HMRF 半监督近似核 k-means 算法

贾洪杰 王良君 宋和平

(江苏大学计算机科学与通信工程学院 江苏 镇江 212013)

摘 要 信息技术的发展催生了海量数据。聚类有助于发现数据的内在联系,从中挖掘有价值的信息。在对数据进行分析时,容易获得一些关于数据的背景知识,使用这些有限的先验信息指导聚类,可以显著改善聚类的结果。基于隐马尔可夫随机场(Hidden Markov Random Fields,HMRF)的半监督聚类使用成对约束作为监督信息,虽然在很多应用场景中有较好的聚类效果,但是其时间和空间复杂度很高,无法满足大规模数据处理的需要。针对该问题,文中首先分析了 HMRF 半监督聚类与核 k-means 的数学联系,使用矩阵的迹将两者的目标函数统一起来;然后,为了降低 HMRF 半监督聚类的复杂度,提出 HMRF 半监督近似核 k-means 算法(HMRF semi-supervised Approximate Kernel K-Means,HMRF-AKKM),通过采样构造近似核矩阵,使用近似核 k-means 优化聚类的目标函数;最后,在基准数据集上将 HMRF-AKKM 算法与相关的聚类算法进行对比,分析不同算法在实验中的聚类表现。实验结果表明,在相同的聚类任务上,HMRF-AKKM 算法与原始的 HMRF 半监督聚类具有类似的聚类质量,但是 HMRF-AKKM 算法的聚类时间更短,说明 HMRF-AKKM 算法继承了 HMRF 半监督聚类与近似核 k-means 的优点。该算法一方面可以充分利用成对约束信息改善聚类质量,另一方面通过采样和矩阵近似提高了聚类效率,而且聚类质量和聚类效率可以通过调节采样比例和成对约束数量来平衡。因此,所提出的 HMRF-AKKM 算法具有良好的可扩展性,适合处理大规模非线性数据的聚类问题。

关键词 半监督聚类,HMRF 模型,近似核 k-means,矩阵的迹,成对约束

中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/jsjkx.190600159

HMRF Semi-supervised Approximate Kernel k-means Algorithm

JIA Hong-jie WANG Liang-jun SONG He-ping

(School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, Jiangsu 212013, China)

Abstract Massive data are produced with the development of information technology. Clustering can help to discover the intrinsic links of data and extract valuable information from them. In data analyzing, it is easy to get some background knowledge about data. Using these limited prior information to guide clustering can significantly improve the clustering results. The semi-supervised clustering based on Hidden Markov Random Fields (HMRF) uses pairwise constraints as the supervision information. Although it has good clustering results in many applications, its time and space complexity are very high, which cannot meet the needs of large-scale data processing. To solve this problem, this paper first analyzed the mathematical relationship between HMRF semi-supervised clustering and kernel k-means, and used matrix trace to unify the objective functions of the two clustering methods. In order to reduce the complexity of HMRF semi-supervised clustering, this paper proposed a HMRF semi-supervised approximate kernel k-means algorithm (HMRF-AKKM), which constructs an approximate kernel matrix by sampling, and used the approximate kernel k-means to optimize the clustering objective function. Finally, the HMRF-AKKM algorithm was compared with the related clustering algorithms on several benchmark datasets and the clustering performances of different algorithms were analyzed in the experiments. The experimental results show that the HMRF-AKKM algorithm has similar clustering quality to the original HMRF semi-supervised clustering on the same clustering task, but the HMRF-AKKM algorithm has shorter clustering time. This indicates that the HMRF-AKKM algorithm inherits the advantages of HMRF semi-supervised clustering and approximate kernel k-means. On the one hand, HMRF-AKKM can make full use of pairwise constraints to achieve high clustering quality. On the other hand, it improves the clustering efficiency by sampling and matrix approximation. Moreover, the clustering quality and clustering efficiency can be balanced by adjusting the sampling ratio

到稿日期:2019-04-15 返修日期:2019-06-12 本文受江苏省高校自然科学研究面上项目(18KJB520009,16KJB520008),国家自然科学基金青年基金项目(61906077,61601202),江苏省自然科学基金青年基金项目(BK20190838,BK20170558)资助。

贾洪杰(1988—),男,博士,讲师,CCF 会员,主要研究方向为机器学习、数据挖掘,E-mail:jiahj@ujs.edu.cn(通信作者);**王良君**(1982—),男,博士,讲师,主要研究方向为图像编码、压缩感知;**宋和平**(1983—),男,博士,副教授,主要研究方向为智能信息处理。

and the number of pairwise constraints. Therefore, the proposed HMRF-AKKM algorithm has good scalability and it is suitable for the clustering problems of large-scale nonlinear data.

Keywords Semi-supervised clustering, HMRF model, Approximate kernel k-means, Matrix trace, Pairwise constraints

1 引言

数据生成、收集和存储技术的发展，导致数字信息呈现爆炸式增长。根据国际数据公司(International Data Corporation，IDC)预测，到 2025 年，全球数据量将从 2018 年的 33ZB 增加到 175ZB。聚类可以对大规模数据进行有效管理，是数据分析的主要工具之一，已被广泛应用于网络搜索、社交网络分析、图像检索、医学影像分析、基因表达分析、推荐系统和市场分析等领域[1]。

很多大规模聚类算法假设数据集中的类簇是线性可分的，并根据数据之间的欧氏距离把数据对象分成不同的类别。而真实环境中采集的大规模数据集通常具有复杂的非线性结构，简单地使用欧氏距离很难描述数据对象之间的复杂关系。核 k-means 算法根据核函数定义非线性距离度量，用以计算数据的相似性，能够发现数据集中的非线性结构，而且通常比基于欧氏距离的聚类算法表现更好[2]。但是核 k-means 算法需要计算和存储核矩阵，其空间复杂度是 $O(n^2)$，在处理大规模数据时会占用大量的内存空间[3]。为了降低核 k-means 的复杂度，Chitta 等[4]提出了近似核 k-means 算法，在聚类过程中仅使用部分核矩阵，而非完整的核矩阵，通过抽样和矩阵近似来减小数据的规模和计算量。但是由于样本数有限，这种加速方法的聚类准确率一般较低，并且对初始类中心敏感。

为了提高聚类的准确率，一些文献采用半监督方式扩展核聚类，利用数据的先验知识改善聚类结果[5]。在半监督学习中，最常用的先验知识是数据点的成对约束，即 must-link 约束(表示成对的数据点属于相同的类)和 cannot-link 约束(表示成对的数据点属于不同的类)。成对约束信息在很多领域都自然存在，例如生物中的蛋白质相互作用数据库(Database of Interacting Proteins，DIP)中包含蛋白质共生的信息，在聚类中就可以视为 must-link 约束。Gan 等[6]提出一种局部齐次一致的安全半监督聚类方法，通过构建基于图的正则化项，使标记样本预测的类标签与局部邻居的类标签一致，从而降低标记样本的风险。Wang 等[7]提出一种半监督子空间聚类框架，将数据标签和亲和度结合起来，使同一子空间的数据点的亲和关系保持一致性，而不同子空间的数据点的类标签具有差异性。Yu 等[8]提出一种基于选择约束投影的双重加权半监督集成聚类方法，将随机子空间技术与约束投影方法相结合来处理高维数据集。Ren 等[9]提出一种半监督深度嵌入聚类模型，使用深度神经网络学习特征表示，并在特征学习过程中结合成对约束，使得相同类簇的数据样本在特征空间中彼此接近。Mei[10]实例化了在模糊聚类框架下用子集分区进行聚类的思想，将许多子集的划分结果视为辅助聚类的附加信息，提出一种半监督模糊聚类方法。Nguyen 等[11]提出一种基于核的距离度量学习方法，并推导出拉格朗日对偶公式来降低训练复杂度，提高 k-means 聚类的实用性。

为了对视频中的人脸图像聚类，Wu 等[12]提出一种基于隐马尔可夫随机场的半监督聚类框架，使用成对约束指导聚类过程。HMRF 框架的目标函数试图最大化 HMRF 模型中数据与约束的联合似然性，可以使用类似 k-means 的迭代优化算法求解。研究表明，基于平方欧氏距离和类大小加权惩罚项的 HMRF 半监督聚类目标函数是核 k-means 目标函数的一个特例，所以可以使用核 k-means 算法优化 HMRF 半监督聚类的目标函数。但是，核 k-means 算法需要计算和存储核矩阵，时间和空间复杂度很高。为了处理大规模的数据集，本文使用近似核 k-means 来求解 HMRF 半监督聚类问题，提出了一种 HMRF 半监督近似核 k-means 算法(HMRF-AKKM)，其通过把类中心限制在由抽样点生成的较小的子空间中来降低 HMRF 半监督聚类的复杂度。本文的主要贡献如下：

(1)分析了 HMRF 半监督聚类与核 k-means 的数学联系，使用矩阵的迹将两者的目标函数统一起来。

(2)将成对约束加入近似核 k-means 的目标函数中，并为违反约束设计了相关的惩罚项，提出 HMRF 半监督近似核 k-means 算法。

(3)利用近似核 k-means 优化 HMRF 半监督聚类的目标函数，从数据集中选择 $m(m\ll n)$个样本构造近似核矩阵进行聚类，将算法的空间复杂度从 $O(n^2)$降低到 $O(mn)$，使算法更容易扩展到大数据环境中。

(4)在多个复杂数据集上测试了所提出的 HMRF-AKKM 算法的聚类性能。实验结果表明，HMRF-AKKM 算法继承了 HMRF 半监督聚类与近似核 k-means 的优点，可以在聚类质量和聚类效率之间取得较好的平衡。

2 核 k-means 聚类

核 k-means 是经典 k-means 算法的一个非线性扩展。它把 k-means 算法中使用的欧氏距离函数 $d^2(x_a,x_b)=\|x_a-x_b\|^2$ 替换成了一个非线性核距离，定义为：

$$d_\kappa^2(x_a,x_b)=\kappa(x_a,x_a)+\kappa(x_b,x_b)-2\kappa(x_a,x_b)$$

其中，$x_a\in\mathbb{R}^d$ 和 $x_b\in\mathbb{R}^d$ 是两个数据点，$\kappa(\cdot,\cdot):\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ 表示核函数。核函数建立了从原(输入)空间到高维核空间的非线性映射，有助于识别输入空间中非线性分布的簇[13]。

令 $X=\{x_1,\cdots,x_n\}$ 表示包含 n 个点的输入数据集，k 是设定的类数，$\boldsymbol{K}\in\mathbb{R}^{n\times n}$ 是核矩阵，$K_{ij}=\kappa(x_i,x_j)$。由核函数 $\kappa(\cdot,\cdot)$诱导出的线性空间，称为再生核希尔伯特空间(Reproducing Kernel Hilbert Space，RKHS)，用 $\mathcal{H}$ 表示。将数据点划分成 k 个不相交的类即$\{V_1,\cdots,V_k\}$，每个类中数据点到类中心的距离的平方和称为聚类误差，核 k-means 的目标是寻找合理的划分，使总的聚类误差最小。因此，核 k-means 问题的目标函数可以表示为：

$$\begin{aligned}\min J(V_1,\cdots,V_k)&=\sum_{i=1}^{k}\sum_{x_j\in V_i}|\kappa(x_j,\cdot)-c_i(\cdot)|_{\mathcal{H}}^2\\&=\sum_{i=1}^{k}\sum_{j=1}^{n}U_{ij}|\kappa(x_j,\cdot)-c_i(\cdot)|_{\mathcal{H}}^2\end{aligned}\quad(1)$$

其中，$|\cdot|$表示 $\mathcal{H}$ 的泛函范数，$c_i(\cdot)\in\mathcal{H}$ 表示 V_i 的类中心，$\boldsymbol{U}\in\{0,1\}^{k\times n}$是数据点的隶属矩阵。设 $\boldsymbol{U}=(\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k)^{\mathrm{T}}$，

$\boldsymbol{U}$ 的第 i 行 $\boldsymbol{u}_i^{\mathrm{T}} \in \mathbb{R}^{1\times n}$ 是类 V_i 的指示向量，若点 x_j 在 V_i 中，$u_i(j)=1$；否则，$u_i(j)=0$。令 $n_i=\boldsymbol{u}_i^{\mathrm{T}}\mathbf{1}_n$ 表示 V_i 中数据点的个数，n_i 的倒数构成对角矩阵 $\boldsymbol{F}\in\mathbb{R}^{k\times k}$：$\boldsymbol{F}=[\mathrm{diag}(n_1,\cdots,n_k)]^{-1}$。利用 $\boldsymbol{F}$ 将 $\boldsymbol{U}$ 归一化，得到 $\overset{\wedge}{\boldsymbol{U}}\in\mathbb{R}^{k\times n}$ 和 $\tilde{\boldsymbol{U}}\in\mathbb{R}^{k\times n}$：

$$\overset{\wedge}{\boldsymbol{U}}=(\overset{\wedge}{\boldsymbol{u}}_1,\cdots,\overset{\wedge}{\boldsymbol{u}}_k)^{\mathrm{T}}=[\mathrm{diag}(n_1,\cdots,n_k)]^{-1}\boldsymbol{U}=\boldsymbol{F}\boldsymbol{U}$$

$$\tilde{\boldsymbol{U}}=(\tilde{\boldsymbol{u}}_1,\cdots,\tilde{\boldsymbol{u}}_k)^{\mathrm{T}}=[\mathrm{diag}(\sqrt{n_1},\cdots,\sqrt{n_k})]^{-1}\boldsymbol{U}=\boldsymbol{F}^{1/2}\boldsymbol{U}$$

直接寻找 U_{ij} 和 $c_i(\cdot)$ 来最小化 $J(V_1,\cdots,V_k)$ 并不容易，不过可以采取迭代的方法：先固定 $c_i(\cdot)$，选择最优的 U_{ij}（容易看出，只要把数据点和离它最近的类中心归为一类，就能保证 $J(V_1,\cdots,V_k)$ 最小）；然后固定 U_{ij}，再求最优的 $c_i(\cdot)$。用 $J(V_1,\cdots,V_k)$ 对 $c_i(\cdot)$ 求偏导，并令导数等于 0，可以推出 $J(V_1,\cdots,V_k)$ 最小时，$c_i(\cdot)$ 应该满足：

$$c_i(\cdot)=\frac{1}{n_i}\sum_{x_j\in V_i}\boldsymbol{\kappa}(x_j,\cdot)=\sum_{j=1}^{n}\hat{U}_{ij}\boldsymbol{\kappa}(x_j,\cdot),i\in[k] \tag{2}$$

式(1)中核 k-means 的目标函数可以写成矩阵迹的形式[4]：

$$J(V_1,\cdots,V_k)=\mathrm{tr}(\boldsymbol{K})-\mathrm{tr}(\tilde{\boldsymbol{U}}\boldsymbol{K}\tilde{\boldsymbol{U}}^{\mathrm{T}}) \tag{3}$$

因为核矩阵 $\boldsymbol{K}$ 是固定的，所以式(3)中的 $\mathrm{tr}(\boldsymbol{K})$ 是一个常量，则最小化式(3)等价于最大化 $\mathrm{tr}(\tilde{\boldsymbol{U}}\boldsymbol{K}\tilde{\boldsymbol{U}}^{\mathrm{T}})$。最后，核 k-means 的目标函数可以表示为：

$$\max_{\tilde{U}}\ \mathrm{tr}(\tilde{\boldsymbol{U}}\boldsymbol{K}\tilde{\boldsymbol{U}}^{\mathrm{T}}) \tag{4}$$

3　HMRF 半监督聚类

使用成对约束形式的监督信息（如属于相同或不同类的有标签的成对实例）可以显著改善无监督聚类的效果。HMRF 半监督聚类模型统一了基于约束的和基于距离的方法。该模型选择欧氏距离的平方度量类的失真程度，把广义 Potts 势（Generalized Potts Potential）作为违反约束的惩罚项，其目标函数表示如下：

$$J(V_1,\cdots,V_k)=\sum_{i=1}^{k}\sum_{x_j\in V_i}\|x_j-m_i\|^2+\sum_{\substack{x_a,x_b\in M\\ \text{s.t. } l_a\neq l_b}}p_{ab}+\sum_{\substack{x_a,x_b\in C\\ \text{s.t. } l_a=l_b}}p_{ab} \tag{5}$$

其中，V_i 表示第 i 个类，$m_i=\frac{1}{n_i}\sum_{x_j\in V_i}x_j$ 是 V_i 的类中心（n_i 是 V_i 中数据点的个数），M 是 must-link 约束的集合，C 是 cannot-link 约束的集合，p_{ab} 是违反 x_a 和 x_b 之间约束的惩罚成本，l_a 是 x_a 的类标签。式(5)的第一项是标准 k-means 目标函数，第二项是违反 must-link 约束的惩罚函数，第三项是违反 cannot-link 约束的惩罚函数。可以使用类似 k-means 的迭代重定位方法来优化式(5)的目标函数。

观察式(5)的第二项，如果属于 must-link 的两个点被划到不同的类中，则为违反 must-link 约束增加一个惩罚项。我们可以修改这个惩罚函数，如果 must-link 的数据点在相同的类中，说明满足约束条件，就给予奖励，从目标函数中减去相应的惩罚项。假如所有 must-link 约束的权值和是一个定值，就相当于给原始的目标函数加了一个常数。因此，最小化式(5)等价于最小化：

$$\sum_{i=1}^{k}\sum_{x_j\in V_i}\|x_j-m_i\|^2-\sum_{\substack{x_a,x_b\in M\\ l_a=l_b}}p_{ab}+\sum_{\substack{x_a,x_b\in C\\ l_a=l_b}}p_{ab}$$

下面考虑这种情况：如果属于 cannot-link 的两个点在同一类里，且相应的类较小，则应该给予更高的惩罚；类似地，如果属于 must-link 的两个点在较小的类中，应该给予更高的奖励。因此，我们引入类大小加权惩罚项，每个 p_{ij} 都除以相应类中数据点的总数，得到：

$$J(V_1,\cdots,V_k)=\sum_{i=1}^{k}\sum_{x_j\in V_i}\|x_j-m_i\|^2-\sum_{\substack{x_a,x_b\in M\\ l_a=l_b}}\frac{p_{ab}}{n_{l_a}}+\sum_{\substack{x_a,x_b\in M\\ l_a=l_b}}\frac{p_{ab}}{n_{l_a}} \tag{6}$$

根据 $\sum_{i=1}^{k}\sum_{x_j\in V_i}2\|x_j-m_i\|^2=\sum_{i=1}^{k}\sum_{x_a,x_b\in V_i}\frac{1}{n_i}\|x_a-x_b\|^2$ 重写式(6)，则 HMRF 半监督聚类的目标函数等价于最小化：

$$\begin{aligned}&J(V_1,\cdots,V_k)\\&=\sum_{i=1}^{k}\sum_{x_a,x_b\in V_i}\frac{\|x_a-x_b\|^2}{n_i}-\sum_{\substack{x_a,x_b\in M\\ l_a=l_b}}\frac{2p_{ab}}{n_{l_a}}+\sum_{\substack{x_a,x_b\in C\\ l_a=l_b}}\frac{2p_{ab}}{n_{l_a}}\\&=\sum_{i=1}^{k}\sum_{x_a,x_b\in V_i}\frac{\|x_a-x_b\|^2}{n_i}-\sum_{i=1}^{k}\sum_{\substack{x_a,x_b\in V_i\\ (x_a,x_b)\in M}}\frac{2p_{ab}}{n_i}+\\&\quad\sum_{i=1}^{k}\sum_{\substack{x_a,x_b\in V_i\\ (x_a,x_b)\in M}}\frac{2p_{ab}}{n_i}\end{aligned} \tag{7}$$

为了简化式(7)，使用隶属矩阵 $\boldsymbol{U}\in\mathbb{R}^{k\times n}$ 来描述类的划分。令 $\boldsymbol{E}$ 表示成对数据点的平方欧氏距离的矩阵，即 $E_{ab}=\|x_a-x_b\|^2$。设 $\boldsymbol{P}$ 是约束惩罚矩阵，对于 must-link，$P_{ab}=p_{ab}$；对于 cannot-link，$P_{ab}=-p_{ab}$；否则，$P_{ab}=0$。然后，将式(7)写成下面的形式：

$$J(V_1,\cdots,V_k)=\sum_{i=1}^{k}\frac{\boldsymbol{u}_i^{\mathrm{T}}(\boldsymbol{E}-2\boldsymbol{P})\boldsymbol{u}_i}{\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{u}_i} \tag{8}$$

其中，$\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{u}_i=n_i$ 是 V_i 中数据点的个数，$\boldsymbol{u}_i^{\mathrm{T}}(\boldsymbol{E}-2\boldsymbol{P})\boldsymbol{u}_i$ 是 V_i 中所有成对数据点 (x_a,x_b) 对应的 $E_{ab}-2P_{ab}$ 之和。

为了把式(8)写成矩阵迹的形式，引入归一化的隶属矩阵 $\tilde{\boldsymbol{U}}=\boldsymbol{F}^{1/2}\boldsymbol{U}=(\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}})^{-1/2}\boldsymbol{U}$，$\tilde{\boldsymbol{U}}$ 的第 i 行 $\tilde{\boldsymbol{u}}_i^{\mathrm{T}}=\boldsymbol{u}_i^{\mathrm{T}}/(\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{u}_i)^{1/2}$，而且 $\tilde{\boldsymbol{U}}$ 是一个正交矩阵，即 $\tilde{\boldsymbol{U}}\tilde{\boldsymbol{U}}^{\mathrm{T}}=\boldsymbol{I}_k$（其中 $\boldsymbol{I}_k\in\mathbb{R}^{k\times k}$ 是单位矩阵），则式(8)可以表示为：

$$J(V_1,\cdots,V_k)=\sum_{c=1}^{k}\tilde{\boldsymbol{u}}_c^{\mathrm{T}}(\boldsymbol{E}-2\boldsymbol{P})\tilde{\boldsymbol{u}}_c=\mathrm{tr}(\tilde{\boldsymbol{U}}(\boldsymbol{E}-2\boldsymbol{P})\tilde{\boldsymbol{U}}^{\mathrm{T}}) \tag{9}$$

对比式(4)与式(9)可知，这两个目标函数的主要区别是：HMRF 半监督聚类是迹的最小化问题，而核 k-means 是迹的最大化问题。为了把式(9)变成一个最大化问题，根据 $E_{ab}=\|x_a-x_b\|^2=x_a^{\mathrm{T}}x_a+x_b^{\mathrm{T}}x_b-2x_a^{\mathrm{T}}x_b$，分别定义矩阵 $\boldsymbol{S}$ 和 $\tilde{\boldsymbol{S}}$，其中 $\boldsymbol{S}_{ab}=x_a^{\mathrm{T}}x_b$，$\tilde{\boldsymbol{S}}_{ab}=\boldsymbol{S}_{aa}+\boldsymbol{S}_{bb}$，则 $\boldsymbol{E}=\tilde{\boldsymbol{S}}-2\boldsymbol{S}$，将其代入式(9)中得：

$$\begin{aligned}J(V_1,\cdots,V_k)&=\mathrm{tr}(\tilde{\boldsymbol{U}}(\boldsymbol{E}-2\boldsymbol{P})\tilde{\boldsymbol{U}}^{\mathrm{T}})\\&=\mathrm{tr}(\tilde{\boldsymbol{U}}(\tilde{\boldsymbol{S}}-2\boldsymbol{S}-2\boldsymbol{P})\tilde{\boldsymbol{U}}^{\mathrm{T}})\\&=\mathrm{tr}(\tilde{\boldsymbol{U}}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{U}}^{\mathrm{T}})-2\mathrm{tr}(\tilde{\boldsymbol{U}}(\boldsymbol{S}+\boldsymbol{P})\tilde{\boldsymbol{U}}^{\mathrm{T}})\\&=2\mathrm{tr}(\boldsymbol{S})-2\mathrm{tr}(\tilde{\boldsymbol{U}}(\boldsymbol{S}+\boldsymbol{P})\tilde{\boldsymbol{U}}^{\mathrm{T}})\end{aligned}$$

注意到 $\mathrm{tr}(\boldsymbol{S})$ 是一个常数，在优化过程中可以忽略，于是 HMRF 半监督聚类的目标函数可以表示为：

$$\max_{\tilde{U}}\ \mathrm{tr}(\tilde{\boldsymbol{U}}(\boldsymbol{S}+\boldsymbol{P})\tilde{\boldsymbol{U}}^{\mathrm{T}}) \tag{10}$$

如果令核矩阵 $\boldsymbol{K}=\boldsymbol{S}+\boldsymbol{P}$，则式(10)可表示成 $\mathrm{tr}(\tilde{\boldsymbol{U}}\boldsymbol{K}\tilde{\boldsymbol{U}}^{\mathrm{T}})$

的最大化，这在数学上等价于核 k-means 的目标函数，因此可以用核 k-means 算法来解决 HMRF 半监督聚类问题。

4 HMRF 半监督近似核 k-means 聚类

利用核 k-means 优化 HMRF 半监督聚类的目标函数，需要构造 $n\times n$ 的核矩阵 $\boldsymbol{K}$，但是存储核矩阵会消耗大量的内存资源，空间复杂度是 $O(n^2)$，较高的复杂度使得其不适合处理大规模数据，这也是核 k-means 算法的一个主要缺点。为了解决这一难题，Chitta 等提出了近似核 k-means 算法[4]，通过把类中心限制在由抽样点生成的较小的子空间中，构造近似核矩阵，以降低算法的空间需求。

给定数据集 $X=\{x_i\}_{i=1}^{n}$，从中随机选出 m 个样本点，组成样本集合 $\hat{X}=\{\hat{x}_i\}_{i=1}^{m}$，其中 $m\ll n$。然后根据 $\hat{X}$ 中成对样本点之间的核相似性构造矩阵 $\boldsymbol{K}_A\in\mathbb{R}^{m\times m}$，根据 X 中的数据点与 $\hat{X}$ 中的样本点之间的核相似性构造矩阵 $\boldsymbol{K}_B\in\mathbb{R}^{n\times m}$，则近似核 k-means 的目标函数可以用矩阵的迹表示为：

$$\max_{\widetilde{U}} \operatorname{tr}(\widetilde{\boldsymbol{U}}\boldsymbol{K}_B\boldsymbol{K}_A^{-1}\boldsymbol{K}_B^{\mathrm{T}}\widetilde{\boldsymbol{U}}^{\mathrm{T}}) \tag{11}$$

将式(11)与式(4)进行对比可知，在近似核 k-means 中，核矩阵 $\boldsymbol{K}\approx\boldsymbol{K}_B\boldsymbol{K}_A^{-1}\boldsymbol{K}_B^{\mathrm{T}}$，算法运行过程中只用到了矩阵 $\boldsymbol{K}_A$ 和 $\boldsymbol{K}_B$，由于 $\boldsymbol{K}_A$ 是 $\boldsymbol{K}_B$ 的一部分，实际上只需要计算 $\boldsymbol{K}_B$，这样就避免了使用整个核矩阵 $\boldsymbol{K}$。在这种情况下，设由采样集合 $\hat{X}$ 得到的类中心为 $m_i=\sum_{j=1}^{m}\alpha_{ij}\kappa(\hat{x}_j,\cdot)$，可以证明矩阵 $\boldsymbol{\alpha}=\hat{\boldsymbol{U}}\boldsymbol{K}_B\boldsymbol{K}_A^{-1}$，其中 $\hat{\boldsymbol{U}}=\boldsymbol{F}\boldsymbol{U}=(\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}})^{-1}\boldsymbol{U}$。

近似核 k-means 的空间复杂度是 $O(mn)$，在处理大规模数据时，由于 $m\ll n$，可以节省大量的内存空间，提高聚类的效率。因此，本文使用近似核 k-means 算法来求解大规模 HMRF 半监督聚类问题，提出了 HMRF 半监督近似核 k-means 算法(见算法 1)。

算法 1 HMRF 半监督近似核 k-means 算法

HMRF-Approx-Kernel-KMeans($X,m,k,\boldsymbol{P},t_{\max}$)

输入：数据集 $X=\{x_1,\cdots,x_n\}$，样本数 m (m≪n)，类数 k，约束惩罚矩阵 $\boldsymbol{P}$，最大迭代次数 $t_{\max}$

输出：聚类结果的隶属矩阵 $\boldsymbol{U}$

1. 从 X 中随机选取 m 个点，得到样本集 $\hat{X}=\{\hat{x}_1,\cdots,\hat{x}_m\}$。
2. 根据 HMRF 半监督聚类对应的核矩阵 $\mathbf{K}=\mathbf{S}+\mathbf{P}$，计算 $K_A=[\kappa(\hat{x}_i,\hat{x}_j)]_{m\times m}$ 和 $K_B=\kappa(x_i,\hat{x}_j)]_{n\times m}$。
3. 初始化隶属矩阵 $\mathbf{U}$=Constraints-Init(k,$\mathbf{P}$)(见算法 2)
4. 令迭代次数 t=0。
5. repeat
6. 令 t=t+1。
7. 为了确定每次迭代的类中心，计算矩阵 $\boldsymbol{\alpha}=\hat{\mathbf{U}}\mathbf{K}_B\mathbf{K}_A^{-1}=(\mathbf{U}\mathbf{U}^{\mathrm{T}})^{-1}\mathbf{U}\mathbf{K}_B\mathbf{K}_A^{-1}$。
8. for j=1,…,n do
9. repeat
10. 为每个点 x_j 找到最近的类中心 i^*：

$$i^*=\arg\min_{i\in[k]}[d(x_j,m_i)]=\arg\min_{i\in[k]}(\boldsymbol{\alpha}_i^{\mathrm{T}}\mathbf{K}_A\boldsymbol{\alpha}_i-2\boldsymbol{\phi}_j^{\mathrm{T}}\boldsymbol{\alpha}_i)$$

其中，$\boldsymbol{\alpha}_i^{\mathrm{T}}$ 是 $\boldsymbol{\alpha}$ 的第 i 行，$\boldsymbol{\phi}_j^{\mathrm{T}}$ 是 $\mathbf{K}_B$ 的第 j 行。

11. until Constraints-Check($x_j,V_{i^*},\mathbf{P}$)=true(见算法 3)
12. 更新 $\mathbf{U}$ 的第 j 列，令其第 i=i^* 行元素 $U_{ij}=1$，而其余元素为 0。
13. end for
14. until 隶属矩阵 $\mathbf{U}$ 不再变化或 $t>t_{\max}$。

算法 2 成对约束初始化算法

Constraints-Init($k,\boldsymbol{P}$)

输入：类数 k，约束惩罚矩阵 $\mathbf{P}$

输出：初始的隶属矩阵 $\mathbf{U}^{(0)}$

1. 计算 $\mathbf{P}$ 中 must-link 约束的传递闭包 T_m。
2. 通过 T_m 中 must-link 连接的点，得到连通分量集合 C_m。
3. 对于 C_m 中两个连通分量 $C_1,C_2\in C_m$，若 C_1 和 C_2 之间存在 cannot-link 约束，就在所有成对点(x_a,x_b)之间增加 cannot-link 约束，其中 $x_a\in C_1,x_b\in C_2$。
4. 根据 cannot-link 和 C_m，进一步确定候选连通分量集合 C_c。
5. 将 C_c 中最大的连通分量作为第 1 类 $V_1^{(0)}$，令当前类数 c=1。
6. repeat
7. 找到离当前已选的类 $\{V_i^{(0)}\}_{i=1}^{c}$ 最远的分量 C_{far}。
8. 令 $V_{c+1}^{(0)}=C_{far}$，令 c=c+1。
9. until c=k。
10. 用隶属矩阵 $\mathbf{U}^{(0)}$ 表示集合 $\{V_i^{(0)}\}_{i=1}^{k}$。

算法 3 成对约束检查算法

Constraints-Check($x_j,V_i,\boldsymbol{P}$)

输入：数据点 x_j，点 x_j 所属的类 V_i，约束惩罚矩阵 $\mathbf{P}$

输出：true 或 false

1. 根据矩阵 $\mathbf{P}$，确定 must-link 的集合 $Con_=$ 和 cannot-link 的集合 $Con_{\neq}$。
2. 对于 $Con_=$ 中的每对数据点($x_j,x_=$)$\in Con_=$，如果 $x_=\notin V_i$，则返回 false；对于 $Con_{\neq}$ 中的每对数据点($x_j,x_{\neq}$)$\in Con_{\neq}$，如果 $x_{\neq}\in V_i$，则返回 false。
3. 否则，返回 true。

在 HMRF-AKKM 算法中，隶属矩阵 $\boldsymbol{U}$ 的初始化非常重要。为了产生好的初始类中心，本文设计了算法 2，使用成对约束和未标记的数据来初始化隶属矩阵 $\boldsymbol{U}$。首先假设所有的约束都是一致的，根据 $\boldsymbol{P}$ 中编码的 must-link 和 cannot-link 推断额外的约束，得到候选连通分量的集合；然后使用 Farthest-first 遍历方法[14]，从候选连通分量中选择 k 个初始的类。为了满足给定的成对约束条件，HMRF-AKKM 在聚类过程中会不断调整数据点的类标签(详见算法 1 的第 9－11 步)。对于任意数据点 x_j，首先尝试把它分配到最近的类 $V_{c(x_j)}$ 中，然后调用算法 3，检查这样的指派是否满足 must-link 和 cannot-link 约束条件。只有当 must-link 集合中所有与 x_j 相连的点 $x_=$ 都属于类 $V_{c(x_j)}$，同时 cannot-link 集合中所有与 x_j 相连的点 $x_{\neq}$ 都不属于类 $V_{c(x_j)}$ 时，类 $V_{c(x_j)}$ 才是合法的；否则，继续测试下一个离 x_j 最近的类，直到算法 3 返回 true 为止。

5 实验与分析

5.1 数据集

为了测试 HMRF-AKKM 算法的有效性，在 5 个基准数据集上进行实验，表 1 给出了这些数据集的基本信息。

表 1　实验中使用的基准数据集

Table 1　Benchmark datasets used in experiments

数据集	数据对象数	维数	类数
Twonorm	7400	20	2
USPS	9298	256	10
Letter Recognition	20000	16	26
MNIST	70000	784	10
Skin Segmentation	245057	3	2

Twonorm[15]的每类数据点都服从方差为 1 的多元正态分布，但是两类数据有重叠。USPS 和 MNIST[16]都是手写数字灰度图片数据集。Letter Recognition[17]包含 26 个英文大写字母的不同字体的图片。Skin Segmentation[18]是从 FERET 数据库和 PAL 数据库的人脸图像中随机选取 B,G,R 值组成的，其中皮肤样本 50859 个，非皮肤样本 194198 个。

5.2　度量标准

本文使用 CRI(Constrained Rand Index)指标来评价聚类结果[19]。设 U_c 是算法聚类后得到的数据点划分，U_t 是原始数据集真实的划分，CRI 指标可以衡量 U_c 和 U_t 两种划分的相符程度。若 X 中包含 n 个数据点，CRI 假设每种划分都有 $n\times(n-1)/2$ 个成对决策。令 a 表示 U_c 和 U_t 中的 (x_i,x_j) 都分到相同类中的决策数，b 表示 (x_i,x_j) 在 U_c 和 U_t 中分别属于不同类的决策数，则 $a+b$ 即为正确的决策数。对于半监督聚类来说，由于加入了成对约束，算法能确保约束闭包中的点对是正确的，因此要去掉已知成对约束的数目 c。CRI 指标的计算公式如下：

$$CRI(U_c,U_t)=\frac{a+b-c}{n\times(n-1)/2-c} \tag{12}$$

其中，分子是正确的自由决策数，分母为总的自由决策数。CRI 值在 0 和 1 之间，1 表示最佳的聚类结果，即所有数据点都被分到正确的类中。

5.3　实验和结果

在实验中，本文将所提出的 HMRF 半监督近似核 k-means 算法(HMRF-AKKM)与 3 种相关的聚类算法进行对比，对比算法分别是基于近似加权核 k-means 的谱聚类算法(AWKK-SC)[20]、基于背景知识约束的 k-means 算法(Cop-Kmeans)[21]、基于 HMRF 模型的半监督近似谱聚类算法(HMRF-ASC)[22]。算法中使用的成对约束是由数据集中获得的样本点生成的，must-link 和 cannot-link 取决于关联的两个点属于相同的类或不同的类[23]。依据成对约束，构造约束惩罚矩阵 $\boldsymbol{P}$。为了使约束惩罚项和数据点与类中心之间的距离惩罚项大约在相同的尺度上，把 $\boldsymbol{P}$ 中每个惩罚权重设为 $p_{ij}=n/(kc)$，其中 n 是所有数据点的数目，k 是数据点的类数，c 是约束的总数[14]。实验电脑的基本配置如下：处理器 AMD Ryzen 5 1600 3.20GHz，内存 16GB，操作系统 Windows 10 64 位，编程环境 MATLAB 2015b。将每种算法在不同数据集上运行 20 次，得到平均聚类结果。

表 2 给出了各种算法在不同采样比例下的 CRI 统计值。Cop-Kmeans，HMRF-ASC 和 HMRF-AKKM 都属于半监督聚类算法，在聚类时都使用了成对约束；而 AWKK-SC 是无监督聚类算法，无须使用数据点的成对限制信息。观察表 2 中的数据，由于缺少了用户先验知识的指导，AWKK-SC 算法无法及时修正划分错误的数据点的类标签，因此在 MNIST 和 Skin Segmentation 数据集上表现不好。总体来看，随着约束中包含的样本点逐渐增多，半监督聚类算法可以更好地了解数据的内在结构，其聚类准确率也在不断提高。Cop-Kmeans 算法在 Twonorm 数据集上的 CRI 指标最高，不过 Cop-Kmeans 主要基于欧氏距离判断数据点的归属，而且容易陷入局部最优，因此在其他几个数据集上的表现不如 HMRF-ASC 和 HMRF-AKKM 算法。

表 2　算法在不同数据集上的 CRI 指标

Table 2　CRI of algorithms on different datasets

算法	采样个数	数据集				
		Twonorm	USPS	Letter Recognition	MNIST	Skin Segmentation
AWKK-SC	50	0.9522(±0.0018)	0.6324(±0.0035)	0.9268(±0.0028)	0.5178(±0.0153)	0.5011(±0.0031)
	100	0.9542(±0.0015)	0.6395(±0.0042)	0.9292(±0.0016)	0.5408(±0.0188)	0.5029(±0.0017)
	200	0.9584(±0.0006)	0.6541(±0.0031)	0.9295(±0.0020)	0.5646(±0.0299)	0.5042(±0.0026)
	400	0.9591(±0.0012)	0.6679(±0.0028)	0.9307(±0.0055)	0.6430(±0.0187)	0.5064(±0.0020)
	800	0.9594(±0.0005)	0.6837(±0.0022)	0.9344(±0.0038)	0.6727(±0.0129)	0.5098(±0.0005)
Cop-Kmeans	50	0.9526(±0.0007)	0.7619(±0.0046)	0.7837(±0.0109)	0.8771(±0.0061)	0.5043(±0.0029)
	100	0.9573(±0.0004)	0.7738(±0.0033)	0.8212(±0.0087)	0.8820(±0.0013)	0.5113(±0.0010)
	200	0.9581(±0.0014)	0.7795(±0.0037)	0.8706(±0.0069)	0.8833(±0.0059)	0.5122(±0.0031)
	400	0.9624(±0.0018)	0.7864(±0.0026)	0.9038(±0.0050)	0.8837(±0.0040)	0.5141(±0.0015)
	800	**0.9635**(±0.0015)	0.7953(±0.0031)	0.9243(±0.0086)	0.8862(±0.0051)	0.5173(±0.0016)
HMRF-ASC	50	0.9515(±0.0019)	0.7842(±0.0042)	0.9302(±0.0092)	0.8896(±0.0027)	0.5406(±0.0012)
	100	0.9544(±0.0015)	0.7916(±0.0038)	0.9349(±0.0074)	0.8924(±0.0044)	0.5427(±0.0007)
	200	0.9582(±0.0008)	0.8231(±0.0025)	0.9368(±0.0020)	0.9167(±0.0031)	0.5498(±0.0027)
	400	0.9605(±0.0010)	0.8472(±0.0021)	0.9387(±0.0019)	0.9358(±0.0023)	0.5529(±0.0006)
	800	0.9610(±0.0013)	0.8585(±0.0017)	**0.9420**(±0.0047)	**0.9587**(±0.0040)	0.5544(±0.0016)
HMRF-AKKM	50	0.9554(±0.0005)	0.8428(±0.0057)	0.9294(±0.0044)	0.8809(±0.0043)	0.5485(±0.0017)
	100	0.9564(±0.0017)	0.8476(±0.0049)	0.9334(±0.0017)	0.8858(±0.0063)	0.5486(±0.0024)
	200	0.9577(±0.0005)	0.8536(±0.0051)	0.9353(±0.0018)	0.8877(±0.0053)	0.5541(±0.0023)
	400	0.9579(±0.0008)	0.8591(±0.0064)	0.9372(±0.0016)	0.8895(±0.0014)	0.5543(±0.0019)
	800	0.9595(±0.0016)	**0.8667**(±0.0044)	0.9386(±0.0025)	0.8978(±0.0037)	**0.5631**(±0.0026)

HMRF-ASC 算法建立在半监督 Graph cut 目标函数基础上，并使用加权核 k-means 算法优化目标函数，它在

Letter Recognition 和 MNIST 数据集上表现较好。所提出的 HMRF-AKKM 算法可以较好地处理 USPS 和 Skin Segmentation 数据集，而且它在 Twonorm 和 Letter Recognition 数据集上的聚类表现与 HMRF-ASC 算法接近，说明使用近似核 k-means 算法求解 HMRF 半监督聚类问题是有效的。

表 3 对比了各个算法在不同数据集上聚类的平均时间。实验中的对比算法 AWKK-SC，Cop-Kmeans，HMRF-ASC 和 HMRF-AKKM 都属于划分式聚类算法，它们都通过反复迭代为每个数据点寻找最近的类中心，使目标函数达到最优[24]。表 3 中的数据说明，这些划分式聚类算法的聚类时间与抽样点的个数和算法迭代次数密切相关。样本规模越大，迭代次数越多，算法花费的聚类时间也越长[25]。这几个算法中，Cop-Kmeans 的运行速度最快，因为它基于 k-means 算法，不用构造核矩阵，所以能在较短的时间内得到聚类结果。AWKK-SC 使用近似加权核 k-means 来优化 Graph cut 的目标函数，需要计算原始数据点与样本点两两之间的核距离，以便通过近似核矩阵确定每个数据点的类别。HMRF-ASC 算法在 AWKK-SC 基础上引入了成对约束，在每次迭代时都会检查并修正数据点对的类属关系，使它们同时满足 must-link 和 cannot-link 约束条件。这样做虽然可以改善聚类精度，但是也增加了聚类时间，尤其是在 Letter Recognition，MNIST，Skin Segmentation 这些包含上万对象的数据集上，当约束中的样本点增加到 500～1000 时，HMRF-ASC 的效率会明显降低。与 HMRF-ASC 相比，HMRF-AKKM 算法使用近似核 k-means 而非近似加权核 k-means 来优化目标函数，不需要为每个数据点计算权重，所以可以在较短的时间内完成聚类任务，聚类效率更高。

表 3 算法在不同数据集上的聚类时间

Table 3 Clustering time of algorithms on different datasets

（单位：s）

算法	采样个数	数据集				
		Twonorm	USPS	Letter Recognition	MNIST	Skin Segmentation
AWKK-SC	50	0.152	1.425	10.025	16.891	13.916
	100	0.166	1.517	14.971	18.120	22.098
	200	0.271	2.346	18.073	21.077	44.287
	400	0.711	3.259	21.435	29.343	66.383
	800	2.230	5.034	25.762	45.258	117.881
Cop-Kmeans	50	0.293	2.168	1.657	25.908	6.339
	100	0.334	2.872	2.646	34.605	7.170
	200	0.355	6.266	4.819	53.180	10.781
	400	0.389	7.151	7.517	60.675	24.067
	800	0.481	7.923	14.416	67.733	49.710
HMRF-ASC	50	1.037	1.192	23.564	14.192	14.188
	100	1.920	2.343	50.329	27.952	24.266
	200	2.262	5.687	61.209	45.084	51.059
	400	3.479	13.151	89.159	105.179	136.358
	800	5.817	28.565	99.034	228.190	331.728
HMRF-AKKM	50	0.234	0.928	7.920	10.829	13.205
	100	0.353	1.644	16.113	19.640	25.164
	200	0.484	4.376	29.552	37.258	40.725
	400	1.074	9.982	48.781	84.675	125.713
	800	3.061	17.853	93.520	151.663	206.942

结束语 在很多实际应用中，人们在得到大量数据信息时，通常能获取一些与这些数据相关的先验知识，例如成对 must-link 或 cannot-link 约束。本文根据 HMRF 半监督聚类和核 k-means 聚类的数学联系，提出了一种 HMRF 半监督近似核 k-means 算法。该算法将成对约束加入近似核 k-means 的目标函数中，并为约束违反设计了相关的惩罚项，利用先验知识指导聚类；而且它使用由抽样点构造的近似核矩阵进行聚类，算法的空间复杂度较低，容易扩展到大数据环境中。本文在多个复杂数据集上测试了 HMRF-AKKM 算法的聚类性能，并将其与流行的半监督聚类算法进行了对比。实验结果表明，HMRF-AKKM 算法可以根据成对约束及时调整数据点的隶属关系，改善生成类簇的质量，并通过计算近似核矩阵提高了 HMRF 半监督聚类的效率。但是也应注意到，当成对约束中包含的样本点较多时，算法会花费较长时间寻找合适的类中心，如何尽量加快检查约束和调整类标签的速度，缩短聚类时间，是下一步需要研究的问题。

参 考 文 献

[1] ZHANG X T，ZHANG X C，LIU H. Weighed Multi-Task Clustering by Feature and Instance Transfer [J]. Chinese Journal of Computers，2019，42(36)：1-17. (in Chinese)
张晓彤，张宪超，刘晗. 基于特征和实例迁移的加权多任务聚类[J]. 计算机学报，2019，42(36)：1-17.

[2] MARIN D，TANG M，AYED I B，et al. Kernel clustering：density biases and solutions [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence，2019，41(1)：136-147.

[3] LIU X，ZHU X，LI M，et al. Multiple kernel k-means with incomplete kernels [OL]. https://doi. org/10. 1109/TPAMI. 2019. 2892416.

[4] CHITTA R，JIN R，HAVENS T C，et al. Scalable kernel clustering：Approximate kernel k-means [J]. arXiv：1402. 3849.

[5] MEHRKANOON S, ALZATE C, MALL R, et al. Multiclass Semisupervised Learning Based Upon Kernel Spectral Clustering [J]. IEEE Transactions on Neural Networks & Learning Systems, 2015, 26(4): 720-733.

[6] GAN H, FAN Y, LUO Z, et al. Local homogeneous consistent safe semi-supervised clustering [J]. Expert Systems with Applications, 2018, 97: 384-393.

[7] WANG W, YANG C, CHEN H, et al. Unified Discriminative and Coherent Semi-Supervised Subspace Clustering [J]. IEEE Transactions on Image Processing, 2018, 27(5): 2461-2470.

[8] YU Z, LUO P, LIU J, et al. Semi-supervised ensemble clustering based on selected constraint projection [J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(12): 2394-2407.

[9] REN Y, HU K, DAI X, et al. Semi-supervised deep embedded clustering [J]. Neurocomputing, 2019, 325: 121-130.

[10] MEI J P. Semi-supervised fuzzy clustering with partition information of subsets [OL]. https://doi.org/10.1109/TFUZZ.2018.2889010.

[11] NGUYEN B, DE BAETS B. Kernel-Based Distance Metric Learning for Supervised k-Means Clustering [OL]. https://doi.org/10.1109/TNNLS.2018.2890021.

[12] WU B, ZHANG Y, HU B G, et al. Constrained clustering and its application to face clustering in videos [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2013: 3507-3514.

[13] WANG S, GITTENS A, MAHONEY M W. Scalable kernel K-means clustering with Nyström approximation: relative-error bounds [J]. The Journal of Machine Learning Research, 2019, 20(1): 431-479.

[14] KULIS B, BASU S, DHILLON I, et al. Semi-supervised graph clustering: a kernel approach [J]. Machine Learning, 2009, 74(1): 1-22.

[15] BÜHLER T, HEIN M. Spectral clustering based on the graph p-Laplacian [C] // Proceedings of the 26th International Conference on Machine Learning. ACM, 2009: 81-88.

[16] CAI D, HE X, HAN J, et al. Graph regularized nonnegative matrix factorization for data representation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(8): 1548-1560.

[17] BAGHSHAH M S, AFSARI F, SHOURAKI S B, et al. Scalable semi-supervised clustering by spectral kernel learning [J]. Pattern Recognition Letters, 2014, 45: 161-171.

[18] BHATT R B, DHALL A. Skin Segmentation Dataset. UCI Machine Learning Repository [OL]. https://archive.ics.uci.edu/ml/index.html.

[19] KLEIN D, KAMVAR S D, MANNING C D. From Instance-level Constraints to Space-Level Constraints: Making the Most of Prior Knowledge in Data Clustering [C] // Proceedings of the 19th International Conference on Machine Learning. ACM, 2002: 307-314.

[20] JIA H J, DING S F, SHI Z Z. Approximate weighted kernel k-means for large-scale spectral clustering [J]. Journal of Software, 2015, 26(11): 2836-2846. (in Chinese)
贾洪杰，丁世飞，史忠植. 求解大规模谱聚类的近似加权核 k-means 算法 [J]. 软件学报，2015，26(11)：2836-2846.

[21] YANG Y, TAN W, LI T, et al. Consensus clustering based on constrained self-organizing map and improved Cop-Kmeans ensemble in intelligent decision support systems [J]. Knowledge-Based Systems, 2012, 32: 101-115.

[22] DING S, JIA H, DU M, et al. A semi-supervised approximate spectral clustering algorithm based on HMRF model [J]. Information Sciences, 2018, 429: 215-228.

[23] SHI Y, OTTO C, JAIN A K. Face clustering: representation and pairwise constraints [J]. IEEE Transactions on Information Forensics and Security, 2018, 13(7): 1626-1640.

[24] HE L, RAY N, GUAN Y, et al. Fast large-scale spectral clustering via explicit feature mapping [J]. IEEE Transactions on Cybernetics, 2018, 49(3): 1058-1071.

[25] ZHAO X, LIANG J, DANG C. A stratified sampling based clustering algorithm for large-scale data [J]. Knowledge-Based Systems, 2019, 163: 416-428.