

基于分而治之策略的稀疏张量多层次数据挖掘

周溜溜, 业宁, 徐昇, 严敏利, 孙伟

(南京林业大学 信息技术学院, 江苏 南京 210037)

摘要: 分析了 MET 算法的局限性以及 out-of-core 方法的特点, 融合了两种算法的思想提出一种基于分而治之策略的多层次数据挖掘算法(DRMET), 避免了计算过程中可能造成的维数灾难问题, 克服了 MET 算法执行效率不高的缺陷, 同时继承了 MET 内存开销小的优点; 实验结果表明: 新算法在不增加存储空间的前提下大大约减了 MET 的时间开销, 其效率大约是 MET 的 1.86~15.85 倍。

关键词: 分而治之; 多层次数据挖掘; DRMET; 时间开销

中图分类号: TP311

文献标识码: A

文章编号: 1000-7180(2011)10-0204-05

Multi-aspect Data Minizing on the Sparse Tensor Based on Dividing and Ruling

ZHOU Liu-liu, YE Ning, XU Sheng, YAN Min-li, SUN Wei

(College of Computer Science and Technology, Nanjing Forestry University, Nanjing 210037, China)

Abstract: Analyzed the limitations of MET algorithm and feature of out-of-core, mixed ideas of the two algorithms and then giving a method called DRMET which based on the strategy of dividing and ruling for multi-aspects data mining; avoid the problem of "dimension disaster" while overcomed the MET's limitation of low efficiency, and inherit its advantage on small memory consumed; experiment gives the result: new algorithm reduced time cost from MET substantially, its efficiency improved about 1.86~15.85 times compared to MET.

Key words: dividing and ruling; multi-aspect data mining; DRMET; time cost

1 引言

随着信息产业的发展和社会进步, 无论在信息领域^[1]或社会网络领域都会产生海量数据, 这些数据的组织形式往往是高维的, 很自然利用张量(高维数组)表示。例如: email 交往数据和因特网流量等; 如何高效分析管理此类数据便成为研究者所关心的问题。目前为止, 科学家们对紧密度高的张量型数据的挖掘给予了较多的关注^[2], 产生了很多实用的算法, 但经研究观察可发现, 以上所列举涉及的数据集具有稀疏性质, 譬如 email 交往数据, 被视为一个四阶张量, 且四个维度分别表示发送者、接收者、关键词、时间戳。在此类数据集中, 也许发送者 A 只与自己的几位好友常常发生通信关系, 而与数据集中其

他人没有任何来往, 这样数据组织者会将 A 与其非好友之间相应值赋为零, 经验可知这样的零值大量存在数据中; 因此设计高效的算法用于此类多层次稀疏数据进行有效挖掘便成了迫切而有意义的工作。这些工作离不开张量的分解, 目前主流的分解方法是 CP 分解^[3]和 Tucker^[4]分解。Kolda^[5]等于 2008 年对稀疏张量计算时内存消耗过大给出了解决方案, 避免数据 Tucker 分解时产生的 intermediate blowup 问题, 提出了 MET^[5]算法。该算法可以令在分解过程中内存消耗较 Tucker ALS^[6]方法节省近 1 000 倍; 但 MET 在设计初始侧重于内存的有效利用, 因此该算法的测试结果表明运行时间尽管与 Tucker ALS 有着可比性, 但无太多的效率提升。此外 Wang^[7]等于 2005 年设计的 out-of-core 算法

收稿日期: 2011-04-25; 修回日期: 2011-05-28

基金项目: 江苏省自然基金项目(BK2009393); 国家自然科学基金(30671639); 江苏省青蓝工程学术带头人项目

是一款为保留图像结构信息而进行图像张量化再逐区域分解方法. 图像是高密度数据, 几乎没有零点, 因此如何将这一思想合适的用于稀疏型张量便成了有意义且充满挑战性的工作.

基于以上考虑, 本文融合 MET 算法与 out-of-core 思想, 提出了基于分而治之策略的算法, 主要步骤先将原始稀疏张量按规则进行分块操作, 拆解为一个较小的子张量, 然后在各子张量上进行 MET 算法操作, 并使得在矩阵化以后利用自身转置乘再进行 SVD 分解, 从而巧妙避免了可能形成的维数灾难问题, 最后将各子张量操作结果按规则拼接, 便得到原张量分解结果, 称此算法为 DRMET, 实验表明 DRMET 不仅继承了 MET 算法内存消耗的性能优势, 且在时间性能上有很大提升, 时间开销大约为 MET 的 5.4%~53.8%.

2 背景知识

2.1 Tucker 分解的研究与应用

张量分解概念由 Hitchcock 于 1927 年在文献[8]中首次提出, 现张量分解已成功应用于数据挖掘、机器学习^[2-3]等领域. Tucker 分解在这些领域发挥着重要作用; 基于 ALS 原则的 Tucker ALS^[6]方法成为数据挖掘领域分析多层次数据的主要手段, 由 Kroonenberg 等分析序为 3 的张量时所提, 后扩展至多层次学习. 但大规模稀疏型张量的学习目前学者们给予的关注还不多; 例如在文献[9]首先给出了张量的一个前提条件: 假设 $\mathbf{A}$ 是非稀疏张量, 而给出此假定目的之一便是防止内存溢出与. Kolda^[5]等于 2008 年提出的 MET 算法, 成功克服了大规模稀疏张量计算可能产生的内存溢出现象, 他们将这一现象称为“intermediate blowup”问题. 但 MET 也存在不足, 就是对张量的运算效率上没有很大提升, 往往不能满足应用即时性要求.

2.2 基本概念和操作定义

定义 1 张量及其表示: 张量数据即多维数组, 称数组维数为张量的序, 用 N 表示; 字母 $\mathbf{R}$ 表示张量, $\mathbf{A}^{(n)}$ 为某矩阵序列 $\{\mathbf{A}\}$ 中第 n 个矩阵; 黑体小写字母表示向量, 如 $\mathbf{a}$; 普通小写字母表示标量, 如 a .

定义 2 张量的范数: 存在序为 N 的张量 $\mathbf{R} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, 则 $\mathbf{R}$ 的范数表示为 $\|\mathbf{R}\|$.

定义 3 张量纤维与张量切片: 张量纤维是张量数据某特定方向的一维分量, 其表现形式为向量; 如对于一个序为 3 的张量型数据 $\mathbf{R}^{2 \times 2 \times 2}$, 其所有沿

第 1 维度张量纤维分别为 $\mathbf{a} = (r_{111}, r_{211})$, $\mathbf{b} = (r_{112}, r_{212})$, $\mathbf{c} = (r_{111}, r_{212})$, $\mathbf{d} = (r_{112}, r_{211})$, 张量切片的概念与张量纤维类似, 它取决于两个特定方向的张量片段, 其表现形式为矩阵.

定义 4 张量的 n 模矩阵乘: 存在张量 $\mathbf{R}^{I_1 \times I_2 \times \dots \times I_N}$ 和矩阵 $\mathbf{U}^{J \times J_N}$, 则 $\mathbf{R}$ 的 n 模矩阵乘表示为 $\mathbf{R} \times_n \mathbf{U}$, 结果仍是序为 N 的张量, 且 $(\mathbf{R} \times_n \mathbf{U})_{i_1 \dots i_{n-1} j_{n+1} \dots i_N} = \sum_{i_n} r_{i_1 i_2 \dots i_n} \times a_{j n}$, 矩阵乘运算满足交换率^[9].

定义 5 张量的 n 模向量乘: 存在张量 $\mathbf{R}$ 和向量 $\mathbf{u} \in \mathbb{R}^{I_n}$, 则 $\mathbf{R}$ 的 n 模向量乘表示为 $\mathbf{R} \times_n \mathbf{v}$, 结果是序为 $N-1$ 的张量, 且 $(\mathbf{R} \times_n \mathbf{U})_{i_1 \dots i_{n-1} i_{n+1} \dots i_N} = \sum_{i_n} r_{i_1 i_2 \dots i_n} \times v_{i_n}$.

3 DRMET 算法

3.1 基础数学推导

Tucker 分解为获得保留原始信息的核张量 $\mathbf{R}_2$ 和列正交矩阵序列 $\{\mathbf{A}\}$, 使得 $\mathbf{R}_1 \approx \mathbf{R}_2 \times \{\mathbf{A}\} = \mathbf{R}_2 \times_1 \mathbf{A}^{(1)} \times \dots \times_N \mathbf{A}^{(N)}$ 且满足信息损失最小, 即:

$$\min_{\mathbf{R}_2, \{\mathbf{A}\}} \|\mathbf{R}_1 - \mathbf{R}_1 \times \{\mathbf{A}\}\|^2 \quad (1)$$

由于 $\mathbf{A}^{(n)}$ 是列正交矩阵, 故存在式(2):

$$\mathbf{R}_2 \approx \mathbf{R}_1 \times_1 \mathbf{A}^{(1)\top} \times \dots \times_N \mathbf{A}^{(N)\top} \quad (2)$$

信息损耗为

$$\min \|\mathbf{R}_1 - \mathbf{R}_2 \times \{\mathbf{A}\}\| \quad (3)$$

其中

$$\|\mathbf{R}_1 - \mathbf{R}_2 \times \{\mathbf{A}\}\| = \|\mathbf{R}_1\|^2 + \|\mathbf{R}_2\|^2 - 2\langle \mathbf{R}_1, \mathbf{R}_2 \times \{\mathbf{A}\} \rangle = \|\mathbf{R}_1\|^2 - \|\mathbf{R}_2\|^2 \quad (4)$$

因 $\mathbf{R}_1$ 已知, 所以式(3)可以转化为最大化 $\|\mathbf{R}_2\|$

$$\text{即: } \max \|\mathbf{R}_2\| = \max \|\mathbf{R}_1 \times \{\mathbf{A}^\top\}\| \quad (5)$$

令 $\mathbf{R}_3 = \mathbf{R}_1 \times_{-n} \{\mathbf{A}^\top\}$, 则式(5)可以写成

$$\max \|\mathbf{R}_1 \times \{\mathbf{A}^\top\}\| = \max_{\mathbf{A}^{(n)}} \|\mathbf{A}^{(n)\top} \mathbf{R}_{3(n)}\| \quad (6)$$

观察式(6)可知求解 $\mathbf{A}^{(n)}$ 很重要, 计算可以转换为对高阶 SVD 分解; 故 $\mathbf{R}_3$ 的求解便至关重要, 对该问题解决的思路迄今大都以 Tucker ALS^[6]为基础.

3.2 算法思想

Kolda^[5]等提出了 MET 算法, 解决了“intermediate blowup”问题. MET 核心思想. 对 $\mathbf{R}_3$ 求过程解将元素间运算改为纤维甚至切片运算; 张量纤维是一个向量, 而张量每做一回向量乘运算序都会减 1, 这样便达到减少内存开销目的; 在此我们利用

序为 3 的张量举例说明: 设 $\mathbf{R}_3$ 求解第一次优化:

$$\mathbf{A}^{(1)}: \mathbf{R}_3 \leftarrow \mathbf{R}_1 \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)};$$

(1) Tucker ALS 标准运算: 该方法是首先矩阵乘做 $\mathbf{R}_1 \times_2 \mathbf{A}^{(2)}$ 得出一个大小为 $I_1 \times J_2 \times I_3$ 张量, 再与 $\mathbf{A}^{(3)}$ 运算, 进而获得 $I_1 \times J_2 \times J_3$ 大小的张量 $\mathbf{R}_3$; 再通过高阶 SVD 分解获得优化后 $\mathbf{A}^{(1)}$;

(2) 切片化计算: 首先将 $\mathbf{A}^{(2)}$ 拆解成 J_2 个行向量, 利用 $\mathbf{R}_1$ 分别于这些向量做向量乘运算后得出一个矩阵再与 $\mathbf{A}^{(3)}$ 做矩阵乘运算, 并将每次得出的矩阵 (张量切片) 组合便获得 $\mathbf{R}_3$;

(3) 纤维化计算: 在切片化基础上将 $\mathbf{A}^{(3)}$ 拆分成向量, 这样 $\mathbf{R}_1$ 连续做两次向量乘运算最终得到的是一个向量且共有 $J_2 \times J_3$ 个纤维, 最后组合成待求张量 $\mathbf{R}_3$.

3.3 DRMET 算法原理与步骤

我们对 MET 研究后发现并没有获得效率提升, 因此结合分而治之思想, 提出 DRMET 算法. DRMET 算法主要思想是将原待分解张量划分成一些小的子张量, 通过利用矩阵自身转置乘再分解的技巧避免了可能产生的维数灾难问题, 再对这些小块的张量利用 MET 算法进行操作, 得到核张量的相应子块, 最后将这些子块按拆分逆过程组合得出所求的结果集合. 现存在一个序为 N 的张量 $\mathbf{R}^{I_1 \times I_2 \times \dots \times I_N}$, 将每一阶大小 I_n 平均划分为 P_n 块, 则每一块大小应为 I_n/P_n (假设该值恰为整数), 那么张量 $\mathbf{R}$ 共可以分为 $P_1 P_2 \dots P_N$ 个子张量, 要求 $I_n/P_n \ll I_n$; 同时矩阵序列的行列拆解划分关系应满足 $J_n/Q_n \leq I_n/P_n$, 其中 Q_n 为矩阵行数的划分数; 正如 out-of-core 方法所设计的那样, 当一个原始张量划分为 $P_1 P_2 \dots P_N$ 块后, 其中某一块记为 $\mathbf{R}_{p_1 p_2 \dots p_N}$; 矩阵 $\mathbf{A}^{(n)}$ 序列中矩阵被划分为 $Q_n \times P_n$ 块, 且块的运算遵循以下规则^[7]:

$$\mathbf{R}'_{p_1 p_2 \dots p_N} = \sum_{p_n=1}^{P_n} \mathbf{R}_{p_1 p_2 \dots p_n \dots p_N} \times \mathbf{A}_{p_n}^{(n)} \times P_n \quad (7)$$

这一运算规则类似于张量的 n 模矩阵乘法; DRMET 同时在矩阵的 SVD 分解时利用了性质: 张量 $\mathbf{R}^{I_1 \times I_2 \times \dots \times I_N}$ 对应矩阵 $\mathbf{A}^{(n)}$ 大小为 $I_n \times I_1 \times I_2 \dots I_{n-1} \times I_{n+1} \dots I_N$, 其中 $I_n \ll I_1 I_2 \dots I_{n-1} I_{n+1} \dots I_N$, 所以直接对 $\mathbf{A}$ 做 SVD 操作的结果为 $\mathbf{U}\mathbf{S}\mathbf{U}^T$, 但只需要其中 $\mathbf{U}$ 矩阵, 但过程中会因为维数较高造成维数灾难, 我们知道 $\mathbf{A}\mathbf{A}^T$ 的大小为 $I_n \times I_n$, 且 $\mathbf{A}\mathbf{A}^T$ 做 SVD 操作结果 $\mathbf{U}\mathbf{S}^2\mathbf{U}^T$ 是, 可以得到所需的 $\mathbf{U}$ 矩阵并避免了维数灾难问题同时减少了计算量. 综合以上推导分析, 现给出 DRMET 算法的具体实现步骤与伪代码如算法 1

所示:

算法 1: DRMET

输入: 序为 N 的张量数据 $\mathbf{R}^{I_1 \times I_2 \times \dots \times I_N}$ 和核张量

大小: $\mathbf{J}_n, n=1, 2, \dots, N$;

输出: 核张量 $\mathbf{R}_2^{J_1 \times J_2 \times \dots \times J_N}$ 以及矩阵序列 $\{\mathbf{A}\}$ 且 $\mathbf{A}^n \in \mathbf{R}^{I_n \times J_n}$;

1 将 $\mathbf{R}_1$ 拆解为 $P_1 P_2 \dots P_n \dots P_N$ 个子块;

2 for $i=1$ to N ;

3 沿着第 i 维度对 $\mathbf{R}_1$ 每一子块矩阵化展开获得 $\mathbf{R}'_{p_1 p_2 \dots p_n \dots p_N}$;

4 $\mathbf{R}_{p_1 \dots p_n \dots p_N} \leftarrow \mathbf{R}'_{p_1 \dots p_n \dots p_N} \leftarrow \mathbf{R}'_{p_1 \dots p_n \dots p_N}^T$;

5 对 $\mathbf{R}_{p_1 p_2 \dots p_n \dots p_N}$ 做 SVD 分解获得 $\mathbf{U}_i$;

6 取前 J_n 个奇异向量得初始矩阵 $\mathbf{A}_{p_i}^{(i)}$;

7 $\mathbf{A}^{(i)} \leftarrow$ 合并每一子矩阵 $\mathbf{A}_{p_i}^{(i)}$;

8 end;

9 构造矩阵序列 $\{\mathbf{A}\}$;

10 将 $\mathbf{R}_1$ 拆解为 $P_1 P_2 \dots P_n \dots P_N$ 个子块;

11 for $n=1$ to N ;

12 $\mathbf{R}_{3[p_1 \dots p_N]} \leftarrow \mathbf{R}_{1[p_1 \dots p_N]} \times_{-n} \{\mathbf{A}_{p_n}^T\}$;

13 沿着第 n 维度对 $\mathbf{R}_{3[p_1 p_2 \dots p_n \dots p_N]}$ 矩阵化展开获得 $\mathbf{R}'_{3n[p_1 \dots p_N]}$;

14 $\mathbf{R}_{3n[p_1 \dots p_N]} \leftarrow \mathbf{R}'_{3n[p_1 \dots p_N]} \mathbf{R}'_{3n[p_1 \dots p_N]}^T$;

15 对 $\mathbf{R}_{3n[p_1 p_2 \dots p_n \dots p_N]}$ 做 SVD 分解;

16 取前 J_n 个奇异向量构成矩阵 $\mathbf{A}_{p_n}^{(n)}$;

17 $\mathbf{A}^{(n)} \leftarrow$ 合并每一子矩阵 $\mathbf{A}_{p_n}^{(n)}$;

18 $\mathbf{R}_3 \leftarrow$ 合并所有子张量块 $\mathbf{R}_{3[p_1 p_2 \dots p_n \dots p_N]}$;

19 end;

20 判断是否满足收敛条件;

21 构造新的矩阵序列 $\{\mathbf{A}\}$;

22 $\mathbf{R}_2 \leftarrow \mathbf{R}_3 \times_N \mathbf{A}^{(N)}$;

23 return $\mathbf{R}_2$ and $\{\mathbf{A}\}$;

算法 1 为 DRMET 主要步骤, 第 1 行为张量拆解; 2~8 行按各维度展开利用 SVD 分解获得奇异矩阵, 尤其 4~5 行利用 $\mathbf{A}\mathbf{A}^T$ 的 SVD 操作也可以得到直接对进行 SVD 操作所需的矩阵, 提高计算效率的同时避免了维数灾难; 第 9 行完成矩阵序列初始化; 10~20 行继续采用分而治之策略, 利用依次优化反复迭代每个矩阵直至收敛; 21~23 行确定矩阵序列后获得核张量完成求解.

4 实验与分析

实验运行环境参数: CPU 是主频为 3.2 GHz 的 Intel Pentium 4, 内存 10 GB, 硬盘 250 GB, 系统 Windows XP SP3, 以 MATLAB R2010b 版本作为测试工

具;实验对 DRMET 算法做了性能测试,同时与 MET 做了比较分析.

4.1 数据生成器

为更好测试 DRMET 性能,令原始张量与核张量为方形,即, $I_1=I_2=\cdots=I_N, J_1=J_2=\cdots=J_N$;先随机生成一个大小 $J\times J\times\cdots\times J$ (共 N 个)的核张量 $\mathbf{R}_2$ 及 N 个随机正交阵 $\mathbf{A}^{(n)}\in\mathbf{R}^{I\times J}$;再将 $\mathbf{R}_2$ 与矩阵依次做乘运算,容易获得大小 $I\times I\times\cdots\times I$ 的张量 $\mathbf{R}$,显然 $\mathbf{R}$ 非稀疏,故我们保留中取得前较大的元素 (D 表示非零元素比),令其他元素为零,则新张量便是一个稀疏型张量数据记为 $\mathbf{R}_1$;该数据生成器产生数据的时间不计算在实验过程.

4.2 实验数据及其性能分析

本节进行性能测试,利用固定参数法评估 MET 和 DRMET 算法执行效率,即分别通过改变 D 、 I 、 N 、 J ,固定剩余参数值的方法获得运行时间;对于算法另一性能信息损耗量原本是极为重要的参考指标,由于在两种算法中损耗较低(约为 $1e-8\sim 1e-16$)而不做比较.下面给出具体实验结果数据,利用图 1~4 的可视化效果呈现,图的纵坐标值均为以 10 为底的对数取值,表中所给出的时间值也是以 10 为底的对数取值.

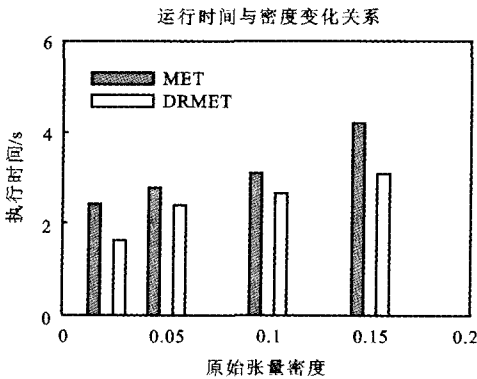


图1 密度与运行时间关系
($N=3, I=400, J=10$)

图 1 中通过固定 N, I, J 值变化 D 值的方法考察了两种算法执行时间,均随着 D 值的变大而变大, DRMET 变化幅度小于 MET,整体效率约为 MET 的 2.51~12.59 倍;

图 2 中通过固定 N, D, J 值变化 I 取值给出了执行时间, DRMET 整体性能约为 MET 的 1.86~7.24 倍.

图 3 中通过改变 N 值得出新算法性能约是 MET 的 3.16~12.59 倍;

图 4 通过改变 J 值观察新算法整体性能约是

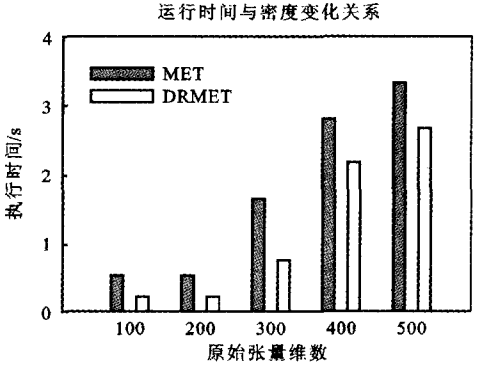


图2 维数与运行时间关系
($D=0.05\%, N=3, J=10$)

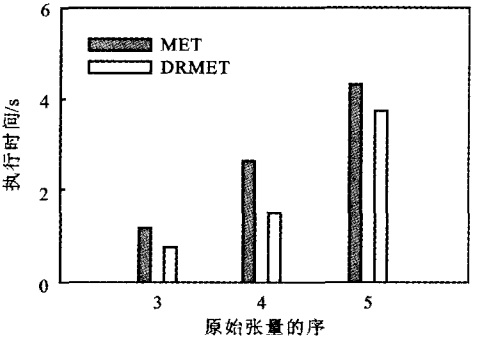


图3 序与运行时间关系
($D=0.05\%, I=400, J=10$)

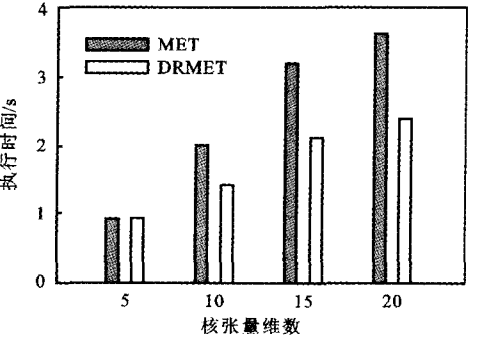


图4 核维数与运行时间关系
($D=0.05\%, I=400, N=3$)

MET 的 3.98~15.85 倍;可见所求核张量大小参数对算法效率有着明显的影响.从图 1~4 分析得出结论:本文算法 DRMET 在对稀疏型张量进行多层次挖掘时较 MET 在效率上有极大改进.纵观所有实验数据, DRMET 算法在本文所研究领域内,运行时间较 MET 方法节省了约 1.86~15.85 倍,所以该算法是高效实用的.

5 结束语

本文基于 out-of-core 算法特点和 MET 思想,提

出采用分而治之策略的算法 DRMET,使之适用于稀疏型张量挖掘;该算法不仅继承了 MET 内存开销小的优点,同时在计算过程中巧妙地避免了因规模较大所造成的维数灾难问题,提升了算法执行效率,克服了 MET 挖掘运行时间过长缺陷;最后我们利用数据集测试了两款算法(MET 和 DRMET)的运行效率并加以比较分析,实验证明:所设计的新算法是高效、充分可适用的,将 MET 执行效率提升了约 1.86 至 15.85 倍。接下来,我们将要做的工作是考虑如何进一步提升其性能,尤其是当所挖掘张量并非方形张量时该如何使之适应。

参考文献:

- [1] Sun J, Papadimitriou S, Yu P. Window-based tensor analysis on high-dimensional and multi-aspect streams. [C]// proceedings of the 6th International Conference on Data Mining. Washington, DC, USA; IEEE Computer Society, 2006: 1076-1080.
- [2] Henrion R. Body diagonalization of core matrices in three-way principal components analysis: Theoretical bounds and simulation[J]. Journal of Chemometrics, 1993 (7): 477-494.
- [3] Carroll J D, Chang J J. Analysis of individual differences in multidimensional scaling via an N-way generalization of 'Eckart-Young' decomposition[J]. Psychometrika, 1970, 35(2): 283-319.
- [4] Tucker L R. Some mathematical notes on three-mode factor analysis[J]. Psychometrika, 1966, 31(2): 279-311.
- [5] Kolda T G, Sun J. Scalable tensor decompositions for multi-aspect data mining[C]// proceedings In ternational Conference on Data Mining. Washington, DC; IEEE, 2008.
- [6] Kolda T G, Bader B W. Tensor decompositions and applications[J]. SIAM Review, 2008(10): 1046-1092.
- [7] Wang H C, Wu Q, Shi L, et al. Out-of-core Tensor Approximation of multi-dimensional matrices of visual data[J]. SIGGRAPH, 2005, 24(3): 340-348.
- [8] Hitchcock F L. The expression of a tensor or a polyadic as a sum of products[J]. Journal of Mathematics and Physics, 1927(6): 164-189.
- [9] Bader B W, Kolda T G. Efficient MATLAB computations with sparse and factored tensors[J]. SIAM Journal on Scientific Computing, 2007, 30(1): 205-231.

作者简介:

周溜溜 男, (1987-), 硕士. 研究方向为数据挖掘机器学习.

(上接第 203 页)

- [4] Kailath T. The divergence and bhattacharyya distance measures in signal selection[J]. IEEE Transactions on Communication Technology, 1967, 15(1): 52-60.
- [5] Asuncion A, Newman D. UCI machine learning repository [EB/OL]. [2011-04-10]. <http://www.ics.uci.edu/mllearn/MLRepository.html>.
- [6] Bradley A P. The use of the area under the roc curve in the evaluation of machine learning algorithms[J]. Pattern Recognition, 1997, 30(7): 1145 - 1159.
- [7] Hand D J, Till R J. A simple generalisation of the area under the roc curve for multiple class classification problems [J]. Machine Learning, 2001, 45(2): 171 - 186.

作者简介:

钱 祺 男, (1986-), 硕士. 研究方向为机器学习与数据挖掘;

姜 远 女, (1976-), 博士, 副教授. 研究方向为机器学习与数据挖掘.