

# 基于图正则化和稀疏约束的增量型非负矩阵分解

孙 静 蔡希彪 姜小燕 孙福明

(辽宁工业大学电子与信息工程学院 锦州 121001)

**摘要** 非负矩阵分解(Nonnegative Matrix Factorization, NMF)不仅可以很好地描述数据而且分解后的矩阵具有直观的物理意义。为了提高算法的有效性和识别率,提出了一种更为合理的算法——基于图正则化和稀疏约束的增量型非负矩阵分解(Graph Regularized and Incremental Nonnegative Matrix Factorization with Sparseness Constraints, GINMFSC)。该算法既保持了数据的几何结构,又充分利用上一步的分解结果进行增量学习,而且对系数矩阵施加了稀疏性约束,最后将它们整合于单个目标函数中,构造了一个有效的更新算法。在多个数据库上的仿真结果表明,相对于NMF,GNMF,INMF,IGNMF等算法,GINMFSC算法在降低运算时间的同时,还具有更好的聚类精度和稀疏性。

**关键词** 非负矩阵分解,图正则,稀疏约束,增量学习

中图分类号 TP37 文献标识码 A DOI 10.11896/j.issn.1002-137X.2017.06.053

## Graph Regularized and Incremental Nonnegative Matrix Factorization with Sparseness Constraints

SUN Jing CAI Xi-biao JIANG Xiao-yan SUN Fu-ming

(School of Electronics and Information Engineering, Liaoning University of Technology, Jinzhou 121001, China)

**Abstract** Nonnegative matrix factorization (NMF) not only is a description of the data, but also has intuitive physical meaning after the decomposition of the matrix. With the aim to enhance the validity and classification accuracy, a more reasonable algorithm was proposed, which is graph regularized and incremental nonnegative matrix factorization with sparseness constraints (GINMFSC). It not only preserves the intrinsic geometry of data, but also makes full use of the last step decomposition results as incremental learning, and introduces sparseness constraint to coefficient matrix. Finally, they are integrated into one single objective function and an efficient updating approach is produced. Compared with NMF, GNMF, INMF and IGNMF, experiments on several databases have shown that the proposed method achieves better clustering accuracy and sparsity while reducing the computation time.

**Keywords** Nonnegative matrix factorization, Graph regularized, Sparseness constraints, Incremental learning

## 1 引言

图像视觉特征与高层语义之间存在的“语义鸿沟”,成为了图像检索领域发展的瓶颈。提高图像内容的利用效率、缩小语义鸿沟的核心问题是如何有效地表示图像的视觉特征,即如何找到一种合理的表示数据的方法。为了处理矩阵中的数据信息,通常将矩阵进行分解。矩阵分解后不但可以使原始矩阵的维数大大消减,同时也可以对原始矩阵中存放的数据进行概括和压缩。矩阵分解技术受到了越来越多的关注,典型的矩阵分解方法主要包括:主成分分析(PCA)<sup>[1]</sup>、线性判别分析(LDA)<sup>[2]</sup>、独立分量分析(ICA)<sup>[3]</sup>、奇异值分解(SVD)<sup>[4]</sup>等。这些方法通常是在一定的限制下对原始数据矩阵进行分解或线性变换,其共同点是在求解时允许分解结果出现负值,从计算角度来看这是正确的,但是在实际应用中负

值是没有任何意义的。

目前,国际上提出了一种新的矩阵分解方法——非负矩阵分解<sup>[5]</sup>。Daniel D. Lee和H. Sebastian Seung<sup>[6-7]</sup>于1999年在《Nature》杂志上发表了非负矩阵分解算法的论文,首次提出了非负矩阵分解的概念。非负矩阵分解方法不同于上述典型的矩阵分解方法,该方法的独特之处在于矩阵分解过程中所有元素均非负。非负矩阵分解方法将原始矩阵分解为左右两个非负矩阵,左面的矩阵称为基矩阵,右面的矩阵称为系数矩阵。所以,原始矩阵中的列向量可以解释为对基矩阵中所有列向量的加权和,而权重系数即为系数矩阵。这样的向量组合具有直观的语义解释,符合人类思维中“局部构成整体”的概念<sup>[8]</sup>。非负矩阵分解通过添加非负限制,不但保证了分解结果的可解释性,还具有实现简便、占用存储空间小等优点,从而更加贴近应用领域。因此,对非负矩阵分解进行探索

到稿日期:2016-04-18 返修日期:2016-09-01 本文受国家自然科学基金(61272214,61472059)资助。

孙 静(1992—),女,硕士生,主要研究领域为移动通信与无线技术,E-mail:694600868@qq.com;蔡希彪(1972—),男,博士,副教授,主要研究领域为移动通信与无线技术,E-mail:xbc1111@126.com;姜小燕(1989—),女,硕士生,主要研究领域为图像语义理解,E-mail:1091061380@qq.com;孙福明(1972—),男,博士,教授,CCF会员,主要研究领域为计算机视觉、图像语义理解,E-mail:sunwenfriend@hotmail.com。

具有一定的实际意义。

NMF 自提出后立即得到了学术界的广泛关注和深入研究,并且获得了大量的研究成果。为了提高 NMF 算法的有效性和识别率,诸多学者对 NMF 算法进行了深入研究,为了满足特定的需求并达到特定的效果,在基本的 NMF 框架中添加各种约束,如稀疏性、流形、正交性、判别性等,提出了若干种改进的算法,并将其成功地应用到各个领域,取得了良好的实验效果。浙江大学计算机学院的蔡登教授等人<sup>[9]</sup>考虑到原始数据中蕴含的几何结构,从而将流形学习与非负矩阵分解相结合,提出了图正则化非负矩阵分解(Graph Regularized Nonnegative Matrix Factorization, GNMF)算法,该算法在分解过程中保留了数据集携带的局部几何信息,使得低维表示很好地保留了原始样本的近邻结构。姜伟等人<sup>[10]</sup>提出了稀疏约束图正则非负矩阵分解算法(GNMFSC),不但加入了数据的几何信息,还对系数矩阵施加了稀疏约束,使分解后的人脸图像有更高的识别率。Liu 等<sup>[11]</sup>提出了一种半监督约束非负矩阵分解(Constrained Nonnegative Matrix Factorization, CNMF)算法,该算法将已知标签信息约束到 NMF 分解中,但缺陷是没有考虑样本的几何结构以及未施加稀疏约束。后来,学者在 CNMF 的基础之上增加了稀疏约束,并提出了稀疏约束的半监督非负矩阵分解(CNMFSC)<sup>[12]</sup>,该方法不仅考虑了标签信息,还对系数矩阵进行了稀疏约束,实验结果表明该算法的聚类精度高。Serhat S. Bucak 等人<sup>[13]</sup>将增量学习与非负矩阵分解相结合,提出了增量式非负矩阵分解算法(Incremental Nonnegative Matrix Factorization, INMF),该算法使用分块矩阵的思想降低了运算规模,利用上一步的分解结果参与迭代运算,避免新加入的训练样本造成重新运算,节省了大量的运算时间。随后 Yu 等人<sup>[14]</sup>将增量学习和图正则化同时引入 NMF 框架中,提出了增量式图正则化非负矩阵分解(IGNMF),该方法在增量学习的框架下保持了样本所在空间的几何结构。

本文将 GNMF 算法和 INMF 算法结合在一起,并在此基础之上增加了稀疏约束,从而提出了基于图正则化和稀疏约束的增量型非负矩阵分解(Graph Regularized and Incremental Nonnegative Matrix Factorization with Sparseness Constraints, GINMFSC)算法。该算法不仅能有效地保持样本的局部结构,使算法具有较高的分类准确率,还能结合增量学习充分利用上一步的分解结果参与迭代运算,避免重复计算从而降低运算时间。最后在多个数据库上的仿真结果表明,GINMFSC 算法优于其他算法。

## 2 非负矩阵分解算法

对给定的矩阵进行分解,其结果往往不是唯一的,利用 NMF 算法进行分解时其结果往往也是不尽相同的。对于给定的  $n$  个非负样本  $x_i (i=1, 2, \dots, n)$ , 组成矩阵  $X=[x_1, x_2, \dots, x_n] \in R^{m \times n}$ , 每一个  $x_i \in R^m$  均是列向量。NMF 算法的目的就是寻找两个非负矩阵  $U \in R^{m \times k}$  和  $V \in R^{n \times k}$ ,  $U$  是特征矩阵或基矩阵,  $V$  是系数矩阵, 其中  $k \leq \frac{mn}{(m+n)}$ , 并且  $U$  和  $V$  中的元素都为非负值, 使得  $X \approx UV^T$ , 也就是使  $X$  与  $UV^T$

的差异值最小,即优化目标函数。NMF 算法有两种描述目标函数的形式:欧氏距离和 K-L 散度。

(1) 目标函数的描述形式:欧氏距离

$$O_F = \|X - UV^T\|_F^2, \text{ s. t. } U \geq 0, V \geq 0 \quad (1)$$

其中,  $\|\cdot\|_F$  是 Frobenius 范数, 当且仅当  $X=UV^T$  时, 目标函数的值为 0。通过求解相对应的两个矩阵的元素差的平方来衡量  $X$  与  $UV^T$  的相似度, 其值越小, 表明  $X$  与  $UV^T$  的差异越小; 反之越大。通常这种描述目标函数的方式使用较为广泛, 而且对数据的处理十分方便。

(2) 目标函数描述形式:K-L 散度

$$O_{KL} = D(X \| UV^T) \\ = \sum_{i,j} (x_{ij} \log \frac{x_{ij}}{(uv^T)_{ij}} - x_{ij} + (uv^T)_{ij}) \quad (2)$$

这种形式描述的是  $X$  与  $UV^T$  离散的程度, 当且仅当  $\sum_{i,j} x_{ij} = \sum_{i,j} (uv^T)_{ij} = 1$  时, 散度为 0, 并且此时  $X$  与  $UV^T$  服从标准正态分布。

上述两个目标函数只有在单独对  $U$  或者  $V$  求解时才是凸函数, 且能得到最优解。因此, 找到目标函数的全局最优解并不是一件容易的事, 反而可以通过寻找局部最优解的方法来优化目标函数。经过推导证明, 以上两个目标函数是收敛的(收敛性的证明过程在这里不做详细介绍)。利用最速下降法和迭代法推导出最小化目标函数(1)的乘法迭代更新公式如下:

$$u_{ik} \leftarrow u_{ik} \frac{(XV)_{ik}}{(UV^T V)_{ik}} \quad (3)$$

$$v_{jk} \leftarrow v_{jk} \frac{(X^T U)_{jk}}{(V U^T U)_{jk}} \quad (4)$$

其中,  $U=[u_{ik}], V=[v_{jk}]$ 。

在计算过程中, 随机选取非负初始矩阵  $U_0$  和  $V_0$ , 在给定迭代终止条件后, 按照式(3)和式(4)交替迭代更新直到满足终止条件, 可得到最终的  $U$  和  $V$ 。

## 3 基于图正则化和稀疏约束的增量型非负矩阵分解算法

### 3.1 图正则化非负矩阵分解

蔡登教授等人提出了图正则化非负矩阵分解(GNMF)算法。GNMF 算法在进行矩阵分解的同时, 要求在低维空间中继续保持样本的几何结构, 即要求分解后的数据保持原始数据固有的结构信息。假设两个数据点  $x_i$  和  $x_j$  在原始空间是邻近点, 那么在新的基下相应的  $v_i$  和  $v_j$  也是邻近点。即该算法在 NMF 算法的基础上, 具有使得分解后的点能够保持原数据上图的流形的特点。

设  $G$  为原始数据点构成的图, 其中  $R_{ij}$  表示权重矩阵,  $N_p(x_i)$  表示  $x_i$  的  $p$  个近邻, 则有:

$$R_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{\sigma}}, & x_i \in N_p(x_j) \text{ 或 } x_j \in N_p(x_i) \\ 0, & \text{其它} \end{cases}$$

定义  $L=D-R, D_{ii}=\sum_j R_{ij}$ ,  $D$  是对角矩阵,  $L$  是拉普拉斯矩阵。

GNMF 算法的最小化目标函数为:

$$O_F = \|X - UV^T\|_F^2 + \lambda \text{Tr}(V^T L V) \quad (5)$$

s. t.  $U \geq 0, V \geq 0$

其中,  $\lambda$  是正则化参数, 且  $\lambda$  为大于 0 的常数。

目标函数(5)的迭代更新规则如下:

$$u_{ik} \leftarrow u_{ik} \frac{(XV)_{ik}}{(UV^T V)_{ik}} \quad (6)$$

$$v_{jk} \leftarrow v_{jk} \frac{(X^T U + \lambda R V)_{jk}}{(V U^T U + \lambda D V)_{jk}} \quad (7)$$

蔡登教授等人证明了在此迭代更新规则下的目标函数是收敛的。这样就把流形学习和 NMF 结合在一起, 并具有良好的性质。

### 3.2 增量非负矩阵分解

传统的 NMF 迭代更新基矩阵和系数矩阵中的所有元素, 所以迭代次数随着新样本数据非线性增加。针对此问题, Bucak 等人将增量学习引入到非负矩阵分解中, 提出了增量非负矩阵分解(INMF)算法。当新的样本加入后, INMF 算法能够增量地调整子空间的基, 其所需的存储空间大小是不变的, 不随样本的增加而增加。它可以使 NMF 算法在应用于逐渐增大的矩阵时能利用上一步的分解结果, 从而降低重新学习阶段的运算时间。下面介绍该算法。

设  $U_k$  和  $V_k$  分别表示样本集合  $X_k$  经非负矩阵分解后得到的基矩阵和系数矩阵, 其中  $k$  表示当前样本集合中存在的样本数。  $F_k$  代表  $X_k$  经过非负矩阵分解后的目标函数, 则  $F_k$  的表达式如下:

$$F_k = \|X - U_k V_k^T\|_F^2 = \sum_{i=1}^m \sum_{j=1}^k (X_{ij} - (U_k V_k^T)_{ij})^2 \quad (8)$$

在新增加样本  $x_{k+1}$  后, 目标函数变为:

$$F_{k+1} = \|X - U_{k+1} V_{k+1}^T\|_F^2 = \sum_{i=1}^m \sum_{j=1}^{k+1} (X_{ij} - (U_{k+1} V_{k+1}^T)_{ij})^2 \quad (9)$$

其中,  $U_{k+1}$  和  $V_{k+1}$  分别表示样本集合  $X_{k+1}$  非负矩阵分解后得到的基矩阵和系数矩阵。

INMF 算法假定当新样本  $x_{k+1}$  到来时, 系数矩阵  $V_{k+1}$  的前  $k$  列是不变的,  $V_{k+1}$  则可以表示为  $V_{k+1} = [V_k, v_{k+1}]$ 。  $F_{k+1}$  可以进一步表示为  $F_k$  与新输入样本  $x_{k+1}$  所引入的残余误差  $f_{k+1}$  之和。

$$\begin{aligned} F_{k+1} &= \sum_{i=1}^m \sum_{j=1}^{k+1} (X_{ij} - (U_{k+1} V_{k+1}^T)_{ij})^2 \\ &\approx \sum_{i=1}^m \sum_{j=1}^k (X_{ij} - (U_k V_k^T)_{ij})^2 + \sum_{i=1}^m ((x_{k+1})_i - \\ &\quad (U_{k+1} v_{k+1}^T)_i)^2 \\ &= F_k + f_{k+1} \end{aligned} \quad (10)$$

其中,  $f_{k+1} = \sum_{i=1}^m ((x_{k+1})_i - (U_{k+1} v_{k+1}^T)_i)^2$ 。

在得到了目标函数  $F_{k+1}$  的增量式表达后, 利用梯度下降法可以推导出式(10)对应的迭代更新公式:

$$\begin{aligned} (v_{k+1})_a &\leftarrow (v_{k+1})_a \frac{(x_{k+1}^T U_{k+1})_a}{(v_{k+1} U_{k+1}^T U_{k+1})_a} \\ (U_{k+1})_{ia} &\leftarrow (U_{k+1})_{ia} \frac{(X_k V_k + x_{k+1} v_{k+1})_{ia}}{(U_{k+1} V_k^T V_k + U_{k+1} v_{k+1}^T v_{k+1})_{ia}} \end{aligned}$$

由上式可知, 基矩阵的更新计算只与  $X_k V_k + x_{k+1} v_{k+1}$  和  $V_k^T V_k + v_{k+1}^T v_{k+1}$  有关。当输入新样本后, 原始矩阵  $X_k$  和系数矩阵  $V_k$  的维数都会增加, 但是  $X_{k+1} V_{k+1}$ ,  $V_{k+1}^T V_{k+1}$  的维数是

固定的, 它们所需要的存储空间也是不变的, 因此降低了存储空间的要求。

### 3.3 基于图正则化和稀疏约束的增量型非负矩阵分解

以上改进的 NMF 算法都存在着一一定的缺陷, 稀疏约束 NMF 算法只是简单地对 NMF 的系数矩阵进行了稀疏。 Bucak 等人提出的 INMF 虽然利用了上一步的分解结果, 降低了重新学习阶段的运算时间, 但是无法保持样本所在空间的几何结构。蔡登教授等人提出的 GNMF 算法虽然将流形中的图正则与 NMF 相结合实现了保持数据空间几何结构不变, 但是很显然当新样本加入时不可避免地需要重复计算, 从而限制了该算法的应用。针对以上各改进算法的优缺点, 本文综合 INMF 算法、GNMF 算法和稀疏约束, 提出了基于图正则化和稀疏约束的增量型非负矩阵分解(GINMFSC)算法。在 INMF 算法中加入图正则化, 不仅能在低维空间保持样本的几何结构信息, 还能结合增量学习充分利用上一步的分解结果, 避免了重复计算, 从而降低了运算时间。在基于图正则化的增量型非负矩阵分解算法的基础之上对系数矩阵进行  $l_p$  ( $p=2$ ) 范数稀疏约束, 从而达到提高数据稀疏度的效果, 节省了存储空间, 提高了分解质量。下面介绍该算法。

设  $U_k$  和  $V_k$  分别表示样本集合  $X_k$  非负矩阵分解后得到的基矩阵和系数矩阵,  $L_k$  表示样本集合  $X_k$  的拉普拉斯矩阵, 其中  $k$  表示当前样本集合中存在的样本数。  $F_k$  代表  $X_k$  经过非负矩阵分解后的目标函数, 则  $F_k$  的表达式如下所示:

$$\begin{aligned} F_k &= \|X - U_k V_k^T\|_F^2 + \lambda \text{Tr}(V_k^T L_k V_k) + \beta \|V_k\|_2^2 \\ &= \sum_{i=1}^m \sum_{j=1}^k (X_{ij} - (U_k V_k^T)_{ij})^2 + \lambda \text{Tr}(V_k^T L_k V_k) + \\ &\quad \beta \|V_k\|_2^2 \\ \text{s. t. } &U \geq 0, V \geq 0 \end{aligned} \quad (11)$$

其中,  $\lambda$  是正则化参数,  $\beta$  是稀疏系数,  $\beta \in (0, 1)$ 。通过对系数矩阵  $V_k$  做稀疏约束可以使样本数据分解后具备较高的稀疏性, 且稀疏程度与  $\beta$  的值成正比。

当第  $k+1$  个样本加入后, 目标函数变为:

$$\begin{aligned} F_{k+1} &= \|X - U_{k+1} V_{k+1}^T\|_F^2 + \lambda \text{Tr}(V_{k+1}^T L_{k+1} V_{k+1}) + \\ &\quad \beta \|V_{k+1}\|_2^2 \\ &= \sum_{i=1}^m \sum_{j=1}^{k+1} (X_{ij} - (U_{k+1} V_{k+1}^T)_{ij})^2 + \lambda \text{Tr}(V_{k+1}^T L_{k+1} \\ &\quad V_{k+1}) + \beta \|V_{k+1}\|_2^2 \end{aligned} \quad (12)$$

其中,  $U_{k+1}$  和  $V_{k+1}$  分别表示样本集合  $X_{k+1}$  非负矩阵分解后得到的基矩阵和系数矩阵,  $L_{k+1}$  是样本集合  $X_{k+1}$  的拉普拉斯矩阵。

GINMFSC 算法假设当新样本  $x_{k+1}$  到来时, 系数矩阵  $V_{k+1}$  的前  $k$  列向量近似等于  $V_k$  的列向量, 即  $V_{k+1} = [V_k, v_{k+1}]$ 。目标函数  $F_{k+1}$  可以重写为如下形式:

$$\begin{aligned} F_{k+1} &= \sum_{i=1}^m \sum_{j=1}^{k+1} (X_{ij} - (U_{k+1} V_{k+1}^T)_{ij})^2 + \lambda \text{Tr}(V_{k+1}^T L_{k+1} \\ &\quad V_{k+1}) + \beta \|V_{k+1}\|_2^2 \\ &\approx \sum_{i=1}^m \sum_{j=1}^k (X_{ij} - (U_k V_k^T)_{ij})^2 + \sum_{i=1}^m ((x_{k+1})_i - (U_{k+1} \\ &\quad v_{k+1}^T)_i)^2 + \lambda \text{Tr}(V_{k+1}^T L_{k+1} V_{k+1}) + \beta \sum_{i=1}^m \sum_{j=1}^k (V_{k+1})_{ij}^2 \\ &\approx F_k + \sum_{i=1}^m ((x_{k+1})_i - (U_{k+1} v_{k+1}^T)_i)^2 + \lambda [\text{Tr}(V_{k+1}^T L_{k+1} \\ &\quad V_{k+1}) + \beta \|V_{k+1}\|_2^2] \end{aligned}$$

$$L_{k+1}V_{k+1}) - Tr(V_k^T L_k V_k)] + \beta \sum_{i=1}^m (\nu_{k+1})_i^2$$

$$= F_k + f_{k+1} \quad (13)$$

其中,  $f_{k+1}$  的表达式如下:

$$f_{k+1} = \sum_{i=1}^m ((x_{k+1})_i - (U_{k+1} \nu_{k+1}^T)_i)^2 + \lambda [Tr(V_{k+1}^T L_{k+1} V_{k+1}) - Tr(V_k^T L_k V_k)] + \beta \sum_{i=1}^m (\nu_{k+1})_i^2$$

在得到了目标函数  $F_{k+1}$  的增量式表达式后,利用梯度下降法,可以推导出式(13)对应的迭代更新公式:

$$(\nu_{k+1})_a \leftarrow \frac{(\nu_{k+1})_a (x_{k+1}^T U_{k+1} + \lambda \nu_{k+1} (R_{k+1})_{k+1, \cdot})_a}{(\nu_{k+1} U_{k+1}^T U_{k+1})_a + \lambda (\nu_{k+1} (D_{k+1})_{k+1, \cdot})_a + \beta (\nu_{k+1})_a} \quad (14)$$

$$(U_{k+1})_{ia} \leftarrow \frac{(U_{k+1})_{ia} (X_k V_k + x_{k+1} \nu_{k+1})_{ia}}{(U_{k+1} V_k^T V_k + U_{k+1} \nu_{k+1}^T \nu_{k+1})_{ia}} \quad (15)$$

迭代更新式(14)和式(15)的推导证明过程如下。

首先求出系数矩阵增量部分  $\nu_{k+1}$  的迭代公式:

$$(\nu_{k+1})_a \leftarrow (\nu_{k+1})_a - \eta_a \frac{\partial F_{k+1}}{\partial (\nu_{k+1})_a} \quad (16)$$

接下来计算  $F_{k+1}$  对  $\nu_{k+1}$  的偏导数:

$$\frac{\partial F_{k+1}}{\partial (\nu_{k+1})_a} \approx \frac{\partial}{\partial (\nu_{k+1})_a} (F_k + f_{k+1}) = \frac{\partial f_{k+1}}{\partial (\nu_{k+1})_a}$$

$$= 2 \sum_{i=1}^m ((x_{k+1})_i - (U_{k+1} \nu_{k+1}^T)_i) (-U_{k+1})_{ia} + 2(\lambda \nu_{k+1} (L_{k+1})_{k+1, \cdot})_a + 2\beta (\nu_{k+1})_a$$

$$= -2(x_{k+1}^T U_{k+1})_a + 2(\nu_{k+1} U_{k+1}^T U_{k+1})_a + 2(\lambda \nu_{k+1} (L_{k+1})_{k+1, \cdot})_a + 2\beta (\nu_{k+1})_a$$

式(16)中的  $\eta_a$  是迭代的步长,表达式为:

$$\eta_a = \frac{(\nu_{k+1})_a}{2(\nu_{k+1} U_{k+1}^T U_{k+1})_a + 2(\lambda \nu_{k+1} (D_{k+1})_{k+1, \cdot})_a + 2\beta (\nu_{k+1})_a}$$

其中,  $(L_{k+1})_{k+1, \cdot}$  是  $L_{k+1}$  所对应的第  $k+1$  行的行向量。同样地,  $(D_{k+1})_{k+1, \cdot}$  和  $(R_{k+1})_{k+1, \cdot}$  分别为  $D_{k+1}$  和  $R_{k+1}$  所对应的第  $k+1$  行的行向量。

将公式  $\eta_a$  和  $\partial F_{k+1} / \partial (\nu_{k+1})_a$  的值代入迭代式(16),得到  $\nu_{k+1}$  的迭代式(14)。

同样地,  $U_{k+1}$  的迭代公式也可以用梯度下降法来推导证明:

$$(U_{k+1})_{ia} \leftarrow (U_{k+1})_{ia} - \eta_{ia} \frac{\partial F_{k+1}}{\partial (U_{k+1})_{ia}} \quad (17)$$

在式(17)中,  $\partial F_{k+1} / \partial (U_{k+1})_{ia}$  的推导如下:

$$\frac{\partial F_{k+1}}{\partial (U_{k+1})_{ia}} = \frac{\partial}{\partial (U_{k+1})_{ia}} \sum_{i=1}^m \sum_{j=1}^{k+1} (X_{ij} - (U_{k+1} V_{k+1}^T)_{ij})^2$$

$$= 2 \sum_{j=1}^{k+1} (X_{ij} - (U_{k+1} V_{k+1}^T)_{ij}) (-V_{k+1})_{ja} + (U_{k+1} \nu_{k+1}^T \nu_{k+1} - x_{k+1} \nu_{k+1})_{ia}$$

$$= 2(U_{k+1} V_k^T V_k - X_k V_k)_{ia} + 2(U_{k+1} \nu_{k+1}^T \nu_{k+1} - x_{k+1} \nu_{k+1})_{ia}$$

式(17)中的  $\eta_{ia}$  是迭代的步长,表达式为:

$$\eta_{ia} = \frac{(U_{k+1})_{ia}}{2(U_{k+1} V_k^T V_k)_{ia} + 2(U_{k+1} \nu_{k+1}^T \nu_{k+1})_{ia}}$$

将公式  $\eta_{ia}$  和  $\partial F_{k+1} / \partial (U_{k+1})_{ia}$  的值代入迭代式(17),得到  $U_{k+1}$  的迭代式(15)。证毕。

GINMFSC 算法的具体步骤如下。

输入:数据集矩阵  $X$

输出:训练集分解后的系数矩阵  $V_{train}$

1. 初始化参数。设定参数  $\beta \in (0, 1)$ 、分解维度  $r$  和最大迭代次数 ( $nIterMax$ )。  $X_{m \times n}$  分为训练集和测试集,并随机初始化基矩阵  $U_{m \times r}$  和系数矩阵  $V_{n \times r}$ ;
2. FOR  $nI=1:nIterMax$  ( $nI$  是迭代次数);
3. 运用迭代规则式(14)和式(16)进行迭代运算;
4. 根据式(12)计算最小化目标函数;
5. END FOR;
6. 输出训练集分解后的系数矩阵  $V_{train}$ 。

## 4 实验与结果分析

实验环境为 Windows 7, CPU 为 2.60GHz, 内存为 4GB, 程序环境为 Matlab R2012b。

### 4.1 数据集

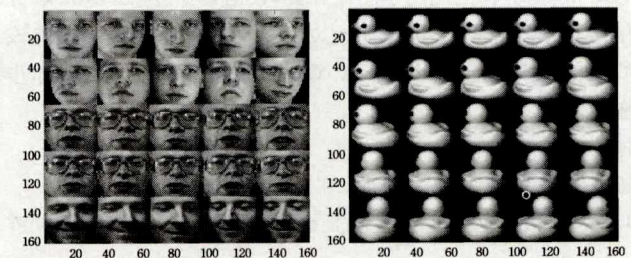
本文在 3 个常用数据集上进行了实验, 3 个数据集分别为: ORL 数据集、COIL20 数据集和 PIE-pose27 数据集。

(1) ORL 人脸数据库是由剑桥大学 AT&T 实验室拍摄的, 包含 40 个不同的人, 每人 10 幅图像, 共计 400 幅灰度图像, 每幅灰度图像有 256 个灰度级, 分辨率为  $112 \times 92$ 。ORL 人脸图像部分表情和细节有变化, 例如生气或高兴、戴或不戴眼镜、睁眼或闭眼等。人脸的姿态也有相当程度的不同, 如低头和抬头等。

(2) COIL20 数据集是由哥伦比亚大学创建的, 包含 20 个不同的对象(如招财猫、杯子、玩具小鸭等), 每个物体在水平方向旋转  $360^\circ$ , 每隔  $5^\circ$  拍摄一幅图片, 因此每个物体总共 72 幅图片, 整个数据库总共 1440 幅图片。

(3) PIE-pose27 数据集是由美国卡耐基梅隆大学创建的, 包含 68 名志愿者的面部表情图像, 每个人的姿态和光照变化都是在严格控制条件下采集的。PIE-pose27 数据集共有 42 种不同的光照条件, 共计 2856 幅图片。

ORL 数据集、COIL20 数据集和 PIE-pose27 数据集的部分图片如图 1 所示, 对比情况如表 1 所列。



(a) ORL

(b) COIL20

(c) PIE-pose27

图 1 3 个数据库中的图像示例

表1 数据集信息

数据集	图片数/n	维度/m	类别/class
ORL	400	1024	40
COIL20	1440	1024	20
PIE-pose27	2856	1024	68

## 4.2 参数选择

从数据集中选取  $k(k=2,3,\dots,10)$  类数据进行实验。然后每个类中随机选取样本比例为  $p(p=10\%,20\%,\dots,90\%)$  的样本作为训练集数据,剩余数据作为测试集数据,随着  $p$  值的增大分类准确率也会增大,本实验取  $p=60\%$ 。稀疏约束系数  $\beta$  的值由实验的结果确定,不同数据集的  $\beta$  值不同,但是  $\beta$  不宜过大。根据实验经验,分别将样本降到  $r(r=20,30,\dots,100)$  维,选择正则项参数  $\lambda$  为 100,最大迭代次数为 200。

对于 ORL 数据集,每个类中取 6 个共 240 个样本作为训练样本,余下的作为测试样本。在 240 个训练样本中取 200 个样本作为初始训练样本,剩余的 40 个样本作为新增训练样本,设置稀疏约束系数  $\beta$  为 0.3。

对于 COIL20 数据集,每个类中取 43 个共 860 个样本作为训练样本,余下的作为测试样本。在 860 个训练样本中取 700 个样本作为初始训练样本,剩余的 160 个样本作为新增训练样本,设置稀疏约束系数  $\beta$  为 0.4。

对于 PIE-pose27 数据集,每个类中取 25 个共 1700 个样本作为训练样本,剩下的作为测试样本。在 1700 个训练样本中取 1400 个样本作为初始训练样本,剩余的 300 个样本作为新增训练样本,设置稀疏约束系数  $\beta$  为 0.2。

本文实验采用支持向量机(SVM)进行分类,SVM 采用的是 LibSVM 工具箱。利用 LibSVM 进行分类预测实验时,需要调节两个重要参数——惩罚函数  $c$  和核函数  $g$ 。本实验采用了 K-fold Cross Validation (K-CV) 确定最佳参数值,即  $c=2, g=1.52$ 。

## 4.3 实验结果

本文将基于图正则化和稀疏约束的增量型非负矩阵分解(GINMFSC)算法应用于图像分类,数据集经过 GINMFSC 分解成基矩阵和系数矩阵,通过对系数矩阵进行聚类可检验该算法的识别性能。GINMFSC 算法的图像分类模型的具体步骤如下。

输入:数据集矩阵  $X$

输出:数据集的基图像、分类准确率

1.  $X_{m \times n}$  分为训练集和测试集,并随机初始化基矩阵  $U_{m \times r}$  和系数矩阵  $V_{n \times r}$ ;
2. 利用 NMF 的算法步骤求出基矩阵  $U_{\text{train}}$  和系数矩阵  $V_{\text{train}}$ ;
3. 利用训练集得到的系数矩阵  $V_{\text{train}}$  训练 SVM 分类器,从而得到分类模型;
4. 测试集经 GINMFSC 算法得到系数矩阵  $V_{\text{test}}$ ,利用分类模型对其进行分类,最终得到基图像和图像分类准确率。

本节选择了 NMF,GNMF,INMF 和 IGNNMF 作为 GINMFSC 算法的对比对象进行实验。为了得到更加合理的结果,对每个  $r$  值均运行 20 次后取平均值作为最终结果。

表 2—表 4 是几种算法在 3 个数据集上不同  $r$  值时的分类准确率。

表2 不同算法在 ORL 数据集上分类的准确率/%

$r$	NMF	GNMF	INMF	IGNMF	GINMFSC
20	65.12	67.19	70.28	74.28	75.48
30	68.48	73.85	74.27	75.01	78.64
40	70.16	78.57	78.43	78.98	79.30
50	73.53	75.16	78.26	78.62	86.62
60	75.21	80.25	80.43	84.78	85.67
70	78.99	81.85	83.69	86.32	85.08
80	79.83	84.70	85.95	85.86	87.89
90	82.35	85.58	86.59	88.41	89.80
100	83.61	84.03	85.06	85.50	88.85
Avg.	75.25	79.02	80.33	81.97	84.15

由表 2 可知,在 ORL 数据集中,GINMFSC 算法比 NMF 算法的分类准确率提高了 8.9%,比 GNMF 算法的分类准确率提高了 5.13%,比 INMF 算法的分类准确率提高了 3.82%,比 IGNNMF 算法的分类准确率提高了 2.18%。

表3 不同算法在 COIL20 数据集上分类的准确率/%

$r$	NMF	GNMF	INMF	IGNMF	GINMFSC
20	67.08	75.27	78.12	78.82	79.74
30	77.50	80.02	83.56	84.26	84.25
40	81.53	84.49	80.58	83.46	86.34
50	82.22	83.10	84.72	85.07	87.62
60	84.03	85.06	86.92	85.42	88.31
70	83.88	85.53	85.69	87.38	87.96
80	84.58	86.22	88.54	86.80	87.50
90	85.27	86.80	85.76	87.62	88.89
100	84.58	88.43	88.65	89.12	89.93
Avg.	81.18	83.88	84.73	85.33	86.73

由表 3 可知,在 COIL20 数据集中,GINMFSC 算法比 NMF 算法的分类准确率提高了 5.55%,比 GNMF 算法的分类准确率提高了 2.85%,比 INMF 算法的分类准确率提高了 2%,比 IGNNMF 算法的分类准确率提高了 1.4%。

表4 不同算法在 PIE-pose27 数据集上分类的准确率/%

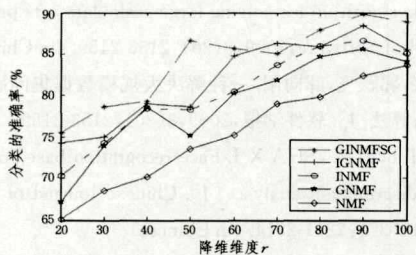
$r$	NMF	GNMF	INMF	IGNMF	GINMFSC
20	61.76	64.84	65.21	66.95	70.31
30	65.16	67.96	68.69	71.30	78.90
40	68.54	71.09	75.65	76.52	80.46
50	71.91	75.01	78.26	79.13	82.81
60	73.03	77.34	79.13	80.86	85.94
70	77.53	82.03	83.47	82.60	90.59
80	75.28	79.68	80.86	81.74	87.56
90	78.65	84.37	84.30	84.35	89.84
100	80.89	81.25	82.60	83.47	87.50
Avg.	72.53	79.95	77.57	78.55	83.77

由表 4 知在 PIE-pose27 数据集中,GINMFSC 算法比 NMF 算法的分类准确率提高了 11.24%,比 GNMF 算法的分类准确率提高了 7.82%,比 INMF 算法的分类准确率提高了 6.2%,比 IGNNMF 算法的分类准确率提高了 5.22%。

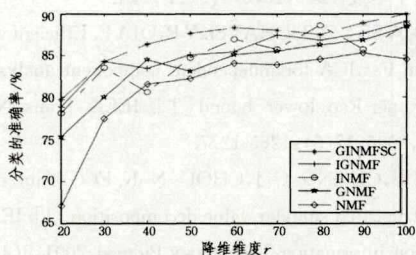
由表 2—表 4 可知,GINMFSC 算法的分类准确率高于其他对比算法。为了更加直观地说明 GINMFSC 算法的分类效果,图 2 给出了 3 个数据集上的分类结果曲线图。

由图 2 可以看出,随着  $r$  值的增加,几种算法的分类准确率大致呈上升趋势,有时会因为实验环境等因素的影响出现

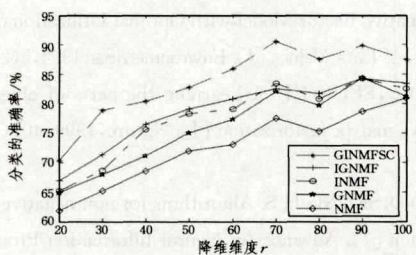
不稳定的情况。与 NMF,GNMF,INMF 和 IGNMF 算法相比,GINMFSC 算法的分类准确率相对较高,但偶尔也会出现一定的波动。



(a)ORL



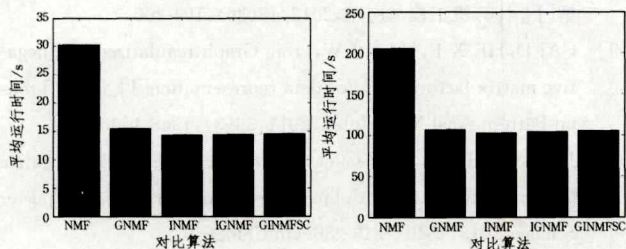
(b)COIL20



(c)PIE-pose27

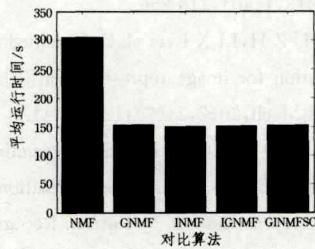
图 2 3 个数据集上的分类准确率

图 3 为 GINMFSC 算法及其对比算法在 ORL 人脸数据库、COIL20 人脸数据库和 PIE-pose27 数据库上的平均运行时间。



(a)ORL

(b)COIL20



(c)PIE-pose27

图 3 3 个数据集上的平均运行时间

从图 3 可以看出,使用增量 NMF 后,时间消耗比标准

NMF 算法有明显的改善。从时间复杂度上分析,改进的 NMF 算法的时间主要用在矩阵分解的迭加上,其他步骤则不会超过 NMF 算法的时间复杂度。而增量 NMF 算法的主要目的是在时间复杂度允许的范围内,使 NMF 算法能够动态地形成其特征矩阵和系数矩阵,并且使相应的目标函数在整个过程中不会增加。又由于流形学习的引入,使得本文提出的 GINMFSC 算法的平均运行时间远远小于 NMF 算法的平均运行时间。同时,从图 3 中不难看出,GINMFSC 与 GNMF,INMF 和 IGNMF 算法的平均运行时间相差不多。但是,由于 GINMFSC 算法加入了稀疏约束,其计算复杂度增加,因此导致其平均运行时间略长于 IGNMF 算法。因为稀疏约束的引入使得 GINMFSC 算法能够得到更好的基于部分的表示,所以其分类的准确率依然是几种算法中最高的。因此,综合考虑分类准确率和运行时间,本文提出的基于图正则化和稀疏约束的增量型非负矩阵分解算法取得了较好的分类效果。

#### 4.4 稀疏度量

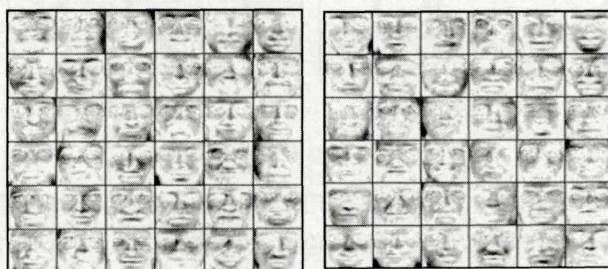
首先定义度量稀疏度的函数为<sup>[17]</sup>:

$$sparseness(x) = \frac{1}{n-1} [n - (\frac{\|x\|_1}{\|x\|_2})^2] \quad (18)$$

其中,  $n$  是向量  $x$  的维度,  $\|\cdot\|_1$  是向量的 1 范数,  $\|\cdot\|_2$  是向量的 2 范数,  $0 \leq sparseness \leq 1$ 。当  $x$  仅有一个非零元素时,  $sparseness(x) = 1$ ; 当所有元素都不为零且相等时,  $sparseness(x) = 0$ 。

本节实验采用 GNMF,INMF,IGNMF 和 GINMFSC 算法对原始的非负矩阵  $X$  进行矩阵分解得到基图像,用式(18)来计算稀疏度。

图 4—图 6 分别表示 3 个数据集上的基图像。其中, ORL 数据集的基矩阵特征维数取 36,COIL20 和 PIE-pose27 数据集的基矩阵特征维数取 20。



(a)GNMF(稀疏度 0.4496)

(b)INMF(稀疏度 0.4739)



(c)IGNMF(稀疏度 0.4977)

(d)GINMFSC(稀疏度 0.5187)

图 4 ORL 数据集的基图像

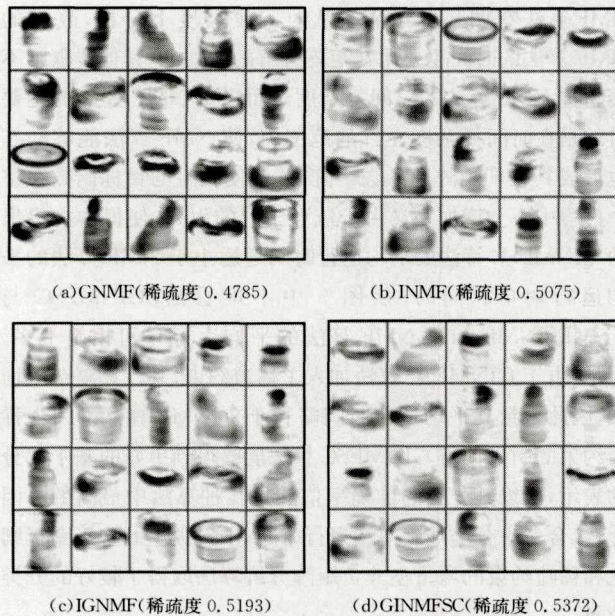


图5 COIL20数据集的基图像

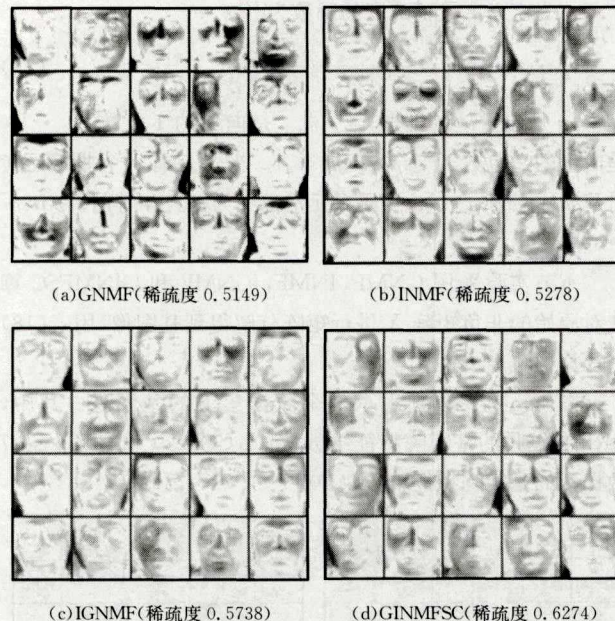


图6 PIE-pose27数据集的基图像

**结束语** 本文提出了基于图正则化和稀疏约束的增量型非负矩阵分解,并给出了相应的迭代求解公式。在 ORL, COIL20 和 PIE-pose27 3 个图像数据库上,对提出的新方法进行了实验,利用分类的准确率和平均运行时间来衡量算法的识别能力。从实验结果可以看出,GINMFSC 算法明显优于标准的 NMF 算法和其他几种改进的 NMF 算法,说明了 GINMFSC 算法的分类效果和有效性。最后,考察了算法的稀疏度,结果显示提出的 GINMFSC 算法稀疏度最高,得出了更好的图像表示。因此,该算法能够得到图像的最佳局部表示,使得基图像具有更好判别能力。但是,算法中依然存在着一定的问题,例如迭代初始值的选取问题、如何有效地选取参数  $\lambda$  等,目前仍然没有统一的标准。这些问题将成为今后研究的重点。

## 参考文献

- [1] SHI W Y, GUO Y F, XUE X Y. A kernel principal component analysis algorithm for solving large scale data sets problem [J]. Journal of Software, 2009, 8(20): 2153-2159. (in Chinese)  
史卫亚, 郭跃飞, 薛向阳. 一种解决大规模数据集问题的核主成分分析算法[J]. 软件学报, 2009, 8(20): 2153-2159.
- [2] YIN H T, FU P, SHA X J. Face recognition based on DCT and linear discriminant analysis [J]. Chinese Journal of Electronic, 2009, 10(37): 2211-2214. (in Chinese)  
尹洪涛, 付平, 沙学军. 基于 DCT 和线性判别分析的人脸识别[J]. 电子学报, 2009, 10(37): 2211-2214.
- [3] KOLDOVSKY Z, TICHAVSKY P, OJA E. Efficient variant of algorithm FastICA for independent component analysis attaining the Cramér-Rao lower bound [J]. IEEE Trans Neural Networks, 2006, 17(5): 1265-1277.
- [4] WEI J J, CHANG C J, CHOU N K. ECG data compression using truncated singular value decomposition [J]. IEEE Transactions on Information Technology Biomed, 2001, 5(4): 290-299.
- [5] PAATERO P, TAPPER U. Positive Matrix Factorization: A Nonnegative Factor Model with Optimal Utilization of Error Estimates of Data Values [J]. Environmetrics, 1994, 5(2): 111-126.
- [6] LEE D D, SEUNG H S. Learning the parts of objects by non-negative matrix factorization [J]. Nature, 1999, 401(6755): 788-791.
- [7] LEE D D, SEUNG H S. Algorithms for non-negative matrix factorization [J]. Advances in Neural Information Processing Systems, 2001, 13: 556-562.
- [8] DU S Q, SHI Y Q, WANG W L, et al. Graph regularized-based semi-supervised non-negative matrix factorization [J]. Computer Engineering and Applications, 2012, 48(36): 194-200. (in Chinese)  
杜世强, 石玉清, 王维兰, 等. 基于图正则化的半监督非负矩阵分解[J]. 计算机工程与应用, 2012, 48(36): 194-200.
- [9] CAI D, HE X F, HAN J W, et al. Graph regularized non-negative matrix factorization for data representation [J]. IEEE Trans on Pattern Anal Mach Intell, 2011, 33(8): 1548-1560.
- [10] JIANG W, LI H, YU Z G, et al. Graph regularized Non-negative Matrix Factorization with Sparseness Constraints [J]. Computer Science, 2013, 1(40): 218-256. (in Chinese)  
姜伟, 李宏, 于震国, 等. 稀疏约束图正则非负矩阵分解 [J]. 计算机科学, 2013, 1(40): 218-256.
- [11] LIU H F, WU Z H, LI X L, et al. Constrained non-negative matrix factorization for image representation [J]. IEEE Trans on Pattern Anal Mach, 2012, 34(7): 1299-1311.
- [12] SHU Z Q, ZHAO C X. Constrained non-negative matrix factorization algorithm based on graph regularization and its application in image representation [J]. Pattern Recognition and Artificial Intelligence, 2013, 26(3): 300-306. (in Chinese)  
舒振球, 赵春霞. 基于图正则化的受限非负矩阵分解算法及其在图像表示中的应用 [J]. 模式识别与人工智能, 2013, 26(3): 300-306.

- [13] BUCAK S S, GUNSEL B. Incremental subspace learning via non-negative matrix factorization [J]. Pattern Recognition, 2009, 42(5):788-797.
- [14] YU Z Z, LIU Y H, B Li, et al. Incremental Graph Regulated Nonnegative Matrix Factorization [J]. Journal of Applied Mathematics, 2014, 2014(1):1-10.
- [15] DING C, LI T, PENG W, et al. Orthogonal nonnegative matrix t-factorizations for clustering [C] // Proceedings of SIGKDD. 2006:126-135.
- [16] LOVASZ L, PLUMMER M. Matching Theory [M]. North Holland, 1986.
- [17] MICHAEL W, SHAKHINA A, STEWART G W. Computing sparse reduced-rank approximations to sparse matrices [J]. ACM Transactions on Mathematical Software, 2004, 19(3):231-235.
- [18] WANG W L, CAI J. An incremental learning algorithm for non-negative matrix factorization under sparse constraints [J]. Computer Science, 2014, 41(8):241-244. (in Chinese)  
王万良, 蔡竞. 稀疏约束下非负矩阵分解的增量学习算法 [J]. 计算机科学, 2014, 41(8):241-244.

(上接第 297 页)

除干扰角点,其实验结果如图 8(b)所示。给定  $\theta=162^\circ$ ,检测到 12 个角点,并依次连线。与凸包算法相比,该算法可以检测到图形的关键点位置,得到图形的大致形状。但是其并没有起到拟合的效果,原有的边缘弯曲及缺失仍然存在。

从图 8(c)中可以看出,本文算法可以得到关键角点,大大弱化了边缘的弯曲度。在提取的原有边界的基础上,提升了拟合效果,使得最终形状恢复至原图效果。与前两种方法相比,本文方法成功地去除了边界的毛刺<sup>[9]</sup>,同时保证了多边形的每个角度都是  $90^\circ$ ,与原图保持一致。因此,本文方法是一种有效的拟合方式。

#### 4.2 时间复杂度对比

通过上面 3 组算法的对比,可以直观地看出基于最小外接矩形的多边形拟合算法的效果提升。同时,由于算法的思路简单,使得计算具有高效性。本文无需计算轮廓点的曲率值,转化坐标系后直接进行简单作差比较,耗时很少。经测试,本算法利用 VS2010+OPENCV 平台,在 Intel Core i5-2320 CPU 上运行一幅图片平均耗时 8ms,如表 1 所列。类似地,测试凸包算法的运行耗时,大约为 16ms。针对求解曲率寻找角点的方法,得到一幅图的轮廓耗时需 1494ms。其中,有 1034ms 的耗时间用在计算边缘各点的曲率值上。由此可见,无需计算曲率大大提高了计算的效率。

表 1 运行耗时/ms

算法类型	本文算法	凸包检测	角点检测
耗时	8	16	1494

**结束语** 本文提出一种基于最小外接矩形的直角多边形拟合算法。针对直角角点的特征,利用了最小外接矩形作为轮廓的外界,并与轮廓作差值找到直角角点。根据矩形在平面内旋转,建立不同角度的笛卡尔坐标系,可以检测出平面内成任意角度的直角多边形形状,修复了原轮廓的局部凹凸性。所提方法拟合后的图形很好地贴近房屋的原始形状,且与传统的基于曲率的检测角点方法相比,具有计算上的高效性。通过测试不同的图片,其得到的准确性高。

#### 参 考 文 献

- [1] MA C Q. Using coordinate system of translation and rotation to simplify the housing axis positioning [J]. Science and Technology Innovation Herald, 2012(24):59, 61. (in Chinese)
- [2] XU H. Research on key points detection of profile curve [D]. Nanjing: Southeast University, 2009. (in Chinese)  
徐虹. 轮廓曲线的关键点检测研究 [D]. 南京:东南大学, 2009.
- [3] LI M, WANG J Z, LI L L, et al. Curvature statistics corner detection method [J]. Mind and Computation, 2009(3):209-214. (in Chinese)  
李明, 王甲卓, 李玲玲, 等. 曲率统计角点检测方法 [J]. 心智与计算, 2009(3):209-214.
- [4] HE X C, YUNG N H. Curvature scale space corner detector with adaptive threshold and dynamic region of support [C] // Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004). 2004, 2:791-794.
- [5] MOKHTARIAN F, MOHANNAN F. Enhancing the curvature scale space corner Detector [C] // Proc. Scandinavian Conf. on Image Analysis. 2001:145-152.
- [6] HU X M, SHEN Z X, WANG S Y, et al. Remote gesture recognition based on polygon bump detection [J]. Information communication, 2015(9):57-58. (in Chinese)  
胡学敏, 沈朝晓, 王司雨, 等. 基于多边形凹凸点检测的远距离手势识别 [J]. 信息通信, 2015(9):57-58.
- [7] LI Z H, JIANG G H, XU S N, et al. Infrared and visible image registration algorithm based on contour polygon fitting [J]. System Engineering and Electronic Technology, 2015, 37(12):2872-2878. (in Chinese)  
李振华, 江耿红, 徐胜男, 等. 基于轮廓多边形拟合的红外与可见光图像配准算法 [J]. 系统工程与电子技术, 2015, 37(12):2872-2878.
- [8] SUN S Z, GU X C, YANG Y, et al. Research on contour extraction and polygon fitting of aircraft image [J]. Journal of Changchun University of Science and Technology (Natural Science Edition), 2009, 32(3):447-449. (in Chinese)  
孙爽滋, 谷欣超, 杨勇, 等. 飞机图像的轮廓提取与多边形拟合研究 [J]. 长春理工大学学报(自然科学版), 2009, 32(3):447-449.
- [9] WANG Y, HAO C Y, HAN P Y, et al. Polygonal Fitting Method to Image Contour Based on Mergence [C] // National Joint Conference on signal and Information Processing Technology and Information Processing. 2004. (in Chinese)  
王毅, 郝重阳, 韩培友, 等. 基于聚合的图象轮廓线多边形拟合方法 [C] // 信号与信息处理技术信号与信息处理全国联合学术会议. 2004.