软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2023,34(2):945−963 [doi: 10.13328/j.cnki.jos.006673]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种非完全的散点图去重叠算法*

赵 颖[1], 秀昱宏[1], 唐 涛[1], 文陈飞宇[1], 陈晓慧[2], 尤 旸[3], 周芳芳[1]

[1](中南大学 计算机学院, 湖南 长沙 410083)

[2](信息工程大学 数据与目标工程学院, 河南 郑州 450002)

[3](明略科技集团, 北京 100020)

通信作者: 周芳芳, E-mail: zff@csu.edu.cn

摘 要: 散点图中数据点重叠现象会严重影响可视分析效率. 现有散点图去重叠算法主要通过调整部分数据点的位置来完全去除重叠, 但普遍存在画布面积增长、轮廓保持不自然、迭代时间较长等问题. 认为完全去除重叠是非必须的, 通过实验发现: 用户能够在散点图有轻微重叠的情况下, 快速、准确地完成数据点选取和区域密度估计等可视分析任务. 因此, 提出了一个非完全的散点图去重叠算法, 该算法通过结合虚拟点临时占位、Voronoi 网格划分、数据点选择性移动和重叠率快速计算等方法, 实现分布紧凑、轮廓自然、高效迭代的散点图去重叠效果. 通过客观实验和主观实验评估了算法性能. 实验结果表明, 该算法在移动距离、面积增长、形状保持、正交顺序、邻域保持这 5 个客观指标和形状相似性、类簇稳定性这 2 个主观指标上都优于现有算法.

关键词: 可视化; 可视分析; 散点图; 高维数据; 降维投影; 去重叠

中图法分类号: TP391

中文引用格式: 赵颖, 秀昱宏, 唐涛, 文陈飞宇, 陈晓慧, 尤旸, 周芳芳. 一种非完全的散点图去重叠算法. 软件学报, 2023, 34(2): 945–963. http://www.jos.org.cn/1000-9825/6673.htm

英文引用格式: Zhao Y, Xiu YH, Tang T, Wen CFY, Chen XH, You Y, Zhou FF. Incomplete Overlapping Removal Algorithm for Scatterplots. Ruan Jian Xue Bao/Journal of Software, 2023, 34(2): 945–963 (in Chinese). http://www.jos.org.cn/1000-9825/6673.htm

Incomplete Overlapping Removal Algorithm for Scatterplots

ZHAO Ying[1], XIU Yu-Hong[1], TANG Tao[1], WEN Chen-Fei-Yu[1], CHEN Xiao-Hui[2], YOU Yang[3], ZHOU Fang-Fang[1]

[1](School of Computer Science and Engineering, Central South University, Changsha 410083, China)

[2](School of Data and Target Engineering, Information Engineering University, Zhengzhou 450002, China)

[3](Mininglamp Technology, Beijing 100020, China)

Abstract: Data point overlapping frequently occurs in scatterplots, resulting in visual clutters to interfere visual analysis. Some overlapping removal algorithms have been proposed to remove data point overlapping completely, however, they have some common shortcomings, mainly including the increasing of canvas size, distortion of data distribution, and dissatisfaction of time consumption. This work proposes that the complete removal of data point overlapping is non-essential, while slight overlapping is acceptable in some data analytical scenarios. Therefore, an incomplete overlapping removal algorithm is designed for scatterplots. First, the algorithm generates virtual data points in the blank areas in a scatterplot by using a semi-random generation method. Second, the algorithm uses a Voronoi diagram to divide each data point into an irregular grid, and then moves data points to grid centers to reduce the rate of data point overlapping and maintain the natural contour of data distribution. At last, the algorithm iteratively runs the step of Voronoi meshing and data point moving until that the rate of data point overlapping reaches a preset threshold. A series of objective and subjective experiments are conducted to evaluate the performance of the proposed algorithm and reference algorithms. The results show that users can quickly and accurately accomplish visual analysis tasks, including data point selection and regional density estimation, in scatterplots with a slight data point overlapping. The results reflect that the proposed algorithm is superior to all of the reference algorithms in the objective and

* 基金项目: 国家重点研发计划(2018YFB1700403); 国家自然科学基金(61872388, 62072470)
收稿时间: 2021-09-24; 修改时间: 2022-02-10; 采用时间: 2022-03-15

subjective indicators.

Key words: visualization; visual analytics; scatterplots; high-dimensional data; dimensionality reduction; overlapping removal

散点图是常见的可视化方法[1−3], 它将数据对象映射到二维直角坐标系, 直观地呈现数据对象在二维平面的分布特征, 例如聚类和离群点[4−6]. 在高维数据分析中, 人们经常使用 PCA、MDS 和 t-SNE 等降维方法将数据投影到散点图中[7,8], 以快速观察和分析高维数据. 图 1(a)展示了通过 t-SNE 降维方法将 Digits 数据集投影到散点图的可视化结果, 每个数据点表示一个手写数字图像, 数据点间的距离反映了图像间的相似程度.

散点图经常存在数据点重叠现象[4,9,10], 如图 1(a)中高亮框①–高亮框③所示. 在散点图中, 数据点通常被绘制成具有一定半径的圆点, 当任意两个数据点间的距离小于它们的半径之和时, 就会产生重叠. 随着数据点数量的增多, 重叠现象会愈加严重, 在一些数据可视分析场景中会干扰用户进行数据分析. 例如: 在聚类分析时, 数据点重叠会影响类簇内数据点密度的视觉估计[11−14]. 又如: 在交互时, 用户难以选中被遮挡的数据点并获取它们的属性信息[15]. 因此, 在散点图中减少或消除数据点重叠具有重要意义.

学者们已经提出了一些散点图去重叠方法[9,10], 可以分为图元编码法、数据转换法和空间移动法. 图元编码法通过调整数据点大小或透明度来减少重叠, 但当重叠数据点数量多、重叠程度高时收效甚微[16−18]. 数据转换法通过采样和超点抽象等方式来减少需绘制的数据点数量, 但这类方法会丢失细节信息[19−26]. 空间移动法是当前主流的散点图去重叠方法, 该类方法通过调整部分数据点在散点图中的位置来去除重叠, 常见的数据点移动策略包括正交移动、中心移动、螺旋移动和网格移动. 图 1(b)–图 1(e)是分别采用 VPSC 算法(正交移动)、PRISM 算法(中心移动)、RWORDLE 算法(螺旋移动)和 DGRID 算法(网格移动)[27−30]去重叠后获得的散点图.

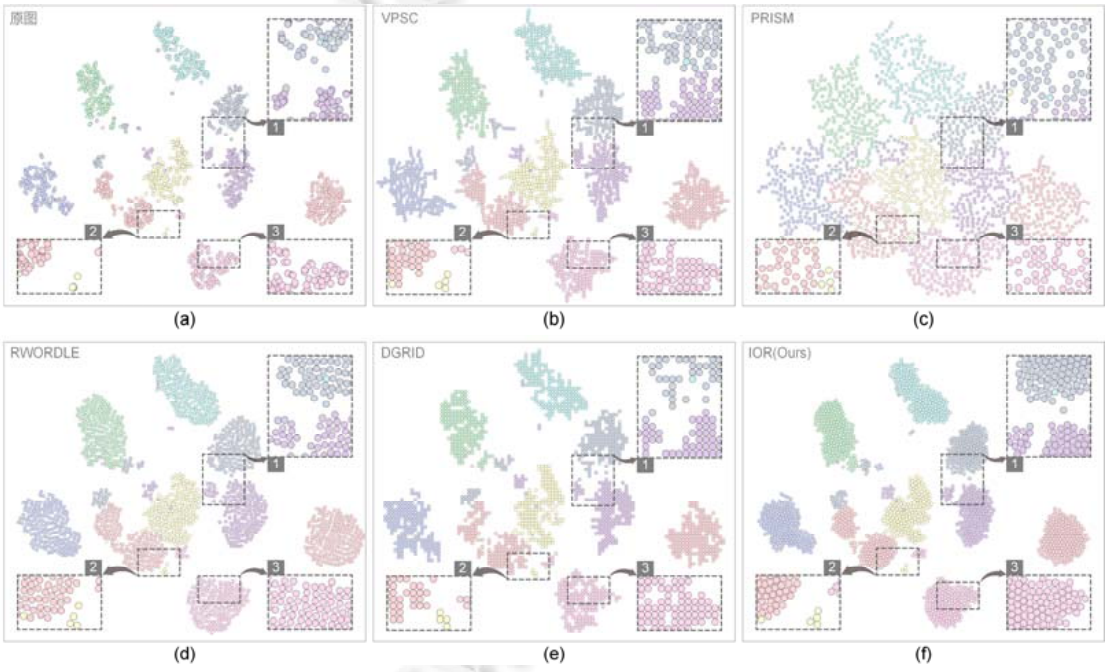


图 1 Digits 数据集使用 t-SNE 投影算法获得的散点图(a)和采用 VPSC 算法(b)、PRISM 算法(c)、RWORDLE 算法(d)、DGRID 算法(e)以及本文 IOR 算法(f)获得的去重叠散点图

现有的基于空间移动的散点图去重叠算法都遵从一个基本原则, 即完全去除重叠, 实现数据点的像素级平铺效果, 如图 1(b) –图 1(e)中高亮框①–高亮框③所示. 但该原则必然会使部分数据点产生较大的空间位移, 而且算法迭代次数不可控, 从而导致如下 3 个问题.

(1) 原屏幕空间难以容纳所有数据点, 需扩大画布或缩小数据点, 比如图 1(c)散点图中数据点半径比图 1(a)原散点图小 0.5%;

(2) 散点图的原数据分布被破坏, 如图 1(b)、图 1(c)中高亮框①处所示, 原本分离的类簇变得边界模糊; 如图 1(c)中高亮框②处所示, 散点图中原离群点(黄色圆点)被融入到附近类簇中; 如图 1(b) –图 1(d)中高亮框③处所示, 类簇轮廓或形状发生了改变;

(3) 算法效率不高, 需迭代多轮.

根据我们的观察和使用经验, 在一些分析场景中无需完全去除数据点重叠, 用户可以在有轻微重叠的散点图上完成可视分析任务. 例如: 在进行数据点选取时, 目标数据点与周围数据点只要能有视觉区分, 用户就可以选中数据点并进行后续交互分析. 又如: 在进行区域密度估计时, 只要不存在完全被覆盖的数据点, 用户就可以进行区域密度的视觉估计. 综上, 我们认为在散点图去重叠过程中, 不一定必须遵循完全去除重叠的原则, 可以适当允许轻微重叠的存在.

本文提出了一种非完全的散点图去重叠(incomplete overlapping removal for scatterplot, IOR)算法. 针对散点图面积增长问题, 我们采用网格移动策略, 将散点图画布进行网格划分, 用网格限定数据点移动边界, 防止数据点长距离移动, 以控制画布面积增长程度. 针对散点图数据分布保持问题, 首先, 我们采用 Voronoi 网格划分方法生成非规则网格, 从而自然地保持数据分布的总体轮廓; 然后, 我们提出了基于矩形格的半随机虚拟点生成方法, 用虚拟点对散点图空白区域进行临时填充占位, 防止 Voronoi 网格划分产生大网格, 消除去重叠后可能产生的类簇边界数据点离群现象和类簇内空洞效应, 保持数据分布的局部特征. 针对散点图去重叠效率问题, 我们综合使用虚拟点生成数量控制、数据点有选择性移动、重叠率快速计算方法和设定重叠率阈值, 多方面地减少算法迭代次数, 提高去重叠效率.

本文进行了 3 组实验来验证算法的有效性.

- 首先, 我们进行了用户评估实验, 邀请了 20 名参与者在不同重叠率的散点图上完成数据点选和区域密度估计任务. 实验结果表明: 当重叠率低于 0.5%时, 参与者完成任务的准确率达到 100%并保持不变, 初步验证了散点图存在轻微重叠不会影响可视分析任务, 同时也给出了 0.5%是一个经验上合适的重叠率阈值.
- 然后, 我们选取了移动距离、面积增长、形状保持、正交顺序、邻域保持和时间消耗这 6 个客观指标, 在多个数据集上对本文的 IOR 算法与 4 种参考算法进行了算法性能客观评估. 实验结果表明: 本文算法在 5 个散点图数据分布结构保持性能指标上都取得了最佳表现; 在时间消耗指标上, 本文算法排名第二.
- 最后, 我们再次邀请 20 名参与者进行主观评估实验, 他们通过视觉观察对 5 种算法的去重叠效果从形状相似性和类簇稳定性这两个主观指标进行打分, 本文算法在两个主观指标上都获得了最佳评分.

综上所述, 本文首次探讨了散点图是否需要完全去除数据点重叠问题. 本文初步验证了散点图存在轻微重叠不会影响数据点选取和聚类分析等可视分析任务, 并推荐了重叠率阈值. 本文提出了一种非完全的散点图去重叠算法, 该算法能够有效保持散点图数据分布, 产生紧凑、自然的去重叠效果.

1 相关工作

1.1 散点图去重叠算法分类

散点图中常存在的数据点重叠现象会干扰用户对数据分布、类簇、离群点的视觉感知与探索分析[4]. 现有的散点图去重叠方法大致可以分为 3 类: 图元编码法、数据转换法以及空间移动法. 图元编码法通过调整数据点的大小、形状以及透明度等视觉编码来减少重叠[16−18], 比如, Christian 等人总结了数据点视觉编码对重叠感知的影响[18]. 这类方法简单、易用, 但不适合重叠点多且重叠度高的场景. 数据转换法通过减少同时显示的数据点数量来去除重叠, 采样[19−22]和视觉抽象[23−26]是这类方法的代表, 比如随机采样[20]、密度偏差采样[21]、蓝噪声采样[22]等一系列散点图采样算法都可以减少重叠; 又如, Mayorga 等人将散点图中重叠程度高的

区域抽象为色块[23], Christian 等人将散点图中多个类簇抽象为半径不一的圆[26], 这样既减少了重叠, 又保持了原始散点图的总体分布. 但是数据转换法会破坏原始数据的完整性, 使得散点图无法呈现细节信息. 近年来, 研究者提出了一些在散点图画布上移动数据点的去重叠方法, 称为空间移动法. 本文提出的散点图去重叠算法属于空间移动法.

1.2 基于空间移动的散点图去重叠算法

空间移动法通过调整散点图中部分数据点的位置, 使重叠的数据点相互分离, 从而完全去除重叠. 该类方法既不减少数据点, 也不改变视觉编码, 能够有效保留数据点的总体分布和局部细节. 空间移动法是散点图去重叠方法的热门研究分支, 已经有一些典型算法, 这些算法的差别主要体现在数据点移动策略上, 大体上可以分为 4 种移动策略: 正交移动、中心移动、螺旋移动和网格移动.

正交移动策略是空间移动法的先驱, 它通过在水平和垂直方向上移动数据点来去除重叠. 代表性算法是 PFS[31,32]和 VPSC 算法[27], 两个算法的差异体现在移动距离控制上. PFS 算法使用推力模型, 而 VPSC 算法使用移动约束模型. 正交移动策略可以有效保持数据点的正交顺序, 但是只能在水平和垂直两个方向上移动数据点, 导致去重叠后的散点图容易在水平和垂直方向上呈狭长分布, 破坏了原始散点图的分布形态.

中心移动策略认为, 数据点的移动方向应该由数据点之间的相对位置决定. PRISM 和 GTree 是采用该策略的经典算法[28,33], 它们采用 Delaunay 三角剖分技术构造相邻数据点间的中心连线, 确定每个数据点的可移动方向. 该策略增加了数据点的可移动方向, 能够有效避免正交移动策略带来的散点图狭长分布问题, 并能较好地保持数据点之间的邻近关系. 但数据点移动仍需沿特定方向进行, 无法充分利用散点图中空白区域, 容易出现局部数据点松散分布现象, 导致散点图整体面积增加.

螺旋移动策略受到词云布局算法的启发, 完全解除了数据点在可移动方向上的限制, 对需要移动的数据点, 通过螺旋转动方式, 搜索最近邻的空白区域进行放置. 该策略可以有效利用数据点间的空白区域, 减少中心移动策略带来的局部松散现象. Mani-Wordle 算法[34]及其改进版本 RWordle-L 和 RWordle-C 算法[29]都是采用该策略的经典算法. 但这类算法不善于处理密集重叠区域中的数据点, 因为寻找空白区域耗时较长, 而且仍然可能造成散点图整体面积的增加.

正交移动、中心移动和螺旋移动策略都将数据点移动方向作为切入点, 不断优化去重叠后散点图的紧凑程度. 但是它们并没有严格界定数据点的移动边界, 难以避免去重叠后散点图整体面积增加的现象. 网格移动策略首先将散点图划分成若干规则矩形网格, 然后将数据点移动到矩形网格中心以去除重叠, 代表性算法是 GridFit[35]. 该策略可以有效防止散点图面积的增加并可一定程度地保持散点图的原始分布形态, 但该策略存在两个缺点: 一是在多个网格共同边界上的数据点的移动方向存在不确定性, 二是数据点的规则化排列导致散点图总体轮廓不自然. 最近, Gladys 等人提出了一种基于空间划分的散点图去重叠算法, 称为 DGRID[30]. 该算法在对散点图进行规则矩形划分时加入了一些虚拟数据点, 这些虚拟点会占据一定数量的空白网格, 从而有效减少原始数据点在移动方向上的不确定性, 但仍然无法解决规则网格划分带来的散点图总体轮廓不自然的问题.

此外, 也有学者将斥力模型用于去重叠. 早期的 PFS[31]和 PFSP[32]算法使用的推力模型可以看作斥力模型的简化版本. 最近, Philipp 等人[36]提出了使用 4 种斥力组合来去重叠, 但在连续斥力参数空间中寻找理想参数组合非常困难, 因此他们设计了一个交互界面 LayoutExOmizer, 让用户手动调整参数并预览去重叠效果. 该方法开拓了交互式去重叠的新思路.

本文提出的 IOR 算法采用了网格移动策略. 本文算法与已有算法的区别在于如下 3 个方面: 首先, 本文算法允许数据点间存在轻微重叠, 这样有利于提高算法效率以及保持散点图的原始结构特征, 且并不会对点选、密度视觉估计、聚类分析等可视分析任务产生明显的负面影响; 然后, 本文算法没有采用规则矩形划分, 而是采用了图形学、图学和地理学等领域常用的 Voronoi 网格划分技术, 这样有利于解决规则矩形划分带来的散点图总体轮廓不自然的问题; 最后, 我们改进了 DGRID 算法的虚拟点生成技术, 更好地控制了虚拟点的生成数量、生成位置和分布形态, 多方面提高了去重叠效果和效率.

2 IOR 算法设计

2.1 设计目标

空间移动法是当前主流的散点图去重叠方法, 已经衍生出了一些具体算法, 这些算法通过调整数据点位置来完全消除散点图中的数据点重叠现象, 实现数据点像素级平铺效果[9,10]. 但是, 已有算法也普遍存在画布面积增长、轮廓保持不自然、迭代时间较长等问题[30]. 本文认为: 完全消除重叠是非必须的, 去重叠过程中可以允许数据点间存在轻微程度的重叠, 这样有利于克服现有算法的缺点, 并且轻微重叠不会对点选、密度视觉估计、聚类分析等可视分析任务产生明显的负面影响. 因此, 本文的总体目标是设计一种非完全的散点图去重叠算法, 具体设计目标包括以下几点.

- T1: 去重叠过程中, 数据点移动距离要尽量短, 以防止数据分布松散和画布面积的增长;
- T2: 去重叠后, 数据点间可以存在轻微重叠, 但是每个数据点要能够被清晰辨识, 以支持点选等交互操作;
- T3: 去重叠后, 可有效保持原散点图的数据分布和自然轮廓, 以支持密度估计和聚类分析等可视分析任务;
- T4: 算法总体运行效率要高.

2.2 设计思路

为了实现上述目标, 我们认真调研了已有算法并进行了多轮先导实验. 根据先导实验结果, 我们对已有算法的优缺点进行了深入分析, 发现新算法设计必须认真思考以下 4 点.

- C1: 数据点移动策略的选择

移动数据点需遵循一定策略. 在现有的 4 种主要移动策略中, 正交移动策略、中心移动策略和螺旋移动策略容易造成数据分布松散、画布面积明显增长和迭代次数较多等问题[27−29]. 网格移动策略因为给定了数据点移动边界, 所以是目前提高去重叠效率、避免数据分布松散和画布面积增长最好的策略(T1)[30]. 因此, 我们将采用网格移动策略, 即先将散点图划分成若干网格, 然后将数据点移动到网格中心以去除重叠.

- C2: 散点图网格划分的选择

网格移动策略需将散点图划分为网格状. 现有的网格移动去重叠算法采用大小一致的规则矩形网格划分[29,34], 如图 2(b)所示, 但存在网格边界上数据点移动方向不确定问题和去重叠后总体轮廓不自然问题. 图形学、图学和地理学等领域常用四叉树网格[37]和 Voronoi 网格[38,39]进行空间划分. 如果用四叉树网格对散点图进行递归网格划分, 如图 2(c)所示, 则稀疏区域网格大, 稠密区域网格小, 可以解决规则矩形网格的两大问题, 但许多数据点会落在网格边缘, 造成数据点需移动较长距离才能到达网格中心. 如果采用 Voronoi 网格划分方法, 如图 2(d)所示, 对新算法设计有 3 个优势: 首先, 非规则网格能够保证轮廓自然(T3); 然后, 每个网格只容纳单个数据点, 能够消除数据点移动方向的不确定性(T4); 最后, 大部分数据点在网格划分后就位于网格中心区域, 只要短距离移动就可以到达网格中心(T1,T2). 因此, 我们决定采用 Voronoi 网格划分技术.

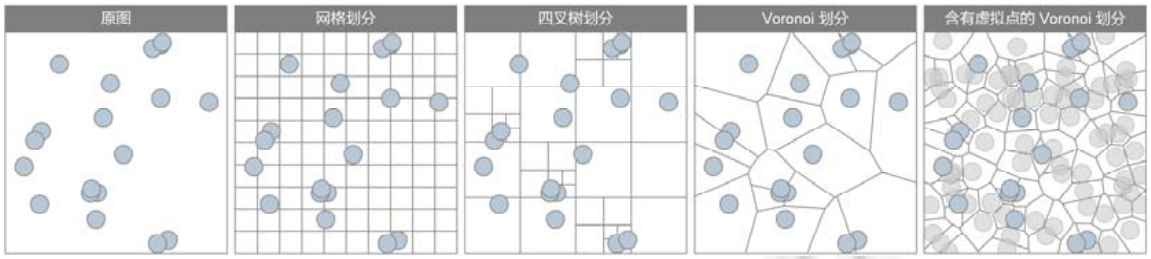
- C3: 大网格的优化

散点图中可能存在一些空白区域, 这些空白区域中会被划入面积较大的 Voronoi 网格, 如图 2(d)右下角所示. 类似大网格的产生, 是源于 Voronoi 网格划分的数据驱动模式. Voronoi 根据数据点的分布按照最邻近原则划分平面, 当区域内数据点分布较密集时, 生成的网格较小; 当区域内数据点分布较稀疏时, 生成的网格较大. 这种大网格会产生两方面的影响: 首先, 大网格中数据点移动距离较长; 其次, 类簇边界数据点容易被移动到远离类簇的空白区域. 受 DGRID 算法启发, 我们将生成一些临时占位的虚拟点来防止大网格的产生, 如图 2(e)所示. 但虚拟数据点生成数量、生成位置和分布形态都会对算法效率(T4)和散点图局部分布特征保持能力(T3)产生影响, 因此, 新算法设计需要重点考虑虚拟点生成方法的细节.

- C4: 算法效率的优化

现有散点图去重叠算法都需要迭代多轮才能完成去重叠任务, 一方面是因为每轮数据点的移动不能保证

完全消除当前重叠而且还可能产生新重叠, 另一方面是因为现有算法要求完全去除重叠. 本文允许存在轻微重叠的思路可以直接减少算法迭代次数, 但仍然需要认真设计数据点移动方法和重叠率计算方法, 并选择合理的重叠率阈值, 从多方面提高散点图去重叠效率(T4).



(a) 原散点图 (b) 规则网格划分 (c) 四叉树网格划分 (d) Voronoi 网格划分 (e) 增加虚拟数据点的 Voronoi 网格划分

图 2 多种散点图网格划分方法示意图

2.3 算法总体流程

本文提出的 IOR 散点图去重叠算法可分为以下 4 步, 算法流程如图 3 所示.

- STEP1. 生成虚拟点: 在散点图的空白区域生成一定数量的虚拟点. 此时, 散点图中同时存在着真实点和虚拟点这两类数据点;
- STEP2. Voronoi 划分: 对散点图中的所有数据点进行 Voronoi 划分, 尽量让每个网格中只存在单个数据点;
- STEP3. 调整数据点位置: 将所有虚拟点和重叠的真实点移动到所在网格的中心, 以减少重叠. 移动完成后, 计算当前真实点的重叠率: 如果重叠率未降至预设的阈值, 返回 STEP2; 否则, 进入 STEP4;
- STEP4. 删除虚拟点: 删除虚拟点并保留所有真实点, 形成去重叠后的散点图效果.

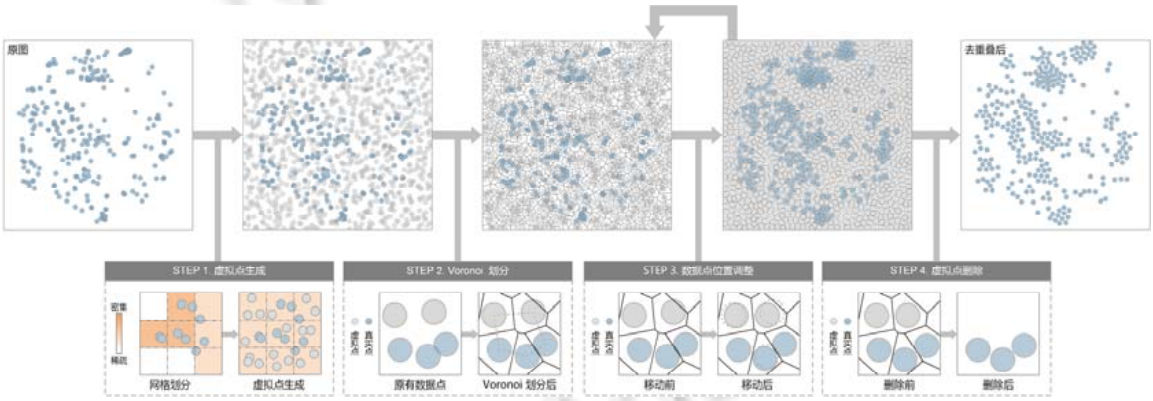


图 3 IOR 散点图去重叠算法流程图

2.4 虚拟点生成

虚拟点生成是本文算法的 STEP1, 通过在散点图空白区域生成一些临时的虚拟点, 防止后续 Voronoi 划分中可能出现的大网格现象(T1,C3). 我们尝试了一些虚拟点生成方法, 比如基于密度估计的虚拟点生成法和基于网格的虚拟点生成法, 但发现这些方法不能同时有效解决虚拟点带来的如下 3 个问题.

(1) 控制虚拟点的生成数量. 如果虚拟点数量太少, 将难以有效覆盖空白区域; 如果虚拟点数量太多, 将影响算法运行效率;

(2) 控制虚拟点的生成位置. 散点图中常存在一些位于真实点之间的小型空白区域, 这些区域一般只能容纳 1–2 个虚拟点. 这些小空白区域如果被虚拟点占位, 真实点将很难移动到这些区域, 从而在去

重叠后形成局部空洞效应. 例如: 在图 4 左图的区域①–区域③中都存在被真实点包围的小型空白区域, 其中, 区域①中的空白区域内有虚拟点, 区域②和区域③中的空白区域内没有虚拟点; 图4右图是去重叠后的效果, 由于区域①的真实点无法移动至被虚拟点占据的空白区域, 从而导致该空白区域在去除重叠后被保留, 甚至有时还会被放大, 形成局部空洞效应;

(3) 控制虚拟点的分布形态. Voronoi 网格是由 Delaunay 边的中垂线相交而成, 这其中存在一种特殊情况: 当Delaunay边恰好也是Voronoi边的中垂线时, 平面空间就进入了一种稳定状态. 此时, 数据点的当前位置和 Voronoi 划分后的网格中心完全相同, 导致数据点在迭代时无法移动. 虚拟点的规则排列会造成稳定状态, 从而严重限制了真实点的可移动方向, 影响了去重叠的质量和效率. 例如: 图 5 左图上半部分的虚拟点呈规则排列, 下半部分的虚拟点呈随机排列, 当移动重叠的真实点时, 真实点只能在下半部分寻找空隙, 如图 5 右图所示, 这不但造成上方大量空白区域被浪费, 而且影响了算法迭代效率.

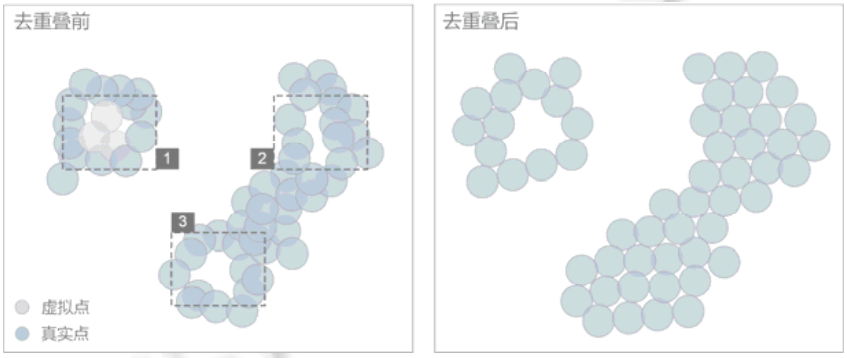


图 4 虚拟点生成位置不当导致散点图去重叠后产生局部空洞效应的示意图

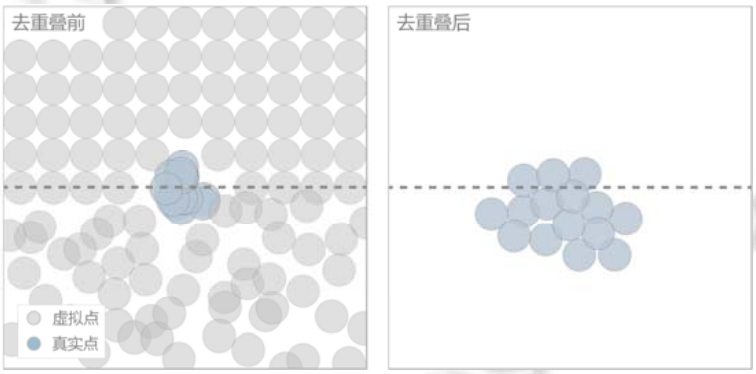


图 5 虚拟点规则排列影响重叠真实点可移动方向的示意图

为解决上述 3 个问题, 我们提出了一种基于矩形格的半随机虚拟点生成方法. 基本思想是: 将整个散点图划分成相同大小的矩形格, 通过矩形格数量确定虚拟点数量, 防止虚拟点数量不可控, 以提高算法的效率(T4,C4); 通过矩形格位置结合点对间空隙检测控制虚拟点可生成区域, 防止虚拟点生成在点间小空白区域造成局部空洞现象, 有利于保持原始散点图局部结构特征(T3,C3); 通过随机函数确定虚拟点在矩形格内的具体位置, 防止虚拟点规则分布, 提高散点图空白区域利用率(T3,C3). 具体做法如下.

首先, 我们将散点图划分为 m 个大小相同的正交矩形格, m 表示无重叠情况下, 该散点图中可容纳数据点的最大数量, m 的计算公式如下:

$$m=\mathrm{int}\left(\frac{width\cdot height}{grid\_area}\right) \tag{1}$$

$$grid\_area=4\cdot point\_radius^2 \tag{2}$$

其中, *width* 和 *height* 分别为散点图的宽和高, *grid_area* 表示矩形格面积, *point_radius* 表示数据点半径. 假设散点图面积为 100, 数据点半径为 1, 则矩形格数量 m 为 25. 我们规定每个矩形格最多只能容纳一个虚拟点, 于是可以预估出虚拟点的生成数量上限$(m-n)$, 其中, n 表示真实点数量. 这样一来, 既保证了有足够数量的虚拟点占据散点图中的空白区域, 又能有效防止虚拟点过量生成. 需要说明的是, 实际生成的虚拟点数量一般会略小于$(m-n)$, 因为一些矩形格并不适于生成虚拟点.

然后, 我们在正交矩形格划分的基础上设计了一种点对间空隙检测方法, 防止虚拟点生成在点间小空白区域. 首先, 设 $Grid_{ij}$ 为任意矩形格中数据点数量, i 和 j 为矩形格的行列索引值, 当 $Grid_{ij}=0$ 时, 矩形格为空白矩形格; 然后, 对于任意空白矩形格, 我们检测其在 4 个方向(水平、垂直、双对角线)上的两个相邻矩形格中是否同时存在数据点, 以判断空白矩形格是否位于点对间, 公式化表达如下:

$$\begin{cases} Grid_{(i-1)(j-1)}>0\wedge Grid_{(i+1)(j+1)}>0 \\ Grid_{(i+1)(j-1)}>0\wedge Grid_{(i-1)(j+1)}>0 \\ Grid_{(i)(j-1)}>0\wedge Grid_{(i)(j+1)}>0 \\ Grid_{(i-1)(j)}>0\wedge Grid_{(i+1)(j)}>0 \end{cases} \tag{3}$$

如果满足上述 4 个条件之一, 则判定 $Grid_{ij}$ 位于点对间, 并禁止虚拟点生成. 该方法能够有效避免虚拟点生成在点间小空白区域而造成的局部空洞现象.

最后, 在某空白矩形格中生成虚拟点时, 我们使用一个随机函数来确定虚拟点的具体位置(R_x,R_Y), 计算公式如下:

$$R_x=x+random(i\times w,(i+1)\times w) \tag{4}$$

$$R_y=y+random(j\times h,(j+1)\times h) \tag{5}$$

其中, i 和 j 表示矩形格的索引值, x 和 y 表示矩形格左上角坐标, w 和 h 表示矩形格的宽和高, *random* 表示随机函数. 该方法能够有效防止多个虚拟点形成规则排列, 从而为后续真实点移动创造了更多可移动方向和移动空间.

2.5 Voronoi划分和数据点位置调整

Voronoi 划分是本文算法的 STEP2. 该步骤同时针对真实点和虚拟点, 采用经典 Voronoi 划分技术[38,39], 将散点图划分为一系列的 Voronoi 多边形网格. 这些网格具有以下特点: (1) 每个网格对应一个数据点; (2) 所有网格大小相近. 经过划分后, 这些网格能够有效控制数据点的移动方向与移动距离, 从而使散点图在去重叠后能够尽量保持数据分布与自然轮廓(T3,C2). 需要注意的是: 在 Voronoi 划分之前, 我们要对完全重叠的点进行小幅度的随机移动, 使它们之间产生一定的距离, 防止完全重叠的数据点造成 Voronoi 划分失败.

数据点位置调整(STEP3)是去除重叠的核心步骤. 经 Voronoi 划分后, 需要通过移动数据点减少重叠. 该步骤利用分而治之的思想, 如图 6 所示, 对真实点与虚拟点采用不同的移动方法, 可在提高算法效率的同时, 尽量保持散点图的原始分布. 设 p_i 表示任意数据点, $Vor(i)$表示其所在的 Voronoi 网格, $Neighbors(i)$表示与其相邻的数据点集. 如果 p_i 为一个虚拟点, 我们直接移动到其所在网格 $Vor(i)$的中心位置. 因为虚拟点仅有填充占位的作用, 在去重叠完成后会被删除, 所以将其直接移动到网格中心不但能为真实点的移动创造更多空间, 还能减少迭代次数(T4,C4). 如果 p_i 为一个真实点, 我们采用“有重叠才移动”的策略, 即: 只有当该点与其邻居点集 $Neighbors(i)$中的任意真实点发生重叠, 才将该点移至其所在网格 $Vor(i)$的中心位置. 不移动非重叠真实点有两方面的好处: 一是保持原始数据分布(T3,C2), 二是提高算法运行速度(T4,C4).

在上述的数据点位置调整过程中, 我们需要检测重叠真实点并计算多边形中心坐标, 具体方法如下.

(1) 重叠真实点的检测方法包括两个步骤, 分别是确定邻居点集合和检测重叠. 我们在 Voronoi 划分过程中, 利用网格的邻近关系记录了每个真实点的邻居点集. 对于任意真实点, 我们只检测其是否与邻居点集中

的真实点发生重叠(T4,C4). 当任意两个数据点间的距离小于它们的半径之和时, 就会产生重叠现象, 表示为 $dist_{ij}<2\times point\_radius$, 其中, $dist_{ij}$ 表示数据点 p_i 和 p_j 的欧氏距离, $point\_radius$ 表示数据点的半径.

(2) 多边形中心坐标的计算方法: Voronoi 网格是一个不规则的多边形, 在二维直角坐标系中, 我们用 N 个顺时针的点$[(x_0,y_0),\ldots,(x_i,y_i),\ldots,(x_N,y_N)]$表示多边形, 其中, (x_N,y_N)与(x_0,y_0)的坐标相同. 对于多边形中心坐标的计算, 我们使用 Bourke 提出的方法[40]. 该方法将多边形视作多个三角形组合而成, 将每一个三角形中心坐标与其面积的乘积累加, 再将累加值与多边形面积相比, 比值即为多边形的中心坐标; 其中, 多边形面积等于所有三角形面积之和.

- 多边形中心坐标(C_x,C_y)的计算公式如下:

$$C_x=\frac{1}{6\times ploygon\_area}\sum_{i=0}^{N-1}(x_i+x_{i+1})(x_iy_{i+1}-x_{i+1}y_i) \tag{6}$$

$$C_y=\frac{1}{6\times ploygon\_area}\sum_{i=0}^{N-1}(y_i+y_{i+1})(x_iy_{i+1}-x_{i+1}y_i) \tag{7}$$

- 多边形面积 $ploygon\_area$ 的计算公式如下:

$$ploygon\_area=\frac{1}{2}\sum_{i=0}^{N-1}(x_iy_{i+1}-x_{i+1}y_i) \tag{8}$$

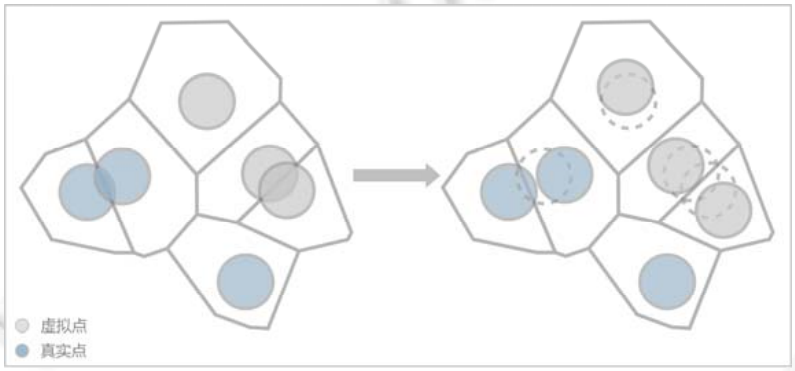


图 6 真实数据点和虚拟数据点移动策略示意图

2.6 重叠率定义与重叠率阈值设置

重叠率阈值是本文算法的关键参数. 我们将重叠率定义为散点图数据点间重叠面积与数据点总面积的比值, 将重叠率阈值定义为去重叠后允许的最高重叠率值. 在完成 STEP3 后, 需要重新计算重叠率并将其与重叠率阈值加以比较, 根据比较结果决定算法是否仍需要继续迭代运行. 如果重叠率低于预先设定的重叠率阈值, 则算法终止迭代并删除虚拟点生成最终布局(STEP4); 否则, 将返回(STEP2).

本文将重叠率γ的计算公式定义如下:

$$\gamma=\frac{overlap\_area}{total\_area} \tag{9}$$

其中, $total\_area$ 表示数据点无重叠时的总面积, 即单个数据点的面积与数据点的数量的乘积; $overlap\_area$ 表示数据点间重叠面积, 我们利用两个数据点距离与重叠面积的函数关系, 计算每对数据点间的重叠面积并进行累加求和, 最终得出所有数据点的重叠面积. 相较于直接计算数据点组成不规则图形面积, 计算数据点重叠面积的时间复杂度为 O(n), 更加简便、快捷, 有助于提高算法效率(T4,C4). 计算公式如下:

$$\begin{cases} total\_area=node\_num\times\pi R^2 \\ overlap\_area=\sum\limits_{i=1}^{node\_num}\sum\limits_{j=1}^{node\_num}\left[\arccos\left(\frac{dist_{ij}}{2\times R}\right)\times R^2-\sqrt{R^2-\frac{dist_{ij}^2}{4}}\times\frac{dist_{ij}}{2}\right] \end{cases} \tag{10}$$

其中, $node\_num$ 表示数据点数量, $dist_{ij}$ 表示数据点 p_i 和 p_j 的欧氏距离, R 表示数据点的半径, acos(·)表示反余弦.

重叠率阈值控制着算法迭代次数和去重叠效果, 重叠率阈值越低, 则算法迭代次数越多, 重叠去除程度也就越高. 若将该阈值设定为 0, 即表示完全去除重叠. 本文主张允许轻微重叠存在, 因此重叠率阈值一般不设定为 0. 为了获得一个合适的重叠率阈值, 我们进行了一个用户评估实验, 具体请见后文第 3.1 节. 实验结果表明: 当散点图重叠率低于 0.5%时, 用户从视觉层面就几乎无法感知到重叠了, 并且用户进行点选和视觉密度估计的准确率可达 100%. 因此, 本文推荐将重叠率阈值设为 0.5%.

3 实验结果与分析

在实验环节, 我们首先探究了重叠率对于用户完成散点图可视分析任务的影响; 然后, 我们通过客观指标评估和主观用户评估, 从多方面比较了本文 IOR 算法和相关算法的性能.

3.1 重叠率实验

重叠率实验是一个用户评估实验. 该实验有两个目的: 一是验证假设, 即散点图存在轻微重叠并不会影响可视分析任务; 二是确定一个合适的重叠率阈值, 它是 IOR 的唯一参数. 我们招募了 20 名参与者(10 名男性, 10 名女性, 年龄在 20–30 岁之间). 我们选择了 2 种常见的基于散点图的可视分析任务: 数据点选取和区域密度判断. 在数据点选取任务中, 散点图会随机高亮 1 个数据点, 要求参与者在规定时间内(5 s)准确点击该数据点, 如果点错或超时, 则任务失败. 在区域密度判断任务中, 散点图会随机显示 2 个矩形框, 参与者需要观察并比较这两个矩形框内的数据点密度, 并选出密度较大的矩形框. 我们选取了 5 个中等规模的数据集(1000–10000 个数据点)和 8 种重叠率, 用 t-SNE 投影算法和本文的 IOR 去重叠算法生成 40 个(5×8)散点图. 在实验中, 每次随机展示一个散点图用于完成上述两种任务, 每个参与者需要进行 40 次实验并完成 80 次(40×2)任务. 所有实验结束后, 我们计算每个重叠率下, 所有参与者在 5 个数据集上完成上述两种任务的准确率, 结果见表 1.

表 1 重叠率对于可视分析任务影响实验的结果(%)

重叠率	点选准确率	密度估计准确率
15	54.6	79
10	63.8	85.4
5	71.6	88.2
1	78.8	93
0.8	86.8	97.7
0.5	100	100
0.4	100	100
0.3	100	100

实验结果初步验证了我们的假设: 随着散点图重叠率的降低, 参与者完成数据点选取和区域密度估计的准确率整体呈现上升趋势; 当重叠率低于 0.5%时, 参与者准确率达到 100%并保持不变. 这说明, 在该重叠率下, 基于散点图的点选和区域密度估计分析任务不会受到数据点重叠的影响. 一些参与者反馈: 当重叠率低于 0.5%时, 视觉上已经难以察觉到重叠的存在. 因此, 我们认为, 0.5%是一个经验上合适的重叠率阈值.

3.2 客观评估实验

散点图去重叠算法的性能主要体现在结构保持能力和时间损耗上. 为了评价本文 IOR 算法的性能, 我们选择了具有代表性的评估指标、参考算法以及数据集, 进行了一组客观评估实验.

对于评估指标, 我们选择了 6 个常用评估指标[10], 分别是:

- ED (Euclidean distance): 移动距离用于衡量去重叠前后所有数据点的移动距离. 该指标越接近 0, 表明移动距离越短, 越容易保持散点图结构;
- SI (size increase): 面积增长用于衡量去重叠前后散点图所占空间面积的变化情况. 该指标越接近 1, 表明散点图面积变化越小;
- SP (shape preservation): 形状保持用于衡量去重叠前后散点图外围轮廓的变化情况. 该指标越接近 0,

表明散点图整体形状保持得越好;

- OO (orthogonal ordering): 正交顺序用于衡量去重叠前后所有数据点相对位置的变化. 该指标越接近0, 表面去重叠后散点图的局部正交结构变化越小;
- NP (neighborhood preservation): 邻域保持用于衡量去重叠前后所有数据点的邻域保持情况. 该指标越接近 1, 表明去重叠后散点图的局部邻域结构保得越好;
- TC (time consumption): 时间消耗体现算法效率. 耗时越少, 效率越高.

对于参考算法, 为了能够覆盖 4 类空间移动策略, 我们从每类策略中选取了一个代表性算法, 分别是VPSC(正交移动)、PRISM(中心移动)、RWORDLE(螺旋移动)和 DGRID(网格移动)算法. 这些参考算法无需参数设置, 本文算法的唯一参数重叠率阈值被设置为 0.5%.

对于数据集, 我们选取了 5 个著名的高维数据集, 见表 2. 这些数据集有如下特点: (1) 数据规模覆盖了1 000 至 10 000 个数据点; (2) 类簇形态包括高斯聚类和流形聚类, 类簇数量在 3–31 个之间; (3) 通过 t-SNE算法投影后形成的散点图的重叠率从 15%–30%不等. 这些特点有利于全面验证算法的性能. 另外, 为了让散点图原始画布尺寸一致, 我们为不同规模的数据集设置了不同的散点图半径.

表 2 实验数据集基本信息

数据集名	数据规模(点数)	数据点半径(像素)	重叠率(%)	画布尺寸(像素)	数据类型
Digits	1 797	5	22.00	1080×1080	人工合成
D31	3 100	5	19.80	1080×1080	真实数据
Abalone	4 177	4	32.46	1080×1080	真实数据
Wine quality	4 898	4	15.28	1080×1080	真实数据
AI4I2020	10 000	3	29.38	1080×1080	真实数据

在实验中, 我们首先将每个算法在每个数据集上运行 20 次, 每次运行计算 6 个指标值; 然后, 我们计算每个指标的20次运行的平均值, 实验结果见表3(表中黑色加粗数值表示在某个数据集和某个指标条件下最好的实验结果). 所有实验运行在同一台电脑上, 其主要软硬件配置是: Interl Core i5-11600K CPU@ 3.9 GHz, 32 GB RAM, Windows10 64 bits, 分辨率为 2560×1440.

表 3 IOR 与 4 个参考算法的结构保持性能实验结果

数据集	算法	评估指标				
		移动距离(ED)	面积增长(SI)	形状保持(SP)	正交顺序(OO)	邻域保持(NP)
Digits	VPSC	13.093	1.113	0.003 3	0.027	0.793
	PRISM	60.603	1.571	0.078 5	0.079	0.76
	RWORDLE	22.393	1.097	0.005 7	0.037	0.736
	DGRID	17.550	1.128	0.013 4	0.019	0.798
	IOR(Ours)	**6.462**	**1.047**	**0.001 3**	**0.016**	**0.849**
D31	VPSC	13.317	1.041	0.009 8	0.024	0.763
	PRISM	81.825	1.702	0.081 0	0.068	0.688
	RWORDLE	18.725	1.004	0.000 4	0.028	0.723
	DGRID	15.777	1.076	0.004 0	0.016	0.778
	IOR(Ours)	**5.867**	**1.004**	**0.000 2**	**0.012**	**0.847**
Abalone	VPSC	29.427	1.302	0.164 0	0.057	0.637
	PRISM	119.951	2.057	0.230 6	0.121	0.632
	RWORDLE	29.427	1.102	0.007 6	0.038	0.614
	DGRID	21.776	1.158	0.018 4	0.03	0.733
	IOR(Ours)	**7.128**	**1.051**	**0.001 0**	**0.016**	**0.775**
Wine_Quality	VPSC	8.212	1.066	0.004 2	0.057	0.764
	PRISM	109.060	2.128	0.187 5	0.121	0.648
	RWORDLE	10.558	1.046	0.002 3	0.038	0.753
	DGRID	25.553	1.151	0.052 5	0.030	0.771
	IOR(Ours)	**4.804**	**1.022**	**0.000 7**	**0.016**	**0.844**
AI4I2020	VPSC	12.040	1.127	0.013 9	0.023	0.663
	PRISM	195.797	2.958	0.339 2	0.069	0.579
	RWORDLE	12.448	1.044	0.001 1	0.018	0.685
	DGRID	21.235	1.149	0.013 4	0.017	0.759
	IOR(Ours)	**4.156**	**1.027**	**0.000 8**	**0.009**	**0.810**

下面我们将 6 个指标分两组进行算法性能分析.

(1) 结构保持性能分析

表 3 给出了 5 个算法在 5 个数据集和 5 个结构保持性能指标上的实验结果. 本文的 IOR 算法在 5 个结构保持性能指标上都取得了最好结果. 在移动距离、形状保持和面积增长指标上, 本文算法大幅度优于其他 4 个参考算法. 主要原因是: 我们将虚拟点生成方法与 Voronoi 网格划分方法有机地结合在一起, 保证了数据点在移动前尽量位于多边形网格的中心区域, 大幅度减少了数据点的移动距离, 有效控制了散点图整体面积的增加幅度, 并且较好地保持了散点图的自然轮廓. 如图 7(f)所示: 本文算法获得的散点图中类簇紧凑且形状自然, 类簇间边界清晰, 整个散点图的轮廓最接近原始散点图. 在正交顺序和邻域保持指标上, 本文算法略优于其他算法, Voronoi 网格划分遵从了数据点的邻近关系, 因此数据点移动后仍然能够较好地保证数据点间的正交顺序和邻近关系.

对于其他 4 个参考算法, VPSC 算法在移动距离指标上的表现不错, 该算法在水平和垂直方向上设置了最小移动距离约束, 能够较好地控制移动距离, 但是由于只有两个可移动方向, 导致算法在形状保持指标上表现不佳, 还可能使原本明显分离的类簇连接起来, 如图 7(b)所示. PRISM 算法在 5 个结构保持性能指标上均表现不佳, 如图 7(c)所示, 去重叠后的数据点近似均匀分布在散点图中, 完全失去了原始结构特征, 主要原因是算法中移动约束条件过多, 导致数据点需要通过较多次的位置调整才能找到优化解. RWORDLE 算法在形状保持和面积增长指标上的表现良好, 该算法的螺旋转动方式能够避免数据点过度移动, 还能保持类簇形状, 但螺旋转动也会破坏数据点间的相对位置, 造成该算法在正交顺序与邻域保持指标上表现不佳, 如图 7(d)所示. DGRID 算法在正交顺序和邻域保持指标上表现较好, 该算法采用正交矩形网格划分, 有效保持了数据点间正交顺序和邻近关系, 但数据点呈现规则的正交分布形态, 导致类簇形状不自然; 另外, 该算法没有解决局部空洞效应, 在许多类簇内部存在不自然的局部性空洞, 如图 7(e)所示.

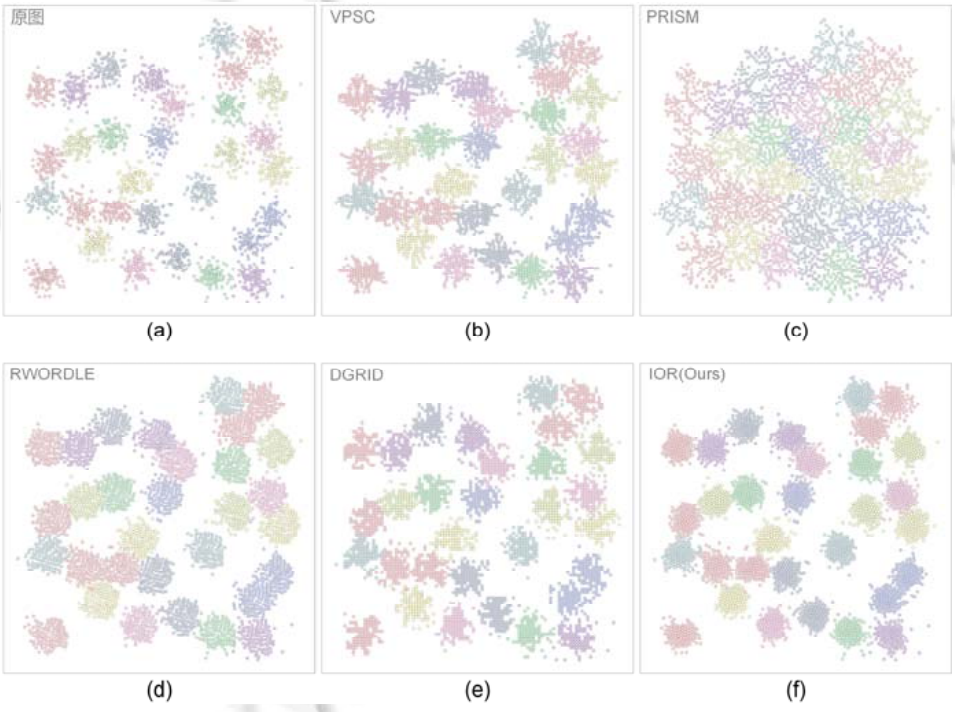


图 7 D31 数据集使用 t-SNE 投影算法获得的散点图(a)和采用 VPSC 算法(b)、PRISM 算法(c)、RWORDLE 算法(d)、DGRID 算法(e)以及本文 IOR 算法(f)获得的去重叠散点图

(2) 时间消耗分析

表 4 的结果表明: 本文 IOR 算法的时间性能总体较好, 在 5 个算法中排名第 2, 仅次于 DGRID 算法. IOR 算法有两个耗时步骤, 分别是时间复杂度为 O($n$log(n))的 Voronoi 划分和时间复杂度为 O(n)的重叠率计算, 其中, n 表示数据点的数量. 这两个耗时步骤时间复杂度不高, 另外, 通过调节重叠率阈值还能进一步减少算法迭代次数, 提高整体运算效率.

表 4 IOR 与 4 个参考算法在不同数据集上的时间消耗(单位: s)

算法	Digits	D31	Abalone	Wine_Quality	AI4I2020
VPSC	0.192	0.431	1.164	1.151	5.441
PRISM	25.434	157.712	237.7	683.832	2 531.733
RWORDLE	6.652	13.498	191.542	29.352	385.633
DGRID	**0.084**	**0.086**	**0.19**	**0.177**	**0.422**
IOR (ours)	0.45	0.472	1.248	0.447	0.996

3.3 主观评估实验

我们再次招募了第 3.1 节重叠率阈值实验中的 20 名参与者, 让他们通过视觉感知的方式对不同算法的去重叠效果进行主观评估. 我们选择了两个主观评估指标: 形状相似性和类簇稳定性. 形状相似性用于评估散点图整体结构的保持程度, 类簇稳定性用于评估去重叠前后类簇总体结构和类簇边界特征的稳定性. 选取了与第 3.2 节实验相同的 5 个数据集和 5 个去重叠算法, 每次向参与者呈现用同一个数据集生成的随机排列的 6 个散点图, 包括初始散点图和 5 个算法去重叠后的散点图, 参与者需要对每个散点图分 2 项指标进行打分, 分数采用里克特五级量表, 其中, 1 分代表最差, 5 分代表最好.

图 8 用堆叠条形图展现了参与者的打分情况, 我们还计算了每个算法的平均评分结果. 在形状相似性指标上, IOR (μ=4.45)和 DGRID (μ=3.96)表现较好; 在类簇稳定性指标上, IOR (μ=4.43)和 DGRID (μ=3.95)表现较好. 这些结果表明, 本文算法在 2 个主观指标上明显优于 4 个参考算法. 大多数参与者反馈: 他们一般通过类簇的紧凑性来判断类簇的形状与结构的保持程度, IOR 算法的类簇最为紧凑且类簇形状非常自然.

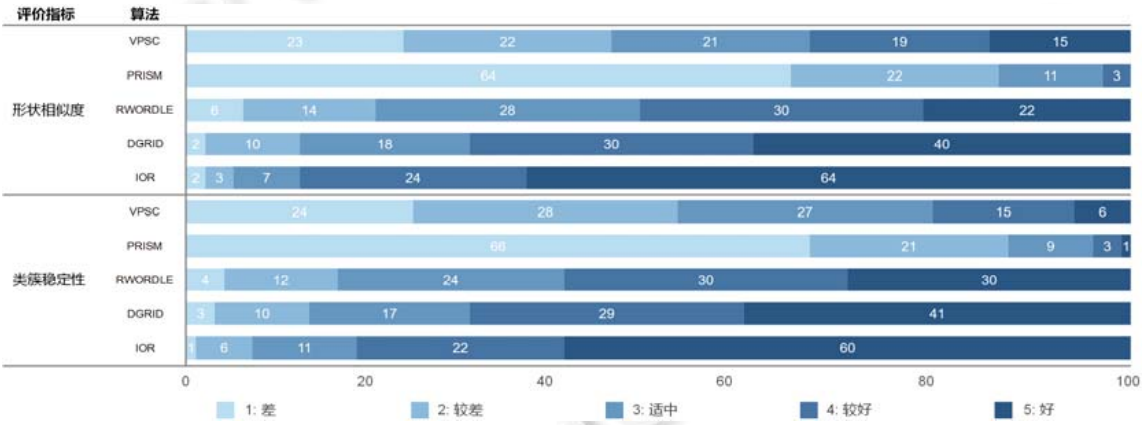


图 8 20 名参与者对 5 种算法的散点图去重叠效果的主观评分

4 总结与展望

本文首次探讨了散点图是否需要完全去除数据点重叠问题. 本文初步验证了散点图存在轻微重叠不会影响数据点选取和聚类分析等可视分析任务, 并推荐了具体的重叠率阈值. 本文提出了一种非完全的散点图去重叠算法, 该算法综合采用基于矩形格的半随机虚拟点生成方法、基于 Voronoi 的散点图网格划分与数据点选择性移动方法、重叠率快速计算方法和重叠率阈值设置, 实现了快速且优质的散点图非完全去重叠. 多组客观和主观实验结果表明, 本文算法在多项指标上都优于已有算法.

本文算法还有一些有待改进之处.

(1) 可视分析任务的局限性. 本文实验环节数据点选取和区域密度估计两个可视分析任务, 所以并不能确定本文算法是否适用于其他可视分析任务;

(2) 视觉编码的局限性. 本文散点图中数据点都为半径较小的圆点, 因此并不能确定本文算法是否适合数据点为大圆点、矩形甚至图片的场景;

(3) 数据集的局限性. 本文实验数据集没有超过 1 万个点规模, 当遇到更大规模数据集时, 扩大画布或减小数据点尺寸将不可避免, 本文没有探讨这方面的细节.

针对以上不足, 我们将进行下一步的研究: 首先, 对更广泛的可视分析任务进行实验, 确定非完全去重叠适用范围, 寻找在不同任务场景下适用的重叠率阈值; 此外, 需要对输入数据集进行预先分析, 根据数据集的规模以及数据点大小, 自适应地扩展画布大小; 最后, 提高算法对散点图可视编码的适用性, 处理数据点为非圆形状带来的去重叠细节问题.

References:

[1] Sarikaya A, Gleicher M. Scatterplots: Tasks, data, and designs. IEEE Trans. on Visualization and Computer Graphics, 2018, 24(1): 402−412.

[2] Wei YT, Mei HH, Zhao Y, Zhou SY, Lin BR, Jiang HJ, Chen W. Evaluating perceptual bias during geometric scaling of scatterplots. IEEE Trans. on Visualization and Computer Graphics, 2020, 26(1): 100−111.

[3] Zhao Y, Luo XB, Lin XR, Wang HR, Kui XY, Zhou FF, Wang JS, Chen Y, Chen W. Visual analytics for electromagnetic situation awareness in radio monitoring and management. IEEE Trans. on Visualization and Computer Graphics, 2020, 26(1): 590−600.

[4] Yuan J, Xiang SX, Xia JZ, Yu LY, Liu SX. Evaluation of sampling methods for scatterplots. IEEE Trans. on Visualization and Computer Graphics, 2021, 27(2): 1720−1730.

[5] Zhao Y, Luo F, Chen MH, Wang YC, Xia JZ, Zhou FF, Wang YH, Chen Y, Chen W. Evaluating multi-dimensional visualizations for understanding fuzzy cluster. IEEE Trans. on Visualization and Computer Graphics, 2019, 25(1): 12−21.

[6] Zhou ZG, Meng LH, Tang C, Zhao Y, Guo ZY, Hu MX, Chen W. Visual abstraction of large scale geospatial origin-destination movement data. IEEE Trans. on Visualization and Computer Graphics, 2018, 25(1): 43−53.

[7] Ingram S, Munzner T, Irvine V, Tory M, Bergner S, Moller T. DimStiller: Workflows for dimensional analysis and reduction. In: Proc. of the IEEE Symp. on Visual Analytics Science and Technology. 2010. 3−10.

[8] van der Maaten L, Hinton G. Visualizing data using *t*-SNE. Journal of Machine Learning Research, 2008, 2579−2605.

[9] Marcı´lio-Jr WE, Eler DM, Garcia RE, Pola IRV. Evaluation of approaches proposed to avoid overlap of markers in visualizations based on multidimensional projection techniques. Information Visualization, 2019, 18(4): 426−438.

[10] Chen FT, Piccinini L, Poncelet P, Sallaberry A. Node overlap removal algorithms: An extended comparative study. Journal of Graph Algorithms and Applications, 2020, 24(4): 683−706.

[11] Zhou FF, Li JC, Huang W, Wang JW, Zhao Y. Extending dimensions in Radviz for visual clustering analysis. Ruan Jian Xue Bao/Journal of Software, 2016, 27(5): 1127−1139 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4951.htm [doi: 10.13328j.cnki.jos.004951]

[12] Tang L, Li XQ, Liu Y. Dimensional density and clustering in scatterplots. Ruan Jian Xue Bao/Journal of Software, 2010, 21: 194−204 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/10021.htm

[13] Zhou ZG, Tang C, Liu YH, Liu YN, Xu JL. Visual analytics for multidimensional time-varying data via dimension reduced visual perception. Journal of Computer-aided Design & Computer Graphics, 2018, 30(7): 1194−1204 (in Chinese with English abstract).

[14] Ding SF, Xu X, Wang YR. Optimized density peaks clustering algorithm based on dissimilarity measure. Ruan Jian Xue Bao/Journal of Software, 2020, 31(11): 3321−3333 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5813.htm [doi: 10.13328/j.cnki.jos.005813]

[15] Yuan Z, Wen JR, Wei ZW, Liu JJ, Yao B, Zheng K. Real-time interactive analysis on big data. Ruan Jian Xue Bao/Journal of Software, 2020, 31(1): 162−182 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5886.htm [doi: 10.13328/j.cnki.jos.005886]

[16] Li J, van Wijk JJ, Martens JB. A model of symbol lightness discrimination in sparse scatterplots. In: Proc. of the IEEE Pacific Visualisation Symp. 2010. 105−112.

[17] Li J, Martens JB, van Wijk JJ. A model of symbol size discrimination in scatterplots. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. 2010. 2553−2562.

[18] van Onzenoodt C, Huckauf A, Ropinski T. On the perceptual influence of shape overlap on data-comparison using scatterplots. Computers & Graphics, 2020, 90: 169−181.

[19] Ellis G, Bertini E, Dix A. The sampling lens: Making sense of saturated visualisations. In: Proc. of the Extended Abstracts on Human Factors in Computing Systems. 2005. 1351−1354.

[20] dos Santos Amorim EP, Brazil EV, Daniels J, Joia P, Nonato LG, Sousa MC. iLAMP: Exploring high-dimensional spacing through backward multidimensional projection. In: Proc. of the IEEE Conf. on Visual Analytics Science and Technology. 2012. 53−62.

[21] Xiang S, Ye X, Xia J, Wu J, Chen Y, Liu S. Interactive correction of mislabeled training data. In: Proc. of the IEEE Conf. on Visual Analytics Science and Technology. 2019. 57−68.

[22] Chen H, Chen W, Mei H, Liu Z, Zhou K, Chen W, Gu W, Ma K. Visual abstraction and exploration of multi-class scatterplots. IEEE Trans. on Visualization and Computer Graphics, 2014, 20(12): 1683−1692.

[23] Mayorga A, Gleicher M. Splatterplots: Overcoming overdraw in scatter plots. IEEE Trans. on Visualization and Computer Graphics, 2013, 19(9): 1526−1538.

[24] Li CH, Baciu G, Han Y. StreamMap: Smooth dynamic visualization of high-density streaming points. IEEE Trans. on Visualization and Computer Graphics, 2018, 24(3): 1381−1393.

[25] Zhao JH, Liu X, Guo C, Cheryl Qian ZY, Victor Chen YJ. Phoenixmap: An abstract approach to visualize 2D spatial distributions. IEEE Trans. on Visualization and Computer Graphics, 2019, 27(3): 2000−2014.

[26] Beilschmidt C, Mattig M, Fober T, Seeger B. An efficient aggregation and overlap removal algorithm for circle maps. Geoinformatica, 2019, 23: 473−498.

[27] Dwyer T, Stuckey PJ, Marriott K. Fast node overlap removal. In: Proc. of the Int'l Symp. on Graph Drawing, Vol.3843. 2005. 153−164.

[28] Gansner E, Hu YF. Efficient, proximity-preserving node overlap removal. Journal of Graph Algorithms and Applications, 2010, 14(1): 53−74.

[29] Strobelt H, Spicker M, Stoffel A, Keim D, Deussen O. Rolled-out wordles: A heuristic method for overlap removal of 2D data representatives. In: Proc. of the IEEE-TVCG Conf. on Visualization. 2012, 31(3): 1135−1144.

[30] Eler DM, Paulovich FV, Hilasaca GM, Martins RM, Marcílio-Jr WE. Overlap removal of dimensionality reduction scatterplot layouts. arXiv:1903.06262, 2021

[31] Misue K, Eades P, Lai W, Sugiyama K. Layout adjustment and the mental map. Journal of Visual Languages & Computing, 1995, 6(2): 183−210.

[32] Hayashi K, Inoue M, Masuzawa T, Fujiwara H. A layout adjustment problem for disjoint rectangles preserving orthogonal order. Systems and Computer in Japan, 1998, 33(2): 183−197.

[33] Nachmanson L, Nocaj A, Bereg S, Zhang LS, Holroyd A. Node overlap removal by growing a tree. Journal of Graph Algorithms and Applications, 2016, 21(5): 857−872.

[34] Koh K, Lee BS, Kim BY, Seo JW. ManiWordle: Providing flexible control over wordle. IEEE Trans. on Visualization and Computer Graphics, 2010, 16(6): 1190−1197.

[35] Keim DA, Herrmann A. The gridfit algorithm: An efficient and effective approach to visualizing large amounts of spatial data. In: Proc. of the IEEE Visualization. 1998.

[36] Schader P, Beckmann R, Graner6 L, Bernard J. LayoutExOmizer: Interactive exploration and optimization of 2D data layouts. In: Proc. of the Vision, Modeling, and Visualization. 2021.

[37] Barnes J, Hut P. A hierarchical o(*n*log*n*) force calculation algorithm. Nature, 1986, 324: 446−449.

[38] Lyons KA, Meijer H, Rappaport D. Algorithms for cluster busting in anchored graph drawing. Journal of Graph Algorithms and Applications, 1998, 2(1): 1−24.

[39] Zhao HS, Lü L, Bo ZT. Variational circular treemaps for hierarchical data. Ruan Jian Xue Bao/Journal of Software, 2016, 27(5): 1103−1113 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4952.htm [doi: 10.13328/j.cnki.jos.004952]

[40] Bourke P. Calculating the area and centroid of a polygon. Swinburne University of Technology, 1988. http://paulbourke.net/geometry/polygonmesh/

附中文参考文献:

[11] 周芳芳, 李俊材, 黄伟, 王俊韡, 赵颖. 基于维度扩展的 Radviz 可视化聚类分析方法. 软件学报, 2016, 27(5): 1127−1139. http://www.jos.org.cn/1000-9825/4951.htm [doi: 10.13328j.cnki.jos.004951]

[12] 唐磊, 李学庆, 刘洋. 基于维密度和聚类得散点图. 软件学报, 2010, 21: 194−204. http://www.jos.org.cn/1000-9825/10021.htm

[13] 周志光, 汤成, 刘玉华, 刘亚楠, 许建龙. 降维空间视觉认知增强的多维时变数据可视分析方法. 计算机辅助设计与图形学学报, 2018, 30(7): 1194−1204.

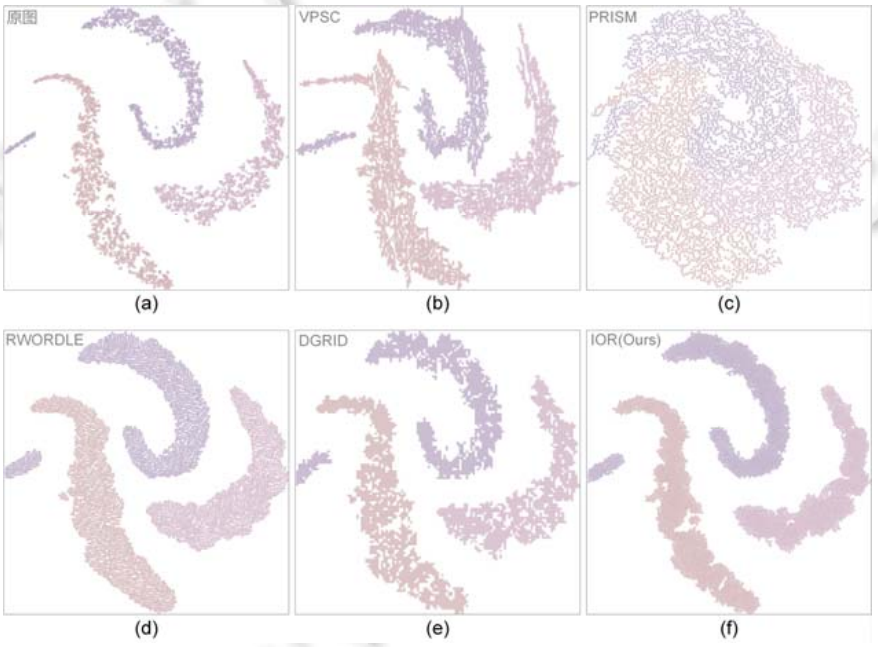
[14] 丁世飞, 徐晓, 王艳茹. 基于不相似性度量优化的密度峰值聚类算法. 软件学报, 2020, 31(11): 3321−3333. http://www.jos.org.cn/1000-9825/5813.htm [doi: 10.13328/j.cnki.jos.005813]

[15] 袁喆, 文继荣, 魏哲巍, 刘家俊, 姚斌, 郑凯. 大数据实时交互式分析. 软件学报, 2020, 31(1): 162−182. http://www.jos.org.cn/1000-9825/5886.htm [doi: 10.13328/j.cnki.jos.005886]

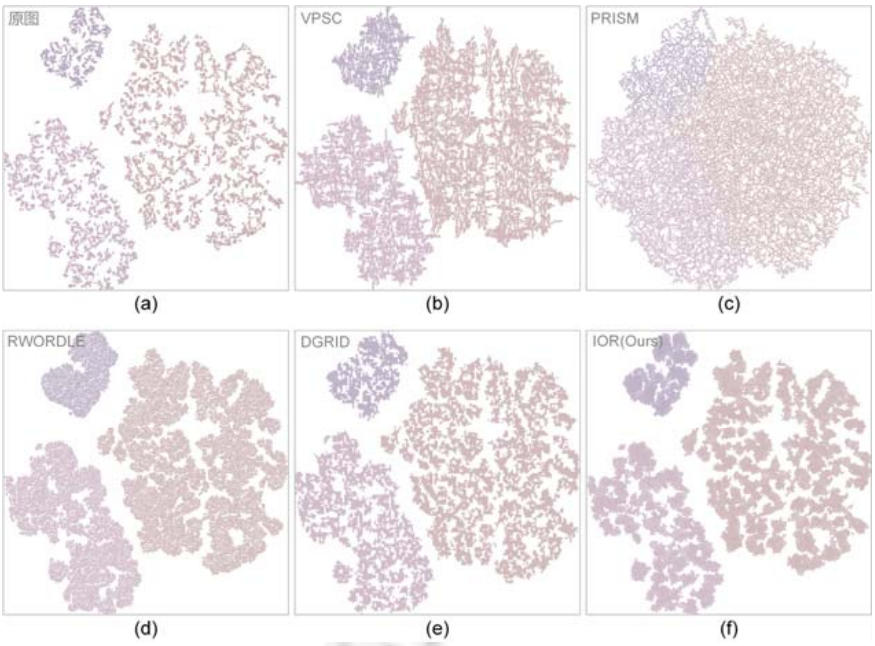
[39] 赵海森, 吕琳, 薄志涛. 面向层次化数据的变分圆形树图. 软件学报, 2016, 27(5): 1103−1113. http://www.jos.org.cn/1000-9825/4952.htm [doi: 10.13328/j.cnki.jos.004952]

附录

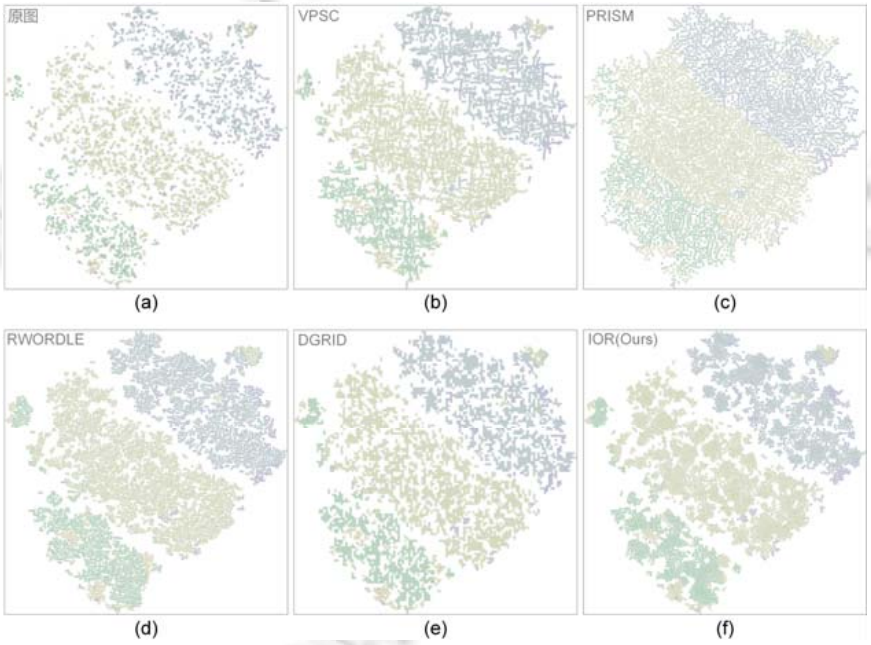
在正文实验中使用了 5 个数据集, 但正文中只提供了 2 个数据集的去重叠效果图, 另外 3 个数据集的去重叠效果图如附图 1–附图 3 所示.



附图 1 Abalone 数据集使用 t-SNE 投影算法获得的散点图(a)和采用 VPSC 算法(b)、PRISM 算法(c)、RWORDLE 算法(d)、DGRID 算法(e)以及本文 IOR 算法(f)获得的去重叠散点图



附图 2　Wine_Quality 数据集使用 t-SNE 投影算法获得的散点图(a)和采用 VPSC 算法(b)、PRISM 算法(c)、RWORDLE 算法(d)、DGRID 算法(e)以及本文 IOR 算法(f)获得的去重叠散点图



附图 3　AI4I2020 数据集使用 t-SNE 投影算法获得的散点图(a)和采用 VPSC 算法(b)、PRISM 算法(c)、RWORDLE 算法(d)、DGRID 算法(e)以及本文 IOR 算法(f)获得的去重叠散点图

除了正文实验外, 我们还在 12 个附加数据集上进行了客观评估实验, 评估指标与正文实验保持一致. 附表 1 给出了实验结果(表中黑色加粗数值表示在某个数据集和某个指标条件下最好的实验结果). 实验结果表明: 本文算法在 5 个结构保持性能指标上表现最佳, 在时间消耗上的表现也相对较好.

附表 1 IOR 与 4 个参考算法在附加数据集上的客观评估实验结果

数据集	算法	评估指标					
		移动距离 (ED)	面积增长 (SI)	形状保持 (SP)	正交顺序 (OO)	邻域保持 (NP)	时间损耗 (TC)
Digits	VPSC	13.093	1.113	0.003 3	0.027	0.793	0.192
	PRISM	60.603	1.571	0.078 5	0.079	0.76	25.434
	RWORDLE	22.393	1.097	0.005 7	0.037	0.736	6.652
	DGRID	17.550	1.128	0.013 4	0.019	0.798	**0.084**
	IOR(Ours)	**6.462**	**1.047**	**0.001 3**	**0.016**	**0.849**	0.45
D31	VPSC	13.317	1.041	0.009 8	0.024	0.763	0.431
	PRISM	81.825	1.702	0.081 0	0.068	0.688	157.712
	RWORDLE	18.725	1.004	0.000 4	0.028	0.723	13.498
	DGRID	15.777	1.076	0.004 0	0.016	0.778	**0.086**
	IOR(Ours)	**5.867**	**1.004**	**0.000 2**	**0.012**	**0.847**	0.472
Abalone	VPSC	29.427	1.302	0.164 0	0.057	0.637	1.164
	PRISM	119.951	2.057	0.230 6	0.121	0.632	237.7
	RWORDLE	29.427	1.102	0.007 6	0.038	0.614	191.542
	DGRID	21.776	1.158	0.018 4	0.03	0.733	**0.19**
	IOR(Ours)	**7.128**	**1.051**	**0.001 0**	**0.016**	**0.775**	1.248
Wine_Quality	VPSC	8.212	1.066	0.004 2	0.057	0.764	1.151
	PRISM	109.060	2.128	0.187 5	0.121	0.648	683.832
	RWORDLE	10.558	1.046	0.002 3	0.038	0.753	29.352
	DGRID	25.553	1.151	0.052 5	0.030	0.771	**0.177**
	IOR(Ours)	**4.804**	**1.022**	**0.000 7**	**0.016**	**0.844**	0.447
AI4I2020	VPSC	12.040	1.127	0.013 9	0.023	0.663	5.441
	PRISM	195.797	2.958	0.339 2	0.069	0.579	2 531.733
	RWORDLE	12.448	1.044	0.001 1	0.018	0.685	385.633
	DGRID	21.235	1.149	0.013 4	0.017	0.759	**0.422**
	IOR(Ours)	**4.156**	**1.027**	**0.000 8**	**0.009**	**0.810**	0.996
Absenteeism at_work	VPSC	3.991	1.012	0.000 5	**0.008**	0.946	**0.045**
	PRISM	15.653	1.156	0.030 0	0.028	0.839	5.476
	RWORDLE	6.036	1.023	0.001 0	0.012	0.923	0.087
	DGRID	13.647	1.073	0.023 1	0.016	0.880	0.087
	IOR(Ours)	**3.796**	**1.010**	**0.000 5**	0.009	**0.946**	0.218
Banana	VPSC	30.248	1.250	0.112 3	0.046	0.641	1.780
	PRISM	147.851	2.432	0.179 2	0.089	0.624	507.163
	RWORDLE	31.573	1.127	0.003 9	0.038	0.599	210.057
	DGRID	25.231	1.186	0.030 7	0.026	0.730	**0.163**
	IOR(Ours)	**8.076**	**1.070**	**0.000 4**	**0.017**	**0.744**	5.708
Car	VPSC	5.802	1.049	0.004 9	0.011	0.876	0.156
	PRISM	42.529	1.473	0.363 8	0.059	0.738	52.972
	RWORDLE	9.820	1.049	0.004 7	0.015	0.837	1.104
	DGRID	19.570	1.107	0.035 5	0.017	0.832	**0.092**
	IOR(Ours)	**4.872**	**1.030**	**0.000 8**	**0.009**	**0.894**	0.352
Contraceptive	VPSC	10.481	1.109	0.016 6	0.021	0.810	**0.123**
	PRISM	43.548	1.392	0.035 6	0.073	0.712	28.43
	RWORDLE	12.339	1.056	0.002 6	0.024	0.791	1.606
	DGRID	17.591	1.095	0.005 2	0.021	0.802	0.164
	IOR(Ours)	**6.147**	**1.032**	**0.001 0**	**0.015**	**0.862**	0.373
Diabetes	VPSC	**0.797**	1.003	0.000 1	**0.002**	0.986	0.025
	PRISM	4.515	1.024	0.001 5	0.011	0.942	1.664
	RWORDLE	2.608	1.006	0.000 9	0.005	0.968	**0.013**
	DGRID	12.209	1.026	0.008 1	0.014	0.883	0.101
	IOR(Ours)	1.124	**1.001**	**0.000 1**	0.004	**0.979**	0.191
Facebook_live	VPSC	34.127	1.285	0.199 2	0.062	0.666	0.321
	PRISM	64.598	1.577	0.167 5	0.092	0.561	92.994
	RWORDLE	26.795	1.121	0.003 1	0.052	0.655	15.560
	DGRID	25.632	1.200	0.024 1	0.030	0.744	**0.093**
	IOR(Ours)	**10.939**	**1.085**	**0.000 9**	**0.027**	**0.780**	1.298
Iris	VPSC	1.485	1.009	0.000 3	**0.007**	0.978	0.011
	PRISM	6.826	1.098	0.012 5	0.021	0.935	0.284
	RWORDLE	2.992	1.016	0.001 5	0.013	0.969	**0.006**
	DGRID	12.074	1.048	0.039 6	0.022	0.894	0.100
	IOR(Ours)	**1.329**	**1.002**	**0.000 2**	0.009	**0.976**	0.137

附表 1 IOR 与 4 个参考算法在附加数据集上的客观评估实验结果(续)

数据集	算法	评估指标					
		移动距离 (ED)	面积增长 (SI)	形状保持 (SP)	正交顺序 (OO)	邻域保持 (NP)	时间损耗 (TC)
Mammographic	VPSC	15.610	1.132	0.018 9	0.027	0.789	0.011
	PRISM	47.909	1.406	0.151 9	0.076	0.672	0.284
	RWORDLE	16.181	1.071	**0.003 0**	0.030	0.798	**0.006**
	DGRID	16.667	1.123	0.006 6	**0.018**	0.838	0.100
	IOR(Ours)	**8.739**	**1.063**	0.003 3	0.020	**0.874**	0.137
Page_blocks	VPSC	22.527	1.282	0.084 8	0.041	0.625	1.805
	PRISM	146.359	2.464	0.381 4	0.081	0.578	624.703
	RWORDLE	21.033	1.082	0.004 3	0.031	0.636	96.412
	DGRID	26.894	1.199	0.030 1	0.018	0.746	**0.197**
	IOR(Ours)	**7.172**	**1.055**	**0.001 6**	**0.016**	**0.770**	1.969
R15	VPSC	5.566	1.008	0.000 5	0.014	0.908	**0.038**
	PRISM	25.619	1.113	0.011 3	0.044	0.826	2.697
	RWORDLE	8.413	1.010	0.001 9	0.022	0.878	0.129
	DGRID	15.522	1.029	0.008 1	0.017	0.862	0.096
	IOR(Ours)	**3.862**	**1.003**	**0.000 2**	**0.012**	**0.924**	0.319
Twonorm	VPSC	44.209	1.162	0.097 2	0.044	0.605	3.145
	PRISM	247.434	3.492	1.493 0	0.138	0.661	728.803
	RWORDLE	82.329	1.167	0.046 1	0.075	0.570	3 556.398
	DGRID	56.023	1.200	0.142 0	0.070	0.743	**0.168**
	IOR(Ours)	**5.444**	**1.008**	**0.000 6**	**0.010**	**0.811**	6.809
Sprial	VPSC	0.000 2	2.394	1.000	0.005	0.963	0.017
	PRISM	0.001 6	21.740	1.032	0.039	0.867	1.277
	RWORDLE	0.000 0	2.507	1.000	0.003	0.982	**0.009**
	DGRID	0.007 1	13.205	1.022	0.014	0.910	0.094
	IOR(Ours)	**0.000 1**	**0.577**	**1.000**	**0.002**	**0.986**	0.207

赵颖(1980－), 男, 博士, 教授, CCF 专业会员, 主要研究领域为可视化, 可视分析.

秀昱宏(1997－), 男, 硕士生, 主要研究领域为可视化, 可视分析.

唐涛(2001－), 男, 本科生, 主要研究领域为可视化, 可视分析.

文陈飞宇(2001－), 男, 本科生, 主要研究领域为机器学习, 可视化.

陈晓慧(1983－), 女, 博士, 副教授, 主要研究领域为可视化, 可视分析.

尤旸(1982－), 女, 设计总监, 主要研究领域为营销分析, 软件工程, 数据可视化.

周芳芳(1980－), 女, 博士, 教授, 主要研究领域为可视化, 虚拟现实.