

A Sparse Factorization Strategy for Third-Order Tensors and Its Application

Liang Wang

School of Applied Mathematics, Guangdong University of Technology, Guangzhou Guangdong
Email: 1065035870@qq.com

Received: Aug. 10th, 2018; accepted: Aug. 23rd, 2018; published: Aug. 30th, 2018

Abstract

This paper mainly studies the decomposition algorithm of tensor and its application. TST-SVD algorithm is proposed on the basis of T-SVD [1] algorithm of tensor, and it solves the case that the T-SVD algorithm may not be able to retain its main information when decomposing. However, our TST-SVD can keep its main information well through the soft threshold method, and achieve a good denoising effect for noise images. It also has a good performance in image recognition and reconstruction.

Keywords

T-SVD Algorithm, TST-SVD Algorithm, Soft Threshold, Image Recognition and Reconstruction

一个三阶张量的稀疏分解方法及其应用

汪亮

广东工业大学, 应用数学学院, 广东 广州
Email: 1065035870@qq.com

收稿日期: 2018年8月10日; 录用日期: 2018年8月23日; 发布日期: 2018年8月30日

摘要

本文主要研究张量的分解算法及其应用。在张量的T-SVD [1]算法的基础上提出TST-SVD算法, 并且解决了T-SVD算法在分解时可能无法保留其主要信息的情况。而我们的TST-SVD则通过软阈值的方法可以很好地保留其主要信息, 并且对于噪音图片也可以达到很好去噪效果。在图片的识别和重构方面也有着不错的表现。

关键词

T-SVD算法, TST-SVD算法, 软阈值, 图片的识别和重构

Copyright © 2018 by author and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

张量分解早在 19 世纪就被提出来了, 随着计算机的发展张量活跃在各个领域。例如在神经影像, 信号处理, 化学统计学, 计算机视觉, 心理测验学, 数据挖掘, 图像处理等领域有着极其重要的应用。由于数据的复杂性以及我们所需要的一些特征, 这就需要我们研究张量的内部结构特征。张量的经典分解有 CP 分解[2]和 Tucker 分解[2], CP 分解表示为由一系列秩一的向量外积之和, Tucker 表示一个核张量[2]与因子矩阵的 n 模积。

Tamara G. Kolda 和 Brett W. Bader 在 2007 发表了一篇关于张量的分解和应用的综述[2], 文章讲解了什么是张量, 如何做张量分解以及张量的一些应用。文章也提到了经典的 CP 分解和 Tucker 分解, 也列举了其他的一些分解方法, 并且用对比试验分析了它们的优缺点, 给予研究人员很大的启发。Genevera I. Allen [3]提出了一种稀疏的 CP 分解算法, 基于张量乘积的稀疏 CP 分解算法, 创新性的使用软阈值操作以达到分解成分稀疏的效果。Eric C. Chi 和 Tamara G. Kolda [4]提出了基于泊松分布的张量分解模型, 利用非线性 Gauss-Seidel 来构造优化函数加入一些约束条件得出 KKT 条件求得最优解。经过迭代到收敛得出新的稀疏非负的分解。Will Wei Sun [5]等人在 Genevera I. Allen 在稀疏 CP 分解的基础上提出了截断的张量 CP 分解, 也取得了不错的效果。Misha E. Kilmer [1]等人提出了对张量做矩阵操作的张量 T-SVD 分解方法。这些算法在不同情况下已经被证实可以取得不错的效果。

这篇文章中我们提出了一个新的张量分解 TST-SVD, 它是在 Misha E. Kilmer 等人的基础上对其算法缺点做了修改并且实验结果也达到了预期的效果。针对 T-SVD 算法中直接对重构张量进行压缩处理而导致某些重要信息缺失的情况, 我们对分解出的成分做软阈值操作以达到稀疏效果和保留重要信息的作用。

本文的组织结构如下: 第 2 章介绍张量; 第 3 章介绍张量的 T-SVD 并且详细讨论 T-SVD 算法及其总流程; 第 4 章详细讲解我们新的算法(TST-SVD); 第 5 章为实验设置和结果分析; 第 6 章总结全文。

2. 张量

张量是一个多维数组[2], 一个零阶张量对应一个标量, 一个一阶对应一个向量, 一个二阶对应一个矩阵。一个三阶对应一个立方体 $A \in R^{I \times J \times K}$ 如图 1, 一个 n 阶张量表示为 $X \in R^{I_1 \times I_2 \times \dots \times I_N}$ 。

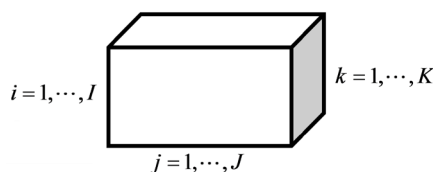


Figure 1. Third order tensor $A \in R^{I \times J \times K}$

图 1. 3 阶张量 $A \in R^{I \times J \times K}$

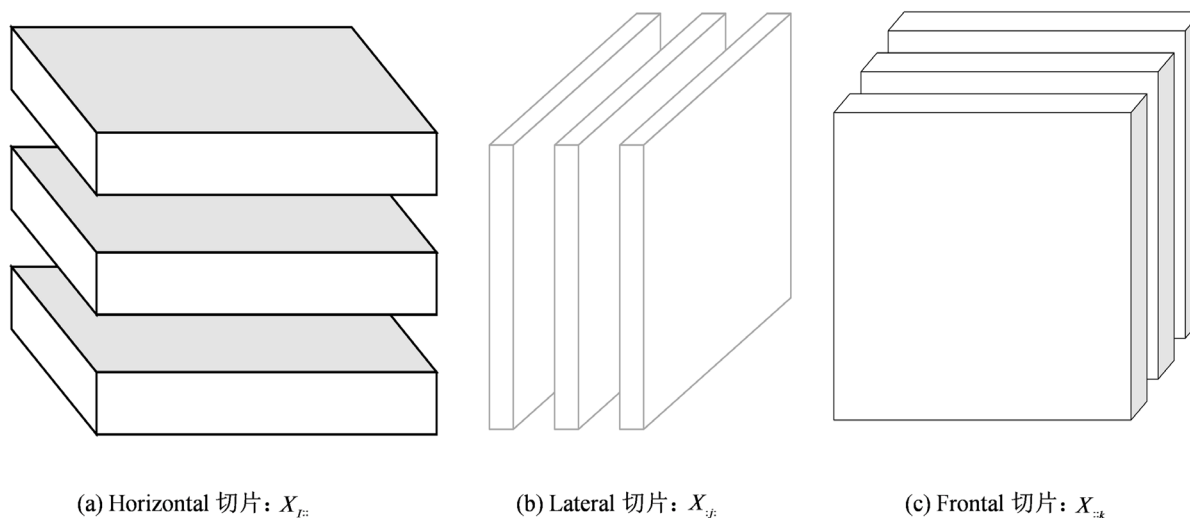


Figure 2. Slices of a 3rd-order tensor
图 2. 3 阶张量的切片

向量的第 i 个元素表示为 a_i ，矩阵的第 (i, j) 个元素表示为 a_{ij} ，同样三阶张量的第 (i, j, k) 个元素可以表示为 a_{ijk} ，高阶张量以此类推。

张量的切片如图 2 所示。

3. 张量的 T-SVD 分解

它是在傅里叶域中一种基于矩阵的新的分解形式，类似与矩阵的奇异值分解他们有一些相似的性质。这对于张量是一个很实用的方法。

张量的 T-SVD 分解， A 是一个 $n_1 \times n_2 \times n_3$ 的实数张量，则 A 可以表示为：

$$A = U * S * V^T \quad (1)$$

3.1. 定义 1

由张量 frontal 切片组成的块循环矩阵：

$$\text{circ}(A) = \begin{bmatrix} A^{(1)} & A^{(n_3)} & A^{(n_3-1)} & \dots & A^{(2)} \\ A^{(2)} & A^{(1)} & A^{(n_3)} & \dots & A^{(3)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A^{(n_3)} & A^{(n_3-1)} & \dots & A^{(2)} & A^{(1)} \end{bmatrix} \quad (2)$$

其中 $A \in R^{n_1 \times n_2 \times n_3}$ 的张量， $A^{(i)}$ 是张量 A 的 $n_1 \times n_2$ 的 frontal 切片， $\text{circ}(A)$ 是一个 $n_1 n_3 \times n_2 n_3$ 的块循环矩阵。

我们定义 MatVec 操作， $\text{MatVec}(A)$ 的作用是把 $n_1 \times n_2 \times n_3$ 的张量变成了一个 $n_1 n_3 \times n_2 n_3$ 的块矩阵，fold 操作是对 MatVec 的一个还原。两个操作如下：

$$\text{MatVec}(A) = \begin{bmatrix} A^{(1)} \\ A^{(2)} \\ \vdots \\ A^{(n_3)} \end{bmatrix}, \quad \text{fold}(\text{MatVec}(A)) = A \quad (3)$$

3.2. 定义 2

我们来定义一个张量的 t-product:

$$A * B = \text{fold}(\text{circ}(A) \cdot \text{MatVec}(B)) \quad (4)$$

其中 $A \in R^{n_1 \times n_2 \times n_3}$, $B \in R^{p \times n_2 \times n_3}$ 。

3.3. 傅里叶变换

前面我们对矩阵做了一些操作使其具备了某些性质: 块矩阵可以通过傅里叶变换变成块对角矩阵。这就意味着, 当 $A \in R^{n_1 \times n_2 \times n_3}$, F 是一个正则化的 $n_3 \times n_3$ 的 DFT (Discrete Fourier Transform) 矩阵, 则存在 n_3 个 $n_1 \times n_2$ 的矩阵 D_i 使得:

$$(F \otimes I) \text{circ}(A) (F^* \otimes I) = \text{blockdiag}(D_1, D_2, \dots, D_{n_3}) \quad (5)$$

3.4. 张量的 T-SVD

当张量是 f-diagonal (对角) 时, 则他的每个 frontal slice 都是对角的。同样地, 当张量是 f-upper triangular 或 f-lower triangular 时, 则张量的 frontal slice 是上三角或者下三角。

定理 1: A 是一个 $n_1 \times n_2 \times n_3$ 的实数张量, 则 A 的分解为:

$$A = U * S * V^T \quad (6)$$

其中 U, V 分别是 $n_1 \times n_2 \times n_3$ 和 $n_2 \times n_2 \times n_3$ 的正交张量, S 是一个 $n_1 \times n_2 \times n_3$ 的 f-diagonal 张量。这就是张量的 T-SVD 分解[6]。

证明: 首先我们把 A 变换成公式(2)的傅里叶域中, 接着我们计算每个 D_i 的 SVD, 则 $D_i = U_i \Sigma_i V_i^T$ ($i=1, 2, \dots, n_3$)

$$\begin{bmatrix} D_1 & & \\ & \ddots & \\ & & D_{n_3} \end{bmatrix} = \begin{bmatrix} U_1 & & \\ & \ddots & \\ & & U_{n_3} \end{bmatrix} \begin{bmatrix} \Sigma_1 & & \\ & \ddots & \\ & & \Sigma_{n_3} \end{bmatrix} \begin{bmatrix} V_1^T & & \\ & \ddots & \\ & & V_{n_3}^T \end{bmatrix} \quad (7)$$

由公式(2)可得(6)式。

3.5. 张量的 T-SVD 的算法

算法 1:

- 1) 输入 $n_1 \times n_2 \times n_3$ 的张量 A 。
- 2) 把 A 映射到傅里叶域中, $D = \text{fft}(A, [1, 3])$ 。
- 3) 对张量 D 的每一个 frontal slice 做矩阵的奇异值分解, 得到的每个 u, s, v 都分别存储在 $U(:, :, i), S(:, :, i), V(:, :, i)$ 中。
- 4) 在分别对 U, S, V 做一个傅里叶逆变换。就可以得到张量 A 的奇异值分解。

3.6. 张量的 T-SVD 的截断定理

一个 $n_1 \times n_2 \times n_3$ 张量的 A , 它的 T-SVD 分解为 $A = U * S * V^T$ 。 $k < \min(n_1, n_2)$, 则有

$$A_k = \sum_{i=1}^k U(:, i, :) * S(i, i, :) * V(:, i, :)^T \quad (8)$$

3.7. 张量的 T-SVD 的截断算法

算法 2:

- 1) 输入 $n_1 \times n_2 \times n_3$ 的张量 A , 截断 k 。
- 2) 用算法 1 得出 $U(:,i,:), S(i,i,:), V(:,i,:)$ 。
- 3) 分别计算 $U(:,i,:)$ 的 k_1 个奇异值分解, $S(i,i,*) * V(:,i,:)$ 的 k_2 个奇异值分解。
- 4) 计算 $\sigma_i \mu^{(j)} \lambda^{(l)} \left\| \text{circ}(q^{(j)}) t^{(l)} \right\|_{\infty} > \text{tol}$ [6] 当满足条件时, 可以得出张量 A 的截断 T-SVD。

3.8. 新视角下的张量

我们定义两个操作 squeeze 和 twist。

- 1) squeeze 可以把张量矩阵化, $X := \text{squeeze}(x) \Rightarrow X(i, j) := x(i, 1, j)$ 。
- 2) twist 把矩阵张量化, $\text{twist}(\text{squeeze}(x)) = x$ 。

其中 $x \in R^{n_1 \times n_2 \times n_3}$ 。我们可以用它们来构造图片的张量矩阵, 当 M 为图片的平均值矩阵, 则 $A(:, j, :) = \text{twist}(X_j - M), j = 1, \dots, m$ 。其中 $A(:, j, :)$ 是张量 A 的 lateral slice, X_j 是第 j 张图片。

4. 张量的 TST-SVD 算法

张量的 TST-SVD 是张量的 T-SVD 算法和张量的软阈值 CP 分解算法的基础上改进的。张量的 TST-SVD 在张量截断 T-SVD 算法的基础上合理的引入了软阈值操作。由于张量的 T-SVD 算法在去噪方面对于某些特征图片并没有很好地优化, 因此加入了软阈值操作可以很好地平滑去噪。TST-SVD 在人脸识别方面与传统的 PCA 和 T-SVD 相比有着较好的效果。

4.1. 软阈值操作

软阈值操作的公式为 $S(\bullet, \rho) = \text{sign}(\bullet)(|\bullet| - \rho) + \bullet$ [3],

$$U = \text{sign}(U)(|U| - \rho_U * I_U) + U$$

$$S = \text{sign}(S)(|S| - \rho_S * I_S) + S$$

$$V = \text{sign}(V)(|V| - \rho_V * I_V) + V$$

其中 ρ 是软阈值参数, 在这里通过多次试验当 $\rho_U = 0.001$ 时软阈值效果最好。 I 是元素全为 1 的张量并且与它们各自的规模相同。

4.2. 张量 TST-SVD 的算法步骤

算法 3:

- 1) 输入 $n_1 \times n_2 \times n_3$ 的张量 A , 截断 k 。
- 2) 用算法 1 得出 $U(:,i,:), S(i,i,:), V(:,i,:)$ 。
- 3) 分别计算 $U(:,i,:)$ 的 k_1 个奇异值分解, $S(i,i,*) * V(:,i,:)$ 的 k_2 个奇异值分解。
- 4) 计算 $\sigma_i \mu^{(j)} \lambda^{(l)} \left\| \text{circ}(q^{(j)}) t^{(l)} \right\|_{\infty} > \text{tol}$ [6] 当满足条件时, 得出 U, S, V 。
- 5) 分别对 U, S, V 做软阈值 $S(\bullet, \rho)$ 操作, 得出最终的张量 TST-SVD 分解。

4.3. 张量 TST-SVD 在人脸识别上的算法步骤

由于 U 是正交的, U_{st} 是 U 的正交集。并且 $U_{\text{st}} * U_{\text{st}}^T$ 是一个正交投影, 它把图片投影到更小的空间

维度中。我们对张量 A 的 lateral slice 进行投影:

$$A(:,j,:) \approx U_{st} * U_{st}^T * A(:,j,:) = \sum_{t=1}^k U_{st} * C(t,j,:) \quad (9)$$

其中 $C(:,j,:)=U_{st}^T * A(:,j,:)$, C 就是我们图片投影的系数。当做人脸识别时,我们只需比较训练图片的投影系数与测试图片的投影系数即可。

算法 4:

- 1) 输入训练集图片 $I_i, i=1,2,\dots,N$; 测试图片 J 。
- 2) 计算训练集图片的均值, 得出均差图片 ζ_i 。然后把每张图片以 $\zeta_i \rightarrow A(:,i,:)$ 的形式存储到张量 A 的 lateral slice 中。
- 3) 由算法 3 得出 U 的软阈值 U_{st} , $U_{st}^T * A$ 表示训练集的正交投影。
- 4) 对测试图片做均值处理, 再做一个 $\text{twist}(J-M) \rightarrow T(:,1,:)$ 的变换, 对 $T(:,1,:)$ 做正交投影 $B \leftarrow U_{st}^T * T$ 得出测试图片的投影系数 C 。
- 5) 对测试图片的系数与训练图片的系数做 2-范数, 直到收敛。
- 6) 找出 2-范数最小的那个就可以找出训练集图片的位置, 从而达到识别的效果。

5. 实验设置和结果分析

为了比较张量的 TST-SVD 算法的性能, 本文将 TST-SVD 算法, T-SVD 算法和经典的 PCA 算法用于 Yale faces [7] 的人脸识别实验, 并对实验结果进行结果对比分析。

5.1. 实验设置

我们选取 10 个不同的人, 每个人有 10 种不同的表情, 图片的大小是 100 px \times 100 px。第一组我们随机选取每个人的 3 个表情作为训练集的 30 张图片, 剩下的 70 张图片作为测试集。第二组我们随机选取每个人的 4 个表情作为训练集的 40 张图片, 剩下的 60 张图片作为测试集。第三组我们随机选取每个人的 5 个表情作为训练集的 50 张图片, 剩下的 50 张图片作为测试集。第四组我们随机选取每个人的 6 个表情作为训练集的 60 张图片, 剩下的 40 张图片作为测试集。第五组我们随机选取每个人的 7 个表情作为训练集的 70 张图片, 剩下的 30 张图片作为测试集。分别使用上述的三种算法, 针对每个组进行实验。并记录实验结果的人脸识别率和人脸重构的情况。实验平台为 MATLAB R2014a。

5.2. 实验结果分析

为了使实验更加有效每组做 20 次实验所得的人脸识别率取平均值。其中人脸识别率等于识别正确的个数与测试集总数之比。在多次实验中 TST-SVD 的软阈值操作 U 的阈值为 0.001 时实验效果最好, 在这里我们也取 $\rho_U = 0.001$ (表 1)。

我们再来比较 TST-SVD 与 PCA 的重构图片效果如何。重构图片的公式为: $J \approx U * U^T * J + M$ [8]。

由图 3 可以看出在相同训练集和测试集的情况下 TST-SVD 的识别率更高。图 4 可以看出在图片重构方面 TST-SVD 也有着不错的表现。

Table 1. The average face recognition probability of The TST-SVD, T-SVD and PCA

表 1. TST-SVD, T-SVD 和 PCA 的平均人脸识别率

算法	30	40	50	60	70
TST-SVD	0.7857	0.8333	0.7600	0.9500	0.7333
T-SVD	0.7100	0.8010	0.7000	0.9000	0.6900
PCA	0.6000	0.7500	0.6777	0.8500	0.6333

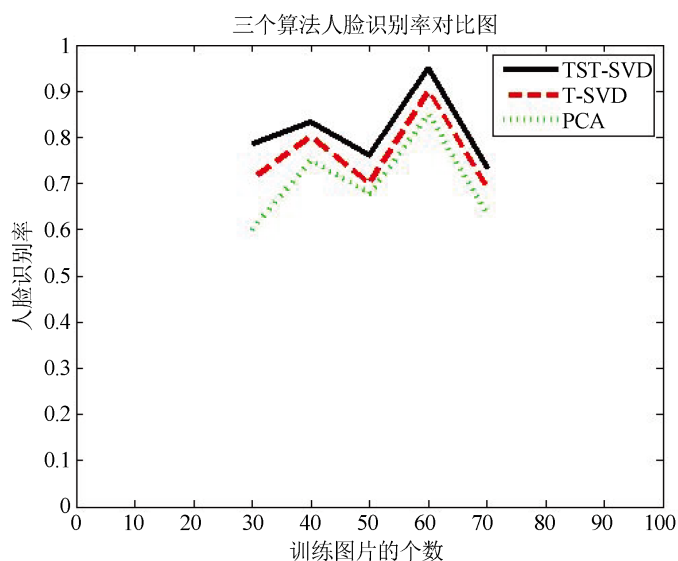


Figure 3. Comparison of algorithm recognition rates of TST-SVD, T-SVD and PCA

图 3. TST-SVD, T-SVD 和 PCA 的算法识别率对比图



Figure 4. Comparison of reconstructed images of TST-SVD and PCA

图 4. TST-SVD 与 PCA 的重构图片对比图

6. 结语

原 T-SVD 算法的截断操作有可能会丢失重要的信息，而我们的 TST-SVD 算法找到了一种可以更好保留重要信息的能力，因为它加入了软阈值操作可以对重要信息进行平滑的去噪。在图像识别方面也取得了不错的效果。当重构图片时，加入软阈值之后的正交投影可以更好地重构图片的主要信息。在跟 PCA 算法做重构图片对比时，TST-SVD 重构的图片明显比 PCA 更好。

致 谢

文章感谢王丽娟老师在写作期间给予我的帮助和鼓励，感谢同门的陪伴和关怀。

参考文献

- [1] Kilmer, M.E., Braman, K., Hao, N. and Hoover, R.C. (2013) Third-Order Tensors as Operators on Matrices: A Theoretical and Computational Framework with Applications in Imaging. *SIAM Journal on Matrix Analysis and Applications*, **34**, 148-172. <https://doi.org/10.1137/110837711>
- [2] Kolad, T. and Bader, B. (2009) Tensor Decompositions and Applications. *SIAM Review*, **51**, 455-500. <https://doi.org/10.1137/07070111X>
- [3] Allen, G. (2012) Sparse Higher-Order Principal Components Analysis. *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, **22**, 27-36.
- [4] Chi, E.C. and Kolda, T.G. (2012) On Tensors, Sparsity, and Nonnegative Factorizations. *SIAM Journal on Matrix Analysis and Applications*, **33**, 1272-1299. <https://doi.org/10.1137/110859063>
- [5] Sun, W.W., Lu, J., Liu, H. and Cheng, G. (2015) Provable Sparse Tensor Decomposition. <https://arxiv.org/pdf/1502.01425v2.pdf>
- [6] Kilmer, M.E. and Martin, C.D. (2011) Factorization Strategies for Third-Order Tensors. *Linear Algebra and its Applications*, **435**, 641-658. <https://doi.org/10.1016/j.laa.2010.09.020>
- [7] The Database of Yale Faces. <http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html>
- [8] Bader, B. and Kolda, T. (2010) Matlab Tensor Toolbox Version 2.4.

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2324-7991, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: aam@hanspub.org