

一种基于极大熵的快速无监督线性降维方法*

王继奎¹, 杨正国¹, 刘学文¹, 易纪海¹, 李冰¹, 聂飞平²

¹(兰州财经大学 信息工程学院, 甘肃 兰州 730020)

²(西北工业大学 光学影像分析与学习中心, 陕西 西安 710072)

通信作者: 聂飞平, feipingnie@gmail.com



摘要: 现实世界中高维数据无处不在, 然而在高维数据中往往存在大量的冗余和噪声信息, 这导致很多传统聚类算法在对高维数据聚类时不能获得很好的性能. 实践中发现高维数据的类簇结构往往嵌入在较低维的子空间中. 因而, 降维成为挖掘高维数据类簇结构的关键技术. 在众多降维方法中, 基于图的降维方法是研究的热点. 然而, 大部分基于图的降维算法存在以下两个问题: (1) 需要计算或者学习邻接图, 计算复杂度高; (2) 降维的过程中没有考虑降维后的用途. 针对这两个问题, 提出一种基于极大熵的快速无监督降维算法 MEDR. MEDR 算法融合线性投影和极大熵聚类模型, 通过一种有效的迭代优化算法寻找高维数据嵌入在低维子空间的潜在最优类簇结构. MEDR 算法不需事先输入邻接图, 具有样本个数的线性时间复杂度. 在真实数据集上的实验结果表明, 与传统的降维方法相比, MEDR 算法能够找到更好地将高维数据投影到低维子空间的投影矩阵, 使投影后的数据有利于聚类.

关键词: 无监督学习; 线性降维; 邻接图; 聚类; 极大熵

中图法分类号: TP18

中文引用格式: 王继奎, 杨正国, 刘学文, 易纪海, 李冰, 聂飞平. 一种基于极大熵的快速无监督线性降维方法. 软件学报, 2023, 34(4): 1779–1795. <http://www.jos.org.cn/1000-9825/6400.htm>

英文引用格式: Wang JK, Yang ZG, Liu XW, Yi JH, Li B, Nie FP. Fast Unsupervised Dimension Reduction Method Based on Maximum Entropy. Ruan Jian Xue Bao/Journal of Software, 2023, 34(4): 1779–1795 (in Chinese). <http://www.jos.org.cn/1000-9825/6400.htm>

Fast Unsupervised Dimension Reduction Method Based on Maximum Entropy

WANG Ji-Kui¹, YANG Zheng-Guo¹, LIU Xue-Wen¹, YI Ji-Hai¹, LI Bing¹, NIE Fei-Ping²

¹(College of Information Engineering, Lanzhou University of Finance and Economics, Lanzhou 730020, China)

²(Center for Optical Imagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: High-dimensional data is widely adopted in the real world. However, there is usually plenty of redundant and noisy information existing in high-dimensional data, which accounts for the poor performance of many traditional clustering algorithms when clustering high-dimensional data. In practice, it is found that the cluster structure of high-dimensional data is often embedded in the lower dimensional subspace. Therefore, dimension reduction becomes the key technology of mining high-dimensional data. Among many dimension reduction methods, graph-based method becomes a research hotspot. However, most graph-based dimension reduction algorithms suffer from the following two problems: (1) most of the graph-based dimension reduction algorithms need to calculate or learn adjacency graphs, which have high computational complexity; (2) the purpose of dimension reduction is not considered in the process of dimension reduction. To address the problem, a fast unsupervised dimension reduction algorithm is proposed based on the maximum entropy-MEDR, which combines linear projection and the maximum entropy clustering model to find the potential optimal cluster structure of high-dimensional data embedded in low-dimensional subspace through an effective iterative optimization algorithm. The MEDR algorithm does not need the adjacency graph as an input in advance, and has linear time complexity of input data scale. A large number of experimental results on real

* 基金项目: 国家自然科学基金 (61772427, 11801345); 甘肃省高等学校创新能力提升项目 (2019B-97); 兰州财经大学校级重点项目 (Lzufe2020B-0010, Lzufe2020B-011)

收稿时间: 2021-02-22; 修改时间: 2021-05-19; 采用时间: 2021-06-16; jos 在线出版时间: 2022-06-15

CNKI 网络首发时间: 2022-11-15

datasets show that the MEDR algorithm can find a better projection matrix to project high-dimensional data into low-dimensional subspace compared with the traditional dimensionality reduction method, so that the projected data is conducive to clustering analysis.

Key words: unsupervised learning; dimension reduction; adjacency graph; clustering; maximum entropy

随着科技的发展,涌现出越来越多的高维数据.传统的机器学习算法在面对高维数据时存在很多问题,比如计算复杂度高、样本采样密度过于稀疏导致距离计算困难等.然而,高维数据有意义的类簇结构往往嵌入在低维子空间中.为了寻找高维数据嵌入在低维空间的类簇结构,研究者们提出了很多降维方法.这些降维方法主要包括无监督降维、半监督降维和有监督降维 3 种.主成分分析 (principal component analysis, PCA)^[1] 是一种经典的无监督线性投影降维方法,因其速度快、适用于不同的场景而被广泛使用.但是 PCA 对异常点敏感,选择投影后数据全局方差最大的投影方向,忽视了不同类别数据的分布情况,有时候效果不佳.为克服这些问题,研究者在 PCA 的基础上提出了一系列改进算法^[2-5].随后,研究人员提出了局部保留投影算法 (locality preserving projection, LPP)^[6], LPP 算法是基于邻接图的线性降维算法,其基本思想是使降维前后数据的近邻关系得到保持.在 LPP 的基础上,研究人员进一步开发了系列算法,如 NPE^[7]、GoLPP^[8]、JGOPL^[9]、DGLPGE^[10] 和 AgLPP^[11] 等.无监督降维算法 AutoEncoder^[12] 利用编码、解码技术,最小化重构误差,使用中间隐层作为样本的抽象表示进行降维.

基于 PCA 的系列算法和基于 LPP 的系列算法都是线性降维算法,不适用于流形数据.为了挖掘流形数据嵌入在低维子空间中的类簇结构,研究人员提出了系列流形学习降维算法,比如 locally linear embedding^[13,14]、ISOMAP^[15]、RSpLPP^[16] 及 Laplacian Eigenmaps^[17] 等.因为流形学习降维算法计算复杂度高,难以适用于新增数据,所以很少应用在实际场景中.除了上述无监督降维算法,研究者们也提出了许多半监督降维算法以及许多有监督降维算法^[18-21].这些降维算法都是基于某种优化目标,旨在保持数据的某种特性,比如,PCA 要求降维后的数据方差尽可能大, LPP 要求降维前后数据的近邻关系得到保持,并没有考虑降维后数据的用途.

聚类也是机器学习领域的研究热点之一. K-means^[22] 是最经典的一个聚类算法.尽管 K-means 算法计算速度快,被广泛使用,但是其也具有若干缺点,比如对异常点敏感、鲁棒性不强、计算速度慢及仅适用于球形分布的数据等.为了解决这些问题,研究者们也进行了系列研究^[23-33].文献 [34-37] 对 K-means 算法进行扩展以完成分类属性数据、混合类型数据聚类.1948 年,Shannon^[38] 引入信息熵,一个系统越有序,信息熵就越低;反之,一个系统越混乱,信息熵就越高.所以说,可以认为信息熵是系统有序化程度的一个度量.最大熵^[39] 建模是以最大熵理论为基础的一种选择模型的方法,即从符合条件的分布中选择熵最大的分布作为最优的分布.此后,研究人员将极大熵的思想用于机器学习中.文献 [40] 提出了一种基于极大熵的聚类算法,引起了广泛关注.文献 [8] 将极大熵的思想与 LPP 结合在一起,提出了图优化的局部近邻保持投影算法.文献 [41,42] 对极大熵聚类算法的收敛性进行了证明.

传统的方法将高维数据的聚类分成降维和对降维后数据的聚类两个独立的阶段分别进行.比如,先用 PCA 进行数据降维,然后采用 K-means 算法进行聚类学习.目前也有一些研究将降维后数据的聚类信息融入降维过程中,用于指导降维^[43-50]. van de Velden 等人^[47] 指出两阶段的高维数据聚类方法不如优化一个目标函数性能好.将降维与降维后的用途结合起来设计降维算法成为新的研究思路.

以提高降维后数据的聚类性能为目标,我们提出了一种将极大熵理论与线性投影技术结合的快速无监督线性降维算法 (fast unsupervised linear dimension reduction method based on maximum entropy, MEDR). MEDR 算法将极大熵模型融合进线性降维过程中,用于监督降维过程.我们提出了一种有效的迭代优化算法进行 MEDR 模型优化.具体做法如下:固定权重关系矩阵,优化投影矩阵和类簇中心;固定投影矩阵和类簇中心,优化权重关系矩阵. MEDR 算法迭代地优化投影矩阵、类簇中心和权重关系矩阵,直至算法收敛. MEDR 算法融合了线性投影技术和极大熵模型,利用极大熵模型监督降维过程,使降维后嵌入在低维子空间的数据更适合聚类.所以, MEDR 算法具有明显的目的,使降维后的数据更适合进行聚类.

1 相关工作

1.1 局部近邻保持算法 (LPP)

LPP 是 Laplacian Eigenmaps 的一个线性扩展, LPP 的目标是使高维数据在低维子空间的投影依然保持原空间

的近邻关系. LPP 最小化以下目标函数:

$$\frac{1}{2} \sum_{i,j=1}^n a_{ij} \|y_i - y_j\|_2^2 \quad (1)$$

公式 (1) 中, $a_{ij} = \exp(-\|x_i - x_j\|_2^2 / 2\delta^2)$, δ 是平滑参数. a_{ij} 表示原空间中样本 x_i 和样本 x_j 之间的权重关系, 用 A 表示样本间的权重矩阵. $y_i = v^T x_i$, v 表示一个投影向量. a_{ij} 的定义表明原空间样本对 (x_i, x_j) 之间的距离越远, a_{ij} 值越小; 反之, 距离越近, a_{ij} 值越大. 从而, 近邻关系得到保持. LPP 目标函数可转化为:

$$\frac{1}{2} \sum_{i,j=1}^n a_{ij} \|y_i - y_j\|_2^2 = \frac{1}{2} \sum_{i,j=1}^n a_{ij} (v^T x_i - v^T x_j)^T (v^T x_i - v^T x_j) = v^T X(D - A)X^T v = v^T XLX^T v \quad (2)$$

公式 (2) 中, D 是一个对角阵, 其第 i 个对角元素 $d_{ii} = \sum_{j=1}^n a_{ij}$ 表示权重矩阵的第 i 行的行和. $L = D - A$ 表示权重矩阵 A 的拉普拉斯矩阵. LPP 用 $v^T XDX^T v = 1$ 对投影后的数据尺度进行约束. 最后, 最小化公式 (1) 的目标函数问题转变为以下有约束的最小化问题:

$$\min_v v^T XLX^T v, \text{ s.t. } v^T XDX^T v = 1 \quad (3)$$

公式 (3) 可以转化为以下广义特征向量求解问题:

$$XLX^T v = \lambda XDX^T v \quad (4)$$

公式 (4) 中, λ 表示特征值, 因此投影矩阵 W 由满足公式 (4) 的最小的 d' 个广义特征值对应的特征向量构成. 与 Laplacian Eigenmaps 相比, LPP 算法能够学到一个投影矩阵 W , 可以处理新增数据. 但是 LPP 存在两个缺点: (1) LPP 的邻接图需预先输入, 邻接图的构图质量决定了降维的性能; (2) LPP 算法的计算复杂度为 $O(n^2 d)$, 时间复杂度高.

1.2 图优化的局部近邻保持算法 (GoLPP)

为了解决 LPP 需预先输入邻接图的问题, 研究人员提出了图优化的局部近邻保持算法 (graph-optimized locality preserving projections, GoLPP). GoLPP 也是一种线性投影算法, 它结合 LPP 和极大熵理论, 可以自动学习权重矩阵 A , 其模型如下:

$$\min_{W,A} \sum_{i=1}^n \sum_{j=1}^n \frac{\|W^T x_i - W^T x_j\|_2^2}{\sum_{i=1}^n \|W^T x_i\|_2^2} a_{ij} + \lambda^{-1} a_{ij} \ln(a_{ij}), \text{ s.t. } A^i \geq 0, A^i 1_n = 1 \quad (5)$$

公式 (5) 中, A^i 表示权重矩阵 A 的第 i 行, $A^i 1_n = 1$ 表示权重矩阵第 i 行的和为 1. λ 是平衡因子, GoLPP 通过调整平衡因子 λ 的值调整权重矩阵 A 中各元素的分布. 与 LPP 相比, GoLPP 主要有两点不同: (1) GoLPP 算法可以学习权重矩阵 A , 而 LPP 算法则是事先给定权重矩阵 A ; (2) 结合极大熵理论, GoLPP 算法可以通过不同的 λ 取值调整权重矩阵 A 中各元素的分布, 使得高维数据在低维子空间的投影数据质量更好. 然而 GoLPP 算法依然需要计算样本间的两两距离, 其时间复杂度为 $O(n^2 d)$. 所以, 与其他基于图的方法一样, GoLPP 时间复杂度高.

1.3 基于锚点图的局部近邻保持投影算法 (AgLPP)

为了解决 LPP 的时间复杂度高, 不适用于大规模数据的问题, 文献 [11] 提出了一种基于锚点图的快速降维算法 (dimensionality reduction on anchorgraph with an efficient locality preserving projection, AgLPP). AgLPP 利用锚点图的权重矩阵 Z 估算原数据集的权重矩阵 A . 锚点图的权重矩阵 Z 计算复杂度为 $O(ndm)$, m 表示锚点的个数, 远低于权重矩阵 A 的计算复杂度 $O(n^2 d)$. AgLPP 具体做法如下.

- (1) 利用一种聚类算法获得 m 个锚点集 B , b_k 表示其第 k 个元素.
- (2) 利用以下公式计算锚点图权重矩阵:

$$z_{ik} = \frac{K_\delta(x_i, b_k)}{\sum_{j=1}^{c'} K_\delta(x_i, b_j)} \quad (6)$$

公式 (6) 中, $K_\delta(x_i, b_k) = \exp(-\|x_i - b_k\|_2^2 / 2\delta^2)$, δ 是平滑参数, c' 表示近邻锚点的个数.

(3) 与 LPP 类似, 利用以下公式求解投影向量:

$$XLX^T v = \lambda XD X^T v \quad (7)$$

公式 (7) 中, $D = Z^T Z$, $L = Z^T Z - Z^T Z \Lambda^{-1} Z^T Z$. Λ 是一个对角阵, 其第 k 个对角元素 $\Lambda_{kk} = \sum_{i=1}^n z_{ik}$. AgLPP 算法的计算复杂度为 $O(nmdt + d^3)$, t 表示迭代次数. AgLPP 算法具有样本个数 n 的线性时间复杂度. 然而, AgLPP 需预先计算锚点图, 锚点图的构图质量决定了 AgLPP 算法的性能.

2 基于极大熵的快速无监督线性降维方法

为了解决 LPP 算法面临的需预先输入邻接图和计算复杂度高的问题, 我们将 GoLPP 和 AgLPP 的思想结合起来, 提出了一种基于极大熵的快速线性降维算法 MEDR. 我们将高维数据嵌入在低维子空间中的虚拟类簇中心作为锚点, 结合极大熵理论, 改变降维后数据的分布, 使 MEDR 算法降维后的数据更适用于聚类. MEDR 降维算法解决了经典 LPP 算法需预先输入邻接图和计算复杂度高的两个问题, 同时使降维后的数据具有更好的类簇结构.

2.1 模型

基于以上分析, 我们提出了如下基于极大熵的快速线性降维模型.

$$\min_{W, M, P} \sum_{i=1}^n \sum_{k=1}^c \|W^T x_i - m_k\|_2^2 p_{ik} + \gamma^{-1} p_{ik} \ln(p_{ik}), \text{ s.t. } W^T S_t W = I, P^i \geq 0, P^i 1_c = 1, \|P^i\|_0 = K \quad (8)$$

公式 (8) 中, n 表示样本的个数, c 表示类簇的个数, $W \in \mathbb{R}^{d \times d'}$ 表示线性投影矩阵, $M = \{m_1, \dots, m_c\} \in \mathbb{R}^{d' \times c}$ 表示降维后的类簇中心, 用 m_k 表示第 k 个类簇中心, $P \in \mathbb{R}^{n \times c}$ 表示样本与类簇中心的权重矩阵, P^i 表示权重矩阵 P 第 i 行, p_{ik} 表示 P^i 的第 k 元素, S_t 表示数据集 X 中心化后的协方差矩阵. 约束 $W^T S_t W = I$ 将线性投影后的各维度数据约束在一定尺度内, 并保证降维后各维度数据统计不相关. 约束 $P^i \geq 0, P^i 1_c = 1$ 使得每个权重非负, 并且行和为 1. $\|P^i\|_0 = K$ 是稀疏约束, 用来约束 P^i 中不为零的元素个数. 公式 (8) 引入负熵项 (极大熵项) $\sum_{i=1}^n \sum_{k=1}^c p_{ik} \ln(p_{ik})$, 用于刻画样本点与类簇中心的权重分布. $\gamma \in (0, \infty)$, 是平衡因子, 用于调节公式 (8) 中极大熵项的重要程度. 当 γ 的取值较大时, γ^{-1} 较小, 公式 (8) 中的第 1 项起主要作用, 公式 (8) 倾向于将样本分配给离其最近的类簇中心, 样本点与离其最近的类簇中心的权值逼近 1. 当 γ 的取值较小时, γ^{-1} 较大, 公式 (8) 中的第 2 项起主要作用, 公式 (8) 将所有类簇中心靠拢, 从而使样本点隶属于不同类簇中心的权重几乎相等. 所以, 通过调整 γ 的取值, 可以改变类簇中心的分布, 寻找最优的投影矩阵, 从而使降维后的数据有更好的类簇结构.

MEDR 降维算法与 GoLPP 等基于图的降维算法形式上很像, 但与它们相比有两点关键不同: (1) MEDR 算法只需计算样本与类簇中心的距离, 时间复杂度为 $O(ndc)$. 而 LPP、GoLPP 等基于图的算法需预先计算邻接图, 时间复杂度为 $O(n^2 d)$. 与 LPP、GoLPP 等基于图的算法相比, MEDR 算法的计算复杂度大幅降低. 而且, MEDR 算法可以动态的学习权重矩阵 P , 不需预先输入邻接图. (2) 由于 MEDR 算法在降维的过程中融合了基于极大熵的聚类模型, 使 MEDR 算法具有明确的目的, 降维后的数据更有利于聚类.

2.2 模型优化

公式 (8) 可以采用迭代的方法求解, 具体来说, 我们先固定一组变量, 优化另一组变量, 迭代的进行优化, 直至算法收敛. 公式 (8) 的优化分两步进行:

步骤 1. 固定 P , 优化 W, M . 固定 P , 公式 (8) 简化为以下模型:

$$\min_{W, M} \sum_{i=1}^n \sum_{k=1}^c \|W^T x_i - m_k\|_2^2 p_{ik}, \text{ s.t. } W^T S_t W = I \quad (9)$$

公式 (9) 关于 m_k 求导并令导数为 0 得:

$$\frac{\partial \sum_{i=1}^n \sum_{k=1}^c p_{ik} \|W^T x_i - m_k\|_2^2}{\partial m_k} = 2 \sum_{i=1}^n p_{ik} W^T x_i - 2 \sum_{i=1}^n p_{ik} m_k = 0 \quad (10)$$

由公式 (10) 可以得到 m_k 的最优解为:

$$m_k = \sum_{i=1}^n W^T x_i p_{ik} / \sum_{i=1}^n p_{ik} = W^T \sum_{i=1}^n x_i p_{ik} / \sum_{i=1}^n p_{ik} = W^T y_k \quad (11)$$

其中, $y_k = \sum_{i=1}^n x_i p_{ik} / \sum_{i=1}^n p_{ik}$, 显然, y_k 表示原空间的类簇中心. 将 $m_k = W^T y_k$ 代入公式 (9) 得:

$$\min_W \sum_{i=1}^n \sum_{k=1}^c p_{ik} \|W^T x_i - W^T y_k\|_2^2, \text{ s.t. } W^T S_t W = I \quad (12)$$

为了得到 W 的最优解, 我们提出以下引理.

引理 1. 令 D^x 为 $n \times n$ 对角阵, 其第 i 个对角元素为 $D_{ii}^x = \sum_{k=1}^c p_{ik}$, $i \in [1, n]$, 令 D^y 为 $c \times c$ 的对角阵, 其第 k 个对角元素为: $D_{kk}^y = \sum_{i=1}^n p_{ki}$, $k \in [1, c]$. 令 $Q = XD^x X^T - 2XPY^T + YD^y Y^T$, $H = (Q + Q^T)/2$, 则有 $\sum_{i=1}^n \sum_{k=1}^c p_{ik} \|W^T x_i - W^T y_k\|_2^2 = \text{tr}[W^T H W]$.

引理 1 的证明见附录 1. 所以公式 (9) 中 W 的最优解由满足方程 $H\alpha = \lambda S_t \alpha$ 的最小的 d' 个广义特征值对应的特征向量给出.

步骤 2. 固定 W, M , 优化 P , 公式 (8) 简化为以下模型:

$$\min_P \sum_{i=1}^n \sum_{k=1}^c d_{ik} p_{ik} + \gamma^{-1} p_{ik} \ln(p_{ik}), \text{ s.t. } P^i \geq 0, P^i 1_c = 1, \|P^i\|_0 = K \quad (13)$$

其中, $d_{ik} = \|W^T x_i - m_k\|_2^2$. 因为约束发生在每 1 行, 则公式 (13) 可以分解为以下 n 个独立的子问题分别求解, 第 i 个子问题表述为:

$$\min_{P^i} \sum_{k=1}^c d_{ik} p_{ik} + \gamma^{-1} p_{ik} \ln(p_{ik}), \text{ s.t. } P^i \geq 0, P^i 1_c = 1, \|P^i\|_0 = K \quad (14)$$

为求解以上模型, 暂时不考虑 $P^i \geq 0, \|P^i\|_0 = K$ 约束条件, 定义以下拉格朗日函数:

$$J(P^i, \alpha_i) = \sum_{k=1}^c d_{ik} p_{ik} + \gamma^{-1} p_{ik} \ln(p_{ik}) + \alpha_i \left(\sum_{j=1}^c p_{ij} - 1 \right)$$

令 p_{ik}^* 表示 p_{ik} 的最优解, α_i^* 表示 α_i 的最优解, 则:

$$\frac{\partial J(P^i, \alpha_i)}{\partial p_{ik}^*} = d_{ik} + \gamma^{-1} (1 + \ln p_{ik}^*) + \alpha_i^* = 0 \quad (15)$$

$$\frac{\partial J(P^i, \alpha_i)}{\partial \alpha_i^*} = \sum_{j=1}^c p_{ij}^* - 1 = 0 \quad (16)$$

由公式 (16) 可得:

$$p_{ik}^* = e^{-\gamma(d_{ik} + \alpha_i^*) - 1} = e^{-\gamma\alpha_i^* - 1} e^{-\gamma d_{ik}} \quad (17)$$

将公式 (17) 代入公式 (16) 可得:

$$\sum_{j=1}^c e^{-\gamma(d_{ij} + \alpha_i^*) - 1} = e^{-\gamma\alpha_i^* - 1} \sum_{j=1}^c e^{-\gamma d_{ij}} = 1 \quad (18)$$

由公式 (18) 可得:

$$e^{-\gamma\alpha_i^* - 1} = 1 / \sum_{j=1}^c e^{-\gamma d_{ij}} \quad (19)$$

将公式 (19) 代入公式 (17) 可得:

$$p_{ik}^* = e^{-\gamma d_{ik}} / \sum_{j=1}^c e^{-\gamma d_{ij}} \quad (20)$$

显然 $p_{ik}^* > 0$, 满足 $P^i \geq 0$ 约束条件, 公式 (13) 不考虑 $\|P^i\|_0 = K$ 约束的最优值为:

$$\begin{aligned} \sum_{i=1}^n \sum_{k=1}^c p_{ik}^* d_{ik} + \gamma^{-1} \sum_{i=1}^n \sum_{k=1}^c p_{ik}^* \ln p_{ik}^* &= \sum_{i=1}^n \sum_{k=1}^c p_{ik}^* (d_{ik} + \gamma^{-1} \ln p_{ik}^*) = \sum_{i=1}^n \sum_{k=1}^c \left(e^{-\gamma d_{ik}} / \sum_{j=1}^c e^{-\gamma d_{ij}} \right) \left(-\gamma^{-1} \ln \left(\sum_{j=1}^c e^{-\gamma d_{ij}} \right) \right) \\ &= -\gamma^{-1} \sum_{i=1}^n \sum_{k=1}^c \left(e^{-\gamma d_{ik}} / \sum_{j=1}^c e^{-\gamma d_{ij}} \right) \left(\ln \left(\sum_{j=1}^c e^{-\gamma d_{ij}} \right) \right) = -\gamma^{-1} \sum_{i=1}^n \left(\ln \left(\sum_{j=1}^c e^{-\gamma d_{ij}} \right) \right) \sum_{k=1}^c \left(e^{-\gamma d_{ik}} / \sum_{j=1}^c e^{-\gamma d_{ij}} \right) \\ &= -\gamma^{-1} \sum_{i=1}^n \left(\ln \left(\sum_{j=1}^c e^{-\gamma d_{ij}} \right) \right) \left(\sum_{k=1}^c (e^{-\gamma d_{ik}}) / \sum_{j=1}^c e^{-\gamma d_{ij}} \right) = -\gamma^{-1} \sum_{i=1}^n \left(\ln \left(\sum_{j=1}^c e^{-\gamma d_{ij}} \right) \right). \end{aligned}$$

由以上推导过程可得:

$$\sum_{i=1}^n \sum_{k=1}^c p_{ik}^* d_{ik} + \gamma^{-1} \sum_{i=1}^n \sum_{k=1}^c p_{ik}^* \ln p_{ik}^* = -\gamma^{-1} \sum_{i=1}^n \left(\ln \sum_{k=1}^c e^{-\gamma d_{ik}} \right) \quad (21)$$

从公式 (21) 可以看出, d_{ik} 越小, 最优值越小. 所以选择最小的前 K 个 d_{ik} 计算 p_{ik} , 其余的 p_{ik} 置为 0, 可得公式 (13) 的最优解. 我们通过分析的方法求出了公式 (13) 的全局最优解.

通过以上分析, MEDR 算法描述如算法 1.

算法 1. MEDR 算法.

输入: 数据集 X , 类簇数 c ;

参数: 降维后数据的维度 d' , 平衡因子 γ , 稀疏约束 K ;

输出: 投影矩阵 W , 权重矩阵 P .

1. 随机初始化 P , 使其满足 $P^i \geq 0, P^i 1_c = 1, \|P^i\|_0 = K$
 2. WHILE 不收敛 DO
 3. 计算原始空间聚类中心 $y_k = \sum_{i=1}^n x_i p_{ik} / \sum_{i=1}^n p_{ik}$
 4. 计算 $D_{ii}^x = \sum_{j=1}^c p_{ij}, D_{kk}^y = \sum_{i=1}^n p_{ki}$
 5. 计算 $Q = XD^x X^T - 2XPY^T + YD^y Y^T, H = (Q + Q^T)/2$
 6. 更新 W , 其最优解由满足方程 $H\alpha = \lambda S_1 \alpha$ 的最小的 d' 个特征值对应的特征向量给出
 7. 根据公式 (11) 更新 m_k
 8. 计算 $d_{ik} = \|W^T x_i - m_k\|_2^2$
 9. 选择最小的前 K 个 d_{ik} 计算 p_{ik} , 利用公式 (20) 更新 p_{ik} , 其余的 p_{ik} 置为 0
 10. END WHILE
 11. 输出 W, P
-

2.3 时间和空间复杂度分析

令 n 表示输入数据的规模, c 表示类簇数, d 表示数据的维度. MEDR 算法在一次迭代中, 由于 D^x 是对角矩阵, 仅需计算其 n 个非零的对角元素, 所以计算 XD^x 的复杂度为 $O(nd)$; 故计算 Q 的时间复杂度为 $O(nd^2)$. 更新 W 的时间复杂度为 $O(d^3)$, 更新 M 的时间复杂度为 $O(ndc)$; 更新 P 的时间复杂度为 $O(ndc)$. 设迭代次数为 t , 在 $d \ll n, c \ll n$ 的假设下, 整个算法的时间复杂度为 $O(nd^2 t)$. 由此可见, MEDR 算法具有样本 n 的线性时间复杂度. MEDR 的时间复杂度远小于基于图的 LPP、ISOMAP、NPE 和 GoLPP.

PCA 算法的空间复杂度为 $O(nd)$; AutoEncoder 算法的空间复杂度为 $O(nd)$; 由于需要计算样本间的两两距离, 所以, LPP、ISOMAP、LLE、NPE 和 GoLPP 算法的空间复杂度均为 $O(n^2)$; AgLPP 算法的空间复杂度为 $O(nm)$, m 表示锚点的数量; MEDR 算法的空间复杂度为 $O(nc)$, 由于 c 表示类簇的个数, 其值远小于 n 和 m 的值, 所以 MEDR 的空间复杂度比较低。

2.4 收敛性分析

令第 t 次迭代的最优解为 $W^{(t)}, M^{(t)}, P^{(t)}$, 模型的最优值为 $J(W^{(t)}, M^{(t)}, P^{(t)})$. MEDR 算法每次迭代包括两个步骤。

第 1 步, 固定 $P^{(t)}$, 优化 W 、 M , 得到其最优解 $W^{(t+1)}, M^{(t+1)}$. 由文献 [51] 可知:

$$J(W^{(t)}, M^{(t)}, P^{(t)}) \geq J(W^{(t+1)}, M^{(t+1)}, P^{(t)}) \quad (22)$$

第 2 步, 固定 $W^{(t+1)}, M^{(t+1)}$, 优化 P . 由公式 (21) 可知: $J(W^{(t+1)}, M^{(t+1)}, P^{(t+1)}) = -\gamma^{-1} \sum_{i=1}^n (\ln \sum_{k=1}^c e^{-\gamma d_{ik}})$, 上式表明, 最优值 $J(W^{(t+1)}, M^{(t+1)}, P^{(t+1)})$ 是 d_{ik} 的单调递增函数. 令 h_i 表示在稀疏约束 K 的情况下最小的前 K 个 d_{ik} 对应的位置, 令 h_{ik} 表示 h_i 的第 k 个元素. 选择 $d_{i,h_{ik}}$ 计算 $p_{i,h_{ik}}$, 则最优值最小, 最优值为 $-\gamma^{-1} \sum_{i=1}^n \ln \sum_{k=1}^K (e^{-\gamma d_{i,h_{ik}}})$. 令 $ind(p_i^{(t)})$ 表示在第 t 次迭代求得的第 i 子问题的最优解 $p_i^{(t)}$ 不为零的位置. 用 ind_k 表示 $ind(p_i^{(t)})$ 第 k 个元素. 根据公式 (21) 可知, $-\gamma^{-1} \sum_{i=1}^n \ln \sum_{k=1}^K (e^{-\gamma d_{i,ind_k}})$ 是 $W^{(t+1)}, M^{(t+1)}$ 和 $ind(p_i^{(t)})$ 固定后的最优解. 所以:

$$J(W^{(t+1)}, M^{(t+1)}, P^{(t)}) = -\gamma^{-1} \sum_{i=1}^n \left(\ln \sum_{k=1}^K e^{-\gamma d_{i,ind_k}} \right) \geq -\gamma^{-1} \sum_{i=1}^n \left(\ln \sum_{k=1}^K e^{-\gamma d_{i,h_{ik}}} \right) = J(W^{(t+1)}, M^{(t+1)}, P^{(t+1)}) \quad (23)$$

联合公式 (22)、公式 (23) 可得:

$$J(W^{(t+1)}, M^{(t+1)}, P^{(t+1)}) \leq J(W^{(t)}, M^{(t)}, P^{(t)}) \quad (24)$$

因为公式 (8) 中 $\sum_{i=1}^n \sum_{k=1}^c \|W^T x_i - m_k\|_2^2 p_{ik}$ 项大于 0, $\sum_{i=1}^n \sum_{k=1}^c \gamma^{-1} p_{ik} \ln(p_{ik})$ 项的下界为 $-(n/\gamma) \ln c$, 所以公式 (8) 的下界为 $-(n/\gamma) \ln c$. 依据公式 (24) 可知, MEDR 算法是收敛的。

3 实验结果与分析

3.1 实验环境

实验环境为 Win7 操作系统、2.7 GHz AMD A12-9800B R7 CPU、Matlab 2012a. 实验使用的 PCA、LPP、ISOMAP、LLE 和 NPE 算法实现均来自 Matlab 的 drtoolbox 库, AutoEncoder 采用 Matlab 官网上的实现, GoLPP^[8] 和 AgLPP^[11] 算法根据原文实现。

3.2 实验

3.2.1 数据集

我们选取 Australian、Cars、Cleve、Diabetes、German、Glass、UPS、ORL、Yale 和 Palm 等 10 个 UCI 基准数据集 (<http://archive.ics.uci.edu/ml/datasets.php>) 进行实验. 数据集的基本信息如表 1 所示。

表 1 数据集信息表

数据集	样本数	维度	类簇数	数据集	样本数	维度	类簇数
Australian	690	14	2	Glass	214	9	6
Cars	392	8	3	Yale	165	1024	15
Cleve	303	13	4	ORL	400	1024	40
Diabetes	768	8	2	UPS	1854	256	10
German	1000	20	2	Palm	2000	256	100

以上 10 个数据集都是经典的用于降维、聚类研究的数据集. 对于大于 100 维的数据集, 先利用 PCA 降到 100 维. 对比的降维方法包括 PCA、LPP、ISOMAP、AutoEncoder、LLE、NPE、GoLPP 和 AgLPP. 利用 K-

means 算法对降维后的数据进行聚类. 选取常用的准确度 (accuracy)、互信息 (NMI) 和 F 分数 (F-score) 作为聚类性能评价指标, accuracy 代表聚类的准确度, 其取值范围为 [0, 1], F-score 是召回率和准确率的加权调和平均的倒数, 其取值范围为 [0, 1], NMI 即归一化互信息, 用来衡量两个数据集分布的吻合程度, 其取值范围为 [0, 1], 上述 3 种评测指标都是值越大, 说明聚类性能越好, 这 3 种指数的具体计算方法可参考文献 [52]. 准确度、互信息和 F 分数 3 种指标可以对平衡和不平衡数据集的聚类性能进行有效度量.

3.2.2 运行参数设置

在实验中各算法的参数设置如下, 对于所有算法 $d' \in 1, \dots, \min(d, 100)$, 其中 d 表示原始空间数据的维度, 对于维度大于 100 的数据集, 先用 PCA 降到 100 维. 对于 LPP、ISOMAP、LLE、NPE、AgLPP 和 MEDR 算法中 $k \in \{2, \dots, 12\}$. LPP 算法中的 $\sigma \in \{2e-0, 2e+2, 2e+4, 2e+6\}$, AgLPP 算法中的 $\sigma \in \{5e-3, 5e-2, 5e-1, 1, 5e+1\}$, AgLPP 算法中的 $m \in \{n/10, n/5\}$, MEDR 算法中的 $\gamma \in \{100, 300, 400, 500, 600, 1000\}$.

3.2.3 实验结果与分析

对于 PCA、LPP、ISOMAP、AutoEncoder、LLE、NPE、GoLPP 和 AgLPP 算法, K-means 算法运行 50 次, 取聚类平均性能最好的结果进行比较. MEDR 算法由于学到了类簇中心, 仅需运行一次 K-means 算法. 实验结果如表 2、表 3 所示, 采用 mean±std(最佳维度) 的形式描述. 表中粗体表示最好的聚类性能指标数据.

表 2 各算法在 Australian、Cars、Cleve、Diabetes 和 German 数据集上的聚类性能

指标	数据集	Australian	Cars	Cleve	Diabetes	German
准确率 (accuracy)	PCA	56.20±0.01(9)	65.05±0.02(1)	67.26±0.12(1)	66.09±0.08(1)	70.00±0.00(1)
	LPP	69.68±0.30(13)	68.29±0.11(5)	68.09±0.15(10)	72.4±0.52(1)	70.05±0.00(16)
	AutoEncoder	84.49±0.26(2)	70.23±0.11(6)	70.69±0.01(12)	65.1±0.41(1)	70.00±0.00(1)
	ISOMAP	56.20±0.21(6)	67.4±0.02(3)	60.4±0.36(1)	65.23±0.31(1)	70.00±0.00(1)
	LLE	69.23±0.37(1)	69.16±0.05(2)	64.39±0.13(4)	67.04±0.04(6)	70.27±0.00(7)
	NPE	72.20±1.02(12)	66.91±0.03(6)	68.42±0.21(9)	74.36±0.11(1)	70.00±0.00(1)
	GoLPP	60.17±0.24(2)	67.91±0.09(7)	62.74±0.19(3)	71.74±0.22(7)	70.00±0.00(1)
	AgLPP	68.55±0.32(1)	69.39±0.16(5)	65.35±0.15(5)	72.92±0.14(1)	70.12±0.00(8)
	MEDR	85.53(1)	73.72(2)	69.93(6)	73.03(2)	70.40(3)
互信息 (NMI)	PCA	3.35±0.06(9)	21.00±0.12(6)	15.35±0.03(2)	3.01±0.02(1)	1.24±0.00(12)
	LPP	17.31±0.88(13)	27.46±0.35(5)	17.38±0.28(13)	11.30±0.03(1)	3.10±0.00(20)
	AutoEncoder	40.50±0.29(2)	26.52±0.10(7)	13.36±0.01(12)	1.09±0.05(1)	2.03±0.00(8)
	ISOMAP	3.35±0.00(6)	21.93±0.01(3)	6.44±0.02(1)	3.00±0.02(3)	1.28±0.00(1)
	LLE	10.56±0.00(1)	26.14±0.14(3)	9.04±0.11(8)	3.85±0.13(6)	2.04±0.00(1)
	NPE	22.17±0.33(12)	27.31±0.63(6)	19.47±0.02(11)	14.35±0.00(1)	3.67±0.05(13)
	GoLPP	6.73±0.10(2)	15.39±0.03(4)	10.97±0.03(3)	12.04±0.08(7)	2.99±0.00(4)
	AgLPP	10.88±0.02(5)	24.36±0.12(8)	9.74±0.03(5)	12.20±0.06(6)	2.21±0.00(5)
	MEDR	42.79(1)	33.63(2)	20.83(3)	12.13(4)	9.32(1)
F分数 (F-score)	PCA	66.94±0.14(6)	51.14±0.23(6)	64.02±0.09(2)	64.36±0.32(1)	67.05±0.00(9)
	LPP	70.68±0.27(13)	63.20±0.63(7)	61.06±0.48(13)	72.12±0.10(1)	67.09±0.00(1)
	AutoEncoder	84.53±0.02(2)	68.25±0.29(8)	62.22±0.13(8)	60.42±0.23(6)	65.96±0.17(17)
	ISOMAP	66.94±0.23(1)	59.78±0.35(1)	61.62±0.71(1)	69.39±0.31(1)	71.46±0.00(1)
	LLE	68.47±0.00(1)	63.58±0.46(7)	62.22±0.16(1)	69.31±0.13(8)	71.29±0.00(16)
	NPE	72.83±0.72(12)	62.93±0.25(6)	61.50±0.14(9)	74.26±0.18(1)	67.15±0.00(2)
	GoLPP	66.99±0.35(2)	55.22±0.31(5)	56.06±0.11(11)	71.89±0.52(7)	68.51±0.00(17)
	AgLPP	68.51±0.57(1)	64.69±0.83(1)	63.49±0.72(2)	72.74±0.62(1)	68.41±0.00(7)
	MEDR	85.57(1)	67.75(4)	64.04(2)	72.67(2)	69.88(2)

表 2 的实验结果表明, 我们提出的 MEDR 算法在 Australian、Cars、German 这 3 个数据集获得了最好的 accuracy 值, 在 Australian、Cars、Cleve 和 German 这 4 个数据集上获得最好的 NMI 值, 在 Australian 和 Cleve 两

个数据集上获得了最好的 F-score 值. 因为 Cars 和 Cleve 为非平衡数据集, 所以 3 个评价指标的度量结果不一致, MEDR 算法在 Cars 数据集上取得了最好的 accuracy 和 NMI 值, 而 AutoEncoder 取得了最好的 F-score 值.

表 3 各算法在 Glass、UPS、ORL、Yale 和 Palm 数据集上的聚类性能

指标	数据集	Glass	UPS	ORL	Yale	Palm
准确率 (accuracy)	PCA	58.79±0.30(1)	72.80±0.07(89)	59.83±0.04(47)	42.79±0.04(8)	77.22±0.02(73)
	LPP	60.42±0.20(3)	76.83±0.09(8)	71.90±0.05(24)	48.30±0.01(15)	86.41±0.08(29)
	AutoEncoder	62.38±0.10(1)	69.67±0.04(68)	58.87±0.04(73)	41.21±0.25(23)	72.51±0.04(92)
	ISOMAP	57.71±0.24(2)	81.71±0.26(74)	61.43±0.01(18)	44.67±0.08(57)	5.95±0.02(1)
	LLE	55.84±0.11(6)	75.96±0.16(4)	63.73±0.04(26)	50.06±0.15(17)	55.33±0.02(62)
	NPE	59.11±0.26(3)	74.75±0.08(11)	71.08±0.02(17)	49.39±0.04(13)	85.14±0.04(30)
	GoLPP	50.23±0.18(5)	47.81±0.05(100)	46.10±0.01(100)	23.45±0.01(100)	79.66±0.01(100)
	AgLPP	56.50±0.32(6)	75.12±0.09(56)	50.83±0.03(28)	43.33±0.15(68)	73.03±0.12(57)
	MEDR	71.03(5)	76.70(18)	74.25(22)	50.91(15)	88.30(53)
互信息 (NMI)	PCA	42.01±0.17(9)	62.01±0.01(23)	76.27±0.01(47)	48.89±0.03(33)	91.31±0.70(90)
	LPP	40.10±0.15(3)	68.97±0.03(6)	83.88±0.01(27)	53.09±0.02(15)	96.07±0.62(40)
	AutoEncoder	39.33±0.10(6)	60.50±0.03(68)	75.67±0.03(73)	45.89±0.06(23)	88.58±0.61(92)
	ISOMAP	42.68±0.16(2)	75.57±0.14(37)	76.82±0.02(20)	50.37±0.01(17)	19.65±0.31(1)
	LLE	44.47±0.05(2)	73.63±0.01(4)	79.36±0.01(26)	53.11±0.07(20)	76.81±0.25(62)
	NPE	40.77±0.13(4)	66.29±0.01(11)	83.63±0.01(17)	52.87±0.01(13)	95.48±0.26(30)
	GoLPP	23.67±0.06(8)	41.58±0.05(100)	61.46±0.01(100)	24.40±0.02(1)	93.07±0.29(100)
	AgLPP	35.64±0.02(4)	66.50±0.21(56)	69.43±0.22(28)	47.66±0.15(66)	86.52±0.31(57)
	MEDR	60.32(5)	68.57(18)	84.20(30)	55.56(16)	96.21(62)
F分数 (F-score)	PCA	57.20±0.01(2)	69.71±0.10(23)	59.07±0.09(53)	46.27±0.06(8)	75.88±0.02(89)
	LPP	56.53±0.07(3)	74.37±0.04(11)	69.98±0.04(26)	52.73±0.01(15)	85.90±0.08(35)
	AutoEncoder	57.85±0.12(1)	67.15±0.13(68)	57.56±0.06(73)	44.03±0.13(23)	71.45±0.05(92)
	ISOMAP	58.00±0.04(2)	80.37±0.50(74)	60.84±0.03(18)	48.15±0.06(17)	2.45±0.03(1)
	LLE	56.69±0.01(2)	77.40±0.05(4)	62.49±0.06(26)	53.61±0.12(20)	53.67±0.01(62)
	NPE	57.42±0.01(3)	71.78±0.05(11)	69.26±0.02(17)	52.90±0.01(13)	84.54±0.03(30)
	GoLPP	48.55±0.17(7)	48.25±0.01(100)	45.28±0.00(100)	23.28±0.01(100)	78.94±0.31(100)
	AgLPP	54.67±0.23(6)	72.48±0.32(56)	50.04±0.46(28)	45.88±0.33(21)	71.63±0.18(57)
	MEDR	65.84(8)	73.75(18)	73.84(16)	52.24 (19)	87.41(42)

尽管 MEDR 算法在 Cleve 数据集上没有取得最好的 accuracy 值, 但取得了最好的 NMI 值和 F-score 值. 在上述 5 个数据集上, MEDR 算法的聚类性能最好, AutoEncoder 算法次之, PCA 算法最差. 对 German 数据集由于其第 5 个维度为 Credit amount, 其取值范围、方差明显大于其余维度, 所以数据集起作用的主要是第 5 个维度, 其余维度的数据被忽略, 所以各算法的聚类性能差不多.

各对比算法在 Glass、ORL、Yale 和 Palm 数据集的实验结果由表 3 给出, 从表 3 中的数据可以看出, MEDR 算法获得了最好的 accuracy 值和 NMI 值, 在 Glass、Yale 和 Palm 数据集上获得最好的 F-score 值. 在上述数据集上, MEDR 算法的聚类性能最好, ISOMAP 算法次之, GoLPP 算法最差. 表 2 和表 3 中的数据表明, MEDR 在 10 个基准数据集集中的 7 个数据集上获得了最好的 accuracy 值, 在 8 个数据集上获得了最好的 NMI 值, 在 6 个数据集上获得了最好的 F-score 值.

表 2、表 3 中的实验结果数据表明对大多数数据集, 其有意义的类簇结构存在于维度较低的低维子空间中, 实验结果证明了 MEDR 算法在降维的过程中融合极大熵模型, 潜在地完成了在低维子空间的聚类. MEDR 算法学习到的投影矩阵, 可将原始空间的数据投影到具有最优类簇结构的子空间中.

3.3 数据可视化

数据可视化是降维技术的一项重要应用. 我们选择 UCI 测试数据库 (<http://archive.ics.uci.edu/ml/datasets.php>) 中的 Wine 基准数据集进行实验, Wine 数据集包含 178 个样本, 维度为 10, 分为 3 类, 其中类一有 59 个样本, 类二

有 71 个样本, 类三有 48 个样本. 对比的算法包括 PCA、LPP、ISOMAP、AutoEncoder、LLE、NPE、GoLPP 和 AgLPP. 各算法参数设置信息如下: 对于 Breast 数据集, GoLPP 的参数 $\eta = 10^{-10}$, AgLPP 的参数 $m = 140$ 、 $\delta = 0.5$. LPP 的参数 $\delta = 200$, LPP、NPE、GoLPP、AgLPP、ISOMAP 和 LLE 构造邻接图的参数 k 均设置为 12. MEDR 的参数 $\gamma = 1000$, K 取类簇个数. 各算法的二维可视化实验结果如图 1 所示.

图 1 中用空心圆表示类一, 五角星表示类二, 星号表示类三. 图 1 表明 MEDR 几乎将 Wine 数据集的 3 类数据完全分开, 可视化效果十分理想. 利用 K-means 算法对图 1 中的数据进行聚类, MEDR 的聚类准确率为 97.53%. 对于 Wine 数据集, 与排名第 2 的 AutoEncoder 和 AgLPP 相比, MEDR 的聚类准确率提升了 25.62%.

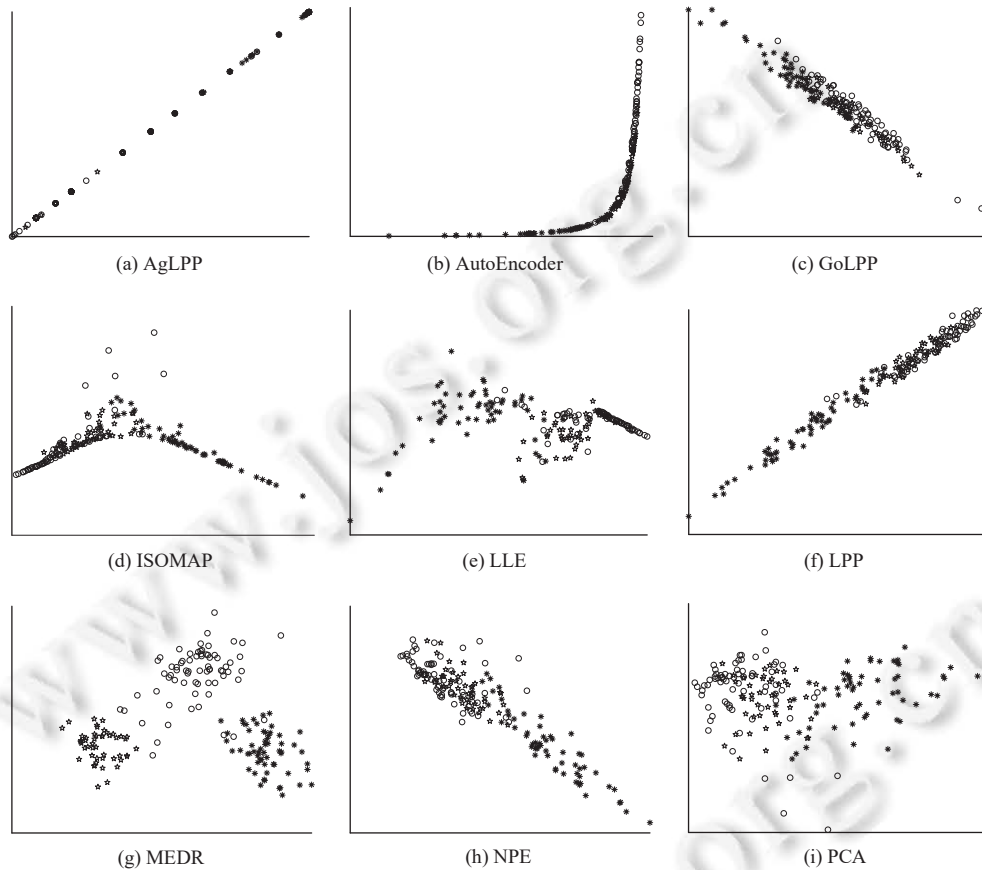


图 1 各算法在 Wine 数据集上的二维可视化结果

3.3.1 各算法运行时间实验

MEDR 具有样本个数线性的时间复杂度. PCA、LPP、ISOMAP、AgLPP、LLE 和 NPE 是非迭代算法, GoLPP、AutoEncoder 和 MEDR 是迭代算法. 表 4 列出了各算法在实验数据集上的运行时间.

在实现中, 迭代次数往往远小于数据规模. 所以, 在时间复杂度实验中, GoLPP、AutoEncoder 和 MEDR 选取一次迭代的平均时间进行比较. 实验在 Australian、Cars、Cleve、Diabetes、German、Glass、UPS、ORL、Yale 和 Palm 等 10 个基准数据集进行, 统一将数据降到 2 维, 各算法从参数取值范围取一组参数运行.

因为 Cars、Cleve、Diabetes 和 Glass 数据集较小, PCA 运行速度很快, 所以用“—”表示. 从表 4 中可以看出, PCA 的运行速度最快. AgLPP 利用锚点近似表示样本间的相似度矩阵, 运行速度也比较快. LPP、NPE 和 GoLPP 随着数据规模的增大, 运行时间明显高于 MEDR 算法. LLE 算法要保持局部线性重构关系, 运行速度也比较慢. ISOMAP 由于要计算最短路径等, 其时间复杂度为 $O(n^3)$, 是所有对比算法中最高的. 实验结果表明 MEDR 具有样本个数的线性时间复杂度, 运行速度明显快于 LPP、ISOMAP、AutoEncoder、LLE、NPE、GoLPP 和 AgLPP.

表 4 各算法在实验数据集上的运行时间比较 (ms)

数据集	Australian	Cars	Cleve	Diabetes	German	Glass	UPS	ORL	Yale	Palm
PCA	28.13	—	—	—	7.81	—	15.63	10.94	6.25	17.19
LPP	125.95	43.79	39.71	124.74	178.73	24.00	764.28	116.97	28.82	738.32
ISOMAP	6050.35	2017.36	1116.32	8142.36	13652.78	564.24	80557.29	1947.92	388.89	58923.61
AutoEncoder	757.81	295.31	307.81	214.06	454.69	262.50	406.25	345.31	265.63	534.38
LLE	302.60	178.30	100.69	443.23	507.64	91.49	1964.06	128.99	56.94	1793.92
NPE	137.85	102.26	62.50	240.28	208.85	59.55	1128.65	70.31	34.20	891.32
GoLPP	98.05	39.45	25.00	113.28	220.70	15.23	960.55	64.84	23.05	913.67
AgLPP	44.28	12.13	11.29	23.81	33.90	9.55	137.58	31.21	23.71	133.88
MEDR	30.51	3.60	2.00	9.60	16.40	1.10	34.00	22.20	16.86	95.22

3.4 参数研究

MEDR 算法涉及降维后的数据维度 d' 、极大熵项平衡因子 γ 和稀疏约束 K 这 3 个超参数, 参数研究实验中采用固定步长搜索的方式寻找 d' 、 γ 和 K . d' 的变化范围为 $[1, d]$, 搜索步长为 1; γ 值的变化范围依据经验设置为 $[100, 300, 400, 500, 600, 1\ 000]$; K 的取值范围设置为 $[2, c]$, 搜索步长为 1, c 表示类簇中心的个数.

(1) γ 值变化对聚类准确度的影响

不同的 γ 值反映了虚拟类簇中心在低维子空间的不同分布, γ 值越小, 则极大熵项对模型的影响越大, 则类簇中心越集中; 反之, 类簇中心越分散. 我们选择能在 10 个基准数据集上获得最优聚类结果的子空间维度进行分析, 图 2 描述了 MEDR 算法在基准数据集上的 F 分数、互信息和聚类准确度随不同的 γ 值在最优子空间的变化情况.

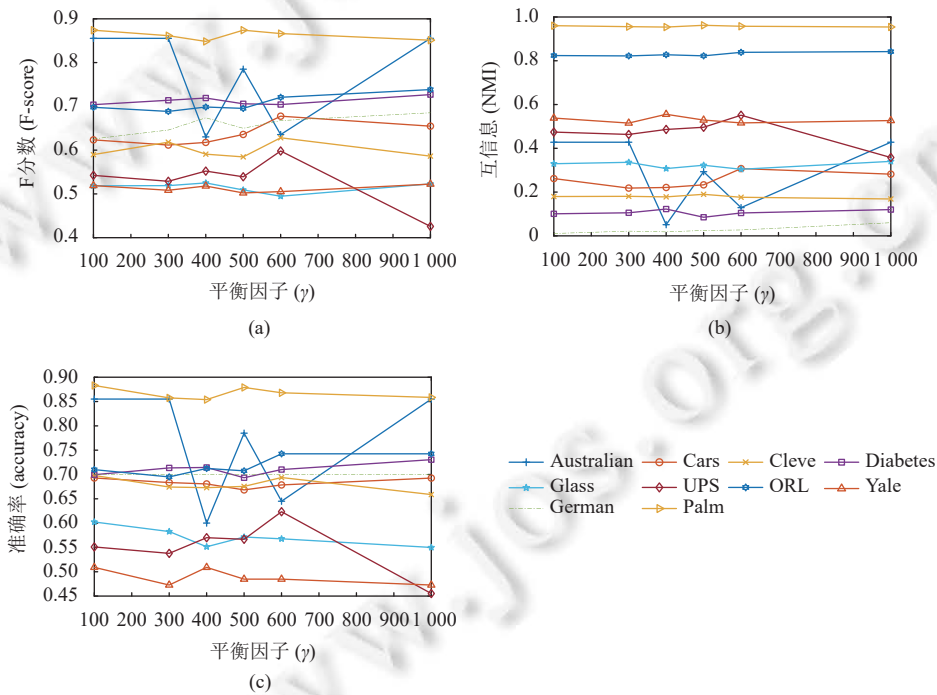


图 2 F 分数、互信息和聚类准确度随 γ 值的变化情况

从图 2 可以看出, γ 的取值对 F-score、NMI 和 accuracy 影响很大, 并且没有明显的相关性. 对于测试数据集, $\gamma = 100$ 时, 降维后的数据可取得较好的聚类性能. γ 的取值对 Palm 数据集降维后的数据的聚类性能影响十分明显, $\gamma = 100$ 时, 聚类性能最好, $\gamma = 400$ 时, 聚类性能最差. 对于 Glass 数据集, 当 $\gamma = 600$ 时聚类性能最佳, $\gamma = 1000$

时聚类性能很低. 除了 Palm 和 Glass 数据集, 其余数据集对 γ 的取值比较鲁棒. 通过调整 γ 的取值, MEDR 算法可以获得很高的聚类性能, 从而找到最优的投影矩阵将高维数据嵌入到低维子空间中.

(2) d' 的取值对聚类准确度的影响

不同的 d' 值对应不同的最优投影矩阵, 也对应着不同的投影数据. 因而, F 分数、互信息和聚类准确度会随着 d' 值的变化而变化. 对于不同的数据集, 其最优的类簇结构存在于不同的低维子空间中, 图 3 显示了 F 分数、互信息和聚类准确度随 d' 值变化的情况.

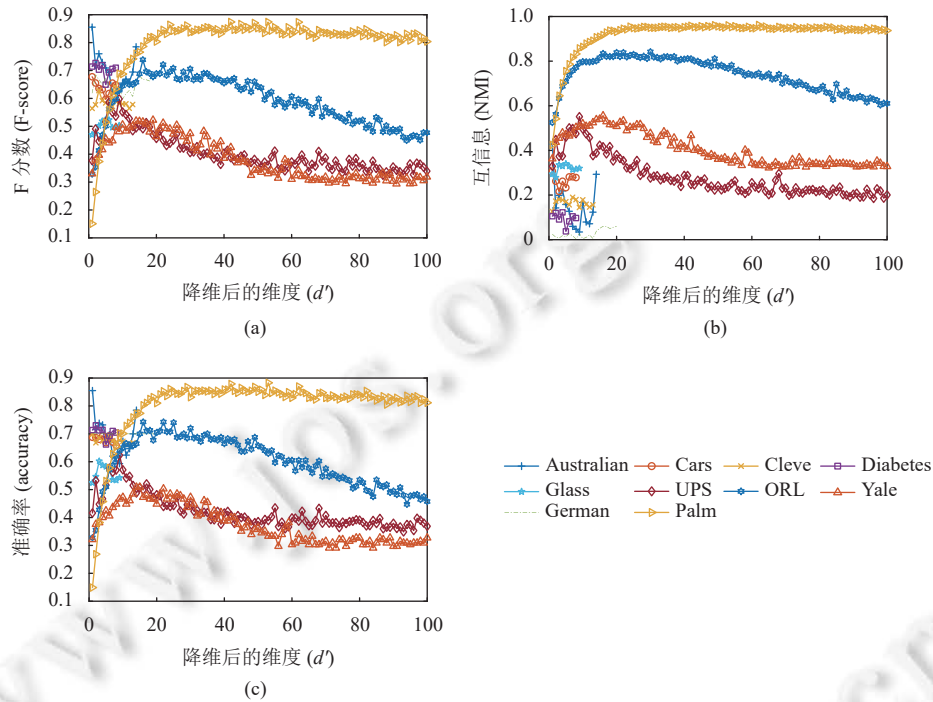


图 3 F 分数、互信息和聚类准确度随 d' 值的变化情况

从图 3 可以看出, 从维度上看 MEDR 算法找到了 Australian、Cars、Cleve、Diabetes、German、Glass、UPS、ORL、Yale 和 Palm 数据集的最优聚类性能所在的低维子空间的维度, 分别为 3、1、2、1、1、2、18、22、20 和 53. 在上述子空间中, K-means 的 accuracy 分别达到了 85.65%、71.38%、61.93%、73.05%、67.57%、56.05%、76.70%、74.25%、52.12% 和 88.30%. 与原空间数据的 accuracy 相比, MEDR 算法在子空间的 accuracy 分别提高了 29.24%、26.48%、17.05%、7.29%、0.37%、1.84%、10.36%、14.25% 和 42.24%. F-score、NMI 的变化情况与 Accuracy 基本一致. 实验结果表明, 高维数据的本质类簇结构往往嵌入在低维子空间中.

(3) 稀疏约束 K 的取值对聚类准确度的影响

K 的取值反映了对样本点与类簇中心之间权矩阵 P 的稀疏程度, K 值越小, 表示权矩阵 P 越稀疏. 图 4 展示了 K 的取值对 MEDR 算法聚类性能的影响. Palm 数据集的 $K \in [2, 33]$.

图 4 表明, 对于类簇数大于 5 的数据集, K 值取 5 时 F 分数、互信息和聚类准确度的值最大, 聚类性能最佳. 实验结果说明, 较稀疏的权重图, 可以获得更好的聚类性能. 通过设置恰当的稀疏约束 K 值, 可以显著地提升 MEDR 算法的聚类性能.

3.5 收敛速度

MEDR 算法在 10 个基准数据集上的迭代运行 100 次, 参数 γ 取值 1000, 降维后的维度为 2. 我们记录每次迭代的目标函数值, 并采用 Min-Max 的方式进行归一化. 图 5 展示了 MEDR 算法在各个数据集上的收敛过程.

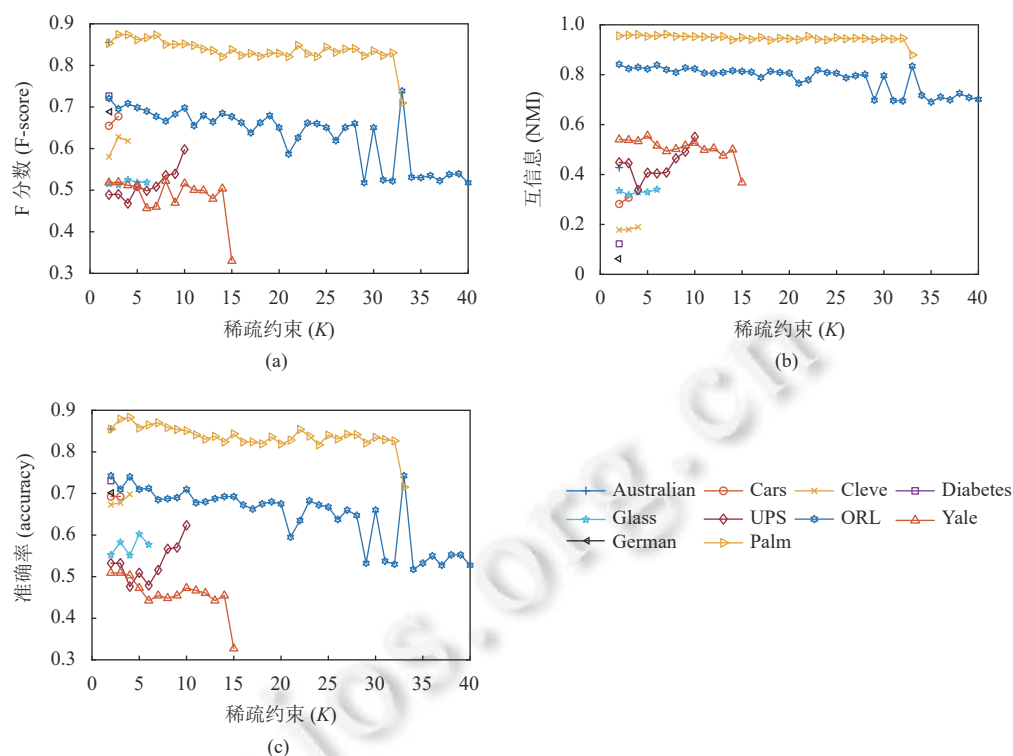
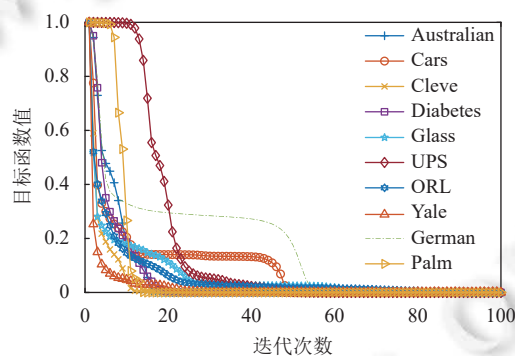
图4 F分数、互信息和聚类准确度随 K 的取值的变化情况

图5 MEDR 算法在 10 个基准数据集的收敛曲线

由图5可以看出, MEDR 算法在 Australian、Cleve、Diabetes、Yale 和 Palm 数据集上, 运行 20 次左右就收敛了. 在 Cars、German、UPS 数据集上运行不到 60 次就收敛了. 在 Glass、ORL 数据集上运行不到 80 次就收敛了, 所以 MEDR 算法收敛速度很快.

4 结束语

针对常用的基于图的降维算法需预先构建或者学习邻接图, 计算复杂度高, 而且降维过程中没有考虑降维后数据的用途等问题, 我们提出了一种基于极大熵的快速线性降维算法 MEDR. MEDR 算法将线性降维过程与聚类模型融合在一起, 使降维过程与聚类过程相互监督. 因为在降维的过程中融合了极大熵聚类模型, 所以降维后的数据具有良好的类簇结构. 我们对权重矩阵 P 施加稀疏约束来提升降维后数据的聚类性能. 大量基准数据集上的实验结果表明, MEDR 算法具有线性的时间复杂度, 具有很好的二维可视化效果, 提升了降维后数据的聚类性能. 因

为 MEDR 算法学习到了一个最优的投影矩阵, 所以 MEDR 算法还可以方便地处理新增数据.

References:

- [1] Hotelling H. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 1933, 24(7): 498–520. [doi: [10.1037/h0070888](https://doi.org/10.1037/h0070888)]
- [2] Ren JE, Li XG, Haupt J. Robust PCA via tensor outlier pursuit. In: *Proc. of the 50th Asilomar Conf. on Signals, Systems and Computers*. Pacific Grove: IEEE, 2016. 1744–1749. [doi: [10.1109/ACSSC.2016.7869681](https://doi.org/10.1109/ACSSC.2016.7869681)]
- [3] Feng DC, Chen F, Xu WL. Learning robust principal components from L1-norm maximization. *Journal of Zhejiang University SCIENCE C*, 2012, 13(12): 901–908. [doi: [10.1631/jzus.C1200180](https://doi.org/10.1631/jzus.C1200180)]
- [4] Ji HJ, Huang S. Kernel entropy component analysis with nongreedy L1-norm maximization. *Computational Intelligence and Neuroscience*, 2018, 2018: 6791683. [doi: [10.1155/2018/6791683](https://doi.org/10.1155/2018/6791683)]
- [5] Kokiopoulou E, Saad Y. Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2007, 29(12): 2143–2156. [doi: [10.1109/TPAMI.2007.1131](https://doi.org/10.1109/TPAMI.2007.1131)]
- [6] He XF, Niyogi P. Locality preserving projections. In: *Proc. of the 16th Int'l Conf. on Neural Information Processing Systems*. Vancouver: MIT Press, 2003. 153–160. [doi: [10.5555/2981345.2981365](https://doi.org/10.5555/2981345.2981365)]
- [7] He XF, Cai D, Yan SC, Zhang HJ. Neighborhood preserving embedding. In: *Proc. of the 10th IEEE Int'l Conf. on Computer Vision*. Beijing: IEEE, 2005. 1208–1213. [doi: [10.1109/ICCV.2005.167](https://doi.org/10.1109/ICCV.2005.167)]
- [8] Zhang LM, Qiao LS, Chen SC. Graph-optimized locality preserving projections. *Pattern Recognition*, 2010, 43(6): 1993–2002. [doi: [10.1016/j.patcog.2009.12.022](https://doi.org/10.1016/j.patcog.2009.12.022)]
- [9] Yi YG, Wang JZ, Zhou W, Fang YM, Kong J, Lu YH. Joint graph optimization and projection learning for dimensionality reduction. *Pattern Recognition*, 2019, 92: 258–273. [doi: [10.1016/j.patcog.2019.03.024](https://doi.org/10.1016/j.patcog.2019.03.024)]
- [10] Gou JP, Yang YY, Yi Z, Lv JC, Mao QR, Zhan Y. Discriminative globality and locality preserving graph embedding for dimensionality reduction. *Expert Systems with Applications*, 2019, 144: 113079. [doi: [10.1016/j.eswa.2019.113079](https://doi.org/10.1016/j.eswa.2019.113079)]
- [11] Jiang R, Fu WJ, Wen L, Hao SJ, Hong RC. Dimensionality reduction on anchorgraph with an efficient locality preserving projection. *Neurocomputing*, 2016, 187: 109–118. [doi: [10.1016/j.neucom.2015.07.128](https://doi.org/10.1016/j.neucom.2015.07.128)]
- [12] Hinton GE, Salakhut DRR. Reducing the dimensionality of data with neural networks. *Science*, 2006, 313(5786): 504–507. [doi: [10.1126/science.1127647](https://doi.org/10.1126/science.1127647)]
- [13] Saul LK, Roweis ST. Fit locally: Unsupervised learning of nonlinear manifolds. *Journal of Machine Learning Research*, 2003, 4: 119–155. [doi: [10.1162/153244304322972667](https://doi.org/10.1162/153244304322972667)]
- [14] Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 2000, 290(5500): 2323–2326. [doi: [10.1126/science.290.5500.2323](https://doi.org/10.1126/science.290.5500.2323)]
- [15] Tenenbaum JB, De Silva V, Langford JC. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000, 290(5500): 2319–2323. [doi: [10.1126/science.290.5500.2319](https://doi.org/10.1126/science.290.5500.2319)]
- [16] Zheng ZL, Huang XQ, Jia J, Yang J. Locality preserving projection with sparse penalty. *Chinese Journal of Computers*, 2014, 37(9): 2038–2046 (in Chinese with English abstract). [doi: [10.3724/SP.J.1016.2014.02038](https://doi.org/10.3724/SP.J.1016.2014.02038)]
- [17] Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 2003, 15(6): 1373–1396. [doi: [10.1162/089976603321780317](https://doi.org/10.1162/089976603321780317)]
- [18] Wang S, Nie FP, Chang XJ, Li X, Sheng QZ, Yao LA. Uncovering locally discriminative structure for feature analysis. In: *Proc. of the 2016 Joint European Conf. on Machine Learning and Knowledge Discovery in Databases*. Riva del Garda: Springer, 2016. 281–295. [doi: [10.1007/978-3-319-46128-1_18](https://doi.org/10.1007/978-3-319-46128-1_18)]
- [19] Zhang XW, Chu DL. Sparse uncorrelated linear discriminant analysis. In: *Proc. of the 30th Int'l Conf. on Machine Learning*. Atlanta: JMLR.org, 2013. 45–52. [doi: [10.5555/3042817.3042824](https://doi.org/10.5555/3042817.3042824)]
- [20] Ye JP, Xiong T. Computational and theoretical analysis of null space and orthogonal linear discriminant analysis. *Journal of Machine Learning Research*, 2006, 7(43): 1183–1204. [doi: [10.5555/1248547.1248590](https://doi.org/10.5555/1248547.1248590)]
- [21] Zheng WS, Lai JH, Yuen PC. GA-Fisher: A new LDA-based face recognition algorithm with selection of principal components. *IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2005, 35(5): 1065–1078. [doi: [10.1109/TSMCB.2005.850175](https://doi.org/10.1109/TSMCB.2005.850175)]
- [22] MacQueen J. Some methods for classification and analysis of multivariate observations. In: *Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability*. Berkeley: University of California, 1967. 281–297.
- [23] Bradley PS, Mangasarian OL, Street WN. Clustering via concave minimization. In: *Proc. of the 9th Int'l Conf. on Neural Information Processing Systems*. Denver: MIT Press, 1997. 368–374. [doi: [10.5555/2998981.2999033](https://doi.org/10.5555/2998981.2999033)]

- [24] Bezdek JC, Coray C, Gunderson R, Watson J. Detection and characterization of cluster substructure I. Linear structure: Fuzzy c-lines. *SIAM Journal on Applied Mathematics*, 1981, 40(2): 339–357. [doi: 10.1137/0140029]
- [25] Arthur D, Vassilvitskii S. K-Means++: The advantages of careful seeding. In: *Proc. of the 18th Annual ACM-SIAM Symp. on Discrete Algorithms*. New Orleans: SIAM, 2007. 1027–1035. [doi: 10.5555/1283383.1283494]
- [26] Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY. An efficient K-means clustering algorithm: Analysis and implementation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2002, 24(7): 881–892. [doi: 10.1109/TPAMI.2002.1017616]
- [27] Frahling G, Sohler C. A fast K-means implementation using coresets. *Int'l Journal of Computational Geometry & Applications*, 2008, 18(6): 605–625. [doi: 10.1142/S0218195908002787]
- [28] Elkan C. Using the triangle inequality to accelerate k-means. In: *Proc. of the 20th Int'l Conf. on Machine Learning*. Washington: AAAI Press, 2003. 147–153. [doi: 10.5555/3041838.3041857]
- [29] Xia SY, Peng DW, Meng DY, Zhang CQ, Wang GY, Glem E, Wei W, Chen ZZ. Ball k-means: Fast adaptive clustering with no bounds. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2022, 44(1): 87–99. [doi: 10.1109/TPAMI.2020.3008694]
- [30] Giffon L, Emiya V, Kadri H, Ralaivola L. QuickK-means: Accelerating inference for K-means by learning fast transforms. *Machine Learning*, 2021, 110(5): 881–905. [doi: 10.1007/s10994-021-05965-0]
- [31] Hicks SC, Liu RX, Ni YW, Purdom E, Risso D. Mbmeans: Fast clustering for single cell data using mini-batch K-means. *PLoS Computational Biology*, 2021, 17(1): e1008625. [doi: 10.1371/journal.pcbi.1008625]
- [32] 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]
- [33] Hu SZ, Lou ZZ, Wang RB, Yan XQ, Ye YD. Dual-weighted multi-view clustering. *Chinese Journal of Computers*, 2020, 43(9): 1708–1720 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2020.01708]
- [34] Huang ZX. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 1998, 2(3): 283–304. [doi: 10.1023/A:1009769707641]
- [35] Huang ZX, Ng MK. A fuzzy k-modes algorithm for clustering categorical data. *IEEE Trans. on Fuzzy Systems*, 1999, 7(4): 446–452. [doi: 10.1109/91.784206]
- [36] Huang JZ, Ng MK, Rong HQ, Li ZC. Automated variable weighting in k-means type clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2005, 27(5): 657–668. [doi: 10.1109/TPAMI.2005.95]
- [37] Li MJ, Ng MK, Cheung YM, Huang JZ. Agglomerative fuzzy k-means clustering algorithm with selection of number of clusters. *IEEE Trans. on Knowledge and Data Engineering*, 2008, 20(11): 1519–1534. [doi: 10.1109/TKDE.2008.88]
- [38] Shannon CE. A mathematical theory of communication. *The Bell System Technical Journal*, 1948, 27(3): 379–423. [doi: 10.1002/j.1538-7305.1948.tb01338.x]
- [39] Jaynes ET. Information theory and statistical mechanics. II. *Physical Review*, 1957, 108: 171–190. [doi: 10.1103/PhysRev.108.171]
- [40] Miyamoto S, Mukaidono M. Fuzzy c-means as a regularization and maximum entropy approach. In: *Proc. of the 7th Int'l Fuzzy Systems Association World Congress*. Prague: University of Economics, 1997. 86–92.
- [41] Zhang ZH, Zheng NN, Shi G. Maximum-entropy clustering algorithm and its global convergence analysis. *Science in China Series E: Technological Sciences*, 2001, 44(1): 89–101 (in Chinese with English abstract). [doi: 10.3969/j.issn.1674-7259.2001.01.009]
- [42] Ren SJ, Wang YD. A proof of the convergence theorem of maximum-entropy clustering algorithm. *Science China Information Sciences*, 2010, 53(6): 1151–1158 (in Chinese with English abstract). [doi: 10.1360/zf2010-40-4-583]
- [43] Yamamoto M, Hwang H. A general formulation of cluster analysis with dimension reduction and subspace separation. *Behaviormetrika*, 2014, 41(1): 115–129. [doi: 10.2333/bhmk.41.115]
- [44] De Soete G, Carroll JD. K-means clustering in a low-dimensional Euclidean space. In: Diday E, Lechevallier Y, Schader M, Bertrand P, Burtshy B, eds. *New Approaches in Classification and Data Analysis*. Berlin: Springer, 1994. 212–219. [doi: 10.1007/978-3-642-51175-2_24]
- [45] Yamamoto M, Hwang H. Dimension-reduced clustering of functional data via subspace separation. *Journal of Classification*, 2017, 34(2): 294–326. [doi: 10.1007/s00357-017-9232-z]
- [46] Zhou J, Pedrycz W, Yue XD, Gao C, Lai ZH, Wan J. Projected fuzzy C-means clustering with locality preservation. *Pattern Recognition*, 2020, 113: 107748. [doi: 10.1016/j.patcog.2020.107748]
- [47] van de Velden M, D'Enza AI, Yamamoto M. Special feature: Dimension reduction and cluster analysis. *Behaviormetrika*, 2019, 46(2): 239–241. [doi: 10.1007/s41237-019-00092-6]
- [48] Wang JK, Shi QF, Yang ZG, Nei FP. Clustering by unified principal component analysis and fuzzy C-means with sparsity constraint. In:

- Proc. of the 20th Int'l Conf. on Algorithms and Architectures for Parallel Processing. New York City: Springer, 2020. 337–351. [doi: 10.1007/978-3-030-60239-0_23]
- [49] Wang R, Nie FP, Wang Z, Hu HJ, Li XL. Parameter-free weighted multi-view projected clustering with structured graph learning. IEEE Trans. on Knowledge and Data Engineering, 2020, 32(10): 2014–2025. [doi: 10.1109/TKDE.2019.2913377]
- [50] Ran RS, Ren YS, Zhang SG, Fang B. A novel discriminant locality preserving projections method. Journal of Mathematical Imaging and Vision, 2021, 63(5): 541–554. [doi: 10.1007/s10851-020-01008-w]
- [51] Bezdek JC, Hathaway RJ. Convergence of alternating optimization. Neural, Parallel & Scientific Computations, 2003, 11(4): 351–368. [doi: 10.5555/964885.964886]
- [52] Huang XH, Ye YM, Guo HF, Cai Y, Zhang HJ, Li Y. DSKmeans: A new kmeans-type approach to discriminative subspace clustering. Knowledge-based Systems, 2014, 70: 293–300. [doi: 10.1016/j.knosys.2014.07.009]

附中文参考文献:

- [16] 郑忠龙, 黄小巧, 贾润, 杨杰. 稀疏局部保持投影. 计算机学报, 2014, 37(9): 2038–2046. [doi: 10.3724/SP.J.1016.2014.02038]
- [32] 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]
- [33] 胡世哲, 姜铮铮, 王若彬, 闫小强, 叶阳东. 一种双重加权的多视角聚类方法. 计算机学报, 2020, 43(9): 1708–1720. [doi: 10.11897/SP.J.1016.2020.01708]
- [41] 张志华, 郑南宁, 史罡. 极大熵聚类算法及其全局收敛性分析. 中国科学 (E辑), 2001, 44(1): 89–101. [doi: 10.3969/j.issn.1674-7259.2001.01.009]
- [42] 任世军, 王亚东. 极大熵聚类算法的收敛性定理证明. 中国科学:信息科学, 2010, 53(6): 1151–1158. [doi: 10.1360/zf2010-40-4-583]

附录 1

证明: 令 $U = W^T X$, $F = W^T Y$, 用 u_i 和 f_k 分别表示 U 和 F 的第 i 个和第 k 个列向量, 则:

$$\begin{aligned} \sum_{i=1}^n \sum_{k=1}^c p_{ik} \|W^T x_i - W^T y_k\|_2^2 &= \sum_{i=1}^n \sum_{k=1}^c p_{ik} \|u_i - f_k\|_2^2 = \sum_{i=1}^n \sum_{k=1}^c p_{ik} (u_i - f_k)^T (u_i - f_k) \\ &= \sum_{i=1}^n \sum_{k=1}^c p_{ik} (u_i^T u_i - 2f_k^T u_i + f_k^T f_k) = \sum_{i=1}^n d_{ii}^x u_i^T u_i + \sum_{k=1}^c d_{kk}^y f_k^T f_k - 2 \sum_{i=1}^n \sum_{k=1}^c p_{ik} f_k^T u_i \end{aligned} \quad (25)$$

显然, 公式 (25) 中的第 1 项和第 2 项用迹的形式表示为:

$$\sum_{i=1}^n d_{ii}^x u_i^T u_i = \text{tr}[D^x U^T X U] = \text{tr}[X U D^x X U^T], \sum_{k=1}^c d_{kk}^y f_k^T f_k = \text{tr}[D^y F^T F] = \text{tr}[F D^y F^T] \quad (26)$$

令 e_i 表示第 i 个规范向量, 可将公式 (25) 中的第 3 项转化为:

$$\sum_{i=1}^n \sum_{k=1}^c p_{ik} f_k^T u_i = \sum_{i=1}^n (F e_i)^T \sum_{k=1}^c p_{ik} u_k = \sum_{i=1}^n (F e_i)^T (U P) e_i = \text{tr}[F^T (U P)] = \text{tr}[U P F^T] \quad (27)$$

结合公式 (25)–公式 (27) 可知:

$$\sum_{i=1}^n \sum_{k=1}^c p_{ik} \|W^T x_i - W^T y_k\|_2^2 = \sum_{i=1}^n \sum_{k=1}^c p_{ik} \|u_i - f_k\|_2^2 = \text{tr}[W^T Q W] = \text{tr}[W^T ((Q + Q^T)/2) W] = \text{tr}[W^T H W]$$

证毕.



王继奎(1978—), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为机器学习, 人工智能.



易纪海(1974—), 男, 讲师, 主要研究领域为机器学习, 人工智能.



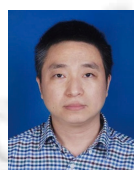
杨正国(1987—), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为机器学习, 人工智能.



李冰(1997—), 女, 硕士生, 主要研究领域为机器学习, 人工智能.



刘学文(1996—), 男, 硕士生, CCF 学生会员, 主要研究领域为机器学习, 人工智能.



聂飞平(1977—), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为机器学习, 人工智能.