基于最大平均熵率的大数据关联聚类算法

张俪文¹, 王涛¹, 罗坚¹, 杨树森², 徐宗本^{2*}

1. 西安交通大学电子与信息学部, 西安 710049

2. 西安交通大学数学与统计学院, 西安 710049

* 通信作者. E-mail: zbxu@mail.xjtu.edu.cn

收稿日期: 2019-06-04; 接受日期: 2019-07-10; 网络出版日期: 2019-12-16

国家自然科学基金 (批准号: 61772410, 61802298, 11690011, U1811461) 和国家重点研发计划 (批准号: 2017YFB1010004) 资助项目

摘要 聚类是数据挖掘和机器学习中的基本任务之一. 传统聚类方法由于其设计中对簇结构假设的限制, 导致算法在不符合其假设的数据集上, 尤其是大型高维数据集上的聚类效果较差. 本文引入了最大平均熵率的概念, 设计了一种基于图的关联聚类算法. 该算法将关联聚类问题分解为多个独立的单类优化问题, 并利用邻域消除了关联聚类对大数据的限制. 算法实现通过启发式邻域搜索和类生成简化了对最优邻域和关联聚类的求解过程, 并且设计了适应分布式计算平台的图迭代方法. 与其他聚类算法相比, 该算法在提高计算效率的同时, 对簇结构假设相对灵活, 可适用于多种分布数据. 在聚类实验中, 算法的 f1-measure 和 purity 指数均好于其他 6 种聚类算法, 而且对于高维大数据集, 算法的运行时间远远低于其他聚类算法.

关键词 聚类, 相关聚类, 熵率, 图聚类, 大数据

1 引言

聚类是数据挖掘中的基础问题, 同时也是机器学习中一类重要的无监督学习问题. 聚类算法广泛应用于计算机视觉、生物医学和交通规划等诸多领域, 也作为变量筛选、维度筛选等预处理步骤出现在多种复杂算法中^[1~3]. 聚类算法的目标是寻找数据集中相似元素聚集成簇. 现有聚类算法可按建模方法分为以下 3 类:

- **簇内建模.** 比如 k -means^[4] 和 mean-shift^[5] 算法假设存在簇中心, 且点和所属簇的簇中心相似度均高于其他簇中心. 其中 k -means 是计算全局的 k 个簇中心, 而 mean-shift 是计算局部的极值点作为簇中心.

- **簇间划分代价建模.** MinCut^[6] 和 NCut^[7] 算法考虑通过边切割对图进行划分, 在保证得到 k 个子图的同时, 切割的代价最小.

引用格式: 张俪文, 王涛, 罗坚, 等. 基于最大平均熵率的大数据关联聚类算法. 中国科学: 信息科学, 2019, 49: 1572–1585, doi: 10.1360/SSI-2019-0117
Zhang L W, Wang T, Luo J, et al. Maximum average entropy-rate based correlation clustering for big data (in Chinese). Sci Sin Inform, 2019, 49: 1572–1585, doi: 10.1360/SSI-2019-0117

• **簇内簇间相对关系建模.** 代表算法有关联聚类 (correlation clustering, CC)^[8] 和 Felzenszwalb-Huttenlocher (FH) 分割算法^[9], 关联聚类的目标是最大化簇内相似边与簇间不相似边的个数之和, 而 FH 算法则是维持簇间相似程度小于簇内相似程度的相对关系.

对于小型数据集, 可以通过分析全部数据之间的相似关系, 或是可视化选择符合簇结构假设的聚类算法, 但随着数据量增加和数据维度增高, 聚类算法的选择面临以下挑战: (1) 部分聚类算法的簇结构假设难以满足. 当数据维度增加, 簇内结构逐渐松散, 使得簇内建模算法的性能下降, 而数据量增加会使簇间边界变得模糊, 导致簇间划分代价建模算法难以准确地划分数据. (2) 算法实现受限于大数据计算平台. 已有的大数据计算平台提供的操作远远少于单机, 同时算法实现需要考虑集群节点之间交互引起的时间和空间代价, 平台限制使得部分算法很难在大数据计算平台运行.

已实现的大数据聚类算法集中在基于簇内建模的聚类算法^[10], 如 k -means 算法, 由于算法迭代过程中传输数据量小, 已在 Map Reduce^[11], HaLoop^[12], Spark^[13] 等多个大数据计算平台实现; Kim^[14] 通过网格技术减少了低维数据聚类中邻域搜索的计算量, 实现了基于 Map Reduce 的 DBSCAN 算法; Chen 等^[15] 通过 KD 树索引减少了高维数据聚类中邻域搜索的计算量, 实现了基于分布式系统的谱聚类. 但基于簇内建模的聚类算法对参数的要求较高, 如 k -means 和谱聚类中的类个数, DBSCAN 中高密度点的密度和邻域阈值, 在大数据环境下较难控制聚类结果.

为了应对大数据带来的挑战, 本文设计了一种基于最大平均熵率的关联聚类算法 (maximum average entropy-rate based correlation clustering, ECC), 该算法对关联聚类问题进行分解, 实现了简化优化过程的目标, 引入邻域图以消除大数据对关联聚类的限制, 提高了算法效率. 与其他聚类算法相比, ECC 对簇结构假设灵活, 处理大数据聚类问题有较高的执行效率和较好的精度.

2 相关工作

关联聚类算法是由 Bansal 等^[8] 提出的基于边标签的图聚类算法, 算法目标为最大化类内相似边与类间不相似边个数 (或最小化类内不相似边和类间相似边的个数), 该算法的特点是不限定类数, 同时允许类边缘有一定的重叠. 本节将介绍关联聚类算法、已有的求解方法以及关联聚类对大数据的限制.

2.1 关联聚类

对于数据集 $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, 生成以 n 个数据作为节点的全连接图 $G(\mathbf{V}, \mathbf{E})$, 其中, 顶点集 $\mathbf{V} = \mathbf{X}$, $\mathbf{E}$ 为带权重的边集, $\mathbf{e}_{ij} \in \mathbf{E}$ 的权重 ω_{ij} 为两节点 i, j 之间的相似关系. 设 $\mathbf{Y}$ 为关联聚类中的一组边的标签,

$$y_{ij} = \begin{cases} +, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are similar;} \\ -, & \text{othersize.} \end{cases} \quad (1)$$

关联聚类的目标为最大化类内 “+” 边和类间 “-” 边的个数 (或最小化类内 “-” 边和类间 “+” 边的个数), 从而实现类内高内聚, 类间低耦合, 同时允许类边缘重合, 如图 1 所示. 使用边的权重和代替边的个数, 其目标函数 $\mathcal{F}(\cdot, \cdot)$ 可以表示为

$$\mathcal{F}(\mathbf{X}, \mathbf{Y}) = \sum_{(i,j) \in \mathbf{E}} \text{Sim}(i, j) y_{ij}, \quad (2)$$

其中 $\text{Sim}(i, j)$ 是衡量 $\mathbf{x}_i$ 与 $\mathbf{x}_j$ 节点相似度的函数, 其值为正, 表示节点相似, 其值为负, 则节点不相似, 即 $\mathbf{x}_i$ 与 $\mathbf{x}_j$ 节点越相似, $\text{Sim}(i, j)$ 的值越大, 反之, 则其值越小. 关联聚类过程可视为不断优化标

图 1 (网络版彩图) 关联聚类示意图

Figure 1 (Color online) Illustration of correlation clustering

签 $\mathbf{Y}$, 最大化目标函数 $\mathcal{F}(\mathbf{X}, \mathbf{Y})$ 的过程,

$$\hat{\mathbf{Y}} = \arg \max_{\mathbf{Y}} \mathcal{F}(\mathbf{X}, \mathbf{Y}). \quad (3)$$

然而, 该优化问题是 NP-Hard 问题, 并不能直接求解.

2.2 已有关联聚类的求解方法

已有的关联聚类求解方法主要有两类, 一类是对原有的目标函数进行松弛, 从不可解问题转化为可解问题, 另一类是通过引入其他可解模型对目标函数进行重新解释.

最常见的松弛方式是通过路径不等式 (cycle inequality) 和中心不等式 (odd-wheel inequality)^[16, 17], 将整数规划问题转化为线性规划问题. Vitaladevuni 等^[18] 通过半定松弛和线性规划得到了对原问题的凸松弛. Tsochantaridis 等^[19] 在关联聚类中引入结构化支持向量机 (structured support vector machine, S-SVM), 实现了优化分类器, 同时训练和学习边的标签值的目标.

Bagon 等^[20] 通过 Potts 模型对最小化类间相似边的个数的问题进行了重新解释, 使原问题可解. Chierichetti 等^[21] 将 Pivot 模型应用于正负边判断中, 通过 MapReduce 计算模型实现了快速关联聚类. Blondel^[22] 提出探测大型网络中社区的 louvain 算法, 利用模块度度量子图的独立程度, 通过多尺度优化子图的模块度, 实现对海量图数据的多尺度聚类.

由于第 1 类求解方式处理复杂, 不能对较大数据集进行处理; 第 2 类求解方式可以处理海量图数据, 但不能处理非图大数据, 且聚类精度依赖解释模型.

2.3 关联算法的大数据的限制

关联聚类对大数据的限制主要体现在: (1) “+”“-” 边标签基于全连接图 $G(\mathbf{V}, \mathbf{E})$ 标注, 即每个节点都有指向所有节点的 n 条边, 当 n 较小时, “+” 边的作用 $f^+(\cdot)$ 与 “-” 边的作用 $f^-(\cdot)$ 大小相差不大, 但随着 n 变大, 体量较小的类中元素的 “-” 边数目剧增, $f^-(\cdot)$ 的值增加, 使得小类被其他大类分解或吞并; (2) 随着 n 增加, 边的数目以平方级增加, 计算量巨大.

目前已有的处理方法有以下两个.

- 将 “-” 边的标签值设为 0, 从而消除 $f^-(\cdot)$ 的影响;

- 使用 k 近邻生成邻接图代替全连接图, 使边的数目随数据量增长而线性增加.

第 1 种处理方法虽然在结果上去除了“-”边的影响, 但实际计算量没有减少, 而第 2 种方法中近邻 k 的取值对结果的影响很大.

本文算法 ECC 将关联聚类问题分解为多个独立的单类优化问题, 实现了关联聚类的简化和并行化; 使用基于最大平均熵率的邻域图代替全连接图, 消除了关联聚类对大数据的限制, 同时提高算法效率. 算法通过启发式邻域搜索和类生成简化了对最优邻域和关联聚类的求解过程, 并且设计了适应分布式计算平台的图迭代方法. 与其他聚类算法相比, 该算法可适用于多种类型分布数据, 且适用于大数据聚类.

3 基于最大平均熵率的关联聚类算法

3.1 关联聚类目标函数分解

本节主要是将关联聚类的全局优化目标 $\mathcal{F}(\mathbf{X}, \mathbf{Y})$ 分解为多个单类优化目标之和 $\sum_{\mathbf{c} \in \mathbf{C}} \mathcal{F}(\mathbf{c}, \mathbf{Y}_{\mathbf{c}})$, 且各个单类优化之间相互独立, 其中 $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k\}$ 为聚类的结果, 且 $\mathbf{c}_1 \cup \mathbf{c}_2 \cup \dots \cup \mathbf{c}_k = \mathbf{V}$, 类数 k 为任意正整数, $\mathbf{Y}_{\mathbf{c}}$ 为类 $\mathbf{c}$ 在 $\mathbf{Y}$ 中的标签.

假设聚类结果 $\mathbf{C}$ 中各类之间允许存在交集, 即 $\|\mathbf{c}_i \cap \mathbf{c}_j\| \geq 0$, $\mathbf{c}_i, \mathbf{c}_j \in \mathbf{C}$, $\mathbf{c}$ 为聚类结果中某个类, 则类 $\mathbf{c}$ 的目标函数 $\mathcal{F}(\mathbf{c}, \mathbf{Y}_{\mathbf{c}})$ 可表示为

$$\mathcal{F}(\mathbf{c}, \mathbf{Y}_{\mathbf{c}}) = \sum_{i \in \mathbf{c}} \sum_{(i,j) \in \mathbf{E}} \text{Sim}(i, j) y_{ij}. \quad (4)$$

由于各类之间允许存在重叠, 单类的目标函数优化是独立的, 互不影响的, 因此, 多个单类的目标函数与全局目标函数之间的关系可以表示为

$$\mathcal{F}(\mathbf{X}, \mathbf{Y}) = \sum_{\mathbf{c} \in \mathbf{C}} \frac{1}{2} \mathcal{F}(\mathbf{c}, \mathbf{Y}_{\mathbf{c}}) - \frac{\lambda}{2} \mathcal{F}(\delta(\mathbf{C}), \mathbf{Y}_{\delta(\mathbf{C})}), \quad (5)$$

其中 $\delta(\mathbf{C})$ 为类之间的交集部分, $\lambda \geq 1$ 是其重叠程度系数. 由于关联聚类中类重叠只出现在类边缘, 且数量较少, 忽略重叠部分 $\delta(\mathbf{C})$ 对目标函数的影响, 可将全局优化分解为多个单类的目标函数优化, $\hat{\mathbf{Y}} \approx \hat{\mathbf{Y}}_{\mathbf{c}_1} \cup \hat{\mathbf{Y}}_{\mathbf{c}_2} \cup \dots \cup \hat{\mathbf{Y}}_{\mathbf{c}_k}$, 单类的优化目标为

$$(\hat{\mathbf{c}}, \hat{\mathbf{Y}}_{\mathbf{c}}) = \arg \max_{\mathbf{c}, \mathbf{Y}_{\mathbf{c}}} \mathcal{F}(\mathbf{c}, \mathbf{Y}_{\mathbf{c}}), \quad (6)$$

至此已将关联聚类的全局优化目标函数分解为多个独立的单类优化目标函数.

3.2 基于最大平均熵率的邻域图

在构造邻域图时, 本文算法引入平均熵率作为节点邻域的判别准则. 熵率是建立在条件熵基础上, 衡量随机过程不确定性变化的指标. 熵率越大, 熵的变化越快. 若将与节点 i 连接的节点看作 i 在下一时刻即将跳转的状态, 边的权重为跳转的概率, 则权重越大, 跳转的可能性越大, 而其中包含的不确定性越小; 反之, 跳转的可能性越小, 则不确定性越大. 文献 [23] 描述了基于图 $\mathbf{G}(\mathbf{V}, \mathbf{E})$ 的熵率可表示为

$$\begin{aligned} \mathcal{H}(\mathbf{G}) &= - \sum_i \mu_i \sum_j \frac{\omega_{ij}}{\omega_i} \log \frac{\omega_{ij}}{\omega_i}, \\ \omega_i &= \sum_{(i,j) \in \mathbf{E}} \omega_{ij}, \quad \omega_T = \sum_{i \in \mathbf{V}} \omega_i, \quad \mu_i = \frac{\omega_i}{\omega_T}, \end{aligned} \quad (7)$$

单节点的公式为

$$\mathcal{H}(\mathcal{N}(i)) = - \sum_{j \in \mathcal{N}(i)} \frac{\omega_{ij}}{\omega_i} \log \omega_{ij} - \log \omega_i, \quad (8)$$

其中 $\mathcal{N}(i)$ 为点 i 的邻域.

随着 $\mathcal{N}(i)$ 扩大, 函数 $\mathcal{H}(\cdot)$ 单调递增, 但每条边引起的增长速率 $\Delta \mathcal{H} = -\frac{\omega \log \omega}{\omega_i}$ 不同, 使用平均熵率来衡量点邻域中所有边的不确定性变化,

$$\bar{\mathcal{H}}(\mathcal{N}(i)) = \frac{\mathcal{H}(\mathcal{N}(i))}{|\mathcal{N}(i)|}, \quad (9)$$

节点的最优邻域为不确定性骤增的边缘, 即最大化该节点的平均熵率的邻域,

$$\hat{\mathcal{N}}(i) = \arg \max_{\mathcal{N}(i)} \bar{\mathcal{H}}(\mathcal{N}(i)). \quad (10)$$

4 优化求解与算法流程

本文设计了启发式邻域搜索和类生成求解最大平均熵率邻域和单类, 给出了算法的实现流程, 针对大数据设计了基于图迭代的方法, 讨论了各算法的计算复杂度.

4.1 最大平均熵率邻域求解方案

算法使用启发式邻域搜索求解最大平均熵率邻域, 求解 (10) 式可以看作寻找节点 i 的连接边中引起不确定性变化最快的边 e_{ij} , 边的权重大于 w_{ij} 的不确定性较小, 而小于 w_{ij} 的不确定性较大, 将与节点 i 相连的边, 按权重从大到小的顺序依次添加到邻域内, 计算对应的平均熵率并在最大点停止, 使平均熵率最大的边集合为所求邻域. 算法伪代码见算法 1.

Algorithm 1 Neighboring by greedy algorithm-nbr $_{\bar{\mathcal{H}}}(\cdot, \cdot)$

Require: Node i , initial edge set of i $E_i = \{e_{ij} \in E\}$;

Ensure: Edge set based on maximum average entropy rate $\mathcal{N}_i$;

```

1:  $\mathcal{N}_i \leftarrow \emptyset$ ;
2:  $E'_i \leftarrow$  Order edges in  $E_i$  by weight in descending order;
3:  $\bar{\mathcal{H}}_{\max} \leftarrow 0$ ; % Variable to store the value of maximum average entropy rate.
4: for Edge  $e_{ij}$  in  $E'_i$  do
5:    $\mathcal{H}_{\text{temp}} \leftarrow \mathcal{H}(\mathcal{N}_i)$ ; % Average entropy rate of current edge set.
6:   if  $\bar{\mathcal{H}}_{\text{temp}} \geq \bar{\mathcal{H}}_{\max}$  then
7:     Add  $e_{ij}$  to  $\mathcal{N}_i$ ;
8:      $\bar{\mathcal{H}}_{\max} \leftarrow \bar{\mathcal{H}}_{\text{temp}}$ ;
9:   else
10:    Return  $\mathcal{N}_i$ ;
11:   end if
12: end for
```

对于每个节点, 算法复杂度由排序算法和连接边的体量决定, 最高为 $\mathcal{O}(|E_i|^3)$. $|E_i|$ 的大小由算法初始化返回的初始邻域中边数决定, 设初始化返回的初始邻域内的平均边数为 $\bar{d}$, 则算法的复杂度最高为 $\mathcal{O}(|V|\bar{d}^3)$.

4.2 单一类求解方案

本文使用启发式类生成方法求解单类元素集合,基本思路是将节点的“+”和“-”边看作类元素和非类元素的吸引作用,通过比较两种作用的大小,决定节点是否属于该类。

对各点的连接边以“+”“-”进行二分,则单类的目标函数(6)式可以写为所有类内元素的“+”“-”边的作用之和,

$$\begin{aligned}\mathcal{F}(\mathbf{c}, \mathbf{Y}_c) &= \sum_{i \in \mathbf{c}} [\mathbf{f}^+(i) + \mathbf{f}^-(i)], \\ \mathbf{f}^+(i) &= \sum_{(i, j^+) \in \mathbf{E}^+} \text{Sim}(i, j^+) y_{ij^+}, \\ \mathbf{f}^-(i) &= \sum_{(i, j^-) \in \mathbf{E}^-} \text{Sim}(i, j^-) y_{ij^-}, \\ \mathbf{E}^+ &= \{e : y_e = +\}, \quad \mathbf{E}^- = \{e : y_e = -\}.\end{aligned}\tag{11}$$

以单节点为目标,若 $\text{Sim}(\cdot, \cdot) \in [-1, 1]$, 则点在全连接图 $\mathbf{G}(\mathbf{V}, \mathbf{E})$ 中受到的作用 $\mathbf{f}(\cdot)$ 可以表示为 $\mathbf{f}^+(\cdot) + \mathbf{f}^-(\cdot)$, 且 $\mathbf{f}(\cdot) \in [-n, n]$, 作用的大小反映了标签的正确与否: 当 $\mathbf{f}(\cdot) = -n$ 时, 点的“+”边和“-”边标签全部标记错误, 且“+”边的权重为 -1 (最不相似), “-”边的权重为 1 (最相似); 当 $\mathbf{f}(\cdot) = n$ 时, 点的“+”边和“-”边标签全部标记正确, 且“+”边的权重为 1 , “-”边的权重为 -1 . 优化目标(6)式可以看作所有满足 $\mathbf{f}(\cdot) \geq 0$ 条件的点的集合, 即 $\hat{\mathbf{c}} = \{i : \mathbf{f}(i) \geq 0\}$.

对数据集 $\mathbf{X}$ 中每个点进行单类生成, 可以获得以每个点为中心的类, 这些类之间存在严重的重复和重叠, 需要筛选出具有代表性的类. 本文参考文献[24]中对类中心的定义——“类中心是密度较高且距离较远的点”——在每次单类生成时, 选择度高且没有被其他类包含的点作为起始点, 算法伪代码见算法 2, 该算法的复杂度由最大迭代次数、节点的度以及节点数量决定 $\mathcal{O}(|\mathbf{E}| |\mathbf{V}|)$.

Algorithm 2 Correlation clustering on single cluster- $\text{Cc}_s(\cdot, \cdot)$

Require: Graph $\mathbf{G}(\mathbf{V}, \mathbf{E})$: $v \in \mathbf{V}$, $v.\text{cluster} = \text{'unset'}$, $e \in \mathbf{E}$, $e.\text{label} = -$, maximum iteration $\text{iter}_{\max}$;

Ensure: Node set of cluster $\mathbf{c}$;

```

1:  $\mathbf{V}_{\text{temp}} \leftarrow$  nodes from  $\mathbf{V}$  with  $.\text{cluster} = \text{'unset'}$ ;           % Unclustered node set.
2:  $i \leftarrow$  node from  $\mathbf{V}_{\text{temp}}$  with maximum degree;
3: Set edge in  $\mathbf{E}_i$   $.\text{label} = +$ ;
4:  $\text{iter} \leftarrow 0$ ;
5: while  $\text{iter} < \text{iter}_{\max}$  do
6:   for each  $v$  in  $\mathbf{V}$  do
7:     if  $\mathbf{f}^+(v) \geq \mathbf{f}^-(v)$  then
8:       Set  $v.\text{cluster} = i$ , set edge label in  $\mathbf{E}_v$  to +;           %  $v$  is contained by cluster  $\mathbf{c}$ .
9:     else
10:      Set  $v.\text{cluster} = \text{'unset'}$ , set edge label in  $\mathbf{E}_v$  to -;       %  $v$  is not contained by cluster  $\mathbf{c}$ .
11:    end if
12:  end for
13:   $\text{iter} \leftarrow \text{iter} + 1$ ;
14: end while
15:  $\mathbf{c} \leftarrow$  nodes from  $\mathbf{V}$  with  $.\text{cluster} = i$ ;
16: Return  $\mathbf{c}$ .
```

算法 2 的计算结果为单个类的元素集合, 可以多次求解单个类获得全局的聚类结果, 算法伪代码见算法 3. 算法 3 的复杂度为算法 2 的 K 倍加上初始化邻域的复杂度, 其中 K 为类数. 需要说明的

是, 类数 K 可以设定, 即返回前 K 类, 也可以迭代至所有节点类属性都被标记时为止, 算法 3 为第 2 种情况; 初始化邻域复杂度由邻域搜索结构和方法决定, 如使用相似度矩阵的复杂度为 $\mathcal{O}(|V|^2)$, k 近邻的复杂度为 $\mathcal{O}(|V| \log |V|)$, Hash 表索引的复杂度为 $\mathcal{O}(1)$.

Algorithm 3 Correlation clustering on multiple clusters- $\text{Cc}_m(\cdot, \cdot, \cdot)$

Require: Dataset $\mathbf{X}$, initial neighbor parameter σ_0 , maximum iteration $\text{iter}_{\max}$;

Ensure: Clustering result $\mathbf{C}$;

```

1:  $\mathbf{E} \leftarrow \emptyset$ ;
2: for each  $x$  in  $\mathbf{X}$  do
3:    $\mathbf{E}_x \leftarrow \mathcal{N}_{\sigma_0}(x)$ ;           %  $\mathcal{N}_{\sigma_0}(\cdot)$  return initial neighbor with parameter  $\sigma_0$ .
4:    $\mathcal{N}_x \leftarrow \text{nbr}_{\mathcal{H}}(x, \mathbf{E}_x)$ ;
5:   Add  $\mathcal{N}_x$  to  $\mathbf{E}$ ;
6: end for
7: Generate graph  $\mathbf{G}(\mathbf{V} = \mathbf{X}, \mathbf{E})$ ;
8: Initialize  $\mathbf{G}$ :  $v \in \mathbf{V}$ ,  $v.\text{cluster} = \text{'unset'}$ ,  $e \in \mathbf{E}$ ,  $e.\text{label} = -$ ;
9:  $\mathbf{C} \leftarrow \{\}$ ;
10: while  $\{v \in \mathbf{V}, v.\text{cluster} = \text{'unset'}\} \neq \emptyset$  do
11:    $\mathbf{c} \leftarrow \text{Cc}_s(\mathbf{G}, \text{iter}_{\max})$ ;
12:   Add  $\mathbf{c}$  to  $\mathbf{C}$ ;
13:   Reset  $\mathbf{G}$ :  $e \in \mathbf{E}$ ,  $e.\text{label} = -$ ;
14: end while
15: Return  $\mathbf{C}$ .

```

4.3 面向大数据的关联聚类算法

设计面向大数据的关联聚类算法, 需要改进以下操作: (1) 起始点的选取需要对全图未聚类的节点按度进行排序; (2) 类是串行生成的, 且通过全图节点迭代产生. 这些操作会造成算法计算和通信量激增, 不适合在大数据平台上运行.

本文将初始化起始点和类生成的操作由图迭代方法实现, 使算法可以在大数据平台运行.

初始化起始点的主要任务是筛选出未聚类点集中度最大的节点, 并将与之连接的边的 label 属性设置为 “-”, 原始算法中为了避免已有类对未生成类的干扰, 每次需要重置所有边的 label 属性. 改进方法是将边的 label 属性从原有的 “+” “-” 标签, 变为节点标签, 初始化过程将其赋值为连接节点中度最大的节点标签, 算法伪代码见算法 4.

Algorithm 4 Start points initialization- $\text{Initspts}(\cdot)$

Require: Graph $\mathbf{G}(\mathbf{V}, \mathbf{E})$: $v \in \mathbf{V}$, $v.\text{cluster} = \text{'unset'}$, $e \in \mathbf{E}$, $e.\text{label} = \text{'unset'}$;

Ensure: Initialized graph $\mathbf{G}(\mathbf{V}, \mathbf{E}')$;

```

1:  $\mathbf{E}' \leftarrow \{\}$ ;
2: for each  $e_{ij}$  in  $\mathbf{E}$  do
3:   if  $e_{ij}.\text{label} == \text{'unset'}$  then
4:      $e_{ij}.\text{label} \leftarrow \text{node with maximum degree between node } i, j$ ;
5:   else
6:      $e_{ij}.\text{label} \leftarrow \text{node with maximum degree among node } i, j \text{ and } e_{ij}.\text{label}$ ;
7:   end if
8:   Add  $e_{ij}$  to  $\mathbf{E}'$ ;
9: end for
10: Return  $\mathbf{G}(\mathbf{V}, \mathbf{E}')$ .

```

对类生成操作的改进是将节点的标签从唯一值改为定长列表, 列表长度 l 为单个节点最多可以属于的类数, 此时节点受到的作用从 f^+ 和 f^- 变为多个节点作用. 当受到多个节点作用时, 将某一节点 i 的作用看作“+”边作用 $f^+ = f_i$, 其他节点的作用之和 $\sum f_{j \neq i}$ 看作 f^- , 根据 f^+ 和 f^- 大小关系判断是否将 i 标签传入该节点; 当有新的节点标签传入时, 保留度较大的节点标签. 该改进在传播了度较大的节点标签的同时, 保证了类之间的重叠关系, 算法伪代码见算法 5.

Algorithm 5 Clusters generation-Gencluster($\cdot, \cdot, \cdot$)

Require: Edge initialized graph $G(V, E)$, overlapping parameter l , maximum iteration $\text{iter}_{\max}$;

Ensure: Node set of clusters C ;

```

1: Initialize  $V$  of  $G(V, E)$ :  $v \in V$ ,  $v.\text{cluster} \leftarrow$  empty list with  $l$  length;
2: while  $\text{iter} < \text{iter}_{\max}$  do
3:   for each  $v$  in  $V$  do
4:      $K \leftarrow$  collection of label value of edge in  $E_v$ ;
5:     templist  $\leftarrow$  empty list;
6:     for each  $k$  in  $K$  do
7:       if  $f_k(v) \geq \sum_{i \neq k} f_i(v)$  then
8:         Add  $k$  to templist;
9:       end if
10:    end for
11:     $v.\text{cluster} \leftarrow$  Keep the top  $l$  items in templist ordered by its force  $f_k(v)$ ;
12:  end for
13:   $\text{iter} \leftarrow \text{iter} + 1$ ;
14: end while
15:  $C \leftarrow$  aggregate nodes by their label;
16: Return  $C$ .
```

改进后的起始点初始化和类生成部分的算法复杂度为 $O((I+1)|V|\bar{d})$, 其中 I 为最大迭代次数, 复杂度是算法 2 复杂度的 $\frac{1}{K}$, 面向大数据的关联聚类的整体算法伪代码见算法 6.

Algorithm 6 Correlation clustering on big data-Cc_b($\cdot, \cdot, \cdot, \cdot$)

Require: Dataset X , initial neighbor parameter σ_0 , maximum iteration $\text{iter}_{\max}$, overlapping parameter l ;

Ensure: Node set of clusters C ;

```

1:  $E \leftarrow \emptyset$ ;
2: for each  $x$  in  $X$  do
3:    $E_x \leftarrow \mathcal{N}_{\sigma_0}(x)$ ;
4:    $\mathcal{N}_x \leftarrow \text{nbr}_{\mathcal{H}}(x, E_x)$ ;
5:   Add  $\mathcal{N}_x$  to  $E$ ;
6: end for
7: Generate graph  $G(V = X, E)$ ;
8: Initespts( $G$ );
9:  $C \leftarrow \text{Gencluster}(G, l, \text{iter}_{\max})$ ;
10: Return  $C$ .
```

5 实验与分析

聚类实验在多个实验数据集和真实大数据上进行测试, 测试平台包括单机环境和分布式计算平台. ECC 算法聚类结果与 6 种聚类算法对比, 对比算法包括 k -means [4], AP [25], Mean-shift [5],

Spectral^[26], HAC^[27] 和 Ncut^[7], 对比内容包括聚类指标和运行效率.

5.1 单机实验配置

单机实验中使用的数据集包括 4 个不同分布的二维标准数据集和 1 个包括 500 张图片的小型人脸数据集.

- 二维标准数据集包括来自 Clustering dataset^[28] 的 4 个数据集: Aggregation, Flame, S3 和 Spiral. 在二维标准数据集上使用 Gauss 核函数 $\omega_{ij} = e^{-\left(\frac{\|v_i - v_j\|_2}{2\sigma^2}\right)^2}$ 衡量两点之间的相似度, 其中 $\|v_i - v_j\|_2$ 为两点之间的欧氏距离, σ 为核参数;

- 人脸数据集的 500 张人脸图片来自人脸探测^[29] 后的 CASIA-Webface 数据集¹⁾, 经过 LResNet34E-IR^[30] 深度网络提取得到 512 维实数稠密向量. 特征向量需进行归一化处理, 使用 Laplace 核函数 $\omega_{ij} = e^{-\left(\frac{|v_i \cdot v_j|}{\sigma}\right)}$ 衡量两点之间的相似度.

核参数 σ 的选取会影响邻域确定和算法效率, 但该参数与数据分布情况相关. 本文在实验选取的初始 σ_0 略大于真实 σ , 通过 σ_0 邻域与点之间的关系, 优化 σ 使之趋近真实值, 优化目标可以表示为

$$\hat{\sigma}_i = \arg \max_{\sigma} \frac{\sum_{j \in \mathcal{N}_{\sigma_0(i)}} \omega_{ij}^{\sigma_0} \sigma_0}{|\mathcal{N}_{\sigma_0(i)}| \sigma} + |\sigma|, \quad (12)$$

其中 $\omega_{ij}^{\sigma_0}$ 为初始的相似度值. 基本思想是通过缩小核参数实现对基于 σ_0 的密度值向的中心聚集, 同时, 为了防止 σ 过小, 加入正则项 $|\sigma|$. 优化前后的结果如图 2 所示, 通过优化各个点的核参数 σ , 各点的邻域范围更加准确.

本文选择两个广泛使用的聚类指数 f1-measure 和 purity 作为聚类衡量指标.

f1-measure 指数是通过聚类精确率和召回率综合评估聚类质量的指标. 对于一个聚类 k , 其基于给定类 c 的 f1-measure 可以表示为该类的精确率 $p(k, c)$ 和召回率 $r(k, c)$ 的调和平均数,

$$\begin{aligned} p(k, c) &= \frac{|k \cap c|}{|k|}, & r(k, c) &= \frac{|k \cap c|}{|c|}, \\ f1(k, c) &= \frac{2 \times p(k, c) \times r(k, c)}{p(k, c) + r(k, c)}, \end{aligned} \quad (13)$$

其中 $|\cdot|$ 是集合的模. 对于聚类结果 $K = \{k_1, k_2, \dots, k_k\}$, 在类 $C = \{c_1, c_2, \dots, c_c\}$ 上的 f1-measure 值可以表示为

$$f1(K, C) = \sum_{i \in [1, c]} \frac{|c_i|}{|C|} \times \max_{j \in [1, k]} f1(k_j, c_i), \quad (14)$$

而 purity 为 f1-measure 指数中的精确率 $p(k, c)$.

5.2 单机实验结果与分析

ECC 算法与其他 6 种聚类算法在标准二维数据集上的聚类结果对比如图 3 所示, 指数对比见表 1^[4, 5, 7, 25~27]. 实验结果表明, 其他聚类算法对于符合簇结构假设的数据集都能取得较好结果, 如 k -means 在 S3 数据集取得了较好的聚类结果, 但对于其他结构的簇结果较差; ECC 算法是对簇内和簇间的相对关系建模, 并不受簇结构限制, 因此适用于多种簇结构.

小型人脸数据集上的聚类指数见表 1. 对于高维数据, 簇结构十分稀疏, 造成部分聚类算法的簇内高密度簇间低密度假设不成立, 算法性能下降; 而在 ECC 算法中点的归属只由邻域内点决定, 因此在簇内部稀疏的情况下依然可以得到较好的聚类结果, 其聚类指数远高于其他聚类算法.

1) <http://www.cbsr.ia.ac.cn/>.

图 2 (网络版彩图) σ 优化前后对比图. 以 Aggregation 数据集中的点为例, (a) 是初始 $\sigma_0 = 10$ 的邻域点范围, (b) 是优化后 $\hat{\sigma} = 0.4$ 的邻域点范围.

Figure 2 (Color online) Comparison of neighbors by original σ_0 and optimized $\hat{\sigma}$. Taking points in Aggregation for instance, neighbors by original $\sigma_0 = 10$ are shown in (a) and neighbors by optimized $\hat{\sigma} = 0.4$ are shown in (b).

图 3 (网络版彩图) 标准二维数据集上的聚类结果对比

Figure 3 (Color online) Clustering results comparison on standard 2D datasets

5.3 大数据实验配置

实验使用 CASIA-Webface 数据集, 共有来自 10575 人脸的 494414 张图片, 经过人脸探测过滤后可用图像为 307784 张.

在大数据实验中, 使用 Minhash 对原始向量进行索引, 索引后的数据可通过 LSH (local sensitive Hashing) 直接返回邻域. Minhash 索引建立在原始向量二值化的基础上, 原始向量 $\mathbf{x} \in \mathbb{R}^d$ 需要通过二值化函数 $\text{bin}(\cdot)$ 转化为二进制编码 $\mathbf{e} \in \{0, 1\}^d$, $\text{bin}(\cdot)$ 通过判断向量 $\mathbf{x}$ 各个维度的值 x_i 与数据集在

表 1 标准二维数据集上的聚类指标对比, 指标包括 f1-measure ($F1 \in [0, 1]$) 和 purity ($P \in [0, 1]$)

Table 1 Clustering metrics comparison on standard 2D datasets, including f1-measure and purity

Datasets	Metrics	ECC	k -means [4]	AP [25]	Mean-shift [5]	Spectral [26]	HAC [27]	Ncut [7]
Flame	$F1$	1.0000	0.8406	0.8243	0.8528	0.9832	0.7732	0.9917
	P	1.0000	0.8647	0.8488	0.8586	0.8787	0.8595	0.9917
Spiral	$F1$	1.0000	0.3502	0.3225	0.3562	1.0000	0.3917	0.5823
	P	1.0000	0.3447	0.3566	0.3475	1.0000	0.3337	0.5992
Aggregation	$F1$	1.0000	0.8504	0.8618	0.9595	0.7491	0.9097	0.9925
	P	1.0000	0.9453	0.9415	0.9685	0.8242	0.9646	0.9930
Face	$F1$	0.9669	0.8637	0.8867	0.6883	0.3478	0.9605	0.1644
	P	0.9727	0.7134	0.7583	0.5445	0.1738	0.8335	0.1952

该维度的中值 m_i 的大小关系来决定,

$$e_i = \begin{cases} 1, & \text{if } x_i > m_i; \\ 0, & \text{othersize.} \end{cases} \quad (15)$$

Minhash 基于 Jaccard 距离, 因此对于向量 x_1, x_2 , 其二进制编码分别是 e_1, e_2 , 则 e_1, e_2 的基于 Jaccard 距离的相似度可以写为

$$j(e_1, e_2) = \frac{|e_1 \cap e_2|}{2d - |e_1 \cap e_2|}, \quad (16)$$

其中 $e_1 \cap e_2$ 为二进制串中值相同的位的集合.

5.4 大数据实验结果分析

大数据实验在通用大数据计算框架 Spark 上进行, 该计算框架广泛应用于科学研究和工业生产中, 是目前比较成熟的通用大数据计算平台之一. 实验将对 ECC, k -means 和 k -NN+Louvain 在 CASIA-Webface 数据集上的聚类结果和运行效率. 其余 6 种聚类算法, 因其计算过程难以在 Spark 平台实现而不参与对比.

ECC, k -means 和 k -NN+Louvain 的 f1-measure 分别为 0.5870, 0.5853 和 0.6419, purity 分别为 0.9912, 0.8980 和 0.9289. 其中, ECC 和 k -NN+Louvain 的指数要高于 k -means, 说明对于 CASIA-Webface 数据集, 簇内簇间相对关系建模的聚类算法要优于簇内建模的聚类算法; ECC 相对于 k -NN+Louvain, 虽然 f1-measure 的值较低, 但获得了更高纯度的簇.

运行时间随数据量增长的变化趋势如图 4 所示. k -means 和 k -NN+Louvain 算法的运行时间随数据量线性增加而超线性增长, 且远高于 ECC 算法运行时间的增长曲线. k -means 算法的运行时间快速增长是由于类数 k 随数据量增加而增加, 引起计算平台的数据计算量和传输量的骤增; k -NN+Louvain 算法的时间开销主要集中在 k 邻域搜索过程, 该过程运行时间随数据量和数据维度增加而快速增长; 而 ECC 算法基于邻接图设计, 数据量的增加对基于最优邻域的邻接图中边数目影响并不大, 同时算法通过图迭代方式计算, 因此算法运行效率较高.

6 结论

本文在相关聚类算法的基础上引入最大平均熵率的概念, 设计了一种基于图的关联聚类算法. 该

图 4 (网络版彩图) ECC 与 k -means 算法在高维大数据上的运行时间对比: 运行时间为相同运行环境下 3 次运行时间的平均值; k -means 的参数 k 分别为 1003, 2119, 3141, 4207, 5181, 6209, 7232, 8252, 9306; ECC 的参数为 $\sigma_0 = 0.5$, $\text{iter}_{\max} = 10$, $l = 1$; k -NN+Louvain 的参数为 $l = 0$, $k = 10$.

Figure 4 (Color online) Comparison on running time of ECC and k -means: running time is the average of 3 times running; values of k for k -means are 1003, 2119, 3141, 4207, 5181, 6209, 7232, 8252 and 9306; parameters for ECC are $\sigma_0 = 0.5$, $\text{iter}_{\max} = 10$, $l = 1$; parameters for k -NN+Louvain are $l = 0$, $k = 10$.

算法将原始关联聚类的全局目标函数分解为多个独立的单类目标函数, 并利用邻域图消除了大数据在关联聚类上的限制. 在算法实现上, 通过启发式邻域搜索和类生成算法实现了最大平均熵率邻域求解和原关联聚类问题的求解, 算法的复杂度为 $\mathcal{O}(Cn)$, 其中 C 为常数, n 为数据量.

为了使算法适用于大数据计算平台, 我们将串行启发式算法改进为并行的图迭代算法, 算法的复杂度进一步下降到单机算法的 $\frac{1}{K}$, K 为常数. 在多组与其他聚类算法对比的实验中, ECC 算法的 f1-measure 和 purity 均好于其他聚类算法, 对于大数据集, ECC 算法不仅拥有较高的聚类指数, 其运行时间也远远低于其他聚类算法.

参考文献

- Kolosnjaji B, Zarras A, Webster G, et al. Deep learning for classification of malware system call sequences. In: Proceedings of Australasian Joint Conference on Artificial Intelligence. Berlin: Springer, 2016. 137–149
- Bühlmann P, Rütimann P, van de Geer S, et al. Correlated variables in regression: clustering and sparse estimation. J Stat Plan Infer, 2013, 143: 1835–1858
- He Z, Liu H, Wang Y W, et al. Generative adversarial networks-based semi-supervised learning for hyperspectral image classification. Remote Sens, 2017, 9: 1042
- Kanungo T, Mount D M, Netanyahu N S, et al. An efficient k-means clustering algorithm: analysis and implementation. IEEE Trans Pattern Anal Mach Intell, 2002, 24: 881–892
- Comaniciu D, Meer P. Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell, 2002, 24: 603–619
- Estrada F, Jepson A, Chennubhotla C, et al. Spectral embedding and min cut for image segmentation. In: Proceedings of BMVC, Kingston, 2004. 1–10
- Shi J B, Malik J. Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell, 2000, 22: 888–905
- Bansal N, Blum A, Chawla S. Correlation clustering. Mach Learn, 2004, 56: 89–113
- Felzenszwalb P F, Huttenlocher D P. Efficient graph-based image segmentation. Int J Comput Vision, 2004, 59: 167–181
- Mohebi A, Aghabozorgi S, Wah T Y, et al. Iterative big data clustering algorithms: a review. Softw Pract Exper, 2016, 46: 107–129

- 11 Zhao W Z, Ma H F, He Q. Parallel k-means clustering based on mapreduce. In: Proceedings of IEEE International Conference on Cloud Computing. Berlin: Springer, 2009. 674–679
- 12 Bu Y Y, Howe B, Balazinska M, et al. HaLoop: efficient iterative data processing on large clusters. Proc VLDB Endow, 2010, 3: 285–296
- 13 Zaharia M, Chowdhury M, Das T, et al. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. In: Proceedings of NSDI, San Jose, 2012. 2
- 14 He Y B, Tan H Y, Luo W M, et al. MR-DBSCAN: an efficient parallel density-based clustering algorithm using MapReduce. In: Proceedings of ICPADS, Tainan, 2011. 473–480
- 15 Chen W Y, Song Y Q, Bai H J, et al. Parallel spectral clustering in distributed systems. IEEE Trans Pattern Anal Mach Intell, 2011, 33: 568–586
- 16 Chopra S, Rao M R. The partition problem. Math Program, 1993, 59: 87–115
- 17 Nowozin S, Jegelka S. Solution stability in linear programming relaxations: graph partitioning and unsupervised learning. In: Proceedings of ICML, Montreal, 2009. 769–776
- 18 Vitaladevuni S N, Basri R. Co-clustering of image segments using convex optimization applied to EM neuronal reconstruction. In: Proceedings of CVPR, San Francisco, 2010. 2203–2210
- 19 Tsochantaridis I, Joachims T, Hofmann T, et al. Large margin methods for structured and interdependent output variables. J Mach Learn Res, 2005, 6: 1453–1484
- 20 Bagon S, Galun M. Large scale correlation clustering optimization. 2011. ArXiv: 1112.2903
- 21 Chierichetti F, Dalvi N, Kumar R. Correlation clustering in mapreduce. In: Proceedings of KDD, New York, 2014. 641–650
- 22 Blondel V D, Guillaume J L, Lambiotte R, et al. Fast unfolding of communities in large networks. J Stat Mech, 2008, 2008: P10008
- 23 Liu M Y, Tuzel O, Ramalingam S, et al. Entropy-rate clustering: cluster analysis via maximizing a submodular function subject to a matroid constraint. IEEE Trans Pattern Anal Mach Intell, 2014, 36: 99–112
- 24 Rodriguez A, Laio A. Clustering by fast search and find of density peaks. Science, 2014, 344: 1492–1496
- 25 Frey B J, Dueck D. Clustering by passing messages between data points. Science, 2007, 315: 972–976
- 26 Ng A Y, Jordan M I, Weiss Y. On spectral clustering: analysis and an algorithm. In: Proceedings of NIPS, Vancouver, 2002. 849–856
- 27 Bouguettaya A, Yu Q, Liu X M, et al. Efficient agglomerative hierarchical clustering. Expert Syst Appl, 2015, 42: 2785–2797
- 28 Fränti P, Sieranoja S. K-means properties on six clustering benchmark datasets. Appl Intell, 2018, 48: 4743–4759
- 29 King D E. Dlib-ml: a machine learning toolkit. J Mach Learn Res, 2009, 10: 1755–1758
- 30 Deng J K, Guo J, Xue N N, et al. Arcface: additive angular margin loss for deep face recognition. In: Proceedings of CVPR, Long Beach, 2019. 4690–4699

Maximum average entropy-rate based correlation clustering for big data

Liwen ZHANG¹, Tao WANG¹, Jian LUO¹, Shusen YANG² & Zongben XU^{2*}

1. *Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China;*

2. *School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China*

* Corresponding author. E-mail: zbxu@mail.xjtu.edu.cn

Abstract Clustering is one of fundamental tasks of data mining and machine learning. Due to the limitation of cluster assumption, lots of clustering algorithms perform poorly on some datasets against their assumptions, especially high-dimensional big data. This paper presents a maximum average entropy-rate based correlation clustering algorithm which is a kind of a graph-based correlation clustering. The objective function of original correlation clustering is decomposed into several single cluster optimizations and the limitation of big data in correlation clustering is removed by the neighboring connected graph. In algorithm implementation, the optimization of proper neighbor searching and correlation clustering are performed by heuristic neighbor searching and cluster generating respectively, and there is also an efficient graph-iterated implementation on distributed computation platform. Compared with other clustering algorithms, the proposed clustering algorithm is more flexible in cluster assumption, when accelerating the clustering process. In an experimental study we demonstrate the performance of the proposed algorithms on several datasets. The proposed clustering algorithm performed better than the other six clustering algorithms on the highest f1-measure and purity values, while its running time on high-dimensional big data is much lower than other clustering algorithms.

Keywords clustering, correlation clustering, entropy-rate, graph-based clustering, big data

ing on big data and clustering algorithms.

Liwen ZHANG received her B.S. and M.S. degrees in geographic information system and mapping technology from China University of Mining and Technology, Xuzhou, China, in 2012 and 2015, respectively. She is currently a Ph.D. candidate at the Faculty of Electronic and Information Engineering, Xi'an Jiaotong University. Her research interests include machine learning on big data and clustering algorithms.

Tao WANG received his B.S. degree in automation from the Faculty of Electronic and Information Engineering, Xi'an Jiaotong University in 2016. He is currently an M.S. candidate at the Faculty of Electronic and Information Engineering, Xi'an Jiaotong University. His research interests include machine learning on big data, clustering algorithms, and distributed system.

Jian LUO received his B.S. degree in software engineering from Nanchang Hangkong University, Nanchang, in 2017. He is currently pursuing his M.S. degree in software engineering at Xi'an Jiaotong University. His research interests include big data, machine learning, and distributed systems.

Zongben XU is a mathematician and an expert in signal and information processing. He received his Ph.D. degree in applied mathematics in 1987 from Xi'an Jiaotong University, China. He currently serves as a professor of mathematics and computer science, and the director of the Institute for Information and System Sciences Xi'an Jiaotong University. He is an academican of the Chinese Academy of Sciences and a member of the New York Academy of Sciences. His current research interests include intelligent information processing, machine learning, and theories in numerical modeling.