DOI：10.11830/ISSN.1000-5013.202110006

结合 PCA 的 *t*-SNE 算法的并行化实现方法

徐旸，王佳斌，彭凯

（华侨大学 工学院，福建 泉州 362021）

摘要： 为了提高大数据环境下高维非线性数据的处理速度和精确度，提出一种结合主成分分析（PCA）的基于 *t* 分布的随机近邻嵌入（*t*-SNE）算法. 首先，通过主成分分析法对原始数据进行预处理，去除噪声点；然后，结合 *t*-SNE 算法，构建 *K* 最邻近（*K*-NN）图，以表示高维空间中数据的相似关系；最后，在 Spark 平台上进行并行化运算，并在 BREAST CANCER，MNIST 和 CIFAR-10 数据集上进行实验. 结果表明：文中算法完成了高维数据至低维空间的有效映射，提升了算法的效率和精确度，可应用于大规模高维数据的降维.

关键词： 高维数据；Spark 平台；降维；可视化；*t*-SNE 算法

中图分类号： TP 391　　**文献标志码：** A　　**文章编号：** 1000-5013(2022)05-0685-08

Parallel Implementation Method of *t*-SNE Algorithm Combined With PCA

XU Yang, WANG Jiabin, PENG Kai

(College of Engineering, Huaqiao University, Quanzhou 362021, China)

Abstract: In order to improve the processing speed and accuracy of high-dimensional nonlinear data based on *t* distribution in the big data environment, a random neighbor embedding (*t*-SNE) algorithm combined with principal component analysis (PCA) is proposed. Firstlly, the original data is preprocessed by the principal component analysis method to remove noise points. Then, combined with the *t*-SNE algorithm, the *K* nearest neighbor (*K*-NN) graph is constructed to represent the similarity relationship of the data in the high-dimensional space. Finally, on the Spark platform carry out parallel operation and experiment on BREAST CANCER, MNIST and CIFAR-10 data sets. The results show that the proposed algorithm complete the effective mapping of high-dimensional data to low-dimensional space, improves the efficiency and accuracy of the algorithm, and can be applied to large-scale high-dimensional data dimensionality reduction.

Keywords: high-dimensional data; Spark platform; dimensionality reduction; visualization; *t*-SNE algorithm

大数据可视化是大数据研究领域的核心内容之一[1]，其中，高维数据可视化尤为关键，降维可视化方法[2]是高维数据可视化的一种重要技术，它将高维数据转换为二维或三维的低维数据，并可视化于散点图中[3]. 数据降维[4]的目标是在低维空间映射数据内部结构，并充分地保留原来高维数据的重要信息. 梁京章等[5]提出核主成分分析（KPCA）法，通过核函数的作用，将数据映射至高于现存的维度中，再通过线性降维的手段进行处理. Roweis 等[6]提出的局部线性嵌入（LLE）和 Tenenbaum 等[7]提出的等距离特征映射（ISOMAP）是流行学习中的代表算法，这两种算法在高维空间中观察数据的最潜层特征

收稿日期： 2021-10-06

通信作者： 王佳斌（1974-），男，副教授，主要从事物联网、云计算和大数据的研究. Email：fatwang@hqu.edu.cn.

基金项目： 国家自然科学基金青年科学基金资助项目（61505059）

后,根据两个维度空间的映射关系,将数据的主要特征关系映射于低维空间. Maaten 等[8]提出基于 t 分布的随机近邻嵌入(t-SNE)算法,通过高维空间和低维空间中的条件概率关系,采用长尾 t 分布实现降维效果. 基于此,本文提出一种结合主成分分析(PCA)的 t-SNE 算法的并行化实现方法.

1 相关工作

1.1 主成分分析法

主成分分析法的目的是在尽可能减小原始信息损失的同时压缩、简化数据,去除冗余的噪声数据. 主成分分析法提取数据的主要特征,将原有数据重构为新的相互无关的综合变量集,新变量集的数据量小于等于原数据量. 主成分分析法能够有效地展示各变量的映射关系和内部结构.

主成分分析法主要有以下 3 个计算步骤.

1) 建立初始关系数据矩阵 $\boldsymbol{X}$,有

$$\boldsymbol{X}=\begin{pmatrix} x_{1,1} & \cdots & x_{1,m} \\ \vdots & & \vdots \\ x_{n,1} & \cdots & x_{n,m} \end{pmatrix}. \tag{1}$$

2) 标准化初始关系数据矩阵元素为

$$x_{i,j}^{*}=\frac{x_{i,j}-\overline{x}_j}{\sqrt{\mathrm{var}(x_j)}}. \tag{2}$$

式(2)中:$\overline{x}_j$,$\mathrm{var}(x_j)$分别为第 j 列向量的均值和方差,即

$$\overline{x}_j=\frac{1}{n}\sum_{i=1}^{n}x_{i,j}, \tag{3}$$

$$\mathrm{var}(x_j)=\frac{1}{n-1}\sum_{1}^{n}(x_{i,j}-\overline{x}_j)^2. \tag{4}$$

使用奇异值分解(SVD)法求解相关系数矩阵 $\boldsymbol{R}$ 的特征值($\lambda_1,\lambda_2,\cdots,\lambda_m$)和相应的特征向量 $\boldsymbol{\alpha}_j$,$\boldsymbol{\alpha}_j=(\alpha_{j,1},\alpha_{j,2},\cdots,\alpha_{j,m})$,$j=1,2,\cdots,m$.

3) 选择重要的主成分分量. 方差贡献率 c_j 为

$$c_j=\frac{\lambda_j}{\sum_{j=1}^{m}\lambda_j}. \tag{5}$$

式(5)中:λ_j 为第 j 个主成分分量的特征值.

累积贡献率 δ_j 为

$$\delta_j=c_1+c_2+\cdots+c_j. \tag{6}$$

主成分分量的筛选标准为 $\delta_j\geqslant 85\%$,可得主成分分量为

$$\left.\begin{aligned} Y_1&=\alpha_{1,1}x_1+\alpha_{1,2}x_2+\cdots+\alpha_{1,m}x_m, \\ Y_2&=\alpha_{2,1}x_1+\alpha_{2,2}x_2+\cdots+\alpha_{2,m}x_m, \\ &\vdots \\ Y_m&=\alpha_{m,1}x_1+\alpha_{m,2}x_2+\cdots+\alpha_{m,m}x_m. \end{aligned}\right\} \tag{7}$$

式(7)中:$x_1,x_2,\cdots,x_m$ 为标准化后的矩阵向量元素.

1.2 t-SNE 算法

t-SNE 算法是对称随机近邻嵌入(SNE)算法的改进[9-10],t-SNE 算法利用条件概率分布替换传统的距离表示,映射数据点在高维和低维空间之间的距离相似关系,在更好地维持初始数据结构的前提下,展示其内部的聚类特点. t-SNE 算法有以下 5 点计算思想.

1) 在高维空间中,高斯分布 $p_{v|u}$ 表示点 x_v,x_u 互为邻近点的概率. 当 x_v 与 x_u 之间的距离越近,$p_{v|u}$ 越大;当 x_v 与 x_u 之间的距离越远,$p_{v|u}$ 越小. $p_{v|u}$ 为

$$p_{v|u}=\frac{\exp\left(-\frac{\|x_u-x_v\|^2}{2\sigma_u^2}\right)}{\sum_{k\neq u}\exp\left(-\frac{\|x_u-x_k\|^2}{2\sigma_u^2}\right)}. \tag{8}$$

式(8)中：定义 $p_{u|u}=0$；σ_u 为高斯分布的方差；x_k 为高维数据.

2) 在对称 SNE 中，高维空间中的离群点 x_u 与其他数据点 x_v 的距离都很远，则 x_u 的联合概率分布 $p_{u,v}$ 均较小. $p_{u,v}$ 为

$$p_{u,v}=\frac{p_{v|u}+p_{u|v}}{2s}. \tag{9}$$

式(9)中：s 为数据点的总数.

3) 同理，在低维空间中，用 t 分布定义数据点之间的关系，则 x_u 的低维表示形式 y_u 的联合概率分布 $q_{u,v}$ 为

$$q_{u,v}=\frac{(1+\|y_u-y_v\|^2)^{-1}}{\sum_{k\neq l}(1+\|y_k-y_l\|^2)^{-1}}. \tag{10}$$

式(10)中：y_v，y_k，y_l 分别为数据点 x_v，x_k，x_l 的低维表示形式；t 分布的自由度设为 1.

4) 利用相对熵(KL)距离可得代价函数 C，有

$$C=\mathrm{KL}(P\|Q)=\sum_u\sum_v p_{u,v}\lg\frac{p_{u,v}}{q_{u,v}}. \tag{11}$$

式(11)中：P，Q 分别为高维空间和低维空间中所有数据点的联合概率分布.

5) 使用梯度下降法进行优化，有

$$\frac{\delta C}{\delta y_u}=4\sum_v(p_{u,v}-q_{u,v})(y_u-y_v)(1+\|y_u-y_v\|^2)^{-1}. \tag{12}$$

t-SNE 算法的具体执行步骤如下.

输入：s 个 D 维的向量 $\boldsymbol{\chi}=\{x_1,x_2,\cdots,x_s\}$ 映射到二维或三维空间，定值困惑度为 Prep，迭代次数为 T，学习率为 η，momentum 项系数为 $\beta(t)$

输出：低维数据 $y=\{y_1,y_2\cdots,y_s\}$

步骤：

步骤 1　点 x_u 的方差 σ_u 使用二分查找进行计算；

步骤 2　通过式(8)，(9)计算成对数据点的 $p_{v|u}$ 和 $p_{u,v}$；

步骤 3　初始化低维数据 $y_1,y_2,\cdots,y_s$；

步骤 4　通过式(10)计算低维数据的 $q_{u,v}$；

步骤 5　计算 $\frac{\delta C}{\delta y_u}$；

步骤 6　更新低维数据，$y^t=y^{t-1}+\eta\frac{\delta C}{\delta y_u}+\beta(t)(y^{t-1}-y^{t-2})$；

步骤 7　重复步骤 4～6，完成初始设置的迭代次数 T.

2　结合 PCA 的 t-SNE 算法及其并行化实现

随着数据体量和数据维数的增长，t-SNE 算法使用梯度下降法进行反复迭代，计算低维空间中数据点的分布情况，此时，产生的中间结果快速增多，内存压力逐渐变大，当内存不足时，只能将结果存储在磁盘中，这将大幅降低算法的效率.

由于 Spark 平台[11-13]是开源的通用分布式内存计算框架，通过驱动主节点程序实现任务的分发、调度执行和聚合结果，可解决内存压力过大导致的算法效率低下问题.

2.1　结合 PCA 法的数据并行化预处理

由于原始数据的维度较高，数据特征值过多，计算数据点之间成对距离的时间复杂度很高，导致算法的整体运行时间增加. 而主成分分析法由于轻量化，收敛速度快，能够快速地找到噪声点，在尽可能减少数据损失的情况下压缩和简化数据，节约内存，从而减少可视化结果受噪声点的干扰，去除冗余信息[14]. 因此，在数据预处理阶段使用 Spark 平台的 Mllib 机器学习库[15]中的分布式主成分分析法减少数据维度，既不会严重扭曲数据点之间的距离，又可以去除噪声数据.

首先,数据被分块存储于不同的分区上,对矩阵 $\boldsymbol{A}$(RowMatrix 类型)的格拉姆矩阵进行求解,矩阵中行和列的提取由 Map 回调函数执行,再发送给各执行节点,其结果由 Reduce 回调函数获得;然后,使用 SVD 法求解协方差矩阵 $\boldsymbol{W}$[16],再用特征值、特征向量生成主成分分量;最后,完成的数据再重新分发并保存到分布式文件系统中.

数据预处理流程图,如表 1 所示.

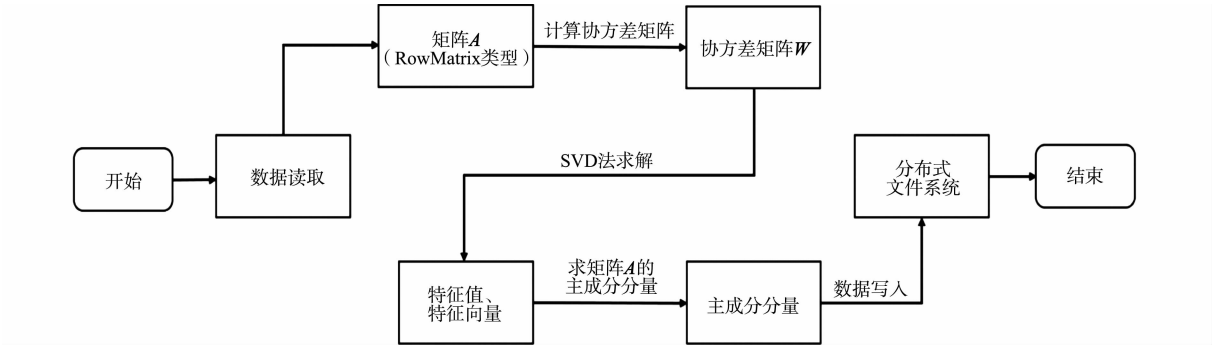


图 1 数据预处理流程图

Fig. 1 Data preprocessing flow chart

2.2 高维空间中点的相似性的 K-NN 图表示

K 最邻近(K-NN)图是基于 K-NN 算法构造的多节点关系图[17].对于空间中的 s 个节点,找出与任一节点 x_s 距离最小的 K 个邻居 $x_1,x_2,\cdots,x_K$,这里的距离度量方式可以自行设定,找到邻居节点后,将其与 x_s 进行连接.空间中其他节点同理,由此可得 K-NN 图.

在 t-SNE 算法中,高维空间中数据点间的相关关系用高斯分布进行表示(式(8)),对每一个数据点都包含 $\sum\limits_{K\in N_u}\exp\left(-\dfrac{\|x_u-x_K\|^2}{2\sigma_u^2}\right)$ 这一项,计算量会与数据规模成正比例上升.然而,在实际应用中,两数据点距离较大且互为邻居的概率 $p_{u,v}$ 几乎可以忽略.因此,描述任何高维数据点之间的相关性不需要精确到所有数据点,仅需使用近邻的一些数据点.文中使用 x_u 相邻的 $\|3U\|$ 个数据点,U 为 x_u 的周围条件概率分布的困惑度,近邻数据点的集合为 N_u,则有

$$p_{v|u}=\frac{\exp\left(-\dfrac{\|x_u-x_v\|^2}{2\sigma_u^2}\right)}{\sum\limits_{K\in N_u}\exp\left(-\dfrac{\|x_u-x_K\|^2}{2\sigma_u^2}\right)}, \tag{13}$$

由此大幅降低了计算量.文中构建 K-NN 图的算法是制高点(VP)树方法,时间复杂度为 $O(UN\lg N)$.

2.3 多节点并行执行 t-SNE 算法

实现 Spark 平台的连接和访问,任务控制节点 Driver Program 创建 SparkConf 和 Spark Context 对象,再对分布式文件系统(HDFS)上预处理完成的数据创建弹性分布式数据集(RDD),并分发至每个工作节点,读取分区中的数据集,并有序选择数据点 x_u,x_v 作为起始点,生成 RDD1,通过 Map 回调函数计算 $p_{u,v}$,生成 RDD2,触发 Cache 缓存算子,通过 Map 回调函数计算低维分布 $q_{u,v}$,生成 RDD3.进入 Combline 阶段,优化成本函数,更新 $q_{u,v}$,生成 RDD4,判断是否继续迭代.达到预先设定的迭代次数后,RDD4 启动 ReduceByKey 算子,将所有结果汇聚到同一分块,输出最终的低维空间中所有数据点的矩阵 $\boldsymbol{Z}$,完成降维目标.在这些执行步骤中,各工作节点的中间结果保存在内存中,完成后的数据集中到任务控制节点,触发 SaveAsTextFile 算子,并将最终结果写入 HDFS.

并行执行算法流程图,如图 2 所示.

在进行并行计算时[18],Spark 平台将 RDD 分发到不同的工作节点上,触发缓存机制可以在内存中实现 RDD 显式持久化,使中间数据重复使用,并将结果缓存到内存中;在计算低维空间中的数据分布时,存储在内存中的数据发挥作用,减少了读取时间,加快迭代过程.因此,对于需要反复迭代计算的算法,内存计算可有效地优化时间成本.在 Spark 平台中,各任务共同使用广播发送变量,变量在每个计算节点上运行和备份,减少了各数据在传输过程中的消耗,提升了算法的效率.

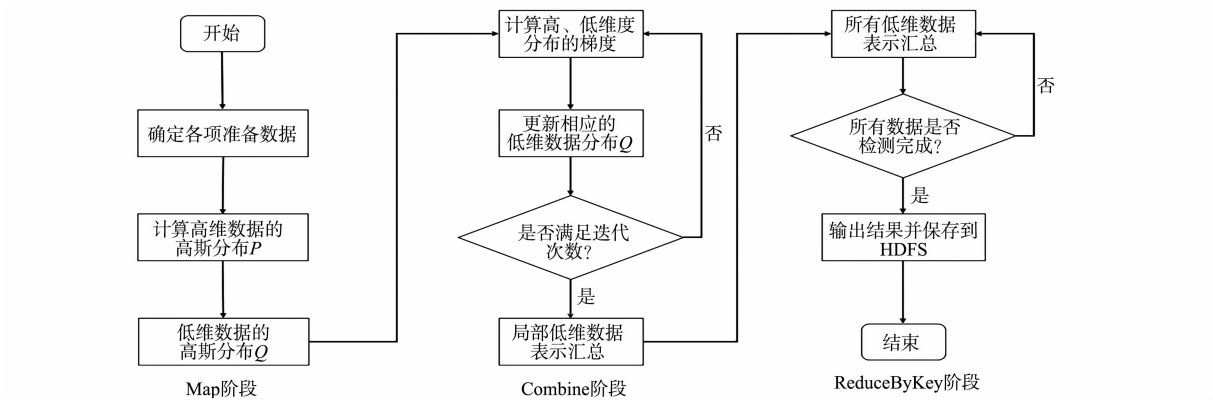


图 2　并行执行算法流程图

Fig. 2　Parallel execution algorithm flow chart

3　实验结果与分析

为测试文中算法的性能，从运行效率、加速比、扩展比、可视化效果和精确度 5 个方面进行分析.

3.1　实验环境

基于 Vmware 虚拟平台，搭建 Spark 平台集群环境. 创建 3 台虚拟机，其中，1 台虚拟机为主要节点，其他两台虚拟机为从节点. 每个节点 CPU 信息为 Intel®Core™ i7-9750，运行内存为 2 GB，硬盘读写速度为 1 TB·s^{-1}，集群操作系统为 Centos7，Hadoop 软件版本为 2.8.3，Spark 平台的版本为 2.3.0. t-SNE 算法的单机实验环境如下：CPU 信息为 Intel®Core™ i7-9750，运行内存为 16 GB，硬盘数据读写速度为 1 TB·s^{-1}.

实验采用的数据集为 BREAST CANCER，MNIST 和 CIFAR-10[19-20]. 根据数量级，将 MNIST 数据集分为训练集和测试集，其中，测试样本 10 000 个，训练样本 60 000 个，每个样本均为 1 个 784 维度的高维数据.

3.2　运行效率

将 MNIST 数据集分别运行于单机环境和 Spark 平台，通过处理时间(t_c)衡量文中算法的运行效率. 不同数量级的数据集在单机环境和 Spark 平台的运行效率对比，如图 3 所示. 图 3 中：w 为节点数.

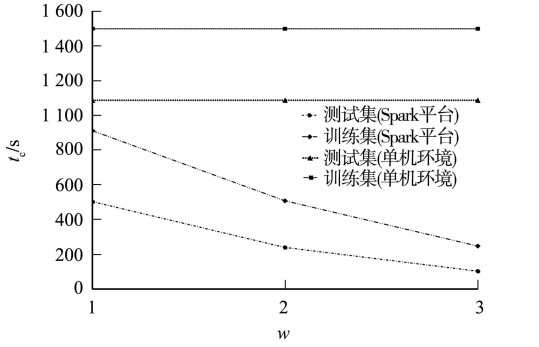


图 3　单机环境和 Spark 平台的运行效率对比

Fig. 3　Comparison of operational efficiency between stand-alone environment and spark platform

由图 3 可知：当使用同一数据集在集群中进行实验时，在 Spark 平台中单个节点的运行效率远高于单机下的运行效率；数据的处理时间随着集群中的节点数的增加而减少，表明算法的执行速度随着节点数的增加而提高，同时，大规模数据集的处理速度随集群中节点数的增加而提高.

3.3　加速比和扩展比

加速比(S)和扩展比(E)是衡量算法并行化的指标. 并行化性能的优劣由单个节点运行的时间与多个节点并行的时间的比值进行量化，并行化性能与加速比成正比. 加速比的计算公式为

$$S=\frac{t_1}{t_w},\qquad w=1,2,3,\cdots. \tag{14}$$

式(14)中：t_1 为算法单个节点运行的时间；t_w 为算法多个节点并行的时间.

扩展比是加速比和节点数的比值，其计算公式为

$$E=S/w. \tag{15}$$

文中算法在 MNIST 数据集的加速比和扩展比，如图 4，5 所示. 由图 4，5 可知：在 Spark 平台中，随

着参与计算节点的增多,加速比随之逐渐增大,扩展比随之逐渐减小;随着数据集的增大,加速比的增长随之变快,而扩展比随之趋于平稳,算法的并行化的优势也愈发明显.

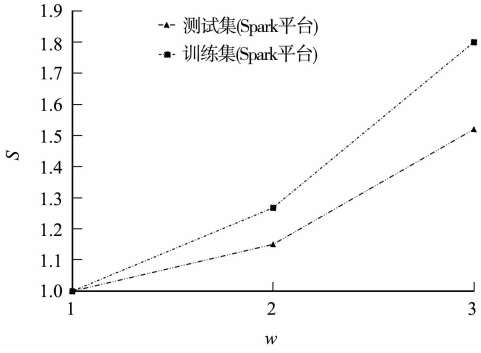


图4 文中算法在MNIST数据集的加速比

Fig. 4 Speedup ratio of proposed algorithm in MNIST data set

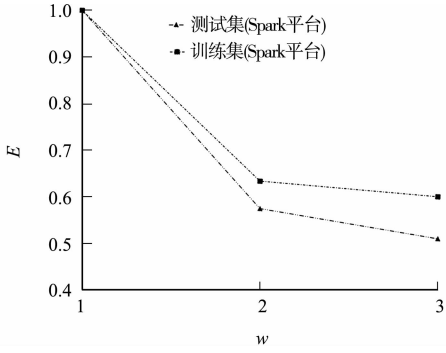


图5 文中算法在MNIST数据集的扩展比

Fig. 5 Expansion ratio of proposed algorithm in MNIST data set

3.4 可视化效果和精确度

对高维数据集进行降维处理,最终将其可视化于二维空间,点的颜色对应不同的对象类别.通过准确率、召回率和相对准确率衡量算法的精确度.

准确率 ξ_P 和召回率 ξ_R 的计算公式分别为

$$\xi_P=\frac{N_{g,h}}{N_g}, \tag{16}$$

$$\xi_R=\frac{N_{g,h}}{N_h}. \tag{17}$$

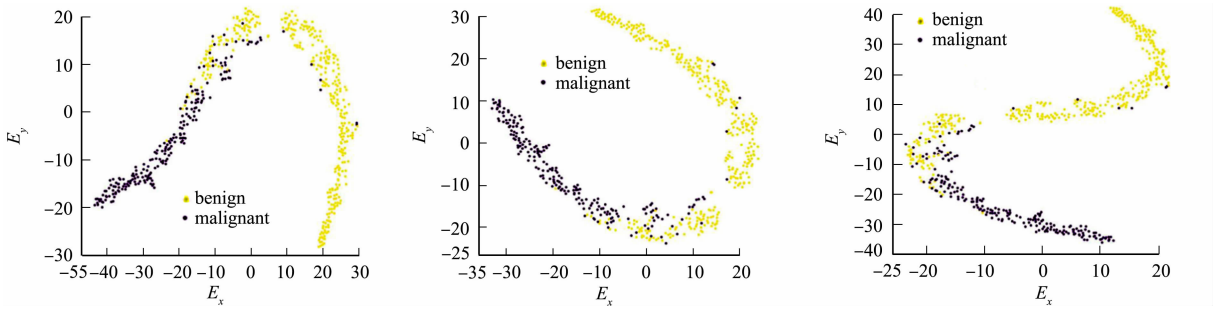
式(16),(17)中:$N_{g,h}$为属于g类,但被划归到h类中的数据数量;N_g 为g类中的全部数据数量;N_h 为h类中的全部数据数量.

由此可得精确度的评价指标相对准确率 ξ_F 为

$$\xi_F=\frac{2\xi_P\times\xi_R}{\xi_P+\xi_R}. \tag{18}$$

为了验证文中算法的可视化效果和精确度,采用 t-SNE 算法(单机环境),基于 Spark 平台的 t-SNE 算法(和文中算法环境相同)、文中算法在 BREAST CANCER,MNIST 和 CIFAR-10 数据集上进行实验,其可视化效果对比,如图6~8所示.图6~8中:E_x,E_y 分别表示原始数据的两个特征值.

由图6~8可知:原始 BREAST CANCER 数据集是30维的向量,有效映射到二维散点图被分为2类;原始 MNIST 数据集是784维的向量,有效映射到二维散点图被分为10类;原始 CIFAR-10 数据集是1 024维的向量,有效映射到二维散点图被分为10类.



(a) t-SNE 算法 (b) 基于 Spark 平台的 t-SNE 算法 (c) 文中算法

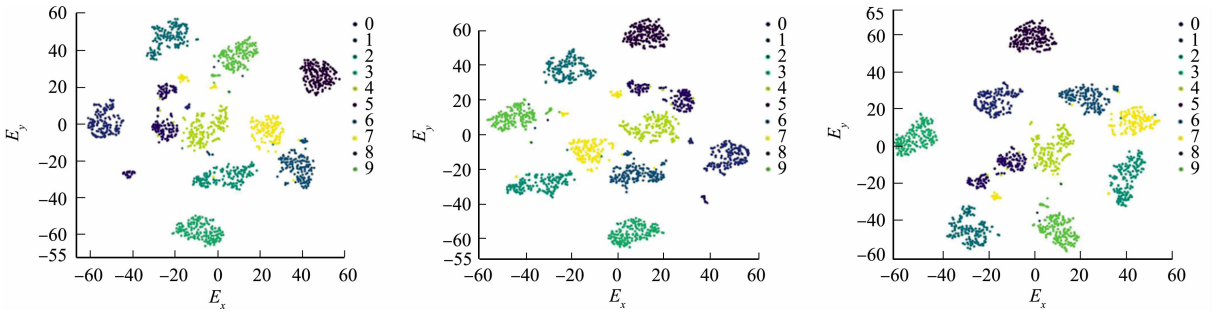
图6 3种算法在BREAST CANCER数据集上的可视化效果对比

Fig. 6 Comparison of visualization effects of three algorithms in BREAST CANCER data set

不同数据集的精确度对比,如表1所示.

表1 不同数据集的精确度对比

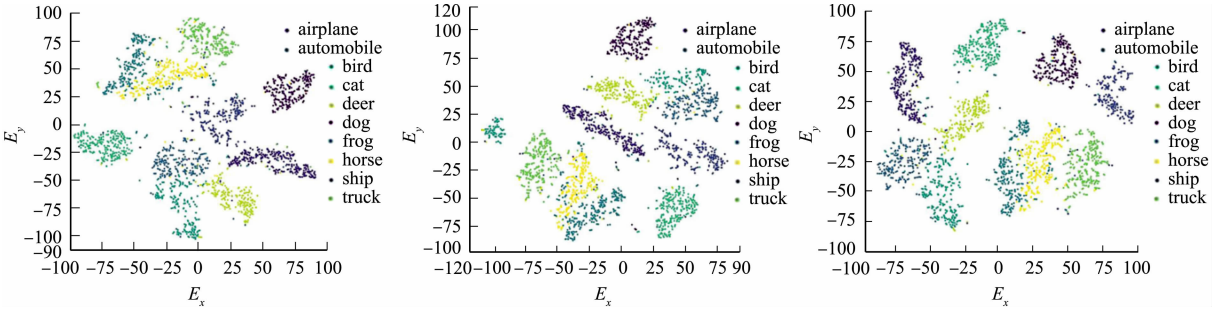
Tab. 1 Accuracy comparison of different data sets



（a）*t*-SNE 算法　　（b）基于 Spark 平台的 *t*-SNE 算法　　（c）文中算法

图 7　3 种算法在 MNIST 数据集上的可视化效果对比

Fig. 7　Comparison of visualization effects of three algorithms in MNIST data set



（a）*t*-SNE 算法　　（b）基于 Spark 平台的 *t*-SNE 算法　　（c）文中算法

图 8　3 种算法在 CIFAR-10 数据集上的可视化效果对比

Fig. 8　Comparison of visualization effects of three algorithms in CIFAR-10 data set

数据集	算法	ξ_P	ξ_R	ξ_F
BREAST CANCER	*t*-SNE 算法	0.875	0.809	0.840
	基于 Spark 平台的 *t*-SNE 算法	0.883	0.842	0.862
	文中算法	0.905	0.871	0.887
MNIST	*t*-SNE 算法	0.863	0.817	0.839
	基于 Spark 平台的 *t*-SNE 算法	0.878	0.801	0.837
	文中算法	0.922	0.867	0.893
CIFAR-10	*t*-SNE 算法	0.871	0.833	0.852
	基于 Spark 平台的 *t*-SNE 算法	0.868	0.839	0.853
	文中算法	0.915	0.878	0.896

由以上分析可知：文中算法在降维后的可视化效果、准确率、召回率和相对准确率均明显优于其他两种算法.

4　结束语

提出一种结合 PCA 的 *t*-SNE 算法的并行化方法. 在 MNIST 数据集中，对文中算法进行实验，验证了文中算法在大规模数据集中可以在提高运行效率和精确度的前提下，高效地完成降维可视化. 然而，降维会使数据被映射到低维空间时产生错误位置，导致其附近信息的丢失，原始高维空间中一些特征未能得到较好的保留. 此外，通过保留数据的周围信息，将数据从高维空间映射至低维空间，并未考虑全局数据之间的关系. 虽然文中算法能够在 Spark 平台下对大规模数据集进行处理，但由于文中算法是将低维数据作为变量进行迭代，一旦更新数据，需要重新启动算法，因此，在灵活性和开销方面仍有不足，今后将针对该问题开展相关研究.

参考文献：

[1]　PEZZOTTI N，THIJSSEN J，MORDVINTSEV A，*et al*. GPGPU linear complexity *t*-SNE optimization[J]. IEEE

Transactions on Visualization and Computer Graphics, 2020, 26(1): 1172-1181. DOI: 10.1109/TVCG.2019.2934307.

[2] 赵学武,吴宁,王军,等.航空大数据研究综述[J].计算机科学与探索,2021,15(6):999-1025. DOI:10.3778/j.issn.1673-9418.2012108.

[3] HEINRICH J, LUO Yuan, KIRKPATRICK A, *et al*. Evaluation of a bundling technique for parallel coordinates[J]. Energy Conversion and Management, 2011, 88(5): 259-266. DOI: 10.1016/j.enconman.2014.08.006.

[4] 途乐,陈彬捷,周志光.OD数据可视分析综述[J].计算机辅助设计与图形学报,2021,33(8):1160-1171. DOI:10.3724/SP.J.1089.2021.18679.

[5] 梁京章,黄星舒,吴丽娟,等.基于KPCA和改进K-means的电力负荷曲线聚类方法[J].华南理工大学学报(自然科学版),2020,48(6):143-150. DOI:10.12141/j.issn.1000-565X.200009.

[6] ROWEIS S, SAUL L. Nonlinear dimensionality reduction by locally linear embedding[J]. Science, 2000, 290(5500): 2323-2329. DOI: 10.1126/science.290.5500.2323.

[7] TENENBAUM J, SILVA V, LANGFORD J. A global geometric framework for nonlinear dimensionality reduction [J]. Science, 2000, 290(5500): 2319-2322. DOI: 10.1126/science.290.5500.2319.

[8] MAATEN L, HINTON G. Visualizingnon-metric similarities in multiple maps[J]. Machine Learing, 2012, 87(1): 33-55. DOI: 10.1007/s10994-011-5273-4.

[9] CHAN D M, RAO R, HUANG F, *et al*. GPU accelerated t-distributed stochastic neighbor embedding[J]. Journal of Parallel and Distribute Computing, 2019, 131(1): 1-13. DOI: 10.1016/j.jpdc.2019.04.008.

[10] 崔文泉,黄禹侨.高维数据情形下的一种基于随机投影的集成分类方法[J].中国科学技术大学学报,2019,49(12):974-984. DOI:10.3969/j.issn.0253-2778.2019.12.004.

[11] 刘东江,黎建辉.基于Spark的并行图聚类算法研究[J].系统仿真学报,2020,32(6):1038-1050. DOI:10.16182/j.issn1004731x.joss.18-0722.

[12] 张文杰,蒋烈辉.基于MapReduce并行化计算的大数据聚类算法[J].计算机应用研究,2020,37(1):53-56. DOI:10.19734/j.issn.1001-3695.2018.05.0496.

[13] 任磊,杜一,马帅,等.大数据可视分析综述[J].软件学报,2014,25(9):1909-1936. DOI:10.13328/j.cnki.jos.004645.

[14] 程宇航,张健钦,李江川,等.交通行业事故文本数据的可视化挖掘分析方法[J].计算机工程与应用,2021,57(21):116-122. DOI:10.3778/j.issn.1002-8331.2010-0269.

[15] 魏占辰,刘晓宇,黄秋兰,等.Spark迭代密集型应用的优化方法研究[J].计算机工程与应用,2020,56(23):68-73. DOI:10.3778/j.issn.1002-8331.1912-0293.

[16] LIU Shusen, MALJOVEC D, WANG Bei, *et al*. Visualizing high-dimensional data: Advances in the past decade [J]. IEEE Transactions on Visualization and Computer Graphics, 2017, 23(3): 1249-1268. DOI: 10.1109/tvcg.2016.2640960.

[17] BELKINA A C, CICCOLELLA C O, ANNO R, *et al*. Automated optimized parameters for *T*-distributed stochastic neighbor embedding improve visuallization and analysis of large datasets[J]. Nature Communications, 2019, 10(1): 1-12. DOI: 10.1038/s41467-019-13055-y.

[18] 崔艺馨,陈晓东.Spark框架优化的大规模谱聚类并行算法[J].计算机应用,2020,40(1):168-172. DOI:10.11772/j.issn.1001-9081.2019061061.

[19] 章蓉,陈谊,张梦录,等.高维数据聚类可视分析方法综述[J].图学学报,2020,41(1):44-56. DOI:10.11996/JG.j.2095-302X.2020010044.

[20] 董安国,张倩,刘洪超,等.基于TSNE和多尺度稀疏自编码的高光谱图像分类[J].计算机工程与应用,2019,55(21):177-182. DOI:10.3778/j.issn.1002-8331.1903-0155.

(**责任编辑**:钱筠 **英文审校**:吴逢铁)