

# 随机增量张量奇异值分解与人脸识别新算法

邱子衿 陈 潇 贾志刚

(江苏师范大学 数学与统计学院,江苏 徐州 221116)

**摘 要** 本文新提出随机增量张量奇异值分解方法,当数据逐步增加时,新方法能够在保持原数据的随机奇异值分解基础上,通过计算新增数据的奇异值分解得到更新后数据的张量奇异值分解.基于随机增量张量奇异值分解建立新的人脸识别模型.数值实验表明新模型与已有人脸识别模型相比具有较高的识别率.

**关键词** 随机张量;增量 T-SVD;人脸识别;张量脸

**中图分类号** O241.6

**文献标识码** A

## 0 绪论

生物特征识别技术<sup>[1]</sup>是一种以每个人自身的生理特征为标准,使用计算机、传感器等电子设备来自动进行身份鉴定的一种技术.基于人脸特征的身份识别技术正被越来越多的应用于公共安全、刑事侦查及商业管理等各行各业中.有各种子空间变换方法来进行人脸识别.Karl Pearson<sup>[2]</sup>在 1901 年提出的主成分分析(PCA)是广泛使用的一种线性子空间变换方法,在投影子空间内最大化特征的方差.线性判别分析(LDA)<sup>[3]</sup>则通过最大化类间协方差,同时最小化投影子空间内的类内协方差来进行信息判别.为了保持灰度图像的空间结构信息,Yang 等人<sup>[4]</sup>提出了一种称为二维主成分分析(2D-PCA)的人脸识别算法,其从表示图像的矩阵中直接计算图像协方差(散射)矩阵.Li 和 Yuan<sup>[5]</sup>使用判别信息提出了 2D-LDA,构建了图像的类间协方差矩阵和类内协方差矩阵.这些人脸识别算法都是将二维的人脸图像数据重排为一维向量,再利用特征提取对图像集进行截断.线性降维方法会破坏了人脸图像原有的几何结构,且在进行数据截断与特征提取时,原有信息可能会有所丢失,也可能导致维数灾难<sup>[6]</sup>.张量表示则可在一定程度上避免上述缺点.

张量是一个多维数组.高阶张量已被应用于化学计量学,心理测量学,图像和信号处理等应用领域.一幅灰度人脸图像可表示为二阶张量,灰度人脸图像集可表示为三阶张量.很多张量分解算法<sup>[7]</sup>被提出,如 CANDECOMP/PARAFAC(CP)分解<sup>[8]</sup>,TUCKER 分解<sup>[9]</sup>,高阶 SVD<sup>[10]</sup>和张量迭代<sup>[11]</sup>等.其中 CANDECOMP/PARAFAC(CP)分解和 TUCKER 分解是最广泛使用的两种张量分解方法.

张量脸<sup>[12]</sup>是第一个通过操作张量来处理人脸识别问题的算法.在这个算法中,不同因素的数据被表示为不同模式的张量.该算法与传统 PCA 一致的是所有图片都是向量化,因此多维性只有通过其他形式去表现数据的其它特性.张量的近似估计可由 Tucker 分解的代表:多维高阶 SVD(HOSVD)完成.但在本文中,我们基于[11]中提出的框架,以张量 SVD 分解为基础,结合随机张量的概念,提出了一种新的随机增量 SVD 方法.当增加训练集图片时,不需对更新后的训练集重新进行分解,只需在原有分解的基础上进行进一步计算.避免重复运算提高计算效率.

本文的其余部分组织如下:在第二部分中,我们给出了张量的相关概念和张量乘法的基本框架;在第三部分,我们将简要回顾张量 SVD 分解(T-SVD)及随机 SVD 分解算法并介绍张量的增量 SVD 和随机 SVD 分解算法;在第四节中引入人脸识别的新张量模型;在第五部分,我们在数据库 ORL 和 Extended Yale B 进行,同时将 T-SVD,随机 SVD,增量 SVD,随机增量 SVD 的结果进行比较,最后在第六节中得出结论,并指出了未来工作的几个问题.

收稿日期:2018-09-05

基金项目:国家自然科学基金项目(11771188)资助

通讯作者:贾志刚,男,汉族,博士,教授,研究方向:数值代数,E-mail:zhgjia@jsnu.edu.cn.

# 1 张量

本节我们介绍张量的基本概念,基本运算和张量乘法的基本框架.

张量(tensor):张量表示多维数组.一般的,一阶张量为向量,二阶张量为矩阵,三阶及以上张量则称为高阶张量.

张量的阶(modes):张成所属张量空间的向量空间的个数被称作张量的阶,可作为表示一个张量的量纲.一个  $p$  阶张量即是一个  $p$  维数组,写作  $\mathbf{A}=(a_{i_1 i_2 \dots i_p}) \in \mathbf{R}^{n_1 \times n_2 \times \dots \times n_p}$ . 张量的每一维度可称为模态. 纤维(fiber):表示张量的一维分量,即固定一个分量外的其余所有分量. 三阶张量有 mode-1(列), mode-2(行), mode-3(管). 切片(Slice):即固定两个分量外的其余分量. 有水平切片,侧面切片,正面切片. 用 MATLAB 符号可表示为  $\mathbf{A}(k, :, :)$ ,  $\mathbf{A}(:, k, :)$ ,  $\mathbf{A}(:, :, k)$ , 其中  $\mathbf{A}(:, :, k)$  可用  $\mathbf{A}^{(k)}$  表示,具体如图 1 所示.

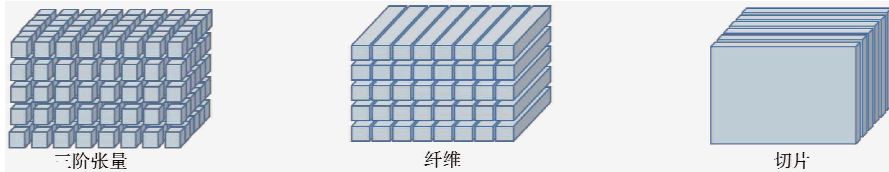


图 1 张量示意图

张量与矩阵相乘:张量  $\mathbf{X} \in \mathbf{R}^{I_1 \times I_2 \times \dots \times I_N}$  和一个矩阵  $\mathbf{U} \in \mathbf{R}^{J \times I_n}$  在第  $n$  个模态下的乘积记为  $(\mathbf{X} \times_n \mathbf{U}) \in \mathbf{R}^{I_1 \times \dots \times I_{n-1} \times J \times I_{n+1} \times \dots \times I_N}$ , 其元素定义为

$$(\mathbf{X} \times_n \mathbf{U})_{i_1 \dots i_{n-1} j_{n+1} \dots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \dots i_n} u_{j_n i_n}. \quad (1)$$

若张量  $\mathbf{A} \in \mathbf{R}^{l \times m \times n}$  则

$$\text{bcirc}(\mathbf{A}) = \begin{bmatrix} \mathbf{A}^{(1)} & \mathbf{A}^{(n)} & \mathbf{A}^{(n-1)} & \dots & \mathbf{A}^{(2)} \\ \mathbf{A}^{(2)} & \mathbf{A}^{(1)} & \mathbf{A}^{(n)} & \dots & \mathbf{A}^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{(n)} & \mathbf{A}^{(n-1)} & \mathbf{A}^{(n-1)} & \dots & \mathbf{A}^{(1)} \end{bmatrix}_{ln \times mn}$$

为  $ln \times mn$  的块循环矩阵. 定义命令  $\text{unfold}(\mathbf{A})$  将大小为  $l \times m \times n$  的三维张量作用为  $ln \times m$  的块矩阵

$$\text{unfold}(\mathbf{A}) = \begin{bmatrix} \mathbf{A}^{(1)} \\ \mathbf{A}^{(2)} \\ \vdots \\ \mathbf{A}^{(n)} \end{bmatrix}, \quad (2)$$

命令  $\text{fold}$  则将  $\text{unfold}(\mathbf{A})$  重新折叠为三阶张量

$$\text{fold}(\text{unfold}(\mathbf{A})) = \mathbf{A}. \quad (3)$$

张量  $\mathbf{A} \in \mathbf{R}^{l \times m \times n}$  的 Frobenius 范数为

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^l \sum_{j=1}^m \sum_{k=1}^n a_{ijk}^2}, \quad (4)$$

其中  $a_{ijk}$  是  $\mathbf{A}$  的第  $i, j, k$  个元素.

定义 1(张量乘法) 设张量  $\mathbf{A} \in \mathbf{R}^{l \times p \times n}$ ,  $\mathbf{B} \in \mathbf{R}^{p \times m \times n}$ , 则  $\mathbf{A} * \mathbf{B}$  [11] 得到维度为  $l \times m \times n$  的张量

$$\mathbf{A} * \mathbf{B} = \text{fold}(\text{bcirc}(\mathbf{A}) \cdot \text{unfold}(\mathbf{B})). \quad (5)$$

记矩阵  $\mathbf{A} \in \mathbf{R}^{I \times J}$ ,  $\mathbf{B} \in \mathbf{R}^{K \times L}$ , 则它们的 Kronecker 乘积为

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11} \mathbf{B} & a_{12} \mathbf{B} & \dots & a_{1J} \mathbf{B} \\ a_{21} \mathbf{B} & a_{22} \mathbf{B} & \dots & a_{2J} \mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1} \mathbf{B} & a_{I2} \mathbf{B} & \dots & a_{IJ} \mathbf{B} \end{bmatrix}.$$

设张量  $\mathbf{A} \in \mathbf{R}^{l \times m \times n}$ ,  $\mathbf{F}_n$  是大小为  $n \times n$  的离散傅里叶变换矩阵. 已知循环矩阵可由规范化的离散傅里叶变换矩阵对角化<sup>[13]</sup>, 即我们可以块对角化块循环矩阵. 现在  $\text{bcirc}(\mathbf{A})$  是大小为  $n \times n$  的块矩阵, 其中每块的大小为  $l \times m$ . 设  $\mathbf{P}_1, \mathbf{P}_2$  为块置换矩阵, 满足

$$\mathbf{P}_1 \cdot \text{bcirc}(\mathbf{A}) \cdot \mathbf{P}_2 = \begin{bmatrix} \mathbf{N}_{11} & \mathbf{N}_{12} & \cdots & \mathbf{N}_{1m} \\ \mathbf{N}_{21} & \mathbf{N}_{22} & \cdots & \mathbf{N}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{N}_{l1} & \mathbf{N}_{l2} & \cdots & \mathbf{N}_{lm} \end{bmatrix}, \quad (6)$$

为  $l \times m$  的块矩阵, 其中块  $\mathbf{N}_{ij}$  为  $n \times n$  的循环矩阵,  $\mathbf{N}_{ij}$  的第一列由  $\mathbf{A}^{(k)}$ ,  $k=1, 2, \dots, n$  的第  $i, j$  项组成, 且

$$\mathbf{P}_2^T \cdot (\mathbf{F}_n^* \otimes \mathbf{I}_m) \cdot \mathbf{P}_2 = \mathbf{I}_m \otimes \mathbf{F}_n^*. \quad (7)$$

显然

$$\mathbf{P}_1 \cdot (\mathbf{F}_n \otimes \mathbf{I}_l) \cdot \mathbf{P}_1^T \cdot \mathbf{P}_1 \cdot \text{bcirc}(\mathbf{A}) \cdot \mathbf{P}_2 \cdot \mathbf{P}_2^T \cdot (\mathbf{F}_n^* \otimes \mathbf{I}_m) \cdot \mathbf{P}_2 \quad (8)$$

等价于

$$(\mathbf{I}_l \otimes \mathbf{F}_n) \begin{bmatrix} \mathbf{N}_{11} & \mathbf{N}_{12} & \cdots & \mathbf{N}_{1m} \\ \mathbf{N}_{21} & \mathbf{N}_{22} & \cdots & \mathbf{N}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{N}_{l1} & \mathbf{N}_{l2} & \cdots & \mathbf{N}_{lm} \end{bmatrix} (\mathbf{I}_l \otimes \mathbf{F}_n^*) = \begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} & \cdots & \mathbf{D}_{1m} \\ \mathbf{D}_{21} & \mathbf{D}_{22} & \cdots & \mathbf{D}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{D}_{l1} & \mathbf{D}_{l2} & \cdots & \mathbf{D}_{lm} \end{bmatrix}, \quad (9)$$

其中  $\mathbf{D}_{ij}$  均是大小为  $n \times n$  的对角矩阵. 对上式左乘  $\mathbf{P}_1^T$ , 右乘  $\mathbf{P}_2^T$ , 有

$$(\mathbf{F}_n \otimes \mathbf{I}_l) \cdot \text{bcirc}(\mathbf{A}) \cdot (\mathbf{F}_n^* \otimes \mathbf{I}_m) = \mathbf{P}_1^T \begin{bmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} & \cdots & \mathbf{D}_{1m} \\ \mathbf{D}_{21} & \mathbf{D}_{22} & \cdots & \mathbf{D}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{D}_{l1} & \mathbf{D}_{l2} & \cdots & \mathbf{D}_{lm} \end{bmatrix} \mathbf{P}_2^T = \begin{bmatrix} \hat{\mathbf{A}}^{(1)} & & & \\ & \hat{\mathbf{A}}^{(2)} & & \\ & & \ddots & \\ & & & \hat{\mathbf{A}}^{(n)} \end{bmatrix}. \quad (10)$$

沿  $\mathbf{A}$  的第三维度(管)应用快速傅里叶变换给出了计算  $\hat{\mathbf{A}}^{(i)}$  的另一种方法. 令  $\hat{\mathbf{A}} = \text{fft}(\mathbf{A}, [], 3)$ , 则对  $\forall i$ , 有  $\hat{\mathbf{A}}^{(i)} = \hat{\mathbf{A}}(:, :, i)$ .

当张量  $\mathbf{A} \in \mathbf{R}^{l \times p \times n}$ ,  $\mathbf{A} \in \mathbf{R}^{p \times m \times n}$  是稀疏的, 我们直接使用定义 1 计算  $\mathbf{C} = \mathbf{A} * \mathbf{B}$ . 但当  $\mathbf{A}, \mathbf{B}$  为稠密张量时, 如果直接使用公式(5)的计算量约为  $O(lpn^2m)$ , 考虑到  $\text{bcirc}(\mathbf{A})$  可以被块对角化, 我们选择对(5)做变形<sup>[11]</sup>

$$(\mathbf{F}_n^* \otimes \mathbf{I}_l) (\mathbf{F}_n \otimes \mathbf{I}_l) \cdot \text{bcirc}(\mathbf{A}) \cdot (\mathbf{F}_n^* \otimes \mathbf{I}_m) (\mathbf{F}_n \otimes \mathbf{I}_m) \cdot \text{unfold}(\mathbf{B}). \quad (11)$$

注意到其中  $(\mathbf{F}_n \otimes \mathbf{I}_m) \text{unfold}(\mathbf{B})$  可以沿  $\mathbf{B}$  的第三维度(管)进行快速傅里叶变换得到, 我们记其结果为  $\hat{\mathbf{B}}$ . 由(10)同样沿  $\mathbf{A}$  的第三维度(管)进行快速傅里叶变换计算得  $\hat{\mathbf{A}}$ . 因此, 将  $\hat{\mathbf{A}}, \hat{\mathbf{B}}$  的正面切片进行相乘再对所得结果沿其第三维度(管)进行反向快速傅里叶变换即可得到所求结果.

**引理 1**<sup>[11]</sup>  $\mathbf{A} * (\mathbf{B} * \mathbf{C}) = (\mathbf{A} * \mathbf{B}) * \mathbf{C}$ .

**定义 2**<sup>[11]</sup> 如果张量  $\mathbf{A} \in \mathbf{R}^{l \times m \times n}$ , 则  $\mathbf{A}^T$  是大小为  $m \times l \times n$  的, 转置  $\mathbf{A}$  的每个正面切片再反转第 2 到  $n$  项的顺序后得到的张量.

**定义 3**<sup>[11]</sup> 一个张量是  $f$ -对角的, 如果它的每个正面切片是对角阵.

类似的, 一个张量是  $f$  上三角张量, 如果它的每个正面切片是上三角矩阵.

**定义 4**<sup>[11]</sup> 张量  $\mathbf{T}_{ln} \in \mathbf{R}^{l \times l \times n}$  是单位张量, 如果它的第一个正面切片是  $l \times l$  的单位矩阵, 其余正面切片均为零.

**定义 5**<sup>[11]</sup> 实张量  $\mathbf{Q} \in \mathbf{R}^{l \times l \times n}$  是正交的, 如果

$$\mathbf{Q}^T * \mathbf{Q} = \mathbf{Q} * \mathbf{Q}^T = \mathbf{I}, \quad (12)$$

这表示所有该张量的侧面切片都是正交的, 即

$$\mathbf{Q}(:, i, :)^T * \mathbf{Q}(:, j, :) = \begin{cases} \mathbf{e}_1, & i=j, \\ 0, & i \neq j, \end{cases} \quad (13)$$

这里  $\mathbf{e}_1$  为管张量(即大小是  $1 \times 1 \times n$ )且其第一个正面切片为 1, 其余正面切片为 0.

## 2 随机张量的增量奇异值分解

本节给出随机张量的增量奇异值分解.

### 2.1 T-SVD 分解

这里首先介绍以定义 1 张量乘法为基础,类似于矩阵奇异值分解的张量 SVD 分解,记为 T-SVD.

**定理 1**<sup>[11]</sup> 设实张量  $\mathbf{A} \in \mathbf{R}^{l \times m \times n}$ , 则  $\mathbf{A}$  可因式分解为

$$\mathbf{A} = \mathbf{U} * \mathbf{S} * \mathbf{V}^T, \quad (14)$$

其中  $\mathbf{U}$  是  $l \times l \times n$  的正交张量,  $\mathbf{V}$  是  $m \times m \times n$  的正交张量,  $\mathbf{S}$  是  $l \times m \times n$  的  $f$ -对角张量. T-SVD 分解示意图如图 2.

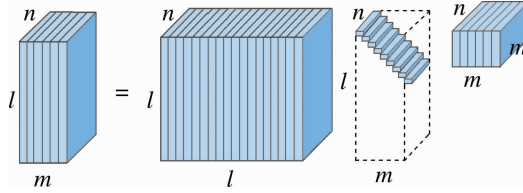


图 2 T-SVD 分解示意图

T-SVD 的特点是它提供了一种方法,类似矩阵 SVD 将矩阵的最佳逼近定义为向量外积的和, T-SVD 将  $k < \min(n_1, n_2)$  个矩阵外积的和作为张量的最佳近似.

**定理 2**<sup>[11]</sup> 张量  $\mathbf{A} \in \mathbf{R}^{l \times m \times n}$  的 T-SVD 分解由  $\mathbf{A} = \mathbf{U} * \mathbf{S} * \mathbf{V}^T$  给出, 且对于  $k < (l, m)$ , 定义

$$\mathbf{A}_k = \sum_{i=1}^k \mathbf{U}(:, i, :) * \mathbf{S}(i, i, :) * \mathbf{V}(:, i, :)^T, \quad (15)$$

则  $\mathbf{A}_k = \arg \min_{\mathbf{A} \in \mathbf{M}} \|\mathbf{A} - \tilde{\mathbf{A}}\|_F$ ,  $\mathbf{M} = \{\mathbf{C} = \mathbf{X} * \mathbf{Y} | \mathbf{X} \in \mathbf{R}^{l \times m \times n}, \mathbf{Y} \in \mathbf{R}^{k \times m \times n}\}$ .

### 2.2 张量的随机 SVD 分解

在本小节中,我们介绍张量的随机 SVD(RT-SVD)<sup>[17]</sup>方法.

**定义 6(高斯随机张量)** 张量  $\mathbf{W} \in \mathbf{R}^{n_1 \times n_2 \times n_3}$  是高斯随机张量,如果  $\mathbf{W}^{(1)}$  的元素服从标准正态分布,且其他正面切片均为零.

沿  $\mathbf{W}$  的第三模态使用快速傅里叶变换算法得到  $\hat{\mathbf{W}}$ , 可以发现  $\hat{\mathbf{W}}$  的每一个正面切片都等于  $\mathbf{W}^{(1)}$ .

**定义 7**<sup>[17]</sup> (T-QR 分解) 设实张量  $\mathbf{A} \in \mathbf{R}^{l \times m \times n}$ , 则  $\mathbf{A}$  可因式分解为

$$\mathbf{A} = \mathbf{Q} * \mathbf{R}, \quad (16)$$

其中  $\mathbf{Q}$  是  $l \times l \times n$  的正交张量,  $\mathbf{R}$  是  $l \times m \times n$  的  $f$ -上三角张量.

RT-SVD 的目的是在张量  $\mathbf{A} \in \mathbf{R}^{l \times m \times n}$  维数过大时找到一个很好的近似分解,使得  $\mathbf{A} \approx \mathbf{U}_k * \mathbf{S}_k * \mathbf{V}_k$ . 计算 RT-SVD 的第一步是生成高斯随机张量  $\mathbf{W} \in \mathbf{R}^{m \times (k+p) \times n}$ , 这里  $k$  为目标截断系数,  $p$  为过采样系数. 张量  $\mathbf{y} = \mathbf{A} * \mathbf{W} \in \mathbf{R}^{l \times (k+p) \times n}$ , 利用式(10)转换到傅里叶域中有  $\hat{\mathbf{y}}^{(i)} = \hat{\mathbf{A}}^{(i)} \mathbf{W}^{(1)}$ , 即对  $\forall i, \hat{\mathbf{y}}^{(i)}$  包含  $\hat{\mathbf{A}}^{(i)}$  列向量的随机线性组合. 计算  $\hat{\mathbf{y}}^{(i)}$  的 thin QR 分解  $\hat{\mathbf{y}}^{(i)} = \hat{\mathbf{Q}}^{(i)} \hat{\mathbf{R}}^{(i)}$ , 很显然  $\hat{\mathbf{y}}^{(i)}$  的列空间和  $\hat{\mathbf{Q}}^{(i)}$  的列空间相等, 所以有  $\hat{\mathbf{y}}^{(i)} \approx \hat{\mathbf{Q}}^{(i)} \hat{\mathbf{Q}}^{(i)H} \hat{\mathbf{A}}^{(i)}$ . 记  $\hat{\mathbf{Q}}, \hat{\mathbf{R}}$  是以  $\hat{\mathbf{Q}}^{(i)}, \hat{\mathbf{R}}^{(i)}$  为对角块的块对角矩阵, 选择合适的单位阵, 分别将  $\mathbf{F}_n^* \otimes \mathbf{I}, \mathbf{F}_n \otimes \mathbf{I}$  作用于两个块对角阵的两边, 张量化所得块循环矩阵的第一列即可得  $\mathbf{Q}, \mathbf{R}$ . 这样就得到低秩近似张量  $\mathbf{A} \approx \mathbf{Q} * \mathbf{Q}^T * \mathbf{A}$ . 接下来即可计算  $\mathbf{B} = \mathbf{Q}^T \mathbf{A}, \mathbf{B} \in \mathbf{R}^{(k+p) \times m \times n}$  的 T-SVD 分解, 取截断系数  $k$  进行截断得到  $\mathbf{B} = \mathbf{U} * \mathbf{S}_k * \mathbf{V}_k^T$ , 有

$$\mathbf{A} \approx \mathbf{Q} * \mathbf{Q}^T * \mathbf{A} = (\mathbf{Q} * \mathbf{U}) * \mathbf{S}_k * \mathbf{V}_k^T = \mathbf{U}_k * \mathbf{S}_k * \mathbf{V}_k^T.$$

因为  $\mathbf{Q}$  和  $\mathbf{U}$  均为正交张量, 所以  $\mathbf{U}_k = \mathbf{Q} * \mathbf{U}$  也为正交张量, 满足条件. 这样就完成  $\mathbf{A}$  的随机 SVD 分解.

### 2.3 张量的增量 SVD 分解

本节将矩阵的增量奇异值分解算法<sup>[18]</sup> 延展到三阶张量空间, 构造张量的增量 SVD 分解算法. 设原张量  $\mathbf{A} \in \mathbf{R}^{l \times m \times n}$ , 已有  $\mathbf{A} \approx \mathbf{U}_k * \mathbf{S}_k * \mathbf{V}_k^T$ , 这里  $\mathbf{U}_k \in \mathbf{R}^{l \times k \times n}, \mathbf{S}_k \in \mathbf{R}^{k \times k \times n}, \mathbf{V}_k \in \mathbf{R}^{m \times k \times n}$ . 现有张量  $\mathbf{Z} \in \mathbf{R}^{l \times t \times n}$  可作为侧面切片添加到  $\mathbf{A}$ , 我们用  $[\mathbf{A} \ \mathbf{Z}]$  表示更新后的张量, 即  $[\mathbf{A} \ \mathbf{Z}](:, 1:m, :) = \mathbf{A}, [\mathbf{A} \ \mathbf{Z}](:, m+1:m+t, :) = \mathbf{Z}$ , 如图 3 所示.

先将增量部分投影至  $U_k$  的侧面切片所张成的空间中有  $P := U_k \tau \omega$ , 其中  $w = U_k^T Z$ . 令  $r = Z - P$ , 显然  $r$  的侧面切片正交于  $U_k$  的侧面切片空间. 这样新的训练集可以表示为

$$[A \ Z] = [U_k S_k V_k \ P + r] = [U_k S_k V_k \ U_k \tau \omega + r]. \quad (17)$$

计算  $r$  的 T-QR 分解  $r = q \bar{r}$ , 这里  $q$  的侧面切片与  $r$  的侧面切片具有相同的空间, 同时  $q$  是正交张量. 式(17)则可以分解为

$$[U_k S_k V_k^T \ U_k w + r] = [U_k \ q] \begin{bmatrix} S_k & w \\ q_r & T \end{bmatrix} \begin{bmatrix} V_k^T \\ I \end{bmatrix}, \quad (18)$$

其中  $q \in \mathbf{R}^{l \times l \times n}$ ,  $w \in \mathbf{R}^{k \times t \times n}$ ,  $q^T r$  是由  $r$  的 T-QR 分解得到的  $f$  上三角张量  $\bar{r}$ , 维度为  $l \times n$ . 事实上将式(18)中的各部分作为一个整体, 仍可理解该式为“分块矩阵”, 只不过这里的每一分块均为张量, 运算时要按定义 1 的张量乘法进行. 记  $Y = \begin{bmatrix} S_k & w \\ q_r & T \end{bmatrix}$ , 对  $Y$  作 T-SVD 计算有  $Y = \tilde{U} \tilde{S} \tilde{V}^T$ , 其中  $\tilde{U}, \tilde{V}$  为正交张量. 因为  $Y$  是上三角张量所以对  $Y$  的 T-SVD 计算时间会小很多. 根据式(18)可得

$$\hat{U} = [U_k \ q] \hat{U}, \hat{S} = \tilde{S}, \hat{V} = \begin{bmatrix} V_k^T \\ I \end{bmatrix} \tilde{V}, \quad (19)$$

且  $\hat{U}^T * \hat{U} = \hat{U} * \hat{U}^T = I, \hat{V} * \hat{V}^T = \hat{V}^T * \hat{V} = I$ , 即正交张量, 满足条件. 这样我们就完成了对更新张量的 SVD 分解  $[A \ Z] = \hat{U} \hat{S} \hat{V}^T$ .

**算法 1** 张量的增量 SVD 算法.

**输入** 张量  $A \in \mathbf{R}^{l \times m \times n}$ , 已有  $A \approx U_k * S_k * V_k$ , 这里  $U_k \in \mathbf{R}^{l \times k \times n}, S_k \in \mathbf{R}^{k \times k \times n}, V_k \in \mathbf{R}^{m \times k \times n}$ . 增加张量  $Z \in \mathbf{R}^{l \times l \times n}$ .

**输出** 张量  $[A \ Z]$  的 SVD 分解  $\hat{U}_k * \hat{S}_k * \hat{V}_k$ ,

$w \leftarrow U^T Z, p \leftarrow U w, r \leftarrow Z - p, q \leftarrow \text{T-QR}(r)$ ,

$\hat{U} \hat{S} \hat{V}^T$  T-SVD  $\left[ \begin{bmatrix} S_k & w \\ q_r & T \end{bmatrix} \right]$ ,

$\hat{U}_k \leftarrow [U_k \ q] \hat{U}, \hat{S}_k \leftarrow \hat{S}, \hat{V}_k \leftarrow \begin{bmatrix} V_k^T \\ I \end{bmatrix} \hat{V}$ .

### 2.4 张量的随机增量 SVD

本节结合 2.2 与 2.3 节提出一种新的张量奇异值分解的计算方法, 即随机增量 SVD 算法. 同样记增加部分的张量为  $Z \in \mathbf{R}^{l \times l \times n}$ , 更新后的数据集记为  $[A \ Z]$ . 随机 SVD 的第一步是找到一个张量  $\tilde{Q}$ , 对  $[A \ Z]$  进行低秩近似表示

$$[A \ Z] \approx \tilde{Q} * \hat{Q}^T * [A \ Z]. \quad (20)$$

首先需要生成随机张量  $W$  作用目标张量. 为了保证增量 SVD 算法能很好的与随机 SVD 算法相结合, 这里要使原张量与增量部分的随机过程相互独立且原张量的随机 SVD 结果能为我们所用. 所以我们构造随机张量  $W = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix}$ , 这里  $W_1 \in \mathbf{R}^{m \times (k+p)n}, W_2 \in \mathbf{R}^{l \cdot (k+p)n}, W \in \mathbf{R}^{(m+l) \cdot 2(k+p)n}$  则

$$[A \ Z] \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} = [A W_1 \ Z W_2]. \quad (21)$$

对于原张量  $A \in \mathbf{R}^{l \times m \times n}$ , 假设我们已经对其进行随机 SVD, 有  $A W_1 = QR, B = Q^T A = U_k S_k V_k^T$ , 那么由式(21)有

$$[A \ Z] \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} = [A W_1 \ Z W_2] = [QR \ Z W_2]. \quad (22)$$

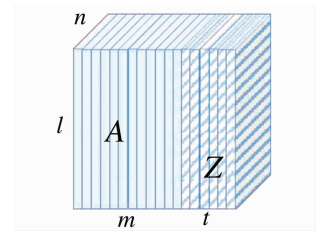


图 3 更新后的张量  $[A \ Z]$

接下来即计算  $[QR \quad ZW_2]$  的 T-QR 分解. 事实上,  $Q$  是正交张量, 将  $ZW$  投影到空间中有投影  $QQ^T$  ( $ZW$ ), 张量  $Q^T(ZQ)$  就成为  $R$  的新的侧面切片空间, 我们就只需计算  $[R \quad Q^T ZW_2]$  的 T-QR 分解. 于是, 有

$$[A \quad Z] \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} = [AW_1 \quad ZW_2] = [QR \quad ZW_2] = Q[R \quad Q^T ZW_2] = QQ\tilde{R} = \hat{Q}\hat{R}, \quad (23)$$

这里  $\hat{Q} = QQ\tilde{Q}$ ,  $\hat{R} = \tilde{R}$ . 这样即求得低秩近似表示  $[A \quad Z] \approx \hat{Q} * \hat{Q}^T * [A \quad Z]$ . 与 2.2 节中步骤相同, 接下来需要计算  $\hat{B} = \hat{Q}^T [A \quad Z]$  的 T-SVD 分解. 首先由  $B = Q^T A = U_k S_k V_k^T$ ,  $\hat{Q} = QQ\tilde{Q}$ , 我们有

$$\hat{B} = \hat{Q}^T [A \quad Z] = \tilde{Q}^T Q^T [A \quad Z] = \tilde{Q}^T [Q^T A \quad Q^T Z] = \tilde{Q}^T [B \quad Q^T Z] = \tilde{Q}^T [U_k S_k V_k^T \quad Q^T Z] \quad (24)$$

对  $[U_k S_k V_k^T \quad Q^T Z]$  则可运用 2.3 节中张量的增量 SVD 算法, 有

$$\hat{B} = \tilde{Q}^T \tilde{U}_k \tilde{S}_k \tilde{V}_k^T = \tilde{U} \tilde{S}_k \tilde{V}_k^T, \quad (25)$$

这里  $\tilde{U} = \tilde{Q}^T \tilde{U}_k$ . 综上, 即有

$$[A \quad Z] \approx \hat{Q} * \hat{Q}^T * [A \quad Z] = \hat{Q} * \hat{B} = (\hat{Q} * \tilde{U}) * \tilde{S}_k * \tilde{V}_k^T = (Q * \tilde{Q} * \tilde{Q}^T * \tilde{U}_k) * \tilde{S}_k * \tilde{V}_k^T, \quad (26)$$

即最终求得的奇异值分解结果为  $U = Q * \tilde{Q} * \tilde{Q}^T * \tilde{U}_k$ ,  $S = \tilde{S}_k$ ,  $V = \tilde{V}_k^T$ .

**算法 2** 张量的随机增量 SVD 算法.

**输入** 张量  $A \in \mathbf{R}^{l \times m \times n}$ , 增加张量  $z \in \mathbf{R}^{l \times l \times n}$ , 高斯随机张量:  $W_1 \in \mathbf{R}^{m \times (k+p) \times n}$ ,  $W_2 \in \mathbf{R}^{l \times (k+p) \times n}$ .

**输出** 张量  $[A \quad Z]$  的 SVD 分解  $U * S * V$ .

$A$  的随机 SVD 分解:  $[Q, R] \leftarrow$  T-QR ( $AW_1$ ),  $B = U_k S_k V_k^T \leftarrow$  T-SVD ( $Q^T A$ ),  $[\tilde{Q}, \tilde{R}] \leftarrow$  T-QR ( $[R \quad Q^T ZW_2]$ ),  $\hat{Q} \leftarrow QQ\tilde{Q}$ ,  $[\tilde{U}_k, \tilde{S}_k, \tilde{V}_k^T] \leftarrow$  增量 SVD ( $[U_k S_k V_k^T \quad Q^T Z]$ ),  $U \leftarrow \hat{Q} * \tilde{Q}^T * \tilde{U}_k$ ,  $S \leftarrow \tilde{S}_k$ ,  $V \leftarrow \tilde{V}_k^T$ .

### 3 人脸识别的张量模型

本节引入人脸识别的新张量模型.

#### 3.1 T-SVD 模型

与基于矩阵的 PCA 算法不同, T-SVD 将数据保存在三阶张量中, 使用基于张量乘法运算的张量分解法. 这种方法以刚刚讨论的 T-SVD 为基础, 基本算法由基于矩阵的 PCA 类比得到. 该算法中不需要矢量化样本和分解过程, 且所有投影均在张量空间完成.

因为将使用面向张量第三模态的矩阵, 首先定义 squeeze 表示将张量  $X \in \mathbf{R}^{l \times 1 \times n}$  压缩为一个矩阵, 即

$$X := \text{squeeze}(X) \Rightarrow X(i, j) := X(i, 1, j), \quad (27)$$

twist 表示对 squeeze 命令的反作用, 即

$$\text{twist}(\text{squeeze}(X)) = X. \quad (28)$$

设  $M$  表示平均脸,  $A(:, j, :) = \text{twist}(X_j - M)$ ,  $j = 1, \dots, m$ , 这里样本仍可理解为由“列向量”表示, 但这些列向量的长度均为  $l$ , 且其中分量为管张量.

由传统 PCA 的协方差矩阵类比得  $K := \frac{1}{m-1} A * A^T$  是对称半正定的<sup>[15]</sup>, 且正交可对角化<sup>[16]</sup>, 即  $K = U * D * U^T$ , 其中  $D$  是对角线为  $K$  的特征值组 (即管张量) 的对角张量,  $U$  的侧面切片  $U(:, j, :)$  可理解为  $K$  的特征矩阵. 显然每个切面  $K(:, :, j)$  表示所有图片第  $j$  列的协方差, 所以该协方差张量给出了侧面切片上的最大方差方向.

与矩阵 PCA 方法相似, 这里不需要计算  $K$  的特征分解, 而是使用张量的奇异值分解  $A = U * S * V^T$ , 即  $K = \frac{1}{m-1} A * A^T = \frac{1}{m-1} U * S * S^T * U^T$ , 左边的正交张量  $U$  即包含相对于  $K$  的主成分.

由于  $U$  是正交的,  $U(:, 1:k, :)$  是侧面切片  $U(:, j, :)$   $\in \mathbf{R}^{l \times 1 \times n}$  的正交集, 且  $U(:, 1:k, :) * U(:, 1:k, :)^T$  定义了一个正交投影<sup>[15]</sup> 可将图片投影至低维空间. 如果我们使用  $U$  左侧的前  $k$  个侧面切片作为新基, 则

$$A(:, j, :) \approx U(:, 1:k, :) * U(:, 1:k, :)^T * A(:, j, :) = \sum_{t=1}^k U(:, t, :) * C(t, j, :), \quad (29)$$

其中  $\mathbf{C}(:, j, :) = \mathbf{U}(:, 1:k, :)^T * \mathbf{A}(:, j, :)$ ,  $\mathbf{C}(t, j, :) = \mathbf{U}(:, t, :)^T * \mathbf{A}(:, j, :)$ ,  $t=1, 2, \dots, k$ . 所以每个图像  $\mathbf{A}(:, j, :)$  可被看作系数为  $\mathbf{C}(t, j, :)$  的  $\mathbf{U}(:, t, :)$ ,  $t=1, 2, \dots, k$  线性组合. 这样识别问题就简化为比较测试图集和训练图集的系数了. 注意到必须存储的元素为  $k, l \times n$  的基矩阵(用来投影图像)和训练集张量  $\mathbf{C} \in \mathbf{R}^{k \times (\text{No. training images}) \times n}$  投影至  $\mathbf{U}_k$  侧面切片空间的系数.

**算法 3** T-SVD 识别算法<sup>[14]</sup>.

**输入** 训练集  $\mathbf{X}_j, j=1, 2, \dots, n$ , 测试图片  $\mathbf{N}$ , 截断系数  $k$ .

**输出** 测试集与训练集在投影空间的系数比较.

for  $i=1$  to  $n$ ,

$$\mathbf{A}(:, i, :) = \mathbf{X}_j,$$

end.

$\mathbf{M} \leftarrow$  样本均值,

$\mathbf{A} \leftarrow \mathbf{A} - \mathbf{M}$ , 平均样本偏差,

$\mathbf{U} \leftarrow$  T-SVD( $\mathbf{A}$ ), 左奇异张量,

$\mathbf{A}_c \leftarrow \mathbf{U}_k^T \mathbf{A}$ , 训练集投影,

$\mathbf{T} \leftarrow \mathbf{N} - \mathbf{M}$ ,  $\mathbf{C} \leftarrow \mathbf{U}_k^T \mathbf{T}$ , 测试投影,

for  $j=1$  to  $n$ ,

$$\text{计算 Frobenious 范数 } \|\mathbf{C} - \mathbf{A}_c(:, j, :)\|_F,$$

end.

测试图片与训练集中系数之差最小的(即距离最短)样本匹配.

**引理 2**<sup>[14]</sup> 当图像数为一维大小(即  $n=1$ ), T-SVD 算法则退化为传统 PCA 算法.

在分析该算法时, 重写式(29)使其只有矩阵记号是很有用的. 为了方便起见, 令  $\mathbf{Y}_j := \text{squeeze}(\mathbf{A}(:, j, :))$ . 令  $\text{circ}(\mathbf{v})$  表示第一列为  $\mathbf{v}$  的循环矩阵, 式(29)可改写为

$$\mathbf{Y}_j \approx \sum_{t=1}^k \mathbf{W}_t \text{circ}(c_{t,j}), \quad (30)$$

这里  $\mathbf{W}_t := \text{squeeze}(\mathbf{U}(:, t, :))$  是大小为  $l \times n$  的矩阵,  $\mathbf{c}_{t,j}^T := \text{squeeze}(\mathbf{C}(t, j, :))^T$  是长度为  $n$  的行向量. 我们有以下结论.

**定理 3**<sup>[14]</sup> 令  $\mathbf{F}$  表示大小为  $n \times n$  的离散傅里叶变换矩阵, 记  $\hat{\mathbf{Y}}_j := \mathbf{Y}_j \mathbf{F}$ ,  $\hat{\mathbf{W}}_j := \mathbf{W}_j \mathbf{F}$ ,  $\hat{c}_{t,j}$  为长度为  $n$  的列向量  $c_{t,j}$  的傅里叶系数,

$$\mathbf{Y}_j \approx [\sum_{i=1}^k \hat{c}_{t,j}^{(1)} \hat{\mathbf{U}}^{(1)}(:, t), \sum_{i=1}^k \hat{c}_{t,j}^{(2)} \hat{\mathbf{U}}^{(2)}(:, t), \dots, \sum_{i=1}^k \hat{c}_{t,j}^{(n)} \hat{\mathbf{U}}^{(n)}(:, t)], \quad (31)$$

这里  $\hat{c}_{t,j}^{(i)}$  表示向量中的第  $i$  个分量,  $\hat{\mathbf{W}}_t(:, i) = \hat{\mathbf{U}}^{(i)}(:, t)$ , 其中  $\hat{\mathbf{U}}^{(i)}$  为酉矩阵.

这个定理的主要结果表明了进一步压缩数据库中信息的存储空间的可行性, 只要我们直接在傅里叶空间中进行操作. 同时我们可以证明在傅里叶空间中操作, 识别过程并没有损失(假设  $n$  为偶数).

测试图像  $J$  与训练集的管系数比较可直接在傅里叶域中进行, 因为使用张量乘法计算  $\mathbf{Z} = \mathbf{U}(:, 1:k, :)^T * \text{twist}(\mathbf{J} - \mathbf{M})$  相当于直接在傅里叶域中计算大小为  $k \times 1 \times n$  的张量  $\hat{\mathbf{Z}}$ .

由张量乘法后的讨论可知, 因为傅里叶域是共轭对称的, 所以只需执行一半的循环内乘法来计算  $\hat{\mathbf{Z}}$ , 即只需存储  $\mathbf{U}(:, 1:k, :)$  的前  $\frac{n}{2} + 1$  个正面切片. 如果我们想将  $\mathbf{Z}$  和  $\mathbf{C}$  的侧面切片进行比较, 我们可以直接

等价的比较  $\hat{\mathbf{Z}}(:, 1, 1:\frac{n}{2}+1)$  和  $\hat{\mathbf{C}}(:, j, 1:\frac{n}{2}+1)$ ,  $j=1, 2, \dots$ .

### 3.2 张量的随机 SVD 模型

当训练集样本过多时, 为提高计算效率可使用 2.2 节随机 SVD 算法替换上小节张量 SVD 模型中的 T-SVD 算法来得到训练集张量  $\mathbf{A} \in \mathbf{R}^{l \times m \times n}$  一个很好的近似分解  $\mathbf{A} \approx \mathbf{U}_k * \mathbf{S}_k * \mathbf{V}_k$ .

使用 RT-SVD 进行识别的想法同样是训练集  $\mathbf{A}$  的主成分可由左奇异张量  $\mathbf{U}$  的前  $k$  个侧面切片表示, 利

用  $U_k U_k^T$  所定义的正交投影将其投影至低维空间进行管系数的比较. 设  $\mathbf{M}$  表示平均脸,  $\mathbf{A}(\cdot, j, \cdot) = \text{twist}(\mathbf{X}_j - \mathbf{M})$ ,  $j=1, \dots, m$ . 由 3.2 节可知 RT-SVD 算法可分为两个主要步骤, 第一步是找到一个张量  $\mathbf{Q}$  得到训练集的低秩近似  $\mathbf{A} \approx \mathbf{Q} * \mathbf{Q}^T * \mathbf{A}$ , 第二步计算  $\mathbf{B} = \mathbf{Q}^T \mathbf{A}$ ,  $\mathbf{B} \in \mathbf{R}^{(k+p) \times m \times n}$  的 T-SVD 分解. 这样就完成训练集  $\mathbf{A}$  的随机 SVD 分解  $\mathbf{A} \approx \mathbf{Q} * \mathbf{Q}^T * \mathbf{A} = (\mathbf{Q} * \mathbf{U}) * \mathbf{S}_k * \mathbf{V}_k^T = \mathbf{U}_k * \mathbf{S}_k * \mathbf{V}_k^T$ , 得到左奇异张量  $\mathbf{U}_k$ , 将  $\mathbf{U}_k$  的侧面切片作为新基代入式(29), 之后即与上节相同完成后面的识别过程.

**算法 4** T-QR.

输入  $\mathbf{A} \in \mathbf{R}^{l \times p \times n}$ .

输出  $\mathbf{Q} \in \mathbf{R}^{l \times l \times n}$ ,  $\mathbf{R} \in \mathbf{R}^{l \times m \times n}$ ,  $\hat{\mathbf{A}} \leftarrow \text{fft}(\mathbf{A}, [], 3)$ .

for  $i=1$  to  $n$ ,

$[\mathbf{Q}, \mathbf{R}] = \text{qr}(\hat{\mathbf{A}}(\cdot, \cdot, i)), \hat{\mathbf{Q}}^{(i)} \leftarrow \mathbf{Q}, \hat{\mathbf{R}}^{(i)} \leftarrow \mathbf{R}$ .

end.

$\mathbf{Q} \leftarrow \text{ifft}(\hat{\mathbf{Q}}, [], 3), \mathbf{R} \leftarrow \text{ifft}(\hat{\mathbf{R}}, [], 3)$ .

**算法 5** 随机张量 SVD 识别算法.

输入 训练集  $\mathbf{X}_j, j=1, 2, \dots, n$ , 测试图片  $\mathbf{N}$ , 截断系数  $k$ , 高斯随机张量  $\mathbf{W} \in \mathbf{R}^{m \times (k+p) \times n}$ .

输出 测试集与训练集在投影空间的系数比较.

for  $i=1$  to  $n$ ,

$\mathbf{A}(\cdot, i, \cdot) = \mathbf{X}_j$ .

end.

$\mathbf{M} \leftarrow$  样本均值,

$\mathbf{A} \leftarrow \mathbf{A} - \mathbf{M}$ , 平均样本偏差,

$\mathbf{y} \leftarrow \mathbf{A} * \mathbf{W}, \mathbf{Q} \leftarrow \text{T-QR}(\mathbf{y})$ , 低秩近似分解空间,

$\mathbf{B} \leftarrow \mathbf{Q}^T \mathbf{A}, \mathbf{U}_k \leftarrow \text{T-SVD}(\mathbf{B}), \mathbf{U} \leftarrow (\mathbf{Q} * \mathbf{U}_k)$ , 投影空间,

$\mathbf{A}_c \leftarrow \mathbf{U}(\cdot, 1:k, \cdot)^T \mathbf{A}$ , 训练集投影,

$\mathbf{T} \leftarrow \mathbf{N} - \mathbf{M}, \mathbf{C} \leftarrow \mathbf{U}(\cdot, 1:k, \cdot)^T \mathbf{T}$ , 测试投影,

for  $j=1$  to  $n$ ,

计算 Frobenious 范数  $\|\mathbf{C} - \mathbf{A}_c(\cdot, j, \cdot)\|_F$ ,

end.

测试图片与训练集中系数之差最小的(即距离最短)样本匹配.

### 3.3 张量的随机增量 SVD 模型

本节提出两个新的人脸识别模型.

在实验中,有时需要对现有训练集进行更新,即向训练集添加新的图片.对于本文中利用张量存储的图片集而言,添加  $t$  张图片相当于在原张量训练集上增加了  $t$  个侧面切片,训练集大小由  $l \times m \times n$  变为  $l \times (m+t) \times n$ ,如图 4 所示.

由 2.3 节可知张量的增量 SVD 算法目的在于当训练集进行更新后,不需对更新后的训练集重新进行 T-SVD 运算,只需在原有的 T-SVD 结果上做进一步运算.使用 2.3 节增量 SVD 算法替换 3.1 节张量 SVD 模型中的 T-SVD 算法我们可以得到张量的增量 SVD 人脸识别模型.

**算法 6** 张量的增量 SVD 识别算法.

输入 训练集  $\mathbf{X}_j, j=1, 2, \dots, n$ , 测试图片  $\mathbf{N}$ , 截断系数  $k$ , 增量集  $\tilde{\mathbf{X}}_j, j=1, 2, \dots, t$ .

输出 测试集与训练集在投影空间的系数比较.

for  $i=1$  to  $n$ ,

$\mathbf{A}(\cdot, i, \cdot) = \mathbf{X}_j$ ,

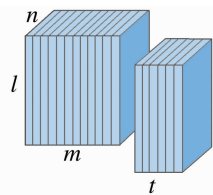


图 4 增加  $t$  个侧面张量

end.

$\mathbf{M} \leftarrow$  样本均值,

$\mathbf{A} \leftarrow \mathbf{A} - \mathbf{M}$ , 平均样本偏差,  $[\mathbf{U}_k, \mathbf{S}_k, \mathbf{V}_k] \leftarrow \text{T-SVD}(\mathbf{A})$ .

for  $i=1$  to  $t$ ,

$\mathbf{Z}(:, i, :) = \tilde{\mathbf{X}}_j$ .

end.

$\mathbf{A} \leftarrow [\mathbf{A} \ \mathbf{Z}]$ , 更新后训练集,

$\mathbf{U}_k \leftarrow$  算法 1 增量 SVD( $\mathbf{U}_k, \mathbf{S}_k, \mathbf{V}_k, \mathbf{Z}$ ), 左奇异张量,

$\mathbf{A}_c \leftarrow \mathbf{U}_k^T \mathbf{A}$ , 训练集投影,

$\mathbf{T} \leftarrow \mathbf{N} - \mathbf{M}$ ,

$\mathbf{C} \leftarrow \mathbf{U}_k^T \mathbf{T}$ , 测试投影.

for  $j=1$  to  $n+t$ ,

计算 Frobenious 范数  $\| \mathbf{C} - \mathbf{A}_c(:, j, :) \|_F$ .

end.

测试图片与训练集中系数之差最小的(即距离最短)样本匹配.

同样,将 2.4 节随机增量 SVD 算法替换 3.1 节张量 SVD 模型中的 T-SVD 算法得到张量的随机增量 SVD 人脸识别模型.

**算法 7** 张量的随机增量 SVD 识别算法.

**输入** 训练集  $\mathbf{X}_j, j=1, 2, \dots, n$ , 测试图片  $\mathbf{N}$ , 截断系数  $k$ , 增量集  $\tilde{\mathbf{X}}_j, j=1, 2, \dots, t$ .

**输出** 测试集与训练集在投影空间的系数比较.

for  $i=1$  to  $n$ ,

$\mathbf{A}(:, i, :) \mathbf{X}_j$ .

end.

$\mathbf{M} \leftarrow$  样本均值,

$\mathbf{A} \leftarrow \mathbf{A} - \mathbf{M}$ , 平均样本偏差.

for  $i=1$  to  $t$ ,

$\mathbf{Z}(:, i, :) = \tilde{\mathbf{X}}_j$ ,

end.

$\mathbf{U}_k \leftarrow$  算法 2 随机增量 SVD( $\mathbf{A}, \mathbf{Z}$ ), 左奇异张量,

$\mathbf{A} \leftarrow [\mathbf{A} \ \mathbf{Z}]$ , 更新后训练集,

$\mathbf{A}_c \leftarrow \mathbf{U}_k^T \mathbf{A}$ , 训练集投影.

$\mathbf{T} \leftarrow \mathbf{N} - \mathbf{M}, \mathbf{C} \leftarrow \mathbf{U}_k^T \mathbf{T}$ , 测试投影,

for  $j=1$  to  $n+t$ ,

计算 Frobenious 范数  $\| (\mathbf{C} - \mathbf{A}_c(:, j, :)) \|_F$ .

end.

测试图片与训练集中系数之差最小的(即距离最短)样本匹配.

## 4 数值实验

本节在数据库 ORL 和 Extended Yale B 进行数值实验分析, CroppedYale 人脸库包含 38 个人, 每个人 20 张照片. ORL 人脸库包含 40 个人, 每个人 10 张照片. 我们首先验证张量的增量 SVD 和张量的随机增量 SVD 算法的可行性, 然后就识别率和运行时间分别将 T-SVD 和增量 SVD, 随机 SVD 和随机增量 SVD 进行比较.

在所有实验中,定义识别率为  $\frac{\text{正确匹配图像的数量}}{\text{测试图像数量}}$ ,运行时间为求投影空间时间和对训练集测试集降

维的时间.

#### 4.1 增量 SVD 与 T-SVD 结果比较

本节在两个数据库下对增量 SVD 算法进行数值实验.图 5(a)是在 CroppedYale 人脸库下,原训练集每人 5 张,测试集每人 10 张照片. addnum 表示更新训练集时每人增加的照片数.图 5(b)是在 ORL 人脸库下,原训练集每人 3 张,测试集每人 3 张照片.两张图可以看出增量 SVD 算法是可行的,且增加照片的张数对识别率有一定影响.

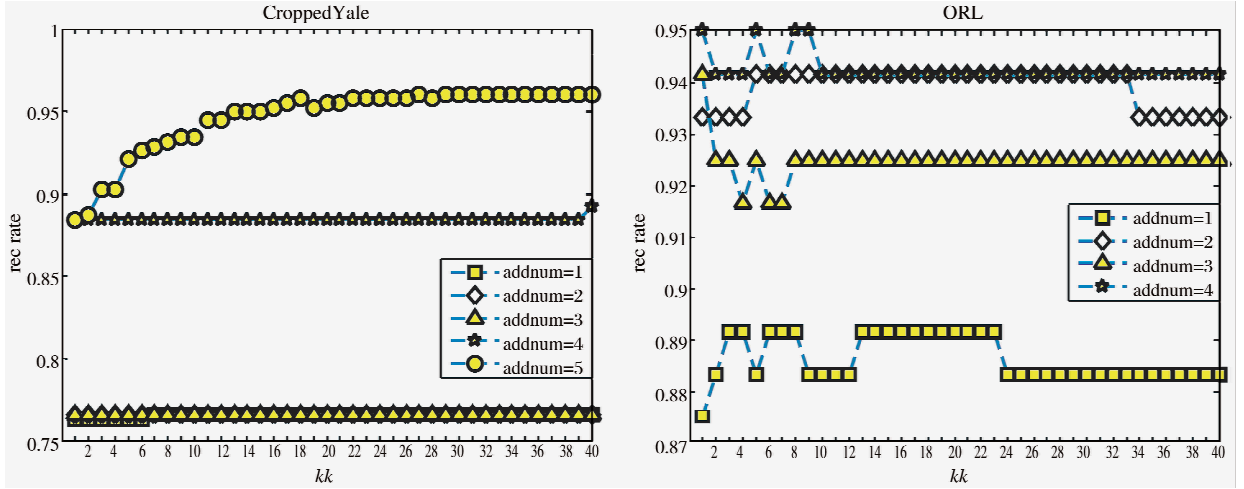


图 5 两个数据库下对增量 SVD 算法进行数值实验结果

图 6-7 是 T-SVD,增量 SVD 算法识别率与运行时间的比较结果,因为增量 SVD 的目的是当对原训练集添加照片时不需要重新进行整体的奇异值分解,所以对增量 SVD 与 T-SVD 运行时间的比较定义为使用 T-SVD 算法计算更新后训练集投影空间的时间和使用增量 SVD 算法针对增量部分计算投影空间的时间.

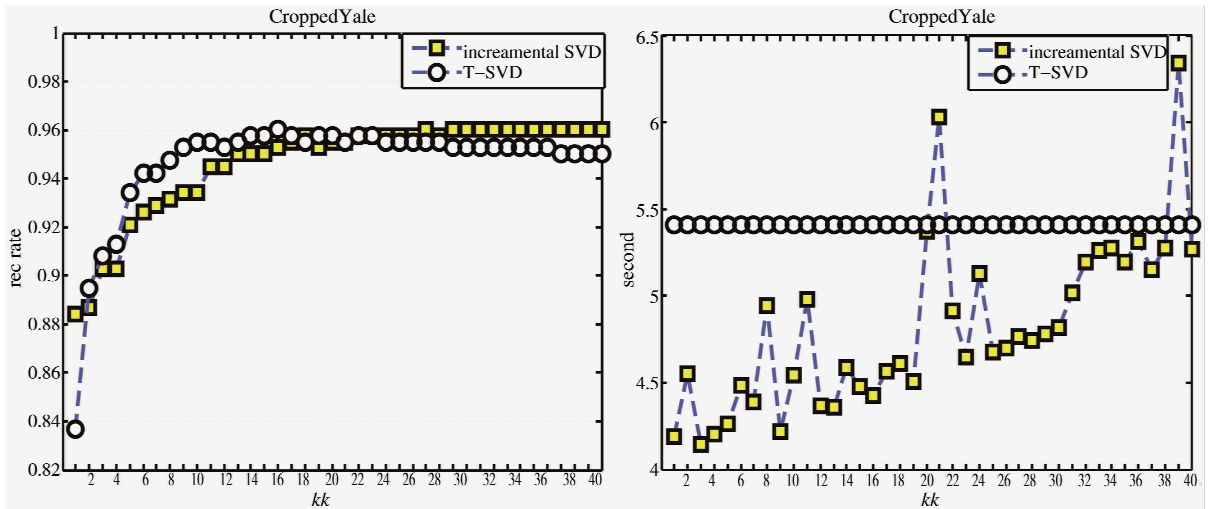


图 6 在 CroppedYale 人脸库下 T-SVD,增量 SVD 算法识别率与运行时间的比较结果

在 CroppedYale 人脸库下 T-SVD 训练集每人 10 张,测试集每人 10 张照片,增量 SVD 原训练集每人 5 张,测试集每人 10 张,增量部分每人增加 5 张照片.由图 6 可以看出截断  $k=16$  时 T-SVD 的识别率最高, $k=29$  时增量 SVD 的识别率最高后达到稳定.而在这两个截断点上增量 SVD 的运行时间都少于 T-SVD.

在 ORL 人脸库下 T-SVD 训练集每人 7 张,测试集每人 3 张照片,增量 SVD 原训练集每人 3 张,测试集每人 3 张,增量部分每人增加 4 张照片.图 7 可以看出增量 SVD 的运行时间在增加的照片数确定时随截断系数的增大呈上升趋势,但在识别率达到最优时的运行时间仍是少于 T-SVD 算法的.

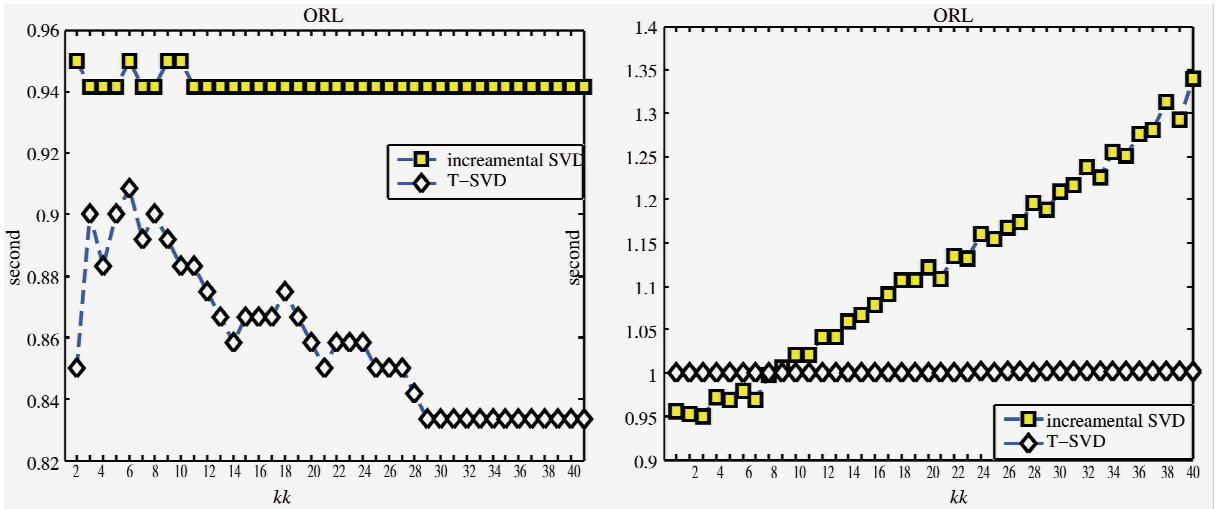


图 7 在 ORL 人脸库下 T-SVD,增量 SVD 算法识别率与运行时间的比较结果

### 4.2 随机 SVD 与随机增量 SVD 结果比较

本节在两个数据库下对随机增量 SVD 算法进行数值实验. 同上节相同,图 8 是在 CroppedYale 人脸库下,原训练集每人 6 张,测试集每人 5 张照片,addnum 表示更新训练集时每人增加的照片数.

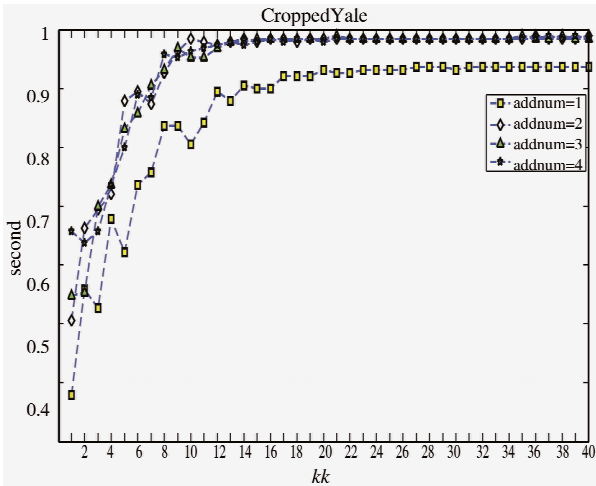


图 8 在 CroppedYale 人脸库下对随机增量 SVD 算法进行数值实验

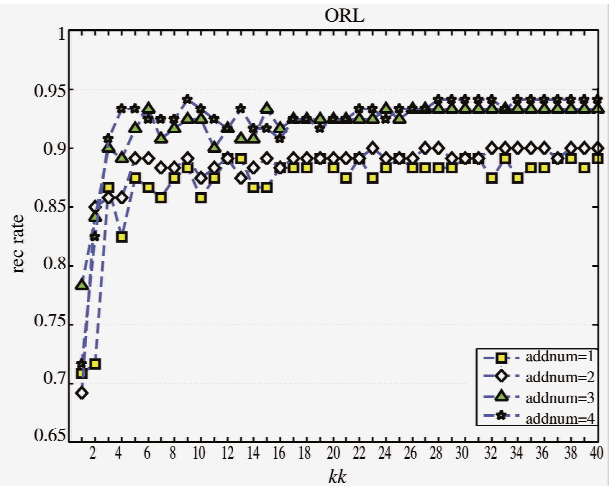


图 9 在 ORL 人脸库下对随机增量 SVD 算法进行数值实验

图 9 是在 ORL 人脸库下,原训练集每人 3 张,测试集每人 3 张照片. 图片证明了随机增量 SVD 算法的可行性且识别率同样随照片数的不断增加呈上升趋势.

图 10,图 11 是随机 SVD,随机增量 SVD 算法识别率与运行时间的比较结果. 与上节相同,运行时间定义为使用随机 SVD 算法计算更新后训练集投影空间的时间和使用随机增量 SVD 算法针对增量部分计算投影空间的时间.

在 CroppedYale 人脸库下随机 SVD 训练集每人 10 张,测试集每人 10 张照片. 随机增量 SVD 原训练集每人 5 张,测试集每人 10 张,增量部分每人增加 5 张照片. 由 3.3 可知随机 SVD 算法的目的在于当样本过多通过随机选取得到样本的近似分解,但随机的不确定性会影响识别结果. 图 10 可以看出在运算时间相当的情况下,随机增量 SVD 算法的识别率高于随机 SVD.

图 11 则是在 ORL 人脸库下随机 SVD 训练集每人 5 张,测试集每人 5 张照片. 增量 SVD 原训练集每人 3 张,测试集每人 5 张,增量部分每人增加 2 张照片. 因为这里增加的照片数少,识别率没有 CroppedYale 人脸库的差距明显,但还是高于随机 SVD 算法的.

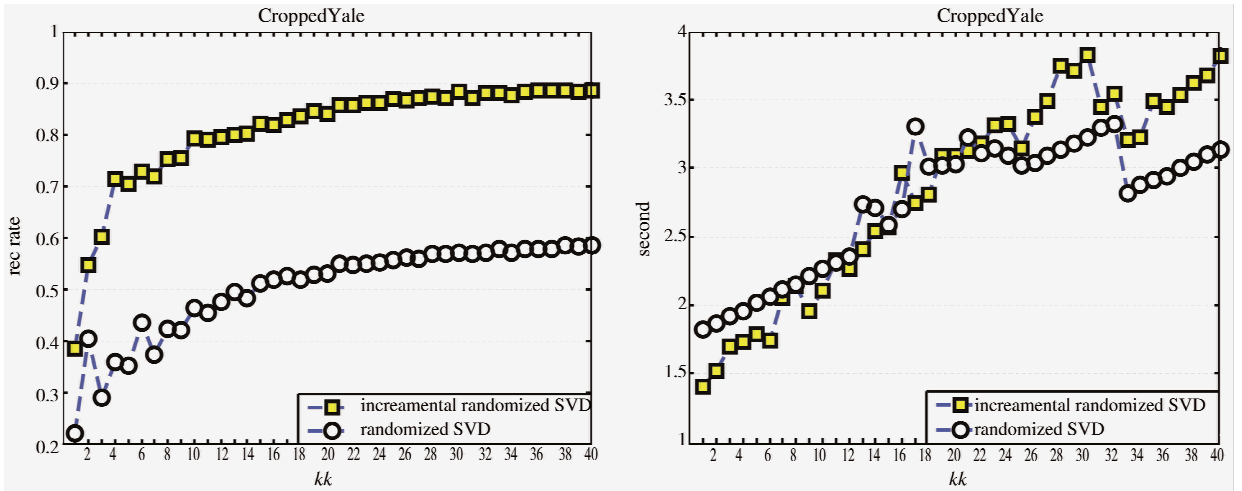


图 10 在 CroppedYale 人脸库下随机 SVD, 随机增量 SVD 算法识别率与运行时间的比较

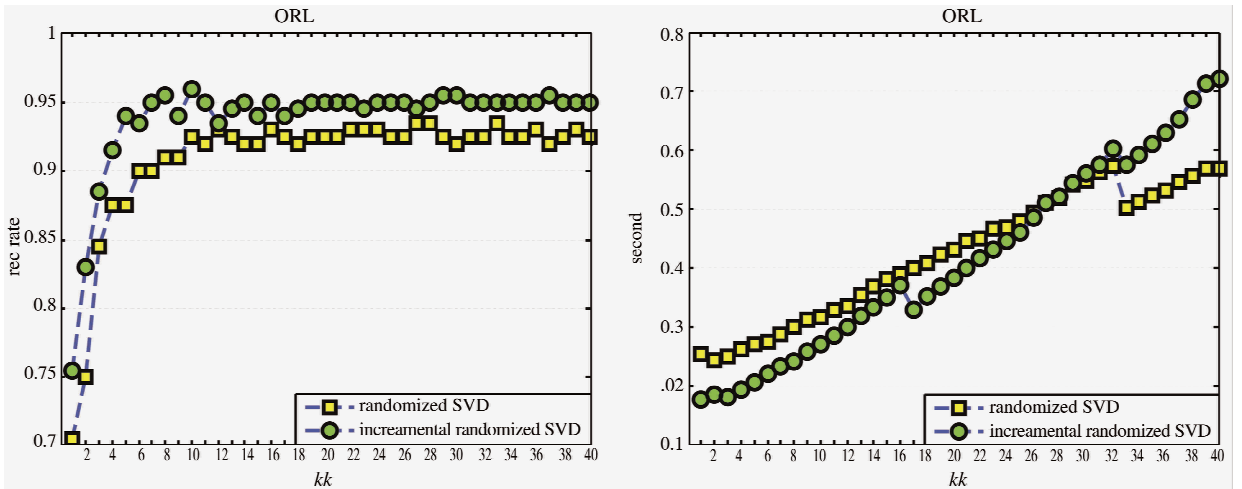


图 11 在 ORL 人脸库下随机 SVD, 随机增量 SVD 算法识别率与运行时间的比较

### 3 结论

利用张量进行图片存储保留更多有效信息在如今的图像处理中被频繁使用, 本文探讨了将矩阵增量奇异值分解算法延展到张量奇异值分解上的新模型, 以及结合了随机张量的相关算法. 实验结果表明增量 SVD 与随机增量 SVD 算法的结果与增加照片数和截断系数有关. 选取恰当的截断系数后张量的增量 SVD 算法可以在更短时间内得到与 T-SVD 算法相当甚至更高的识别率. 随机增量 SVD 算法则弥补了随机 SVD 近似分解时的不确定性对结果的影响, 较好的提高了识别效果.

### 参 考 文 献

- [1] 韩玉峰, 王小林, 张传文. 生物特征识别技术研究及应用[J]. 微计算机信息, 2012(3): 33-35.
- [2] Karl Pearson. On lines and planes of cloest fit to systems of points in space[J]. Philosophical Magazine, 1901(2): 559-572.
- [3] 马小龙. 数据降维方法综述[D]. 北京: 清华大学, 2005.
- [4] Yang J, Zhang D, Frangi A F, et al. Two-dimensional PCA: a new approach to appearance based face representation and recognition[J]. IEEE Trans on Patten Analysis and Machine Intelligence, 2004, 26(1): 131-137.
- [5] Li M, Yuan B Z. 2D-LDA: A statistical linear discriminant analysis for image matrix[J]. Pattern Recognition Letters, 2005, 26(5): 527-532.
- [6] Yan S C, Xu D, Yang Q, et al. Multilinear discriminant analysis for face recognition[J]. IEEE Trans on Image Processing, 2007, 16(1): 212-220.

- [7] Kolda T G, Bader B W. Tensor decompositions and applications[J]. *Siam Review*, 2009, 51 (3): 455-500.
- [8] Carroll J D, Chang J. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition[J]. *Psychometrika*, 1970, 35(3): 283-319.
- [9] Tucker L R. Some mathematical notes on three-mode factor analysis[J]. *Psychometrika*, 1966, 31(3): 279-311.
- [10] Lathauwer L D, Moor B D, Vandewalle J. A multilinear singular value decomposition[J]. *SIAM Journal of Matrix Analysis and Applications*, 2000, 21(4): 1253-1278.
- [11] Kilmer M E, Martin C D. Factorization strategies for third-order tensors[J]. *Linear Algebra and Its Applications*, 2011, 435 (3): 641-658.
- [12] Vasilescu M A O, Terzopoulos D. Multilinear analysis of image ensembles: tensorfaces[J]. *European Conference on Computer Vision*, 2002 (2350): 447-460.
- [13] Gene G, Charles V L. *Matrix Computations*[M]. 3rd edition. Baltimore: Johns Hopkins University Press, 1996.
- [14] Hao N, Kilmer M E, Braman K, et al. Facial recognition using tensor-tensor decompositions[J]. *Siam Journal on Imaging Sciences*, 2013, 6 (1): 437-463.
- [15] Kilmer M E, Braman K, Hao N, et al. Third order tensors as operators on matrices: a theoretical and computational framework with applications in imaging[J]. *Siam Journal on Matrix Analysis & Applications*, 2013, 34 (1): 148-172.
- [16] Braman K. Third-order tensors as linear operators on a space of matrices[J]. *Linear Algebra and Its Applications*, 2010, 433 (7): 1241-1253.
- [17] Zhang J, Saibaba A K, Kilmer M, et al. A randomized tensor singular value decomposition based on the t-product[J]. *Numerical Linear Algebra with Applications*, 2018, 25(5): e2179.
- [18] Bunch J R, Nielsen C P. Updating the singular value decomposition[J]. *Numerische Mathematik*, 1978, 31 (2): 111-129.

## A Randomized Tensor Singular Value Decomposition with Increment for Face Recognition

QIU Zi-jin   CHEN Xiao   JIA Zhi-gang

(School of Mathematics and Statistics, Jiangsu Normal University, Xuzhou 221116, China)

**Abstract** In this paper, the randomized tensor singular value decomposition with increment is proposed. In the process of updating the training datasets, the existing results of the original datasets can be maintained and utilized to obtain the SVD decomposition of the new datasets. Based on the randomized tensor singular value decomposition with increment, a new model of face recognition is established. Numerical experiments show that the new model has higher recognition rate than existing face recognition models.

**Key words** randomized tensor; incremental T-SVD; facial recognition; tensor face