

基于Lp范数的非负矩阵分解并行优化算法

黄路路, 唐舒宇, 张伟, 代祥光

引用本文

黄路路, 唐舒宇, 张伟, 代祥光. 基于Lp范数的非负矩阵分解并行优化算法[J]. 计算机科学, 2024, 51(2): 100-106.

HUANG Lulu, TANG Shuyu, ZHANG Wei, DAI Xiangguang. Non-negative Matrix Factorization Parallel Optimization Algorithm Based on Lp-norm [J]. Computer Science, 2024, 51(2): 100-106.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

基于组合结构的逻辑回归点击预测算法

Logical Regression Click Prediction Algorithm Based on Combination Structure
计算机科学, 2024, 51(2): 73-78. <https://doi.org/10.11896/jsjcx.230100052>

基于对比学习的时间序列聚类方法

Time Series Clustering Method Based on Contrastive Learning
计算机科学, 2024, 51(2): 63-72. <https://doi.org/10.11896/jsjcx.221200038>

基于异构特征融合的多维时间序列分类算法

Multivariate Time Series Classification Algorithm Based on Heterogeneous Feature Fusion
计算机科学, 2024, 51(2): 36-46. <https://doi.org/10.11896/jsjcx.230100135>

联合ZINB模型与图注意力自编码器的自优化单细胞聚类

Self-optimized Single Cell Clustering Using ZINB Model and Graph Attention Autoencoder
计算机科学, 2023, 50(12): 104-112. <https://doi.org/10.11896/jsjcx.221000167>

基于时间聚类和用户动态相似度的自适应位置推荐算法

Adaptive Location Recommendation Based on Time Slots Clustering and User Dynamic Similarity
计算机科学, 2023, 50(12): 113-122. <https://doi.org/10.11896/jsjcx.230200105>

基于 L_p 范数的非负矩阵分解并行优化算法

黄路路¹ 唐舒宇² 张伟³ 代祥光³

1 重庆三峡学院电子与信息工程学院 重庆 404100

2 重庆三峡学院计算机科学与工程学院 重庆 404100

3 重庆三峡学院智能信息处理与控制重庆高校市级重点实验室 重庆 404100

(huangllu0@163.com)

摘要 非负矩阵分解算法可以从高维数据中提取出低维和稀疏的有用信息,是处理图像聚类、数据压缩和特征提取等问题的重要手段。传统非负矩阵分解算法大多采用欧几里得距离来度量重构误差,尽管其在许多任务中已经显示出有效性,但在解决实际应用问题时仍面临着聚类效果欠佳、收敛速度慢、稳定性较差等问题。为解决这些问题,文中采用 L_p 范数作为非负矩阵分解的损失函数,通过调节系数 p 来获得更好的聚类结果。基于协同优化理论和 Majorization-Minimization 算法,使用粒子群优化算法来并行求解基于 L_p 范数的非负矩阵分解问题,并在多个真实数据集上验证了所提方法的可行性和有效性。实验结果表明所提算法明显提升了程序的执行效率且一系列评价指标均优于传统非负矩阵分解算法。

关键词: 非负矩阵分解; L_p 范数; 聚类; 并行优化; 收敛速度

中图分类号 TP391

Non-negative Matrix Factorization Parallel Optimization Algorithm Based on L_p -norm

HUANG Lulu¹, TANG Shuyu², ZHANG Wei³ and DAI Xiangguang³

1 School of Electronics and Information Engineering, Chongqing Three Gorges University, Chongqing 404100, China

2 School of Computer Science and Engineering, Chongqing Three Gorges University, Chongqing 404100, China

3 Key Laboratory of Intelligent Information Processing and Control, Chongqing Three Gorges University, Wanzhou, Chongqing 404100, China

Abstract Non-negative matrix factorization algorithm is an important tool for image clustering, data compression and feature extraction. Traditional non-negative matrix factorization algorithms mostly use Euclidean distance to measure reconstruction error, which has shown its effectiveness in many tasks, but still has the problems of suboptimal clustering results and slow convergence. To solve these problems, the loss function of non-negative matrix factorization is reconstructed by L_p -norm to obtain better clustering results by adjusting the coefficient p . Based on the collaborative optimization theory and Majorization-Minimization algorithm, this paper uses the particle swarm optimization to solve the non-negative matrix factorization problem of reconstruction in parallel. The feasibility and effectiveness of the proposed method is verified in real datasets, and the experimental results show that the proposed algorithm significantly improves program execution efficiency and outperforms the traditional non-negative matrix decomposition algorithm in a series of evaluation metrics.

Keywords Non-negative matrix factorization, L_p -norm, Clustering, Parallel optimization, Rate of convergence

1 引言

随着网络和多媒体技术的高速发展,如何从海量的图像数据中挖掘出更深层次和更具结构性的信息成为一个具有挑战性的问题。由于图像数据是高维的且存储形式具有非负性,因此,非负矩阵分解(Non-negative Matrix Factorization, NMF)^[1]算法在处理图像聚类和特征提取这类问题时具有

良好的可解释性。对于给定的任意一个 $m \times n$ 维的非负矩阵 $V \in \mathbf{R}_{m \times n}^+$ 和分解因子 $r (r \ll \min(m, n))$, 非负矩阵分解算法都能够找到两个非负矩阵 $W \in \mathbf{R}_{m \times r}^+$ 和 $H \in \mathbf{R}_{r \times n}^+$ 使得 $V \approx WH$, 即将原始数据矩阵近似分解为更低秩的基矩阵和系数矩阵。NMF 算法可以有效消除计算过程中的冗余信息,更好地实现对隐含信息的挖掘。与主成分分析^[2]、奇异值分解^[3]等其他矩阵分解算法相比, NMF 更加简单直观,被广泛应用于图像

到稿日期:2023-03-06 返修日期:2023-06-15

基金项目:重庆市教委科学技术研究项目(KJZD-M202201204, KJZD-K202201205);重庆万州区科学技术局科技创新智慧农业项目(2022-17)

This work was supported by the Science and Technology Research Project of Chongqing Municipal Education Commission(KJZD-M202201204, KJZD-K202201205) and Science and Technology Innovation Smart Agriculture Project of Science and Technology Department, Wanzhou District of Chongqing(2022-17).

通信作者:张伟(cqec126@126.com)

聚类^[4]、模式识别^[5]、机器学习^[6]等领域。

目前,针对不同的应用需求,研究者们提出了许多改进方案以提升 NMF 的聚类效果,Guan 等^[7]提出了基于曼哈顿距离的非负矩阵分解,用一个光滑函数去逼近 L₁ 范数表示的非光滑损失函数,减弱了离群点对结果的影响,并利用 Nesterov's 方法^[8]进行求解,解决了非光滑目标函数增加计算开销的问题。Ding 等^[9]对非负矩阵分解进行了系统的分析和扩展,证实了 K-Means 聚类与 NMF 在某种约束条件下的等价关系,此后基于 NMF 和 K-Means 的聚类算法也引起了广泛关注^[10-11]。Kong 等^[12]利用 L_{2,1} 范数来重建目标函数损失,减弱了异常值对 NMF 的影响。Hoyer^[13]提出了稀疏非负矩阵分解算法,增加了表达系数的可解释性。Nakamura 等^[14]提出了一种用于语音增强的 L_p 范数非负矩阵分解算法,提升了增强效果。Zhang 等^[15]提出了 L₁ 范数正则化的 RNMF,通过稀疏误差矩阵,减弱了噪声污染对图像聚类结果的影响。

虽然 NMF 在实际应用中已经取得了诸多成功,但是在处理某些复杂的图像数据时,仍旧面临着聚类效果欠佳、求解效率低且优化结果多为局部最优的问题。本文针对这些缺陷,使用 L_p 范数对非负矩阵分解的目标函数进行了重构,重构后的非凸非光滑目标函数使用 Majorization-Minimization (MM)^[16] 算法进行求解,通过调整系数 p 来获得更好的聚类结果。此外,为提高算法的执行效率,采用粒子群优化算法对重构后的目标函数并行优化求解。在多个真实的人脸数据集上进行测试,验证了所提算法的可行性和有效性。将所提算法与其他经典 NMF 算法进行比较,实验结果表明,所提出的基于 L_p 范数的非负矩阵分解算法(Non-negative Matrix Factorization Parallel Optimization Algorithm Based on L_p-norm, CL_pNMF)能够有效缩减程序的执行时间并获得更好的图像聚类结果。

2 预备知识

本章主要介绍与本文构建目标函数和优化求解相关的定义和算法,包括 L_p 范数、Majorization-Minimization 算法和 OGM 算法。

2.1 L_p 范数定义

假设 $\mathbf{x}=[x_{(1)}, \dots, x_{(n)}]$ 是一个含有 n 个实数的向量,则 \mathbf{x} 的 L_p 范数的定义如下:

$$M_p(\mathbf{x}) = \left(\sum_{i=1}^n |x_{(i)}|^p \right)^{\frac{1}{p}} \quad (1)$$

当 $p=1$ 时,上式表示曼哈顿距离,当 $p=2$ 时,表示欧氏距离。在多种机器学习的任务中,为了防止过拟合,提高算法的泛化能力,常对目标函数中的参数向量引入 L_p 范数来进行正则化。

2.2 Majorization-Minimization 算法求解 L_p 范数问题

MM 算法是一种迭代优化方法,其主要思想是,针对较难优化的目标函数问题,构造一个易于求解的替代函数,对这个替代函数求解,使得替代函数的最优解不断逼近目标函数的最优解。目前该算法已经在统计和机器学习等大规模优化问题中得到了广泛应用^[17-19]。

用 $|\mathbf{Q}|$ 代替上述问题(1)中的 \mathbf{x} , 则其表达式可以重写为式(2):

$$M_p(\mathbf{Q}) = \left(\sum_{i=1}^n Q_{(i)}^p \right)^{\frac{1}{p}} \quad (2)$$

对任意一点 $\mathbf{Q}_{(j)}$ ($1 \leq j \leq n$) 求偏导数:

$$\frac{\partial M_p(\mathbf{Q})}{\partial \mathbf{Q}_{(j)}} = \left(\sum_{i=1}^n Q_{(i)}^p \right)^{\frac{1}{p}-1} Q_{(j)}^{p-1} \quad (3)$$

为了推导出一个合适的替代函数,使用 MM 算法来逼近目标函数的最优解,我们利用函数的凹凸性进行推理。设 O 为函数的任意一点,当 $p < 1$ 时,问题(2)是一个凹函数^[20]。根据凹函数的一阶特性可以得到式(4):

$$M_p(\mathbf{Q}) \leq M_p(O) + \sum_{i=1}^n \frac{\partial M_p(O)}{\partial O_{(j)}} (Q_{(j)} - O_{(j)}) \quad (4)$$

如何推导出一个合适的替代函数,并对该替代函数进行求解,使其逼近目标函数的最优值,后续 3.2 小节会给出更详细的介绍。

2.3 Nesterov's 最优梯度法 (OGM)

基于 Nesterov's^[21] 平滑方法,Guan 等^[7]针对多目标非凸优化问题,提出了一种 OGM 算法。对于 $t+1$ 次的求解目标值 \mathbf{H}_{t+1}^* 而言,该方法可以归纳总结为算法 1。

算法 1 OGM

输入: $\mathbf{V} \in \mathbf{R}_{m \times n}^+, \mathbf{W} \in \mathbf{R}_{r \times m}^+, \mathbf{H} \in \mathbf{R}_{r \times n}^+, \lambda$

输出: \mathbf{H}_{t+1}^*

1. 初始化 $\mathbf{1}$ 是一个 $1 \times n$ 的全一向量, $\mathbf{d} = \sqrt{\sum_{i=1}^r \mathbf{W}(\cdot, i) \in \mathbf{R}^m}$, $\mathbf{Q} = \frac{\sum_{i=1}^m \mathbf{d}}{\lambda}$, 和 $L_\lambda = \frac{1}{\lambda}$.
2. for $k \leftarrow 1$ to ϕ do /* ϕ -循环上限 */
 - 2.1. 计算 $\mathbf{U}_k = \text{med} \left\{ 1, -1, \frac{\mathbf{W}^T \mathbf{H}_k - \mathbf{V}}{\lambda \mathbf{Q}} \right\}$
 - 2.2. 计算 $\nabla f_\lambda(\mathbf{W}, \mathbf{H}_k) = \mathbf{W} \mathbf{U}_k$
 - 2.3. 计算 $\mathbf{Y}_k = \max \left(0, \mathbf{H}_k - \frac{1}{L_\lambda} \nabla f_\lambda(\mathbf{W}, \mathbf{H}_k) \right)$
 - 2.4. 计算 $\mathbf{Z}_k = \max \left(0, -\frac{1}{L_\lambda} \sum_{i=0}^{k-1} \frac{i+1}{2} \nabla f_\lambda(\mathbf{W}, \mathbf{H}_k) \right)$
 - 2.5. 更新 $\mathbf{H}_{k+1} = \frac{2}{k+3} \mathbf{Z}_k + \frac{k+1}{k+3} \mathbf{Y}_k$
3. 检查是否满足结束循环条件
4. 更新 $\mathbf{H}_{t+1}^* = \mathbf{H}_{k+1}$

3 基于 L_p 范数的非负矩阵分解算法

非负矩阵分解的目标是使得原始矩阵与分解后得到的矩阵之间的误差尽可能小。为进一步减小非负矩阵重构后的误差,文中引入 L_p 范数来重构非负矩阵分解的损失函数。

3.1 模型建立

基于上述问题(1)中的 L_p 范数的定义,我们引入 L_p 范数对一个 m 行 n 列的数据矩阵 $\mathbf{V} \in \mathbf{R}_{m \times n}^+$ 的非负矩阵分解问题进行重构,其目标函数可以写为:

$$\min_{\mathbf{W} \geq 0, \mathbf{H} \geq 0} F_p(\mathbf{W}, \mathbf{H}) = (|\mathbf{V} - \mathbf{W}^T \mathbf{H}|^p)^{\frac{1}{p}} \quad (5)$$

s. t. $\mathbf{W} \in \mathbf{R}_{r \times m}^+, \mathbf{H} \in \mathbf{R}_{r \times n}^+$

将问题(5)分解为对 \mathbf{W} 和 \mathbf{H} 每一行或列的优化问题,则目标函数可以被写为:

$$\begin{aligned}
F_s(\mathbf{W}, \mathbf{H}) &= \{ (|\mathbf{V}_{(11)} - \mathbf{W}^{(1)\top} \mathbf{H}^{(1)}|^\rho)^\frac{1}{\rho} + \dots + \\
&\quad (|\mathbf{V}_{(mm)} - \mathbf{W}^{(m)\top} \mathbf{H}^{(m)}|^\rho)^\frac{1}{\rho} \} \\
&= (\sum_{i=1}^m \sum_{j=1}^n |\mathbf{V}_{(ij)} - \mathbf{W}^{(i)\top} \mathbf{H}^{(j)}|^\rho)^\frac{1}{\rho} \quad (6)
\end{aligned}$$

由于目标函数(5)是一个非凸问题,为方便后续求解,采用递归的方式,固定其中的一个变量,对另外一个变量进行求解。对于任意 $t > 0$ 次迭代,可以通过式(7)和式(8),对 \mathbf{H} 和 \mathbf{W} 进行更新。

$$\mathbf{H}_{t+1} = \arg \min_{\mathbf{H} \geq 0} (\sum_{i=1}^m \sum_{j=1}^n |\mathbf{V}_{(ji)} - \mathbf{W}_t^{(j)\top} \mathbf{H}^{(i)}|^\rho)^\frac{1}{\rho} \quad (7)$$

$$\mathbf{W}_{t+1} = \arg \min_{\mathbf{W} \geq 0} (\sum_{i=1}^m \sum_{j=1}^n |\mathbf{V}_{(ij)} - \mathbf{H}_{t+1}^{(j)\top} \mathbf{W}^{(i)}|^\rho)^\frac{1}{\rho} \quad (8)$$

3.2 优化求解

3.2.1 MM 算法求解重构后的目标函数

研究发现,直接对 L_p 范数重构后的问题(7)和问题(8)进行求解是较为困难和繁琐的,通过 MM 算法构造一个可替代函数,可以有效减少工作量。用 $\mathbf{V}_{(ji)} - \mathbf{W}_t^{(j)\top} \mathbf{H}^{(i)}$ 代替上述问题(2)中的 \mathbf{Q} ,则对于矩阵 \mathbf{H} 的优化,第 i 列子问题的目标函数定义如下:

$$\begin{aligned}
F_s(\mathbf{H}^{(i)}) &= M_p(\Theta) = M_p(\Theta_{(1)}, \dots, \Theta_{(m)}) \\
&= M_p(|\mathbf{V}_{(1i)} - \mathbf{W}^{(1)\top} \mathbf{H}^{(i)}|, \dots, |\mathbf{V}_{(mi)} - \mathbf{W}^{(m)\top} \mathbf{H}^{(i)}|) \\
&= (\sum_{j=1}^m |\mathbf{V}_{(ji)} - \mathbf{W}^{(j)\top} \mathbf{H}^{(i)}|^\rho)^\frac{1}{\rho} \quad (9)
\end{aligned}$$

设 O 为任意一点,基于问题(4)的表达式,对目标函数进行替换求解。

$$M_p(\Theta) \leq M_p(O) + \sum_{i=1}^n \frac{\partial M_p(O)}{\partial O_{(j)}} (\Theta_{(j)} - O_{(j)}) \quad (10)$$

其中, $M_p(\Theta)$ 和 $M_p(O)$ 对于问题(9)而言,可以理解为本次迭代期望求得的目标值 $F_s(\mathbf{H}_{t+1}^{(i)})$ 和上一次迭代求得的目标值 $F_s(\mathbf{H}_t^{(i)})$ 。将式(9)中的 $\mathbf{Q}_{(j)}$ 用 $|\mathbf{V}_{(ji)} - \mathbf{W}^{(j)\top} \mathbf{H}_{t+1}^{(i)}|$ 代替,同理,用 $|\mathbf{V}_{(ji)} - \mathbf{W}^{(j)\top} \mathbf{H}_t^{(i)}|$ 代替 $O_{(j)}$,可以得到下列优化不等式:

$$\begin{aligned}
F_s(\mathbf{H}_{t+1}^{(i)}) &\leq \sum_{j=1}^m e_{(j)} |\mathbf{V}_{(ji)} - \mathbf{W}^{(j)\top} \mathbf{H}_{t+1}^{(i)}| + \\
&\quad \underbrace{F_s(\mathbf{H}_t^{(i)}) - \sum_{j=1}^m e_{(j)} |\mathbf{V}_{(ji)} - \mathbf{W}^{(j)\top} \mathbf{H}_t^{(i)}|}_{\text{只与 } \mathbf{H}_t^{(i)} \text{ 有关}} \quad (11)
\end{aligned}$$

其中, $e_{(j)}$ 表示如下:

$$\begin{aligned}
e_{(j)} &= (\sum_{j=1}^m |\mathbf{V}_{(ji)} - \mathbf{W}^{(j)\top} \mathbf{H}_t^{(i)}|^\rho)^\frac{1}{\rho-1} \\
&\quad |\mathbf{V}_{(ji)} - \mathbf{W}^{(j)\top} \mathbf{H}_t^{(i)}|^\rho \quad (12)
\end{aligned}$$

根据上述推导过程,问题(7)的第 i 列优化问题等价于求解下列目标函数:

$$\mathbf{H}_{t+1}^{(i)} = \arg \min_{\mathbf{H}^{(i)} \geq 0} \sum_{j=1}^m e_{(j)} |\mathbf{V}_{(ji)} - \mathbf{W}^{(j)\top} \mathbf{H}_{t+1}^{(i)}| \quad (13)$$

因此,问题(7)和(8)可以表述为:

$$\begin{aligned}
\min_{\mathbf{H} \geq 0} F_s(\mathbf{H}) &= \sum_{i=1}^m \sum_{j=1}^n \mathbf{E}_{(ji)} |\mathbf{V}_{(ji)} - \mathbf{W}^{(j)\top} \mathbf{H}^{(i)}| \\
&= \sum_{i=1}^m |\mathbf{E}^{(i)} \odot (\mathbf{V}^{(i)} - \mathbf{W}^\top \mathbf{H}^{(i)})| \\
&= \sum_{i=1}^m |\mathbf{E}^{(i)} \odot \mathbf{V}^{(i)} - \text{diag}(\mathbf{E}^{(i)}) \mathbf{W}^\top \mathbf{H}^{(i)}| \quad (14)
\end{aligned}$$

同理,

$$\min_{\mathbf{W} \geq 0} F_s(\mathbf{W}) = \sum_{i=1}^m \sum_{j=1}^n \mathbf{E}_{(ij)} |\mathbf{V}_{(ij)} - \mathbf{H}^{(j)\top} \mathbf{W}^{(i)}|$$

$$\begin{aligned}
&= \sum_{i=1}^m |\mathbf{E}_{(i)}^\top \odot (\mathbf{V}_{(i)}^\top - \mathbf{H}^\top \mathbf{W}^{(i)})| \\
&= \sum_{i=1}^m |\mathbf{E}_{(i)}^\top \odot \mathbf{V}_{(i)}^\top - \text{diag}(\mathbf{E}_{(i)}^\top) \mathbf{H}^\top \mathbf{W}^{(i)}| \quad (15)
\end{aligned}$$

其中, \odot 表示点乘, $\text{diag}(\boldsymbol{\tau})$ 表示 $\boldsymbol{\tau}$ 的对角矩阵。 $\mathbf{E}_{(ji)}$ 的表达式如下:

$$\mathbf{E}_{(ji)} = (\sum_{j=1}^m |\mathbf{V}_{(ji)} - \mathbf{W}^{(j)\top} \mathbf{H}_t^{(i)}|^\rho)^\frac{1}{\rho-1} |\mathbf{V}_{(ji)} - \mathbf{W}^{(j)\top} \mathbf{H}_t^{(i)}|^\rho \quad (16)$$

为求解 L_p 范数重构后非负矩阵分解,对式(13)中第 $t+1$ 次矩阵 \mathbf{H} 和 \mathbf{W} 的优化问题,我们在算法 1 的基础上设计了算法 2。通过观察可以发现,问题(7)和问题(8)是对称的,因此调用 OGM 算法交替优化 \mathbf{H} 和 \mathbf{W} ,则重构后的目标函数即问题(5)的优化得以解决。

算法 2 L_p 范数重构后的 NMF 优化算法(LPNMF)

输入: $\mathbf{V} \in \mathbf{R}_{+}^{m \times n}$, $\mathbf{W} \in \mathbf{R}_{+}^{n \times m}$, $\mathbf{H} \in \mathbf{R}_{+}^{m \times n}$, λ, ζ, ρ, r , inner, outer

输出: \mathbf{H}_{t+1}^* , \mathbf{W}_{t+1}^*

1. for $t \leftarrow 1$ to ϕ do / * ϕ -最外层循环上限 * /
2. 根据式(18)计算 $\mathbf{E}^{(i)}$
3. for $k \leftarrow 1$ to μ do / * μ -内层循环上限 * /
4. for $i \leftarrow 1$ to n do / * 对 n 列数据进行更新 * /
 - 4.1. 计算 $\vec{\mathbf{V}} = \mathbf{E}^{(i)} \odot \mathbf{V}^{(i)}$
 - 4.2. 计算 $\bar{\mathbf{W}} = \text{diag}(\mathbf{E}^{(i)}) \mathbf{W}^\top$
 - 4.3. 计算 $\mathbf{H}_{k+1}^{(i)} = \text{OGM}(\vec{\mathbf{V}}, \bar{\mathbf{W}}, \mathbf{H}^{(i)}, \lambda, \epsilon)$
5. 若满足条件 $\frac{|F_s(\mathbf{H}_{k+1}) - F_s(\mathbf{H}_k)|}{|F_s(\mathbf{H}_{k+1}) - F_s(\mathbf{H}_0)|} \leq \epsilon$, 则跳出循环, 否则继续更新
6. 更新 $\mathbf{H}_{t+1} = [\mathbf{H}_{k+1}^{(1)}, \dots, \mathbf{H}_{k+1}^{(n)}]$
7. for $k \leftarrow 1$ to μ do / * μ -内层循环上限 * /
8. for $i \leftarrow 1$ to m do / * 对 m 行数据进行更新 * /
 - 8.1. 计算 $\vec{\mathbf{V}} = \mathbf{E}_{(i)}^\top \odot \mathbf{V}_{(i)}^\top$
 - 8.2. 计算 $\bar{\mathbf{H}} = \text{diag}(\mathbf{E}_{(i)}^\top) \mathbf{H}^\top$
 - 8.3. 计算 $\mathbf{W}_{k+1}^{(i)} = \text{OGM}(\vec{\mathbf{V}}, \bar{\mathbf{H}}, \mathbf{W}^{(i)}, \lambda, \epsilon)$
9. 若满足条件 $\frac{|F_s(\mathbf{W}_{k+1}) - F_s(\mathbf{W}_k)|}{|F_s(\mathbf{W}_{k+1}) - F_s(\mathbf{W}_0)|} \leq \epsilon$, 则跳出循环, 否则继续更新
10. 更新 $\mathbf{W}_{t+1} = [\mathbf{W}_{k+1}^{(1)}, \dots, \mathbf{W}_{k+1}^{(m)}]$
11. 检查是否满足结束循环条件
12. 更新 $\mathbf{H}_{t+1}^* = \mathbf{H}_{t+1}$, $\mathbf{W}_{t+1}^* = \mathbf{W}_{t+1}$

算法 2 中步骤 3-步骤 6 主要是对矩阵 \mathbf{H} 的每一列进行优化即对问题(9)求解。步骤 7-步骤 10 是对矩阵 \mathbf{W} 的每一行进行优化。当满足 ϕ 达到迭代上限或结束循环条件(17)时停止,得到优化后的基矩阵 \mathbf{W} 和系数矩阵 \mathbf{H} 。对于最外层循环,整个优化过程停止的条件为:

$$\frac{|F_s(\mathbf{W}_{t+1}, \mathbf{H}_{t+1}) - F_s(\mathbf{W}_t, \mathbf{H}_t)|}{|F_s(\mathbf{W}_{t+1}, \mathbf{H}_{t+1}) - F_s(\mathbf{W}_0, \mathbf{H}_0)|} \leq \zeta \quad (17)$$

3.2.2 粒子群优化算法

非负矩阵分解在进行优化求解时,常面临着收敛速度慢、容易陷入局部最优解的问题,因此我们使用粒子群优化算法对其进行改进。该算法利用粒子间的交互信息在整个复杂空间进行搜索,求得全局最优解,目前已经扩展到诸多领域,如多目标检测^[22]、信号处理^[23]、无线网络^[24-25]、文本分类^[26-27]以及

控制系统^[28],有效地解决了多类具有挑战性的优化问题。

粒子群算法的两个核心要素分别是粒子的更新速度和改变的位置,其定义如下:

$$\mathbf{Y}_{t+1}^{(i)} = c_0 \mathbf{Y}_t^{(i)} + c_1 r_1 (\mathbf{X}_{t^*}^{(i)} - \mathbf{X}_t^{(i)}) + c_2 r_2 (\vec{\mathbf{x}}_* - \mathbf{X}_t^{(i)}) \quad (18)$$

$$\mathbf{X}_{t+1}^{(i)} = \mathbf{X}_t^{(i)} + \mathbf{Y}_{t+1}^{(i)} \quad (19)$$

其中, $\mathbf{Y}_{t+1}^{(i)}$ 对于问题(11)而言,表示第 i 个子问题在 $t+1$ 次迭代时的更新速度,它由 3 个式子组成。第一项是惯性部分,表示粒子对当前自身的运动状态的依赖程度;第二项是认知部分,表示粒子群位置与自身局部最优解之间的差距;第三项是社会部分,表示粒子群当前位置与全局最优解之间的差距。 c_0, c_1, c_2 分别表示惯性权重、个体学习因子和群体学习因子, r_1 和 r_2 的取值是 $0 \sim 1$ 之间的随机数,用来增加搜索的随机性。 $\mathbf{X}_{t^*}^{(i)}$ 对于问题(11)的第 i 个子问题而言,表示粒子在第 t 次迭代过程中的最优位置, $\mathbf{X}_t^{(i)}$ 表示粒子在第 t 次迭代时的状态向量, $\vec{\mathbf{x}}_*$ 表示全局最优解的位置。

非负矩阵分解优化可以简单地理解为通过更新迭代使得目标最终收敛的过程,该方法运行效率不高。本文引入粒子群算法来并行求解目标函数的全局最优值,进一步缩短程序运行时间。在算法 2 的基础上,根据粒子群优化原理,设计了算法 3,在 Python 编译软件中调用多核 CPU,并行执行粒子的迭代寻优过程。

算法 3 基于 L_p 范数的 NMF 并行优化算法 (CL_pNMF)

输入: $\mathbf{V} \in \mathbf{R}_{m \times n}^+, \xi, p, r, \lambda, c_0, c_1, c_2$, 粒子群数量 N , 最大迭代次数 M

输出: $\mathbf{H}_{t+1}^*, \mathbf{W}_{t+1}^*$

1. 初始化

1.1. 初始化矩阵 \mathbf{W} 和 \mathbf{H} 的当前粒子群位置:

$$\hat{\mathbf{W}} \in \mathbf{R}_{r \times m \times N}, \hat{\mathbf{H}} \in \mathbf{R}_{r \times n \times N}$$

1.2. 初始化粒子群的当前最优位置:

$$\hat{\mathbf{W}}_*^{(1)}, \dots, \hat{\mathbf{W}}_*^{(N)} = \hat{\mathbf{W}}^{(1)}, \dots, \hat{\mathbf{W}}^{(N)}$$

$$\hat{\mathbf{H}}_*^{(1)}, \dots, \hat{\mathbf{H}}_*^{(N)} = \hat{\mathbf{H}}^{(1)}, \dots, \hat{\mathbf{H}}^{(N)}$$

1.3. 初始化当前粒子群的全局最佳位置:

$$\mathbf{W}_0^*, \mathbf{H}_0^* = \underset{\hat{\mathbf{W}}^{(i)}, \hat{\mathbf{H}}^{(i)}}{\operatorname{argmin}} \{F_s(\hat{\mathbf{W}}^{(0)}, \hat{\mathbf{H}}^{(0)}), \dots, F_s(\hat{\mathbf{W}}^{(N)}, \hat{\mathbf{H}}^{(N)})\}$$

1.4. 初始化 \mathbf{W} 和 \mathbf{H} 的前进速度:

$$\hat{\mathbf{U}} \in \mathbf{R}_{r \times m \times N} = \mathbf{0}, \hat{\mathbf{P}} \in \mathbf{R}_{r \times n \times N} = \mathbf{0}$$

1.5. 初始化迭代次数 $k=1$

2. 如果 $K < M / *$ M -迭代次数上限,则开始并行优化,粒子群数量与 CPU 核数相对应 $*/$

3. for $j \leftarrow 1$ to N do $*/ N$ -粒子群数量 $*/$

3.1. $[\mathbf{W}, \mathbf{H}] = \text{LPNMF}(\mathbf{W}, \hat{\mathbf{W}}^{(j)}, \hat{\mathbf{H}}^{(j)}, \lambda, \text{inner}, \text{outer}, \zeta, \epsilon, p)$

3.2. 如果 $F_s(\mathbf{W}, \mathbf{H}) < F_s(\hat{\mathbf{W}}_*^{(j)}, \hat{\mathbf{H}}_*^{(j)})$, 则有 $\hat{\mathbf{W}}_*^{(j)} = \mathbf{W}, \hat{\mathbf{H}}_*^{(j)} = \mathbf{H}$

4. 更新 $\{\mathbf{W}_k^*, \mathbf{H}_k^*\} = \underset{\hat{\mathbf{W}}_*^{(i)}, \hat{\mathbf{H}}_*^{(i)}}{\operatorname{argmin}} \{F_s(\hat{\mathbf{W}}_*^{(1)}, \hat{\mathbf{H}}_*^{(1)}), \dots, F_s(\hat{\mathbf{W}}_*^{(N)}, \hat{\mathbf{H}}_*^{(N)})\}$

5. 如果 $F_s(\mathbf{W}_k^*, \mathbf{H}_k^*) < F_s(\mathbf{W}_{k-1}^*, \mathbf{H}_{k-1}^*)$

则有 $k = k+1$, 否则 $k=1$

6. for $j \leftarrow 1$ to N do $*/$ 对 $\hat{\mathbf{W}}$ 更新前进速度 $*/$

6.1. 初始化 r_1 和 r_2 为 $0 \sim 1$ 之间的随机数

6.2. 计算 $\hat{\mathbf{W}}$ 的前进速度:

$$\hat{\mathbf{U}}^{(j)} = c_0 \hat{\mathbf{U}}^{(j)} + c_1 r_1 (\mathbf{W}_k^* - \hat{\mathbf{W}}^{(j)}) + c_2 r_2 (\hat{\mathbf{W}}_*^{(j)} - \hat{\mathbf{W}}^{(j)})$$

6.3. 更新 $\hat{\mathbf{W}}$ 的位置: $\hat{\mathbf{W}}^{(j)} = \hat{\mathbf{W}}^{(j)} + \hat{\mathbf{U}}^{(j)}$

7. for $j \leftarrow 1$ to N do $*/$ 对 $\hat{\mathbf{H}}$ 更新前进速度 $*/$

7.1. 初始化 r_1 和 r_2 为 $0 \sim 1$ 之间的随机数

7.2. 计算 $\hat{\mathbf{H}}$ 的前进速度:

$$\hat{\mathbf{P}}^{(j)} = c_0 \hat{\mathbf{P}}^{(j)} + c_1 r_1 (\mathbf{H}_k^* - \hat{\mathbf{P}}^{(j)}) + c_2 r_2 (\hat{\mathbf{H}}_*^{(j)} - \hat{\mathbf{P}}^{(j)})$$

7.3. 更新 $\hat{\mathbf{H}}$ 的位置: $\hat{\mathbf{H}}^{(j)} = \hat{\mathbf{H}}^{(j)} + \hat{\mathbf{P}}^{(j)}$

8. 更新 $\mathbf{W}_* \leftarrow \mathbf{W}_k^*$

9. 更新 $\mathbf{H}_* \leftarrow \mathbf{H}_k^*$

算法 3 中,步骤 3.1 是对 N 个粒子群求解局部最优解,步骤 3.2 更新当前粒子群的最佳位置;步骤 4 更新全局最佳位置;步骤 6 和步骤 7 依据对应公式,重置非负矩阵分解模型的每个初始解。

4 实验结果及分析

4.1 数据集与实验设置

4.1.1 数据集

ORL 数据集包含 400 张灰度图像,共采集了 40 个不同年龄、性别、种族的人物对象在不同光照、表情、姿态变化下的图片。该数据集的部分人脸图像如图 1 所示。



图 1 ORL 数据集样图

Fig. 1 Sample images of ORL dataset

Jaffe 数据集共包含 213 张照片,由 10 位女性在不同心情状态时做出的 7 种表情组成,部分图像如图 2 所示。



图 2 Jaffe 数据集样图

Fig. 2 Sample images of Jaffe dataset

Yale 数据集由 165 张灰度图片组成,对 15 位志愿者在不同的光照、表情和姿态变化下进行采集,对每位志愿者采集了 11 张图片,部分图像如图 3 所示。



图3 Yale数据集样图

Fig. 3 Sample images of Yale dataset

4.1.2 实验环境及参数设置

本文提出的基于 L_p 范数的非负矩阵并行优化算法,在以上 3 个广泛使用的人脸标准数据集中进行了验证。使用 Python3.8(64-bit)编译软件实现,运行于 Intel(R) Core(TM) i5-10500 CPU @ 3.10GHz 处理器的个人电脑,运行环境为 Windows 10。

运用算法 3 并行迭代求解时,文中的一些具体参数配置如下:矩阵 \mathbf{W} 和 \mathbf{H} 的当前粒子群位置均采用随机初始化的方式,其中分解因子 r 与数据样本的类别相关; p 表示 L_p 范数重构后目标函数中的 p 值,取值范围为 $0 \sim 1$; c_0, c_1, c_2 表示每一项的量子因子,本实验分别取 1, 2, 2; ξ 表示结束循环的阈值,设置为 e^{-6} ; λ 用来控制 OGM 算法中的目标函数平滑度,取值为 1; 最大迭代次数 M 设置为 500。

4.2 技术指标与对比算法

4.2.1 技术指标

本实验采用了准确率(ACC)与归一化信息(NMI)这两种最常用的指标,来验证所提算法在进行图像聚类时的性能。图像聚类的效果与 ACC 和 NMI 成正相关,数值越大意味着图像聚类的效果越好。这两种评价指标的计算方式如下:

$$ACC = \frac{B}{N} \quad (20)$$

其中, B 表示正确分类的样本数, N 表示样本的总个数。

$$NMI = 2 \frac{MI(\mathbf{U}, \mathbf{V})}{H(\mathbf{U}) + H(\mathbf{V})} \quad (21)$$

其中, $H(\mathbf{U})$ 和 $H(\mathbf{V})$ 分别表示 \mathbf{U} 和 \mathbf{V} 的信息熵, $MI(\mathbf{U}, \mathbf{V})$ 为两个标签集合的互信息。

4.2.2 对比算法

为了验证所提算法的有效性,将 CL $_p$ NMF 算法与 3 个不同标准衡量的经典 NMF 算法进行对比。对比算法分别是 Lee 和 Seung 基于欧几里得距离和 Kullback-Leibler 散度提出的两种经典的 NMF 模型^[29], 和 Guan 等基于曼哈顿距离的 NMF 模型。不同算法的目标函数如表 1 所列。

表 1 对比算法的名称及目标函数

Table 1 Names of comparison algorithms and objective functions

| 算法名称 | 目标函数 |
|-------------|---|
| CL $_p$ NMF | $\min_{\mathbf{W}, \mathbf{H}} (\ \mathbf{V} - \mathbf{W}^T \mathbf{H}\ _p)^{\frac{1}{p}}$ |
| NMF | $\min_{\mathbf{W}, \mathbf{H}} \ \mathbf{V} - \mathbf{W} \mathbf{H}\ _F^2$ |
| NML-KL | $\min_{\mathbf{W}, \mathbf{H}} \sum_{i=1}^m \sum_{j=1}^n \left(v_{ij} \ln \frac{v_{ij}}{[\mathbf{W} \mathbf{H}]_{ij}} - v_{ij} + [\mathbf{W} \mathbf{H}]_{ij} \right)$ |
| MahNMF | $\min_{\mathbf{W}, \mathbf{H}} \ \mathbf{V} - \mathbf{W}^T \mathbf{H}\ _M$ |

4.3 实验结果

4.3.1 函数收敛性实验

本文提出的基于 L_p 范数的非负矩阵分解并行优化算法本质上是迭代的,优化的目标函数为 $\mathbf{E} \odot |\mathbf{V} - \mathbf{W}^T \mathbf{H}|$,选用的优化迭代收敛准则为超过设定的最大迭代次数 500 或者满足条件(17)。为了更好地观察优化目标函数的变化情况,所提算法在 3 个数据集上进行了相关验证,选取相同的参数 p (此处选为 0.5),不同数据集中目标函数的变化情况如图 4 所示。

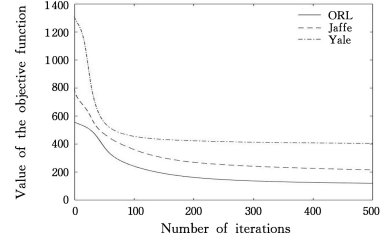


图4 不同数据集的目标函数值变化情况

Fig. 4 Variation of objective function values in different datasets

4.3.2 聚类结果对比实验

为了验证所提算法(CL $_p$ NMF)的有效性,本文在不同数据集上进行了多次实验,在相同的参数 p 下进行了 20 次实验,取 20 次结果的平均值作为实验结果。对经过不同 NMF 算法优化得到的矩阵,使用 K-Means 算法进行聚类。同时,为保证实验结果的客观性,从 $0 \sim 1$ 的数值区间共选择了 20 组不同的参数 p 进行实验,其 ACC 和 NMI 随参数 p 的具体变化情况如图 5—图 7 所示。

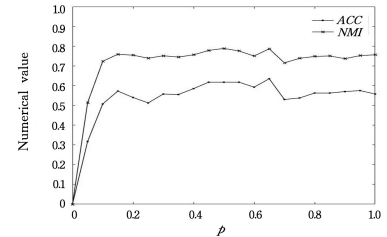


图5 ORL数据集聚类指标随参数 p 变化情况

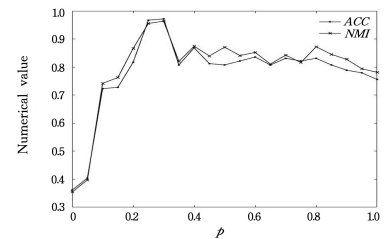
Fig. 5 Variation of clustering metrics with p on ORL dataset

图6 Jaffe数据集聚类指标随参数 p 变化情况

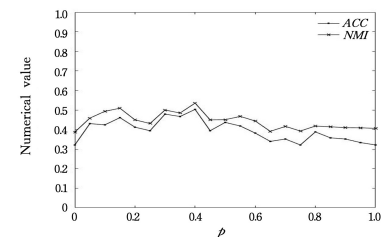
Fig. 6 Variation of clustering metrics with p on Jaffe dataset

图7 Yale数据集聚类指标随参数 p 变化情况

Fig. 7 Variation of clustering metrics with p on Yale dataset

由图5—图7可知,当 p 在数据集 ORL, Jaffe, Yale 上分别取 $0.5 \sim 0.6, 0.2 \sim 0.3, 0.3 \sim 0.4$ 时,聚类效果更好。

表2列出了各算法在不同数据集上的 ACC 和 NMI 值。

表2 不同算法在数据集上的比较

Table 2 Comparison of different algorithms on three datasets

| 数据集 | 算法 | ACC | NMI |
|-----------|--------|---------------|---------------|
| ORL 数据集 | CLpNMF | 0.5925 | 0.7746 |
| | NMF | 0.5850 | 0.7023 |
| | NMF-KL | 0.5825 | 0.7469 |
| | MahNMF | 0.5875 | 0.7575 |
| Jaffe 数据集 | CLpNMF | 0.9577 | 0.9459 |
| | NMF | 0.6948 | 0.7306 |
| | NMF-KL | 0.8357 | 0.8297 |
| | MahNMF | 0.8545 | 0.9066 |
| Yale 数据集 | CLpNMF | 0.5273 | 0.5621 |
| | NMF | 0.4244 | 0.4612 |
| | NMF-KL | 0.3828 | 0.4546 |
| | MahNMF | 0.4727 | 0.4925 |

表3 各数据集粒子数量不相同并行与未并行优化耗时

Table 3 Parallel and unparallel optimization time consumption with different number of particles in datasets

| 数据集 | 计算方式 | 粒子数 | | | | | | | | | |
|-----------|-----------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | | N=2 | N=3 | N=4 | N=5 | N=6 | N=7 | N=8 | N=9 | N=10 | N=11 |
| ORL 数据集 | 并行优化时间/s | 24.38 | 27.80 | 35.88 | 43.87 | 54.29 | 66.20 | 79.54 | 94.08 | 107.46 | 120.14 |
| | 未并行优化时间/s | 36.48 | 104.12 | 120.49 | 160.09 | 192.06 | 222.52 | 276.65 | 287.96 | 317.85 | 414.55 |
| Jaffe 数据集 | 并行优化时间/s | 7.82 | 11.26 | 15.16 | 18.85 | 21.97 | 25.65 | 28.93 | 32.60 | 36.85 | 39.36 |
| | 未并行优化时间/s | 10.82 | 16.80 | 23.98 | 30.76 | 37.71 | 47.13 | 51.10 | 57.11 | 62.90 | 65.10 |
| Yale 数据集 | 并行优化时间/s | 25.16 | 30.84 | 47.95 | 63.82 | 70.61 | 79.75 | 87.68 | 91.69 | 109.75 | 117.16 |
| | 未并行优化时间/s | 30.93 | 57.62 | 77.94 | 136.09 | 161.86 | 199.76 | 203.60 | 239.31 | 233.16 | 246.49 |

结束语 本文提出的基于 L_p 范数的非负矩阵并行优化算法利用 MM 算法将原优化目标函数等价转换为了多个加权非负矩阵分解的子问题。在多个真实人脸数据集上进行对比实验,证实了所提算法在应用于图像聚类时,相比传统 NMF 模型显示出了更优的聚类性能。此外,为进一步加快目标函数的收敛速度,本文在粒子群优化算法的基础上,调用多核 CPU 并行执行程序。对比实验结果表明,该并行优化算法明显缩短了程序执行时间。然而,所提算法只是人为设定多组参数进行实验,对于某些数据集的泛化能力不强。为提高该算法的普适性,未来将对参数 p 的最优选择策略展开进一步研究。

参考文献

- [1] LEE D D, SEUNG H S. Learning the parts of objects by non-negative matrix factorization[J]. Nature, 1999, 401(6755): 788-791.
- [2] ADDI H, WILLIAMS L J. Principal component analysis[J]. Wiley Interdisciplinary Reviews: Computational Statistics, 2010, 2(4): 433-459.
- [3] PAATEROP P, TAPPER U. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values[J]. Environmetrics, 1994, 5(2): 111-126.
- [4] LI X L, JIA M X. Nonnegative Matrix Factorization Algorithm with Hypergraph Based on pertreatments [J]. Computer Science, 2020, 47(7): 71-77.
- [5] JAIN A K, DUIN R P W, MAO J. Statistical pattern recognition: A review[J]. IEEE Transactions on pattern analysis and machine intelligence, 2000, 22(1): 4-37.
- [6] MITCHELL T, BUCHANAN B, DEJONG G, et al. Machine learning[J]. Annual Review of Computer Science, 1990, 4(1): 417-433.
- [7] GUAN N, TAO D, LUO Z, et al. MahNMF: Manhattan Non-negative Matrix Factorization[J]. Journal of Machine Learning Research, 2012, 1050: 14-59.
- [8] NESTEROV Y. Smooth minimization of non-smooth functions [J]. Mathematical Programming, 2005, 103(1): 127-152.
- [9] DING C, HE X, SIMON H D. On the equivalence of non-negative matrix factorization and spectral clustering [C] // Proceedings of the 2005 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2005: 606-610.
- [10] WANG S, CHANG T H, CUI Y, et al. Clustering by orthogonal NMF model and non-convex penalty optimization [J]. IEEE Transactions on Signal Processing, 2021, 69: 5273-5288.
- [11] NAGAYAMA M, ARITAKE T, HINO H, et al. Detecting cell assemblies by NMF-based clustering from calcium imaging data [J]. Neural Networks, 2022, 149: 29-39.
- [12] KONG D, DING C, HUANG H. Robust non-negative matrix factorization using $L_{2,1}$ -norm [C] // The 20th ACM International Conference on Information and Knowledge Management, 2011: 673-682.
- [13] HOYER P O. Non-negative matrix factorization with sparseness constraints [J]. Journal of Machine Learning Research, 2004, 5(9): 1457-1469.
- [14] NAKAMURA T, KAMEOKA H. L_p -norm non-negative matrix factorization and its application to singing voice enhancement [C] // 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015: 2115-2119.
- [15] ZHANG L, CHEN Z, ZHENG M, et al. Robust non-negative

- matrix factorization[J]. *Frontiers of Electrical and Electronic Engineering in China*, 2011, 6:192-200.
- [16] LANGE K. MM optimization algorithms[M]. Society for Industrial and Applied Mathematics, 2016.
- [17] YAO Y, LI Z, LIU H, et al. Robust Transceiver Optimization Against Echo Eclipsing Via Majorization-Minimization[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2022, 59(3):2464-2479.
- [18] PANWAR K, KATWE M, BABU P, et al. A Majorization-Minimization Algorithm for Hybrid T-OA-RSS Based Localization in NLOS Environment[J]. *IEEE Communications Letters*, 2022, 26(5):1017-1021.
- [19] LIU F, SHANG X, ZHU H. Efficient majorization-minimization-based channel estimation for onebit massive MIMO systems[J]. *IEEE Transactions on Wireless Communications*, 2021, 20(6):3444-3457.
- [20] XU J, LANGE K. Power k-means clustering[C]// *International Conference On Machine Learning*. PMLR, 2019:6921-6931.
- [21] NESTEROV Y E. A method for solving the convex programming problem with convergence rate $O(1/k \wedge 2)$ [J]. *Dokl. Akad. Nauk SSSR* 269, 1983, 269(3):543-547.
- [22] QU L C, LYU J, QU Y H, et al. Intelligent Assignment and Positioning Algorithm of Moving Target Based on Fuzzy Neural Network[J]. *Computer Science*, 2021, 48(8):246-252.
- [23] YU M, LI G, JIANG D, et al. Application of PSO-RBF neural network in gesture recognition of continuous surface EMG signals[J]. *Journal of Intelligent & Fuzzy Systems*, 2020, 38(3):2469-2480.
- [24] QABOUCHE H, SAHEL A, BADRI A, et al. Energy efficient PSO-based routing protocol for large WSN[C]// *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology(IRASET)*. IEEE, 2022:1-7.
- [25] LI Y. An Improved Particle Swarm Optimization Algorithm Based on Simulated Annealing for Large-Scale Node Location of WSN[C]// *2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology(ICCASIT)*. IEEE, 2022:971-974.
- [26] ALHAJ Y A, DAHOU A, ALQANESS M A, et al. A novel text classification technique using im-proved particle swarm optimization: A case study of Arabic language[J]. *Future Internet*, 2022, 14(7):194-211.
- [27] WANG L Z, MU X D, LIU H L. Using SVM Method Optimized by Improved Particle Swarm O-ptimization to Analyze Emotion of Chinese Text[J]. *Computer Science*, 2020, 47(1):231-236.
- [28] PAN Y, WU Y, LAM H K. Security based fuzzy control for nonlinear networked control systems with DoS attacks via a resilient event-triggered scheme[J]. *IEEE Transactions on Fuzzy Systems*, 2022, 30(10):4359-4368.
- [29] LEE D D, SEUNGH S. Algorithms for non-negative matrix factorization[J]. *Advances in Neural Information Processing Systems*, 2000, 13:556-562.



HUANG Lulu, born in 1997, postgraduate. Her main research interests include computer vision and algorithm optimization.



ZHANG Wei, born in 1970, Ph.D, professor. His main research interests include information security and computational intelligence.

(责任编辑:杨雪敏)