TLWCC：一种双层子空间加权协同聚类算法

肖龙飞[1,2] 陈小军[1,2]

[1]（ 深圳市高性能数据挖掘重点实验室 深圳 518055 ）
[2]（ 中国科学院深圳先进技术研究院 深圳 518055 ）

摘 要 协同聚类是对数据矩阵的行和列两个方向同时进行聚类的一类算法。本文将双层加权的思想引入协同聚类，提出了一种双层子空间加权协同聚类算法（TLWCC）。TLWCC对聚类块（co-cluster）加一层权重，对行和列再加一层权重，并且算法在迭代过程中自动计算块、行和列这三组权重。TLWCC考虑不同的块、行和列与相应块、行和列中心的距离，距离越大，认为其噪声越强，就给予小权重；反之噪声越弱，给予大权重。通过给噪声信息小权重，TLWCC能有效地降低噪声信息带来的干扰，提高聚类效果。本文通过四组实验展示TLWCC算法识别噪声信息的能力、参数选取对算法聚类结果的影响程度，算法的聚类性能和时间性能。

关键词 协同聚类；加权；聚类；数据挖掘

TLWCC: A Two-Level Subspace Weighting Co-clustering Algorithm

XIAO Long-fei[1,2] CHEN Xiao-jun[1,2]

[1](*Shenzhen Key Laboratory of High Performance Data Mining, Shenzhen* 518055, *China*)
[2](*Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen* 518055, *China*)

Abstract Co-clustering algorithms cluster a data matrix into row clusters and column clusters simultaneously. In this paper, we propose TLWCC, a two-level subspace weighting co-clustering algorithm, and introduces the idea of a two-level subspace weighting method into the co-clustering process. TLWCC adds the first level of weights on co-clusters, and then adds the second level of weights on rows and columns. The three types of weights (co-cluster, row and column weights) are computed in the clustering progress, according to the distances between co-clusters (or rows, columns) and their centers. The larger the distance is, the stronger noise it implies, so a smaller weight is given and vice verse. Thus, by giving small weights to noise, TLWCC filters out the noise and improves the co-clustering result. We propose an iterative algorithm to optimize the model. We carried out four experiments to learn more about TLWCC. The first experiment investigated the properties of three types of weights. The second experiment studied how the clustering result was influenced by the parameters. The third experiment compared the clustering performance of TLWCC with other three algorithms. The fourth experiment examined the computational efficiency of our proposed algorithm.

Keywords co-clustering; weighting; clustering; data mining

1 引 言

协同聚类（Co-clustering，或biclustering）是对数据矩阵的行和列两个方向同时进行聚类的一类算法。由于能够同时提供两个方向的聚类结果和处理高维、稀疏数据时具有很好的性能等优点，最近十年来，协同聚类被广泛应用于基因表达数据分析[1]，文本挖掘[2]，协同过滤[3]，多媒体信息挖掘[4]和计算神经科学[5]等诸多领域。

真实数据含有大量噪声信息，传统的单方向聚类算法在处理含有噪声信息的数据时，普遍采用子空间加权方法（Subspace Weighting）[6-9]。子空间加权方法通过给予相关信息大权重，给予噪声信息小权重

作者简介：肖龙飞，硕士研究生，主要研究方向为数据挖掘、子空间聚类和协同聚类。E-mail：lf.xiao@siat.ac.cn；陈小军，助理研究员，主要研究方向为机器学习、聚类与分类、分布式数据挖掘算法。

的方法，降低噪声信息的干扰。EWKM[9]算法是一种基于熵的子空间加权聚类算法，该算法在聚类过程中自动更新这些权重。FG-k-means[7]算法第一次提出将双层子空间加权的思想应用于聚类算法，该算法先对含有相似特征的变量分组加一层权重，然后对每个分组内的变量再加一层权重，这样就可以在粗和细两个粒度对数据加权重。

1976年，Hartigan首次提出一种基于划分的协同聚类算法[10]。2001年，Dillon等提出了信息论协同聚类算法，它的聚类目标是使得聚类后的互信息损失最少[11]。2004年，Cho等提出了一种最小化平方残差和的协同聚类算法[12]。2007年，Banerjee等基于Bregman距离，提出了一种一般化的最大化熵的协同聚类框架，通过这个框架可以得到很多协同聚类算法的特例。然而，现有的协同聚类算法都没有考虑在聚类过程中计算权重，也就不能对数据信息的重要性进行区分，在处理有噪声的数据时效果往往不理想。

本文将FG-k-means的双层子空间加权的思想引入协同聚类，提出了一种双层子空间加权协同聚类算法（TLWCC）。TLWCC对聚类块加一层权重，对行和列再加一层权重，并且算法在迭代过程中自动计算块、行和列这三组权重。我们提出了一个新的优化模型，并通过一个迭代算法来求解这个模型。我们通过四组实验，分别展示TLWCC权重的特性、参数对聚类精度影响、算法的聚类性能和算法的时间性能。

本文第二部分对聚类问题进行描述，第三部分提出双层加权协同聚类算法，第四部分通过实验分析算法的特性和性能，第五部分做出总结与并对未来的工作进行了展望。

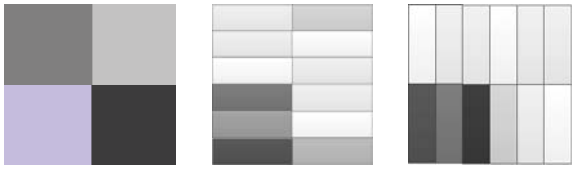
2　问题描述

设$\boldsymbol{X}=[x_{i,j}]_{N\times M}$是一个$N$行$M$列的数据矩阵。协同聚类的目标就是将该矩阵在行方向上聚成K个类簇，同时在列方向上聚成L个类簇。协同聚类的类簇信息用两个划分矩阵 $U=[u_{i,g}]_{N\times K}$ 和 $V=[v_{j,h}]_{M\times L}$ 表示。对于行划分矩阵U，如果第i行属于第g行类簇，那么$u_{i,g}$就置为1，否则置为0。对于列划分矩阵V，如果第j列属于第h列类簇，那么$v_{j,h}$就置为1，否则置为0。另外，$Z=[z_{g,h}]_{K\times L}$表示 $K\times L$个聚类块的中心的值，一般通过求聚类块的均值来计算。当数据的N和M变得很大是，不可避免地会含有大量噪声，而噪声过多会淹没很多重要的信息，导致聚类结果不理想。对于传统单方向聚类算法而言，加权是一种普遍采用的降低噪声干扰的方法。本文首次提出了一种基于双层加权思想的协同聚类算法，来解决协同聚类算法对噪声信息无法进行区分的问题。

3　双层加权的协同聚类算法

3.1　算法模型

TLWCC算法在聚类的过程中，先对聚类块加一层权重，然后对行和列加另一层权重，三组的权重效果如图1：



(a) 块权重　　(b) 行权重　　(c) 列权重

图1　TLWCC权重示意图

为了将数据矩阵X划分成K个行类簇和L个列类簇，本文考虑同时加块、行和列三组权重，提出了以下的优化问题，并将协同聚类问题转化为求解优化问题。优化目标函数如下：

$$\begin{aligned}&J(U,V,Z,R,C,W)\\&=\frac{1}{MN}\sum_{g=1}^{K}\sum_{h=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{M}u_{i,g}v_{j,h}r_{h,i}c_{g,j}w_{g,h}d(x_{i,j},z_{g,h})\\&+\frac{\lambda}{N}\sum_{h=1}^{L}\sum_{i=1}^{N}r_{h,i}\log r_{h,i}\\&+\frac{\eta}{M}\sum_{g=1}^{K}\sum_{j=1}^{M}c_{g,j}\log c_{g,j}+\varphi\sum_{g=1}^{K}\sum_{h=1}^{L}w_{g,h}\log w_{g,h}\end{aligned}\tag{1}$$

subject to

$$\begin{cases}\sum_{g=1}^{K}u_{i,g}=1,\quad u_{i,g}\in\{0,1\},1\le i\le N\\\sum_{h=1}^{L}v_{j,h}=1,\quad v_{j,h}\in\{0,1\},1\le j\le M\\\sum_{i=1}^{N}r_{h,i}=1,\quad 0<r_{h,i}<1,1\le h\le L\\\sum_{j=1}^{M}c_{g,j}=1,\quad 0<c_{g,j}<1,1\le g\le K\\\sum_{g=1}^{K}\sum_{h=1}^{L}w_{g,h}=1,\quad 0<w_{g,h}<1\end{cases}\tag{2}$$

其中

- $U=[u_{i,g}]_{N\times K}$ 表示行划分矩阵
- $V=[v_{j,h}]_{M\times L}$ 表示列划分矩阵
- $Z=[z_{g,h}]_{K\times L}$ 表示聚类块中心的值
- $R=[r_{h,i}]_{L\times N}$ 表示行权重

- $C=[c_{g,j}]_{K\times M}$ 表示列权重
- $W=[w_{g,h}]_{K\times L}$表示块权重
- $\lambda>0$，$\eta>0$和$\varphi>0$是三个参数，分别是用来调整三组权重R，C和W的分布。
- $d(x_{i,j},z_{g,h})$表示矩阵中的元素$x_{i,j}$与它所对应的聚类块$z_{g,h}$之间的距离。在本论文中，我们使用平方欧氏距离：

$$d(x_{i,j},z_{g,h})=(x_{i,j}-z_{g,h})^2 \tag{3}$$

3.2 优化算法

我们通过迭代地解决以下6个优化子问题来优化目标函数（1）：

（1）P_1：固定$\hat{V}$，$\hat{Z}$，$\hat{R}$，$\hat{C}$和$\hat{W}$，求子问题$P(U,\hat{V},\hat{Z},\hat{R},\hat{C},\hat{W})$;

（2）P_2：固定$\hat{U}$，$\hat{Z}$，$\hat{R}$，$\hat{C}$和$\hat{W}$，求子问题$P(\hat{U},V,\hat{Z},\hat{R},\hat{C},\hat{W})$;

（3）P_3：固定$\hat{U}$，$\hat{V}$，$\hat{R}$，$\hat{C}$和$\hat{W}$，求子问题$P(\hat{U},\hat{V},Z,\hat{R},\hat{C},\hat{W})$;

（4）P_4：固定$\hat{U}$，$\hat{V}$，$\hat{Z}$，$\hat{C}$和$\hat{W}$，求子问题$P(\hat{U},\hat{V},\hat{Z},R,\hat{C},\hat{W})$;

（5）P_5：固定$\hat{U}$，$\hat{V}$，$\hat{Z}$，$\hat{R}$和$\hat{W}$，求子问题$P(\hat{U},\hat{V},\hat{Z},\hat{R},C,\hat{W})$;

（6）P_6：固定$\hat{U}$，$\hat{V}$，$\hat{Z}$，$\hat{R}$和$\hat{C}$，求子问题$P(\hat{U},\hat{V},\hat{Z},\hat{R},\hat{C},W)$;

这六个优化子问题通过以下方法进行求解：

解决子问题P_1，使用公式(4)；

$$\begin{cases} u_{i,g}=1 & \text{if } J_g\le J_s \text{ for } 1\le s<K \text{ where} \\ & J_s=\sum_{h=1}^{L}\sum_{j=1}^{M}\hat{v}_{j,h}\hat{c}_{s,j}\hat{r}_{h,i}\hat{w}_{g,h}d(x_{i,j},\hat{z}_{s,h}) \\ u_{i,s}=0 & \text{for } s\ne g \end{cases} \tag{4}$$

解决子问题P_2，使用公式(5)；

$$\begin{cases} v_{j,h}=1 & \text{if } J'_h\le J'_t \text{ for } 1\le t<L \text{ where} \\ & J'_t=\sum_{g=1}^{K}\sum_{i=1}^{N}\hat{u}_{i,g}\hat{c}_{g,j}\hat{r}_{t,i}\hat{w}_{g,h}d(x_{i,j},\hat{z}_{g,t}) \\ v_{j,t}=0 & \text{for } t\ne h \end{cases} \tag{5}$$

解决子问题P_3，使用公式(6)；

$$z_{g,h}=\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\hat{u}_{i,g}\hat{v}_{j,h}\hat{r}_{h,i}\hat{c}_{g,j}x_{i,j}}{\sum_{i=1}^{N}\sum_{j=1}^{M}\hat{u}_{i,g}\hat{v}_{j,h}\hat{r}_{h,i}\hat{c}_{g,j}} \tag{6}$$

解决子问题P_4，使用公式(7)和(8)；

$$r_{h,i}=\frac{\exp\{\frac{-F_{h,i}}{\lambda}\}}{\sum_{i'=1}^{M}\exp\{\frac{-F_{h,i'}}{\lambda}\}} \tag{7}$$

$$F_{h,i}=\frac{1}{N}\sum_{g=1}^{K}\sum_{j=1}^{M}\hat{u}_{i,g}\hat{v}_{j,h}\hat{c}_{g,j}\hat{w}_{g,h}d(x_{i,j},\hat{z}_{g,h}) \tag{8}$$

解决子问题P_5，使用公式(9)和(10)；

$$c_{g,j}=\frac{\exp\{\frac{-E_{g,j}}{\eta}\}}{\sum_{j'=1}^{N}\exp\{\frac{-E_{g,j'}}{\eta}\}} \tag{9}$$

$$E_{g,j}=\frac{1}{M}\sum_{h=1}^{L}\sum_{i=1}^{N}\hat{u}_{i,g}\hat{v}_{j,h}\hat{r}_{h,i}\hat{w}_{g,h}d(x_{i,j},\hat{z}_{g,h}) \tag{10}$$

解决子问题P_6，使用公式(11)和(12)；

$$w_{g,h}=\frac{\exp\{\frac{-D_{g,h}}{\varphi}\}}{\sum_{g'=1}^{K}\sum_{h'=1}^{L}\exp\{\frac{-D_{g',h'}}{\varphi}\}} \tag{11}$$

$$D_{g,h}=\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}\hat{u}_{i,g}\hat{v}_{j,h}\hat{r}_{h,i}\hat{c}_{g,j}d(x_{i,j},\hat{z}_{g,h}) \tag{12}$$

因此，通过(4)~(12)和算法1，我们可以求解带约束(2)的最优化问题(1)：

算法 1 双层加权协同聚类算法（TLWCC）

Input: $\boldsymbol{X}$, K, L, λ, η, φ
Output: U, V, Z, R, C, W
Init: Start with an arbitrary co-clustering, assign every element of R, C and W with equal values, respectively.
t := 0
repeat:
  Update U by (4)
  Update V by (5)
  Update Z by (6)
  Update R by (7) and (8)
  Update C by (9) and (10)
  Update W by (11) and (12)
  t := t+1
Until: the objective function (1) obtains its local minimum value.

在本算法中，$\varphi>0$，$\lambda>0$和$\eta>=0$三个参数用来调整三组权重W、R和C的分布。在一定区间内：

φ越大，块权重分布越平均，否则块权重W分布越集中。

λ越大，行权重分布越平均，否则行权重R分布越集中；

η越大，列权重分布越平均，否则列权重C分布越集中；

参数取值超过一定区间，即小于区间下限或大于区间上限时，权重的分布几乎不再发生变化。

因为TLWCC产生的求解子问题序列(P_1, P_2, …)保证目标函数值是严格递减的，所以算法会在有限次迭代后收敛到一个局部最优解。假设算法需要经过r次迭代达到收敛，TLWCC算法的时间复杂度为$O(rNM(K+L))$，即保证了TLWCC算法具有良好的可扩展性。

4　实验分析

4.1　实验数据

为了研究TLWCC算法的权重特性和参数特点，我们生成了一个30×30的模拟数据D_1。该数据被放入3个显著的聚类块结构，聚类块中的元素在(0.45, 0.55)之间随机取值，其余元素在(0,1)之间随机取值。数据D_1如图2所示，其中灰度越深表示元素值越大，灰度越浅表示元素值越小。

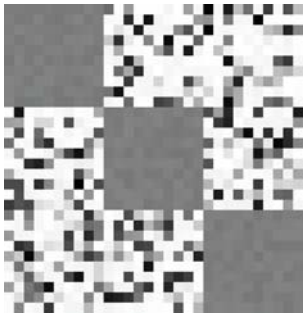


图2　数据D_1

我们使用Lomet数据集[13]（http://www.hds.utc.fr/coclustering/）来测试TLWCC算法的聚类性能。Lomet等依据隐藏块模型（Latent block model）设计了一种用来评价协同聚类算法性能的模拟数据生成方法，该方法是使用错分率（Error rate）来衡量数据被错分的可能性，错分率越大，越难通过聚类找出原始的类簇结构；反之，聚类越容易。

为了测试TLWCC算法的时间性能，我们设计了两组模拟数据S_1和S_2：第一组S_1，固定列数为200，行数在{200,800,1600,2400,3200,6400,9600,12800}中取值，共8个数据；第二组S_2，固定行数为200，列数在{200,800,1600,2400,3200,6400,9600,12800}中取值，共8个数据；其中每一个数据都有3个行类簇和3个列类簇的结构。

4.2　权重特性实验

本实验我们分析作用在数据上的三组权重的特点。实验使用模拟数据D_1，设置参数φ为1.0E-5，η为3.0E-5，λ为3.0E-5。图3为TLWCC算法结束时，得到的块、行和列三组权重。

图中颜色越深表示权重值越大，颜色越浅表示权重值越小。可以看出，块、行和列三组权重都分别在块、行和列三个角度、两个层次上反应了数据D_1的内部结构，有效地区分了相关信息和噪声信息：图3.a中块权重反应了在粗粒度上每个聚类块的权重；图3.b和图3.c分别在细的粒度上反应了每一行和每一列的权重。三组权重联合作用，有效提高了相关信息的权重，降低了噪声的权重，很好地过滤了噪声信息，提高算法抵抗噪声干扰的能力。

(a)块权重　　(b)行权重　　(c)列权重

图3　TLWCC得到的块、行和列三组权重

4.3　参数选择对聚类性能的影响

我们使用正规化互信息[14]（Normalized Mutual Information，NMI）作为衡量聚类精度的指标，定义如下：$NMI(\Omega,\mathbb{C})=\frac{I(\Omega,\mathbb{C})}{(H(\Omega)+H(\mathbb{C}))/2}$，$\Omega$和$\mathbb{C}$分别是真实的类簇信息和聚类得到的类簇信息，$I(\Omega,\mathbb{C})=\sum_k\sum_j P(\omega_k\cap c_j)\log\frac{P(\omega_k\cap c_j)}{P(\omega_k)P(c_j)}$表示Ω和C的互信息，$H(\Omega)=-\sum_k P(\omega_k)\log P(\omega_k)$表示Ω的熵。R-NMI和C-NMI分别表示行和列两个方向聚类结果正规化互信息。

本实验分析影响权重分布的三个参数φ、λ和η的取值对算法聚类结果的影响。实验设置如下：使用模拟数据D_1；参数η和λ的取值范围是(1.0E-5,100)，参数在该范围内按步长为2的等比数列取值；参数φ在{1.0E-6,1.0E-5,1.0E-4,1.0E-3}中取值；给定理想的初始划分。实验结果如图4所示：

从图4中我们可以看出：（1）当参数φ和λ固定不变时，随着η增大，行方向聚类结果R-NMI随之降低；当φ和η固定不变时，随着λ的增大，列方向的聚类结果C-NMI随之降低；（2）随着参数φ从1OE-6到1.0E-3取值不断增大，行和列方向上的聚类精度都随之降低。当参数Φ小于1.0E-6，λ小于1.0E-5或η小于1.0E-5时，权重会集中在很少的块、行或块上，这时算法会变得不稳定，也不能得到好的聚类结果。

实验结果出现上述情况的原因是：当参数过小时，权重过于集中，丢失了很多重要的信息；当参数过大时，权重分布又过于平均，也起不到使噪声信息的权重降低的作用。

综上所述，TLWCC的聚类结果随三个参数的变化而变化，当三组参数都在合适的范围内取值的时候，算法能得到行和列方向上都很好的聚类结果，但是如果参数取值过小或者过大时，那么聚类结果也会变差。

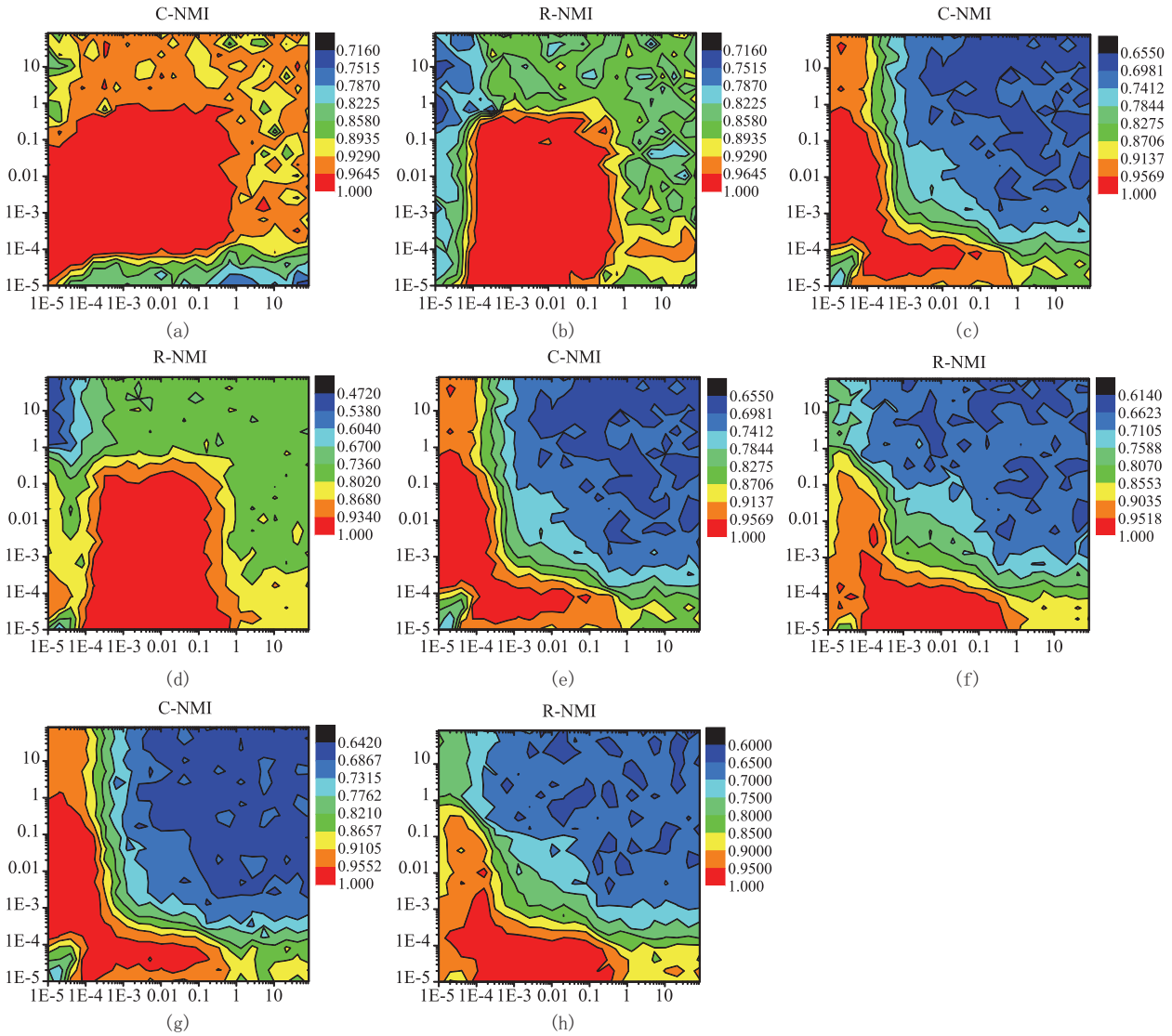


图4 参数选取对聚类结果的影响。各子图中，横轴为λ，纵轴为η。(a)和(b)分别为φ=1.0E-6时，列和行方向的聚类精度；(c)和(d)分别为φ=1.0E-5时，列和行方向的聚类精度；(e)和(f) 分别为φ=1.0E-4时，列和行方向的聚类精度；(g)和(h) 分别为φ=1.0E-3时，列和行方向的聚类精度。

4.4 聚类精度实验

本实验使用Lomet数据集，并按照数据大小为200×200和500×500，错分率为5%，12%和20%共选取其中6个数据。在这6个数据上，本文比较TLWCC与

表1 TLWCC算法聚类精度实验结果

数据		k-means	WKM	BBAC	TLWCC
数据大小	错分率				
200×200	5%	-0.43(0.00)*	-0.43(0.00)*	-0.32(0.02)*	0.50(0.05)
	12%	-0.19(0.00)*	-0.19(0.00)*	-0.12(0.00)*	0.24(0.03)
	20%	-0.53(0.00)*	-0.53(0.00)*	-0.52(0.00)*	0.57(0.05)
500×500	5%	-0.38(0.00)*	-0.38(0.00)*	-0.32(0.00)*	0.40(0.07)
	12%	-0.51(0.00)*	-0.51(0.00)*	-0.49(0.00)*	0.52(0.05)
	20%	-0.52(0.00)*	-0.52(0.00)*	-0.52(0.00)*	0.53(0.06)

a. TLWCC的值是100个聚类结果计算NMI的平均值，其他的值是相应算法NMI均值与TLWCC的差值。括号中的值是100个结果的标准差，“*”表示这个差值是显著的。

k-means[15]，WKM[16]和BBAC[17]等3种算法的聚类性能。对于WKM和TLWCC等含有参数的算法，本实验使用如下方法来设置合适的参数：给定参数范围是(1.0E-6,100)，参数以2为步长的等比数列取值，对于每一组参数运行得到一个结果，选出使得聚类结果最好的参数作为该算法的最优参数。

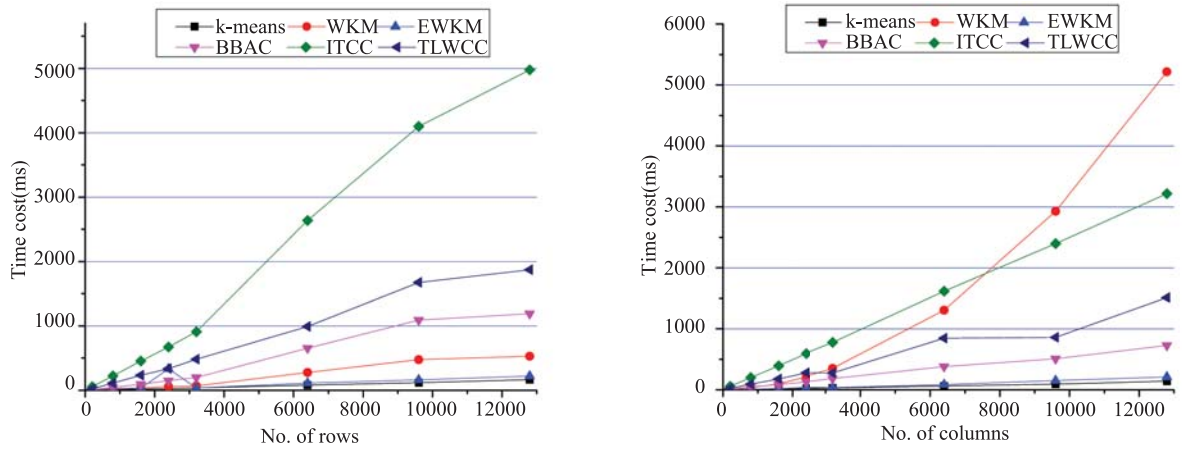
使用上面得到的最优参数，对每个算法的聚类结果计算正规化互信息，将运行100次后的结果进行处理，可以得到如表1的结果。

从表1中我们可以看出：（1）总体上看，TLWCC在这6个数据上比其他三个算法都有显著的优势，尤其是BBAC算法，TLWCC通过对BBAC引入双层权重，显著提高了算法抵抗噪声的能力；（2）对于k-means、WKM和BBAC这3个不考虑子空间权重的算法，随着错分率的提高，这三个算法聚类结果下降非常明显，而使用子空间加权方法的TLWCC在错分率增加时，聚类结果变化不是很大，这也说明了TLWCC算法有较好的抗噪声干扰能力。

4.5　时间复杂度实验

为了分析算法的时间性能和可扩展性，我们使用两组模拟数据S_1和S_2，并对比了其他5种聚类或协同聚类算法——k-means、WKM、EWKM、BBAC和ITCC。在本实验中，每个算法都采用随机初始化，运行100遍，然后将运行时间取平均，结果如图5所示：

从图5中我们可以看出：在(a)图中，对于数据集S_1，数据的列数固定，行数变化，TLWCC算法的运行时间与行数成正比；在(b)图中，对于数据集S_2，数据的行数固定，列数变化，TLWCC算法的运行时间与列数成正比。该运行时间基本符合时间复杂度公式$O(rNM(K+L))$。另外，TLWCC算法的运行时间也不到BBAC的两倍，仅为ITCC的1/3至1/2，具有良好的时间性能。结合图（a）和（b）可以看出，WKM算法运行时间随着数据的行数增长而线性增长，而随着列数的增长，运行时间呈加速增长的趋势。其主要原因是，WKM算法计算权重步骤耗时非常多，随着列数增加，需要计算的权重数量也随之增加，导致整体运行时间加速增加。



(a)　　(b)

图5　(a)保持数据的列数不变，变化行数，各算法运行时间。(b)保持数据的行数不变，变化列数，各算法的运行时间。

5　总结与展望

本文提出了一种双层子空间加权协同聚类算法TLWCC，该算法通过在聚类过程中自动计算块、行、列三组权重，能够有效地降低噪声信息的权重，提高协同聚类抗噪声干扰的能力。我们在理论上和实验上分别分析了算法三组权重的特点、参数选取如何影响聚类结果、算法的聚类性能和算法的时间性能。实验结果表明：该算法能较好地识别出噪声，并减小噪声信息地权重；新算法的聚类结果随参数变化会发生显著变化，在给定参数合适的区间范围时，算法能取得理想的聚类效果；在处理某些噪声数据时，算法比起其他算法有更好的抗干扰能力；算法具备良好计算时间性能和可扩展性。

同时，该算法也存在一些问题：算法的三个参数如何给定，目前没有比较好的方法。如何提出一种合适的选取参数的方法，使得TLWCC算法能够得到稳定的良好的聚类结果，这还有待于以后更加深入的研究。

参 考 文 献

[1] Madeira S C, Oliveira A L. Biclustering algorithms for biological data analysis: a survey [J]. Computational Biology and Bioinformatics, IEEE/ACM Transactions on, IEEE, 2004, 1(1): 24–45.

[2] Song Y, Pan S, Liu S, et al. Constrained Text Co-Clustering with Supervised and Unsupervised Constraints [J]. IEEE Transactions on Knowledge and Data Engineering, IEEE, 2012.

[3] George T, Merugu S. A scalable collaborative filtering framework based on co-clustering [C] // Data Mining, Fifth IEEE International Conference on. 2005: 4–pp.

[4] Li J, Shao B, Li T, et al. Hierarchical Co-Clustering: A New Way to Organize the Music Data [J]. IEEE Transactions on Multimedia, 2012, 14(2): 471–481.

[5] Fan N, Boyko N, Pardalos P M. Recent advances of data biclustering with application in computational neuroscience [J]. Computational Neuroscience, Springer, 2010: 105–132.

[6] Guo G, Chen S, Chen L. Soft subspace clustering with an improved feature weight self-adjustment mechanism [J]. International Journal of Machine Learning and Cybernetics, Springer, 2012, 3(1): 39–49.

[7] Chen X J, Ye Y M, Huang J Z. A feature group weighting method for subspace clustering of high-dimensional data [J]. Pattern Recognition, 2012, 45(1): 434–446.

[8] Deng Z, Choi K S, Chung F L, et al. Enhanced soft subspace clustering integrating within-cluster and between-cluster information [J]. Pattern Recognition, Elsevier, 2010, 43(3): 767–781.

[9] Jing L, Ng M, Huang Z. An Entropy Weighting k-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data [J]. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(8): 1026–1041.

[10] Hartigan J A. Direct clustering of a data matrix [J]. Journal of the American Statistical Association, JSTOR, 1972: 123–129.

[11] Dhillon I S, Mallela S, Modha D S. Information-theoretic co-clustering [C] // Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. 2003: 89–98.

[12] Cho H, Dhillon I S, Guan Y, et al. Minimum sum-squared residue co-clustering of gene expression data [C] // Proceedings of the fourth SIAM international conference on data mining. 2004, 114.

[13] Lomet A, Govaert G, Grandvalet Y. Design of Artificial Data Tables for Co-Clustering Analysis [R]. France: 2012.

[14] Manning C D, Raghavan P, Schütze H. Introduction to Information Retrieval [M]. Cambridge University Press Cambridge, 2008, 1.

[15] MacQueen J B. Some methods for classification and analysis of multivariate observation [J]. Proceedings of the 5th Berkeley Symposium on Mathematical Statistica and Probability, 1967: 281–297.

[16] Huang Z, Ng M, Rong H, et al. Automated Variable Weighting in k-Means Type Clustering [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(5): 657–668.

[17] Banerjee A, Dhillon I, Ghosh J, et al. A Generalized Maximum Entropy Approach to Bregman Co-clustering and Matrix Approximation [J]. Journal of Machine Learning Research, 2007, 8: 1919–1986.