deepshare.net
深度之眼

《Deep Learning》
——数学基础

导师：Johnson

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分解以及应用

deepshare.net
深度之眼

实用性质：

A(B+C)=AB+AC A(B+C)=AB+AC （分配率）

A(BC)=(AB)C A(BC)=(AB)C （结合律）

AB≠BA AB≠BA （一般不满足交换律）

(AB)T=BTAT (AB)T=BTAT (转置)

xTy=(xTy)T=yTx (转置)

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。

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

矩阵逆： 矩阵（方阵）的逆满足如下条件

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

deepshare.net
深度之眼

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λi都是正数)

曾经一道面试题矩阵的压缩表示

最小n+1

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

所以：$B = P^TAP$

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

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

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

$$= \underline{\lambda_1 u_1 {u_1}^T} + \lambda_2 u_2 {u_2}^T + \lambda_3 u_3 {u_3}^T + \cdots + \lambda_n u_n {u_n}^T$$

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

一般矩阵的svd分解：

A^TA为对称正定矩阵

$(A^TA)^T = A^T(A^T)^T = A^TA \Rightarrow$ 对称性

$x^TA^TAx = (x^TA^T)(Ax) = (Ax)^T(Ax) \geq 0 \Rightarrow$ 正定性

deepshare.net
深度之眼

$$A_{m\times n}$$

$$(A^TA)_{n\times n} = U^TDU$$

$$(AA^T)_{m\times m} = V^TDV$$

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

其中λ_i为D的对角元素

令${V^T}_{m\times m} = (v_1, v_2, \cdots, v_m)$

$${U^T}_{n\times n} = (u_1, u_2, \cdots, u_n)$$

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

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

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

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

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

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

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

$$= \begin{pmatrix} v_1 & v_2 & \cdots & v_{10} \end{pmatrix}_{200\times 10} \begin{pmatrix} \lambda_1^{\frac{1}{2}} & & \\ & \lambda_2^{\frac{1}{2}} & \\ & & \ddots \\ & & \end{pmatrix}_{10\times 10} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{10} \end{pmatrix}_{10\times 100}$$

$$\approx M_{200\times 10} N_{10\times 100}$$

$A_{200\times 10} X_{100\times 1}$算$200\times 100$次乘法

$M_{200\times 10} N_{10\times 100} X_{100\times 1}$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

2.逆矩阵，伪逆矩阵，最小二乘解，最小范数解

deepshare.net
深度之眼

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

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

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

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

$\vdots$

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

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

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

当$N = n$且$X_{N\times n}$可逆时：

$a = X^{-1}Y$s

一般情况：$N \neq \mathrm{n}$

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

$x^T xa = x^T Y \qquad x^T x$是否可逆？

1. $\mathrm{N} > \mathrm{n}$

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

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

2. $N < n$

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

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

故$x^T x$不可逆

此刻 $J=\|xa-Y\|^2+\lambda\|a\|^2$

$$\frac{\partial J}{\partial a}=x^Txa-x^TY+\lambda a=0$$

$$(x^Tx+\lambda I)a=x^TY$$

$$a=(x^Tx+\lambda I)^{-1}x^TY$$

x^Tx为对称矩阵可对角化

$$x^Tx=\mathrm{p}^{-1}\begin{pmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_n\end{pmatrix}\mathrm{p}$$

$$|x^Tx|=\lambda_1\lambda_2\cdots\lambda_\mathrm{n}$$

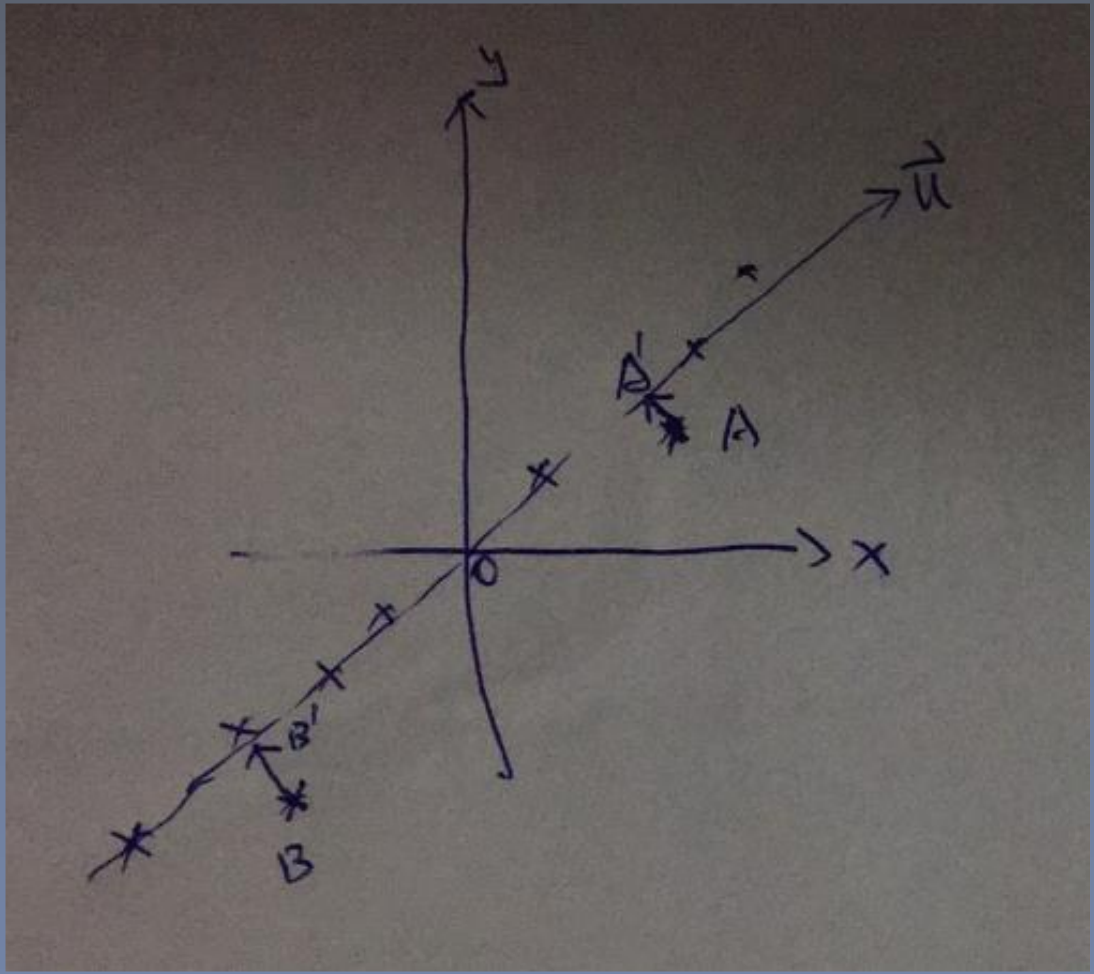
1. $a^T(x^Tx)a=(xa)^T(xa)\geq 0 \rightarrow \lambda_i\geq 0$

   $|x^Tx|$仍可能为0，不一定可逆

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

   $|x^Tx+\lambda I|>0$恒成立，一定可逆

3.PCA原理与推导



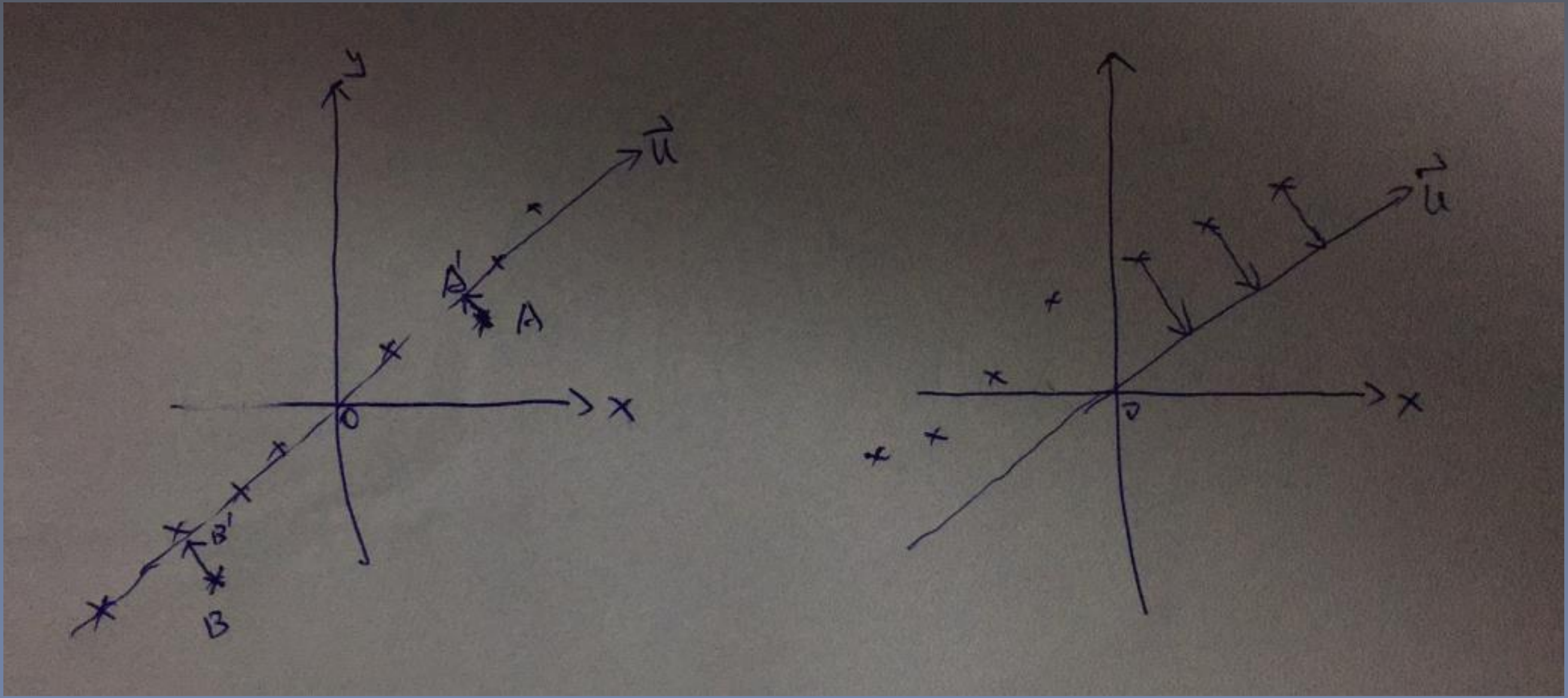
PCA仍然是一种数据压缩的算法

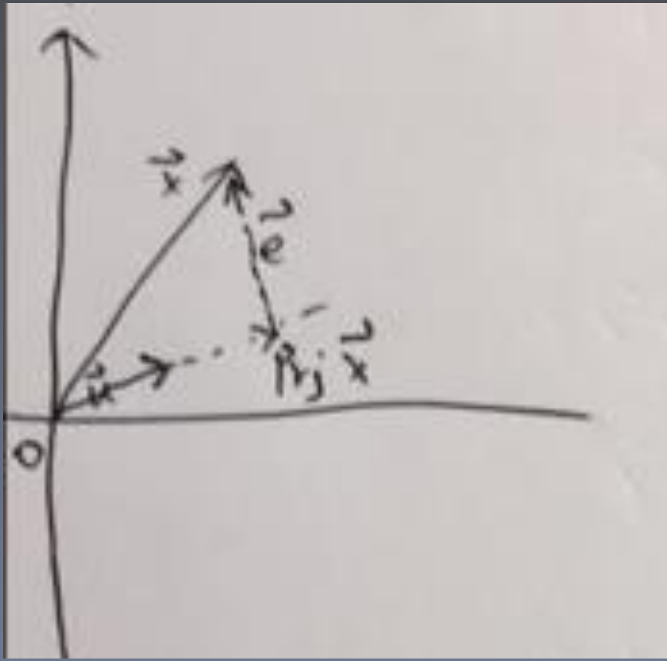
A点需要x,y两个坐标来表示，假设A在向量u上面的投影点为A′，则A′仅仅需要一个参数就能表示，就是OA′的长度(即A′在u上的坐标),我们就想着用A′来替换A，这样N个点(原来要2*N个参数),现在只需要(N+2)个参数(u也需要2个参数)

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





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

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

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

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

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

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

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

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

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

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

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

共有N个样本

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

$$x_i{}^T x_i = X$$

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

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

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

$$Xu = \lambda u$$

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

deepshare.net
深度之眼

联系我们:

电话：18001992849

邮箱：service@deepshare.net

Q Q： 2677693114

公众号

客服微信