一种基于同类约束的半监督近邻反射传播聚类方法

徐明亮 [1, 2] 王士同 [1] 杭文龙 [1]

摘 要 以近邻反射传播 (Affinity propagation, AP) 聚类算法为基础, 提出了一种基于同类约束的半监督近邻反射传播聚类方法 (Semi-supervised affinity propagation clustering method with homogeneity constraints, HCSAP). 该方法在聚类目标函数中引入同类约束项, 以保证聚类结果与同类集先验信息一致. 利用最大和信任传播 (Max-sum belief propagation) 优化过程对目标函数进行求解, 导出同类约束下的吸引度 (Responsibility) 和归属度 (Availability) 的迭代方程. 人工数据集和真实数据集上的实验结果表明本文所提方法的有效性.

关键词 半监督聚类, 近邻反射传播, 最大和, 信任传播, 同类约束

引用格式 徐明亮, 王士同, 杭文龙. 一种基于同类约束的半监督近邻反射传播聚类方法. 自动化学报, 2016, **42**(2): 255−269

DOI 10.16383/j.aas.2016.c150059

A Semi-supervised Affinity Propagation Clustering Method with Homogeneity Constraint

XU Ming-Liang[1, 2] WANG Shi-Tong[1] HANG Wen-Long[1]

Abstract In this paper, a semi-supervised affinity propagation (AP) clustering algorithm with homogeneity constraint, called HCSAP (semi-supervised affinity propagation clustering method with homogeneity constraints), is proposed. To keep consistency between the clustering results and the priori information about homogeneity sets, the constraint terms are introduced to the objection function of algorithm AP. With the max-sum belief propagation procedure, the objection function can be resolved into the corresponding responsibility and availability update equations. Experiments on synthetic dataset and real-world datasets indicate the effectiveness of the proposed HCSAP.

Key words Semi-supervised clustering, affinity propagation (AP), max-sum, belief propagation, homogeneity constraints

Citation Xu Ming-Liang, Wang Shi-Tong, Hang Wen-Long. A semi-supervised affinity propagation clustering method with homogeneity constraints. *Acta Automatica Sinica*, 2016, **42**(2): 255−269

近邻反射传播聚类 (Affinity propagation, AP)[1] 是以数据之间的相似度为基础, 通过迭代更新吸引度 (Responsibility) 和归属度 (Availability) 进行数据聚类. 与 K-means 聚类相比, 其特点是: 1) 不需要预先指定类中心, 由吸引度和归属度的自动确定类中心; 2) 聚类的类数也不需要预先指定, 在设置好参数 Preference 后由算法本身自动决定聚类的类数; 3) 既适用于相似度对称的场合也适用于相似度非对称的场合, 因此应用范围更广; 4) 结合稀疏处理方式可适合于大规模数据聚类; 5) 聚类速度较快. 近邻反射传播聚类已成功应用于航线规划[1]、图像分割[2]、基因组识别[3]、目标识别[4] 等领域, 受到众多研究者的重视[5].

近邻反射传播聚类是一种无监督学习方法, 而在聚类分析中往往具有部分数据的先验知识, 如已知某些数据属于同一类或部分数据的标签信息等. 利用好这些先验知识将有助于聚类性能的提高. 将先验知识用于聚类, 一般有两种途径[6]: 一类是基于相似度的半监督聚类, 即利用先验知识对数据点之间的相似度进行修改, 然后再进行聚类. 如 Bijral 等[7] 在密度距离估计的基础上, 提出一种图上最短路径的简单有效的计算方法, 并将该方法用于稠密的全连接图, 进而有效提高聚类速度. 在近邻反射传播聚类范畴中, 基于相似度的半监督近邻反射传播聚类算法的工作已较为丰富. 该类方法的主要特点是根据 must-link 和 cannot-link 成对约束对相似性矩阵进行局部调整. must-link 成对约束是指受约束的两个点属于同一类, 而 cannot-link 成对约束是指受约束的两个点不在同一类中[8].

收稿日期 2015-01-30 录用日期 2015-08-17
Manuscript received January 30, 2015; accepted August 17, 2015
国家自然科学基金 (61170122, 61202311, 61272210), 江苏省自然科学基金 (BK2012552) 资助
Supported by National Natural Science Foundation of China (61170122, 61202311, 61272210) and Natural Science Foundation of Jiangsu Province (BK2012552)
本文责任编委 封举富
Recommended by Associate Editor FENG Ju-Fu
1. 江南大学数字媒体学院无锡 214122 2. 无锡城市职业技术学院无锡 214153
1. School of Digital Media, Jiangnan University, Wuxi 214122 2. Wuxi City College of Vocational Technology, Wuxi 214153

据此, 肖宇等提出了 SAP 算法[9]. SAP 算法将 cannot-link 成对约束点之间的相似性度量置为 $-\infty$, 将 must-link 成对约束点之间的相似性度量置为 0, 同时根据最短路径调整 must-link 约束点和其他点之间的相似性度量. 然后在新的相似度矩阵上使用近邻反射传播聚类算法. 在 SAP 算法基础上, 张震等提出了基于分层的 SAP-SC 方法[10], 该方法将 AP 聚类过程等分成若干层聚类, 通过构造成对点约束和使用子簇标签映射进行半监督学习, 采用基于组合提升的方法将各层聚类结果加权叠加. 文献 [11] 根据通过数据流密度变化, 以加权方式对相似性度量进行调整, 然后再进行 AP 聚类. Givoni 等根据 must-link 成对约束所具有传递性和闭包特性, 提出 SSAP 算法[12], 该算法将具有相同标签的数据点构成同类集, 再由同类集派生出一个虚点, 利用虚点对原始相似性矩阵进行扩展. 实点之间的相似性度量不变, 实点和虚点之间的相似性度量则根据实点是否包含于虚点所对应的同类集来决定. 当实点是虚点所对应的同类集中的元素时, 两者之间的距离置 0, 否则相似度量值置为实点与同类集中各个数据点之间相似度的最大值. 同时将 cannot-link 成对约束点间的相似性度量置为 $-\infty$. 基于相似度的半监督近邻反射传播聚类存在以下不足: 1) 并不能保证聚类结果与成对约束一致[13]. 如将 must-link 成对约束点之间的相似性度量置 0, 能提高约束点选择彼此作类中心点的可能性, 但不能保证约束点选择的中心点相同. 特别地, 当数据点 A, B, C 两两之间均存在 must-link 成对约束时, 如果数据点 A 和 B 被选为不同类别的类中心点时, 由于数据点 C 和数据点 A, B 的相似度都为 0, 因此数据点 C 可以任意选择其中之一作为自己的类中心点而不会改变聚类目标函数值, 而且也会导致聚类后的调整方法失效[6]. 这样就会导致 must-link 成对约束点并不在同一类中. 2) 这种处理方式容易产生多个并列中心点, 因此可能导致更多的震荡现象[13]. 3) 相似性度量置 0 改变了算法适用于相似性矩阵不对称的特点. 另一类是基于约束的半监督聚类, 即利用先验信息对聚类算法添加约束, 使得聚类结果与先验信息保持一致. 在文献 [14] 中的 MPCK-MEANS 聚类方法在考虑簇与簇之间存在的差异的基础上, 借助于无标记数据和成对约束样本, 在进行簇指派的迭代过程中进行度量学习, 同时实现了约束聚类和局部马氏距离的学习. 尹学松等在文献 [15] 中提出了 DSCA 方法, 该方法首先利用 must-link 成对约束和 cannot-link 成对约束得到投影矩阵, 在投影空间中对数据聚类得到聚类标号, 然后利用线性判别分析 (Linear discriminant analysis, LDA) 选择子空间, 最后使用基于成对约束的 K 均值算法对子空间中的数据聚类. 基于约束的半监督聚类根据先验信息在聚类算法中添加适当的约束, 确保聚类结果与先验信息一致, 因此较之于基于相似度的半监督聚类, 基于约束的半监督策略更显有效. 在近邻反射传播聚类算法范畴内, 基于约束的半监督近邻反射传播聚类算法目前还未发现有相关研究. 本文以 must-link 成对约束为基础, 提出一种基于同类约束的半监督近邻反射传播聚类方法.

1 近邻反射传播算法

在近邻反射传播学习算法中, 通过吸引度和归属度迭代, 让每个数据点在全体数据点集中找自己最优的类中心点, 最终使所有数据点到各自的类中心点的相似度之和最大, 同时满足一致性条件. 吸引度 $r(i,j)$ 和归属度 $a(i,j)$ 迭代方程如式 (1) 和式 (2) 所示.

$$r\left(i,j\right)=s(i,j)-\max_{k\neq j}\left[s(i,k)+a\left(i,k\right)\right] \tag{1}$$

$$a\left(i,j\right)=\begin{cases}\min[0,r\left(j,j\right)+\\ \quad\sum\limits_{k\notin\{i,j\}}\max\left[0,r\left(k,j\right)\right]], & j\neq i\\ \sum\limits_{k\neq j}\max\left[0,r\left(k,j\right)\right], & j=i\end{cases} \tag{2}$$

式中, $s(i,j)$ 为数据点 i, j 之间的相似度. 吸引度 $r(i,k)$ 由数据点 i 指向数据点 k, 表示数据点 k 适合作为数据点 i 的类中心点的程度; 归属度 $a(i,k)$ 是从数据点 k 指向数据点 i, 表示数据点 i 选择数据点 k 作为其类代表点的合适程度. 迭代开始时吸引度和归属度均初始化为 0, 使每一个数据点都成为潜在的类中心点, 通过迭代使得式 (3) 所示的聚类目标函数最大化.

$$S\left(C\right)=\sum_{i=1}^{N}s\left(i,c_i\right)+\sum_{k=1}^{N}\delta_k\left(C\right) \tag{3}$$

式中, c_i 为数据点 i 的类中心点, C 为由 c_i 构成的分配向量, $i=1,\cdots,N$ (N 为数据点个数). $S(C)$ 为所有数据点到各自的类中心点的相似度之和. $\delta_k(C)$ 如式 (4) 所示.

$$\delta_k\left(C\right)=\begin{cases}-\infty, & c_k\neq k \text{ 但 } \exists\, i,\ c_i=k\\ 0, & \text{其他}\end{cases} \tag{4}$$

该项为一致性约束惩罚项, 若有某个数据点 i 选择 k 作为其类代表点, 即 $c_i=k$, 那么数据点 k 必须选择自身作为类代表点, 即 $c_k=k$, 否则函数取值

$-\infty$, 使数据点 i 在下一次迭代中不再选择数据点 k 作为自己的中心点.

各个数据点的类中心点 c_i 按式 (5) 计算得到:

$$c_i = \arg\max_k \left[a\left(i,k\right) + r\left(i,k\right)\right] \tag{5}$$

即数据点选择彼此间吸引度和归属度累加最大的数据点为自己的类中心点.

2　同类约束近邻反射传播半监督聚类

类属相同的数据点所构成的集合称为同类集. 为保证聚类结果和同类集先验信息保持一致, 在聚类目标函数中引入同类约束. 并将此方法称为基于同类约束的半监督近邻反射传播聚类方法 (Semi-supervised affinity propagation clustering method with homogeneity constraints, HCSAP).

2.1　部分符号说明

为方便描述, 文中所涉及的部分符号示例说明如表 1 所示.

2.2　HCSAP 聚类目标函数

为描述方便, 记待聚类数据点集 $W = \{1, 2, \cdots, N\}$. 对同类点集 $p_m \subseteq W$, $m = 1, 2, \cdots, M$, 有 $\forall i, j \in p_m$, $c_i = c_j$. 记 $P = \{p_m | m = 1, 2, \cdots, M\}$, $\bar{P} = W - \bigcup_{m=1}^{M} p_m$.

HCSAP 聚类目标函数定义如下:

$$\begin{aligned} T\left(C\right) = & \sum_{c_{ij}} S_{ij}c_{ij} + \sum_{i=1}^{N} I_i\left(c_{i1}, \cdots, c_{iN}\right) + \\ & \sum_{j=1}^{N} G_j\left(c_{1j}, \cdots, c_{Nj}\right) + \\ & \sum_{k=1}^{N} F_k\left(c_{1k}, \cdots, c_{Nk}\right) \end{aligned} \tag{6}$$

其中,

$$S_{ij} = \begin{cases} 0, & c_{ij} = 0, \\ s\left(i,j\right), & c_{ij} = 1, \end{cases} \quad i, j = 1, \cdots, N \tag{7}$$

$$\sum_{i=1}^{N} I_i\left(c_{i1}, \cdots, c_{iN}\right) = \begin{cases} -\infty, & \sum\limits_{j=1}^{N} c_{ij} \neq 1 \\ 0, & \text{其他} \end{cases} \tag{8}$$

$$\sum_{j=1}^{N} G_j\left(c_{1j}, \cdots, c_{Nj}\right) = \begin{cases} -\infty, & c_{jj} = 0 \text{ 且 } \exists i \neq j \text{ s.t. } c_{ij} = 1 \\ 0, & \text{其他} \end{cases} \tag{9}$$

$$\sum_{k=1}^{N} F_k\left(c_{1k}, \cdots, c_{Nk}\right) = \begin{cases} -\infty, & \exists m, \exists h_1, h_2 \in p_m, \\ & \text{s.t. } c_{h_1k} \oplus c_{h_2k} = 1, \ m \in \{1, \cdots, M\} \\ 0, & \text{其他} \end{cases} \tag{10}$$

表 1　部分符号说明

Table 1　The explanation of some symbol

符号	意义
N	聚类数据点个数
M	同类集个数
h_1, h_2	数据点 h_1, h_2
c_{ij}	变量节点, 为 0 表示 j 不是 i 的类中心点; 为 1 表示 j 是 i 的类中心点
$E_{ij}(\cdot)$	$E_j(c_{1j}, \cdots, c_{Nj})$ 数据点 j 的同类约束与一致性约束函数
ρ_{ij}	表示变量节点 c_{ij} 向函数节点 E_j 所发送的标量信息
c_i	数据点 i 的类中心点
P	全体同类集所构成的集合
p^i	数据点 i 所在的同类约束集
$I_i(\cdot)$	$I_i(c_{i1}, \cdots, c_{iN})$ 为数据点 i 的唯一性约束函数
α_{ij}	表示函数节点 E_j 向变量节点 c_{ij} 所发送的标量信息
β_{ij}	表示变量节点 c_{ij} 向函数节点 I_i 所发送的标量信息
p_v	第 v 个同类集
$\oplus$	异或
$\bar{P}$	无同类约束的数据点集
$S_{ij}(\cdot)$	定义在数据点 i, j 之间的相似度函数
$s(i,j)$	数据点 i, j 之间的相似度
η_{ij}	表示函数节点 I_i 向变量节点 c_{ij} 所发送的标量信息

与式 (3) 不同, 式 (6) 中的 C 为由 c_{ij} 构成的分配矩阵, 且 c_{ij} 只取 0 或 1. $\sum_{c_{ij}} S_{ij}c_{ij}$ 为定义在分配矩阵上的函数, 表示数据点到各自聚类中心点的距离之和. $\sum_i I_i\left(c_{i1}, \cdots, c_{iN}\right)$ 为唯一性约束惩罚项. 唯一性约束是要求每个数据点只能有一个中心点, 即数据点只能属于某一个类. 如果违背这一约束, 则函数值为 $-\infty$. $\sum_j E_j\left(c_{1j}, \cdots, c_{Nj}\right)$ 为一致

性约束惩罚项. 一致性约束项是指如果该数据点不是中心点, 即 $c_{jj}=0$, 那么其余数据点不可以选择该数据点为中心点. 如果违反该规则, 即 $\exists i\neq j$, 有 $c_{ij}=1$, 则函数值为 $-\infty$. 该约束也要求被选作中心点的数据点必须选择自身为中心点. $\sum_k F_k(c_{1k}, \cdots, c_{Nk})$ 为同类约束惩罚项. 同类约束要求同类集中的所有数据点选择的类中心点必须相同. 如果同类集中存在两个数据点的中心点不相同, 即 $\exists m$, $\exists h_1, h_2 \in p_m$, 使得 $c_{h_1k} \oplus c_{h_2k}=1$, 则违反了该约束, 函数取值为 $-\infty$, 进而保证聚类结果和同类约束相一致.

为简化因子图, 将式 (9) 和式 (10) 合并为式 (11).

$$\sum_{j=1}^{N} E_j\left(c_{1j}, \cdots, c_{Nj}\right)= \begin{cases} -\infty, & c_{jj}=0 \text{ 且 } \exists i\neq j, \\ & \text{s.t. } c_{ij}=1,\ i\in\{1,\cdots,N\} \\ & \text{或 } \exists m,\ h_1\in p_m,\ h_2\in p_m, \\ & \text{s.t. } c_{h_1j}\oplus c_{h_2j}=1,\ m\in\{1,\cdots,M\} \\ 0, & \text{其他} \end{cases} \tag{11}$$

则聚类目标函数变为式 (12).

$$T(C)=\sum_{c_{ij}} S_{ij}c_{ij} + \sum_{i=1}^{N} I_i\left(c_{i1}, \cdots, c_{iN}\right)+ \sum_{j=1}^{N} E_j\left(c_{1j}, \cdots, c_{Nj}\right) \tag{12}$$

2.3 目标函数求解

聚类目标函数 (12) 求解是一个 NP-hard 寻优过程, 目前有效的求解方法是利用最大和 (max-sum) 置信传播算法求解出最优解. 置信传播算法将全局函数表示为由变量节点、函数节点以及函数节点和变量节点之间的邻接所构成的因子图, 通过在变量节点和函数节点之间传递标量信息实现置信度最大化, 最终实现全局函数的最小化[16−17]. 变量节点 x 向函数节点 f 的发送的标量信息 $\mu_{x\to f}(x)$ 如式 (13) 所示[16]:

$$\mu_{x\to f}(x)=\sum_{\{f'|f'\in ne(x)\backslash f\}} \mu_{f\to x}(x) \tag{13}$$

式中, $ne(x)$ 为与变量节点 x 相邻接的所有函数节点所构成的集合, $\mu_{f\to x}(x)$ 为函数节点 f 向变量节点 x 的发送的标量信息. 由式 (13) 可知变量 x 到函数 f 的标量信息为该变量所有其他邻接函数节点发送给该变量的标量信息之和.

对函数节点向变量节点的发送的标量信息 $\mu_{f\to x}(x)$ 有[16−17]:

$$\mu_{f\to x}(x)=\max_{x_1,\cdots,x_k}\left[f(x,x_1,\cdots,x_k)+\sum_{\{x'|x'\in ne(f)\backslash x\}}\mu_{x'\to f}(x)\right] \tag{14}$$

式中, $ne(f)$ 为与函数节点 f 相邻接的所有变量节点所构成的集合. 由式 (14) 可知函数节点 f 到变量 x 节点的标量信息为全体邻接变量节点的发送至本函数节点标量信息和函数节点自身信息累加的最大值[18]. max-sum 置信传播算法的具体内容可参见文献 [16−17].

聚类目标函数 $T(C)$ 的因子图如图 1 所示. 其中矩形框表示函数节点, 分别和式 (7)、式 (8)、式 (11) 所示的函数相对应. 圆形框表示变量节点, 即函数变量 c_{ij}. 变量节点和函数节点之间的信息传递如图 2 所示. 其中 ρ_{ij} 和 β_{ij} 为变量节点 c_{ij} 向函数节点 E_j, I_i 所发送的信息 (变量节点 c_{ij} 向函数节点 $s(i,j)$ 发送的信息为 0, 图中未列出). α_{ij}, η_{ij}, $s(i,j)$ 分别为函数节点 E_j, I_i, S_{ij} 向变量节点 c_{ij} 所发送的信息.

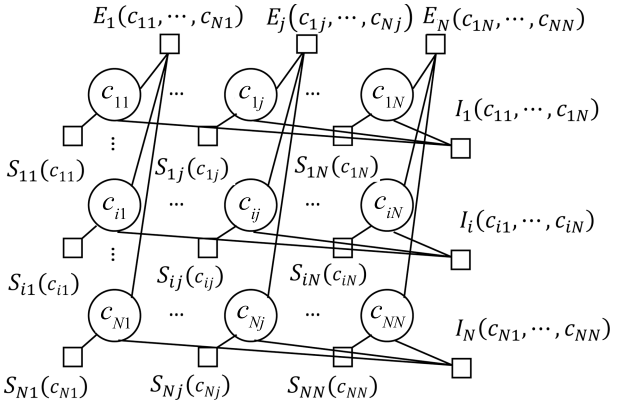


图 1 HCSAP 因子图

Fig. 1 Factor graph of HCSAP

由于变量节点只取 0 和 1 两个值, 因此节点间传播的标量信息只需考虑这两种取值情况下的差值[12−13], 即:

$$\rho_{ij}=\rho_{ij}(c_{ij}=1)-\rho_{ij}(c_{ij}=0)$$
$$\alpha_{ij}=\alpha_{ij}(c_{ij}=1)-\alpha_{ij}(c_{ij}=0)$$
$$\beta_{ij}=\beta_{ij}(c_{ij}=1)-\beta_{ij}(c_{ij}=0)$$
$$\eta_{ij}=\eta_{ij}(c_{ij}=1)-\eta_{ij}(c_{ij}=0)$$

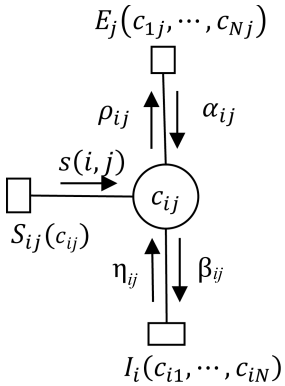


图 2　HCSAP 信息

Fig. 2　Message of HCSAP

对于变量节点 c_{ij} 至函数节点 E_j 的信息 ρ_{ij}, 至函数节点 I_i 的信息 β_{ij}, 由式 (13) 分别有:

$$\beta_{ij} = \alpha_{ij} + s(i,j) \tag{15}$$

$$\rho_{ij} = \eta_{ij} + s(i,j) \tag{16}$$

函数节点 I_i 到变量节点 c_{ij} 的信息 η_{ij} 和相邻节点之间的信息关系如图 3 所示.

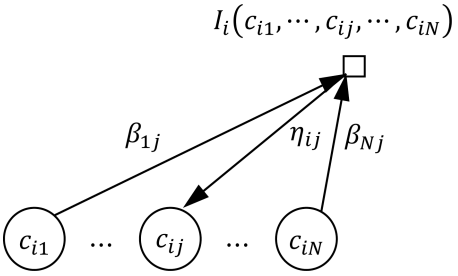


图 3　η_{ij} 与其相关信息关系图

Fig. 3　Relationship among η_{ij} and its correlative message

根据式 (14) 和图 3 有 (具体推导详见附录 A):

$$\eta_{ij} = \eta(1) - \eta(0) = -\max_{t\neq j}(\beta_{it}) \tag{17}$$

函数节点 E_j 到变量节点 c_{ij} 的信息 α_{ij} 和相邻节点之间的信息关系如图 4 所示. 由于 α_{ij} 求解与 c_{ij} 在因子图中的位置以及数据点 i, j 和同类集之间的关系有关, 因此需要按照不同的情况予以讨论. 同理, 根据式 (14) 可以推导出 α_{ij} 与相关信息的关系, 如式 (18) 所示 (其推导过程详见附录 A).

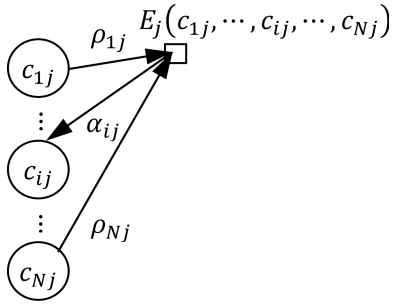


图 4　α_{ij} 与其相关信息关系图

Fig. 4　Relationship among α_{ij} and its correlative message

在式 (18) 中, p^i, p^j 分别表示数据点 i, j 所在的同类集.

为避免震荡现象, 引入阻尼因子对信息进行更新, 即对任意信息 μ, 按式 (19) 进行更新:

$$\mu = \lambda\mu_{old} + (1-\lambda)\mu_{new} \tag{19}$$

其中, λ 为阻尼因子, 取值范围在 $0\sim1$ 之间.

$$\alpha_{ij} = \begin{cases} \min\left[\rho_{jj} + \sum\limits_{k\in\bar{P}\backslash i,j}\max[\rho_{kj},0] + \sum\limits_{p_v\in P}\max\left[\sum\limits_{k\in p_v}\rho_{kj},0\right],0\right], & i\neq j,\ i\in\bar{P},\ j\in\bar{P} \\ \min\left[\sum\limits_{k\in p^j}\rho_{kj} + \sum\limits_{k\in\bar{P}\backslash i}\max[\rho_{kj},0] + \sum\limits_{p_v\in P-p^j}\max\left[\sum\limits_{k\in p_v}\rho_{kj},0\right],0\right], & i\neq j,\ i\in\bar{P},\ j\in p^j \\ \min\left[\sum\limits_{k\in p^i\backslash i}\rho_{kj} + \rho_{jj} + \sum\limits_{k\in\bar{P}\backslash j}\max[\rho_{kj},0] + \sum\limits_{p_v\in P-p^i}\max\left[\sum\limits_{k\in p_v}\rho_{kj},0\right],\sum\limits_{k\in p^i\backslash i}\rho_{kj}\right], & i\neq j,\ i\in p^i,\ j\in\bar{P} \\ \min\left[\sum\limits_{k\in p^i\backslash i}\rho_{kj} + \sum\limits_{k\in\bar{P}}\max[\rho_{kj},0] + \sum\limits_{p_v\in P-p^i}\max\left[\sum\limits_{k\in p_v}\rho_{kj},0\right],0\right], & i\neq j,\ i\in p^i,\ j\in p^j,\ p^i=p^j \\ \min\left[\sum\limits_{k\in p^j}\rho_{kj} + \sum\limits_{k\in p^i\backslash i}\rho_{kj} + \sum\limits_{k\in\bar{P}}\max[\rho_{kj},0] + \sum\limits_{p_v\in P-p^i}\max\left[\sum\limits_{k\in p_v}\rho_{kj},0\right],0\right], & i\neq j,\ i\in p^i,\ j\in p^j,\ p^i\neq p^j \\ \sum\limits_{k\in p^j\backslash i}\rho_{kj} + \sum\limits_{k\in\bar{P}}\max[\rho_{kj},0] + \sum\limits_{p_v\in P-p^j}\max\left[\sum\limits_{k\in p_v}\rho_{kj},0\right], & i=j,\ j\in p^j \\ \sum\limits_{k\in\bar{P}\backslash i}\max[\rho_{kj},0] + \sum\limits_{p_v\in P}\max\left[\sum\limits_{k\in p_v}\rho_{kj},0\right], & i=j,\ j\in\bar{P} \end{cases} \tag{18}$$

2.4 分配矩阵计算

数据点 i 类中心点 c_i 是由接收信息总量最大的变量节点 c_{ij} 节点所对应的 j 来确定, 其计算按式 (20) 进行

$$c_i = \arg\max_j \left(\alpha_{ij} + \eta_{ij} + s(i,j)\right) \tag{20}$$

由式 (16) 有

$$c_i = \arg\max_j \left(\alpha_{ij} + \rho_{ij}\right) \tag{21}$$

根据式 (15) ∼ (17) 有

$$\rho_{ij} = s(i,j) - \max_{t\neq j}\left[\alpha_{it} + s(i,t)\right] \tag{22}$$

由式 (22) 可知, ρ_{ij} 仅与 α_{ij} 和 $s(i,j)$ 有关; 由式 (18) 可知 α_{ij} 仅与 ρ_{ij} 有关. 结合式 (21)、式 (22) 和式 (18) 可知只需迭代计算 α_{ij} 和 ρ_{ij} 即可. 对照式 (1) 和式 (2), ρ_{ij} 和 α_{ij} 即为 HCSAP 的吸引度和归属度.

3 实验及结果分析

3.1 聚类评价参数及相关设置

为验证 HCSAP 算法的有效性, 利用人工数据集及部分真实数据集对 HCSAP 算法进行了聚类测试, 并与基于距离的近邻反射传播半监督聚类 SAP[5]、SSAP[6], 以及基于约束的 MPCK-MEANS[13]、DSCA[14] 聚类算法进行比较.

聚类数据均进行归一化处理. 聚类实验前, 先按照一定的抽取率以均匀随机的方式从聚类数据中抽取一定数量的样本, 然后由这些样本的标签来构造算法中所需要的同类集和 must-link 成对约束. SAP、SSAP、HCSAP 聚类算法中阻尼因子取值为 0.9, 迭代次数取为 100, 聚类初始时, 吸引度和归属度均初始化为 0. MPCK-MEANS 和 DSCA 聚类的类数取 HCSAP 聚类数, 其他相关参数设置与文献 [13−14] 相同.

每个聚类测试运行 10 次, 取平均值和方差进行比较. 同时以 HCSAP 为参照, 与对比算法进行成对 t 检验. 聚类的评价指标采用成对 F 评测指标和 Pure 指标, 其中成对 F 评测指标计算如式 (23) 所示.

$$F = \frac{2\times x\times y}{x+y} \tag{23}$$

式中, x 为准确率 (Precision), y 为召回率 (Recall), 计算分别按式 (24) 和式 (25) 进行.

$$x = \frac{\text{pairs correctly predicted in same cluster}}{\text{total pairs predicted in same cluster}} \tag{24}$$

$$y = \frac{\text{pairs correctly predicted in same cluster}}{\text{total pairs in same cluster}} \tag{25}$$

Pure 指标按式 (26) 进行计算.

$$purity(\Omega,\Psi) = \frac{1}{N}\sum_k \max_j |\omega_k \cap \psi_j| \tag{26}$$

式中, $\Omega = \{\omega_1, \cdots, \omega_K\}$ 为分类集合, N 为样本数据个数, $\Psi = \{\psi_1, \cdots, \psi_N\}$, 为聚类集合, ω_k 为表示第 k 类的数据点集合.

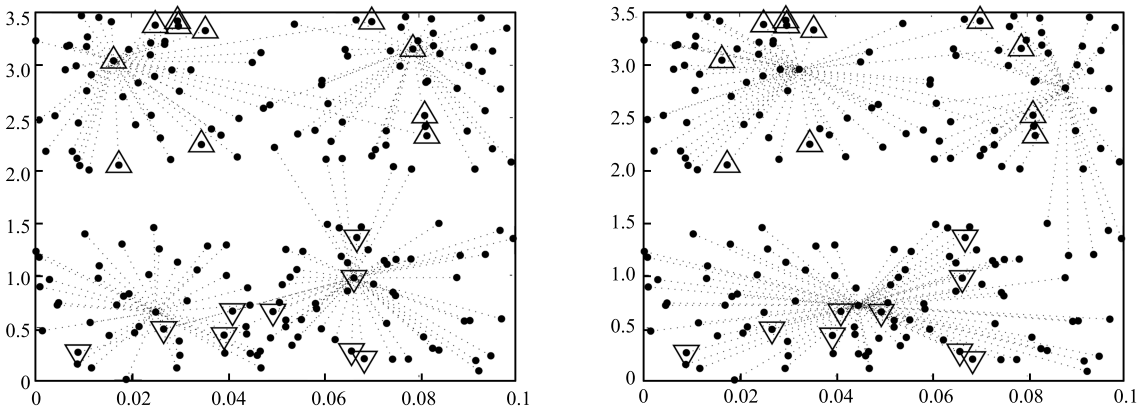
实验环境为戴尔 XPS 8700-r398 型台式电脑, 配置为 Visual Studio 2012 Ultimate, Windows 7 64 位操作系统, Intel 酷睿 i7-4790 3.4 GHz 处理器, 16 GB DDR3 1 600 MHz 内存.

3.2 人工数据集实验

随机产生 200 个二维数据, 数据集横坐标均匀分布在 0 ∼ 0.1 之间, 其中 100 数据点纵坐标均匀分布在 0 ∼ 1.5 之间, 另外 100 个数据点纵坐标均匀分布在 2 ∼ 3.5 之间. SAP、SSAP、HCSAP 聚类参数 (Preference) 设为相似性矩阵中值的 20 倍.

图 5 (a) 和图 5 (b) 给出了抽取率为 10 % 时的 SAP 和 HCSAP 一次聚类结果. 图中上三角和下三角分别为随机抽取的 20 个数据所构成两个同类集. 即所有上三角的数据为一类, 所有下三角数据为一类. 从图 5 (a) 的 SAP 聚类结果来看, 下三角同类数据点都聚在一类中, 这与约束是一致的. 但所有上三角数据点并没有聚为一类. 根据聚类结果, 图中右上角中的一个被选为类中心点的上三角数据点和其他上三角点分属两个不同的类别, 因此聚类结果和先验信息并不一致. 图 5 (b) 给出了 HCSAP 聚类结果. 上三角数据点被聚为一类, 下三角数据点也被聚为一类, 因此聚类结果与先验信息一致. 可见采用基于同类约束的 HCSAP 半监督聚类方法, 能够保证聚类结果和同类集先验信息一致.

表 2 列出了各算法在抽取率分别为 0 % ∼ 50 % 时 10 次聚类结果的 F-评测指标和 Pure 指标的平均值、方差和成对 t 检验值. 抽取率为 0 时, SAP、SSAP、HCSAP 都退化为近邻反射传播聚类算法. MPCK-mean、DSCA 算法退化为 K-means 算法, 从结果可以看出 AP 聚类效果好于 K-means. 在引入先验信息后, HCSAP 聚类结果的聚类 F-measure 参数与 Pure 参数即优于基于距离的 SAP 和 SSAP 算法, 也优于基于约束的 MPCK-mean 和



(a) 抽样率 10% SAP 聚类结果
(a) The clustering result of SAP with 10% sample

(b) 抽样率 10% HCSAP 聚类结果
(b) The clustering result of HCSAP with 10% sample

图 5　人工数据集聚类实例

Fig. 5　The instance of clustering on man-made dataset

表 2　人工数据上的聚类结果参数对比

Table 2　Performance comparison on man-made dataset

Sample rate (%)	Item	F-measure (%)					Pure (%)				
		HCSAP	SAP	SSAP	MPCK-MEAN	DSCA	HCSAP	SAP	SSAP	MPCK-MEAN	DSCA
0	Mean	**72.12**	**72.12**	**72.12**	70.31	69.3	**56.50**	**56.50**	**56.50**	53.43	55.0
	std	(0)	(0)	(0)	(1.3)	(3.6)	(0)	(0)	(0)	(1.1)	(2.1)
	p-value	–	–	–	4.2E−3	6.6E−3	–	–	–	1.8E−1	3.4E−1
10	Mean	**87.24**	82.74	80.22	85.27	81.66	**80.47**	72.41	70.50	77.39	75.4
	std	(6.6)	(4.7)	(0.8)	(5.2)	(9.2)	(1.7)	(1.1)	(2.4)	(2.1)	(3.7)
	p-value	–	3.1E−2 (+)	2.2E−5 (+)	9.7E−2	6.3E−3 (+)	–	3.9E−2 (+)	4.1E−5 (+)	6.4E−2	6.8E−2
20	Mean	**96.15**	80.45	81.78	90.00	88.6	**95.75**	72.57	73.66	90.41	76.8
	std	(1.0)	(1.6)	(1.8)	(4.1)	(4.5)	(4.0)	(1.6)	(2.7)	(0.5)	(1.4)
	p-value	–	9.4E−4 (+)	7.8E−5 (+)	9.2E−3 (+)	9.2E−3 (+)	–	1.7E−7 (+)	2.3E−6 (+)	1.4E−2 (+)	7.5E−6 (+)
30	Mean	**96.24**	91.24	92.47	90.36	91.33	**97.58**	89.00	85.74	90.06	89.6
	std	(2.0)	(4.1)	(5.1)	(0.8)	(1.6)	(0.2)	(6.6)	(4.1)	(0.2)	(2.8)
	p-value	–	4.9E−2 (+)	5.1E−2	8.4E−3 (+)	7.4E−3 (+)	–	3.4E−3 (+)	9.5E−8 (+)	1.6E−2 (+)	4.7E−4 (+)
40	Mean	**96.66**	88.57	87.98	90.21	89.2	**97.35**	88.65	86.97	90.33	90.5
	std	(1.3)	(3.0)	(2.7)	(0.2)	(4.5)	(2.0)	(1.2)	(0.9)	(0.5)	(7.7)
	p-value	–	5.7E−3 (+)	1.1E−3 (+)	6.6E−3 (+)	3.9E−3 (+)	–	1.1E−4 (+)	2.4E−7 (+)	1.8E−2 (+)	7.1E−3 (+)
50	Mean	**98.05**	90.34	88.84	90.70	90.8	**98.25**	89.65	90.45	88.87	90.0
	std	(0.2)	(7.1)	(1.4)	(0.6)	(2.3)	(0.2)	(2.8)	(9.6)	(0.7)	(3.4)
	p-value	–	5.1E−2	7.2E−4 (+)	9.0E−5 (+)	1.4E−4 (+)	–	42E−4 (+)	3.7E−7 (+)	8.4E−3 (+)	6.7E−3 (+)

注: 表中 p-value 为 5% 显著性水平下的 t 检验值, "+" 表示 HCSAP 在 5% 显著性水平下优于对比聚类算法. "–" 表示在 5% 显著性水平下 HCSAP 劣于对比聚类算法. 粗体字表示对比较优者 (下同) .

DSCA 算法.

3.3　标准数据集实验

从 UCI 数据库中选取 Optdigit, Ionosphere, Iris, Pendigits 和 Letter-recognition 数据集中的 I, J, L 字母数据子集[1], 以及 KEEL 网站[2] 中的 glass, wine, wdbc 数据集进行聚类实验. 表 3 中给出了这些数据集的相关信息和聚类参数 (Preference) 设置. 表中 Mid 是相似性矩阵中值.

Optdigit 数据集聚类结果的 F-measure、Pure 相关参数统计结果如表 4 所示. 从结果来看, 除了在 20% 抽取率时, HCSAP 聚类结果的 F-measure 参数不是最优, 但在其他抽取率时相关参数的均值都是最优的. 同时 Pure 结果均优于其他算法.

各算法在 Optdigit 数据集上平均运行用时如图 6 所示. 从图 6 可以看出, 以 AP 为基础的半监督聚类方法速度要快于以 K-MEANS 为基础的半监督聚类方法. 此外, 从迭代方程 (1) 和 (22) 来看, AP

[1] http://archive.ics.uci.edu/ml/-datasets.html
[2] http://www.keel.es

聚类和 HCSAP 聚类的吸引度迭代方程是相同的. 从迭代方程 (2) 和 (18) 来看, 当 $i = j$ 时 AP 聚类的 $\alpha(i,j)$ 迭代操作为 $n-1$ 次 max 运算和 $n-2$ 次加法运算 (n 为数据点个数), 而在 HCSAP 中 $\alpha(i,j)$ 迭代中首先要进行检索操作, 以便进行不同的迭代. 以 $j \in p^j$ 为例, 在迭代方程中, 运算量为 $\left|\overline{P}\right| + |p^j|$ 次 max 运算 ($|\cdot|$ 表示集合元素个数) 和 $n-2$ 次加法运算. 因 $\left|\overline{P}\right| + |p^j| < n-1$, 因此 HCSAP 的 max 运算次数少于 AP.

表 3 实验数据集

Table 3 Dataset used in experiment

Item	Number of instance	Dimension	Class	Preference
Optdigit	1 797	64	10	$1 \times Mid$
Iris	150	4	3	$3 \times Mid$
Ionosphere	351	34	2	$10 \times Mid$
Letter recognition {I, J, L}	2 241	16	3	$1 \times Mid$
Pendigits	3 498	16	10	$1 \times Mid$
glass	214	9	6	$5 \times Mid$
wine	178	13	3	$5 \times Mid$
wdbc	768	8	2	$50 \times Mid$

不过 HCSAP 增加了数据在同类集中的检索操作. 同理在当 $i \neq j$ 时也存在相同的情况. 由于检索运算时间一般少于加法运算, 且 max 运算复杂度要大于加法运算, 因此相比于 AP 而言, HCSAP 最多只是增加了检索运算. AP 的时间复杂度为 $O(n^2)$, HCSAP 时间复杂度不超过 $(n\sum_v |p_v|)/2 + n^2$. 因此 HCSAP 聚类时间只是略多于 AP, 实验结果也表明了这一点. SAP 和 SSAP 相比于 AP 聚类算法时间也较长, 主要是在相似度矩阵调整上进行了一些额外的操作, 因此运算时间要多于 AP. 此外由于 SSAP 增加了虚点, 运算量也有所增加, 因此运算时

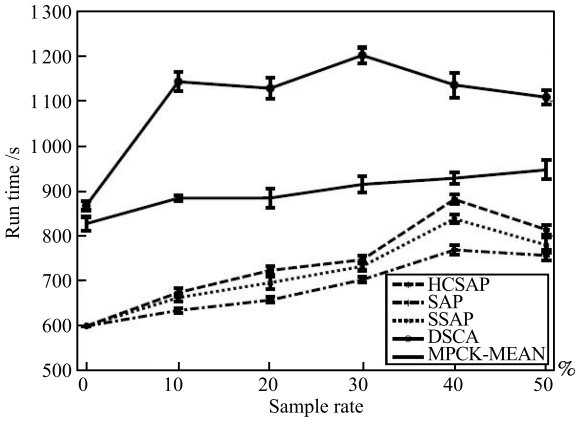


图 6 Optdigit 数据集的运行时间

Fig. 6 The run time of the algorithms on Optdigit dataset

表 4 Optdigit 数据集上的聚类结果对比

Table 4 Performance comparison on Optdigit dataset

Sample rate (%)	Item	F-measure (%)					Pure (%)				
		HCSAP	SAP	SSAP	MPCK-MEAN	DSCA	HCSAP	SAP	SSAP	MPCK-MEAN	DSCA
0	Mean	**22.35**	**22.35**	**22.35**	19.14	20.61	**12.57**	**12.57**	**12.57**	10.68	11.46
	std	(0)	(0)	(0)	(1.25)	(3.87)	(0)	(0)	(0)	(1.6)	(0.9)
	p-value	–	–	–	4.3E−3	4.9E−2	–	–	–	4.1E−2 (+)	4.8E−2
10	Mean	**31.86**	30.03	29.30	27.98	30.41	**18.97**	17.75	15.34	14.68	16.35
	std	(4.69)	(1.30)	(2.47)	(1.50)	(4.94)	(5.02)	(5.92)	(6.1)	(5.7)	(2.2)
	p-value	–	3.1E−1	3.3E−1	2.6E−1	5.1E−1	–	2.7E−1	6.1E−1	1.4E−1	9.9E−2
20	Mean	42.57	43.15	**44.63**	40.55	41.55	27.10	27.71	**27.96**	27.30	24.63
	std	(7.4)	(8.01)	(6.91)	(7.65)	(9.52)	(9.00)	(3.36)	(4.86)	(7.96)	(8.14)
	p-value	–	2.2E−1	6.6E−1	1.0E−1	2.9E−1	–	5.7E−1	6.7E−1	4.3E−1	7.4E−2)
30	Mean	**54.41**	52.5	51.23	49.87	52.44	**36.61**	35.78	34.79	30.76	31.87
	std	(9.02)	(2.01)	(4.31)	(2.44)	(3.75)	(2.67)	(0.79)	(6.6)	(8.2)	(7.8)
	p-value	–	2.6E−1	1.7E−1	4.6E−2 (+)	5.7E−1	–	4.3E−1	8.7E−2	4.4E−2 (+)	2.5E−2 (+)
40	Mean	**62.67**	61.35	61.22	59.57	61.24	**45.85**	44.46	45.20	41.85	40.27
	std	(2.39)	(1.71)	(1.6)	(4.21)	(8.34)	(0.86)	(3.12)	(4.20)	(7.14)	(9.48)
	p-value	–	4.0E−1	4.3E−1	2.9E−1	1.6E−1	–	1.7E−1	2.7E−1	7.7E−2	2.1E−2 (+)
50	Mean	**71.75**	68.84	68.8	69.54	69.33	**56.09**	52.69	51.27	48.62	50.11
	std	(9.58)	(6.32)	(9.96)	(8.47)	(10.20)	(2.45)	(4.79)	(2.70)	(5.47)	(5.50)
	p-value	–	2.4E−1	2.4E−11	8.1E−2	9.8E−2	–	1.4E−2 (+)	1.1E−1	2.5E−3 (+)	5.8E−2

间要多于 SAP 算法. 在空间复杂度方面, 相比于 AP 而言 HCSAP 增加了同类约束集的存储开销, 但这个存储开销要少于 SAP 和 SSAP 算法中的 must-link 成对约束的存储开销.

其他数据集的聚类结果如表 5～11 所示.

在设定的参数下, AP 聚类效果基本上都优于 K-means 聚类方法. 在 Ionosphere 数据集上, 基于距离的 SAP 和 SSAP 方法在抽取率为 10%, 20%, 40% 时聚类结果均优于 HCSAP、MPCK-MEANS 和 DSCA. 在 Iris 数据集和 Letter-recognition {I, J, L} 数据集中, 当同类集规模稍大时, HCSAP 聚类效果总体上要优于所对比的其他算法. 在 Pendigits、glass、wine、wdbc 这四个数据集中 HCSAP 在 F-measure 和 Pure 均有较大优势.

诚然在聚类结果的 F-measure 和 Pure 参数的 t 检验中, 由于 HCSAP 聚类结果和对比算法聚类结果的评价参数分布区间存在重合, 因此部分 p-value 值均高于 5%. 但综合实验结果来看, HCSAP 在保证聚类结果与同类先验信息一致的同时, 也能获得较好的聚类效果.

4　结论与展望

综合人工数据集和标准数据集聚类实验, HCSAP 作为一种基于约束的半监督聚类方法, 具有聚类结果遵循同类先验知识的特点; 在同类集规模较大时, 可以提升聚类性能; HCSAP 不改变相似性度量矩阵, 保留了近邻反射传播算法可用于相似性度量不对称或不满足三角不等性的场合的优点; 在时间复杂度上, HCSAP 方法与 AP 聚类相比增加了同类集中的数据的检索操作, 但检索操作时间与同类集中数据点的个数为线性关系, 运算时间只是略有增加; 由于同类集相比于 must-link 成对约束集合而言, 不存在数据重复存储的情况, 因此 HCSAP 在空间复杂度上要优于基于 must-link 成对约束的方法.

HCSAP 方法只适用于同类约束, 而不像 SAP 和 SSAP 方法那样既适用于同类约束也适用于非同类约束. 拓展 HCSAP 应用范围的一种途径是采用混合策略, 即对非同类约束采用与 SAP 相同的方式来处理, 将非同类约束 (即 cannot-link 约束) 点之间的相似度设为 $-\infty$, 再采用 HCSAP 进行聚类. 但该处理方法虽然能够避免 cannot-link 成对约束点选择对方为类中心点, 但不能排除它们选择相同的数据点作为各自的类中心点. 文献 [10−11] 指出在聚类时实现 cannot-link 成对约束是一个 NP-complete 问题, 因此 cannot-link 成对约束在实际应用时受到很大限制, 难以给出一般性的方法. 同样, 非同类约束也是一个 NP-complete 问题, 如何在近邻反射传播聚类方法中引入非同类约束进而提出一般性的方法是本文今后努力的方向.

表 5　Iris 数据集的聚类结果对比

Table 5　Performance comparison on Iris dataset

Sample rate	Item	F-measure (%)					Pure (%)				
(%)		HCSAP	SAP	SSAP	MPCK-MEAN	DSCA	HCSAP	SAP	SSAP	MPCK-MEAN	DSCA
0	Mean	**61.87**	**61.87**	**61.87**	56.27	55.38	**55.33**	**55.33**	**55.33**	52.61	53.70
	std	(0)	(0)	(0)	(3.1)	(3.6)	(0)	(0)	(0)	(2.30)	(4.81)
	p-value	–	–	–	1.4E−2	2.2E−2	–	–	–	1.6E−1	4.8E−1
10	Mean	**73.36**	72.93	72.55	71.84	69.57	**75.60**	56.53	55.22	60.77	58.45
	std	(2.69)	(0.93)	(1.21)	(6.40)	(3.72)	(5.3)	(0.87)	(4.11)	(2.57)	(12.72)
	p-value	–	1.5E−4 (+)	1.3E−4 (+)	5.3E−4 (+)	2.2E−4 (+)	–	1.3E−3 (+)	1.4E−3 (+)	5.6E−3 (+)	4.2E−2 (+)
20	Mean	79.04	**79.78**	76.99	**81.21**	74.31	**75.33**	69.11	66.22	64.21	70.40
	std	(3.22)	(8.02)	(5.90)	(8.3)	(6.61)	(7.33)	(12.10)	(18.9)	(14.1)	(11.23)
	p-value	–	2.8E−1	1.4E−1	8.2E−1	5.4E−3	–	2.7E−1	2.3E−1	5.4E−1	9.1E−1
30	Mean	**88.46**	80.33	81.24	80.74	75.33	**81.33**	72.22	71.25	72.11	69.44
	std	(1.75)	(9.15)	(9.47)	(4.4)	(10.1)	(2.91)	(12.15)	(11.41)	(13.15)	(10.9)
	p-value	–	2.1E−1	3.4E−1	1.1E−2	1.2E−2	–	2.3E−1	2.0E−1	2.8E−1	1.1E−1
40	Mean	**94.72**	89.62	88.25	85.74	84.27	**92.93**	84.40	84.00	81.23	83.78
	std	(3.64)	(3.79)	(4.8)	(5.9)	(5.6)	(5.77)	(6.95)	(8.3)	(7.9)	(9.1)
	p-value	–	1.5E−1	4.6E−2 (+)	4.9E−1	5.9E−1	–	1.7E−1	1.1E−1	1.4E−1	2.4E−1
50	Mean	**94.43**	92.38	93.50	90.22	89.01	**94.44**	88.22	87.20	86.33	90.01
	std	(0.39)	(1.39)	(2.0)	(7.1)	(5.1)	(0.03)	(5.00)	(6.89)	(6.12)	(10.88)
	p-value	–	1.0E−1	1.7E−1	5.5E−1	6.4E−1	–	1.5E−1	1.2E−1	2.7E−1	1.4E−1

表 6 Ionosphere 数据集上的聚类结果对比

Table 6 Performance comparison on Ionosphere dataset

Sample rate (%)	Item	F-measure (%)					Pure (%)				
		HCSAP	SAP	SSAP	MPCK-MEAN	DSCA	HCSAP	SAP	SSAP	MPCK-MEAN	DSCA
0	Mean	**51.96**	**51.96**	**51.96**	45.89	46.31	**40.35**	**40.35**	**40.35**	43.48	41.25
	std	(0)	(0)	(0)	(5.2)	(3.4)	(0)	(0)	(0)	(2.50)	(1.31)
	p-value	–	–	–	7.3E−1	6.9E−1	–	4.4E−1	5.2E−1	1.3E−1	2.3E−1
10	Mean	55.24	56.74	57.30	**58.21**	54.21	58.21	**59.38**	57.66	55.24	58.00
	std	(2.45)	(7.87)	(1.2)	(7.2)	(3.6)	(7.32)	(2.78)	(7.27)	(6.17)	(5.48)
	p-value	–	3.5E−2 (−)	4.8E−2 (−)	5.4E−1	7.9E−1	–	8.2E−1	6.3E−1	5.2E−1	4.5E−1
20	Mean	56.45	**61.13**	60.34	54.29	55.12	63.41	**64.12**	64.30	62.34	61.25
	std	(1.97)	(2.36)	(1.90)	(4.87)	(3.57)	(5.79)	(3.18)	(5.54)	(7.53)	(4.74)
	p-value	–	5.40E−3 (−)	3.47E−2 (−)	4.4E−1	5.1E−1	–	2.8E−1	3.5E−1	6.4E−1	4.1E−1
30	Mean	**66.61**	61.16	65.14	57.02	59.54	63.74	**67.20**	61.42	61.28	60.11
	std	(3.25)	(5.47)	(3.32)	(5.42)	(4.14)	(5.20)	(1.44)	(8.54)	(3.15)	(7.21)
	p-value	–	2.8E−1	4.2E−1	3.3E−1	2.4E−1	–	1.9E−1	2.9E−1	1.6E−1	1.7E−1
40	Mean	67.61	**71.68**	68.50	64.71	69.87	62.66	**64.04**	57.99	60.34	61.23
	std	(4.23)	(3.03)	(1.20)	(4.56)	(1.57)	(8.17)	(7.24)	(4.8)	(1.36)	(5.4)
	p-value	–	2.4E−2 (−)	3.7E−2 (−)	1.7E−1	5.6E−1	–	2.5E−1	3.6E−1	2.2E−1	1.4E−1
50	Mean	**84.16**	83.17	80.26	80.66	84.78	72.53	71.33	**72.55**	70.21	69.88
	std	(6.12)	(4.87)	(2.80)	(2.69)	(4.31)	(5.60)	(7.53)	(5.33)	(2.42)	(4.69)
	p-value	–	3.6E−1	3.4E−1	4.7E−1	2.1E−1	–	1.9E−1	3.9E−1	2.2E−1	1.0E−1

表 7 Letter-recognition I, J, L 上的聚类结果对比

Table 7 Performance comparison on Letter-recognition dataset

Sample rate (%)	Item	F-measure (%)					Pure (%)				
		HCSAP	SAP	SSAP	MPCK-MEAN	DSCA	HCSAP	SAP	SSAP	MPCK-MEAN	DSCA
0	Mean	**49.87**	**49.87**	**49.87**	41.30	42.33	**33.33**	**33.33**	**33.33**	31.54	30.36
	std	(0)	(0)	(0)	(1.9)	(4.1)	(0)	(0)	(0)	(2.5)	(3.1)
	p-value	–	–	–	5.7E−2	6.9E−2	–	–	–	1.7E−1	1.0E−1
10	Mean	**54.42**	55.01	54.36	**55.11**	57.21	39.00	38.10	**40.23**	39.98	40.05
	std	(8.90)	(2.63)	(5.7)	(6.4)	(3.89)	(8.08)	(2.78)	(1.4)	(2.0)	(3.7)
	p-value	–	6.2E−1	2.1E−1	4.8E−1	1.2E−1	–	2.3E−1	5.6E−1	7.4E−1	6.1E−1
20	Mean	**59.47**	52.01	51.69	57.20	55.31	**42.66**	35.33	34.88	37.90	35.87
	std	(1.92)	(5.73)	(4.8)	(6.8)	(5.0)	(9.57)	(7.26)	(5.8)	(3.2)	(2.8)
	p-value	–	4.2E−2 (+)	3.2E−2 (+)	1.1E−1	3.1E−1	–	4.9E−2	2.3E−2	4.7E−2	2.6E−2
30	Mean	**67.90**	65.36	66.30	64.87	60.45	**52.66**	50.00	49.68	50.33	51.74
	std	(0.41)	(6.44)	(3.2)	(5.6)	(4.9)	(1.84)	(1.54)	(1.74)	(2.53)	(3.40)
	p-value	–	2.6E−1	3.0E−1	1.7E−1	2.6E−1	–	2.3E−1	1.2E−1	2.9E−1	5.4E−1
40	Mean	**77.05**	72.97	73.33	71.4	70.24	**64.66**	60.00	59.40	58.67	51.77
	std	(2.00)	(4.43)	(1.5)	(2.6)	(2.4)	(8.17)	(4.26)	(7.26)	(8.11)	(14.25)
	p-value	–	3.3E−1	4.2E−1	1.1E−1	1.5E−1	–	3.4E−1	2.9E−1	4.1E−1	5.3E−1
50	Mean	**84.16**	83.17	84.12	83.00	81.47	**73.33**	71.33	70.11	68.25	67.49
	std	(4.55)	(7.41)	(5.4)	(4.9)	(4.4)	(2.62)	(7.53)	(5.1)	(7.6)	(5.3)
	p-value	–	4.9E−1	5.0E−1	2.3E−1	2.3E−1	–	3.3E−1	2.8E−1	1.0E−1	4.4E−2 (+)

表 8 Pendigits 数据集的聚类结果对比

Table 8 Performance comparison on Pendigits dataset

Sample rate (%)	Item	F-measure (%)					Pure (%)				
		HCSAP	SAP	SSAP	MPCK-MEAN	DSCA	HCSAP	SAP	SSAP	MPCK-MEAN	DSCA
0	Mean	**19.23**	**19.23**	**19.23**	17.21	16.32	**11.20**	**11.20**	**11.20**	9.79	9.64
	std	(0)	(0)	(0)	(1.42)	(2.51)	(0)	(0)	(0)	(0.25)	(0.36)
	p-value	–	–	–	9.4E−3 (+)	2.8E−3 (+)	–	–	–	3.9E−2 (+)	2.8E−2 (+)
10	Mean	**27.40**	23.17	24.65	24.68	21.97	**16.72**	13.75	12.99	13.58	11.95
	std	(1.38)	(8.62)	(7.64)	(9.17)	(4.11)	(5.03)	(3.15)	(2.87)	(2.77)	(1.44)
	p-value	–	2.3E−1	5.6E−2	2.8E−1	5.8E−2	–	4.1E−1	9.6E−2	6.7E−1	6.1E−1
20	Mean	**38.66**	35.19	34.67	33.74	34.21	**36.14**	25.58	24.71	21.95	27.64
	std	(0.88)	(3.67)	(4.25)	(5.50)	(6.67)	(3.13)	(8.79)	(3.34)	(5.61)	(4.13)
	p-value	–	5.6E−1	2.4E−1	2.7E−1	3.1E−1	–	8.3E−3	2.3E−3	9.8E−4	1.9E−3
30	Mean	**60.54**	57.56	55.82	51.64	56.83	**46.19**	43.08	40.27	39.87	41.56
	std	(9.90)	(0.26)	(1.13)	(4.21)	(6.57)	(0.54)	(2.86)	(3.41)	(4.12)	(3.49)
	p-value	–	8.3E−1	5.3E−1	5.6E−2	6.6E−1	–	5.8E−1	4.1E−1	8.6E−2	2.9E−1
40	Mean	**68.49**	62.12	63.77	60.16	62.57	**55.46**	48.51	44.21	39.48	41.67
	std	(1.28)	(5.29)	(6.84)	(3.46)	(4.57)	(6.09)	(1.93)	(1.53)	(4.85)	(3.34)
	p-value	–	5.6E−1	6.8E−1	4.9E−1	1.4E−1	–	4.6E−2 (+)	5.3E−3 (+)	2.1E−4 (+)	9.4E−3 (+)
50	Mean	**75.75**	66.30	67.38	67.22	64.14	**65.23**	53.60	54.69	55.21	53.96
	std	(4.58)	(8.38)	(7.52)	(7.31)	(5.63)	(2.58)	(6.82)	(7.39)	(4.61)	(9.42)
	p-value	–	6.1E−2	7.6E−1	4.9E−1	2.0E−1	–	3.8E−2 (+)	4.2E−2 (+)	1.6E−2 (+)	6.4E−3 (+)

表 9 glass 数据集的聚类结果对比

Table 9 Performance comparison on glass dataset

Sample rate (%)	Item	F-measure (%)					Pure (%)				
		HCSAP	SAP	SSAP	MPCK-MEAN	DSCA	HCSAP	SAP	SSAP	MPCK-MEAN	DSCA
0	Mean	**31.02**	**31.02**	**31.02**	28.66	27.14	**31.66**	**31.66**	**31.66**	28.51	30.69
	std	(0)	(0)	(0)	(2.80)	(3.74)	(0)	(0)	(0)	(1.90)	(2.45)
	p-value	–	–	–	2.63E−1	1.77E−1	–	–	–	2.1E−1	2.6E−1
10	Mean	**37.24**	35.85	35.62	31.89	33.57	**38.00**	**38.00**	37.15	35.46	34.76
	std	(8.90)	(3.17)	(4.66)	(5.78)	(9.51)	(8.08)	(2.78)	(3.64)	(5.21)	(4.23)
	p-value	–	6.0E−1	3.4E−1	1.9E−1	2.4E−1	–	8.4E−1	5.6E−1	3.6E−1	3.1E−1
20	Mean	**40.98**	37.80	35.70	36.44	37.15	**42.66**	35.33	34.36	31.93	32.19
	std	(0.06)	(0.02)	(0.08)	(1.62)	(3.48)	(9.57)	(7.26)	(6.19)	(8.42)	(2.96)
	p-value	–	1.1E−1	6.4E−2	8.3E−2	4.8E−1	–	5.4E−2	4.9E−2 (+)	2.5E−2 (+)	3.4E−2 (+)
30	Mean	**46.05**	43.32	44.22	40.35	45.11	**70.87**	54.52	55.21	52.94	57.14
	std	(0.15)	(3.21)	(3.90)	(5.44)	(7.16)	(1.50)	(6.36)	(4.83)	(8.91)	(7.88)
	p-value	–	2.9E−1	5.2E−1	9.7E−2	4.5E−1	–	4.6E−2 (+)	2.0E−2 (+)	9.4E−3 (+)	5.6E−2
40	Mean	**53.23**	47.71	46.25	42.18	47.84	**80.06**	61.06	64.37	59.81	60.56
	std	(1.46)	(5.89)	(4.77)	(4.65)	(7.21)	(5.00)	(7.27)	(8.36)	(7.24)	(4.98)
	p-value	–	1.8E−1	1.3E−1	7.6E−2	5.5E−1	–	5.7E−2	1.7E−1	8.7E−3 (+)	7.68E−2
50	Mean	**54.67**	50.70	51.28	49.57	51.14	**79.63**	64.02	61.44	62.37	59.87
	std	(7.43)	(4.98)	(6.40)	(7.15)	(8.44)	(8.83)	(5.42)	(6.48)	(7.21)	(9.18)
	p-value	–	1.6E−1	3.6E−1	7.4E−2	6.1E−1	–	5.0E−2 (+)	4.4E−2 (+)	4.5E−2 (+)	6.9E−3 (+)

表 10 wine 数据集的聚类结果对比

Table 10 Performance comparison on wine dataset

Sample rate (%)	Item	F-measure (%)					Pure (%)				
		HCSAP	SAP	SSAP	MPCK-MEAN	DSCA	HCSAP	SAP	SSAP	MPCK-MEAN	DSCA
0	Mean	**60.21**	**60.21**	**60.21**	54.39	57.82	**70.54**	**70.54**	**70.54**	64.87	61.49
	std	(0)	(0)	(0)	(5.67)	(4.10)	(0)	(0)	(0)	(5.10)	(6.54)
	p-value	–	–	–	2.9E−1	4.6E−1	–	–	–	1.3E−1	9.3E−2
10	Mean	**79.16**	68.98	69.33	65.47	67.26	**84.94**	72.36	73.66	75.19	70.58
	std	(5.450)	(3.82)	(5.44)	(8.21)	(4.94)	(8.35)	(5.26)	(5.87)	(4.24)	(6.29)
	p-value	–	6.3E−3 (+)	9.2E−4 (+)	6.1E−3 (+)	8.1E−2 (+)	–	4.7E−2 (+)	4.9E−2 (+)	6.7E−2	8.1E−3 (+)
20	Mean	**81.36**	71.82	69.88	68.15	60.47	**84.83**	73.88	71.20	69.64	65.88
	std	(3.65)	(5.49)	(5.40)	(8.14)	(6.47)	(8.22)	(4.55)	(6.88)	(7.52)	(10.37)
	p-value	–	7.4E−2	4.8E−2	1.3E−1	9.1E−2	–	1.5E−1	8.3E−2	7.2E−2	4.3E−2 (+)
30	Mean	**84.78**	83.27	80.31	81.55	82.41	**92.06**	89.40	85.94	90.31	88.49
	std	(2.93)	(5.21)	(6.40)	(3.70)	(7.52)	(1.68)	(6.34)	(5.80)	(1.23)	(3.54)
	p-value	–	1.8E−1	1.1E−1	3.4E−1	4.4E−1	–	1.7E−1	8.3E−2	4.6E−1	9.7E−2
40	Mean	**90.22**	84.24	80.64	81.47	72.63	**95.06**	89.66	88.33	90.27	87.92
	std	(2.81)	(3.51)	(4.36)	(1.42)	(0.99)	(1.51)	(5.82)	(6.42)	(8.66)	(9.11)
	p-value	–	4.0E−2 (+)	2.2E−3 (+)	3.6E−2 (+)	9.4E−3 (+)	–	1.3E−1	8.4E−2	5.2E−2	4.3E−2 (+)
50	Mean	**89.76**	86.68	85.85	80.74	81.69	**88.82**	85.74	85.63	84.22	80.75
	std	(1.99)	(4.99)	(6.41)	(6.28)	(7.11)	(2.62)	(7.53)	(8.90)	(7.24)	(9.11)
	p-value	–	2.7E−1	2.0E−1	5.7E−2	3.3E−1	–	2.7E−1	1.1E−1	3.4E−1	2.1E−1

表 11 wdbc 数据集的聚类结果对比

Table 11 Performance comparison on wdbc dataset

Sample rate (%)	Item	F-measure (%)					Pure (%)				
		HCSAP	SAP	SSAP	MPCK-MEAN	DSCA	HCSAP	SAP	SSAP	MPCK-MEAN	DSCA
0	Mean	**52.31**	**52.31**	**52.31**	48.42	47.49	**55.74**	**55.74**	**55.74**	51.78	52.69
	std	(0)	(0)	(0)	(1.10)	(1.67)	(0)	(0)	(0)	(0.62)	(0.37)
	p-value	–	–	–	4.5E−2 (+)	3.1E−2 (+)	–	–	–	4.8E−2 (+)	5.7E−2
10	Mean	**66.35**	50.72	48.39	51.46	49.21	**61.39**	47.80	47.92	44.91	51.68
	std	(13.38)	(1.42)	(2.82)	(2.6)	(5.06)	(11.73)	(3.50)	(6.41)	(17.46)	(14.25)
	p-value	–	1.5E−1	9.2E−2	3.4E−1	1.1E−1	–	1.2E−1	2.4E−1	4.2E−1	2.9E−1
20	Mean	**74.27**	66.58	67.24	64.21	60.37	**72.16**	61.16	60.58	57.26	59.34
	std	(13.78)	(14.87)	(11.35)	(15.87)	(9.58)	(14.45)	(17.51)	(15.68)	(11.34)	(8.48)
	p-value	–	5.0E−1	6.2E−1	9.8E−2	4.1E−1	–	4.1E−1	5.4E−1	1.2E−1	2.6E−1
30	Mean	**85.90**	59.10	58.22	57.31	56.74	**84.23**	52.42	50.72	48.64	51.77
	std	(0. 86)	(17.4)	(20.55)	(8.27)	(4.90)	(1.26)	(4.28)	(5.60)	(4.89)	(2.77)
	p-value	–	9.0E−03 (+)	2.5E−2 (+)	5.1E−3 (+)	6.4E−3 (+)	–	7.4E−04 (+)	5.6E−5 (+)	4.9E−6 (+)	1.8E−7 (+)
40	Mean	**88.09**	71.83	71.27	69.58	64.96	**87.39**	70.61	71.33	64.99	68.41
	std	(1.35)	(8.86)	(7.89)	(10.54)	(9.73)	(3.0)	(8.14)	(6.64)	(10.37)	(12.85)
	p-value	–	4.70E−2 (+)	2.8E−2 (+)	4.2E−3 (+)	7.7E−3 (+)	–	5.3E−2	5.1E−2	4.8E−3 (+)	2.5E−2 (+)
50	Mean	**87.53**	84.82	84.32	81.62	82.44	**88.32**	84.74	80.16	75.32	72.19
	std	(7.18)	(9.02)	(10.33)	(8.27)	(9.11)	(7.80)	(9.83)	(8.12)	(10.77)	(11.65)
	p-value	–	4.8E−2 (+)	3.4E−2 (+)	8.9E−3 (+)	7.7E−3 (+)	–	4.1E−2 (+)	2.2E−2 (+)	3.6E−2 (+)	1.8E−2 (+)

附录 A　η_{ij} 和 α_{ij} 迭代公式推导

证明.

对于信息 η_{ij}, 根据图 3 所给出的其与函数 $I_i(c_{i1},\cdots,c_{ij},\cdots,c_{iN})$ 节点、变量节点 c_{ij} 以及其他邻接节点之间信息的关系以及式 (14) 有:

$$\eta_{ij}(c_{ij}) = \max_{\{c_{ik},k\neq j\}}\left(I_i(c_{i1},\cdots,c_{ij},\cdots,c_{iN})+\sum_{t\neq j}\beta_{it}(c_{it})\right)$$

当 $c_{ij}=1$, 根据中心点唯一性约束, 分配有效时 (即函数 I_i 取值为 0), 则对 $\forall k\neq j$ 时, $c_{ik}=0$.

当 $c_{ij}=0$, 根据中心点唯一性约束, $\exists t\neq j$, s.t. $c_{it}=1$, 同样分配有效时 (即函数 I_i 取值为 0), 则对 $\forall k\neq t$ 时, $c_{ik}=0$.

$$\eta_{ij}(0) = \max_{\{c_{it}=1,c_{ik}=0,k\neq t,j\}}\left(I_i(c_{i1},\cdots,c_{ij}=0,\cdots,c_{iN})+\beta_{it}(1)+\sum_{v\neq t,j}\beta_{iv}(0)\right) = \max_{t\neq j}\left(\beta_{it}(1)+\sum_{v\neq j,t}\beta_{iv}(0)\right) = \max_{t\neq j}(\beta_{it}(1))+\sum_{v\neq j,t}\beta_{iv}(0)$$

则有:

$$\eta_{ij} = \eta_{ij}(1)-\eta_{ij}(0) = \sum_{t\neq j}\beta_{it}(0)-\max_{t\neq j}(\beta_{it}(1))+\sum_{v\neq j,t}\beta_{iv}(0) = -\max_{t\neq j}(\beta_{it}(1)-\beta_{it}(0)) = -\max_{t\neq j}(\beta_{it})$$

对于信息 α_{ij}, 根据图 4 给出的其与函数 $E_j(c_{1j},\cdots,c_{ij},\cdots,c_{Nj})$ 节点、变量节点 c_{ij} 以及其他邻接节点之间信息的关系以及式 (14) 有:

$$\alpha_{ij}(c_{ij}) = \max_{\{c_{ik},k\neq j\}}\left(E_j(c_{1j},\cdots,c_{ij},\cdots,c_{Nj})+\sum_{t\neq j}\rho_{tj}(c_{it})\right)$$

当 $i\neq j$, 且 $i\in\bar{P}$, $j\in\bar{P}$, 令 $c_{ij}=1$, 由中心点一致性约束 $E_j(c_{1j},\cdots,c_{Nj})$, 则必有 $c_{ij}=1$. 则

$$\alpha_{ij}(1) = \rho_{jj}(1)+\sum_{k\in\bar{P}\backslash i,j}\max_{c_{kj}}\rho_{kj}(c_{kj})+\sum_{p_v\in P}\max\left[\sum_{k\in p_v}\rho_{kj}(1),\sum_{k\in p_v}\rho_{kj}(0)\right]$$

$$\alpha_{ij}(0) = \max[\alpha_{ij}(c_{ij}=0,c_{jj}=0),\alpha_{ij}(c_{ij}=0,c_{jj}=1)] = \max\left[\sum_{k\neq i}\rho_{kj}(0),\rho_{jj}(1)+\sum_{k\in\bar{P}\backslash i,j}\max_{c_{kj}}\rho_{kj}(c_{kj})+\sum_{p_v\in P}\max\left[\sum_{k\in p_v}\rho_{kj}(1),\sum_{k\in p_v}\rho_{kj}(0)\right]\right]$$

$$\alpha_{ij} = \alpha_{ij}(1)-\alpha_{ij}(0) = \min\left[\rho_{jj}+\sum_{k\in\bar{P}\backslash i,j}\max[\rho_{kj},0]+\sum_{p_v\in P}\max\left[\sum_{k\in p_v}\rho_{kj},0\right],0\right]$$

当 $i\neq j$, 且 $i\in\bar{P}$, $j\in p^j$ ($p^j\subset P$, 即 j 包含于某个同类约束集, 下同)

$$\alpha_{ij}(1) = \sum_{k\in p^j}\rho_{kj}(1)+\sum_{k\in\bar{P}\backslash i}\max_{c_{kj}}\rho_{kj}(c_{kj})+\sum_{p_v\in P-p^j}\max\left[\sum_{k\in p_v}\rho_{kj}(1),\sum_{k\in p_v}\rho_{kj}(0)\right]$$

$$\alpha_{ij}(0) = \max[\alpha_{ij}(c_{ij}=0,c_{jj}=0),\alpha_{ij}(c_{ij}=0,c_{jj}=1)] = \max\left[\sum_{k\neq i}\rho_{kj}(0),\sum_{k\in p^j}\rho_{kj}(1)+\sum_{k\in\bar{P}\backslash i}\max_{c_{kj}}\rho_{kj}(c_{kj})+\sum_{p_v\in P-p^j}\max\left[\sum_{k\in p_v}\rho_{kj}(1),\sum_{k\in p_v}\rho_{kj}(0)\right]\right]$$

$$\alpha_{ij} = \alpha_{ij}(1)-\alpha_{ij}(0) = \min\left[\sum_{k\in p^j}\rho_{kj}+\sum_{k\in\bar{P}\backslash i}\max[\rho_{kj},0]+\sum_{p_v\in P-p^j}\max\left[\sum_{k\in p_v}\rho_{kj},0\right],0\right]$$

当 $i\neq j$, 且 $i\in p^i$, $j\in\bar{P}$

$$\alpha_{ij}(1) = \sum_{k\in p^i\backslash i}\rho_{kj}(1)+\rho_{jj}(1)+\sum_{k\in\bar{P}\backslash j}\max_{c_{kj}}\rho_{kj}(c_{kj})+\sum_{p_v\in P-p^i}\max\left[\sum_{k\in p_v}\rho_{kj}(1),\sum_{k\in p_v}\rho_{kj}(0)\right]$$

$$\alpha_{ij}(0) = \max[\alpha_{ij}(c_{ij}=0,c_{jj}=0),\alpha_{ij}(c_{ij}=0,c_{jj}=1)] = \max\left[\sum_{k\neq i}\rho_{kj}(0),\rho_{jj}(1)+\sum_{k\in p^i\backslash i}\rho_{kj}(0)+\sum_{k\in\bar{P}\backslash j}\max_{c_{kj}}\rho_{kj}(c_{kj})+\sum_{p_v\in P-p^i}\max\left[\sum_{k\in p_v}\rho_{kj}(1),\sum_{k\in p_v}\rho_{kj}(0)\right]\right]$$

$$\alpha_{ij} = \alpha_{ij}(1) - \alpha_{ij}(0) = \min\left[\sum_{k\in p^i\setminus i}\rho_{kj} + \rho_{jj} + \sum_{k\in\bar{P}\setminus j}\max\left[\rho_{kj},0\right] + \sum_{p_v\in P-p^i}\max\left[\sum_{k\in p_v}\rho_{kj},0\right], \sum_{k\in p^i\setminus i}\rho_{kj}\right]$$

当 $i\neq j$, 且 $i\in p^i$, $p^i = p^j$

$$\alpha_{ij}(1) = \sum_{k\in p^i\setminus i}\rho_{kj}(1) + \sum_{k\in\bar{P}}\max_{c_{kj}}\rho_{kj}(c_{kj}) + \sum_{p_v\in P-p^i}\max\left[\sum_{k\in p_v}\rho_{kj}(1), \sum_{k\in p_v}\rho_{kj}(0)\right]$$

由于 $c_{ij}=0$, i, j 同类, 由列约束, 则 c_{jj}=0.

$$\alpha_{ij}(0) = \alpha_{ij}(c_{ij}=0, c_{jj}=0) = \sum_{k\neq i}\rho_{kj}(0)$$

当 $i\neq j$, 且 $i\in p^i$, $j\in p^j$, 且 $p^i\neq p^j$, 当 $c_{ij}=1$, 由列约束, 必有 $c_{jj}=1$

$$\alpha_{ij}(1) = \sum_{k\in p^j}\rho_{kj}(1) + \sum_{k\in p^i\setminus i}\rho_{kj}(1) + \sum_{k\in\bar{P}}\max_{c_{kj}}\rho_{kj}(c_{kj}) + \sum_{p_v\in P-p^i-p^j}\max\left[\sum_{k\in p_v}\rho_{kj}(1), \sum_{k\in p_v}\rho_{kj}(0)\right]$$

由于 $c_{ij}=0$, i, j 同类, 由列约束, 则 $c_{jj}=0$.

$$\alpha_{ij}(0) = \max[\alpha_{ij}(c_{ij}=0, c_{jj}=0), \alpha_{ij}(c_{ij}=0, c_{jj}=1)] = \max\left[\sum_{k\neq i}\rho_{kj}(0), \sum_{k\in p^j}\rho_{kj}(1) + \sum_{k\in p^i\setminus i}\rho_{kj}(0) + \sum_{k\in\bar{P}}\max_{c_{kj}}\rho_{kj}(c_{kj})\right] + \sum_{p_v\in P-p^i-p^j}\max\left[\sum_{k\in p_v}\rho_{kj}(1), \sum_{k\in p_v}\rho_{kj}(0)\right]$$

$$\alpha_{ij} = \alpha_{ij}(1) - \alpha_{ij}(0) = \min\left[\sum_{k\in p^j}\rho_{kj} + \sum_{k\in p^i\setminus i}\rho_{kj} + \sum_{k\in\bar{P}}\max\left[\rho_{kj},0\right] + \sum_{p_v\in P-p^i}\max\left[\sum_{k\in p_v}\rho_{kj},0\right], 0\right]$$

当 $i=j$, 且 $j\in p^j$

$$\alpha_{ij}(1) = \sum_{k\in p^j\setminus i}\rho_{kj}(1) + \sum_{k\in\bar{P}}\max_{c_{kj}}\rho_{kj}(c_{kj}) + \sum_{p_v\in P-p^j}\max\left[\sum_{k\in p_v}\rho_{kj}(1), \sum_{k\in p_v}\rho_{kj}(0)\right]$$

$$\alpha_{ij}(0) = \sum_{k\neq j}\rho_{kj}(0)$$

$$\alpha_{ij} = \alpha_{ij}(1) - \alpha_{ij}(0) = \sum_{k\in p^j\setminus i}\rho_{kj} + \sum_{k\in\bar{P}}\max\left[\rho_{kj},0\right] + \sum_{p_v\in P-p^j}\max\left[\sum_{k\in p_v}\rho_{kj},0\right]$$

当 $i=j$, 且 $j\in\bar{P}$

$$\alpha_{ij}(1) = \sum_{k\in\bar{P}\setminus i}\max_{c_{kj}}\rho_{kj}(c_{kj}) + \sum_{p_v\in P}\max\left[\sum_{k\in p_v}\rho_{kj}(1), \sum_{k\in p_v}\rho_{kj}(0)\right]$$

$$\alpha_{ij}(0) = \sum_{k\neq i}\rho_{kj}(0)$$

$$\alpha_{ij} = \alpha_{ij}(1) - \alpha_{ij}(0) = \sum_{k\in\bar{P}\setminus i}\max\left[\rho_{kj},0\right] + \sum_{p_v\in P}\max\left[\sum_{k\in p_v}\rho_{kj},0\right]$$

由此完成 α_{ij}, η_{ij} 迭代公式的推导. □

References

1 Frey B J, Dueck D. Clustering by passing messages between data points. *Science*, 2007, **315**(5814): 972−976

2 Xu Xiao-Li, Lu Zhi-Mao, Zhang Ge-Sen, Li Chun, Zhang Qi. Color image segmentation based on improved affinity propagation clustering. *Journal of Computer-Aided Design & Computer Graphics*, 2012, **24**(4): 514−519
(许晓丽, 卢志茂, 张格森, 李纯, 张琦. 改进近邻传播聚类的彩色图像分割. 计算机辅助设计与图形学学报, 2012, **24**(4): 514−519)

3 Borile C, Labarre M, Franz S, Sola C, Refrégier G. Using affinity propagation for identifying subspecies among clonal organisms: lessons from *M. tuberculosis*. *BMC Bioinformatics*, 2011, **12**: 224

4 Chu Yue-Zhong, Xu Bo, Gao You-Tao, Tai Wei-Peng. Technique of remote sensing image target recognition based on affinity propagation and kernel matching pursuit. *Journal of Electronics and Information Technology*, 2014, **36**(12): 2923−2928
(储岳中, 徐波, 高有涛, 邰伟鹏. 基于近邻传播聚类与核匹配追踪的遥感图像目标识别方法. 电子与信息学报, 2014, **36**(12): 2923−2928)

5 Wang Kai-Jun, Zhang Jun-Ying, Li Dan, Zhang Xin-Na, Guo Tao. Adaptive affinity propagation clustering. *Acta Automatica Sinica*, 2007, **33**(12): 1242−1246
(王开军, 张军英, 李丹, 张新娜, 郭涛. 自适应仿射传播聚类. 自动化学报, 2007, **33**(12): 1242−1246)

6 Liu Jian-Wei, Liu Yuan, Luo Xiong-Lin. Semi-supervised learning methods. *Chinese Journal of Computers*, 2015, **38**(8): 1592−1617
(刘建伟, 刘媛, 罗雄麟. 半监督学习方法. 计算机学报, 2015, **38**(8): 1592−1617)

7 Bijral A S, Ratliff N, Srebro N. Semi-supervised learning with density based distances. [Online], available: http://ttic.uchicago.edu/~nati/Publications/SemiSupDBD.pdf, October 10, 2014

8 Wagstaff K, Cardie C. Clustering with instance-level constraints. In: Proceedings of the 17th International Conference on Machine Learning (ICML2000). Stanford: Morgan Kaufmann Publishers, 2000. 1103−1110

9 Xiao Yu, Yu Jian. Semi-supervised clustering based on affinity propagation algorithm. *Journal of Software*, 2008, **19**(11): 2803−2813
(肖宇, 于剑. 基于近邻传播算法的半监督聚类. 软件学报, 2008, **19**(11): 2803−2813)

10 Zhang Zhen, Wang Bin-Qiang, Yi Peng, Lan Ju-Long. Semi-supervised affinity propagation clustering algorithm based on stratified combination. *Journal of Electronics and Information Technology*, 2013, **35**(3): 645−651
(张震, 汪斌强, 伊鹏, 兰巨龙. 一种分层组合的半监督近邻传播聚类算法. 电子与信息学报, 2013, **35**(3): 645−651)

11 Zhang Jian-Peng, Chen Fu-Cai, Li Shao-Mei, Liu Li-Xiong. Data stream clustering algorithm based on density and affinity propagation techniques. *Acta Automatica Sinica*, 2014, **40**(2): 277−288
(张建朋, 陈福才, 李邵梅, 刘力雄. 基于密度与近邻传播的数据流聚类算法. 自动化学报, 2014, **40**(2): 277−288)

12 Givoni I E, Frey B J. Semi-supervised affinity propagation with instance-level constraints. In: Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS). Clearwater Beach, Florida, USA: JMLR W&CP5, 2009. 161−168

13 Zhao Xian-Jia, Wang Li-Hong. Analysis and improvement of semi-supervised clustering algorithm based on affinity propagation. *Computer Engineering and Applications*, 2010, **46**(36): 168−170
(赵宪佳, 王立宏. 近邻传播半监督聚类算法的分析与改进. 计算机工程与应用, 2010, **46**(36): 168−170)

14 Wagstaff K, Cadrie C, Rogers S, Schroedl S. Constrained K-means clustering with background knowledge. In: Proceedings of the 18th International Conference on Machine Learning (ICML2001). Williamstown: Morgan Kaufmann Publishers, 2001. 577−584

15 Yin Xue-Song, Hu En-Liang, Chen Song-Can. Discriminative semi-supervised clustering analysis with pairwise constraints. *Journal of Software*, 2008, **19**(11): 2791−2802
(尹学松, 胡恩良, 陈松灿. 基于成对约束的判别型半监督聚类分析. 软件学报, 2008, **19**(11): 2791−2802)

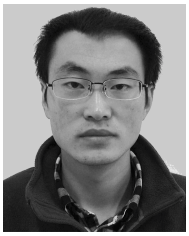
16 Kschischang F R, Frey B J, Loeliger H A. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 2001, **47**(2): 498−519

17 Weiss Y, Freeman W T. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 2001, **47**(2): 736−744

18 Givoni I E, Frey B J. A binary variable model for affinity propagation. *Neural Computation*, 2009, **21**(6): 1589−1600

徐明亮 江南大学数字媒体学院博士后, 无锡城市职业技术学院副教授. 主要研究方向为模式识别, 计算机控制. 本文通信作者. E-mail: xml1973@126.com
(**XU Ming-Liang** Postdoctor at the School of Digital Media, Jiangnan University and associate professor at Wuxi City College of Vocational Technology. His research interest covers pattern recognition and computer control. Corresponding author of this paper.)

王士同 江南大学数字媒体学院教授. 主要研究方向为人工智能, 模式识别和生物信息.
E-mail: wxwangst@yahoo.com.cn
(**WANG Shi-Tong** Professor at the School of Digital Media, Jiangnan University. His research interest covers artificial intelligence, pattern recognition, and bioinformatics.)



杭文龙 江南大学数字媒体学院博士研究生. 主要研究方向为人工智能, 模式识别. E-mail: hwl881018@163.com
(**HANG Wen-Long** Ph. D. candidate at the School of Digital Media, Jiangnan University. His research interest covers articial intelligence and pattern recognition.)