



deepshare.net

深度之眼

法律声明

本课件包括演示文稿、示例、代码、题库、视频和声音等内容，深度之眼和讲师拥有完全知识产权；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或者机构不得盗版、复制、仿造其中的创意和内容，我们保留一切通过法律手段追究违反者的权利。

课程详情请咨询

- 微信公众号：深度之眼
- 客服微信号：deepshare0920



公众号



微信

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料



deepshare.net

深度之眼

《 Deep Learning 》

数学基础

导师: Johnson

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料

数学基础



Machine Learning Basics

1. 矩阵对角化, SVD分解以及应用
2. 逆矩阵, 伪逆矩阵
3. PCA原理与推导
4. 极大似然估计, 误差的高斯分布与最小二乘估计的等价性
5. 最优化, 无约束, 有约束, 拉格朗日乘子的意义, KKT条件

1. 矩阵对角化, SVD分解及应用



deepshare.net

深度之眼

实用性质:

$$A(B+C)=AB+AC \quad (\text{分配率})$$

$$A(BC)=(AB)C \quad (\text{结合律})$$

$$AB \neq BA \quad (\text{一般不满足交换律})$$

$$(AB)^T=B^TA^T \quad (\text{转置})$$

$$x^Ty=(x^Ty)^T=y^Tx \quad (\text{转置}) \quad \text{学员附注: } x,y \text{ 都是列向量, } x^T \text{ 是行向量, } x^Ty \text{ 的结果是一个标量}$$

关注公众号深度之眼, 后台回复花书, 获取AI必学书籍及完整学习资料



单位矩阵：任意向量或矩阵和单位矩阵相乘，都不会改变，记为 I 。所有沿主对角线的元素都是1，而所有其他位置的元素都是 0。

$$I_1 = [1], I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \dots, I_n = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

矩阵逆：矩阵（方阵）的逆满足如下条件

$$A^{-1}A = AA^{-1} = I_n$$



矩阵B（方阵）的对角化 $P^{-1}AP = B$

其中A为对角矩阵，P为单位正交矩阵

一般的矩阵不一定能对角化，但是对称矩阵一定可以对角化(特别是对称正定矩阵，得到的 λ_i 都是正数)

曾经一道面试题矩阵的压缩表示
最小 $n+1$

$$P^T P = P P^T = I$$
$$P^T = P^{-1}$$

所以: $B = P^T A P$

设 $P^T = (u_1, u_2, \dots, u_n), u_i \in \mathbb{R}^n$

$$A = \begin{pmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{pmatrix}$$

$$\text{则 } B = (u_1, u_2, \dots, u_n) \begin{pmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{pmatrix} \begin{pmatrix} u_1^T \\ \vdots \\ u_n^T \end{pmatrix}$$

$$= \lambda_1 u_1 u_1^T + \lambda_2 u_2 u_2^T + \lambda_3 u_3 u_3^T + \dots + \lambda_n u_n u_n^T$$



$$\begin{pmatrix} & & \\ & & \\ & & \end{pmatrix}_{n \times 1} \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix}_{1 \times n}$$



一般矩阵的svd分解: $A^T A$ 为对称正定矩阵
 $(A^T A)^T = A^T (A^T)^T = A^T A \Rightarrow$ 对称性
 $x^T A^T A x = (x^T A^T)(Ax) = (Ax)^T (Ax) \geq 0 \Rightarrow$ 正定性

$$A_{m \times n}$$

$$(A^T A)_{n \times n} = U^T D U$$

$$(A A^T)_{m \times m} = V^T D V$$

$$\Rightarrow A_{m \times n} = V_{m \times m}^T \begin{pmatrix} \lambda_1^{\frac{1}{2}} & & \\ & \lambda_2^{\frac{1}{2}} & \\ & & \ddots \end{pmatrix} U_{n \times n}$$

其中 λ_i 为 D 的对角元素

$$\text{令 } V_{m \times m}^T = (v_1, v_2, \dots, v_m)$$

$$U_{n \times n}^T = (u_1, u_2, \dots, u_n)$$

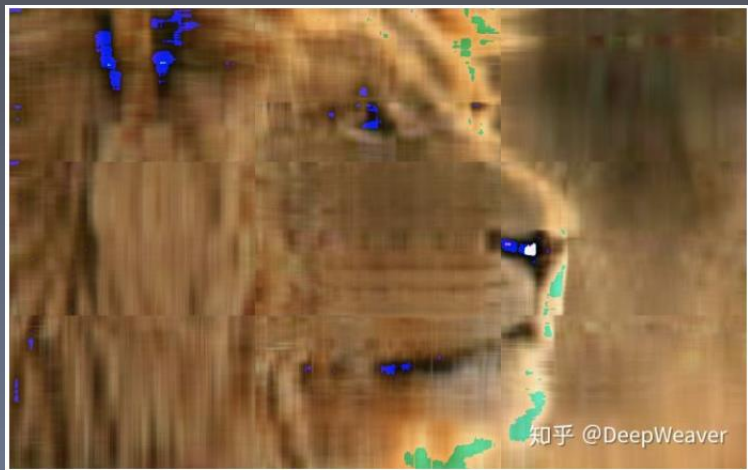
$$\Rightarrow A_{m \times n} = \lambda_1^{\frac{1}{2}} v_1 u_1^T + \lambda_2^{\frac{1}{2}} v_2 u_2^T + \dots$$

图像的压缩存储最小需要 $m+n+1$

当压缩存储量为 $(m+n+1)*K$ 时, 误差为

$$error = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^{\min(m,n)} \lambda_i}$$

$$A_{m \times n} = \lambda_1^{\frac{1}{2}} v_1 u_1^T + \lambda_2^{\frac{1}{2}} v_2 u_2^T + \dots$$



保留了前10个(压缩率122)



前30个(压缩率31)



前50个(压缩率17)

传统网络图片传输与现代传输的原理



关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料

$$A_{m \times n} = \lambda_1^{\frac{1}{2}} v_1 u_1^T + \lambda_2^{\frac{1}{2}} v_2 u_2^T + \dots$$



deepshare.net

深度之眼

$$A_{200 \times 100} = \lambda_1^{\frac{1}{2}} \begin{pmatrix} v_1 \\ \vdots \\ v_{100} \end{pmatrix}_{200 \times 1} \begin{pmatrix} u_1 & \dots & u_{100} \end{pmatrix}_{1 \times 100} + \lambda_2^{\frac{1}{2}} \begin{pmatrix} v_2 \\ \vdots \\ v_{100} \end{pmatrix}_{200 \times 1} \begin{pmatrix} u_2 & \dots & u_{100} \end{pmatrix}_{1 \times 100} + \dots$$

$$= \begin{pmatrix} v_1 & v_2 & \dots & v_{100} \end{pmatrix}_{200 \times 100} \begin{pmatrix} \lambda_1^{\frac{1}{2}} & & & \\ & \lambda_2^{\frac{1}{2}} & & \\ & & \ddots & \\ & & & \lambda_{100}^{\frac{1}{2}} \end{pmatrix}_{100 \times 100} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{100} \end{pmatrix}_{100 \times 100}$$

$$\approx M_{200 \times 100} N_{100 \times 100}$$

$A_{200 \times 100} X_{100 \times 1}$ 算 200×100 次乘法

$M_{200 \times 100} N_{100 \times 100} X_{100 \times 1}$ 算 $10 \times 100 + 10 \times 200 = 3000$ 次

能够极大的减小算法复杂度，
在深度神经网络中有着广泛
的应用

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料



2. 逆矩阵, 伪逆矩阵, 最小二乘解, 最小范数解

$$x_1, x_2, \dots, x_N, x_i \in \mathbb{R}^n$$

$$y_1, y_2, \dots, y_N, y_i \in \mathbb{R}^1$$

$$y_1 = x_{11}a_1 + x_{12}a_2 + \dots + x_{1n}a_n$$

$$y_2 = x_{21}a_1 + x_{22}a_2 + \dots + x_{2n}a_n$$

$$\vdots$$

$$y_N = x_{N1}a_1 + x_{N2}a_2 + \dots + x_{Nn}a_n$$

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nn} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

$$X_{N \times n} a_{n \times 1} = Y_{N \times 1}$$

当 $N = n$ 且 $X_{N \times n}$ 可逆时:

$$a = X^{-1}Y_S$$

一般情况: $N \neq n$

$$\min \|xa - Y\|^2 = J \quad \frac{\partial J}{\partial a} = x^T(xa - Y) = 0$$

$$x^T xa = x^T Y \quad x^T x \text{ 是否可逆?}$$

1. $N > n$

如 $N = 5, n = 3$ $(x^T x)_{3 \times 3}$ 一般是可逆的

$$a = (x^T x)^{-1} x^T Y$$

2. $N < n$

如 $N = 3, n = 5$ $(x^T x)_{5 \times 5}$

$$R(x^T x) \leq R(x) \leq 3$$

故 $x^T x$ 不可逆



此刻 $J = \|xa - Y\|^2 + \lambda \|a\|^2$

$$\frac{\partial J}{\partial a} = x^T x a - x^T Y + \lambda a = 0$$

$$(x^T x + \lambda I)a = x^T Y$$

$$a = (x^T x + \lambda I)^{-1} x^T Y$$

$x^T x$ 为对称矩阵可对角化

$$x^T x = P^{-1} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} P$$

$$|x^T x| = \lambda_1 \lambda_2 \cdots \lambda_n$$

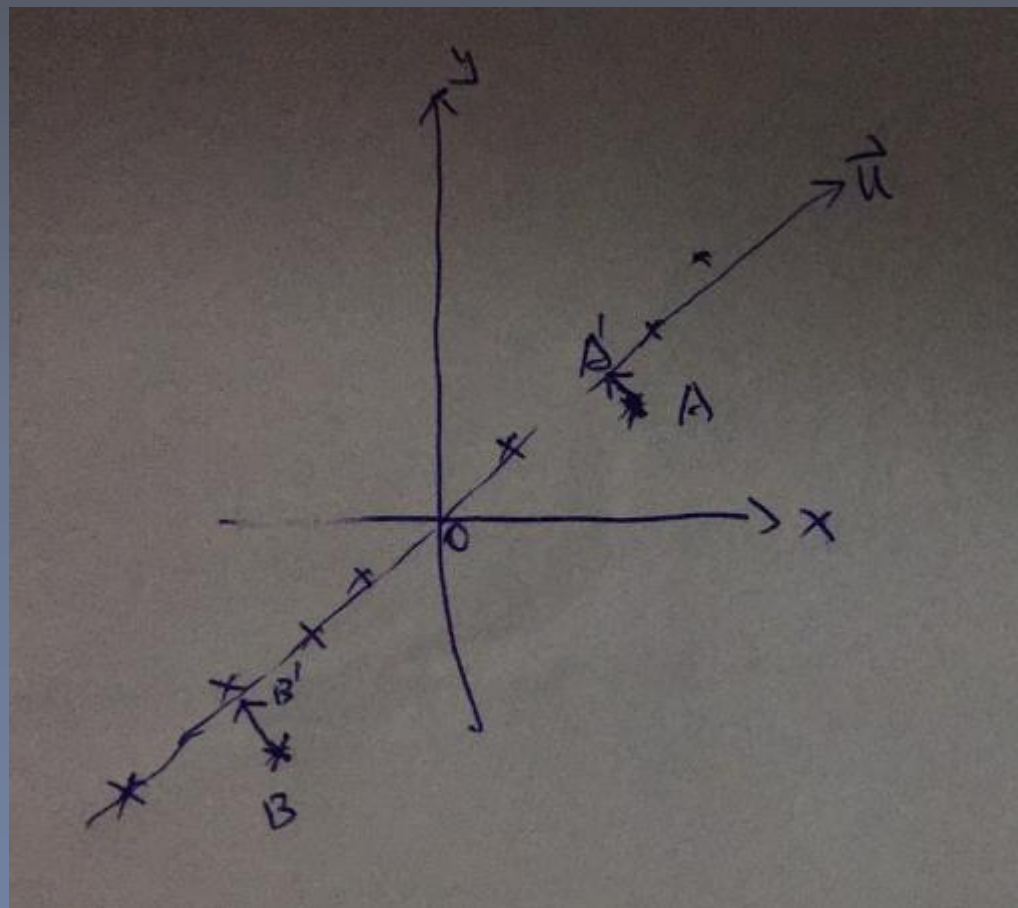
1. $a^T (x^T x) a = (xa)^T (xa) \geq 0 \rightarrow \lambda_i \geq 0$

$|x^T x|$ 仍可能为0, 不一定可逆

2. $a^T (x^T x + \lambda I) a = (xa)^T (xa) + \lambda a^T a > 0 \rightarrow \lambda_i > 0$

$|x^T x + \lambda I| > 0$ 恒成立, 一定可逆

3.PCA原理与推导



PCA仍然是一种数据压缩的算法

A点需要 x, y 两个坐标来表示，假设A在向量 u 上面的投影点为 A' ，则 A' 仅仅需要一个参数就能表示，就是 OA' 的长度(即 A' 在 u 上的坐标)，我们就想着用 A' 来替换A，这样 N 个点(原来要 $2 \times N$ 个参数)，现在只需要 $(N+2)$ 个参数(u 也需要2个参数)

但是此时就带来了误差，如 AA' 和 BB' ，所以我们要能够找到这样一个方向 u ，使得所有原始点与投影点之间的误差最小

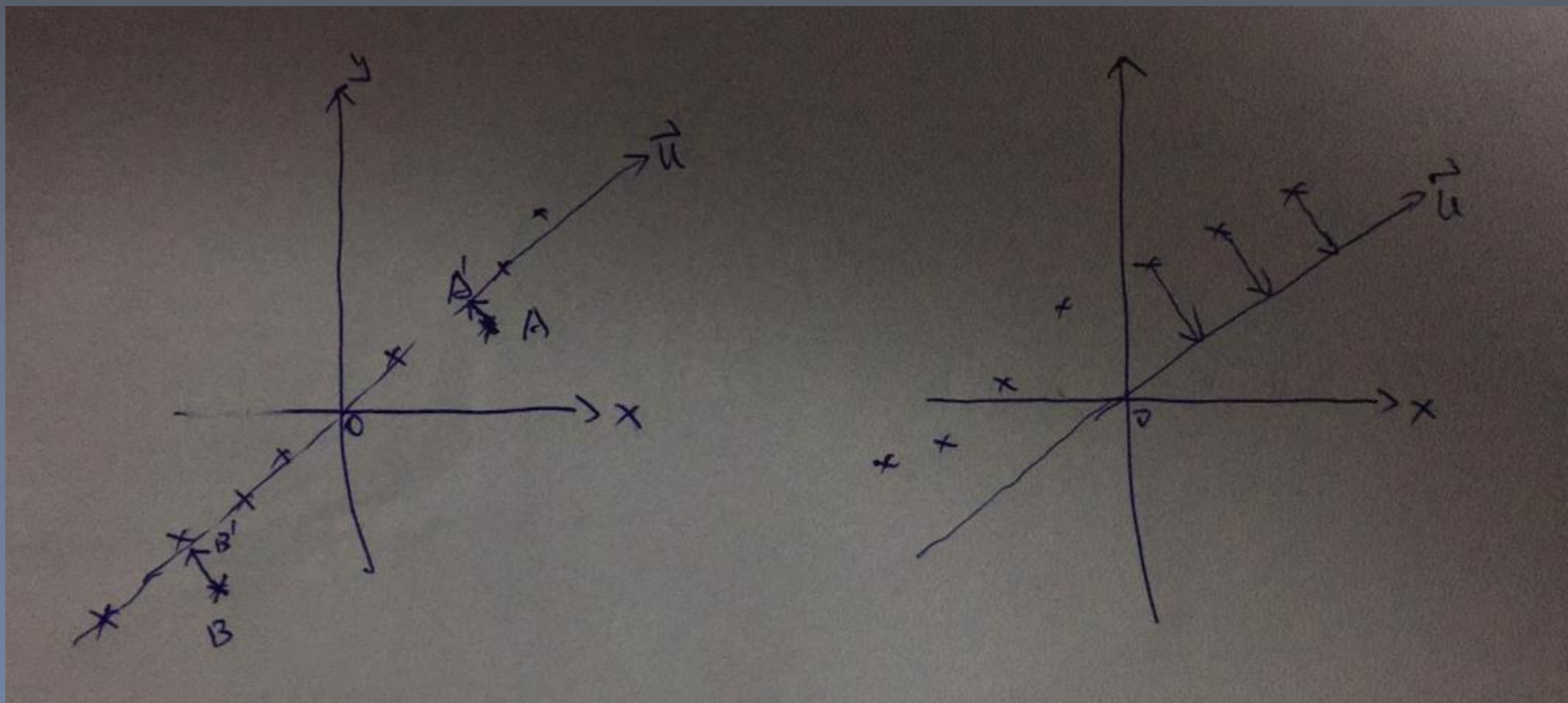
最小重构误差

样本点中心化

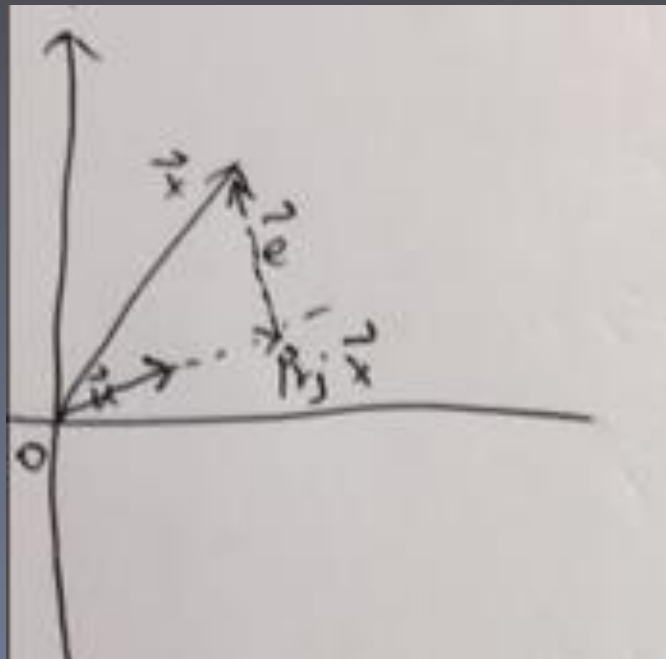


deepshare.net

深度之眼



关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料



$$\vec{e} = \vec{x} - p_{rj} \vec{x}$$

$$= \vec{x} - \langle \vec{x}, \vec{u} \rangle \vec{u}$$

$$= x - (x^T u)u; x, u \in \mathbb{R}^n \quad \text{且} \|u\| = 1, u^T u = 1$$

$$J = \|\vec{e}\|^2 = e^T e = [x - (x^T u)u]^T [x - (x^T u)u]$$

$$= [x^T - (x^T u)u^T][x - (x^T u)u]$$

$$= x^T x - (x^T u)(x^T u) - (x^T u)(x^T u) + (x^T u)^2 u^T u$$

$$= \|x\|^2 - (x^T u)^2 - (x^T u)^2 + (x^T u)^2$$

$$= \|x\|^2 - (x^T u)^2$$

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料

$$\max(x^T u)^2$$

$$\Leftrightarrow \max(x^T u)(x^T u) \Leftrightarrow \max(u^T x)(x^T u)$$

$$\Leftrightarrow \max u^T (x x^T) u$$

共有 N 个样本

$$\max \sum_{i=1}^N u^T (x_i x_i^T) u = u^T \left(\sum_{i=1}^N x_i x_i^T \right) u, \text{ 且 } \|u\| = 1$$

✗

$$\max u^T (X) u, \text{ st: } \|u\| = 1$$

$$L(u, \lambda) = u^T X u + \lambda(1 - u^T u)$$

$$\frac{\partial L}{\partial u} = 0 \Rightarrow Xu - \lambda u = 0$$

$$Xu = \lambda u$$

$$\frac{\partial L}{\partial \lambda} = 0 \Rightarrow u^T u = 1$$

极大似然估计，误差的高斯分布与最小二乘估计的等价性



deepshare.net

深度之眼

假设随机变量 $X \sim P(x; \theta)$

现有样本 x_1, x_2, \dots, x_N

定义似然函数为 $\tilde{L} = P(x_1; \theta)P(x_2; \theta) \cdots P(x_N; \theta)$

对数似然函数为 $L = \ln \tilde{L} = \ln[P(x_1; \theta)P(x_2; \theta) \cdots P(x_N; \theta)]$

极大似然估计为 $\max L$

高斯分布: $P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-u)^2}{2\sigma^2}}$

样本 x_1, x_2, \dots, x_N

$$L = \ln \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_1-u)^2}{2\sigma^2}} \dots \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_N-u)^2}{2\sigma^2}} \right]$$

$$L = -N \ln \sqrt{2\pi} - N \ln \sigma - \left[\frac{(x_1-u)^2}{2\sigma^2} + \dots + \frac{(x_N-u)^2}{2\sigma^2} \right]$$

$$\frac{\partial L}{\partial u} = 2(x_1-u) + 2(x_2-u) + \dots + 2(x_N-u) = 0$$

$$\Rightarrow u = \frac{x_1 + x_2 + \dots + x_N}{N}$$

$$\frac{\partial L}{\partial \sigma} = 0 \Rightarrow -\frac{N}{\sigma} + \frac{\sum_{i=1}^N (x_i - u)^2}{\sigma^3} = 0$$

$$\Rightarrow \sigma^2 = \frac{\sum_{i=1}^N (x_i - u)^2}{N}$$



deepshare.net

深度之眼

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料

$$x_1, x_2, \dots, x_N, x_i \in \mathbb{R}^n$$

$$y_1, y_2, \dots, y_N, y_i \in \mathbb{R}$$

$$y_i = w^T x_i, w \in \mathbb{R}^n$$

$$\text{拟合误差: } e_i = y_i - w^T x_i$$

$$\text{若设 } e_i \sim N(0, 1)$$

$$\text{即 } e_i \sim \frac{1}{\sqrt{2\pi}} e^{-\frac{e_i^2}{2}}$$

$$\begin{aligned} \text{似然函数 } L &= \ln \left[\frac{1}{\sqrt{2\pi}} e^{-\frac{e_1^2}{2}} \frac{1}{\sqrt{2\pi}} e^{-\frac{e_2^2}{2}} \dots, \frac{1}{\sqrt{2\pi}} e^{-\frac{e_N^2}{2}} \right] \\ &= -N \ln \sqrt{2\pi} - \frac{1}{2} (e_1^2 + e_2^2 + \dots + e_N^2) \end{aligned}$$

最大化 L 等价于

$$\text{最小化 } (e_1^2 + e_2^2 + \dots + e_N^2)$$

$$J = \min (y_1 - w^T x_1)^2 + (y_2 - w^T x_2)^2 + \dots + (y_N - w^T x_N)^2$$



$$J = \min (y_1 - w^T x_1)^2 + (y_2 - w^T x_2)^2 + \cdots + (y_N - w^T x_N)^2$$

$$\frac{\partial J}{\partial w} = -2(y_1 - w^T x_1)x_1 + \cdots + (y_N - w^T x_N)x_N = 0$$

$$\sum_{i=1}^N w^T x_i x_i = \sum_{i=1}^N x_i y_i$$

$$w^T x_i x_i = x_i (w^T x_i) = x_i (x_i^T w) = (x_i x_i^T) w$$

$$w = \left(\sum_{i=1}^N x_i y_i \right) \left(\sum_{i=1}^N x_i^T x_i \right)^{-1}$$

对比之前 $a = (x^T x)^{-1} x^T Y$

最优化，无约束，有约束，拉格朗日乘子的意义，KKT条件



deepshare.net

深度之眼

无约束优化问题是机器学习中最普遍、最简单的优化问题。

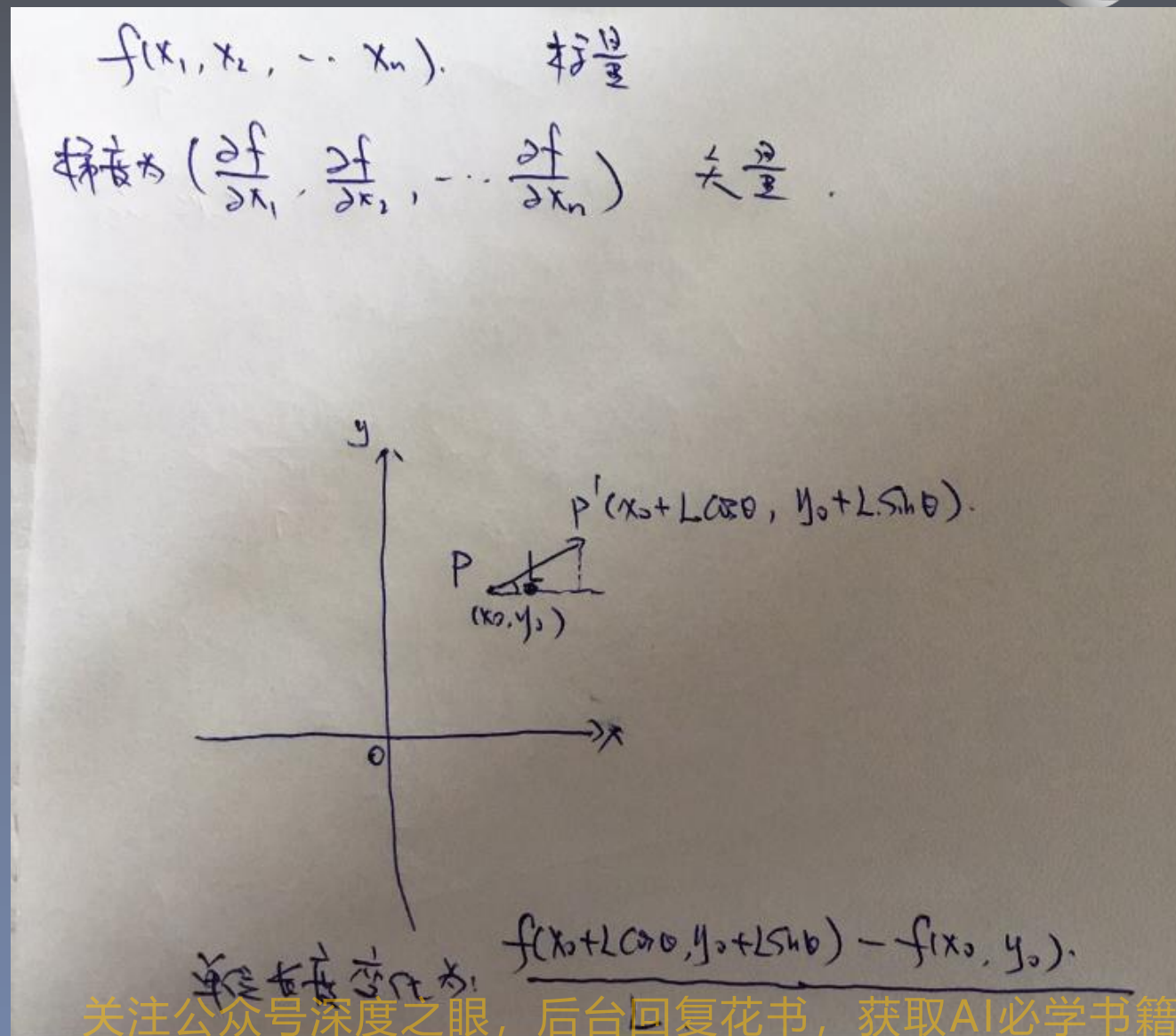
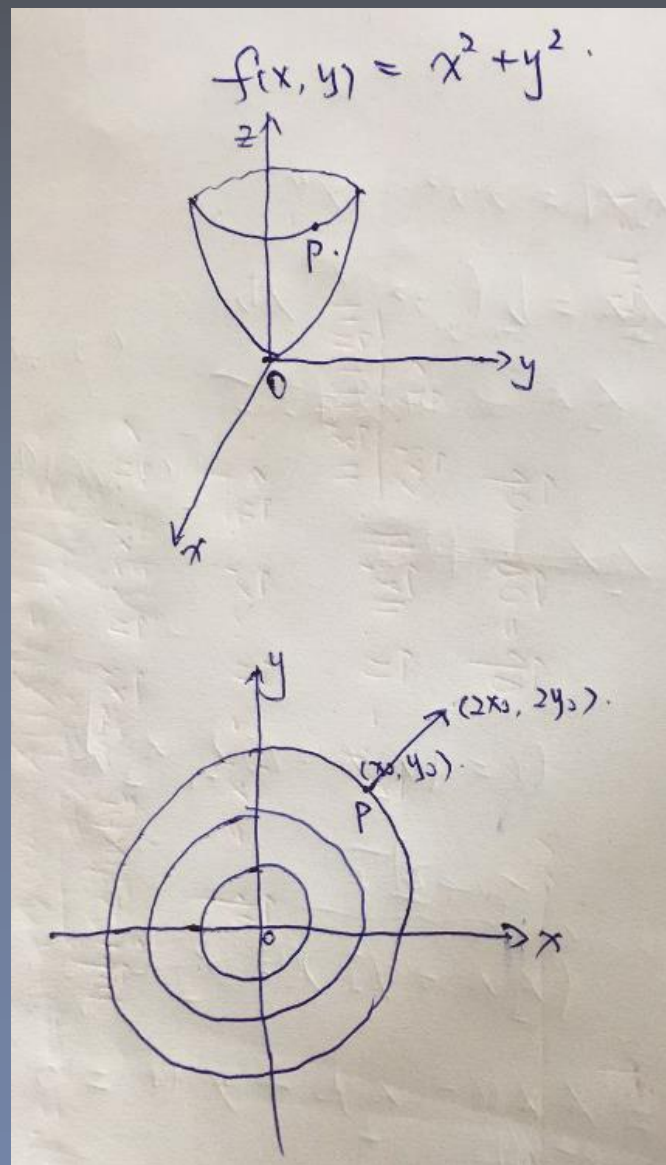
$$x^* = \min_x f(x), x \in R^n$$

梯度下降法



deepshare.net

深度之眼



关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料

单位长度变化为: $\frac{f(x_0 + L\cos\theta, y_0 + L\sin\theta) - f(x_0, y_0)}{L}$

$$= \frac{f(x_0 + L\cos\theta, y_0 + L\sin\theta) - f(x_0 + L\cos\theta, y_0)}{L} + \frac{f(x_0 + L\cos\theta, y_0) - f(x_0, y_0)}{L}$$

$$= \sin\theta \cdot \frac{f(x_0 + L\cos\theta, y_0 + L\sin\theta) - f(x_0 + L\cos\theta, y_0)}{L \cdot \sin\theta} + \cos\theta \cdot \frac{f(x_0 + L\cos\theta, y_0) - f(x_0, y_0)}{L \cos\theta}$$

$$L \rightarrow 0: \sin\theta \cdot f_x(x_0, y_0) + \cos\theta \cdot f_y(x_0, y_0)$$

$$(\sin\theta \cdot f_x + \cos\theta \cdot f_y)^2 \leq (\sin^2\theta + \cos^2\theta)(f_x^2 + f_y^2)$$

$$\langle x, y \rangle = \|x\| \cdot \|y\| \cos\theta$$

$$\langle x, y \rangle^2 = \|x\|^2 \|y\|^2 \cos^2\theta$$

$$\leq \|x\|^2 \|y\|^2$$

Cauchy 不等式

$$\hat{x} = (\sin\theta, \cos\theta)$$

$$y = (f_x, f_y)$$

$$\text{"=" 号} \Rightarrow \frac{\sin\theta}{\cos\theta} = \frac{f_x}{f_y}$$

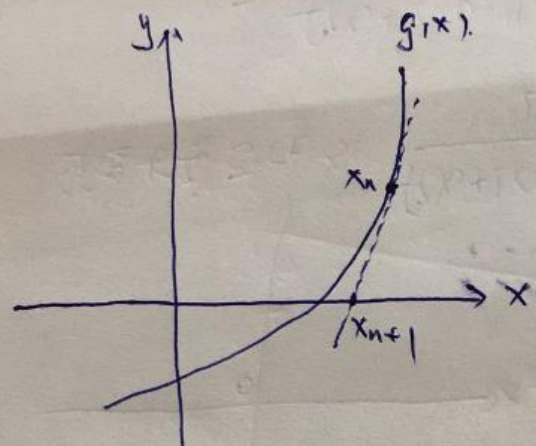


deepshare.net

深度之眼

牛顿法 (两种解释)

$$\min f(x) \\ f'(x) = 0.$$

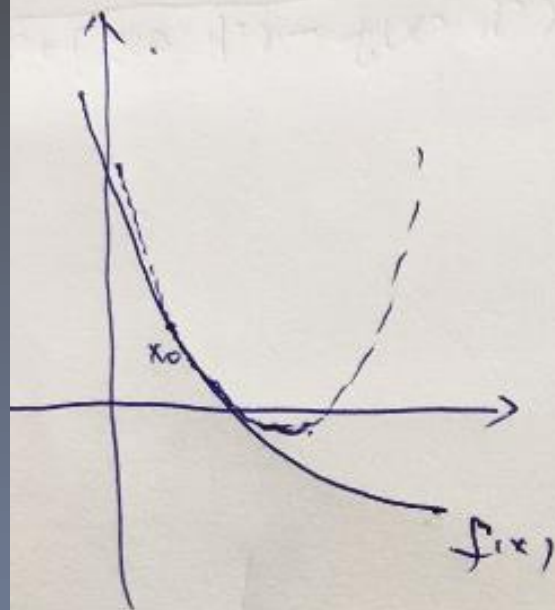


$$y - g(x_n) = g'(x_n)(x - x_n).$$

$$\text{令 } y = 0 \Rightarrow x = x_n - \frac{g(x_n)}{g'(x_n)}.$$

用 $f'(x)$ 替换 $g(x)$.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$



$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \dots$$

$$= \frac{f''(x_0)}{2}x^2 + (f'(x_0) - x_0 f''(x_0))x + \dots$$

$$-\frac{b}{2a} = -\frac{f'(x_0) - x_0 f''(x_0)}{f''(x_0)} = x_0 - \frac{f(x_0)}{f'(x_0)}$$



收敛速度比较，梯度下降是一次收敛，牛顿法是二次收敛（速度快，但也有缺陷，要在比较接近最优点的时候才能收敛，否则可能发散）

$$\|x_{n+1} - x^*\| \leq K \|x_n - x^*\|$$

$$\|x_{n+1} - x^*\| \leq K \|x_n - x^*\|^2$$

有约束，拉格朗日乘子的意义，KKT条件



deepshare.net

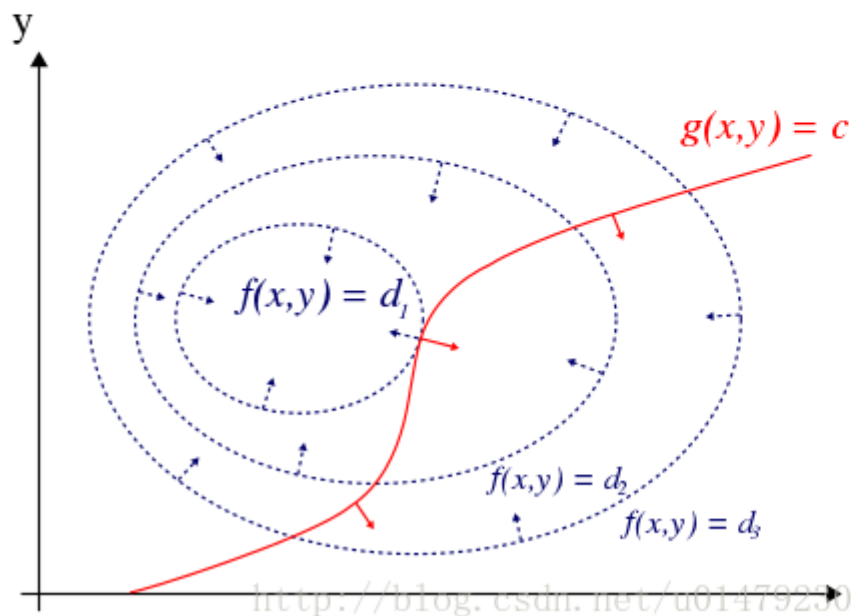
深度之眼

经典拉格朗日乘子法是下面的优化问题（注： \mathbf{x} 是一个向量）：

$$\min_{\mathbf{x}} f(\mathbf{x})$$

$$s.t. g(\mathbf{x}) = 0$$

直观上理解，最优解 $\mathbf{x}_{optimal}$ 一定有这样的性质，以 \mathbf{x} 是二维变量为例：（网上下的图。为了符合行文风格，这里的 $g(x, y) = c$ 应为 $g(x, y) = 0$ ）



$$\begin{cases} \nabla f(\mathbf{x}) = \lambda \nabla g(\mathbf{x}) \\ g(\mathbf{x}) = 0 \end{cases}$$

这时引入拉格朗日函数：

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x})$$

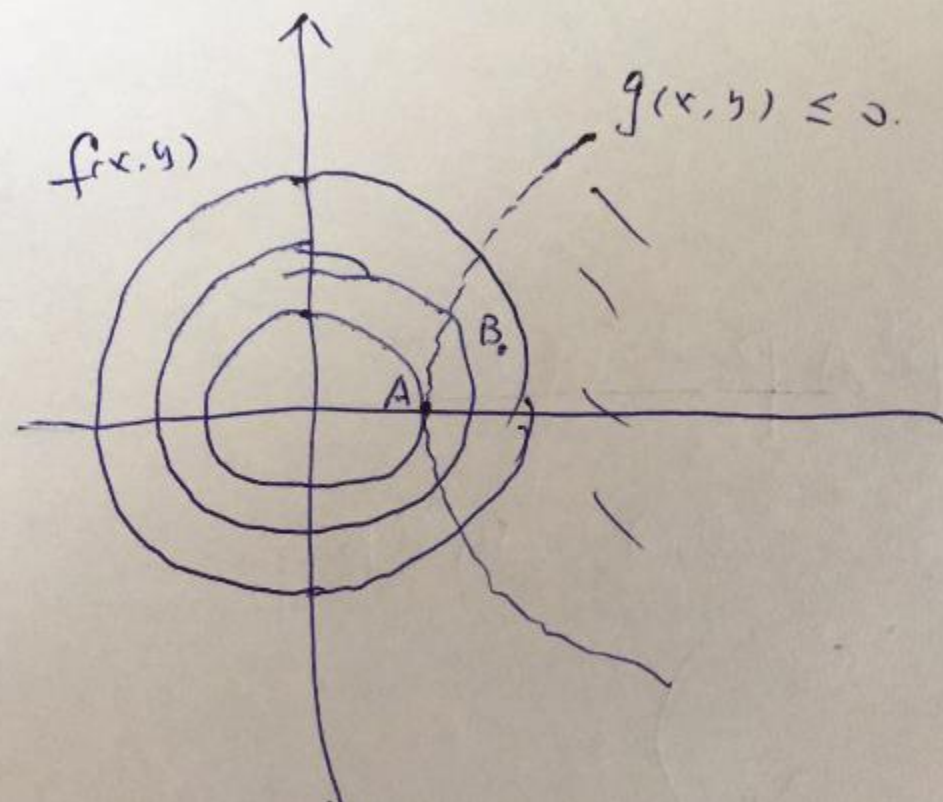
<http://blog.csdn.net/u014792304>

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料



deepshare.net

深度之眼



$$\nabla f(x^*, y^*) = \lambda \nabla g(x^*, y^*)$$

$$\lambda g(x^*, y^*) = 0$$

↓
KKT 条件

$$L = f(x) - \lambda g(x)$$

$$\begin{cases} \frac{\partial L}{\partial x} = 0 \\ \lambda g(x) = 0 \end{cases}$$

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料



deepshare.net

深度之眼

$$\begin{cases} \min f(x), \\ g_i(x) \geq 0, \quad (i=1, 2, \dots, n) \end{cases}$$

$$L = f(x) - \sum_{i=1}^n \lambda_i g_i(x)$$

$$\begin{cases} \frac{\partial L}{\partial x} = 0 \Rightarrow \nabla f(x) = \sum_{i=1}^n \lambda_i \nabla g_i(x), \\ \lambda_i g_i(x) = 0, \quad (\text{KKT 条件}) \end{cases}$$

λ_i

≥ 0

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料

件都加工。

例 5.1.1 求下列非线性规划问题的 K-T 点：

$$\min f(\mathbf{x}) = 2x_1^2 + 2x_1x_2 + x_2^2 - 10x_1 - 10x_2;$$

$$\text{s. t } \begin{cases} x_1^2 + x_2^2 \leq 5, \\ 3x_1 + x_2 \leq 6. \end{cases}$$

解 将上述问题的约束条件改写为 $g_i(\mathbf{x}) \geq 0$ 的形式：

$$\text{s. t } \begin{cases} g_1(\mathbf{x}) = -x_1^2 - x_2^2 + 5 \geq 0, \\ g_2(\mathbf{x}) = -3x_1 - x_2 + 6 \geq 0. \end{cases}$$

设 K-T 点为 $\mathbf{x}^* = (x_1, x_2)^T$, 有

$$\nabla f(\mathbf{x}^*) = \begin{bmatrix} 4x_1 + 2x_2 - 10 \\ 2x_1 + 2x_2 - 10 \end{bmatrix},$$

$$\nabla g_1(\mathbf{x}^*) = \begin{bmatrix} -2x_1 \\ -2x_2 \end{bmatrix},$$

$$\nabla g_2(\mathbf{x}^*) = \begin{bmatrix} -3 \\ -1 \end{bmatrix}.$$

定理 5.1.2, 且将 (5.1.14) 式中第 1 个向量方程拆成分量形式, 有:

$$\begin{cases} 4x_1 + 2x_2 - 10 + 2\gamma_1 x_1 + 3\gamma_2 = 0, \\ 2x_1 + 2x_2 - 10 + 2\gamma_1 x_2 + \gamma_2 = 0, \\ \gamma_1(5 - x_1^2 - x_2^2) = 0, \\ \gamma_2(6 - 3x_1 - x_2) = 0, \\ \gamma_1 \geq 0, \\ \gamma_2 \geq 0. \end{cases} \quad (5.1.17)$$



deepshare.net

深度之眼

$$\begin{cases} x_1 = 1, \\ x_2 = 2, \\ \gamma_1 = 1, \\ \gamma_2 = 0. \end{cases}$$



deepshare.net

深度之眼

联系我们：

电话：18001992849

邮箱：service@deepshare.net

QQ：2677693114



公众号



客服微信

关注公众号深度之眼，后台回复花书，获取AI必学书籍及完整学习资料