西安电子科技大学
XIDIAN UNIVERSITY

数据分析与挖掘

聚类综述

专业名称	软件工程
姓名	郑健
学号	15130110047
班级	1513011 云

2018 年 12 月

摘要

聚类是数据挖掘、模式识别等研究方向的重要研究内容之一,在识别数据的内在结构方面具有极其重要的作用.聚类主要应用于模式识别中的语音识别、字符识别等,机器学习中的聚类算法应用于图像分割和机器视觉,图像处理中聚类用于数据压缩和信息检索.聚类的另一个主要应用是数据挖掘(多关系数据挖掘)、时空数据库应用(GIS等)、序列和异类数据分析等.本文简要对聚类进行说明，并整理了常见的聚类算法，具体算法细节可详读“参考”出处。

1. 介绍

1.1 聚类概念

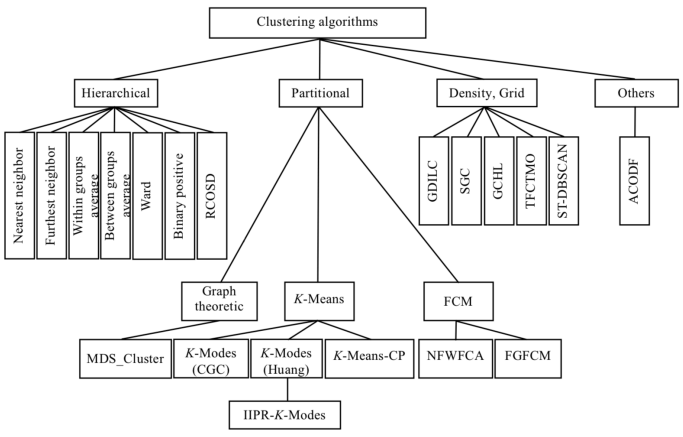
Everitt在 1974 年关于聚类所下的定义:一个类簇内的实体是相似的,不同类簇的实体是不相似的;一个类簇是测试空间中点的会聚,同一类簇的任意两个点间的距离小于不同类簇的任意两个点间的距离;类簇可以描述为一个包含密度相对较高的点集的多维空间中的连通区域,它们借助包含密度相对较低的点集的区域与其他区域(类簇)相分离.

聚类是一个无监督的分类,它没有任何先验知识可用.聚类过程如下:

1. 数据准备:包括特征标准化和降维.
2. 特征选择:从最初的特征中选择最有效的特征,并将其存储于向量中.
3. 特征提取:通过对所选择的特征进行转换形成新的突出特征.
4. 聚类(或分组):首先选择合适特征类型的某种距离函数(或构造新的距离函数)进行接近程度的度量;而后执行聚类或分组.
5. 聚类结果评估:是指对聚类结果进行评估.评估主要有 3 种:外部有效性评估、内部有效性评估和相关性测试评估.

1.2 算法分类

没有任何一种聚类技术(聚类算法)可以普遍适用于揭示各种多维数据集所呈现出来的多种多样的结构.根据数据在聚类中的积聚规则以及应用这些规则的方法,有多种聚类算法.聚类算法有多种分类方法,本文将聚类算法大致分成层次化聚类算法、划分式聚类算法、基于密度和网格的聚类算法和其他聚类算法.

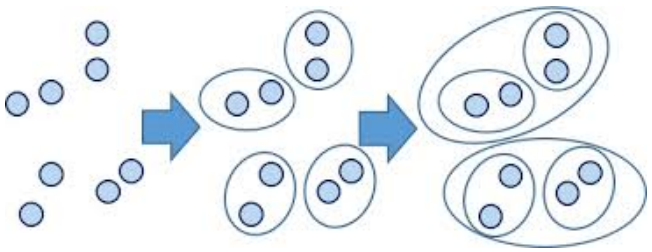


2. 聚类算法

2.1 层次聚类算法

层次聚类算法又称为树聚类算法,它使用数据的联接规则,透过一种层次架构方式,反复将数据进行分裂或聚合,以形成一个层次序列的聚类问题解.本文仅以层次聚类算法中的层次聚合算法为例进行介绍.层次聚合算法的计算复杂性为 $O(n^2)$,适合于`小型数据集`的分类.

2.1.1 层次`聚合`算法



该算法由树状结构的底部开始逐层向上进行聚合,假定样本集$S=o_1,o_2,\ldots,o_n$共有 n 个样本.

HA1[初始化]. 置每个样本 o_i 为一个类; /*共形成 n 个类:o1,o2,...,on*/ HA2[找最近的两个类]. $distance(o_r,o_k)=min_{\forall ou,ov\in S,ou\neq ov}distance(o_u,o_v)$; /*从现有的所有类中找出距离最近(相似度最大)的两个类 o_r 和 o_k*/ HA3[合并 o_r 和 o_k]. 将类 o_r 和 o_k 合并成一个新类 o_{rk}; /*现有的类数将减1*/ HA4. 若所有的样本都属于同一个类,则终止本算法;否则,返回步骤 HA2.

2.1.2 `传统`聚合规则

两个类之间距离的度量方法是传统层次聚合算法的重要组成部分,它主要包括两个重要参数`相似性度量方法`和`联接规则`.这里采用欧式距离作为相似性度量方法,联接规则主要包括单联接规则、完全联接规则、类间平均联接规则、类内平均联接规则和沃德法.这几种联接规则可定义如下 (其中,含$||x-y||$是欧几里德范数, n_i和 n_k 分别指类o_r 和o_k中的样本个数,$C(n_i+n_k,2)$表示从n_i+n_k个元素中抽出两个元素的不同组合的方法总数):

`单联接`聚合规则:$d(o_i,o_k)=min_{x\in o_i,y\in o_k}||x-y||$;

`全联接`聚合规则:$d(o_i,o_k)=max_{x\in o_i,y\in o_k}||x-y||$;

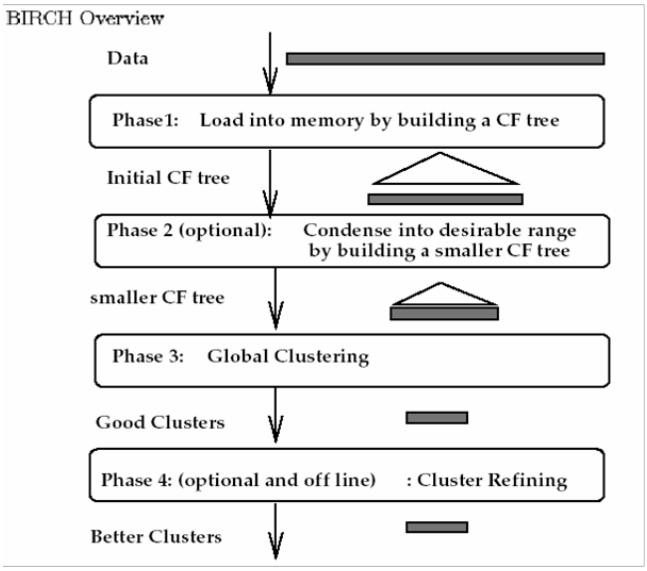
`类间`平均联接聚合规则:$d(o_i,o_k)=(1/n_in_k)\sum_{x\in o_i}(\sum_{y\in o_k}||x-y||)$;

`类内`平均联接聚合规则:$d(o_i,o_k)=(1/C(n_i+n_k,2))\sum_{x,y\in(o_i,o_k)}||x-y||$;

`沃德法`: $d(o_i,o_k)=(1/(n_i+n_k))\sum_{x\in(o_i,o_k)}||x-n||^2$,其中n 是融合聚类的中心.

2.1.3 BIRCH算法

BIRCH（Balanced Iterative Reducing and Clustering Using Hierarchies）主要是在数据体量很大的时候使用，而且数据类型是numerical。

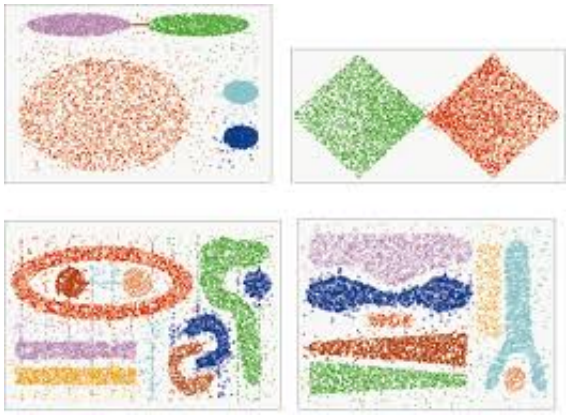


整个算法的实现分为四个阶段：

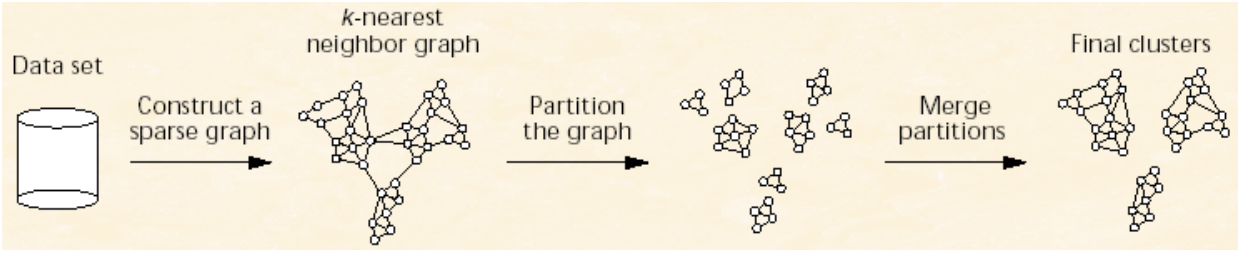
> （1）扫描所有数据，建立初始化的CF树，把稠密数据分成簇，稀疏数据作为孤立点对待
> （2）这个阶段是可选的，阶段3的全局或半全局聚类算法有着输入范围的要求，以达到速度与质量的要求，所以此阶段在阶段1的基础上，建立一个更小的CF树
> （3）补救由于输入顺序和页面大小带来的分裂，使用全局/半全局算法对全部叶节点进行聚类
> （4）这个阶段也是可选的，把阶段3的中心点作为种子，将数据点重新分配到最近的种子上，保证重复数据分到同一个簇中，同时添加簇标签

2.1.4 Chameleon算法

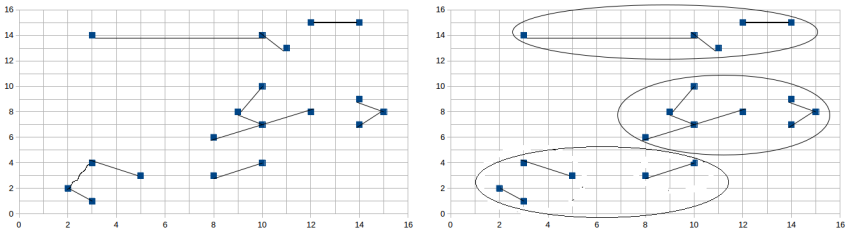
Chameleon（A Hierarchical Clustering Algorithm Using Dynamic Modeling）里用到的linkage是kNN（k-nearest-neighbor）算法，并以此构建一个graph，Chameleon的聚类效果被认为非常强大，比BIRCH好用，但运算复杂还是很高，$O(n^2)$。看个Chameleon的聚类效果图，其中一个颜色代表一类，可以看出来是可以处理非常复杂的形状的。



算法过程：



算法实现结果：(左图为第一阶段形成小簇集的结果，右图为第二阶段合并的结果)

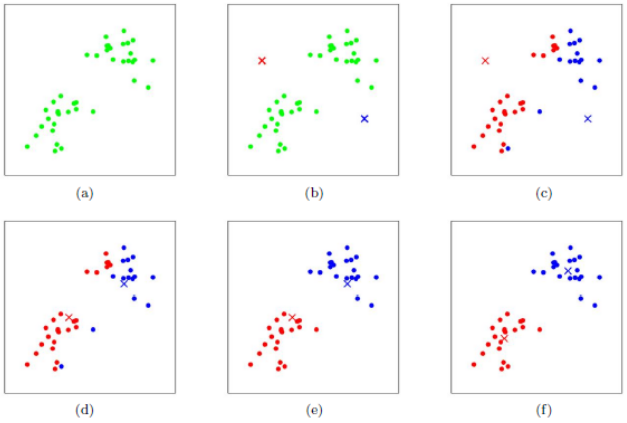


2.2 划分式聚类算法

其原理简单来说就是，想象你有一堆散点需要聚类，想要的聚类效果就是“类内的点都足够近，类间的点都足够远”。首先你要确定这堆散点最后聚成几类，然后挑选几个点作为初始中心点，再然后依据预先定好的启发式算法（heuristic algorithms）给数据点做迭代重置（iterative relocation），直到最后到达“类内的点都足够近，类间的点都足够远”的目标效果。也正是根据所谓的“启发式算法”，形成了k-means算法及其变体包括k-medoids、k-modes、k-medians、kernel k-means等算法。

2.2.1 K-Means聚类

1967 年,MacQueen 首次提出了 K 均值聚类算法(K-means 算法).迄今为止,很多聚类任务都选择该经典算法.该算法的核心思想是找出 K 个聚类中心 $c_1, c_2, \ldots, c_K$,使得每一个数据点 x_i 和与其最近的聚类中心 c_v 的平方距离和被最小化(该平方距离和被称为偏差 D).



K 均值(K-means)聚类算法(对 n 个样本进行聚类)

K1[初始化]. 随机指定 K 个聚类中心($c_1, c_2, \ldots, c_K$); K2[分配 x_i]. 对每一个样本 x_i,找到离它最近的聚类中心 c_v,并将其分配到 c_v 所标明类; K3[修正 c_w]. 将每一个 c_w 移动到其标明的类的中心; K4[计算偏差]. $D = \sum_{i=1}^{n}[min_{r=1,\ldots,K} d(x_i, c_r)^2]$; K5[D 收敛?]. 如果 D 值收敛,则 $return(c_1, c_2, \ldots, c_K)$ 并终止本算法;否则,返回步骤 K2.

优点: 能对 大型数据集 进行高效分类,其计算复杂性为 $O(tKmn)$,其中,t 为迭代次数,K 为聚类数,m 为特征属性数,n 为待分类的对象数,通常,K,m,t<<n.在对大型数据集聚类时,K-means 算法比层次聚类算法 快 得多.

缺点:通常会在获得一个 局部最优 值时终止;仅适合对数值型数据聚类;只适用于聚类结果为凸形(即类簇为凸形)的数据集.

- k-means对初始值的设置很敏感，所以有了k-means++、intelligent k-means、genetic k-means。
- k-means对噪声和离群值非常敏感，所以有了k-medoids和k-medians。
- k-means只用于numerical类型数据，不适用于categorical类型数据，所以k-modes。
- k-means不能解决非凸（non-convex）`数据，所以有了kernel k-means。

2.2.2 K-Modes 算法

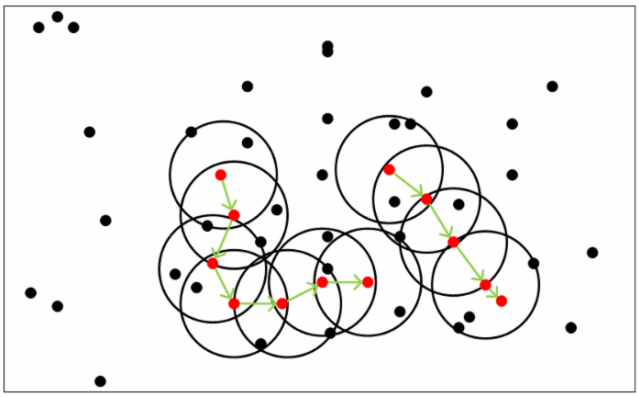
在阐述 K-modes 算法之前,先对 Means(均值) 与 Modes(众数) 做简单介绍. 在 K-means 算法中,mean 为类簇中心或称为质心,是指一个类簇中所有对象关于属性的均值,最初可随机指定.在 K-modes 算法中,modes 可定义如下:设 $X=X_1,X_2,\ldots,X_n$ 是一个数据集,$\forall X_i \in X$ 由 m 个 分类属性 $A_1,A_2,\ldots,A_m$ 来描述,X_i 可表示成向量 $\langle x_{i1},x_{i2},\ldots,x_{im}\rangle$,又可表示成属性-值对的合取 $[A_1=x_i1]\wedge\ldots\wedge[A_m=x_{im}]$;$Q$ 是 X 的一个 mode,Q 可表示成向量 $\langle q_1,q_2,\ldots,q_m\rangle$,也可表示成属性-值对的合取式 $[A_1=q_1]\wedge\ldots\wedge[A_m=q_m]$,$Q$ 需使 $\sum_{i=1,\ldots,n} d_1(X_i,Q)$ 取最小值,$d_1(X_i,Q)$ 表示 X_i 与 Q 之间的距离, Q 不必是 X 的一个元素. 1998 年,Huang 为克服 K-means 算法仅适合于数值属性数据聚类的局限性,提出了一种适合于分类属性数据聚类的 K-modes 算法.该算法对 K-means 进行了 3 点扩展:引入了处理分类对象的 新的相异性度量方法 (简单的相异性度量匹配模式),使用 `modes` 代替 `means`,并在聚类过程中使用基于频度的方法 修正 `modes`,以使聚类代价函数值最小化.

这些扩展允许人们能够直接使用 K-means 范例聚类有分类属性的数据,无须对数据进行变换.K-modes 算法的另一个优点是 modes 能给出类的特性描述,这对聚类结果的解释是非常重要的.事实上,K-modes 算法比 K-means 算法能够 更快收敛 .与 K-means 算法一样,K-modes 算法也会产生 局部最优解 ,依赖于初始化 modes 的选择和数据集中数据对象的次序.1999 年,Huang 等人证明了经过有限次迭代 K-modes 算法仅能收敛于局部最小值.

2.3 基于密度的聚类算法

k-means解决不了不规则形状的聚类。于是就有了Density-based methods来系统解决这个问题。该方法同时也对噪声数据的处理比较好。其原理简单说画圈儿，其中要定义两个参数，一个是圈儿的最大半径，一个是一个圈儿里最少应容纳几个点。只要邻近区域的密度（对象或数据点的数目）超过某个阈值，就继续聚类,最后在一个圈里的，就是一个类。

2.3.1 DBSCAN算法



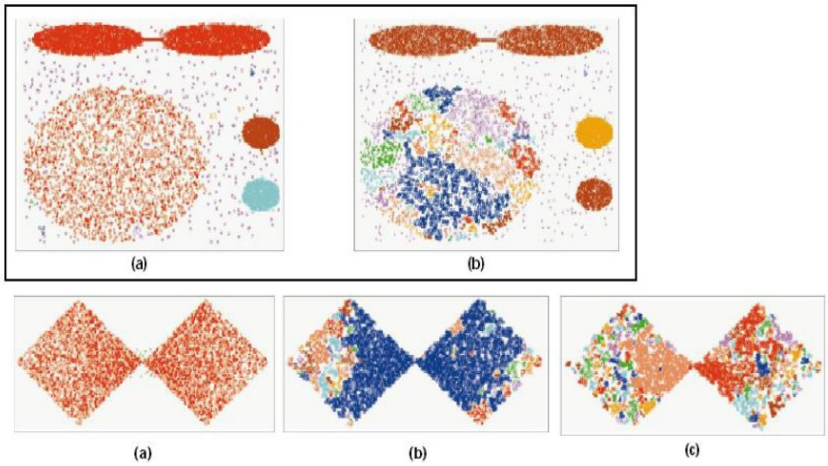
算法流程：

1. 从任一对象点p开始；
2. 寻找并合并核心p对象直接密度可达的对象；

3. 如果p是一个核心点，则找到了一个聚类，如果p是一个边界点(即从p没有密度可达的点)则寻找下一个对象点；
4. 重复2、3，直到所有点都被处理

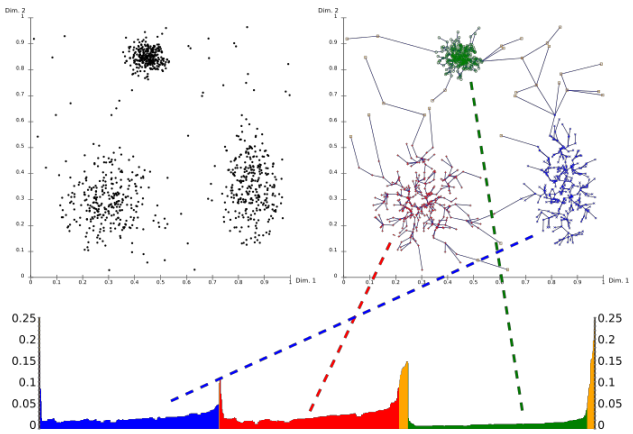
优点：对噪声不敏感；能发现任意形状的聚类。

缺点：聚类的结果与`参数`有很大的关系；DBSCAN用固定参数识别聚类，但当聚类的稀疏程度不同时，相同的判定标准`可能会破坏聚类的自然结构`，即较稀的聚类会被划分为多个类或密度较大且离得较近的类会被合并成一个聚类。下图就是表现了DBSCAN对参数设置的敏感。



2.3.2 OPTICS算法

OPTICS聚类算法是基于密度的聚类算法，全称是Ordering points to identify the clustering structure，目标是将空间中的数据按照密度分布进行聚类，其思想和DBSCAN非常类似，但是和DBSCAN不同的是，OPTICS算法可以获得不同密度的聚类，直接说就是经过OPTICS算法的处理，理论上可以获得任意密度的聚类。因为OPTICS算法输出的是样本的一个有序队列，从这个队列里面可以获得任意密度的聚类。



计算过程为：

1、从结果队列中按顺序取出点，如果该点的可达距离不大于给定半径ϵ，则该点属于当前类别，否则至步骤2；
2、如果该点的核心距离大于给定半径ϵ，则该点为噪声，可以忽略，否则该点属于新的聚类，跳至步骤1；
3、结果队列遍历结束，则算法结束。

2.4 基于网络的方法

将数据空间划分为`网格单元`，将数据对象集映射到网格单元中，并计算每个单元的密度。根据预设的阈值判断每个网格单元是否为高密度单元，由邻近的稠密单元组形成”类“。

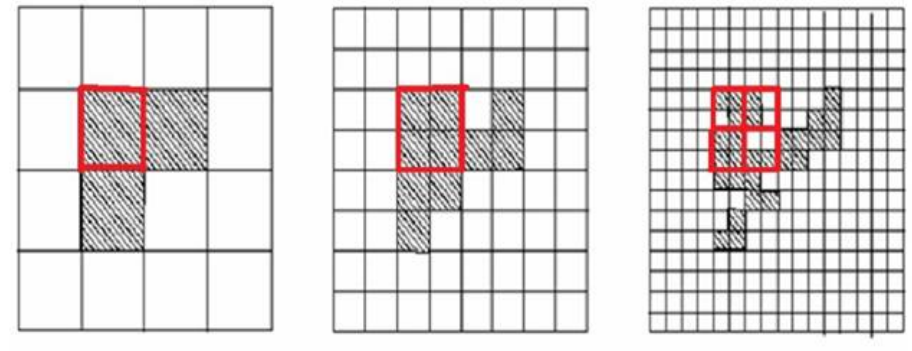
这些算法用不同的网格划分方法，将数据空间划分成为有限个单元（cell）的网格结构,并对网格数据结构进行了不同的处理，但核心步骤是相同的： 1、 `划分网格` 2、 使用网格单元内数据的统计信息对数据进行`压缩表达` 3、 基于这些统计信息`判断`高密度网格单元 4、 最后将相连的高密度网格单元`识别`为簇

优点：速度很快，因为其速度与数据对象的个数无关，而只依赖于数据空间中每个维上单元的个数。

缺点：参数敏感、无法处理不规则分布的数据、维数灾难等；这种算法效率的提高是以聚类结果的精确性为代价的。经常与基于密度的算法结合使用。

2.4.1 STING算法

STING（STatistical INformation Grid）基于网格多分辨率，将空间划分为方形单元，对应不同分辨率

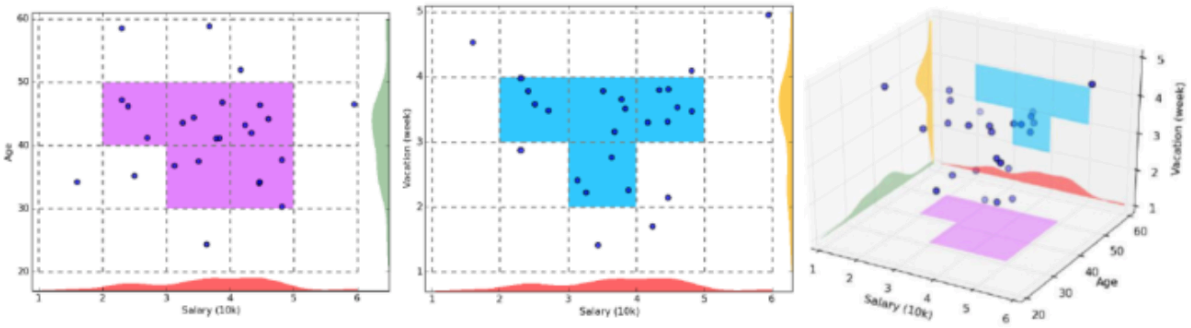


a cell of (i-1)th level corresponds to 4 cells of (i)th level

STING算法的核心思想：首先我们先划分一些层次，每个层次上我们根据维度或者概念分层不同的cell，实际上这里的每个层次对应的是样本的一个分辨率。每个高层的cell在其下一层中被对应得划分成多个cell，每个cell我们都计算出它的统计信息，估计出它的分布。利用这样的结构，我们很容易进行查询，比如我们查询具有某些属性的样本，我们从上到下开始，根据cell的统计信息计算query在每个cell的置信区间，找出最大的那个cell，然后到下一层，依次直至到最底层。这样的好处是，我们不用计算所有的样本，算法每进一层都会抛弃不相关的样本，所需的计算量会越来越少，那么速度就会很快。

2.4.2 CLIQUE算法

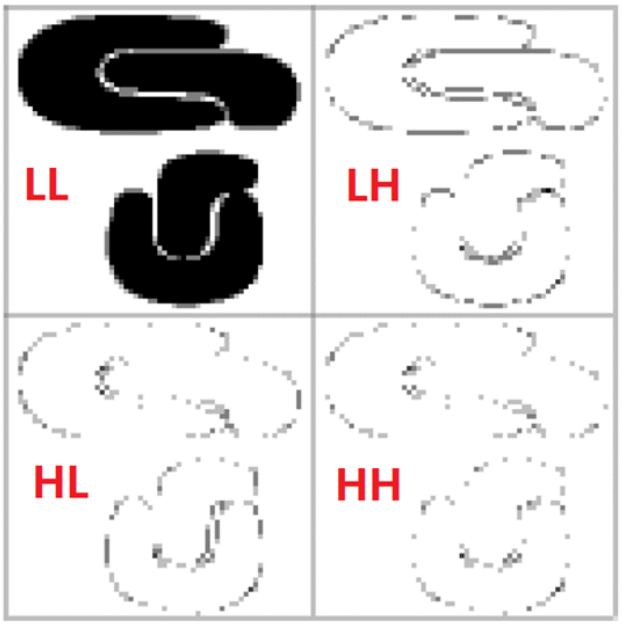
CLIQUE（CLustering In QUEst）结合网格和密度聚类的思想，子空间聚类处理大规模高维度数据



CLIQUE算法核心思想：首先扫描所有网格。当发现第一个密集网格时，便以该网格开始扩展，扩展原则是若一个网格与已知密集区域内的网格邻接并且其其自身也是密集的，则将该网格加入到该密集区域中，知道不再有这样的网格被发现为止。（密集网格合并） 算法再继续扫描网格并重复上述过程，直到所有网格被遍历。以自动地发现最高维的子空间，高密度聚类存在于这些子空间中，并且对元组的输入顺序不敏感，无需假设任何规范的数据分布，它随输入数据的大小线性地扩展，当数据的维数增加时具有良好的可伸缩性。

2.4.3 WAVE-CLUSTER算法

WaveCluster：用小波分析使簇的边界变得更加清晰



WaveCluster算法的核心思想是将数据空间划分为网格后，对此网格数据结构进行小波变换，然后将变换后的空间中的高密度区域识别为簇。基于数据点数目大于网格单元数目（N≥K）的假设，WaveCluster的时间复杂度为O(N)，其中N为数据集内数据点数目，K为网格内的网格单元数目。

2.5 其他聚类

2.5.1 基于约束的方法

真实世界中的聚类问题往往是具备多种约束条件的，然而由于在处理过程中不能准确表达相应的约束条件、不能很好地利用约束知识进行推理以及不能有效利用动态的约束条件，使得这一方法无法得到广泛的推广和应用。这里的约束可以是对个体对象的约束，也可以是对聚类参数的约束，它们均来自相关领域的经验知识。该方法的一个重要应用在于对存在障碍数据的二维空间数据进行聚类。

$COD(ClusteringwithOb2structedDistance)$就是处理这类问题的典型算法，其主要思想是`用两点之间的障碍距离取代了一般的欧氏距离来计算其间的最小距离`。

2.5.2 量子聚类

受物理学中量子机理和特性启发，可以用量子理论解决聚类记过依赖于初值和需要指定类别数的问题。一个很好的例子就是**基于相关点的 $Pott$ 自旋和统计机理提出的量子聚类模型**。它把聚类问题看做一个物理系统。并且许多算例表明，对于传统聚类算法无能为力的几种聚类问题，该算法都得到了比较满意的结果。

2.5.3 核聚类

核聚类方法增加了对样本特征的优化过程，`利用 Mercer 核 把输入空间的样本映射到高维特征空间`，并在特征空间中进行聚类。核聚类方法是普适的，并在性能上优于经典的聚类算法，它通过非线性映射能够较好地分辨、提取并放大有用的特征，从而实现更为准确的聚类；同时，算法的收敛速度也较快。在经典聚类算法失效的情况下，核聚类算法仍能够得到正确的聚类。代表算法有SVDD算法，SVC算法。

2.5.4 谱聚类

首先根据给定的样本数据集定义一个描述成对数据点相似度的`亲合矩阵`,并计算矩阵的特征值和特征向量,然后选择合适的特征向量聚类不同的数据点。谱聚类算法最初用于**计算机视觉、VLSI设计**等领域,最近才开始用于机器学习中,并迅速成为国际上机器学习领域的研究热点。

谱聚类算法建立在图论中的谱图理论基础上,其本质是**将聚类问题转化为图的最优划分问题**,是一种点对聚类算法。

3. 方法扩展

3.1 数据简化方法

3.1.1 变换（Data Transformation）

离散傅里叶变换（Discrete Fourier Transformation）可以提取数据的`频域（frequency domain）信息`，离散小波变换（Discrete Wavelet Transformation）除了频域之外，还可以提取到`时域（temporal domain）信息`。

3.1.2 降维（Dimensionality Reduction）

在降维的方法中，`PCA（Principle Component Analysis）`和`SVD（Singular Value Decomposition）`作为线性方法，受到最广泛的应用。处理非线性降维的算法主要是`流形学习（Manifold Learning）`。关于降维在聚类中的应用，最著名的应该就是谱聚类（Spectral Clustering）效果通常比k-means好，计算复杂度还低，这都要归功于降维的作用。

3.1.3 抽样（Sampling）

最常用的就是`随机抽样（Random Sampling）`，如果你的数据集特别大，随机抽样就越能显示出它的低复杂性所带来的好处。比如CLARA（Clustering LARge Applications）就是因为k-medoids应对不了大规模的数据集，所以采用sampling的方法。一般抽样法采用的不多。

3.2 相似性衡量

3.2.1 距离

距离主要就是指Minkovski距离。这个名字虽然听起来陌生，但其算法就是Lp norm的算法，如果是L1 norm，那就是绝对值/曼哈顿距离（Manhattan distance）；如果是L2 norm，那就是著名的欧式距离（Euclidean distance）了，也是应用最广泛的；如果L无穷范式，supremum距离，好像也有叫切比雪夫距离的，但就很少有人用了。另外，还有Mahalanobis距离，目前来看主要应用于Gaussian Mixture Model（GMM），还有Lance&Williams距离等等。

3.2.2 相似系数

主要有夹角余弦和相关系数。相关系数的应用也非常广泛，其主要优势是它不受原线性变换的影响，而且可以轻松地转换为距离，但其运算速度要比距离法慢得多，当维数很高的时候。

3.2.3 核函数K(x,y)

定义在RdXRd上的二元函数，本质上也是反映x和y的距离。核函数的功能就是把数据从低维空间投影（project）到高维空间去。

3.2.4 DTW（dynamic time warping）

这是一种非常特殊的距离算法，它可以计算两个不同长度的向量的距离，也可以对两对向量中不同时间段内的数据做匹配。DTW主要用在时间序列的部分场合里。

4. 衡量聚类算法优劣的标准

不同聚类算法有不同的优劣和不同的适用条件。大致上从跟数据的属性（是否序列输入、维度），算法模型的预设，模型的处理能力上看。具体如下： 1、**算法的处理能力**：处理大的数据集的能力（即算法复杂度）；处理数据噪声的能力；处理任意形状，包括有间隙的嵌套的数据的能力； 2、**算法是否需要预设条件**：是否需要预先知道聚类个数，是否需要用户给出领域知识； 3、**算法的数据输入属性**：算法处理的结果与数据输入的顺序是否相关，也就是说算法是否独立于数据输入顺序；算法处理有很多属性数据的能力，也就是对数据维数是否敏感，对数据的类型有无要求。

参考

- 用于数据挖掘的聚类算法有哪些，各有何优势? https://www.zhihu.com/question/34554321
- 各种聚类算法的系统介绍和比较 https://blog.csdn.net/abc200941410128/article/details/78541273
- k-modes聚类算法介绍 https://blog.csdn.net/tyh70537/article/details/78158674
- BIRCH算法学习 https://blog.csdn.net/qll125596718/article/details/6895291
- Chameleon两阶段聚类算法 https://blog.csdn.net/Androidlushangderen/article/details/44569077
- DBSCAN密度聚类算法 https://www.cnblogs.com/pinard/p/6208966.html
- OPTICS聚类算法原理 https://blog.csdn.net/xuanyuansen/article/details/49471807
- 机器学习：基于网格的聚类算法 https://cloud.tencent.com/developer/article/1005263
- 谱聚类（spectral clustering）原理总结 https://www.cnblogs.com/pinard/p/6221564.html