第 40 卷 第 5 期
Vol.40 No.5

计 算 机 工 程
Computer Engineering

2014 年 5 月
May 2014

• 先进计算与数据处理 •　文章编号：1000-3428(2014)05-0045-04　文献标识码：A　中图分类号：TP301

基于张量分解的多维数据填充算法

朱彦君，吴向阳

(杭州电子科技大学计算机学院，杭州 310018)

摘　要：在多维数据分析和处理中，经常会出现部分数据丢失或者部分数据未知的情况，如何利用已知数据的潜在结构对这些缺失数据进行填充是一个亟待解决的问题。目前对于缺失数据填充的研究大多是针对矩阵或者向量形式的低维数据，而对于三维以上高维数据填充的研究则很少。针对该问题，提出一种基于张量分解的多维数据填充算法，利用张量分解中 CP 分解模型的结构特性和分解的唯一性，实现对多维数据中缺失数据的有效填充。通过实验对以三维形式存储的部分数据缺失图像进行填充修复，并与 CP-WOPT 算法进行比较，结果表明，该算法具有较高的准确度以及较快的运行速度。

关键词：缺失数据填充；张量分解；多维数据填充；多维数据分析；多维数据处理；图像修复

Multi-dimensional Data Filling Algorithm Based on Tensor Decomposition

ZHU Yan-jun, WU Xiang-yang

(School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China)

【Abstract】On the multi-dimensional data analysis and processing, data with missing or unknown values is ubiquitous. How to use the potential structure of the known data to reconstruct the missing data is an urgent problem to be solved. Previously, the missing data filling mostly aims at low-dimensional data in matrix or vector format, while research on high-dimensional data above 3D is very few. To solve this problem, this paper proposes a multi-dimensional data filling algorithm based on tensor decomposition, adequately using tensor decomposition's structure and uniqueness of CP model, to realize the multi-dimensional data filling effectively. Filling image with missing data stored in 3D format by experiment and comparison with CP-WOPT algorithm, it proves that this algorithm is not only accurate but also rapid.

【Key words】missing data filling; tensor decomposition; mulit-dimensional data filling; multi-dimensional data analysis; multi-dimensional data processing; image inpainting

DOI: 10.3969/j.issn.1000-3428.2014.05.010

1 概述

在生物信号处理、化学数据分析、数据挖掘、机器学习等方面，经常需要处理部分数据缺失的多维数据[1]。这些缺失的数据可能是因为丢失或者无法观察到而引起的，是不可避免的。现阶段大部分处理数据的算法都是基于完整的数据。例如在数据挖掘中，对含有缺失数据的多维数据进行分析，有可能建立错误的挖掘模型。处理缺失数据的方法大致分为 3 类：删除元组，数据填充和不处理[2]。目前，最有效地分析和处理这些多维数据的方法，就是给缺失数据一个填充值[3-4]。过去，数据填充局限于二维矩阵或者向量形式的低维数据[5-6]，对于高维数据填充的研究很少。Acar E 等人于 2011 年提出了 CP-WOPT 算法[7]，可以对高维数据进行填充，但它是基于梯度值最优的方法对缺失数据进行估计的，因此，填充数据非常不准确。

本文提出一种基于张量分解[8]的多维数据填充算法，称为 CPWF(Candecomp/Parafac Weighted Filling)算法。它可对高维数据进行填充，并且能够充分挖掘所有已知数据的潜在结构[9]，实现对缺失数据的准确填充。为了便于观察和描述，实验中将彩色图像按三维形式存储并进行处理，但该算法不局限于三维数据，任意维数据均可处理。

2 预备知识

2.1 张量的矩阵化

一个多维张量即一个多维数组[8]。张量的矩阵化就是把一个多维数组转化为矩阵的过程。例如，一个 2×3×4 的张量可以转化成 6×4 或者 3×8 的矩阵。张量的元素 $x_{i_1 i_2 \cdots i_N}$ 映射到矩阵元素 (i_n, j) 的方式按照式(1)进行转化：

基金项目：国家自然科学基金资助项目(61003193)；浙江工业大学重中之重学科开放基金资助项目。

作者简介：朱彦君(1988－)，男，硕士研究生，主研方向：大规模数据处理，图形图像处理；吴向阳，副教授、博士。

收稿日期：2013-05-09　**修回日期：**2013-06-09　**E-mail：**156372288@qq.com

$$j=1+\sum_{\substack{k=1\\k\neq n}}^{N}(i_k-1)J_k \tag{1}$$

其中，$J_k=\prod_{\substack{m=1\\m\neq n}}^{k-1}I_m$。

由于张量矩阵化的形式不唯一，不同的矩阵化方法会导致完全不同的计算结果，因此本文中的矩阵化方法如无特殊说明，均指本节所说的矩阵化方法。

为了便于理解，这里用一个实例作为说明。假设有一个张量 $\boldsymbol{X}(3\times4\times2)$：

$$\boldsymbol{X}_{:,:,1}=\begin{bmatrix}1&4&7&10\\2&5&8&11\\3&6&9&12\end{bmatrix}$$

$$\boldsymbol{X}_{:,:,2}=\begin{bmatrix}13&16&19&22\\14&17&20&23\\15&18&21&24\end{bmatrix}$$

如果 $\boldsymbol{X}_{(n)}$ 表示张量 $\boldsymbol{X}$ 的第 n 维展开式，那么有：

$$\boldsymbol{X}_{(1)}=\begin{bmatrix}1&4&7&10&13&16&19&22\\2&5&8&11&14&17&20&23\\3&6&9&12&15&18&21&24\end{bmatrix}$$

$$\boldsymbol{X}_{(2)}=\begin{bmatrix}1&2&3&13&14&15\\4&5&6&16&17&18\\7&8&9&19&20&21\\10&11&12&22&23&24\end{bmatrix}$$

$$\boldsymbol{X}_{(3)}=\begin{bmatrix}1&2&3&4&5&6&7&8&9&10&11&12\\13&14&15&16&17&18&19&20&21&22&23&24\end{bmatrix}$$

2.2 CP 分解模型

一个张量可看成一个多维数组。张量特别是高维的张量，在应用上实现对数据和实际情况较为准确的建模，但是这种建模方法使得张量的计算成为一个巨大问题。因此，需要对张量进行分解，以降低其维数，减少计算的复杂度，并能够最大程度保留原始数据的特征。现有的张量分解法主要有 CP 分解法和 Tucker 分解法。本文只关注 CP 分解法。

为了便于理解，本文以一个三维张量 $\boldsymbol{X}(I\times J\times K)$为例，高于三维的情况可以很容易的推导出来。$\boldsymbol{X}$ 的 CP 分解模型如图 1 所示。

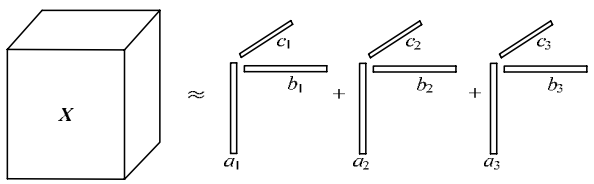


图 1 三维张量的 CP 分解模型

张量的元素值与分解后的向量元素值之间的关系如下：

$$x_{ijk}\approx\sum_{r=1}^{R}a_{ir}b_{jr}c_{kr} \tag{2}$$

其中，R 为给定的正整数；i=1,2,⋯,I；j=1,2,⋯,J；k=1,2,⋯,K。

如果将矩阵 $\boldsymbol{A}(I\times R)$看成由 $\boldsymbol{a}_1$~$\boldsymbol{a}_R$ 共 R 个列向量构成的矩阵，将矩阵 $\boldsymbol{B}(J\times R)$看成由 $\boldsymbol{b}_1$~$\boldsymbol{b}_R$ 共 R 个列向量构成的矩阵，将矩阵 $\boldsymbol{C}(K\times R)$看成由 $\boldsymbol{c}_1$~$\boldsymbol{c}_R$ 共 R 个列向量构成的矩阵，那么 $\boldsymbol{X}$、$\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$ 之间有如下性质：

$$\boldsymbol{X}_{(1)}\approx\boldsymbol{A}(\boldsymbol{C}\odot\boldsymbol{B})^{\mathrm{T}} \tag{3}$$

$$\boldsymbol{X}_{(2)}\approx\boldsymbol{B}(\boldsymbol{C}\odot\boldsymbol{A})^{\mathrm{T}} \tag{4}$$

$$\boldsymbol{X}_{(3)}\approx\boldsymbol{C}(\boldsymbol{B}\odot\boldsymbol{A})^{\mathrm{T}} \tag{5}$$

其中，$\boldsymbol{X}_{(n)}$ 表示张量 $\boldsymbol{X}$ 的第 n 维矩阵化[8]；$\odot$ 表示 Khatri-Rao 积[10]；T 表示矩阵的转置。

CP 分解的目的是求出 $\boldsymbol{A}$，$\boldsymbol{B}$，$\boldsymbol{C}$ 这 3 个矩阵。假设 R 值是给定的，那么问题转化为：

$$\min_{\hat{\boldsymbol{X}}}\left\|\boldsymbol{X}-\hat{\boldsymbol{X}}\right\| \tag{6}$$

其中，$\hat{\boldsymbol{X}}=[\![\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}]\!]$。解决这个问题较好的算法是 ALS 算法[8]。该算法首先初始化 $\boldsymbol{A}$，$\boldsymbol{B}$，$\boldsymbol{C}$，通过 $\boldsymbol{B}$ 和 $\boldsymbol{C}$ 求解 $\boldsymbol{A}$；然后利用 $\boldsymbol{A}$ 和 $\boldsymbol{C}$ 求 $\boldsymbol{B}$；最后使用 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 求 $\boldsymbol{C}$，重复这个过程直到函数值收敛。迭代过程如下：

$$\hat{\boldsymbol{A}}=\boldsymbol{X}_{(1)}\left[(\boldsymbol{C}\odot\boldsymbol{B})^{\mathrm{T}}\right]^{\dagger} \tag{7}$$

$$\hat{\boldsymbol{B}}=\boldsymbol{X}_{(2)}\left[(\boldsymbol{C}\odot\boldsymbol{A})^{\mathrm{T}}\right]^{\dagger} \tag{8}$$

$$\hat{\boldsymbol{C}}=\boldsymbol{X}_{(3)}\left[(\boldsymbol{B}\odot\boldsymbol{A})^{\mathrm{T}}\right]^{\dagger} \tag{9}$$

其中，†表示矩阵的 Moore-Penrose 伪逆[10]。

为减少求 Moore-Penrose 伪逆的代价，式(7)等价于：

$$\hat{\boldsymbol{A}}=\boldsymbol{X}_{(1)}(\boldsymbol{C}\odot\boldsymbol{B})\left(\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C}*\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}\right)^{\dagger} \tag{10}$$

其中，“*”表示矩阵的 Hadamard 积[8]。

由于 CP 分解是唯一的[11]，因此迭代一定会收敛。当 R 大于张量的秩[12]时，$\boldsymbol{X}=[\![\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}]\!]$。

3 CP-WOPT 算法简介

CP-WOPT 算法[7]可以对高维数据进行填充。

给定一个部分数据缺失的张量 $\boldsymbol{X}$，定义一个记录缺失信息的权值张量 $\boldsymbol{W}$ 如下：

$$w_{i_1i_2\cdots i_N}=\begin{cases}1 & \text{如果}x_{i_1i_2\cdots i_N}\text{已知}\\0 & \text{如果}x_{i_1i_2\cdots i_N}\text{缺失}\end{cases} \tag{11}$$

目标函数定义为：

$$f_{\boldsymbol{W}}\left(\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)},\cdots,\boldsymbol{A}^{(n)}\right)=\frac{1}{2}\left\|\left(\boldsymbol{X}-[\![\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)},\cdots,\boldsymbol{A}^{(n)}]\!]\right)\right\|_{\boldsymbol{W}}^{2} \tag{12}$$

式(12)等价于：

$$f_w(\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)},\cdots,\boldsymbol{A}^{(N)})=\frac{1}{2}\|\boldsymbol{Y}-\boldsymbol{Z}\|^2 \tag{13}$$

其中，$\boldsymbol{Y}=\boldsymbol{W}*\boldsymbol{X}$；$\boldsymbol{Z}=\boldsymbol{W}*[\![\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)},\cdots,\boldsymbol{A}^{(N)}]\!]$。问题转化为求 $\boldsymbol{A}^{(1)},\boldsymbol{A}^{(2)},\cdots,\boldsymbol{A}^{(N)}$ 使得目标函数值 f 最小。

$$\frac{\partial f_w}{\partial \boldsymbol{A}^{(n)}} = \left(\boldsymbol{Z}_{(n)} - \boldsymbol{Y}_{(n)}\right)\boldsymbol{A}^{(-n)} \tag{14}$$

在梯度值式(14)中：

$$\boldsymbol{A}^{(-n)} = \boldsymbol{A}^{(N)} \square \cdots \square \boldsymbol{A}^{(n+1)} \square \boldsymbol{A}^{(n-1)} \square \cdots \square \boldsymbol{A}^{(1)} \tag{15}$$

根据式(14)的梯度值，任意基于梯度值的求最优解的方法都可以用于求出式(13)中 $\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \cdots, \boldsymbol{A}^{(N)}$，使得目标函数值 f 最小。

由于 CP 分解是唯一的，而 CP-WOPT 算法则基于局部数据的梯度值估计因子矩阵 $\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \cdots, \boldsymbol{A}^{(N)}$ 的每个元素值。因此只要有一个数据估计不准确，则会使得整个分解模型的数据都不准确。实验证明，该算法确实存在这一问题，当原始张量 $\boldsymbol{X}(20\times20\times20)$ 的每个元素的值在 0~256 之间，f 的值大于 10^7。

4　CPWF 算法

4.1　算法步骤

本文提出算法称为 CPWF 算法。为了便于理解，以三维张量为例，高于三维的情况可以很容易推导出来。

为了记录哪些数据丢失，定义一个和原始张量大小相同的权值张量 $\boldsymbol{W}$ 如下：

$$w_{ijk} = \begin{cases} 1 & \text{如果} x_{ijk} \text{已知} \\ 0 & \text{如果} x_{ijk} \text{缺失} \end{cases} \tag{16}$$

该文提出的 CPWF 算法具体如下：

(1)给定张量 $\boldsymbol{X}$ 和记录缺失数据位置的张量 $\boldsymbol{W}$；

(2)随便填充张量 $\boldsymbol{X}$ 的缺失数据(随机数或者平均值)；

(3)初始化 $\boldsymbol{A}$，$\boldsymbol{B}$，$\boldsymbol{C}$ 和 R；

(4)Repeat

{

$A=X_{(1)}(C\square B)\left(C^{\mathrm{T}}C*B^{\mathrm{T}}B\right)^{\dagger}$;

$B=X_{(2)}(C\square A)\left(C^{\mathrm{T}}C*A^{\mathrm{T}}A\right)^{\dagger}$;

$C=X_{(3)}(B\square A)\left(B^{\mathrm{T}}B*A^{\mathrm{T}}A\right)^{\dagger}$;

for(Every element x_{ijk})

{

If(w_{ijk}==0)

$x_{ijk} = \sum_{r=1}^{R} a_{ir}b_{jr}c_{kr}$;

}

}Until 张量 X 的缺失数据不再发生明显变化

4.2　算法原理

针对三维张量，4.1 节步骤(4)每迭代一次，就把原始张量 $\boldsymbol{X}$ 的每个元素值分散到 $\boldsymbol{A}$，$\boldsymbol{B}$，$\boldsymbol{C}$ 3 个矩阵中，即按式(2)进行映射。此时，随机填充的缺失数据值和已知数据的值都映射到矩阵 $\boldsymbol{A}$，$\boldsymbol{B}$，$\boldsymbol{C}$ 中，并且在一定程度上混合了，即矩阵 $\boldsymbol{A}$，$\boldsymbol{B}$，$\boldsymbol{C}$ 中每个元素值不仅与已知数据相关，而且与随机填充的缺失数据相关。

由于数据具有一定的潜在结构，因此每个缺失位置的元素值根据第一轮迭代后的 $\boldsymbol{A}$，$\boldsymbol{B}$，$\boldsymbol{C}$ 值按照式(2)还原后，会含有一定的新信息，这些信息就是通过 CP 分解模型挖掘出的理论填充值。

重复 4.1 节的步骤(4)，每迭代一次，都会根据已知数据不断修正填充值。由于 CP 分解的唯一性，因此每迭代一次，填充值的变化会越来越小，直到收敛。

5　实验结果与分析

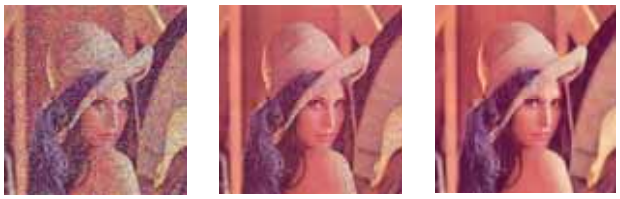
5.1　算法准确性

为了验证算法的准确性，本文实验将部分数据丢失的 BGR 图像像素值写入张量 $\boldsymbol{X}$ 中，$\boldsymbol{X}$ 的第一维为通道数(BGR 图像时长度为 3)，第二三维为图像的高度和宽度，x_{ijk} 表示第 i 通道在位置(j,k)处的像素值。实验结果如图 2、图 3 所示。图 2(a)为按三维形式存储的挖掉一块数据的原始图像，图 2(b)为 CP-WOPT 算法填充结果，图 2(c)为 CPWF 算法填充结果。由于 CP-WOPT 算法不够准确，因此图 2(b)中填充的效果很不好，当局部块缺失时，该算法不能准确挖掘已知数据的潜在结构去填充缺失部分。实验结果表明，对于局部整块缺失的多维数据，CPWF 算法填充效果很好。

(a)原始图像　(b)CP-WOPT 算法填充效果　(c)CPWF 算法填充效果

图 2　R=4 时图像填充效果

图 3(a)为按三维形式存储的添加 30%噪声的原始图像，图 3(b)为 CP-WOPT 算法填充结果，图 3(c)为 CPWF 算法填充结果。被噪声污染的部分被认为是数据缺失的部分。显然，CP-WOPT 算法填充有一些效果，但是帽子和肩膀部分的填充仍然不是很好。而本文提出的 CPWF 算法填充效果明显比 CP-WOPT 算法要好，不仅平滑，而且细节更明显。该实验表明，本文提出的 CPWF 算法对于离散型缺失的多维数据填充效果同样非常好。



(a)原始图像　(b)CP-WOPT 算法填充效果　(c)CPWF 算法填充效果

图 3　R=11 时图像填充效果

5.2　算法效率

针对图 2、图 3 中三维形式的数据，CP-WOPT 算法和 CPWF 算法运行时间如表 1 所示。本文所有实验数据都在联想 Y460N 笔记本上测试运行，CPU 为 i3 380M，2.53 GHz。

内存 4 GB，64 位 Win7 操作系统。由表 1 可以看出，CPWF 算法运算速度明显比 CP-WOPT 算法快。

表 1 CP-WOPT 和 CPWF 算法运行时间比较 s

对比指标	运算时间	
	CP-WOPT 算法	CPWF 算法
图 2 R=4(3×100×88 张量)	7.5	0.7
图 3 R=11(3×240×240 张量)	78.3	6.8

6 结束语

本文提出一种基于张量分解的多维数据填充算法，利用张量分解中 CP 分解模型的结构特性和分解的唯一性，实现了对多维数据中缺失数据的有效填充。实验结果证明，不管是局部数据整块缺失还是全局离散缺失，本文提出的 CPWF 算法都可以很好地填充缺失数据，不仅比 CP-WOPT 算法准确，而且运行速度更快。对于没有规律或者规律不明显的多维数据，如果是全局离散缺失，本文算法仍然可以很好地填充，但当局部数据整块缺失时，算法效果并不理想，因此，需要作进一步研究。另外，本文中的 R 值需要手动调整以达到最佳填充效果，如果取值过大，填充效果反而不好，因此，如何选取一个合适的 R 值也需要作进一步研究。

参考文献

[1] Smilde A, Bro R, Geladi P. Multi-way Analysis: Applications in the Chemical Sciences[M]. [S. l.]: Wiley, 2004.

[2] Pearson R K. The Problem of Disguised Missing Data[J]. ACM SIGKDD Explorations Newsletter, 2006, 8(1): 83-92.

[3] Witten I H, Frank E. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations[M]. 2nd ed. [S. l.]: Morgan Kaufmann, 2005.

[4] Han Jiawei, Kamber M. Data Mining Concepts and Techniques[M]. 2nd ed. [S. l.]: Morgan Kaufmann, 2006.

[5] Nocedal J, Wright S J. Numerical Optimization[M]. [S. l.]: Springer, 1999.

[6] 邹 薇, 王会进. 基于朴素贝叶斯的 EM 缺失数据填充算法[J]. 微型机与应用, 2011, 30(16): 75-77, 81.

[7] Acar E, Dunlavy D M, Kolda T G, et al. Scalable Tensor Factorizations with Missing Data[C]//Proceedings of SIAM International Conference on Data Mining. Columbus, USA: [s. n.], 2011: 41-56.

[8] Kolda T G, Bader B W. Tensor Decompositions and Applications[J]. SIAM Review, 2009, 51(3): 455-500.

[9] 廖志芳, 李 玲, 刘丽敏, 等. 三部图张量分解标签推荐算法[J]. 计算机学报, 2012, 35(12): 2625-2632.

[10] Golub G H, Vanloan C F. Matrix Computations[M]. [S. l.]: Johns Hopkins University Press, 1996.

[11] Kruskal J B. Rank, Decomposition, and Uniqueness for 3-way and N-way Arrays[M]. Amsterdam, the Netherlands: North-Holland Publishing Co., 1989: 7-18.

[12] Hastad J. Tensor rank is NP-complete[J]. Journal of Algorithms, 1990, 11(4): 644-654.

编辑 陆燕菲

(上接第 44 页)

GPU 的发展势头异常迅猛，随着时间的推移，以及人们对计算机并行处理能力要求的进一步提高，通过基于 CUDA 的 GPU 并行处理热问题会越来越多，因此，如何提高 GPU 的并行计算能力将是未来学者们的一个重要研究课题。相信随着硬件的不断发展，GPU 的处理能力也将进一步提高。虽然本文提出的一维热传导 GPU 并行算法较之前的 CPU 串行算法在时间效率上提高了几百倍，但这还远没有发挥出 GPU 并行计算的优势。若将此算法扩展为 GPU 集群上的并行算法，那么计算能力会比单 GPU 的算法能力强得多，下一步将研究工作环境更高的加速比。

参考文献

[1] 王 梁. 二维稳态热传导问题 CPU/GPU 并行求解[EB/OL]. [2013-05-13]. http://tech.it168.com/a2010/0722/1081/000001081200.shtml.

[2] Frezzotti A, Ghiroldi G P. Solving the Boltzmann Equation on GPUs[EB/OL]. (2010-05-28). http://arxiv.org/abs/1005.5405.

[3] Rumpf M, Strzodka R. Using Graphics Cards for Quantized FEM Computations[C]//Proceedings of VIIP'01. Marbella, Spain: [s. n.], 2001: 98-107.

[4] 百度百科. 热传导[EB/OL]. [2013-05-13]. http://baike.baidu.com/view/348360.htm.

[5] 李夏云, 陈传淼. 用龙格-库塔法求解非线性方程组[J]. 数学理论与应用, 2008, 28(2): 62-65.

[6] 林 茂, 董玉敏, 邹 杰. GPGPU 编程技术初探[J]. 软件开发与设计, 2010, (2): 15-17, 23.

[7] 孙敏杰. CPU 架构和技术的演变看 GPU 未来发展[EB/OL]. [2013-05-13]. http://www.pcpop.com/doc/0/521/521832_all.shtml.

[8] 孟小华, 刘坚强, 区业祥. 基于 CUDA 的拉普拉斯边缘检测算法[J]. 计算机工程, 2012, 38(18): 190-193.

[9] 李 超. 论 GPU-CPU 协作计算模式的应用研究[J]. 电子商务, 2010, (11): 54.

[10] 郁志辉. 高性能计算并行编程技术-MPI 并行程序设计[M]. 北京: 清华大学出版社, 2001.

[11] 朱丽莎. 基于 GPU 的一维体系热传导算法研究[D]. 广州: 暨南大学, 2011.

[12] 朱宗柏. 多重网格方法的并行化及其在传热数值分析中的应用[J]. 武汉交通科技大学学报, 2000, 24(4): 351-354.

编辑 任吉慧

基于张量分解的多维数据填充算法

作者： 朱彦君， 吴向阳， ZHU Yan-jun， WU Xiang-yang
作者单位： 杭州电子科技大学计算机学院,杭州,310018
刊名： 计算机工程 ISTIC PKU
英文刊名： Computer Engineering
年，卷(期)： 2014(5)