软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2022,33(4):1326−1337 [doi: 10.13328/j.cnki.jos.006465]

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

基于随机近邻嵌入的判别性特征学习*

赵 辉[1,2], 王红军[1,2], 彭 博[1,2], 龙治国[1,2], 李天瑞[1,2]

[1](西南交通大学 计算机与人工智能学院, 四川 成都 611756)

[2](综合交通大数据应用技术国家工程实验室, 四川 成都 611756)

通信作者: 王红军, E-mail: wanghongjun@swjtu.edu.cn

摘 要: 特征学习是机器学习中的一项重要技术, 研究从原始数据中学习后置任务所需的数据表示. 目前, 多数特征学习算法侧重于学习原始数据中的拓扑结构, 忽略了数据中的判别信息. 基于此, 提出了基于随机近邻嵌入的判别性特征学习模型. 该模型将对判别信息的学习与对拓扑结构的学习融合在一起, 通过迭代求解的方式, 同时完成对这两者的学习, 从而得到原始数据具有判别性的特征表示, 可以显著提升机器学习算法的性能. 多个公开数据集上的实验结果验证了该模型的有效性.

关键词: 特征学习; 随机近邻嵌入; 判别性学习

中图法分类号: TP181

中文引用格式: 赵辉, 王红军, 彭博, 龙治国, 李天瑞. 基于随机近邻嵌入的判别性特征学习. 软件学报, 2022, 33(4): 1326−1337. http://www.jos.org.cn/1000-9825/6465.htm

英文引用格式: Zhao H, Wang HJ, Peng B, Long ZG, Li TR. Discriminant Feature Learning Based on Stochastic Neighbor Embedding. Ruan Jian Xue Bao/Journal of Software, 2022, 33(4): 1326−1337 (in Chinese). http://www.jos.org.cn/1000-9825/6465.htm

Discriminant Feature Learning Based on Stochastic Neighbor Embedding

ZHAO Hui[1,2], WANG Hong-Jun[1,2], PENG Bo[1,2], LONG Zhi-Guo[1,2], LI Tian-Rui[1,2]

[1](School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China)

[2](National Engineering Laboratory of Integrated Transportation Big Data Application Technology, Chengdu 611756, China)

Abstract: Feature learning is an important technique in machine learning, which studies data representation learning required by the post task from raw data. At present, most feature learning algorithms focus on learning topological structure of the original data, but ignore the discriminant information in the data. This study proposes a novel model called discriminant feature learning based on *t*-distribution stochastic neighbor embedding (DTSNE). In this model, the learning of discriminant information and the learning of topology structure are fused together, so both of them are learned to obtain the discriminant feature representation of the original data through iterative solution, which can significantly improve the performance of the machine learning algorithm. Experimental results on multiple open data sets demonstrate the effectiveness of the proposed model.

Key words: feature learning; stochastic neighbor embedding; discriminant learning

在自然语言处理[1]、计算机视觉[2]、机器学习[3]以及生物信息[4]等领域, 存在着大量的高维数据. 高维数据蕴含着大量有价值的信息, 但也存在噪声及冗余信息, 会导致后续的数据分析与挖掘效果不理想. 同时, 随着数据维度的增加, 数据的分布将会变得稀疏, 产生“维数灾难”[5], 严重影响后续机器学习算法的效果.

目前, 多数机器学习算法的性能依赖于数据表示, 一个好的数据表示能够显著提升算法的性能[6]. 基于

* 基金项目: 国家自然科学基金(61806170, 61773324); 国家重点研发计划(2017YFB1401401)
本文由“面向开放场景的鲁棒机器学习”专刊特约编辑陈恩红教授、李宇峰副教授、邹权教授推荐.
收稿时间: 2021-03-10; 修改时间: 2021-07-16; 采用时间: 2021-08-27; jos 在线出版时间: 2021-10-26

此, 特征工程首先被引入到机器学习领域. 特征工程是利用人类经验与领域专业知识从原始数据中获取有效数据表示的一种技术, 曾经得到广泛应用. 但是, 特征工程需要大量的领域专业知识, 并且随着后置任务的变换, 特征工程也要做相应调整, 无法自动地从原始数据中提取出后置任务所需的数据表示. 因此, 特征工程逐渐被特征学习所取代.

特征学习研究如何从原始数据中自动发现并获取后置任务所需的数据表示, 是机器学习领域中的一项重要技术. 根据关注点的不同, 特征学习算法大致可以分为 3 类: 基于线性投影的特征学习、基于流形的特征学习和基于深度学习模型的特征学习. 基于线性投影的特征学习关注于捕获原始数据中的某一投影方向; 而流形学习则主要关注于保持原始空间中的拓扑结构, 如样本之间的距离或局部的线性结构; 基于深度学习模型的特征学习主要是追求最小化重构误差, 使得学习到的特征可以最大化地重构出原始数据.

线性特征学习研究起步较早, 目前已有比较成熟的方法. 但是因为存在对数据的线性假设, 无法处理实际中广泛存在的非线性数据. 因此, 本文专注于非线性特征学习算法. 基于流形的非线性特征学习基于流形假设, 认为观察到的数据实际上分布在嵌入高维空间的低维流形上, 这个低维流形就蕴含了数据的本征维度和本征结构[7]. 流形学习算法一般通过保持局部拓扑结构来实现, 因此在训练时无需海量数据, 广泛适用于各类特征学习场景. 基于深度学习模型的非线性特征学习算法基于数据的分布式表示假设, 认为观察到的数据是由多个层次上的许多不同因子交互产生的, 而隐藏单元能够学习到可以解释数据的潜在因子[8]. 因此, 它通过堆叠多层学习节点来进行特征学习, 并且将最后一层的输出结果作为数据的特征表示. 由于学习节点堆叠层数较深, 深度学习模型往往需要海量数据, 否则极易出现数据过拟合; 同时, 由于数据量大且模型参数多, 对于计算设备的性能也有一定要求. 因此, 基于深度学习模型的非线性特征学习算法广泛适用于有海量数据与高性能计算设备的场景. 本文重点研究无需海量数据且适用范围更广的基于流形的非线性特征学习算法.

现有的大部分流形学习算法只注重于在特征学习的过程中保持原始数据的拓扑信息, 却忽略了数据中的判别信息. 在部分情况下, 数据中的判别信息比拓扑结构等其他信息更加重要, 如对一张人脸图片进行特征学习时, 五官及轮廓等信息就相当重要, 如果学习到的特征可以让五官及轮廓更加分明, 则说明特征学习的效果较好. 这里的五官轮廓信息就可以看作是数据中的判别信息.

因此, 本文提出了基于随机近邻嵌入的判别性特征学习(discriminant feature learning based on *t*-distribution stochastic neighbor, DTSNE)模型. 该模型将随机近邻嵌入的优化目标与判别信息的优化目标相融合, 作为新的优化目标. 接下来应用凸优化相关方法, 求解目标函数的最小值. 当目标函数值最小时, 模型即可最大程度地同时学习到原始数据中的拓扑结构和判别信息, 从而得到原始数据更具判别性的特征表示.

本文主要贡献如下:

(1) 提出了 DTSNE 模型, 同时学习原始数据中的拓扑结构和判别信息, 得到原始数据的特征表示, 可以显著提升后续机器学习算法的性能;

(2) 使用动量梯度下降法对 DTSNE 模型进行推理求解, 得到了各参数的更新公式, 然后根据推理求解过程设计了 DTSNE 模型的算法;

(3) 在 12 个真实的数据集上, 使用准确率、纯度和 *F*1-score 这 3 个评价指标, 验证了本文所提出 DTSNE 模型的性能.

本文第 1 节介绍特征学习领域的相关工作. 第 2 节介绍基于随机近邻嵌入的判别性特征学习模型, 包括理论分析、目标公式、模型优化、算法描述以及复杂度分析. 第 3 节是实验与分析, 介绍实验数据集、评价指标、实验设置以及结果分析. 第 4 节进行全文总结, 并对未来的工作进行展望.

1 相关工作

为了获取数据更好的表示, 以提升机器学习算法的性能, 有很多的特征学习算法被提出. 按照数据变换方式的不同, 特征学习可以分为线性特征学习与非线性特征学习两类.

1.1 线性特征学习

线性特征学习假设从原始空间到特征空间的映射函数是线性的, 基本思想是: 找到一个投影矩阵 W, 然后将数据投影到特征空间, 并且捕捉到所关注的数据信息, 如方差、相关性等. 如果数据满足假设条件, 则线性特征学习算法可以获得原始数据的准确表示. 线性特征学习算法理论完备, 解释性好, 一般都可转换为特征值或广义特征值问题求解, 求解效率高. 经典的线性特征学习方法有主成分分析(principal component analysis, PCA)[9]和线性判别分析(linear discriminant analysis, LDA)[10].

PCA 认为, 投影后数据的方差反映该投影方向的优劣. 如果原始数据沿着某一方向投影后方差较小, 说明该投影方向无法很好地反映原始数据的特点. 因此, PCA 通过寻找一组线性正交基来使得投影后的数据方差最大, 从而完成数据从原始空间到其特征空间的映射. LDA 从样本类别的角度出发, 要求映射后类内距离尽可能小, 类间距离尽可能大. 它通过定义类间散度矩阵和类内散度矩阵, 寻找使得变换后类间散度矩阵与类内散度矩阵迹的比值最大的正交投影, 从而有监督地完成数据映射.

1.2 非线性特征学习

线性特征学习方法简单, 计算效率高, 在实际中取得了广泛的应用. 但是由于存在对数据的线性假设, 线性特征学习算法无法很好地处理图像等非线性数据. 为了应对生活中广泛存在的非线性数据, 非线性特征学习方法应运而生, 并逐渐成为学者研究的热点. 常见的非线性特征学习方法可以分为三大类: 基于核方法的特征学习、基于流形的特征学习和基于深度学习模型的特征学习.

核方法是将核技巧应用在线性特征学习算法上得到的, 即: 将非线性数据投影至更高维的核空间, 使其在核空间中呈现线性结构, 然后再应用线性特征学习算法, 如核主成分分析(kernel principal component analysis, KPCA)[11].

流形学习自 2000 年被正式提出以来受到了广泛关注, 各种流形学习算法被不断提出. Tenenbaum 等人使用测地距离替代欧氏距离, 提出了等度量映射(isometric mapping, ISOMAP)[12]算法. ISOMAP 认为高维空间的直线距离具有误导性, 因此使用测地线来度量样本间的距离; 然后应用经典的多维尺度放缩(multidimensional scaling, MDS)[13]算法得到嵌入的低维流形. Roweis 等人基于邻域内样本线性结构的保持, 提出了局部线性嵌入(locally linear embedding, LLE)[14]算法. LLE 将高维空间中的样本点表示为其邻域内其他样本点的线性组合, 并计算出最佳重构系数, 希望在低维空间中线性关系及重构系数都能得到保持. Belkin 等人基于图谱理论, 提出了拉普拉斯特征映射(Laplacian eigenmaps, LE)[15]算法. LE 通过数据集的邻域信息构建一个邻接图, 通过最小化基于图的代价函数来确保流形上相互靠近的样本点在低维空间中也彼此靠近, 从而保留数据的局部结构. Hinton 等人将概率方法应用到特征学习, 提出了随机近邻嵌入(stochastic neighbor embedding, SNE)[16]算法. 该算法使用概率分布来描述样本结构, 通过最小化高维空间与低维空间样本分布的 KL 散度来保持数据的全局结构. Donoho 等人将 LLE 的邻域内线性重构替换为 Hessian 变换, 提出了 Hessian LLE (HLLE)[17]算法, 在非凸样本集上有很好的表现. Li 等人提出了黎曼流形学习(Riemannian manifold learning, RML)[18]算法, 该算法通过为给定的黎曼流形构造坐标图来获取嵌入的低维流形. 除了上述经典的非线性特征学习算法外, 近年来, 新的流形学习算法也在不断被提出, 如增强局部投影保持(enhanced locality preserving, ELP)[19]算法、无监督大规模图嵌入(unsupervised large graph embedding, ULGE)[20]算法、图优化半监督投影(semisupervised projection with graph optimization, SPGO)[21]算法等.

2006 年, 深度学习被正式提出, 基于深度学习模型的特征学习算法也受到了人们的广泛关注. 基于深度学习模型的特征学习可以分为两类: 一类是受限玻尔兹曼机(restricted Boltzmann machine, RBM)[22], 另一类则是自动编码器(autoencoder)[23]. RBM 通常用作深度学习模型的基本构建块, 由 RMB 堆叠而成的深度信念网络(deep belief net, DBN)[24]具有很好的特征学习能力. 自动编码器由编码器和解码器组成, 编码器负责从原始数据中提取特征, 解码器负责利用提取的特征重构原始数据, 通过最小化重构误差, 自动编码器可以学习到原始数据的特征表示.

2 基于随机近邻嵌入的判别性特征学习

2.1 理论分析

DTSNE 在获取数据的特征表示的时候不仅可以保持原始数据的拓扑结构, 还可以从原始数据中学习得到判别信息, 并将判别信息融入数据的特征表示中, 提升机器学习算法的性能. 本节对 DTSNE 进行理论分析, 主要分为两个方面: (1) 如何在特征学习的过程中保持原始数据的拓扑结构; (2) 如何定义数据中的判别信息, 并将其学到的判别信息融入到数据的特征表示中.

(1) 保持拓扑结构

数据的理想表示应该能够反映数据的本质特征, 而流形学习旨在发现数据的本征维度和实际分布的流形, 因此, 本文基于流形学习中的随机近邻嵌入算法来保持原始数据的拓扑结构.

随机近邻嵌入通过计算数据点对在高维空间中的联合分布, 并选择具有相似分布的低维流形来完成数据点从高维空间到低维空间的映射. 本节介绍 t 分布随机近邻嵌入(t-distributed stochastic neighbor embedding, TSNE)[25].

假设样本集为 $X=\{x_1,x_2,\ldots,x_N\}$, $x_i\in\mathbb{R}^H$, X 的低维嵌入结果为 $Y=\{y_1,y_2,\ldots,y_N\}$, $y_i\in\mathbb{R}^L$, 定义如下条件概率:

$$p_{j|i}=\begin{cases}\dfrac{\exp(-\|x_i-x_j\|/(2\sigma_i^2))}{\sum_{k\neq i}\exp(-\|x_i-x_k\|/(2\sigma_i^2))}, & \text{if } i\neq j\\ 0, & \text{if } i=j\end{cases} \tag{1}$$

对于任意的 $i(1\leqslant i\leqslant N)$, 显然有 $\sum_j p_{j|i}=1$. $p_{j|i}$ 表示高维空间中数据点 x_i 选择 x_j 作为它的邻居的概率, 与 x_j 在以 x_i 为中心的高斯分布下的概率密度成正比. σ_i 表示高斯分布的标准差, 可以通过条件分布的困惑度[23]等于预设值来二分搜索得到.

由于条件概率 $p_{j|i}$ 一般不等于 $p_{i|j}$,为了方便后面的计算,定义联合概率:

$$p_{ij}=\frac{p_{j|i}+p_{i|j}}{2N} \tag{2}$$

其中, N 表示样本个数. 显然有 $p_{ij}=p_{ji}$, $p_{ii}=0$, $\sum_{i,j}p_{ij}=1$.

由于数据从高维空间映射到低维空间的时候存在拥挤问题, 所以 TSNE 使用了重尾的 t 分布来描述低维流形上数据点间的联合概率, 定义如下:

$$q_{ij}=\frac{(1+\|y_i-y_j\|^2)^{-1}}{\sum_{k\neq l}(1+\|y_k-y_l\|^2)^{-1}} \tag{3}$$

最后, TSNE 使用 KL 散度来衡量两个联合分布 P 与 Q 的差异, 目标函数为

$$\min_{y_i}C=KL(P\|Q)=\sum_i\sum_j p_{ij}\log\frac{p_{ij}}{q_{ij}} \tag{4}$$

(2) 学习判别信息

线性判别分析将样本的标签作为判别信息, 用来指导特征学习. 而在无监督特征学习算法中不存在样本标签, 也就无法从先验知识中引入判别信息. 因此, 如何在无监督特征学习中定义判别信息, 就成为一个亟待解决的问题.

聚类是机器学习中的一个重要研究领域, 其基本思想是: 按照一定的准则, 将相似的数据放在一起, 称为一个簇; 而相异的数据则在不同的簇里. 由聚类算法从数据中学到的簇结构具有重要的意义, 尤其是在缺乏对数据的先验知识的时候. 比如: 我们要对一些数据进行分析, 但是没有任何关于这些数据的先验知识, 此时, 我们可以使用聚类算法将数据划分为不同的簇, 以了解原始数据的结构, 并通过对每个簇中代表点的深入研究, 来探究原始数据的性质.

受此启发, 本文在无监督的条件下将判别信息定义为数据的簇结构, 并且希望特征学习算法学习到的数

据表示不仅可以保持原始数据的拓扑结构, 而且可以学习到数据中的判别信息. 关于判别信息的优化目标为

$$\min_{y_i} D=\sum_k \sum_{y_i\in C_k} \| y_i-c_k \|^2 \tag{5}$$

其中, y_i 为样本点 x_i 的特征表示, c_k 为样本点特征空间的簇中心.

2.2 目标公式

经过上面的理论分析, 本节给出 DTSNE 的目标公式. 假设给定的数据集为 $X=\{x_1,x_2,\dots,x_N\}$, $x_i\in\mathbb{R}^H$, 对应的低维嵌入为 $Y=\{y_1,y_2,\dots,y_N\}$, $y_i\in\mathbb{R}^L$, 则目标公式定义如下:

$$\begin{aligned}&\min_{y_i,c_k} E=\alpha KL(P\|Q)+\beta\frac{1}{M}\sum_k\sum_{y_i\in C_k}\| y_i-c_k \|^2=\alpha\sum_i\sum_j p_{ij}\log\frac{p_{ij}}{q_{ij}}+\beta\frac{1}{M}\sum_k\sum_{y_i\in C_k}\| y_i-c_k \|^2=\alpha E_1+\frac{\beta}{M}E_2\\ &\text{s.t. } 0<\alpha<1,\ 0<\beta<1,\ \alpha+\beta=1, 1\leqslant i\leqslant N,\ 1\leqslant j\leqslant N,\ 1\leqslant k\leqslant K\end{aligned} \tag{6}$$

其中, E_1 表示特征学习要保持原始数据的拓扑结构; E_2 表示特征学习要学习到数据中的判别信息; α, β为调节因子, 用于调节两项的权重; M 为放缩因子, 用来将 E_2 放缩到与 E_1 相同的数据尺度上, 本文取 $M=\dfrac{\max\{\| y_i-c_k \|^2\}}{\max\{p_{ij}\log(p_{ij}/q_{ij})\}}$; C_k 表示第 k 个簇; c_k 表示第 k 个簇在特征空间的簇中心.

2.3 模型优化

针对目标函数(6), 可以使用梯度下降法直接求解. 为了加快收敛速度, 本文使用动量梯度下降法[26], 它是在梯度下降法的基础上增加了一个动量项, 即: 不仅考虑当前的梯度, 还要考虑历史梯度. 动量梯度下降法的基本形式如下所示:

$$m^{(t-1)}=\mu m^{(t-2)}-\eta\frac{\partial E(y_i^{(t-1)})}{y_i},\ y_i^{(t)}=y_i^{(t-1)}+\lambda m^{(t-1)} \tag{7}$$

其中, $m^{(t-1)}$表示第 t–1 轮更新时的动量; $y_i^{(t)}$ 表示参数 y_i 第 t 轮更新后的结果; E 表示目标函数; μ, η为调节因子, 用于调节历史梯度和当前梯度的权重; λ为学习率, 表示梯度下降时的迭代步长.

接下来需要求出目标函数关于各参数的导数, 这里的参数有两类: y_i 和 c_k. 目标函数由两部分组成并且是相加的关系, 所以可以分别求梯度, 然后再将其相加. E_1, E_2 前面的系数是常数, 求导时可以略去. 为了推理时表述的方便, 做如下标记:

$$d_{ij}=\|y_i-y_j\| \tag{8}$$

则 q_{ij} 可以被重写为一种更为简洁的形式:

$$q_{ij}=\frac{(1+d_{ij}^2)^{-1}}{\sum_{k\neq l}(1+d_{kl}^2)^{-1}} \tag{9}$$

在求 E_1 关于 y_i 的偏导时, 使用链式法则, 先求 E_1 关于中间变量 d_{ij} 的偏导, 然后再求 d_{ij} 关于 y_i 的偏导. 求解过程如下:

$$\begin{aligned}\frac{\partial E_1}{\partial y_i}&=\sum_{j\neq i}\left(\frac{\partial E_1}{\partial d_{ij}}\frac{\partial d_{ij}}{\partial y_i}+\frac{\partial E_1}{\partial d_{ji}}\frac{\partial d_{ji}}{\partial y_i}\right)\\ &=2\sum_{j\neq i}\frac{\partial E_1}{\partial d_{ij}}\frac{\partial d_{ij}}{\partial y_i}\\ &=2\sum_{j\neq i}2d_{ij}(p_{ij}-q_{ij})(1+d_{ij}^2)^{-1}\cdot\frac{y_i-y_j}{d_{ij}}\\ &=4\sum_{j\neq i}(p_{ij}-q_{ij})(1+\| y_i-y_j \|^2)^{-1}(y_i-y_j)\end{aligned} \tag{10}$$

E_2 中有两次求和, 展开后为若干项相加, 只有与 y_i 有关的那一项求导结果不为 0. 求解过程如下:

$$\frac{\partial E_2}{\partial y_i}=\frac{\partial(\sum_k\sum_{y_l\in C_k}\| y_l-c_k\|^2)}{\partial y_i}=\frac{\partial(\| y_i-c_k\|^2)}{\partial y_i}=2(y_i-c_k) \tag{11}$$

由上述两项, 可以得到 E 关于 y_i 的偏导:

$$\frac{\partial E}{\partial y_i}=\alpha\frac{\partial E_1}{\partial y_i}+\frac{\beta}{M_1}\frac{\partial E_2}{\partial y_i}=4\alpha\sum_{j\neq i}(p_{ij}-q_{ij})(1+\| y_i-y_j\|^2)^{-1}(y_i-y_j)+2\frac{\beta}{M}(y_i-c_k) \tag{12}$$

将公式(12)代入公式(7), 即可得到参数 y_i 的更新公式.

观察 E_2 的表达式, 可以发现 y_i 和 c_k 具有一定的对称性. 因此, 可以很容易地求出 E_2 关于 c_k 的偏导, 从而得到 E 关于 c_k 的导数:

$$\frac{\partial E}{\partial c_k}=2\frac{\beta}{M}\sum_{y_l\in C_k}(c_k-y_l) \tag{13}$$

将公式(13)代入公式(7), 即可得到 c_k 的迭代更新公式.

2.4 算法描述

在上述两节中, 本文详细介绍了 DTSNE 模型的目标函数, 并完成了模型的推理优化, 得到了参数的更新公式. 下面给出 DTSNE 模型的算法流程.

算法 1. DTSNE 算法.

输入: 数据集 $X=\{x_1,x_2,\ldots,x_N\}$, $x_i\in\mathbb{R}^H$, 聚类簇数 K, 特征空间维度 L, 权重α, β, 最大迭代次数 T;

输出: 特征学习结果 $Y=\{y_1,y_2,\ldots,y_N\}$, $y_i\in\mathbb{R}^L$.

1. 由正态分布随机生成 N 个特征空间中的点 $Y^{(0)}=\{y_1^{(0)},y_2^{(0)},...,y_N^{(0)}\}$ 作为初始的特征表示;
2. 选取 $Y^{(0)}$的前 K 个点作为初始簇中心;
3. 使用公式(1)和公式(2)计算联合概率 P;
4. 执行下面的循环

   for t=1,2,…,T **do**

   (a) 使用公式(3)计算特征空间中的联合概率 Q;

   (b) 使用公式(12)和公式(7)更新 y_i;

   (c) 使用公式(13)和公式(7)更新 c_k;

   End for
5. $Y=(y_1,y_2,\ldots,y_N)$, **return** Y

2.5 复杂度分析

本节对 DTSNE 模型的时间和空间复杂度进行分析. 为了描述的简洁性及推理时的方便, 在不影响结果的前提下, 本节推理时省略了某些低阶项.

在时间复杂度方面, 首先要计算原始空间中任意两点间的联合分布, 需要执行 $\frac{N(N-1)}{2}H$ 次运算; 然后是对参数进行迭代求解. 每次迭代可分为 3 步: (1) 计算特征空间中任意两点间的联合概率, 需要执行 $\frac{N(N-1)}{2}L$ 次运算; (2) 对每个样本点在特征空间的表示进行更新, 需要执行 $\frac{N(N-1)}{2}L$ 次运算; (3) 更新特征空间中的簇中心, 需执行 $L(N+C)$次运算. 执行的运算总次数为

$$\frac{N(N-1)}{2}H+T\left(\frac{N(N-1)}{2}L+\frac{N(N-1)}{2}L+L(N+C)\right)=\frac{N(N-1)}{2}H+TL(N^2+C) \tag{14}$$

因此, DTSNE 的时间复杂度为 $O(N^2\times\max\{H,TL\})$.

在空间复杂度方面, 首先, 模型要存储原始数据, 需要 NH 的空间; 然后要存储数据的特征表示和特征空间的簇中心, 分别需要 NL 和 CL 的空间; 在计算的时候, 要分别存储数据点在原始空间和特征空间的联合概率, 均需要 N^2 的空间. 需要的总空间为

$$NH+NL+CL+2N^2=N(2N+H+L)+CL \tag{15}$$

因为 $L<H$, $C<N$, 因此, DTSNE 的空间复杂度为 $O(N\times\max\{N,H\})$.

3 实验与分析

为了验证DTSNE模型的有效性, 本节选取了多个经典的聚类与分类算法, 并设置了3个对照组: (1) 不进行特征学习; (2) 使用 TSNE 进行特征学习; (3) 使用 DTSNE 进行特征学习. 通过所选算法在这 3 个对照组上的性能, 来验证特征学习的效果.

3.1 实验数据集

本文的实验在 12 个真实的数据集上进行, 数据集均来自微软亚洲研究院多媒体 MSRA-MM (Microsoft research asia multimedia)和 UCI 机器学习数据集, 各数据集的详细信息见表 1.

表 1 实验数据集

数据集	样本数	样本维数	类别数
Ballon	830	892	3
Beer	870	892	3
Airplane	855	892	3
Bus	910	892	3
Ufo	881	899	3
Venus	891	899	3
Webcam	790	899	3
Amber	880	892	3
Bike	839	892	3
Birthdaycake	932	892	3
sEMG_sub	600	2 500	3
arcene_sub	100	10 000	2

3.2 实验设置

为了验证 DTSNE 可以学习到原始数据更具判别性的特征表示, 本文从机器学习的常见任务聚类和分类两个角度出发, 分别进行实验.

- 在聚类方面, 本文选取了 3 种经典聚类算法: k-mean[27], Affinity Propagation(AP)[28]和 Density Peaks(DP)[29], 从而得到 3 组对比实验: (1) 经典的聚类算法组; (2) 基于 TSNE 特征学习的聚类算法组; (3) 基于DTSNE特征学习的聚类算法组. 为了保证实验结果的准确性, 每种方法运行10次, 结果取其平均值;
- 在分类方面, 本文也选取了 3 种经典分类算法: 支持向量机(support vector machine, SVM)[30]、决策树(decision tree, DT)[31]和 K 近邻(k-nearest neighbors, KNN)[32], 从而得到与聚类相似的 3 组对比实验. 因为分类属于有监督学习, 所以本文采取了十折交叉验证来确保实验结果的准确性.

3.3 评价指标

用于评价聚类与分类算法性能的指标有很多, 它们都可以从不同角度反映出算法的优劣. 本文采用经典的准确率[33]同时作为聚类和分类的评价指标.

为了使得评价更加客观充分, 本文还引入了纯度[34]和 $F1$-score[35]分别作为聚类和分类的评价指标.

准确率表示预测标签与真实标签对应正确的样本占总样本的比例, 计算公式如下:

$$Acc=\frac{\sum_{i=1}^{N}\delta(map(r_i),l_i)}{n} \tag{16}$$

其中, n 为样本总数, r_i 为样本 i 的预测标签, l_i 为样本 i 的真实标签, $map(r_i)$表示与预测标签 r_i 相对应的真实标签. 预测标签与真实标签的对应关系是一个组合优化问题, 可以使用匈牙利算法(Hungarian algorithm)[36]求解. $\delta(x,y)$是一个 delta 函数, 当 $x=y$, $\delta(x,y)=1$; 否则, $\delta(x,y)=0$.

纯度表示聚类得到的簇中包含单个类别的程度, 计算公式如下:

$$Purity=\frac{1}{N}\sum_{k=1}^{r}\max_{1\leqslant l\leqslant q}n_k^l \tag{17}$$

其中, r 为聚类得到的簇数, q 为样本类别数, n_k^l 表示类簇 k 中包含的标签为 l 的样本个数. 一般取 $r=q$, 使聚类簇数等于样本类别数.

$F1$-score 是一种同时考虑精度与召回率的综合评价指标, 其计算公式如下:

$$F1\text{-score}=\frac{2\times P\times R}{P+R} \tag{18}$$

其中, $P=\frac{m_{ij}}{|K_j|}$ 表示精确率, $R=\frac{m_{ij}}{|C_i|}$ 表示召回率, C_i 表示真实类别, K_j 表示预测类别, m_{ij} 表示 K_j 中 C_i 的总数.

准确率、纯度及 $F1$-score 的取值范围均为[0,1], 并且数值越大, 表示算法效果越好. 为了更清楚地说明算法间的差异性, 本文采用了 Friedman 检验来评价不同算法的性能差异. Friedman 检验使用的统计量定义如下:

$$\chi_F^2=\frac{12n}{k(k+1)}\left[\sum_j R_j^2-\frac{k(k+1)^2}{4}\right] \tag{19}$$

其中, n 为数据集个数, k 为算法个数, r_{ij} 表示第 j 个算法在第 i 个数据集上的 rank 值, $R_j=\frac{1}{n}\sum_i r_{ij}$ 为第 j 个算法在所有数据集上的平均 rank 值. 为了充分说明算法间性能差异的显著性, 本文将 Friedman 检验的显著性水平设置为 0.01.

3.4 结果分析

本节将对聚类与分类算法在 3 个对照组上的实验结果进行展示并展开详尽的分析, 从而充分验证 DTSNE 模型特征学习的有效性.

表 2 和表 3 分别展示了不同聚类算法在各实验数据集上的准确率和纯度, 同一数据集上的最佳值被加粗显示.

表 2 不同聚类算法准确率对比

数据集	k-means	AP	DP	k-means TSNE	AP TSNE	DP TSNE	k-means DTSNE	AP DTSNE	DP DTSNE
Balloon	0.443 1	0.445 8	0.428 9	0.414 5	0.349 4	0.456 6	0.433 7	0.462 7	**0.592 8**
Beer	0.396 9	0.314 9	0.474 7	0.440 2	0.354 0	0.428 7	0.450 6	**0.485 1**	0.432 2
Airplane	0.437 0	0.364 9	0.375 4	0.480 7	0.393 0	0.438 6	0.422 2	0.429 2	**0.488 9**
Bus	0.452 4	0.418 7	0.445 1	0.468 1	0.379 1	0.572 5	0.467 0	0.375 8	**0.589 0**
Ufo	0.411 7	0.354 1	0.382 5	0.398 4	0.373 4	0.422 2	0.395 0	0.354 1	**0.426 8**
Venus	0.562 5	0.358 0	0.461 3	0.551 1	0.487 1	0.571 3	0.547 7	0.346 8	**0.574 6**
Webcam	0.462 9	0.506 3	0.403 8	0.511 4	0.453 2	0.477 2	**0.512 7**	0.469 6	0.486 1
Amber	0.626 1	0.364 8	0.531 8	0.573 9	0.418 2	0.535 2	0.558 0	0.367 0	**0.686 4**
Bike	0.407 6	0.401 7	0.382 6	0.410 0	0.343 3	0.486 3	0.444 6	0.303 9	**0.501 8**
Birthdaycake	0.498 9	0.365 9	0.445 3	0.495 7	0.417 4	0.495 7	0.471 0	0.421 7	**0.633 0**
sEMG_sub	0.335 0	0.338 3	0.338 3	0.365 0	0.385 0	0.351 7	0.376 7	**0.391 7**	0.370 0
arcene_sub	**0.690 0**	**0.690 0**	0.560 0	0.560 0	0.560 0	0.560 0	**0.690 0**	**0.690 0**	**0.690 0**
average	0.477 0	0.410 3	0.435 8	0.472 4	0.409 4	0.483 0	0.480 8	0.424 8	**0.539 3**

在准确率方面, DP DTSNE 在 Balloon 等 9 个数据集上取得了最佳准确率, 所有数据集的最佳准确率均出现在使用 DTSNE 进行特征学习的对照组中. 因此, 在准确率指标下, DTSNE 的效果明显优于前两者. 在 Friedman 检验中, DP DTSNE 取得了最佳平均 rank 值 8.00, χ_F^2 统计量为 38.05, χ_F^2 服从自由度为 8 的卡方分布, 由 $\chi_F^2(8)$ 计算得到对应的 p 值为 7.0×10^{-6}, 小于设定的显著性水平 0.01, 因此拒绝原假设, 接受备择假设, 即 DP DTSNE 显著优于其他对比算法.

表 3 不同聚类算法纯度对比

数据集	*k*-means	AP	DP	*k*-means TSNE	AP TSNE	DP TSNE	*k*-means DTSNE	AP DTSNE	DP DTSNE
Balloon	0.578 1	0.575 9	0.575 9	0.575 9	0.583 1	0.588 0	0.575 9	0.575 9	**0.594 0**
Beer	0.543 7	0.543 7	0.543 7	0.571 3	**0.573 6**	0.567 8	0.552 9	0.563 2	0.567 8
Airplane	0.478 7	**0.559 1**	0.458 5	0.511 1	0.519 3	0.460 8	0.458 5	0.552 0	0.518 1
Bus	**0.663 2**	0.624 2	0.624 2	0.628 6	0.633 0	0.658 2	0.630 8	0.646 2	0.660 4
Ufo	0.488 5	0.444 9	0.441 5	0.483 5	0.489 2	0.467 7	0.493 8	**0.502 8**	0.471 1
Venus	0.598 2	0.565 7	0.508 4	0.566 8	0.590 3	0.573 5	0.590 3	**0.607 2**	0.576 9
Webcam	0.486 8	0.546 8	0.451 9	0.557 0	0.564 6	0.534 2	0.562 0	**0.569 6**	0.540 5
Amber	0.722 7	0.727 3	0.580 7	0.734 1	0.706 8	0.722 7	0.731 8	**0.738 6**	0.717 0
Bike	0.510 1	0.510 1	0.510 1	0.510 1	0.514 9	0.512 5	0.510 1	**0.536 4**	0.510 1
Birthdaycake	0.679 2	0.658 8	0.596 6	0.680 3	**0.685 6**	0.665 2	0.657 7	0.680 3	0.649 1
sEMG_sub	0.336 7	0.340 0	0.338 3	0.380 0	0.385 0	0.360 0	0.378 3	**0.391 7**	0.381 7
arcene_sub	**0.690 0**	**0.690 0**	0.560 0	**0.690 0**	0.560 0	0.560 0	**0.690 0**	**0.690 0**	**0.690 0**
average	0.564 7	0.565 5	0.515 8	0.574 1	0.567 1	0.555 9	0.569 3	**0.587 8**	0.573 1

在纯度方面, AP DTSNE 在 Ufo 等 7 个数据集上取得了最佳纯度, 同时也取得了最佳平均纯度. 从整体上看, 在 12 个数据集中, 有 8 个数据集的最佳纯度出现在 DTSNE 对照组中, 且 AP 和 DP 的最佳平均纯度均出现在 DTSNE 对照组中. 因此, 在纯度指标下, DTSNE 的效果优于前两者. 在 Friedman 检验中, AP DTSNE 取得了最佳平均 rank 值 7.50, χ_F^2 统计量为 36.06, 对应的 p 值为 1.7×10^{-5}, 小于设定的显著性水平 0.01, 因此拒绝原假设, 接受备择假设, 即 AP DTSNE 显著优于其他对比算法.

值得注意的是: 在 arcene_sub 数据集上, 各算法的准确率和纯度均为 0.69 或 0.56, 但是只有 DTSNE 对照组中, 各算法的实验结果均为 0.69; 而在其他对照组中, 总是存在实验结果为 0.56 的算法. 这也从侧面说说明了 DTSNE 模型的效果.

表 4 和表 5 分别展示了不同分类算法在各实验数据集上的准确率和 *F*1-score, 同一数据集上的最佳值被加粗显示.

表 4 不同分类算法准确率对比

数据集	SVM	DT	KNN	SVM TSNE	DT TSNE	KNN TSNE	SVM DTSNE	DT DTSNE	KNN DTSNE
Balloon	0.575 9	0.537 3	0.554 2	0.596 4	0.503 6	0.566 3	**0.597 6**	0.521 7	0.533 7
Beer	0.543 7	0.525 3	0.603 4	0.605 7	0.563 2	0.597 7	0.609 2	0.574 7	**0.643 7**
Airplane	0.457 0	0.561 4	**0.645 3**	0.562 3	0.601 0	0.591 7	0.483 9	0.523 9	0.588 2
Bus	0.624 2	0.609 9	0.678 0	0.662 6	0.580 2	0.687 9	0.703 3	0.626 4	**0.725 3**
Ufo	0.421 8	0.436 9	0.453 9	0.425 4	0.441 5	0.478 9	0.438 2	0.471 9	**0.480 0**
Venus	0.533 5	0.591 4	0.636 3	0.667 9	0.608 4	0.671 1	0.671 3	0.631 9	**0.688 9**
Webcam	0.451 9	0.583 5	**0.588 6**	0.557 0	0.526 6	0.569 6	0.558 2	0.564 6	0.560 8
Amber	**0.744 3**	0.634 1	0.719 3	0.656 8	0.654 5	0.736 4	0.647 7	0.660 2	0.704 5
Bike	0.510 2	0.450 6	0.498 2	0.524 5	0.449 4	0.493 5	**0.559 5**	0.535 7	0.523 8
Birthdaycake	0.604 0	0.593 3	0.676 0	0.653 4	0.624 5	**0.695 3**	0.648 0	0.638 3	0.655 5
sEMG_sub	0.421 3	0.458 3	0.423 3	0.425 0	0.424 9	0.456 7	0.446 7	**0.463 3**	0.460 0
arcene_sub	0.790 0	0.740 0	0.820 0	0.750 0	0.720 0	0.790 0	0.740 0	0.700 0	**0.830 0**
average	0.556 5	0.560 2	0.608 1	0.590 6	0.558 1	0.611 2	0.592 0	0.576 1	**0.616 2**

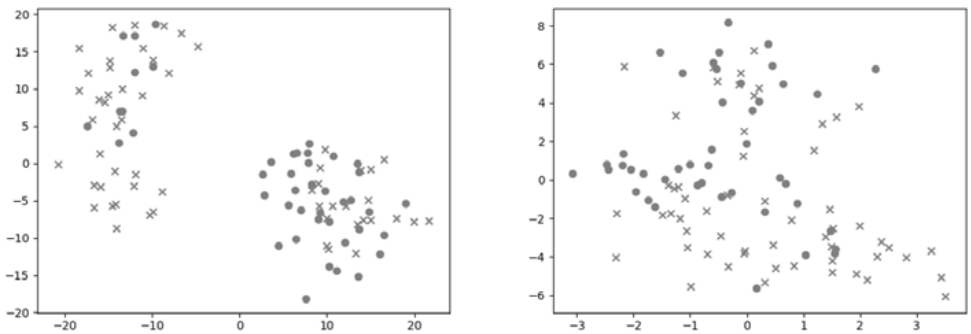
表 5 不同分类算法 *F*1-socre 对比

数据集	SVM	DT	KNN	SVM TSNE	DT TSNE	KNN TSNE	SVM DTSNE	DT DTSNE	KNN DTSNE
Balloon	0.373 1	0.425 6	0.459 4	0.396 6	0.459 7	0.447 8	0.389 5	0.451 5	**0.462 1**
Beer	0.414 4	0.493 9	0.505 5	0.447 8	0.496 3	0.521 9	0.439 0	0.502 0	**0.535 4**
Airplane	0.359 2	0.527 5	0.567 5	0.403 4	0.559 9	**0.583 2**	0.437 2	0.511 9	0.567 5
Bus	0.346 1	0.474 0	0.515 9	0.373 8	0.481 7	0.532 8	0.372 8	0.521 9	**0.539 2**
Ufo	0.351 7	0.436 1	0.435 8	0.354 3	0.445 1	0.438 7	0.348 6	**0.462 1**	0.436 1
Venus	0.560 8	0.586 3	0.653 9	0.610 2	0.597 6	0.647 2	0.613 5	0.574 5	**0.664 6**
Webcam	0.386 7	0.594 9	**0.619 1**	0.474 7	0.544 7	0.585 7	0.490 9	0.545 9	0.588 9
Amber	0.474 4	0.502 3	0.505 9	0.403 6	0.468 1	0.513 6	0.401 8	0.489 3	**0.519 0**
Bike	0.344 8	0.369 3	0.386 4	0.341 6	0.371 1	0.393 6	0.352 5	0.380 4	**0.399 6**
Birthdaycake	0.358 7	0.479 2	0.497 4	0.424 0	0.497 3	0.519 6	0.419 3	0.468 0	**0.521 1**
sEMG_sub	0.509 7	0.479 2	0.315 4	0.507 8	0.478 2	0.475 2	**0.525 9**	0.511 7	0.522 2
arcene_sub	0.735 5	0.711 6	0.760 2	0.753 5	0.688 7	0.773 8	0.754 8	0.681 4	**0.824 1**
average	0.434 6	0.506 7	0.518 5	0.457 6	0.507 4	0.536 1	0.462 1	0.508 4	**0.548 3**

在准确率方面, KNN DTSNE 在 Beer 等 5 个数据集上取得了最佳准确率, 同时也取得了最佳平均准确率. 就整体而言, 12 个数据集中, 8 个数据集的最佳准确率出现在 DTSNE 对照组中, 且 SVM, DT 与 KNN 的最佳平均准确率均出现在 DTSNE 对照组中. 因此, 在准确率指标下, DTSNE 的效果优于前两者. 在 Friedman 检验中, KNN DTSNE 取得了最佳平均 rank 值 6.90, χ_F^2 统计量为 31.80, 对应的 p 值为 1.0×10^{-4}, 小于设定的显著性水平 0.01, 因此拒绝原假设, 接受备择假设, 即 KNN DTSNE 显著优于其他对比算法.

在 $F1$-score 方面, KNN DTSNE 在 Balloon 等 8 个数据集上取得了最佳 $F1$-score, 且 SVM, DT 与 KNN 的最佳平均 $F1$-score 均出现在 DTSNE 对照组中. 因此, 在 $F1$-score 指标下, DTSNE 的效果优于前两者. 在 Friedman 检验中, KNN DTSNE 取得了最佳平均 rank 值 6.90, χ_F^2 统计量为 51.86, 对应的 p 值为 1.8×10^{-8}, 小于设定的显著性水平 0.01, 因此拒绝原假设, 接受备择假设, 即 KNN DTSNE 显著优于其他对比算法.

最后, 我们再从可视化的角度来展示 DTSNE 的效果. arcene_sub 数据集的可视化结果如图 1 所示, 从图中可以发现: 使用 DTSNE 进行可视化后, 数据呈现出了明显的簇结构, 而 TSNE 的可视化结果则相对分散, 未展现出数据中的判别信息.



(a) DTSNE 可视化结果 (b) TSNE 可视化结果

图 1 可视化结果对比

从上面的分析可以看出: 基于 DTSNE 的聚类与分类算法具有显著的优越性, 且可视化的效果也优于 TSNE, 这充分说明了 DTSNE 模型的有效性.

4 总结与展望

本文在对特征学习的研究中, 提出了一种基于随机近邻嵌入的判别性特征学习模型. 该模型将聚类算法融入到随机近邻嵌入算法特征学习的过程中, 随机近邻嵌入算法学习原始数据的拓扑结构, 聚类算法学习原始数据中的判别信息, 最终将两者融合, 得到原始数据更具判别性的特征表示. 从与主流聚类与分类算法融合对比实验的结果可以看出: 使用 DTSNE 模型进行特征学习后, 所得到的数据表示能显著提升这些算法的性能, 这充分说明了 DTSNE 模型的有效性.

DTSNE 属于流形学习模型的改进, 流形学习一般通过局部拓扑结构的保持来完成特征学习, 当引入新数据时, 局部拓扑结构已经发生改变, 因此需要重新训练, 这导致 DTSNE 的泛化性能还有待提高. 在接下来的研究中, 我们将重点研究如何提高 DTSNE 的泛化性能、如何优化 DTSNE 的时间复杂度以及如何将弱监督信息融入该模型中.

References:

[1] Ayesha S, Hanif MK, Talib R. Overview and comparative study of dimensionality reduction techniques for high dimensional data. Information Fusion, 2020, 59: 44−58.

[2] Wang H, Ahuja N. A tensor approximation approach to dimensionality reduction. Int'l Journal of Computer Vision, 2008, 76(3): 217−229.

[3] Singh A, Ganapathysubramanian B, Singh AK, *et al*. Machine learning for high-throughput stress phenotyping in plants. Trends in Plant Science, 2016, 21(2): 110−124.

[4] Saeys Y, Inza I, Larrañaga P. A review of feature selection techniques in bioinformatics. Bioinformatics, 2007, 23(19): 2507−2517.

[5] Kuo FY, Sloan IH. Lifting the curse of dimensionality. Notices of the AMS, 2005, 52(11): 1320−1328.

[6] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798−1828.

[7] Cayton L. Algorithms for manifold learning. University of California at San Diego Tech. Rep, 2005, 12(1-17): 1.

[8] Zhong G, Wang LN, Ling X, *et al*. An overview on data representation learning: From traditional feature learning to recent deep learning. The Journal of Finance and Data Science, 2016, 2(4): 265−278.

[9] Karamizadeh S, Abdullah SM, Manaf AA, *et al*. An overview of principal component analysis. Journal of Signal and Information Processing, 2013, 4(3B): 173.

[10] Pohar M, Blas M, Turk S. Comparison of logistic regression and linear discriminant analysis: A simulation study. Metodoloski Zvezki, 2004, 1(1): 143.

[11] Schölkopf B, Smola A, Müller KR. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, 1998, 10(5): 1299−1319.

[12] Tenenbaum JB, De Silva V, Langford JC. A global geometric framework for nonlinear dimensionality reduction. Science, 2000, 290(5500): 2319−2323.

[13] Jaworska N, Chupetlovska-Anastasova A. A review of multidimensional scaling (MDS) and its utility in various psychological domains. Tutorials in Quantitative Methods for Psychology, 2009, 5(1): 1−10.

[14] Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. Science, 2000, 290(5500): 2323−2326.

[15] Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation, 2003, 15(6): 1373−1396.

[16] Hinton GE, Roweis S. Stochastic neighbor embedding. Advances in Neural Information Processing Systems, 2002, 15: 857−864.

[17] Donoho DL, Grimes C. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. Proc. of the National Academy of Sciences, 2003, 100(10): 5591−5596.

[18] Lin T, Zha H. Riemannian manifold learning. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2008, 30(5): 796−809.

[19] Yu G, Peng H, Wei J, *et al*. Enhanced locality preserving projections using robust path based similarity. Neurocomputing, 2011, 74(4): 598−605.

[20] Nie F, Zhu W, Li X. Unsupervised large graph embedding. In: Proc. of the 31st AAAI Conf. on Artificial Intelligence. San Francisco: Assoc Advancement Artificial Intelligence, 2017. 2422−2428.

[21] Nie F, Dong X, Li X. Unsupervised and semisupervised projection with graph optimization. IEEE Trans. on Neural Networks and Learning Systems, 2020, 32(4): 1547−1559.

[22] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science, 2006, 313(5786): 504−507.

[23] Shao H, Jiang H, Zhao H, *et al*. A novel deep autoencoder feature learning method for rotating machinery fault diagnosis. Mechanical Systems and Signal Processing, 2017, 95: 187−204.

[24] Lee H, Pham P, Largman Y, *et al*. Unsupervised feature learning for audio classification using convolutional deep belief networks. Advances in Neural Information Processing Systems, 2009, 22: 1096−1104.

[25] Van der Maaten L, Hinton G. Visualizing data using *t*-SNE. Journal of Machine Learning Research, 2008, 9(11): 2579−2605.

[26] Qian N. On the momentum term in gradient descent learning algorithms. Neural Networks, 1999, 12(1): 145−151.

[27] Krishna K, Murty MN. Genetic *K*-means algorithm. IEEE Trans. on Systems, 1999, 29(3): 433−439.

[28] Frey BJ, Dueck D. Clustering by passing messages between data points. Science, 2007, 315(5814): 972−976.

[29] Rodriguez A, Laio A. Clustering by fast search and find of density peaks. Science, 2014, 344(6191): 1492−1496.

[30] Noble WS. What is a support vector machine. Nature Biotechnology, 2006, 24(12): 1565−1567.

[31] Safavian SR, Landgrebe D. A survey of decision tree classifier methodology. IEEE Trans. on Systems, 1991, 21(3): 660−674.

[32] Gou J, Du L, Zhang Y, *et al*. A new distance-weighted *k*-nearest neighbor classifier. Journal of Information & Computational Science, 2012, 9(6): 1429−1436.

[33] Qi MM, Xiang Y. Pairwise constraint projections inosculating sparsity preserving. Computer Science, 2012, 39(11): 212−215 (in Chinese with English abstract).

[34] Fu Z, Wang HJ, Li TR, *et al*. Weakly supervised learning framework based on *k* labeled samples. Ruan Jian Xue Bao/ Journal of Software, 2020, 31(4): 981−990 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5919.htm [doi: 10.13328/j.cnki.jos.005919]

[35] Huang S, Wang H, Li T, *et al*. Constraint co-projections for semi-supervised co-clustering. IEEE Trans. on Cybernetics, 2015, 46(12): 3047−3058.

[36] Wright MB. Speeding up the Hungarian algorithm. Computers & Operations Research, 1990, 17(1): 95−96.

附中文参考文献:

[33] 齐鸣鸣, 向阳. 融合稀疏保持的成对约束投影. 计算机科学, 2012, 39(11): 212−215.

[34] 付治, 王红军, 李天瑞, 等. 基于 *k* 个标记样本的弱监督学习框架. 软件学报, 2020, 31(4): 981−990. http://www.jos.org.cn/1000-9825/5919.htm [doi: 10.13328/j.cnki.jos.005919]

赵辉(1996－), 男, 硕士生, 主要研究领域为机器学习, 数据挖掘.

王红军(1977－), 男, 博士, 副研究员, CCF 高级会员, 主要研究领域为人工智能, 机器学习, 数据挖掘.

彭博(1980－), 女, 博士, 副教授, CCF 专业会员, 主要研究领域为图像分割, 机器学习.

龙治国(1989－), 男, 博士, 讲师, CCF 专业会员, 主要研究领域为知识表示, 空间推理, 机器学习.

李天瑞(1969－), 男, 博士, 教授, CCF 杰出会员, 主要研究领域为人工智能, 数据挖掘与知识发现, 云计算与大数据, 粒计算, 粗糙集.