，但是预测的可靠度也下降了

合适的模型

从上面的推导和例子看出，对于给定的数据$(x_k, f(x_k))$，
$k = 0, 1, \cdots, m$，选择合适的模型，亦即合适的函数空间Φ以及参数的个数至关重要。如果对给定的数据分布有一个大致的了解，对选用合适的模型有很大的帮助。
如果数据呈指数分布，即

$$s(x) = be^{ax},$$

这是一个非线性的模型。如果直接用曲线拟合的最小二乘法确定a, b:

$$F(a, b) = \sum_{j=0}^{m} \rho(x_j)[f(x_j) - be^{ax_j}]^2$$

极值的必要条件

$$
\begin{aligned}
\frac{\partial F}{\partial b} &= 2\sum_{j=0}^{m}\rho(x_j)[f(x_j)-be^{ax_j}](-e^{ax_j})=0\\
\frac{\partial F}{\partial a} &= 2\sum_{j=0}^{m}\rho(x_j)[f(x_j)-be^{ax_j}](-bx_je^{ax_j})=0
\end{aligned}
$$

得到一个非线性方程组

$$
\begin{cases}
\sum_{j=0}^{m}\rho(x_j)[f(x_j)-be^{ax_j}]e^{ax_j}=0\\
\sum_{j=0}^{m}\rho(x_j)[f(x_j)-be^{ax_j}]bx_je^{ax_j}=0
\end{cases}
$$

若数据呈指数分布的情况，可两边取对数

$$\ln s(x) = \ln b + ax$$

求出a和$\ln b$是一个线性问题，取$\Phi = \mathrm{span}(1, x)$，用线性最小二乘拟合得到$\ln s^*(x)$，变换回去得到

$$s^*(x) = e^{\ln s^*(x)}$$

注记

非线性的模型形式多样，在很多情况下可以通过变换变成线性模型，以及数据线性化，然后对线性化的数据有线性的最小二乘拟合。教材*P263*表*7.4* 给出了一些常用的可线性化的模型。

用$1, x, \cdots, x^n$来作最小二乘得到的法方程的系数矩阵一般是病态的，实际计算很困难。如果$\varphi_0, \varphi_1, \cdots, \varphi_n$在$\mathbb{X} = \{x_0, \cdots, x_m\}$上是正交的，即

$$(\varphi_k, \varphi_l) = \sum_{j=0}^{m} \rho(x_j)\varphi_k(x_j)\varphi_l(x_j) = \begin{cases} 0, & l \neq k, \\ A_k, & l = k. \end{cases}$$

其中$A_k = \sum_{j=0}^{m} \rho(x_j)[\varphi_k(x_j)]^2$。由法方程可得

$$a_k^* = \frac{(f, \varphi_k)}{(\varphi_k, \varphi_k)} = \frac{\sum_{j=0}^{m} \rho(x_j)f(x_j)\varphi_k(x_j)}{A_k}.$$

而在$\mathbb{X}$上可以用Gram-Schmidt方法来构造两两正交的函数。
设$x_0, x_1, \cdots, x_n$为给定节点，ρ为权函数，设$\rho(x_j) > 0$，$j = 0, 1, \cdots, m$。
可以用递推关系来得到关于$\rho(x_j), 0 \le j \le m$正交的多项式序列$\{\varphi_k, k = 0, 1, \cdots, n\}$。

令

$$
\begin{aligned}
\varphi_0(x) &= 1, \\
\varphi_1(x) &= (x-\alpha_1)\varphi_0(x), \\
\varphi_{k+1}(x) &= (x-\alpha_{k+1})\varphi_k - \beta_k\varphi_{k-1}(x), k=1,2,\cdots,n-1
\end{aligned}
$$

其中φ_k为k次多项式，其最高系数为1。

$$
\alpha_{k+1} = \frac{(x\varphi_k,\varphi_k)}{(\varphi_k,\varphi_k)} = \frac{\sum_{j=0}^{m}\rho(x_j)x_j\varphi_k^2(x_j)}{\sum_{j=0}^{m}\rho(x_j)\varphi_k^2(x_j)}, k=0,1,\cdots,n-1,
$$

$$
\beta_k = \frac{(\varphi_k,\varphi_k)}{(\varphi_{k-1},\varphi_{k-1})} = \frac{\sum_{j=0}^{m}\rho(x_j)\varphi_k^2(x_j)}{\sum_{j=0}^{m}\rho(x_j)\varphi_{k-1}^2(x_j)}, k=1,2,\cdots,n-1.
$$

定理

例

对下面的数据用正交多项式来作最小二乘拟合

j	0	1	2	3	4
x_j	0.00	0.25	0.50	0.75	1.00
$f(x_j)$	1.0000	1.2840	1.6487	2.1170	2.7183

解：用递推公式求出在点集$\{0.00, 0.25, 0.50, 0.75, 1.00\}$上关于$\rho(x_j) = 1, j = 0, 1, 2, 3, 4$的正交多项式$\varphi_0, \varphi_1, \varphi_2$：

$$\begin{aligned}
\varphi_0(x) &= 1, \alpha_1 = \frac{1}{2} \Rightarrow \varphi_1(x) = x - \frac{1}{2}; \\
\alpha_2 &= \frac{1}{2}, \beta_1 = \frac{1}{8} \Rightarrow \varphi_2(x) = (x - \frac{1}{2})(x - \frac{1}{2}) - \frac{1}{8}.
\end{aligned}$$

计算可得

$$\begin{aligned}
(\varphi_0, \varphi_0) &= 5, (f, \varphi_0) = 8.7680; (\varphi_1, \varphi_1) = 0.625, \\
(f, \varphi_1) &= 1.0674, (\varphi_2, \varphi_2) = 0.0547, (f, \varphi_2) = 0.0461;
\end{aligned}$$

直接有

$$\begin{aligned}
a_0^* &= 1.7536, a_1^* = 1.7079, a_2^* = 0.8437. \Rightarrow \\
s^*(x) &= 1.0051 + 0.8642x + 0.8437x^2
\end{aligned}$$

数据拟合

为讨论一般的线性最小二乘问题，可把数据拟合叙述如下：

设给定m个节点$t_1, t_2, \cdots, t_m \in [a, b]$以及在$m$个节点上的函数值$b_1, b_2, \cdots, b_m$。又设$\psi_i \in L^2_\rho[a, b], i = 1, 2, \cdots, n$为线性无关函数组。它们的线性组合

$$\psi(x, t) = x_1\psi_1(t) + x_2\psi_2(t) + \cdots + x_n\psi_n(t),$$

希望在$t_1, t_2, \cdots, t_m$上$\psi(x, t)$能“最佳的”逼近$b_1, b_2, \cdots, b_m$。

理想情况$\psi(x, t_i) = b_i, i = 1, \cdots, m$，亦即$A\boldsymbol{x} = \boldsymbol{b}$，其中

$$A = \begin{bmatrix} \psi_1(t_1) & \psi_2(t_1) & \cdots & \psi_n(t_1) \\ \psi_1(t_2) & \psi_2(t_2) & \cdots & \psi_n(t_2) \\ \vdots & \ddots & \ddots & \vdots \\ \psi_1(t_m) & \psi_2(t_m) & \cdots & \psi_n(t_m) \end{bmatrix} \in \mathbb{R}^{m \times n},$$

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n, \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \in \mathbb{R}^m$$

一般$m > n$，上述方程是超定的方程，不会有解存在。

所以一般情况下$\boldsymbol{Ax} \neq \boldsymbol{b}$,定义残量

$$r_i(x) = b_i - \sum_{j=1}^{n} x_j \psi_j(t_i), i = 1, 2, \cdots, m, \tag{0.6}$$

因而退而求其次：估计参数$x_1, x_2, \cdots, x_n$使得残量尽可能小。(0.6)表示成向量形式

$$\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{b} - A\boldsymbol{x}, , \boldsymbol{r}(\boldsymbol{x}) = (r_1(x), r_2(x), \cdots, r_m(x))^T \in \mathbb{R}^m$$

定义 (线性最小二乘)

给定矩阵$A \in \mathbb{R}^{m\times n}$及向量$\boldsymbol{b} \in \mathbb{R}^m$，确定$\boldsymbol{x}$使得

$$\|\boldsymbol{b} - A\boldsymbol{x}\|_2 = \|\boldsymbol{r}(x)\|_2 = \min_{\boldsymbol{y}\in\mathbb{R}^n} \|\boldsymbol{r}(\boldsymbol{y})\|_2 = \min_{\boldsymbol{y}\in\mathbb{R}^n} \|A\boldsymbol{y} - \boldsymbol{b}\|_2$$

称该问题为线性最小二乘问题。

定理

设$A \in \mathbb{R}^{m\times n}, m \geq n$，假定$A$为列满秩的，则$\forall \boldsymbol{x} \in \mathbb{R}^n$有

$$\|A\boldsymbol{x}^* - \boldsymbol{b}\|_2 \leq \|A\boldsymbol{x} - \boldsymbol{b}\|_2 \Longleftrightarrow A^T A\boldsymbol{x}^* = A^T\boldsymbol{b},$$

注记

前面的数据拟合中的法方程就是

$$A^T A\boldsymbol{x} = A^T\boldsymbol{b}$$

当A为列满秩的假设下，A^TA为对称正定矩阵，因此可以用Cholesky方法求解。

1. 计算$C = A^TA, \boldsymbol{d} = A^T\boldsymbol{b}$;
2. 作Cholesky分解$C = LL^T$;
3. 求解三角方程组$L\boldsymbol{y} = \boldsymbol{d}$和$L^T\boldsymbol{x} = \boldsymbol{y}$。

计算量为$n^2m + \frac{1}{3}n^3$，一般$m \gg n$，所以主要的计算量为n^2m。
算法的精度比较差，

$$A = \begin{bmatrix} 1 & 1 & 1 \\ \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix} \Rightarrow A^TA = \begin{bmatrix} 1+\epsilon^2 & 1 & 1 \\ 1 & 1+\epsilon^2 & 1 \\ 1 & 1 & 1+\epsilon^2 \end{bmatrix}$$

从条件数的角度考虑

$$\mathrm{cond}_2(A^TA) = [\mathrm{cond}(A)]^2$$

一般来说最小二乘问题条件数不会很好！
很多时候借助于QR分解来求解最小二乘问题。

定理

设$A \in \mathbb{R}^{m\times n}, m \geq n$是列满秩的，则存在一个唯一的正交阵$Q \in \mathbb{R}^{m\times n}(Q^TQ = I \in I^{n\times n})$和唯一的具有正对角元$r_{ii} > 0$的上三角阵$R \in \mathbb{R}^{n\times n}$使得

$$A = QR$$

设$A \in \mathbb{R}^{m\times n}, m \geq n$列满秩，$A = QR$，将$Q \in \mathbb{R}^{m\times n}$加上$m-n$个相互正交的列得到

$$[\underbrace{Q}_{n}, \underbrace{\tilde{Q}}_{m-n}]$$

则

$$\begin{aligned}
\|A\boldsymbol{x}-\boldsymbol{b}\|_2^2 &= \|[Q,\tilde{Q}]^T(A\boldsymbol{x}-\boldsymbol{b})\|_2^2 \\
&= \left\| \begin{bmatrix} Q^T \\ \tilde{Q}^T \end{bmatrix} (QR\boldsymbol{x}-\boldsymbol{b}) \right\|_2 \\
&= \left\| \begin{bmatrix} I^{n\times n} \\ O^{(m-n)\times n} \end{bmatrix} R\boldsymbol{x} - \begin{bmatrix} Q^T\boldsymbol{b} \\ \tilde{Q}^T\boldsymbol{b} \end{bmatrix} \right\|_2^2 \\
&= \left\| \begin{bmatrix} R\boldsymbol{x} - Q^T\boldsymbol{b} \\ \tilde{Q}^T\boldsymbol{b} \end{bmatrix} \right\|_2 \\
&= \|R\boldsymbol{x}-Q^T\boldsymbol{b}\|_2^2 + \|\tilde{Q}^T\boldsymbol{b}\|_2^2 \\
&\geq \|\tilde{Q}^T\boldsymbol{b}\|_2^2
\end{aligned}$$

因而$\boldsymbol{x}^*$是原来最小二乘问题的解当且仅当$\boldsymbol{x}^*$是下面问题的解

$$R\boldsymbol{x} = \boldsymbol{c}_1 = Q^T\boldsymbol{b}.$$

利用法方程也可以导出最小二乘解

$$\begin{aligned}
\boldsymbol{x} &= (A^TA)^{-1}A^T\boldsymbol{b} \\
&= (R^TQ^TQR)^{-1}R^TQ^T\boldsymbol{b} \\
&= (R^TR)^{-1}R^TQ^T\boldsymbol{b} \\
&= R^{-1}R^{-T}R^TQ^T\boldsymbol{b} \\
&= R^{-1}Q^T\boldsymbol{b}
\end{aligned}$$

基于QR分解的最小二乘法

从上面的推导得到利用QR分解求解最小二乘法的算法

用QR分解求解线性最小二乘问题

1. 计算A的QR分解$A = QR$;
2. 计算$\boldsymbol{c}_1 = Q^T\boldsymbol{b}$;
3. 求解上三角方程 $R\boldsymbol{x} = \boldsymbol{c}_1$来得到线性最小二乘解。

QR分解求解线性最小二乘问题的计算量为

$$2n^2m - \frac{2}{3}n^3$$

如果$m \gg n$，其计算量为法方程的2倍;
若$m = n$则计算量相同。

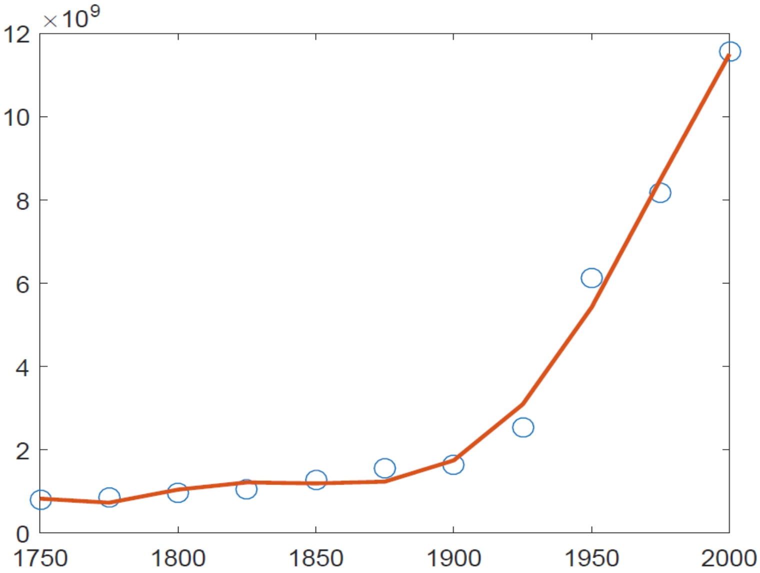
Matlab多项式拟合

Matlab中作多项式曲线拟合的命令是polyfit

```
year = (1750:25:2000)';
pop = 1e6*[791 856 978 1050 1262 1544 1650 2532 6122 8170 11560]';
T = table(year, pop);
plot(year,pop,'o')
[p, ,mu] = polyfit(T.year, T.pop, 5);
f = polyval(p,year,[],mu);
hold on
plot(year,f)
hold off
```

注记

Matlab 还有曲线拟合工具箱 *Curve Fitting Toolbox*。在*Matlab*中输入*cftool*即可。


×10^9
12
10
8
6
4
2
0
1750
1800
1850
1900
1950
2000

统计学中更一般的是fit命令：

load census;

f=fit(cdate,pop,'poly2')

plot(f,cdate,pop)

f = Linear model Poly2:

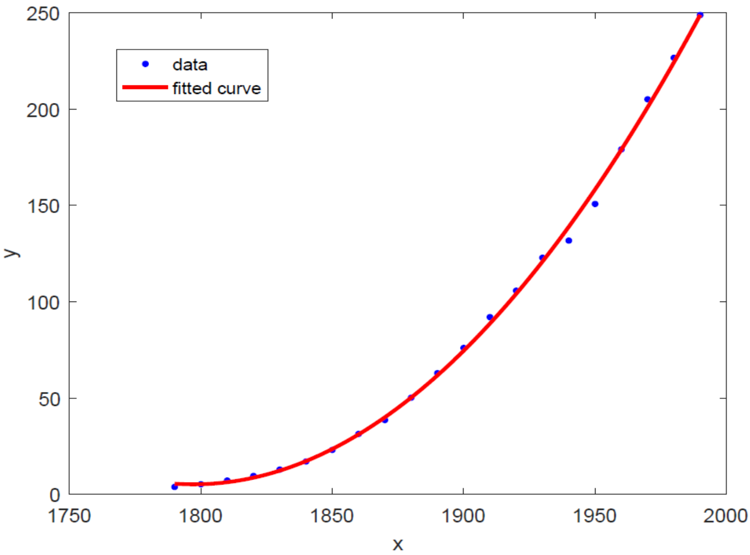
$f(x) = p1 * x^2 + p2 * x + p3$

Coefficients (with 95% confidence bounds):

$p1 = 0.006541(0.006124, 0.006958)$

$p2 = -23.51(-25.09, -21.93)$

$p3 = 2.113e + 04(1.964e + 04, 2.262e + 04)$


data
fitted curve
250
200
150
100
50
0
y
1750
1800
1850
1900
1950
2000
x

fit命令可以拟合二维曲面:

load franke

sf = fit([x, y],z,'poly23')

plot(sf,[x,y],z)

Linear model Poly23:

$sf(x,y) = p00 + p10*x + p01*y + p20*x^2 + p11*x*y + p02*y^2 + p21*x^2*y + p12*x*y^2 + p03*y^3$

Coefficients (with 95% confidence bounds):

p00 = 1.118 (0.9149, 1.321)

p10 = -0.0002941 (-0.000502, -8.623e-05)

p01 = 1.533 (0.7032, 2.364)

p20 = -1.966e-08 (-7.084e-08, 3.152e-08)

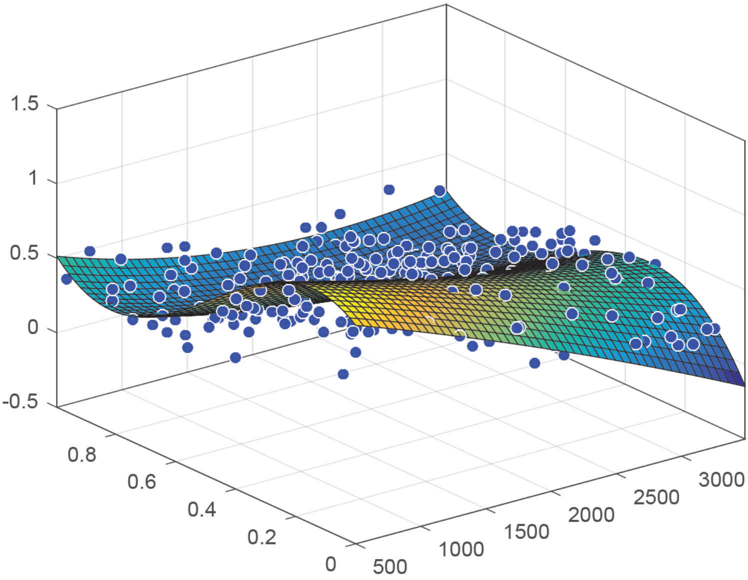
p11 = 0.0003427 (-0.0001009, 0.0007863)

p02 = -6.951 (-8.421, -5.481)

p21 = 9.563e-08 (6.276e-09, 1.85e-07)

p12 = -0.0004401 (-0.0007082, -0.0001721)

p03 = 4.999 (4.082, 5.917)


1.5
1
0.5
0
-0.5
0.8
0.6
0.4
0.2
0
500
1000
1500
2000
2500
3000

对线性的最小二乘问题，有

定理

设$A \in \mathbb{R}^{m\times n}, m \geq n$，假定$A$为列满秩的，则$\forall \boldsymbol{x} \in \mathbb{R}^n$有

$$\|A\boldsymbol{x}^* - \boldsymbol{b}\|_2 \leq \|A\boldsymbol{x} - \boldsymbol{b}\|_2 \Longleftrightarrow A^TA\boldsymbol{x}^* = A^T\boldsymbol{b},$$

若A不是列满秩的，法方程的解不唯一。此时则转而求解下面的极小化问题

$$\begin{aligned} \text{given} \quad & A \in \mathbb{R}^{m\times n}, \boldsymbol{b} \in \mathbb{R}^m, \\ \min_{\boldsymbol{x}\in\mathbb{R}^m} \|\boldsymbol{x}\|_2 \quad & \text{使得} \ \|A\boldsymbol{x} - \boldsymbol{b}\|_2 \leq \|A\hat{\boldsymbol{x}} - \boldsymbol{b}\|_2, \forall \hat{\boldsymbol{x}} \in \mathbb{R}^m \end{aligned}$$

这个问题可以用奇异值分解（SVD, singular value decomposition）来求解。

定理 (奇异值分解)

任意矩阵$A \in \mathbb{R}^{m\times n}$，令 $r = rank(A)$，则存在正交矩阵$U \in \mathbb{R}^{m\times r}$ 和 $V \in \mathbb{R}^{n\times r}$ 使得

$$A = U\Sigma V^T,$$

这里$\Sigma \in \mathbb{R}^{r\times r}$是对角阵，其对角元

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0,$$

此处

- $\sigma_i, i = 1, \cdots, r$称为$A$的奇异值；
- 相应的分解称为奇异值分解；
- U的列向量称为左奇异向量；
- V的列向量称为右奇异向量。

证明概要：

- 令

$$\sigma_1 = \|A\|_2 = \max_{\|\boldsymbol{x}\|_2=1} \|A\boldsymbol{x}\|_2,$$

则存在一对单位向量$(\boldsymbol{v}_1, \boldsymbol{u}_1)$满足：$A\boldsymbol{v}_1 = \sigma_1\boldsymbol{u}_1$。

- 令

$$\sigma_2 = \max_{\|\boldsymbol{x}\|_2=1, \boldsymbol{x}\perp\boldsymbol{v}_1} \|A\boldsymbol{x}\|_2.$$

同理有$A\boldsymbol{v}_2 = \sigma_2\boldsymbol{u}_2$。

- 直到找到向量$\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_r$和奇异值$\sigma_1, \sigma_2, \cdots, \sigma_r$使得

$$\max_{\boldsymbol{x}\perp\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_r\}, \|\boldsymbol{x}\|_2=1} \|A\boldsymbol{x}\|_2 = 0$$

时停止。

- 通过验证

$$Av = \left(\sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T\right) \boldsymbol{v}, \qquad \forall \boldsymbol{v} \in \mathbb{R}^n.$$

来证明分解

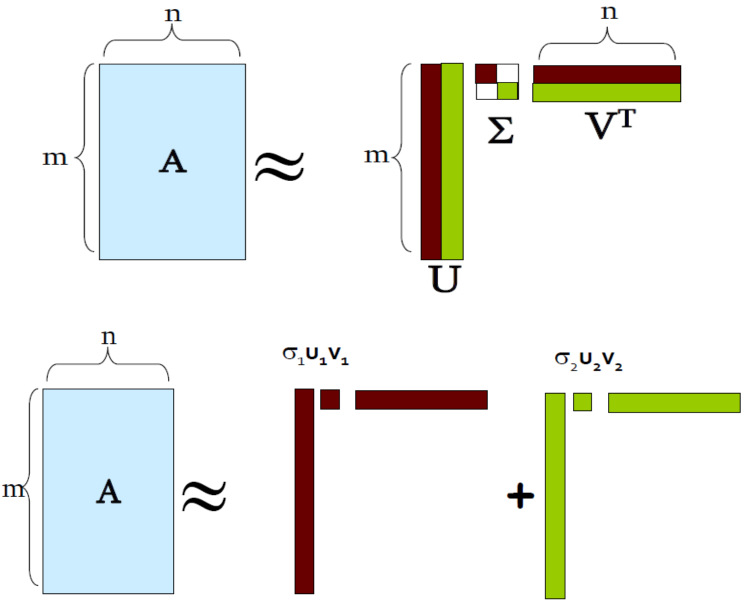
$$A = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T.$$

- 验证$\boldsymbol{u}_j$的正交性：

$$\begin{aligned}
(\boldsymbol{u}_i, \boldsymbol{u}_j) &= \left(\frac{A\boldsymbol{v}_i}{\sigma_i}, \frac{A\boldsymbol{v}_j}{\sigma_j}\right) \\
&= \frac{1}{\sigma_i\sigma_j}(A^T A\boldsymbol{v}_i, \boldsymbol{v}_j) \\
&= \frac{1}{\sigma_i\sigma_j}\left(\sum_{l=1}^{r} \sigma_l^2 \boldsymbol{v}_l \boldsymbol{v}_l^T \boldsymbol{v}_i, \boldsymbol{v}_j\right) \\
&= \delta_{ij}
\end{aligned}$$

SVD的性质

- $\mathrm{rank}(A) = r$，r是非零奇异值的个数。
- 矩阵A有奇异值展开
$$A = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \boldsymbol{v}_i^T.$$
- $\|A\|_2 = \sigma_1 =$最大的奇异值。
- $\|A\|_F = (\sum_{i=1}^{r} \sigma_i^2)^{1/2}$。
- 若$A \in \mathbb{R}^{n\times n}$非奇异，则$\|A^{-1}\|_2 = 1/\sigma_r$。


n
m
A
≈
n
m
Σ
V^T
U
n
m
A
≈
$\sigma_1 u_1 v_1$
+
$\sigma_2 u_2 v_2$

奇异向量

左奇异向量和右奇异向量分别满足

$$\begin{aligned} A\boldsymbol{v}_i &= \sigma_i \boldsymbol{u}_i, \\ A^T \boldsymbol{u}_j &= \sigma_j \boldsymbol{v}_j. \end{aligned}$$

则有

$$\begin{aligned} A^T A\boldsymbol{v}_i &= V\Sigma U^T U\Sigma V^T \boldsymbol{v}_i = \sigma_i^2 \boldsymbol{v}_i, \\ AA^T \boldsymbol{u}_i &= U\Sigma V^T V\Sigma U^T \boldsymbol{u}_i = \sigma_i^2 \boldsymbol{u}_i. \end{aligned}$$

- 右奇异向量$\boldsymbol{v}_i$是A^TA的特征向量。
- 左奇异向量$\boldsymbol{u}_i$是AA^T的特征向量。
- 可以通过求解A^TA和AA^T的特征值问题来得到奇异值分解。但是？

最小二乘问题中，若A是秩亏损的有下面的结论

推论

假定$A \in \mathbb{R}^{m\times n}, m \geq n,$而$rank(A) = r < n$。则存在一个$n-r$维的向量$\boldsymbol{x}$的集合极小化 $\|A\boldsymbol{x} - \boldsymbol{b}\|_2$。

证明： 因为满足方程$Az = 0$的解空间为$n-r$维，如果$\boldsymbol{x}$极小化$\|A\boldsymbol{x} - \boldsymbol{b}\|_2$则

$$\|A(\boldsymbol{x} + z) - \boldsymbol{b}\|_2 = \|A\boldsymbol{x} - \boldsymbol{b}\|_2$$

同样也是极小解。

注记

这说明当A不满秩时，最小二乘解不唯一。

若A秩亏损，即$A = U\Sigma V^T$的秩$r < n$，则A可表示为

$$A = [U, U_1] \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} [V, V_1]^T = U\Sigma V^T$$

这里

$$[U, U_1] \in \mathbb{R}^{m\times m}, \qquad [V, V_1] \in \mathbb{R}^{n\times n}$$

是正交矩阵。

$$\begin{aligned}
\|A\boldsymbol{x} - \boldsymbol{b}\|_2^2 &= \left\| \begin{bmatrix} U^T \\ U_1^T \end{bmatrix} (A\boldsymbol{x} - \boldsymbol{b}) \right\|_2^2 \\
&= \left\| \begin{bmatrix} U^T \\ U_1^T \end{bmatrix} (U\Sigma V^T\boldsymbol{x} - \boldsymbol{b}) \right\|_2^2 \\
&= \left\| \begin{bmatrix} \Sigma V^T\boldsymbol{x} - U^T\boldsymbol{b} \\ U_1^T\boldsymbol{b} \end{bmatrix} \right\|_2^2 \\
&= \|\Sigma V^T\boldsymbol{x} - U^T\boldsymbol{b}\|_2^2 + \|U_1^T\boldsymbol{b}\|_2^2
\end{aligned}$$

从上面的式子可知当

$$\Sigma V^T \boldsymbol{x} = U^T \boldsymbol{b}, \text{ 或者 } \boldsymbol{x} = V\Sigma^{-1}U^T\boldsymbol{b} + V_1\boldsymbol{z}$$

时$\|A\boldsymbol{x} - \boldsymbol{b}\|_2$达到极小，这里用到了

$$V^T V_1 \boldsymbol{z} = 0, \forall \boldsymbol{z}.$$

因为V和V_1相互正交，所以

$$\|\boldsymbol{x}\|_2^2 = \|V\Sigma^{-1}U^T\boldsymbol{b}\|_2^2 + \|V_1\boldsymbol{z}\|_2^2$$

则

$$\begin{aligned} \min_{\boldsymbol{x}} \|\boldsymbol{x}\|_2^2 &= \min_{\boldsymbol{x}} \left\{ \|V\Sigma^{-1}U^T\boldsymbol{b}\|_2^2 + \|V_1\boldsymbol{z}\|_2^2 \right\} \\ &= \|V\Sigma^{-1}U^T\boldsymbol{b}\|_2^2 \end{aligned}$$

亦即当$\boldsymbol{z} = 0$时$\|\boldsymbol{x}\|_2$达到极小。

最小二乘问题的极小模解为

$$x = V\Sigma^{-1}U^T b = A^+ b.$$

此处

定义 (Moore-Penrose 逆 (pseudo-inverse 伪逆，广义逆))

假设$A \in \mathbb{R}^{m\times n}$有奇异值分解$A = U\Sigma V^T$，则$A$的伪逆定义为

$$A^+ = V\Sigma^{-1}U^T,$$

易验证

$$A^+A = V\Sigma^{-1}\Sigma V^T = VI_rV^T = \sum_{i=1}^{r} v_i v_i^T$$

$$AA^+ = U\Sigma\Sigma^{-1}U^T = UI_rU^T = \sum_{i=1}^{r} u_i u_i^T$$

$$A = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T, \qquad A^+ = \sum_{i=1}^{r} \frac{1}{\sigma_i} \boldsymbol{v}_i \boldsymbol{u}_i^T,$$
$$AA^+A = A, \qquad A^+AA^+ = A^+$$

且AA^+和A^+A是对称矩阵。
当A列满秩时，最小二乘解为

$$\begin{aligned} \boldsymbol{x} &= (A^TA)^{-1}A^T\boldsymbol{b} \\ &= (V\Sigma^2V^T)^{-1}A^T\boldsymbol{b} \\ &= V\Sigma^{-2}V^TV\Sigma U^T\boldsymbol{b} \\ &= V\Sigma^{-1}U^T\boldsymbol{b} \\ &= A^+\boldsymbol{b}. \end{aligned}$$

最小二乘问题的条件数

假定测量数据有误差，亦即实际计算的是$\boldsymbol{b}+\boldsymbol{\delta b}$，则对最小二乘解有什么影响？

因为

$$(\boldsymbol{x}+\boldsymbol{\delta x}) = V\Sigma^{-1}U^T(\boldsymbol{b}+\boldsymbol{\delta b}),$$

所以有

$$\boldsymbol{\delta x} = V\Sigma^{-1}U\boldsymbol{\delta b}$$

而

$$\begin{aligned} \|V\Sigma^{-1}U\delta\boldsymbol{b}\|_2 &\leq \|\Sigma^{-1}\|_2\|\delta\boldsymbol{b}\|_2 \\ &= \delta\boldsymbol{b}/\sigma_r \end{aligned}$$

其中σ_r是最小奇异值。

Q：最小二乘问题是否是病态问题？

例

求矩阵$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}$ 的奇异值分解

在Matlab 中输入命令$[U, S, V] = svd(A)$可得

$$U = \begin{bmatrix} -0.1409 & 0.8247 & 0.5473 & -0.0221 \\ -0.3439 & 0.4263 & -0.7133 & 0.4373 \\ -0.5470 & 0.0278 & -0.2153 & -0.8085 \\ -0.7501 & -0.3706 & 0.3813 & 0.3932 \end{bmatrix}$$

$$S = \begin{bmatrix} 25.4624 & 0 & 0 \\ 0 & 1.2907 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.5045 & -0.7608 & -0.4082 \\ -0.5745 & -0.0571 & 0.8165 \\ -0.6445 & 0.6465 & -0.4082 \end{bmatrix}$$

因此

$$\sigma_1 = 25.4624, \sigma_2 = 1.2907, \sigma_3 = 0.$$

这说明矩阵是秩亏损的。

在Matlab中输入X=pinv(A)求得A的广义逆

$$X=\begin{bmatrix}-0.4833 & -0.2444 & -0.0056 & 0.2333\\ -0.0333 & -0.0111 & 0.0111 & 0.0333\\ 0.4167 & 0.2222 & 0.0278 & -0.1667\end{bmatrix}$$

假定观测数据为$b=1,2,3$，则其欧氏范数极小的最小二乘解为

$$\boldsymbol{x}=A^{+}\boldsymbol{b}=X\boldsymbol{b}=\begin{bmatrix}3.0000\\ 1.9167\\ 1.4333\end{bmatrix}$$

当然也可以求出A的QR分解，在Matlab中输入[Q,R]=qr(A)可得

$$
\begin{aligned}
Q &= \begin{bmatrix} -0.0776 & -0.8331 & 0.5473 & -0.0221 \\ -0.3105 & -0.4512 & -0.7133 & 0.4373 \\ -0.5433 & -0.0694 & -0.2153 & -0.8085 \\ -0.7762 & 0.3124 & 0.3813 & 0.3932 \end{bmatrix} \\
R &= \begin{bmatrix} -12.8841 & -14.5916 & -16.2992 \\ 0 & -1.0413 & -2.0826 \\ 0 & 0 & -0.0000 \\ 0 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

HITS(Hyperlink–Induced Topic Search)算法

Hub页面（枢纽页面）和Authority页面（权威页面）是HITS算法最基本的两个定义。

- 所谓“Authority”页面，是指与某个领域或者某个话题相关的高质量网页，比如搜索引擎领域，Google和百度首页即该领域的高质量网页，比如视频领域，优酷和土豆首页即该领域的高质量网页。
- 所谓“Hub”页面，指的是包含了很多指向高质量“Authority”页面链接的网页，比如hao123首页可以认为是一个典型的高质量“Hub”网页。

HITS算法的目的即是通过一定的技术手段，在海量网页中找到与用户查询主题相关的高质量“Authority”页面和“Hub”页面，尤其是“Authority”页面，因为这些页面代表了能够满足用户查询的高质量内容，搜索引擎以此作为搜索结果返回给用户。

SVD和HITS

枢纽值，指的是页面上所有导出链接指向页面的权威值之和。权威值是指所有导入链接所在的页面中枢纽之和。网页看成有向图的顶点，链接关系定义为有向边，有如下假设：

- 一个好的“Authority”页面会被很多好的“Hub”页面指向；
- 一个好的“Hub”页面会指向很多好的“Authority”页面。

各网页的权威值和枢纽值组成权威向量$\boldsymbol{a}$和$\boldsymbol{h}$。L是有向图的链接矩阵，则有

$$\boldsymbol{h} = L\boldsymbol{a}, \quad \boldsymbol{a} = L^T\boldsymbol{h},$$

所以

$$\boldsymbol{h} = LL^T\boldsymbol{h}, \quad \boldsymbol{a} = L^TL\boldsymbol{a}.$$

所以$\boldsymbol{h}$和$\boldsymbol{a}$分别是链接矩阵L的第一个左奇异向量和右奇异向量。

低秩逼近–Low-Rank Approximation

令$\mathcal{M}_{m,n}^{(r)}$是秩为r的$m \times n$阶矩阵，设$A \in \mathcal{M}_{m,n}^{(r)}$。想求$B \in \mathcal{M}_{m,n}^{(k)}$，使得

$$\|A - B\|_F = \min_{rank(C)=k} \|A - C\|_F, \quad \text{或} \quad \|A - B\|_2 = \min_{rank(C)=k} \|A - C\|_2.$$

定理

设$k < r$和

$$A_k = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T,$$

则

$$\min_{rank(B)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}.$$

为求解

$$\|A-B\|_F=\min_{rank(C)=k}\|A-C\|_F,$$

考虑

$$A=U\Sigma V^T,\quad A_k=U\Sigma_k V^T$$

其中

$$\Sigma_k=\begin{bmatrix}\sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_k & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0\end{bmatrix}$$

所以

$$\begin{aligned}\|A-A_k\|_F^2 &= \|U(\Sigma-\Sigma_k)V^T\|_F^2 = \|\Sigma-\Sigma_k\|_F^2 \\ &= \sigma_{k+1}^2+\cdots+\sigma_r^2\end{aligned}$$

如果想找A_k使得

$$\|A-A_k\|_F^2 \le \epsilon^2,\ \ \mathrm{rank}(A_k)=\min$$

则$A_k=U\Sigma_k V^T$有

$$\|A-A_k\|_F^2 = \sigma_{k+1}^2+\cdots+\sigma_r^2$$

若

$$\sigma_{k+1}^2+\cdots+\sigma_r^2 \le \epsilon^2,\quad \sigma_k^2+\cdots+\sigma_r^2 > \epsilon^2$$

则A_k是最优解，且

$$\|A^+-A_k^+\|_F^2 = \left(\frac{1}{\sigma_{k+1}^2}+\cdots+\frac{1}{\sigma_r^2}\right)$$

数据挖掘的主要工具：降维

降维的主要目标不仅仅是为了减小数据的数量（维度），而是

- 在进行下一步的数据分析[如人脸识别等]前，尽可能的去掉数据的噪音和冗余信息。
- 提取重要的“特征”和“参数”。

定义 (数据降维)

对数据$X=[\boldsymbol{x}_1,\cdots,x_n]\in\mathbb{R}^{m\times n}$求其低维的表示$Y=[\boldsymbol{y}_1\cdots,\boldsymbol{y}_n]\in\mathbb{R}^{d\times n}$。

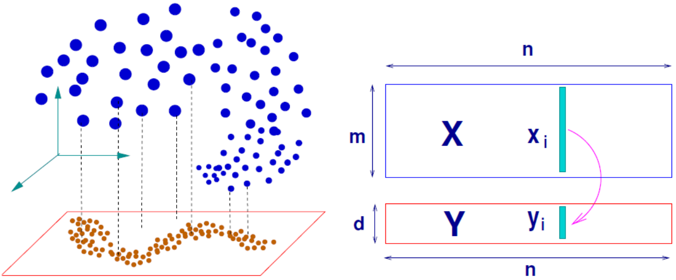
数据降维可通过构造映射：

$$\Phi:\boldsymbol{x}\in\mathbb{R}^m\longrightarrow\boldsymbol{y}\in\mathbb{R}^d.$$

使其满足

$$\Phi_i(\boldsymbol{x}_i)=\boldsymbol{y}_i,\quad i=1,\cdots,n.$$

来实现。



- Φ可能是线性的：$y_i = W^T x_i$，即$Y = W^T X$；
- Φ也有可能是非线性的（隐式）；
- Φ要求满足一定的要求：如距离最小，方差最大等。

主成分分析是求W使得投影数据的方差最大，即

$$\max_{W\in\mathbb{R}^{m\times d}, W^TW=I} \sum_{i=1}^{n} \left\| \boldsymbol{y}_i - \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{y}_j \right\|_2^2, \quad \boldsymbol{y}_i = W^T\boldsymbol{x}_i.$$

此问题和求如下的极大值问题等价：

$$\mathbf{Tr}\left[W^T(X-\boldsymbol{\mu}\boldsymbol{e}^T)(X-\boldsymbol{\mu}\boldsymbol{e}^T)^TW\right], \quad \boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i.$$

而W是协方差矩阵的主特征向量，同时也是$\overline{X} = X - \boldsymbol{\mu}\boldsymbol{e}^T$的左奇异向量。

主成分分析-Principal components analysis（PCA）

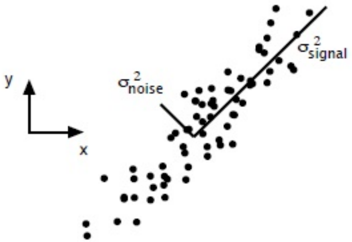
低秩逼近可以用来做主成分分析，主成分分析是一种分析、简化数据集的技术。

定义 (主成分分析)

*PCA*的数学定义是：一个正交变换，把数据变换到一个新的坐标系中，使得这一数据的任何投影的第一大方差在第一个坐标（称为第一主成分）上，第二大方差在第二个坐标（第二主成分）上，依次类推。

主成分分析经常用于减少数据集的维数，同时保持数据集中的对方差贡献最大的特征。其方法主要是通过对协方差矩阵进行特征分解，以得出数据的主成分（即特征向量）与它们的权值（即特征值或奇异值）。

如图所示，通过正交变换将X-Y坐标映射到signal和noise上：



一般来说，方差大的方向是信号的方向，方差小的方向是噪声的方向，在数据挖掘中或者数字信号处理中，往往要提高信号与噪声的比例，也就是信噪比。上图中通过坐标变换后，找出方差最大的方向为第一个坐标（signal），然后在其正交的平面上找出方差最大的方向为第二个坐标（noise）。这样就可以通过选取最大的 k个坐标方向来达到对维度（特征）的提炼。

数学上，m行n维的数据可用矩阵$A \in \mathbb{R}^{m\times n}$表示，坐标变换为$P \in \mathbb{R}^{n\times n}$：

$$A_{m\times n}P_{n\times n} = \widetilde{A}_{m\times n}.$$

PCA将上述中的维度n进行提炼，降维成$k(k < n)$：

$$A_{m\times n}P_{n\times k} = \widetilde{A}_{m\times k}.$$

可以通过SVD实现降维

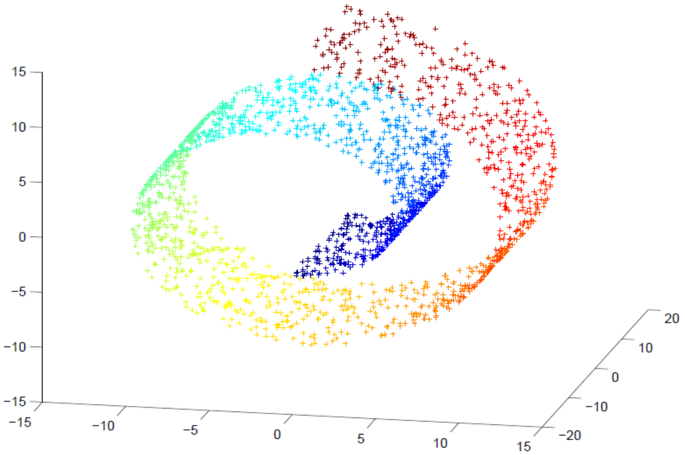
$$A_{m\times n} \approx U_{m\times k}\Sigma_{k\times k}V_{k\times n}^T$$

由于 $V_{n\times k}$ 为正交矩阵，所以

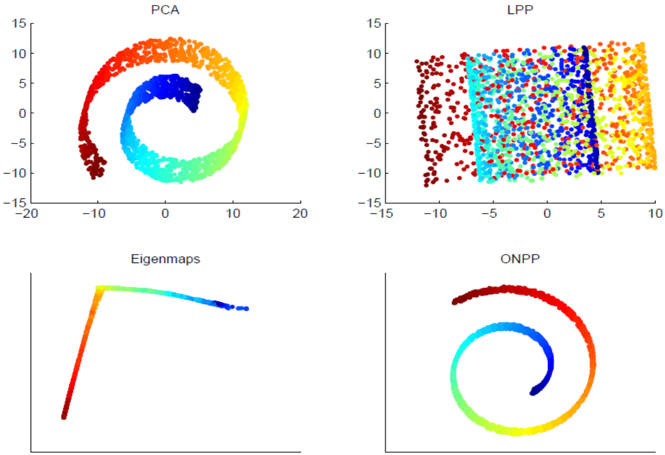
$$A_{m\times n}V_{n\times k} \approx U_{m\times k}\Sigma_{k\times k} \approx \widetilde{A}_{m\times k}.$$

这样通过SVD实现了PCA的坐标系变换和特征的提炼。

3D 数据



2D降维



此外，SVD还可以进行数据的压缩

$$U^T_{k\times m}A_{m\times n}\approx U^T_{k\times m}U_{m\times k}\Sigma_{k\times k}V^T_{k\times n}$$

同样由于 $U_{n\times k}$ 为正交矩阵，所以

$$U^T_{k\times m}A_{m\times n}\approx \Sigma_{k\times k}V^T_{k\times n}\approx \widetilde{A}_{k\times n}$$

如此则将m行数据压缩成k行数据。行压缩可以理解为，将一些相似的数据采样合并在一起，或者将一些没有太大价值的采样去掉。
可以看出，其实PCA几乎可以说是对SVD的一个包装，如果实现了SVD，那也就实现了PCA了，而且SVD可以得到两个方向的PCA，如果对A^TA进行特征值的分解，只能得到一个方向的PCA。

其中一个应用就是图像处理。

可以把$n \times n$的灰度图视为灰度强度的$n \times n$的矩阵。下面的命令先读入一个图像，把它转化成灰度图，最后把图像转化为矩阵。

```
A = imread(’koala.jpg’);   % read image from a file
A = rgb2gray(A);   % convert from color to grayscale
A = im2double(A);   % convert to double precision matrix
imshow(A);   % display the image in a window
```

下面的命令计算A的SVD，用奇异值分解去计算其秩r的最佳逼近，并给出图像。

```
[U, S, V] = svd(A);
for r = [1 2 5 10 25 50 100 298]
= U(:, 1:r) * S(1:r, 1:r) * V(:, 1:r)’;
imshow(Ar);
pause;
imwrite(Ar, sprintf(’koala-%d.jpg’, r));
end
```

主成分分析在图像压缩和人脸识别等领域有重要的应用。

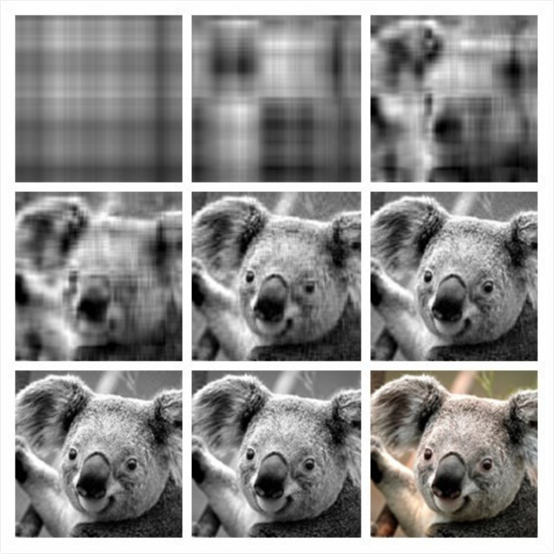
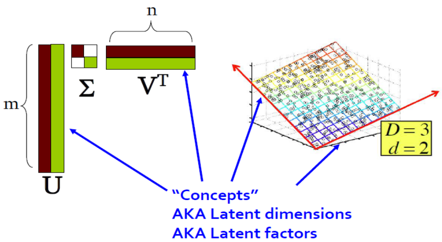


图: $k = 1, 2, 5, 10, 25, 50, 100, 298$

$$
\begin{array}{c|c}
 & \text{Matrix} \quad \text{Alien} \quad \text{Serenity} \quad \text{Casablanca} \quad \text{Amelie} \\
\begin{array}{c} \uparrow \\ \text{SciFi} \\ \downarrow \\ \uparrow \\ \text{Romnce} \\ \downarrow \end{array} &
\begin{bmatrix}
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\
\mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\
\mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\
0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\
0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\
0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2}
\end{bmatrix}
\end{array}
=
$$



$$
\begin{array}{c|c}
 & \text{Matrix} \quad \text{Alien} \quad \text{Serenity} \quad \text{Casablanca} \quad \text{Amelie} \\
\begin{array}{c} \uparrow \\ \text{SciFi} \\ \downarrow \\ \uparrow \\ \text{Romnce} \\ \downarrow \end{array} &
\begin{bmatrix}
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\
\mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\
\mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\
0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\
0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\
0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2}
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
\mathbf{0.13} & 0.02 & -0.01 \\
\mathbf{0.41} & 0.07 & -0.03 \\
\mathbf{0.55} & 0.09 & -0.04 \\
\mathbf{0.68} & 0.11 & -0.05 \\
0.15 & \mathbf{-0.59} & \mathbf{0.65} \\
0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\
0.07 & \mathbf{-0.29} & \mathbf{0.32}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 & 0 \\
0 & \mathbf{9.5} & 0 \\
0 & 0 & \mathbf{1.3}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\
0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09
\end{bmatrix}
$$

$$
\begin{array}{cc}
 & \begin{array}{ccccc}\text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie}\end{array} \\
\begin{array}{c}\uparrow\\ \text{SciFi}\\ \downarrow\\ \uparrow\\ \text{Romnce}\\ \downarrow\end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0\\
3 & 3 & 3 & 0 & 0\\
4 & 4 & 4 & 0 & 0\\
5 & 5 & 5 & 0 & 0\\
0 & 2 & 0 & 4 & 4\\
0 & 0 & 0 & 5 & 5\\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\end{array}
=
\begin{array}{c}
\begin{array}{ccc}\text{SciFi-concept} & \text{Romance-concept} & \end{array}\\
\begin{bmatrix}
\mathbf{0.13} & 0.02 & -0.01\\
\mathbf{0.41} & 0.07 & -0.03\\
\mathbf{0.55} & 0.09 & -0.04\\
\mathbf{0.68} & 0.11 & -0.05\\
0.15 & \mathbf{-0.59} & \mathbf{0.65}\\
0.07 & \mathbf{-0.73} & \mathbf{-0.67}\\
0.07 & \mathbf{-0.29} & \mathbf{0.32}
\end{bmatrix}
\end{array}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 & 0\\
0 & \mathbf{9.5} & 0\\
0 & 0 & \mathbf{1.3}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09\\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69}\\
0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09
\end{bmatrix}
$$

$$
\begin{array}{cc}
 & \begin{array}{ccccc}\text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie}\end{array} \\
\begin{array}{c}\uparrow\\ \text{SciFi}\\ \downarrow\\ \uparrow\\ \text{Romnce}\\ \downarrow\end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0\\
3 & 3 & 3 & 0 & 0\\
4 & 4 & 4 & 0 & 0\\
5 & 5 & 5 & 0 & 0\\
0 & 2 & 0 & 4 & 4\\
0 & 0 & 0 & 5 & 5\\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\end{array}
=
\begin{array}{c}
\begin{array}{ccc}\text{SciFi-concept} & & \end{array}\\
\begin{bmatrix}
\mathbf{0.13} & 0.02 & -0.01\\
\mathbf{0.41} & 0.07 & -0.03\\
\mathbf{0.55} & 0.09 & -0.04\\
\mathbf{0.68} & 0.11 & -0.05\\
0.15 & \mathbf{-0.59} & \mathbf{0.65}\\
0.07 & \mathbf{-0.73} & \mathbf{-0.67}\\
0.07 & \mathbf{-0.29} & \mathbf{0.32}
\end{bmatrix}
\end{array}
\times
\begin{array}{c}
\text{“strength” of the SciFi-concept}\\
\begin{bmatrix}
\mathbf{12.4} & 0 & 0\\
0 & \mathbf{9.5} & 0\\
0 & 0 & \mathbf{1.3}
\end{bmatrix}
\end{array}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09\\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69}\\
0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09
\end{bmatrix}
$$

V is "movie-to-concept" similarity matrix

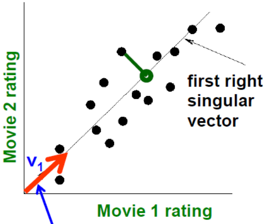
SciFi-concept

$$
\begin{array}{c}
 \\ \text{SciFi} \\ \\ \\ \text{Romnce} \\ \\
\end{array}
\begin{array}{c}
\begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
\mathbf{0.13} & 0.02 & -0.01 \\
\mathbf{0.41} & 0.07 & -0.03 \\
\mathbf{0.55} & 0.09 & -0.04 \\
\mathbf{0.68} & 0.11 & -0.05 \\
0.15 & \mathbf{-0.59} & \mathbf{0.65} \\
0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\
0.07 & \mathbf{-0.29} & \mathbf{0.32}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 & 0 \\
0 & \mathbf{9.5} & 0 \\
0 & 0 & \mathbf{1.3}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\
0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09
\end{bmatrix}
$$

SciFi-concept

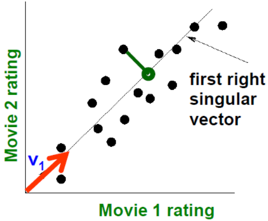
主成分分析的应用

- $A = U\,\Sigma\,V^T$ - example:
  - V: “movie-to-concept” matrix
  - U: “user-to-concept” matrix



$$
\begin{bmatrix}
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\
\mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\
\mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\
0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\
0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\
0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2}
\end{bmatrix}
=
\begin{bmatrix}
\mathbf{0.13} & 0.02 & -0.01 \\
\mathbf{0.41} & 0.07 & -0.03 \\
\mathbf{0.55} & 0.09 & -0.04 \\
\mathbf{0.68} & 0.11 & -0.05 \\
0.15 & \mathbf{-0.59} & \mathbf{0.65} \\
0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\
0.07 & \mathbf{-0.29} & \mathbf{0.32}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 & 0 \\
0 & \mathbf{9.5} & 0 \\
0 & 0 & \mathbf{1.3}
\end{bmatrix}
\times
\begin{bmatrix}
\boxed{\mathbf{0.56} \quad \mathbf{0.59} \quad \mathbf{0.56} \quad 0.09 \quad 0.09} \\
0.12 \quad -0.02 \quad 0.12 \quad \mathbf{-0.69} \quad \mathbf{-0.69} \\
0.40 \quad \mathbf{-0.80} \quad 0.40 \quad 0.09 \quad 0.09
\end{bmatrix}
$$

- $A = U\,\Sigma\,V^T$ - example:

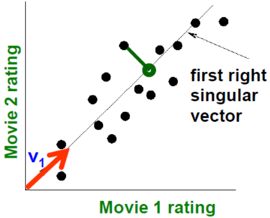


variance ('spread') on the v_1 axis

$$\begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\ \mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\ \mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\ 0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\ 0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\ 0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2} \end{bmatrix} = \begin{bmatrix} \mathbf{0.13} & 0.02 & -0.01 \\ \mathbf{0.41} & 0.07 & -0.03 \\ \mathbf{0.55} & 0.09 & -0.04 \\ \mathbf{0.68} & 0.11 & -0.05 \\ 0.15 & \mathbf{-0.59} & \mathbf{0.65} \\ 0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\ 0.07 & \mathbf{-0.29} & \mathbf{0.32} \end{bmatrix} \times \begin{bmatrix} \mathbf{12.4} & 0 & 0 \\ 0 & \mathbf{9.5} & 0 \\ 0 & 0 & \mathbf{1.3} \end{bmatrix} \times \begin{bmatrix} \mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\ 0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09 \end{bmatrix}$$

$A = U\,\Sigma\,V^T$ - example:

- $U\,\Sigma$: Gives the coordinates of the points in the projection axis



$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix}$$

Projection of users on the "Sci-Fi" axis $(U\,\Sigma)^T$:

$$\begin{bmatrix} \boxed{\begin{matrix} 1.61 \\ 5.08 \\ 6.82 \\ 8.43 \\ 1.86 \\ 0.86 \\ 0.86 \end{matrix}} & \begin{matrix} 0.19 \\ 0.66 \\ 0.85 \\ 1.04 \\ -5.60 \\ -6.93 \\ -2.75 \end{matrix} & \begin{matrix} -0.01 \\ -0.03 \\ -0.05 \\ -0.06 \\ 0.84 \\ -0.87 \\ 0.41 \end{matrix} \end{bmatrix}$$

数据降维

$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0\\
3 & 3 & 3 & 0 & 0\\
4 & 4 & 4 & 0 & 0\\
5 & 5 & 5 & 0 & 0\\
0 & 2 & 0 & 4 & 4\\
0 & 0 & 0 & 5 & 5\\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\approx
\begin{bmatrix}
0.13 & 0.02 & \cancel{-0.01}\\
0.41 & 0.07 & \cancel{-0.03}\\
0.55 & 0.09 & \cancel{-0.04}\\
0.68 & 0.11 & \cancel{-0.05}\\
0.15 & -0.59 & \cancel{0.65}\\
0.07 & -0.73 & \cancel{-0.67}\\
0.07 & -0.29 & \cancel{0.32}
\end{bmatrix}
\times
\begin{bmatrix}
12.4 & 0 & 0\\
0 & 9.5 & 0\\
0 & 0 & \cancel{1.3}
\end{bmatrix}
\times
\begin{bmatrix}
0.56 & 0.59 & 0.56 & 0.09 & 0.09\\
0.12 & -0.02 & 0.12 & -0.69 & -0.69\\
\cancel{0.40} & \cancel{-0.80} & \cancel{0.40} & \cancel{0.09} & \cancel{0.09}
\end{bmatrix}
$$

$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0\\
3 & 3 & 3 & 0 & 0\\
4 & 4 & 4 & 0 & 0\\
5 & 5 & 5 & 0 & 0\\
0 & 2 & 0 & 4 & 4\\
0 & 0 & 0 & 5 & 5\\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\approx
\begin{bmatrix}
0.13 & 0.02\\
0.41 & 0.07\\
0.55 & 0.09\\
0.68 & 0.11\\
0.15 & -0.59\\
0.07 & -0.73\\
0.07 & -0.29
\end{bmatrix}
\times
\begin{bmatrix}
12.4 & 0\\
0 & 9.5
\end{bmatrix}
\times
\begin{bmatrix}
0.56 & 0.59 & 0.56 & 0.09 & 0.09\\
0.12 & -0.02 & 0.12 & -0.69 & -0.69
\end{bmatrix}
$$

降维

$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\approx
\begin{bmatrix}
\mathbf{0.92} & \mathbf{0.95} & \mathbf{0.92} & 0.01 & 0.01 \\
\mathbf{2.91} & \mathbf{3.01} & \mathbf{2.91} & -0.01 & -0.01 \\
\mathbf{3.90} & \mathbf{4.04} & \mathbf{3.90} & 0.01 & 0.01 \\
\mathbf{4.82} & \mathbf{5.00} & \mathbf{4.82} & 0.03 & 0.03 \\
0.70 & \mathbf{0.53} & 0.70 & \mathbf{4.11} & \mathbf{4.11} \\
-0.69 & 1.34 & -0.69 & \mathbf{4.78} & \mathbf{4.78} \\
0.32 & \mathbf{0.23} & 0.32 & \mathbf{2.01} & \mathbf{2.01}
\end{bmatrix}
$$

Query

- Q: Find users that like ‘Matrix’
- A: Map query into a ‘concept space’ – how?

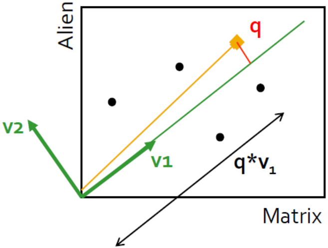
$$
\begin{array}{c|c}
 & \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\
\begin{array}{c} \\ \\ \text{SciFi} \\ \\ \\ \text{Romnce} \\ \\ \end{array} &
\begin{bmatrix}
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\
\mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\
\mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\
0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\
0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\
0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2}
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
\mathbf{0.13} & 0.02 & -0.01 \\
\mathbf{0.41} & 0.07 & -0.03 \\
\mathbf{0.55} & 0.09 & -0.04 \\
\mathbf{0.68} & 0.11 & -0.05 \\
0.15 & \mathbf{-0.59} & \mathbf{0.65} \\
0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\
0.07 & \mathbf{-0.29} & \mathbf{0.32}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 & 0 \\
0 & \mathbf{9.5} & 0 \\
0 & 0 & \mathbf{1.3}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\
0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09
\end{bmatrix}
$$

Query

- Q: Find users that like 'Matrix'
- A: Map query into a 'concept space' – how?

$$q = \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \\ \end{array}$$
$$q = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Project into concept space:
Inner product with each 'concept' vector $\mathbf{v}_i$



$$q_{concept} = q\ V$$

E.g.:

$$q = \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \\ \end{array}$$
$$q = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0.56 & 0.12 \\ 0.59 & -0.02 \\ 0.56 & 0.12 \\ 0.09 & -0.69 \\ 0.09 & -0.69 \end{bmatrix} = \begin{bmatrix} \overset{\text{SciFi-concept}}{2.8} & 0.6 \end{bmatrix}$$

movie-to-concept similarities (V)

$$d_{concept} = d\ V$$

E.g.:

$$q = \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \\ \end{array}$$
$$q = \begin{bmatrix} 0 & 4 & 5 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0.56 & 0.12 \\ 0.59 & -0.02 \\ 0.56 & 0.12 \\ 0.09 & -0.69 \\ 0.09 & -0.69 \end{bmatrix} = \begin{bmatrix} \overset{\text{SciFi-concept}}{5.2} & 0.4 \end{bmatrix}$$

movie-to-concept similarities (V)

潜在语义索引LSI(Latent Semantic Indexing)

LSA潜在语义分析的目的，就是要找出词(terms)在文档和查询中真正的含义，也就是潜在语义。降维是LSA分析中最重要的一步，通过降维，去除了文档中的“噪音”，也就是无关信息（比如词的误用或不相关的词偶尔出现在一起），语义结构逐渐呈现。

Index Words	Titles								
	T1	T2	T3	T4	T5	T6	T7	T8	T9
book			1	1					
dads						1			1
dummies		1						1	
estate							1		1
guide	1					1			
investing	1	1	1	1	1	1	1	1	1
market	1		1						
real							1		1
rich						2			1
stock	1		1					1	
value				1	1				

潜在语义索引LSI

- 左奇异向量的第一列表示每一个词的出现频繁程度；
- 右奇异向量中第一行表示每一篇文档中的出现词的个数的近似；
- 中间的矩阵表示词的分类和文章的类之间的相关性，即左奇异向量的一行与右奇异向量的一列的相关性。

book	0.15	-0.27	0.04
dads	0.24	0.38	-0.09
dummies	0.13	-0.17	0.07
estate	0.18	0.19	0.45
guide	0.22	0.09	-0.46
investing	0.74	-0.21	0.21
market	0.18	-0.30	-0.28
real	0.18	0.19	0.45
rich	0.36	0.59	-0.34
stock	0.25	-0.42	-0.28
value	0.12	-0.14	0.23

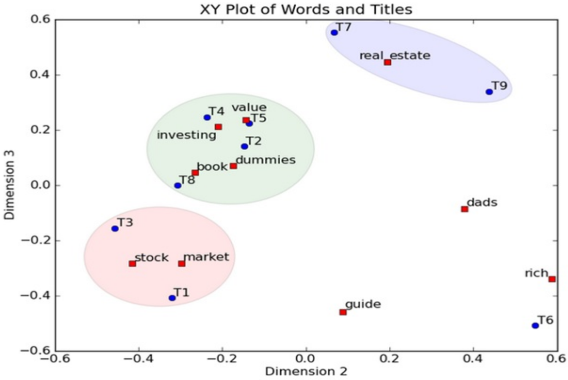
*

3.91	0	0
0	2.61	0
0	0	2.00

*

T1	T2	T3	T4	T5	T6	T7	T8	T9
0.35	0.22	0.34	0.26	0.22	0.49	0.28	0.29	0.44
-0.32	-0.15	-0.46	-0.24	-0.14	0.55	0.07	-0.31	0.44
-0.41	0.14	-0.16	0.25	0.22	-0.51	0.55	0.00	0.34

将左奇异向量和右奇异向量都取后2维（之前是3维的矩阵），投影到一个平面上，可以得到：



在图上，每一个红色的点，都表示一个词，每一个蓝色的点，都表示一篇文档，这样可以对这些词和文档进行聚类。

Matlab命令svds

- s = svds(A) returns a vector of the six largest singular values of matrix A.
- s = svds(A,k) returns the k largest singular values.
- s = svds(A,k,sigma) returns k singular values based on the value of sigma. For example, svds(A,k,'smallest') returns the k smallest singular values.
- s = svds(A,k,sigma,opts) additionally specifies options using a structure.
- s = svds(Afun,n,___) specifies a function handle Afun instead of a matrix A.
- [U,S,V] = svds(___) returns the left singular vectors U, diagonal matrix S of singular values, and right singular vectors V.
- [U,S,V,flag] = svds(___) also returns a convergence flag. If flag is 0, then all the singular values converged.

计算稀疏矩阵最大和最小奇异值

计算稀疏矩阵的最大奇异值

A = delsq(numgrid('C',15));
s = svds(A)
s =
7.8666
7.7324
7.6531
7.5213
7.4480
7.3517

计算稀疏矩阵的最小奇异值

A = delsq(numgrid('C',15));
s =svds(A,5,'smallest')
s =
0.5520
0.4787
0.3469
0.2676
0.1334

具体可以在Matlab命令窗口敲： help svds 看帮助。