控制与决策
Control and Decision

全相似高阶规范割算法研究

张敬茂，沈艳霞

文章编号: 1001-0920(2020)04-0852-09
DOI: 10.13195/j.kzyjc.2018.0938

全相似高阶规范割算法研究

张敬茂, 沈艳霞†

(江南大学 物联网技术与应用教育部工程研究中心，江苏 无锡 214122)

摘 要: 已有的高阶算法中,构建相似模型时仅使用少量超边构建稀疏相似模型,同时高阶相似模型仅考虑使用单阶的高阶相似关系. 为解决这两个问题,以规范割算法为基础,采用直推式学习技术,从标准化和非标准化拉氏矩阵两个角度分别构建全相似高阶模型和全相似多阶相似模型. 根据规范割算法构建直推式学习框架,然后展示该框架如何在算法中训练全相似关系. 研究结果显示,在所提出的算法中超边之间的全相似关系能以一个简洁的形式应用. 以此为基础,将多阶全相似关系进行融合,提出融合多阶信息的全相似多阶相似模型. 将构建的全相似高阶相似模型和全相似多阶相似模型应用到规范割算法框架中,提出全相似高阶规范割算法和全相似多阶规范割算法. 在两种高阶相似模型中,全相似张量采用稀疏张量逆的形式,并且该逆矩阵可以转换为规范割框架中稀疏张量特征分解问题. 将所提出的算法应用于运动分割,并与现有的高阶算法进行对比,实验结果显示,所提出的算法具有一定的优势.

关键词: 全相似高阶模型；全相似多阶模型；直推式学习；规范割；运动分割

中图分类号: TP301.6　　**文献标志码:** A

Studying full higher order affinity normalized cut

ZHANG Jing-mao, SHEN Yan-xia†

(Engineering Research Center of IoT Technology and Application of MOE，Jiangnan University，Wuxi 214122，China)

Abstract: In the higher-order algorithms, only a small number of super edges are used to construct the sparse affinity model, and the higher-order affinity model only considers the single-order higher-order similarity relationship. In order to solve these two problems, based on the normalized cut algorithm and by using the transductive learning technique, a full higher-order affinity model and a full multi-order affinity model are constructed from the two perspectives of normalized and non-normalized Laplacian matrixes. A Transductive learning framework is constructed based on the normalized cut algorithm, and it is showed how the framework trains the full similarity in the algorithm. The final result shows that the full similarity between the super edges in the algorithm can be applied as a simple form. Based on this, the multi-order full similarity relation is merged, and a fully similar multi-order affinity model with multi-order information is proposed. The constructed full higher-order affinity model and the full multi-order affinity model are applied to the framework of the normalized cut algorithm, and a full affinity higher-order normalized cut algorithm and a fully affinity multi-order normalized cut algorithm are proposed. In the two higher-order affinity models, the full affinity tensor is in the form of sparse tensor inverse and the inverse matrix can be transformed into the sparse tensor feature decomposition problem in the normalized cut. In the experimental part, the proposed algorithm is applied to motion segmentation and compared with the existing higher-order algorithms. The results show the advantages of the propesed algorithm.

Keywords: full higher order affinity model；full multi-orders affinity model；transductive inference；normalized cut；motion segmentation

0 引 言

谱聚类算法在机器学习中占有重要地位. 在谱聚类算法中,通常只使用数据之间成对的相似关系来构建相似模型,如规范割算法[1]. 但是在很多问题中,如混合线性建模问题[2],或者更具体的,如图匹配问题[3-4]、子空间聚类[5-6]、运动分割[7-9] 等,仅考虑成对的相似关系往往会导致较差的结果[4,8].

研究人员提出了一类高阶算法解决这些问题. 这类高阶算法的基本思想是将成对的相似度量替换为高阶相似度量(如3阶相似度量). 通常,超图技术用来处理这类相似度量问题. 近年来,大量相关工作已经发表. 文献[5]使用加权图来近似超图;文献

收稿日期: 2018-07-10；**修回日期:** 2018-11-08.

基金项目: 国家自然科学基金项目(61573167, 61572237)；江苏省研究生科研与实践创新计划项目(KYCX17_1454).

†通讯作者. E-mail: shenyx@jiangnan.edu.cn.

[8,10] 中定义了不同的优化准则; 文献 [11] 中提出使用 Ihara zeta 函数; 文献 [6,12] 中提出使用低秩矩阵代表; 文献 [13-14] 中对图切框架使用超图学习; 文献 [15] 中使用非合作博弈解决超图聚类问题. 这些成果已经应用到了很多场景之中, 如子空间聚类[6]、图像分割[16-17]、图匹配[3]、人脸聚类[2]、运动分割[7-9] 等, 在这些应用中, 高阶相似度量往往能够使实验效果显著提高. 目前, 也有对谱聚类中高级算法的理论分析研究工作[18-19].

通常, 为了构建高阶相似模型, 现有的高阶方法可以总结为以下4种:

1) 全数据模型. 全数据模型是指使用所有的数据来构建高阶相似模型, 模型将会计算所有的超边. 在这类模型中, 随着数据的增多, 计算和存储该模型都是十分不现实的. 所以这类模型往往仅用于理论分析[2,7].

2) 随机抽样模型. 随机抽样模型[20] 随机抽取所有高阶相似关系的一小部分来估计原始的高阶相似关系. 该模型在计算上十分高效, 并且不需要预先计算完整的原始高阶相似模型, 具有实际应用价值.

3) 受限随机抽样模型. 受限随机抽样模型是随机抽样模型的改进版本. 该模型根据一个调整参数来过滤掉部分随机抽取的高阶相似关系, 这种过滤是根据有价值的抽样往往包含有价值的信息这一事实进行的. 因此, 在该模型中, 价值权重较低的抽样将会被过滤掉.

4) 迭代抽样模型. 迭代抽样模型仅考虑高阶相似模型中纯净的高阶相似关系. 理想情况下, 一个完美的相似矩阵应该是分块对角结构的, 在算法中使用这种纯净的数据能够得到完美的聚类结果[21]. 在该模型中, 迭代地对数据进行抽样能够保证最终的相似矩阵以纯净的数据构建.

在这些构建高阶相似模型的方法中, 仅仅考虑了一阶的相似关系, 例如, 3阶相似矩阵仅考虑3阶相似关系. 另外, 除了用于理论分析的全数据模型, 仅少量的高阶相似关系用来构建稀疏高阶相似模型.

为了解决以上问题, 本文试图设计一种不使用全部数据的全相似高阶相似模型和一种融合多阶高阶相似关系的多阶相似模型. 本文的主要贡献描述如下:

1) 在规范割框架下, 构建一种新型的全相似高阶相似模型 (FHAM). 在该模型中, 仅使用少量的高阶相似关系来学习全相似高阶相似关系. 为了实现该目的, 本文使用分类问题中的直推式学习技术. 首先, 根据直推式学习框架构建一个价值函数; 然后, 从该价值函数中推导出 FHAM. 值得注意的是, FHAM 在这里是个稠密矩阵的形式, 在实际操作中是不实用的. 不过本文在规范割框架下对 FHAM 进行谱分析后发现, 它可以使用稀疏矩阵进行高效的特征分解, 从而可以避免对稠密矩阵的操作.

2) 在规范割框架下, 设计一种新型的全相似多阶模型 (FMAM). 在所有已知的高阶相似模型中, 仅考虑使用一阶的高阶相似关系 (如3阶相似模型). 考虑到不同阶的相似关系包含不同的聚类信息, 本文设计一种融合多阶关系的高阶相似模型, 如融合3阶和4阶相似关系, 或者更多阶的相似关系. 由于 FMAM 是在 FHAM 的基础上构建的, FMAM 也是一种全相似高阶相似模型. 以 FHAM 和 FMAM 为基础, 本文在规范割框架下提出两种算法: 全相似高阶规范割算法 (FHNcut) 和全相似多阶规范割算法 (FMNcut). 对于这两种算法, 本文分别设计两类算法: 标准化 (normalized) 的算法和非标准化 (unnormalized) 的算法. 在实验部分, 将所有设计的算法都应用到运动分割中. 实验结果显示本文算法具有一定的优势.

1 全相似高阶相似模型

在分类算法中, 直推式学习技术被用来学习标记和未标记问题, 能够通过整合标记的数据估计全相似关系. 本文将直推式学习技术用于高阶聚类问题以估计全相似高阶相似关系. 本文所提全相似高阶模型中, 将抽样得到的数据作为标记的数据, 剩余的数据作为未标记的数据.

1.1 学习全相似高阶相似模型

根据文献 [14], 直推式学习的一般框架可以表示为

$$\arg\min_{f}\{\Omega(f)+\lambda\Theta(f)\}. \tag{1}$$

其中: $\Omega(f)$ 为目标函数, $\Theta(f)$ 为经验损失, f 表示分类函数, $\lambda>0$ 为权衡参数.

对于非标准化拉式矩阵 L, 谱聚类目标函数 $\Omega(f)$ 定义为如下形式:

$$\Omega(f)=f^{\mathrm{T}}Lf. \tag{2}$$

其中: $L = D - W$ 为非标准化拉式矩阵, W 为加权矩阵, D 为 W 的对角矩阵. 目标函数 $\Omega(f)$ 中相似关系为成对相似形式. 为了得到高阶相似形式的目标函数, 本文使用了高阶奇异值分解 (SVD) 理论[22-23] 和该理论的应用[2,9,20]. 简要地说, 考虑一个张量矩阵 $\mathcal{P}$ 和它的展开矩阵 P, 则加权矩阵可以定义为 $W = PP^{\mathrm{T}}$, 选择使用最小二乘损失函数作为经验损失函

数$\Theta(f)$. 为了方便计算,损失函数$\Theta(f)$中增加了矩阵D. y表示初始的数据标记. 在本文中,对于抽样得到的数据,y设置为1,其余数据标记为0. 综上所述,$\Theta(f)$损失函数可定义为

$$\Theta(f)=D\|f-D^{-1}y\|^2. \tag{3}$$

因此,计算全相似高阶模型的价值函数定义为

$$\arg\min_f(f^{\mathrm{T}}Lf+\lambda D\|f-D^{-1}y\|^2). \tag{4}$$

将$L=D-W$代入式(4),可以得到

$$\begin{aligned}&\arg\min_f(f^{\mathrm{T}}Lf+\lambda D\|f-D^{-1}y\|^2)=\\&\arg\min_f(f^{\mathrm{T}}(D-W)f+\lambda D\|f-D^{-1}y\|^2)=\\&\arg\min_f\Big(\sum_{i=1}^n D_{ii}f_i^2-\sum_{i,j=1}^n W_{ij}f_if_j+\\&\qquad\lambda D\|f-D^{-1}y\|^2\Big)=\\&\arg\min_f\Big(\frac{1}{2}\sum_{i,j=1}^n W_{ij}(f_i-f_j)^2+\\&\qquad\lambda\sum_{i=1}^n D_{ii}\Big\|f_i-\frac{y_i}{D_{ii}}\Big\|^2\Big).\end{aligned} \tag{5}$$

为了方便表示,定义$Q(f)=f^{\mathrm{T}}Lf+\lambda D\times\|f-D^{-1}y\|^2$,则全相似高阶模型可表示为$F=\arg\min_f Q(f)$. 计算$Q(f)$对$f$的偏导,有

$$\begin{aligned}&\frac{\partial Q}{\partial F}\Big|_{f=F}=\frac{\partial}{\partial F}[f^{\mathrm{T}}Lf+\lambda D\|f-D^{-1}y\|^2]=\\&2LF+2\lambda D(F-D^{-1}y)=\\&2(D-W)F+2\lambda DF-2\lambda y=\\&2[(1+\lambda)DF-WF-\lambda y]=0,\end{aligned} \tag{6}$$

从而可得

$$F=\frac{\lambda}{1+\lambda}\Big(D-\frac{1}{1+\lambda}W\Big)y=\alpha(D-\beta W)^{-1}y. \tag{7}$$

显然,式(7)中$\beta=1-\alpha$. 因此,全相似高阶模型O定义为

$$O=\alpha(D-\beta W)^{-1}=\alpha Z^{-1}, \tag{8}$$

其中$Z=D-\beta W$为稀疏矩阵. 该稀疏矩阵的逆矩阵为稠密矩阵,导致O也为稠密矩阵. 这也导致对O的计算十分困难,而对O进行特征分解既不实用,也不切实际.

式(8)是从非标准化拉氏矩阵L推导得到的. 现在推导标准化拉氏矩阵的情况,首先将矩阵L替换为标准化拉氏矩阵$L_{\mathrm{sym}}=D^{-1/2}LD^{-1/2}$.

定义$Q_{\mathrm{sym}}(f)=f^{\mathrm{T}}L_{\mathrm{sym}}f+\lambda\|f-y\|^2$,计算其对$f$的偏导,有

$$\begin{aligned}&\frac{\partial Q_{\mathrm{sym}}}{\partial F}\Big|_{f=F}=\frac{\partial}{\partial F}[f^{\mathrm{T}}L_{\mathrm{sym}}f+\lambda\|f-y\|^2]=\\&2L_{\mathrm{sym}}F+2\lambda(F-y)=\\&2(I-D^{-1/2}WD^{-1/2})F+2\lambda F-2\lambda y=\\&2[(1+\lambda)F-D^{-1/2}WD^{-1/2}F-\lambda y]=0.\end{aligned} \tag{9}$$

可以得到

$$\begin{aligned}F=&\frac{\lambda}{1+\lambda}\Big(I-\frac{1}{1+\lambda}D^{-1/2}WD^{-1/2}\Big)y=\\&\alpha(I-\beta D^{-1/2}WD^{-1/2})^{-1}y.\end{aligned} \tag{10}$$

在标准化情况下,本文所提出的全相似高阶相似模型可以定义为

$$O_{\mathrm{sym}}=\alpha(I-\beta D^{-1/2}WD^{-1/2})^{-1}=\alpha Z_{\mathrm{sym}}{}^{-1}, \tag{11}$$

其中$Z_{\mathrm{sym}}=I-\beta D^{-1/2}WD^{-1/2}$也是稀疏矩阵. 因此,式(8)和(11)便构成全相似高阶模型.

1.2 全相似高阶规范割算法

规范割算法解决了如下广义特征值问题:

$$L\boldsymbol{u}=\varepsilon D\boldsymbol{u}. \tag{12}$$

其中:$\boldsymbol{u}$为前K个最小的广义特征向量,ε为前K个最小的特征值. 在非标准化拉氏矩阵$L=D-W$的情况下,式(12)能够转变为标准特征分解形式

$$D^{-1/2}WD^{-1/2}\boldsymbol{l}=(1-\varepsilon)\boldsymbol{l}, \tag{13}$$

其中$\boldsymbol{l}=D^{1/2}\boldsymbol{u}$. 将式(13)中矩阵$W$使用公式(8) FHAM的$O$代替. 进行简单的变换可得

$$\begin{aligned}&D^{1/2}ZD^{1/2}\boldsymbol{l}=\frac{\alpha}{(1-\varepsilon)}\boldsymbol{l}\Rightarrow\\&D^{1/2}(D_W-\beta W)D^{1/2}\boldsymbol{l}=\frac{\alpha}{(1-\varepsilon)}\boldsymbol{l}.\end{aligned} \tag{14}$$

其中:矩阵D是矩阵O的对角矩阵,矩阵D_W是矩阵W的对角矩阵. 由于Z是稀疏矩阵,式(14)所表示的特征分解问题可以高效解决. 而规范割算法所需要的广义特征向量可以简单地通过$\boldsymbol{u}=D^{-1/2}\boldsymbol{l}$得到. 对于标准化拉氏矩阵的情况,只需要将式(14)中的矩阵Z替换为矩阵Z_{sym}. 推导可得

$$\begin{aligned}&D^{1/2}Z_{\mathrm{sym}}D^{1/2}\boldsymbol{l}=\frac{\alpha}{(1-\varepsilon)}\boldsymbol{l}\Rightarrow\\&D^{1/2}(I-\beta D_W^{-1/2}WD_W^{-1/2})D^{1/2}\boldsymbol{l}=\frac{\alpha}{(1-\varepsilon)}\boldsymbol{l}.\end{aligned} \tag{15}$$

其中:矩阵D是矩阵O_{sym}的对角矩阵,矩阵D_W是矩阵W的对角矩阵.

根据上文分析,本文提出的全相似高阶规范割算法(FHNcut)的流程可总结如下.

算法1 全相似高阶规范割算法(FHNcut).

输入:数据集X,高阶相似矩阵阶数,聚类类别数

k,参数α和参数β;

输出: k类聚类结果.

step 1: 从数据集X中进行抽样,对抽样数据进行高阶相似度量;

step 2: 构建张量$\mathcal{P}$和其展开矩阵P,计算$W = PP^{\mathrm{T}}$, L或L_{sym};

step 3: 通过式(8)或(11)计算全相似高阶相似矩阵O或O_{sym};

step 4: 解决式(14)或(15)中的特征分解问题;

step 5: 通过$\boldsymbol{u} = D^{-1/2}\boldsymbol{l}$计算$k$个广义特征向量;

step 6: 使用k-means算法将$\boldsymbol{u}$聚类为k类.

2 全相似多阶相似模型

2.1 学习全相似多阶相似模型

根据前文描述,一阶相似模型是通过价值函数$Q(f)$得到的,对价值函数$Q(f)$求偏导,可以保证得到最优的F. 在本节,最优的F需要使用多阶相似数据得到. 当有$c = (c_1, c_2, \ldots, c_m)$阶数据时,定义直推式学习框架为

$$\arg\min_f [Q^{(c_1)}(f) + Q^{(c_2)}(f) + \ldots + Q^{(c_m)}(f)], \tag{16}$$

其中$Q^{(c_1)}(f)$代表c_1阶的$Q(f)$.

为了方便表示,定义$Q^{\mathrm{multi}}(f) = Q^{(c_1)}(f) + Q^{(c_2)}(f) + \ldots + Q^{(c_m)}(f)$. 在非标准化情况下,为了得到最优的$F$,计算$Q^{\mathrm{multi}}(f)$对$f$的偏导. 可以得到

$$\begin{aligned}
&\frac{\partial Q^{\mathrm{multi}}(f)}{\partial F}\Big|_{f=F} = \\
&\frac{\partial}{\partial F}[Q^{(c_1)}(f) + Q^{(c_2)}(f) + \ldots + Q^{(c_m)}(f)] = \\
&[2L^{(c_1)}F + 2\lambda D^{(c_1)}(F - (D^{(c_1)})^{-1}y)] + \\
&[2L^{(c_2)}F + 2\lambda D^{(c_2)}(F - (D^{(c_2)})^{-1}y)] + \ldots + \\
&[2L^{(c_m)}F + 2\lambda D^{(c_m)}(F - (D^{(c_m)})^{-1}y)] = \\
&2\Big[(1+\lambda)\sum_{i=1}^{m} D^{(c_i)}F - \sum_{i=1}^{m} W^{(c_i)}F - m\lambda y\Big] = 0.
\end{aligned} \tag{17}$$

式(17)可以转化为

$$\begin{aligned}
&2\Big[(1+\lambda)\sum_{i=1}^{m} D^{(c_i)}F - \sum_{i=1}^{m} W^{(c_i)}F - m\lambda y\Big] = 0 \Rightarrow \\
&(1+\lambda)\sum_{i=1}^{m} D^{(c_i)}F - \sum_{i=1}^{m} W^{(c_i)}F = m\lambda y \Rightarrow \\
&\sum_{i=1}^{m} D^{(c_i)}F - \frac{1}{1+\lambda}\sum_{i=1}^{m} W^{(c_i)}F = \frac{m\lambda}{1+\lambda}y \Rightarrow \\
&F = \frac{m\lambda}{1+\lambda}\Big(\sum_{i=1}^{m} D^{(c_i)} - \frac{1}{1+\lambda}\sum_{i=1}^{m} W^{(c_i)}\Big)^{-1} y.
\end{aligned} \tag{18}$$

因此,全相似多阶相似模型定义为

$$\begin{aligned}
O^{\mathrm{multi}} &= \frac{m\lambda}{1+\lambda}\Big(\sum_{i=1}^{m} D^{(c_i)} - \frac{1}{1+\lambda}\sum_{i=1}^{m} W^{(c_i)}\Big)^{-1} = \\
&m\alpha\Big(\sum_{i=1}^{m} D^{(c_i)} - \beta\sum_{i=1}^{m} W^{(c_i)}\Big)^{-1} = \\
&m\alpha(Z^{\mathrm{multi}})^{-1},
\end{aligned} \tag{19}$$

其中$Z^{\mathrm{multi}} = \sum_{i=1}^{m} D^{(c_i)} - \beta\sum_{i=1}^{m} W^{(c_i)}$.

对于标准化情况, $Q_{\mathrm{sym}}^{\mathrm{multi}}(f) = Q_{\mathrm{sym}}^{(c_1)}(f) + Q_{\mathrm{sym}}^{(c_2)}(f) + \ldots + Q_{\mathrm{sym}}^{(c_m)}(f)$.

计算$Q_{\mathrm{sym}}^{\mathrm{multi}}(f)$对$f$的偏导过程与非标准化情况类似,即

$$\begin{aligned}
&\frac{\partial Q_{\mathrm{sym}}^{\mathrm{multi}}(f)}{\partial F}|_{f=F} = \\
&\frac{\partial}{\partial F}[Q_{\mathrm{sym}}^{(c_1)}(f) + Q_{\mathrm{sym}}^{(c_2)}(f) + \ldots + Q_{\mathrm{sym}}^{(c_m)}(f)] = \\
&[2L_{\mathrm{sym}}^{(c_1)}F + 2\lambda D_{\mathrm{sym}}^{(c_1)}(F - (D_{\mathrm{sym}}^{(c_1)})^{-1}y)] + \\
&[2L_{\mathrm{sym}}^{(c_2)}F + 2\lambda D_{\mathrm{sym}}^{(c_2)}(F - (D_{\mathrm{sym}}^{(c_2)})^{-1}y)] + \ldots + \\
&[2L_{\mathrm{sym}}^{(c_m)}F + 2\lambda D_{\mathrm{sym}}^{(c_m)}(F - (D_{\mathrm{sym}}^{(c_m)})^{-1}y)] = \\
&2\Big[m(1+\lambda)F - \\
&\sum_{i=1}^{m}(D^{-1/2})^{(c_i)}W^{(c_i)}(D^{-1/2})^{(c_i)}F - m\lambda y\Big] = 0.
\end{aligned} \tag{20}$$

式(20)能够转换为如下形式:

$$\begin{aligned}
&2\Big[m(1+\lambda)F - \sum_{i=1}^{m}(D^{-1/2})^{(c_i)}W^{(c_i)}(D^{-1/2})^{(c_i)}F - \\
&m\lambda y\Big] = 0 \Rightarrow \\
&F - \frac{1}{m(1+\lambda)}\sum_{i=1}^{m}(D^{-1/2})^{(c_i)}W^{(c_i)}(D^{-1/2})^{(c_i)}F = \\
&\frac{\lambda}{1+\lambda}y \Rightarrow \\
&F = \frac{\lambda}{1+\lambda}\Big(I - \frac{1}{m(1+\lambda)}\times \\
&\quad\sum_{i=1}^{m}(D^{-1/2})^{(c_i)}W^{(c_i)}(D^{-1/2})^{(c_i)}\Big)^{-1}y.
\end{aligned} \tag{21}$$

从而,标准化的全相似多阶模型定义为

$$\begin{aligned}
&O_{\mathrm{sym}}^{\mathrm{multi}} = \\
&\frac{\lambda}{1+\lambda}\Big(I - \frac{1}{m(1+\lambda)}\times \\
&\sum_{i=1}^{m}(D^{-1/2})^{(c_i)}W^{(c_i)}(D^{-1/2})^{(c_i)}\Big)^{-1} = \\
&\alpha\Big(I - \frac{\beta}{m}\sum_{i=1}^{m}(D^{-1/2})^{(c_i)}W^{(c_i)}(D^{-1/2})^{(c_i)}\Big)^{-1} = \\
&\alpha(Z_{\mathrm{sym}}^{\mathrm{multi}})^{-1},
\end{aligned} \tag{22}$$

其中

$$Z_{\rm sym}^{\rm multi}=I-\frac{\beta}{m}\sum_{i=1}^{m}(D^{-1/2})^{(c_i)}W^{(c_i)}(D^{-1/2})^{(c_i)}.$$

综上所述,式(19)和(22)即为本文所提出的全相似高阶模型.

2.2 全相似多阶规范割算法

考虑式(12)中的广义特征分解问题和其变换形式(13),将其中矩阵W替换为式(19)中提出的全相似多阶相似矩阵$O^{\rm multi}$,本文提出全相似多阶规范割算法. 通过简单的变换,可得

$$D^{1/2}Z^{\rm multi}D^{1/2}\boldsymbol{l}=\frac{m\alpha}{(1-\varepsilon)}\boldsymbol{l}\Rightarrow$$

$$D^{1/2}\Big(\sum_{i=1}^{m}D_W^{(c_i)}-\beta\sum_{i=1}^{m}W^{(c_i)}\Big)D^{1/2}\boldsymbol{l}=\frac{m\alpha}{(1-\varepsilon)}\boldsymbol{l}. \tag{23}$$

其中:矩阵D为矩阵$O^{\rm multi}$的对角矩阵,矩阵$D_W^{(c_i)}$为矩阵$W^{(c_i)}$的对角矩阵,矩阵$W^{(c_i)}$是c_i阶下的权重矩阵,m表示阶数. 与前文类似,规范割中广义特征向量可以通过$\boldsymbol{u}=D^{-1/2}\boldsymbol{l}$计算得到. 对于标准化情况,仅需将式(23)中的矩阵$Z^{\rm multi}$替换为矩阵$Z_{\rm sym}^{\rm multi}$,即

$$D^{1/2}Z_{\rm sym}^{\rm multi}D^{1/2}\boldsymbol{l}=\frac{m\alpha}{(1-\varepsilon)}\boldsymbol{l}\Rightarrow$$

$$D^{1/2}\Big(I-\frac{\beta}{m}\sum_{i=1}^{m}(D_W^{(c_i)})^{-1/2}W^{(c_i)}(D_W^{(c_i)})^{-1/2}\Big)D^{1/2}\boldsymbol{l}=\frac{m\alpha}{(1-\varepsilon)}\boldsymbol{l}. \tag{24}$$

其中:矩阵D为矩阵$O_{\rm sym}^{\rm multi}$的对角矩阵,矩阵$D_W^{(c_i)}$为矩阵$W^{(c_i)}$的对角矩阵,矩阵$W^{(c_i)}$是c_i阶下的权重矩阵. 如式(23)和(24)所示,全相似多阶模型同样避免了计算稠密矩阵,即稀疏矩阵$Z^{\rm multi}$和$Z_{\rm sym}^{\rm multi}$的逆矩阵.

根据上述分析,本文提出的全相似多阶规范割算法(FMNcut)的流程可总结如下.

算法2 全相似多阶规范割算法(FMNcut).

输入:数据集X,阶数集$c=(c_1,c_2,\ldots,c_m)$,聚类类别数k,参数α和参数β;

输出:k类聚类结果.

step 1: 从数据集X中进行抽样,在$c=(c_1,c_2,\ldots,c_m)$阶下对抽样数据进行高阶相似度量;

step 2: 在$c=(c_1,c_2,\ldots,c_m)$阶下构建张量$\boldsymbol{P}$和其展开矩阵P,计算$W=PP^{\rm T}$, L或$L_{\rm sym}$;

step 3: 通过式(19)或(22)计算全相似高阶相似矩阵$O^{\rm multi}$或$O_{\rm sym}^{\rm multi}$;

step 4: 解决式(23)或(24)中的特征分解问题;

step 5: 通过$\boldsymbol{u}=D^{-1/2}\boldsymbol{l}$计算$k$个广义特征向量;

step 6: 使用k-means算法将$\boldsymbol{u}$聚类为k类.

3 实验和分析

为验证本文所提出算法的有效性,将算法应用于运动分割. 实验数据采用Hopkins 155运动分割数据库[24]. 其中: 2-运动数据选取1R2RCR_g13, 1R2RCR_g23, 2T3RCRT_g23, 2T3RCRT_g13 数据; 3-运动数据选取2R3RTC, 2RT3RC, 2RT3RCR, 2T3-RCRT数据. 对比算法选择高阶算法: k-means[7]、k-flats[25]、LRR[26]、SCC[2]、SGC[12]、SSC-OMP[27]、TSC[28]、NSN[29]、HOSVD[18]和Tetris[7]. 所有算法的定量评估准则使用聚类误差[2,7,30]. 在评估结果中同样对所有数据计算其平均误差和均值误差. 对于高阶边$e=\{i_1,i_2,\ldots,i_m\}$,使用极化曲率[21]计算其权重,即

$$w_e(\{i_1,i_2,\ldots,i_m\})=\exp\Big(-\frac{f_{pc}(i_1,i_2,\ldots,i_m)}{\sigma^2}\Big). \tag{25}$$

其中: $f_{pc}(\cdot)$表示极化曲率,相似张量w_e称为极化张量[18],调整参数σ使用估计算法[2]进行自动估计计算. 由于$\beta=1-\alpha$,只需要确定参数α的值. 在实验中,本文测试了如下取值$\alpha\in\{10^{-2},10^{-3},10^{-4}\}$. 如表1所示,参数$\sigma$越小,实验显示算法结果越好,因此,本文定值$\alpha=10^{-4}$. 在实验中,所有算法运行20遍,并取其平均结果作为最终结果,该操作是为了平衡算法随机初始化对实验结果的影响.

表1 在数据1R2RCR_g23上对参数α不同取值测试结果

α	10^{-2}	10^{-3}	10^{-4}
meanError	0.048 7	0.044 1	0.039 2

3.1 分析不同阶数据对算法的影响

为了分析不同阶数对算法的影响,本节设计了一系列实验.

对于使用非标准化O和标准化$O_{\rm sym}$的FHNcut算法,选取阶数$c=\{5,6,7,8\}$,对每一阶分别构建对应的高阶相似模型,使用两种情况的FHNcut对数据进行聚类,并对得到的结果进行定量评估. 表2和表3给出了定量评估结果. 其中: O和$O_{\rm sym}$分别代表算法FHNcut_O和算法FHNcut_$O_{\rm sym}$,最优评估结果用加粗数字表示. 如表2和表3所示,通常情况下,低阶数据可以得到更好的实验结果,但这并不是一定的. 例如: 对于数据1R2RCR_g23,在标准化和非标准化情况下,$c=5$都能得到比$c=8$更好的实验结果;但是对于数据2T3RCRT_g23, $c=6$时的结果略好于$c=5$时的结果. 这说明更高的阶数不一定能保证更好的结果,但是如果实验中不能确定最好的阶数,

表 2　2-运动数据下不同阶FHNcut算法结果

算法	2-motion	1R2RCR_g13		1R2RCR_g23		2T3RCRT_g23		2T3RCRT_g13	
		mean	median	mean	median	mean	median	mean	median
O_{sym}	5	0.0030	**0.0026**	**0.0051**	**0.0046**	0.1157	0.1126	0.1838	0.1820
	6	**0.0021**	0.0027	0.0068	0.0091	**0.0101**	**0.0109**	0.1736	0.1758
	7	0.0093	0.0106	0.0082	0.0091	0.1303	0.1423	**0.0525**	**0.0566**
	8	0.1114	0.1167	0.3986	0.4018	0.1780	0.1776	0.1090	0.1172
O	5	**0.0028**	**0.0026**	**0.0022**	**0.0023**	0.1110	0.1136	0.1388	0.1430
	6	0.0034	0.0027	0.0043	0.0046	**0.0157**	**0.0182**	0.0141	0.0117
	7	0.0118	0.0133	0.0071	0.0091	0.1185	0.1168	**0.0078**	**0.0078**
	8	0.1081	0.1088	0.3911	0.3973	0.1843	0.1934	0.0848	0.0957

表 3　3-运动数据下不同阶FHNcut算法结果

算法	3-motion	2R3RTC		2RT3RC		2RT3RCR		2T3RCRT	
		mean	median	mean	median	mean	median	mean	median
O_{sym}	5	0.1465	0.1723	0.0864	0.0907	0.1814	0.1830	0.1566	0.1565
	6	**0.0504**	**0.0491**	**0.0484**	**0.0508**	**0.1705**	**0.1800**	0.1493	0.1529
	7	0.1662	0.1723	0.2495	0.2541	0.2230	0.2319	**0.1307**	**0.1344**
	8	0.1715	0.1723	0.3077	0.3067	0.3700	0.3748	0.3459	0.3481
O	5	**0.0411**	0.0411	**0.0471**	**0.0463**	0.1764	0.1791	0.1434	0.1492
	6	0.0443	**0.0361**	0.0582	0.0572	**0.1348**	**0.1389**	0.1415	0.1409
	7	0.1659	0.1703	0.2549	0.2559	0.2174	0.2231	**0.1257**	**0.1262**
	8	0.1687	0.1743	0.3125	0.3149	0.3698	0.3806	0.3491	0.3527

表 4　2-运动数据下不同阶FMNcut算法结果

算法	2-motion	1R2RCR_g13		1R2RCR_g23		2T3RCRT_g23		2T3RCRT_g13	
		mean	median	mean	median	mean	median	mean	median
O_{sym}	(5,6)	0.0024	**0.0027**	0.0059	**0.0046**	0.0273	0.0268	0.3715	0.3789
	(5,10)	0.0060	0.0066	0.0048	0.0046	0.0320	0.0328	0.3625	0.3691
	(6,7)	**0.0021**	0.0027	0.0059	0.0046	0.0418	0.0487	0.3658	0.3750
	(6,10)	0.0054	0.0053	**0.0043**	0.0046	**0.0226**	**0.0243**	**0.3150**	**0.3320**
	(5,6,7)	**0.0020**	**0.0027**	0.0087	0.0091	**0.0298**	**0.0292**	0.3768	0.3770
	(6,7,8)	0.0164	0.0159	**0.0041**	**0.0046**	0.0496	0.0389	0.3385	0.3438
	(7,8,9)	0.0170	0.0172	0.0094	0.0091	0.0474	0.0487	0.2795	0.3027
	(8,9,10)	0.1325	0.1353	0.4103	0.4247	0.4213	0.4270	**0.2342**	**0.2617**
O	(5,6)	0.0024	**0.0027**	0.0059	**0.0046**	0.0273	0.0268	0.3715	0.3789
	(5,10)	0.0060	0.0066	0.0048	0.0046	0.0320	0.0328	0.3625	0.3691
	(6,7)	**0.0021**	0.0027	0.0059	0.0046	0.0418	0.0487	0.3658	0.3750
	(6,10)	0.0054	0.0053	**0.0043**	0.0046	**0.0226**	**0.0243**	**0.3150**	**0.3320**
	(5,6,7)	**0.0020**	**0.0027**	0.0087	0.0091	**0.0298**	**0.0292**	0.3768	0.3770
	(6,7,8)	0.0167	0.0186	**0.0064**	**0.0046**	0.0655	0.0608	0.3307	0.3301
	(7,8,9)	0.0171	0.0186	0.0082	0.0091	0.0519	0.0535	0.2898	0.3027
	(8,9,10)	0.1286	0.1300	0.4059	0.4155	0.3758	0.3856	**0.2387**	**0.2441**

表 5 3-运动数据下不同阶FMNcut算法结果

算法	3-motion	2R3RTC		2RT3RC		2RT3RCR		2T3RCRT	
		mean	median	mean	median	mean	median	mean	median
O_{sym}	(5,6)	**0.1135**	**0.1293**	**0.0680**	**0.0708**	**0.2007**	**0.2055**	0.1591	0.1565
	(5,10)	0.1747	0.1743	0.2159	0.2178	0.2582	0.2661	0.1727	0.1842
	(6,7)	0.1736	0.1743	0.2449	0.2559	0.2661	0.2701	**0.1475**	**0.1473**
	(6,10)	0.1736	0.1743	0.2387	0.2414	0.2659	0.2681	0.1787	0.1897
	(5,6,7)	0.1739	0.1743	**0.2334**	**0.2377**	**0.2460**	**0.2701**	**0.1441**	**0.1455**
	(6,7,8)	0.1686	0.1743	0.2530	0.2559	0.2655	0.2701	0.1549	0.1584
	(7,8,9)	**0.1547**	**0.1703**	0.2576	0.2577	0.2691	0.2701	0.1757	0.1851
	(8,9,10)	0.1755	0.1764	0.3887	0.3929	0.4621	0.4638	0.4084	0.4098
O	(5,6)	**0.1135**	**0.1293**	**0.0680**	**0.0708**	**0.2007**	**0.2055**	0.1591	0.1565
	(5,10)	0.1747	0.1743	0.2159	0.2178	0.2582	0.2661	0.1727	0.1842
	(6,7)	0.1736	0.1743	0.2449	0.2559	0.2661	0.2701	**0.1475**	**0.1473**
	(6,10)	0.1736	0.1743	0.2387	0.2414	0.2659	0.2681	0.1787	0.1897
	(5,6,7)	**0.1722**	**0.1723**	**0.2022**	**0.2015**	0.2575	0.2720	**0.1261**	**0.1271**
	(6,7,8)	0.1736	0.1743	0.2463	0.2477	**0.2550**	**0.2701**	0.1745	0.1713
	(7,8,9)	0.1723	0.1723	0.2573	0.2577	0.2634	0.2701	0.1863	0.1888
	(8,9,10)	0.1741	0.1743	0.3783	0.3793	0.4437	0.4452	0.3834	0.3849

则应优先选取低阶数据. 在算法FHNcut_O和算法FHNcut_O_{sym}的对比中,虽然直观的感觉下标准化的算法应该更优,但在大多数数据下,算法FHNcut_O显然要优于算法FHNcut_O_{sym}.

对于使用非标准化O和标准化O_{sym}的FMNcut算法,分别选取4组二阶组合$c = \{(5,6), (5,10), (6,7), (6,10)\}$和4组三阶组合$c = \{(5,6,7), (6,7,8), (7,8,9), (8,9,10)\}$进行实验验证. 表4和表5为定量评估结果. 其中:O和O_{sym}分别代表算法FMNcut_O和算法FMNcut_O_{sym},最优定量评估结果用加粗数字表示. 对于三阶情况,实验结果显示:融合更小阶的数据往往能够得到更优的结果;但是对于二阶的情况,这种现象并不明显. 阶数的选择对于算法的结果有很大的影响. 例如:$c = \{(5,6,7)\}$时FMNcut的结果优于$c = \{5,6,7\}$时FHNcut的结果;但是$c = \{(5,6)\}$时FMNcut 的结果并不比$c = \{5\}$时FHNcut的结果好. 另一个可以从表2～表5得到的结论是:多阶情况下算法对不同阶数组合实验结果的波动程度要低于一阶情况下算法不同阶数实验结果的波动程度. 这是因为多阶算法能够平衡不同阶的特征信息,但也导致部分特征被平衡后消失,进而导致部分多阶算法实验结果差于单阶实验结果.

3.2 与其他高阶算法对比

为了验证本文所提出算法的有效性,将本文算法与现有的高阶算法进行对比. 所有算法中,阶数设置为$c = 5$. 表6和表7为定量评估结果. 其中:O和

表 6 2-运动数据下各算法结果

算法	1R2RCR_g13		1R2RCR_g23		2T3RCRT_g23		2T3RCRT_g13	
	mean	median	mean	median	mean	median	mean	median
k-means	0.2375	0.2375	0.2060	0.2060	0.3766	0.3766	0.4681	0.4681
k-flats	0.0613	0.0211	0.1939	0.2315	0.2338	0.2251	0.3586	0.3582
LRR	0.0237	0.0237	0.0116	0.0116	**0.0022**	**0.0022**	**0**	**0**
SCC	0.0140	0.0053	0.0074	**0.0023**	0.3611	0.3680	0.1063	0.0697
SSC-OMP	0.3984	0.3984	0.2431	0.2431	0.3377	0.3377	0.1749	0.1749
TSC	0.3931	0.3931	0.2708	0.2708	0.2554	0.2554	0.4374	0.4374
NSN	0.0053	0.0053	0.2153	0.2153	0.3788	0.3788	0.0071	0.0071
SGC	0.0090	**0**	0.0397	0.0394	0.1287	0.0509	**0**	**0**
HOSVD	**0.0017**	**0**	0.0170	0.0174	0.2195	0.2305	**0**	**0**
Tetris	0.0053	0.0013	0.0347	0.0289	0.2279	0.2630	0.2436	0.2482
O_{sym}	0.0030	0.0026	**0.0051**	0.0046	0.1157	**0.1126**	0.1838	0.1820
O	**0.0028**	0.0026	**0.0022**	**0.0023**	**0.1110**	0.1136	0.1388	0.1430

表 7 3-运动数据下各算法结果

算法	2R3RTC		2RT3RC		2RT3RCR		2T3RCRT	
	mean	median	mean	median	mean	median	mean	median
k-means	0.4515	0.4509	0.4882	0.4882	0.3796	0.3796	0.4475	0.4475
k-flats	**0.0315**	0.0190	0.0585	0.0481	0.2486	0.2368	0.3855	0.3849
LRR	**0.0220**	0.0220	0.1198	0.1198	0.2681	0.2681	**0.0166**	**0.0166**
SCC	0.0681	**0.0010**	**0.0469**	0.0499	0.3378	0.3542	0.3122	0.3039
SSC-OMP	0.0782	0.0782	0.4065	0.4065	0.5675	0.5675	0.2063	0.2063
TSC	0.4469	0.4469	0.5753	0.5753	0.3366	0.3366	0.2284	0.2284
NSN	0.3347	0.3347	0.0617	0.0617	**0.1057**	**0.1057**	0.3260	0.3260
SGC	0.0904	0.0170	0.0719	0.0472	0.3022	0.2877	0.3633	0.3646
HOSVD	0.0811	**0.0120**	0.0593	**0.0445**	0.3002	0.2994	0.3397	0.3582
Tetris	0.1727	0.0802	0.0897	0.0935	0.2704	0.3532	0.2742	0.2726
$O_{\rm sym}$	0.1465	0.1723	0.0864	0.0907	0.1814	0.1830	0.1566	0.1565
O	0.0411	0.0411	**0.0471**	**0.0463**	**0.1764**	**0.1791**	**0.1434**	**0.1492**

$O_{\rm sym}$ 的意义同前所述, 最优的两个定量评估结果使用加粗数字表示. 如表 6 和表 7 所示, 本文所提出的 FHNcut 算法给出了具有竞争力的结果. 在与谱聚类相关的高阶算法 Tetris 进行对比时, 本文算法几乎在所有数据下都占据优势, 仅个别情况略微低于 Tetris 算法. 而在与其他高阶算法对比时, 在大多数情况下, FHNcut 算法的结果比其他算法更好. 其中, 数据 1R2RCR_ g23 中最好的两个平均误差结果都是本文算法得到的. 特别地, FHNcut_O 比 FHNcut_$O_{\rm sym}$ 结果更好. 这说明本文所提出的全相似高阶模型对于谱聚类算法具有明显的提升效果, 尤其是使用非标准化拉氏矩阵时. 同时, 也验证了在谱聚类中全相似高阶相似关系比稀疏的高阶相似关系能够提供更全面的数据特征信息.

4 结 论

本文提出了两种高阶相似模型: 全相似高阶相似模型 (FHAM) 和全相似多阶相似模型 (FMAM). 这两种模型通过在高阶数据上使用直推式学习技术学习全相似信息得到, 从而保证了模型可以使用稀疏相似矩阵的计算优势, 其中 FMAM 融合了不同阶的高阶相似信息. 以这两种模型为基础, 本文提出了两种聚类算法: 一种是全相似高阶规范割算法 (FHNcut); 另一种是全相似多阶规范割算法 (FMNcut). 将本文提出的聚类算法应用于运动分割, 并与现有的高阶算法进行对比. 通过对实验结果的定量评估显示, 所提出的算法在大多数情况下都能给出具有竞争性的结果. 通过不同阶对算法影响的分析得到了选择阶数的模糊准则, 但不是精确的准则. 因此, 在后续工作中, 作者将研究如何在所提出的高阶算法中选择合适的阶数.

参考文献 (References)

[1] Shi J, Malik J. Normalized cuts and image segmentation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2000, 22(8): 888-905.

[2] Chen G, Lerman G. Spectral curvature clustering (SCC)[J]. International Journal of Computer Vision, 2009, 81(3): 317-330.

[3] Chertok M, Keller Y. Efficient high order matching[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2010, 32(12): 2205-2215.

[4] Duchenne O, Bach F, Kweon I S, et al. A tensor-based algorithm for high-order graph matching[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(12): 2383-2395.

[5] Agarwal S, Lim J, Zelnik-Manor L, et al. Beyond pairwise clustering[C]. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Diego: IEEE, 2005, 2: 838-845.

[6] Elhamifar E, Vidal R. Sparse subspace clustering: Algorithm, theory, and applications[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(11): 2765-2781.

[7] Ghoshdastidar D, Dukkipati A. Uniform hypergraph partitioning: Provable tensor methods and sampling techniques[J]. Journal of Machine Learning Research, 2017, 18(50): 1-41.

[8] Ochs P, Brox T. Higher order motion models and spectral clustering[C]. IEEE Conference on Computer Vision and Pattern Recognition. Providence: IEEE, 2012: 614-621.

[9] Ghoshdastidar D, Dukkipati A. Spectral clustering using multilinear SVD: Analysis, approximations and applications[C]. The 21th AAAI Conference on Artificial Intelligence. Austin, 2015: 2610-2616.

[10] Liu H, Latecki L J, Yan S. Robust clustering as ensembles of affinity relations[C]. Advances in Neural Information Processing Systems. Vancouver, 2010: 1414-1422.

[11] Ren P, Aleksi T, Wilson R C, et al. A polynomial characterization of hypergraphs using the ihara zeta function[J]. Pattern Recognition, 2011, 44(9): 1941-1957.

[12] Jain S, Madhav Govindu V. Efficient higher-order clustering on the grassmann manifold[C]. Proceedings of the IEEE International Conference on Computer Vision. Sydney, 2013: 3511-3518.

[13] Hein M, Setzer S, Jost L, et al. The total variation on hypergraphs-learning on hypergraphs revisited[C]. Advances in Neural Information Processing Systems. Lake Tahoe, 2013: 2427-2435.

[14] Zhou D, Huang J, Schölkopf B. Learning with hypergraphs: Clustering, classification, and embedding[C]. Advances in Neural Information Processing Systems. Vancouver, 2007: 1601-1608.

[15] Rota B S, Pelillo M. A game-theoretic approach to hypergraph clustering[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, 35(6): 1312-1327.

[16] Ducournau A, Bretto A, Rital S, et al. A reductive approach to hypergraph clustering: An application to image segmentation[J]. Pattern Recognition, 2012, 45(7): 2788-2803.

[17] Kim S, Yoo C D, Nowozin S, et al. Image segmentation using higher-order correlation clustering[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(9): 1761-1774.

[18] Ghoshdastidar D, Dukkipati A. Consistency of spectral partitioning of uniform hypergraphs under planted partition model[C]. Advances in Neural Information Processing Systems. Montréal, 2014: 397-405.

[19] Ghoshdastidar D, Dukkipati A. Consistency of spectral hypergraph partitioning under planted partition model[J]. The Annals of Statistics, 2017, 45(1): 289-315.

[20] Govindu V M. A tensor decomposition for geometric grouping and segmentation[C]. IEEE Computer Society Conference Computer Vision and Pattern Recognition. San Diego: IEEE, 2005, 1: 1150-1157.

[21] Chen G, Lerman G. Foundations of a multi-way spectral clustering framework for hybrid linear modeling[J]. Foundations of Computational Mathematics, 2009, 9(5): 517-558.

[22] Kofidis E, Regalia P A. On the best rank-1 approximation of higher-order supersymmetric tensors[J]. SIAM Journal on Matrix Analysis and Applications, 2002, 23(3): 863-884.

[23] De Lathauwer L, De Moor B, Vandewalle J. On the best rank-1 and rank-$(r_1, r_2, \ldots, r_n)$ approximation of higher-order tensors[J]. SIAM Journal on Matrix Analysis and Applications, 2000, 21(4): 1324-1342.

[24] Tron R, Vidal R. A benchmark for the comparison of 3-d motion segmentation algorithms[C]. IEEE Conference on Computer Vision and Pattern Recognition. Minneapolis: IEEE, 2007: 1-8.

[25] Bradley P S, Mangasarian O L. K-plane clustering[J]. Journal of Global Optimization, 2000, 16(1): 23-32.

[26] Liu G, Lin Z, Yu Y. Robust subspace segmentation by low-rank representation[C]. Proceedings of the 27th International Conference on Machine Learning. Haifa: IMLS, 2010: 663-670.

[27] Dyer E L, Sankaranarayanan A C, Baraniuk R G. Greedy feature selection for subspace clustering[J]. Journal of Machine Learning Research, 2013, 14(1): 2487-2517.

[28] Heckel R, Bölcskei H. Subspace clustering via thresholding and spectral clustering[C]. IEEE International Conference on Acoustics, Speech and Signal Processing. Vancouver: IEEE, 2013: 3263-3267.

[29] Park D, Caramanis C, Sanghavi S. Greedy subspace clustering[C]. Advances in Neural Information Processing Systems. Montréal, 2014: 2753-2761.

[30] Ghoshdastidar D, Dukkipati A. A provable generalized tensor spectral method for uniform hypergraph partitioning[C]. International Conference on Machine Learning. Lille, 2015: 400-409.

作者简介

张敬茂 (1990−), 男, 博士生, 从事聚类算法及其应用的研究, E-mail: jmzhang19@sina.cn;

沈艳霞 (1973−), 女, 教授, 博士生导师, 从事新能源技术、非线性控制、模式识别等研究, E-mail: shenyx@jiangnan.edu.cn.

(责任编辑：李君玲)