聚类分析

于 剑 肖 宇
北京交通大学

关键词： 聚类算法 类的复杂性

引言

“分而治之”历来是人类处理复杂问题的重要手段，也是处理海量数据的有效方式。如何将数据集分而治之（即将一个数据集划分成一些子数据集），是值得研究的基础性问题。

人们通常期望数据划分后形成的子集内样本具有同质性，即类内的样本是相似的，不同类之间的样本是不相似的。这就是所谓的聚类分析或分群技术。该名词出自中国的一句老话“物以类聚，人以群分”。

与分类技术不同，聚类不要求对数据进行事先标定。机器学习中非常重要的无监督学习的一个主要方法就是在数据的分类结构未知时，利用聚类分析期望能够发现数据集中自身隐藏的内蕴结构信息。聚类分析源于许多研究领域，受到很多应用需求的推动。例如，在复杂网络分析中，人们希望发现具有内在紧密联系的社团；在图像分析中，人们希望将图像分割成具有类似性质的区域；在文本处理中，人们希望发现具有相同主题的文本子集；在有损编码技术中，人们希望找到信息损失最小的编码；在顾客行为分析中，人们希望发现消费方式类似的顾客群，以便制订有针对性的客户管理方式和提高营销效率。这些情况都可以在适当的条件下归为聚类分析。

目前，文献中有关聚类的综述已经非常专业。本文的主要目的是提供一个包含聚类分析最新发展技术的快速导读，以引起有关人士的兴趣。

聚类的基本步骤和相关术语

对聚类分析的基本要求是每个样本只属于一个类别，并且只属于与其相似性最大的类别，因而相似性的定义和计算对于聚类分析来说至关重要。但是由于相似性的定义与需求密切相关，而需求在各种情形下各不相同，所以非常可惜，至今文献中都没有相似性的统一定义。既然如此，人们就将相似性用相异性来表示，相异性越大，相似性越小，反之亦然。

模式识别中的丑小鸭定理[1]明确指出，如果不选取合适的特征，则丑小鸭与白天鹅之间的相似性在逻辑上与白天鹅之间的相似性一样。因而，选取特征与任务相关。比如，在生物学上，鲸鱼与牛同属于哺乳类偶蹄目。但是，按产业界划分，鲸属于水产业，牛属于畜牧业，两者完全不是一类。一个形象的例子可以参见图1，如果按形状分类，则可以分为三角形和方形两类；如果按颜色分类，可以分为红、蓝、黄三类。因此，如何选择特征和定义相似性，通常与人们聚类的目的和标准直接相关。如何定义任务相关的相似性始终是聚类分析的重要任务。

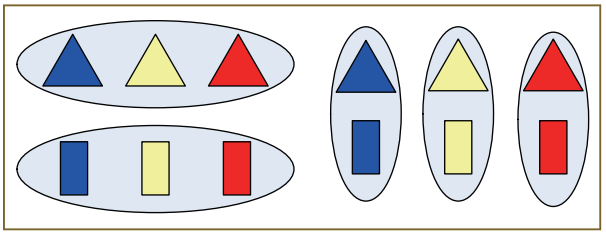

图1 聚类标准与聚类结果

在论述具体的聚类算法之前，本文先介绍与聚类分析相关的一些术语和数学表示方法。

样本 指要进行聚类的数据集中的单个数据。样本一般是一个多维向量。向量的每个分量是数值型或名词型的数据，一般称其为特征或者属性。特征或属性有时也称作变量。

样本集 或称数据集，是由单个样本所组成的集合，即是需要聚类操作的数据整体。数据集通常表示成一个矩阵。

相异矩阵 由矩阵中的元素表示样本集中每对样本之间的相异程度，一般是非负值。

相似矩阵 由矩阵中的元素表示样本集中每对样本之间的相似程度，一般是非负值。

类 指通过聚类而形成的一组数据样本。具体而言，同一类中的样本都具有相似的特征。通常用K或C表示要聚成多少个类，用$C_1, C_2, \cdots, C_K$表示这些类。

类原型 能够代表某个类性质的数据元，可以是样本集中某一个或者多个具体样本，可以是某部分样本的一个或者多个加权值，也可以是能描述一个类特征的向量。

划分矩阵 矩阵中的每个元素表示每个样本属于样本集划分后的各个类的情况，如果第i个样本属于第k类，则$p(i,k)=1$，否则$p(i,k)=0$。

模糊划分（软划分） 用于表示每个样本属于样本集划分后的各个类的模糊隶属度（条件概率），通常用U表示，$U=[u(i,k)]_{n\times K}$，其中$0 \le u(i,k) \le 1$。

类标 用于表示每个样本属于样本集划分后的各个类的情形，通常用I表示。$I \in \{1,2,\cdots,K\}$，其中，如果第i个样本属于第k类，则$I(i)=k$。

因此，聚类可简单描述为：给定一个数据集$X=\{x_1, x_2, \cdots, x_n\}$，或者一个相似矩阵$S$，将其划分为$k$个相似的子集类，其中$C_1, C_2, \cdots, C_K$，$C_i \subseteq X$，且$\bigcup_{i=1}^{K} C_i = X$，$\forall i \neq j, C_i \cap C_j = \varnothing$，$C_i \neq \varnothing$。

典型的数据聚类的基本步骤如下：

1. 对数据集进行表示和预处理，包括数据清洗、特征选择或特征抽取；

2. 给定数据之间的相似度或相异度及其定义方法；

3. 根据相似度，对数据进行划分，即聚类；

4. 对聚类结果进行评估。

本文将重点介绍第3步，即聚类过程，其他部分从略。

聚类算法的基本分类

对类的定义是根据任务而变化的，这决定了聚类算法的多姿多彩。聚类算法可以有多种不同的分类方式，比如文献[2 ~ 3]的分类方式。限于篇幅，本文没有采用文献[2 ~ 3]的方式，而是根据聚类结果的不同表示，将聚类算法分为类原型聚类算法、层次聚类法、连通型聚类算法和划分矩阵型聚类算法。

类原型聚类算法

类原型聚类算法是指每个类可以由类原型来代表。主要目的是对原来的数据集进行压缩编码，从而获得比较简略的描述。文献中常见的类原型有点、超平面和超球等等。根据类原型的不同，可以将算法分为点原型聚类算法、超平面原型聚类算法和超球原型聚类算法等等。

点原型聚类算法 指每个类用一个点来代表，在这三种算法中是文献中研究最多的。如果进一步细分，又可以将其分为虚拟样本点原型聚类算法和样本点原型聚类算法。虚拟样本点原型聚类算法使用的点一般不能保证是样本集中的样本（如图2所示），而样本点原型聚类算法使用的点一定是样本集中的样本。

虚拟样本点原型聚类算法 其基本思想是用类原型来代替每个类中的样本时误差最小，关键部分是定义合适的误差函数，函数

的定义不同，得到的聚类算法就不同。如果误差函数为平方误差函数，则可以导出由麦克奎恩（J. MacQueen）提出的C-均值聚类算法。当使用其他的误差函数时，则可以分别得到模糊C-均值算法（Fuzzy C Mean，FCM）、决定性退火聚类算法（Deterministic Annealing）、均值漂移（Mean Shift）、可能性C-均值聚类（Possibilistic C-Means，PCM）、条件模糊C-均值算法、加权C-均值和期望最大化（Expectation Maximization，EM）聚类算法等变型。文献[4]设计了一种比较普适的误差函数，可以证明上述各种算法都是一些特例。当然，有些文献没有使用误差函数而设计出虚拟样本点原型聚类算法，典型的如自组织映射聚类算法（Self-Organizing Map）等等。

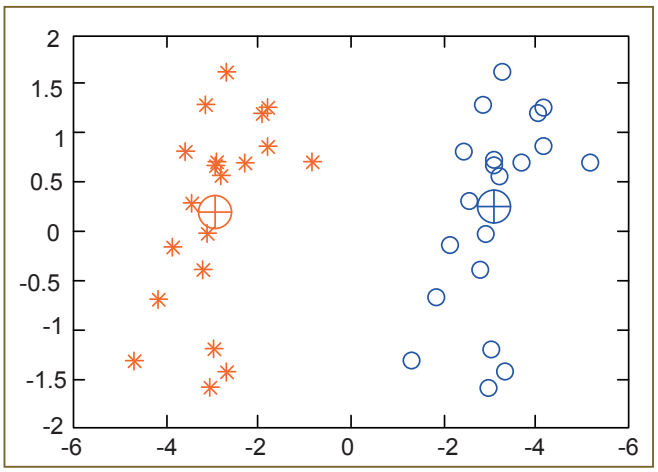

图2 C-均值聚类算法的聚类结果（其中○与*代表真正的样本点，⊕代表类中心。图中的类中心显然是虚拟样本点。）

样本点原型聚类算法 其基本思想是用类中的某个具体样本来代替类中其他样本时的误差最小。误差函数不同时，导出的聚类算法也不同。常见的有基于K-中心点的PAM（Partitioning Around Medoids）算法以及基于选择的CLARA（Clustering Large Applications，大规模应用聚类）、CLARANS（Clustering Large Applications based on RANdomized Search，基于随机搜索的大规模应用聚类）。近年提出的近邻传播聚类算法（Affinity Propagation）[5]是样本点原型聚类算法的一个变形。

超平面原型聚类算法 其类原型是超平面，其他的思路与点原型聚类算法类似。文献报道过的算法有k-平面聚类（k-Plane Clustering，kPC）、模糊k-平面聚类（Fuzzy k-Plane Clustering，FkPC）[6]和GK（Gustafson-Kessel）算法。

超球原型聚类算法 其类原型是超球，文献报道过的算法包括模糊核聚类（Fuzzy C-Shell）和“Fuzzy C-Ellipsoidal Shells”等。

总之，类原型聚类算法的实现简单，收敛速度也很快。但是，在判断数据具有类原型表示的聚类结构方面，该类算法需要很多的先验知识。另外，这类算法对“噪声”和孤立点数据比较敏感。

当聚类结构明显与类原型匹配时，类原型聚类算法能够得到很好的聚类效果。但是，很多情形下，数据的聚类结构在原来的特征空间并不与类原型匹配。此时，文献中有两种思路：一种是设计一个合适的映射，将数据映射到具有明显类原型结构的新特征空间中聚类，核映射下的聚类算法就是这样一种思想的体现。文献中的谱聚类算法如正则化割（Normalized Cut）等就是一种核映射下的点原型聚类算法[7]；另一种思路则考虑到在高维空间情况下，因样本之间的距离几乎相同，使得通常依据距离的聚类算法无法使用，而这时不同的子类实际上处于不同的特征子空间里，利用此特性开发出了子空间聚类（Subspace Clustering）算法。如果将不同的子空间看作类原型，子空间聚类也可以看作是类原型聚类的一种。有关子空间聚类，更详细的内容可以参考文献[8]。

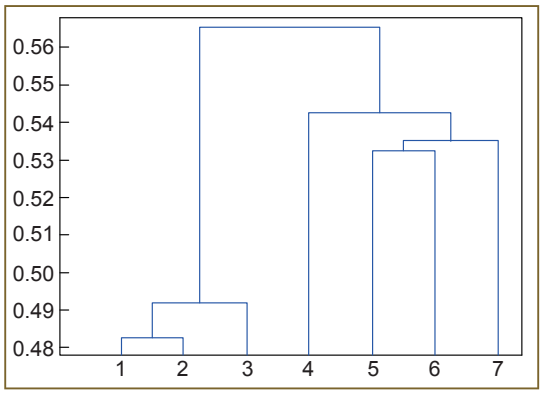

图3 聚类的树状表示

层次聚类算法

如果聚类结果用一个树状结构来表示（如图3），就可以得到层次聚类算法。层次聚类算法的设计可以从两个方向上进行，一是合并相似度较高的单个样本或类；二是分裂相异度较高的样本或类，即从合并和分裂的角度来逐步完成对样本的聚类操作。这两种算法的聚类过程可以用分层的树状图来表示，因此分别称为凝聚的层次聚类算法和分裂的层次聚类算法。

凝聚的层次聚类 这种方法采用的是自底向上的策略。首先将样本集的每个样本作为一个类，然后根据相似度的大小对某些类进行合并，形成新的较大的类，不断重复这一过程，直到所有的样本都属于同一个类，或者某个终结条件满足时结束。绝大部分的层次聚类算法都属于凝聚的层次聚类，区别在于类间相似度和距离的定义不同。如果类与类之间距离定义为类中样本之间的最小距离值，就可以导出单连接（Single-linkage）算法；如果类与类之间距离定义为类中样本之间的最大距离值，就可以导出完全连接（Complete-linkage）算法。类似的算法还有组平均连接（Average-Linkage）、质心连接（Centroid-Linkage）和中值连接（Median-Linkage）等算法。沃德（J. H. Ward）在1963年提出来的沃德层次聚类可以导出上述算法。

分裂的层次聚类 与凝聚的层次聚类正相反，分裂的层次聚类是一种自顶向下的策略。它首先将整个样本集看作为一个类，然后逐渐将较大的类分裂为较小的类，重复这一过程直到每个样本都变成一个类，或者达到了某个终结条件为止。相对而言，此类算法较少。在复杂网络的社区发现问题研究中，著名的GN（Girvan and Newman）[9]算法是一个分裂的层次聚类算法。该算法最重要的部分是定义了无向图上边介数的概念（Edge Betweenness）（边介数是指图中通过该边的最短路径的条数），通过依次删去图上具有最高边介数的边，直至最后每个连通分支中只有一个顶点。

层次聚类算法的思想比较简单，但也存在一定的缺点。首先该算法的时间和空间复杂度都是$O(n^2)$（n为样本的个数）；其次层次聚类是按照合并或分裂的次序进行的，具有不可逆转性和不可更改性，因而一旦某一步合并或分裂选择的不恰当，就会影响下一步的操作，直到影响到最终的聚类效果。

上述介绍的都是层次聚类中基本的算法，没有考虑大型数据库的可伸缩性等问题。对于大型数据库和数据挖掘等问题中的应用，人们提出了许多改进的层次聚类算法，主要有BIRCH（Balanced Iterative Reducing and Clustering using Hierarchies）、CURE（Clustering Using REprisentatives）、COBWEB、ROCK和CHAMELEON等。具体介绍可以参看文献[3]。

连通型聚类算法

对于使用图的连通分支来表示类的聚类算法，本文统称为连通型聚类算法。因此，连通性聚类算法与图密切相关。连通聚类算法中最典型的是最小生成树（Minimum Spanning Tree，MST）算法。其过程是，首先根据样本集的相异（似）矩阵构造一个具有n个顶点全联接图，其顶点之间边的权值大小表示样本之

间相似度或相异度的大小；然后计算此完全图的最小（大）生成树；最后删去不合适的边，得到的连通分支就是最终的聚类结果。可以证明，最小生成树的建立过程与单连接算法的合并过程完全一致。因此，可以认为最小生成树算法是单连接算法的变形。

在连通型聚类算法中还有最小切聚类算法（Minimum Cut）、DBSCAN（Density-Based Spatial Clustering of Application with Noise）和DENCLUE（DENsity based CLUstEring，基于密度分布函数的聚类）等。当使用相似（异）矩阵时，最小切聚类算法是删去不同连通分支之间具有最小（大）权重的连接边。虽然谱聚类算法是最小切的修正，但是修正之后的计算复杂度已经是NP问题，因此必须应用近似计算，但是这种近似运算已经不能用图的连通性来解释，因此，谱聚类一般不能归为连通型聚类算法。

DBSCAN是一个基于高密度连通区域的密度聚类算法。其思想是寻找具有足够高密度的连通区域划分作为类，而低密度区域的点作为孤立点。在这个算法中，***n***个顶点代表***n***个样本，每个顶点只与其距离小于ε的顶点有边相连，并且核心顶点的度数不能低于一个阈值MinPts。如果只考虑图上的核心顶点，它们之间的连通分支个数即为聚类数，这样得到的连通分支上的点加上其上的ε-邻域就形成了类。不在这些类内的点称为野值。采用空间索引或R*-树等技术，可以提高DBSCAN算法的效率，其计算复杂度为***O(nlogn)***。实验也反映了算法超线性的运行时间，可以将算法应用于大型的数据库中。但DBSCAN算法对参数ε和MinPts比较敏感，需要用户慎重选择。

OPTICS算法是对DBSCAN算法的一种扩展，其改进之处在于降低了算法对参数的敏感程度。OPTICS算法的思想是考虑到在DBSCAN聚类时，会发生更高密度的区域被混杂在较低密度的区域里，并出现在同一个聚类中的情况。因此，提出应先完成对更高密度区域的聚类，再选择较低密度的区域进行聚类。也就是说，对于恒定的MinPts值，高密度区域的ε值较小，因此，需要先选择最小的ε值所对应的样本进行聚类，然后再逐步增大ε值。OPTICS算法并没有给出一个真正的聚类结果，而是创建了对数据集进行聚类的次序，再利用其他算法根据次序信息来抽取聚类。对于低维的数据集，这个次序可以用图形的方式来帮助理解。OPTICS算法与DBSCAN具有相同的时间复杂度***O(nlogn)***。

DENCLUE也是一个基于密度分布的聚类算法。该算法的主要思想是，先为每个数据点定义一个影响函数，用于描述数据点在其邻域内的影响；然后构建数据空间的整体密度模型；最后用求得全局密度函数的密度吸引点来进行聚类。利用密度函数，可以定义该函数的梯度和密度吸引点。通常密度吸引点是指全局密度函数在该点取得局部最大值的点。该算法还形式化定义了关于密度吸引点的中心定义的类和任意形状的类。在类内密度比较均匀的情况下，DENCLUE算法能够检测任意形状的类。同时，该算法使用网格单元来保存只包含实际数据点的网格单元信息，并用树的存储结构来处理这些单元，因此提高了算法速度，其计算复杂度为***O(n)***。但是，DENCLUE算法的结果受密度参数δ和噪声阈值E的影响较大，需要仔细选择。

QT（Quality Threshold，质量阈值）[10]聚类算法是通过限定类内两样本的最短距离来聚类的。其主要思想是，如果定义了相异矩阵对应的图，则类内任意两顶点的最短路径长度不大于给定的阈值。因此，该算法也可以用连通分支来表示，只是限定了连通分支内任意两点间最短路径的最大长度或者类直径而已。其流程也非常简单，选定一个样本，逐渐合并与其最相似的样本，直到再增加的样本导致类内样

本间最短路径超过给定的阈值，然后选定下一个样本，重新聚类。

划分矩阵型聚类算法

如果用划分矩阵（包括软划分）来表示类，则发展出来的算法可以称为划分矩阵型聚类算法。目前常见的算法有三种：第一种是基于矩阵分解技术的方法，其算法的输入是相似矩阵，计算的主要依据是将相似矩阵分解成划分矩阵乘积的形式。基于非负矩阵分解的聚类算法[11]和异质聚类算法中的块值分解聚类算法[12]属于此类。可加性聚类算法（Additive Clustering）也可勉强归为此类；第二种是基于信息论的方法，其算法的输入是概率分布矩阵。文献中的算法有信息瓶颈（Information Bottleneck）聚类算法[13]和异质聚类算法的互信息联合（Mutual Information Clustering）聚类算法[14]；第三种是基于间隔（Margin）理论直接对数据进行无监督标定的方法。现有的方法有支持向量机聚类算法（Support Vector Clustering）和最大间隔聚类算法（Maximum Margin Clustering）。由于类标与划分矩阵等价，这类算法也归为划分矩阵聚类算法。

这类算法的主要缺点是可理解性和直观性比其他三种类型的算法要差，算法容易受到初始值设置的影响。但是这些算法有的具有强烈的应用背景，比如NMF（Non-negative Matrix Factorization）聚类和异质聚类的互信息联合聚类算法在文本聚类、基因数据分析等方面的应用；有的具有很好的理论基础，比如支持向量机聚类算法和最大间隔聚类算法。

类的复杂性讨论

通过上述介绍可知，定义类是一件非常复杂的事情。文献[15]曾经讨论了相切类的简单情形，并由此修正了最小生成树聚类算法。但是实际数据中的类的复杂度更高。下面将讨论四种常见的类，借以说明聚类任务具有的挑战性。

任意形状类 指的是不能用一个简单类原型描述的类。它在几何上的形状是任意的，如图4所示。如果各个类都是同质的，并且类内的密度大致均匀，则在参数设置较好的情形下，连通型聚类算法可以很好地处理这种类。

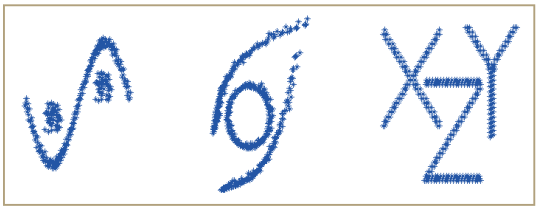

图4 任意形状类

非同质类 有两种定义。弱定义是指不同类内样本的密度不相同，如图5中的左图所示。强定义是指存在一个类，其类内的密度相似度变化很大，如图5的右图所示。

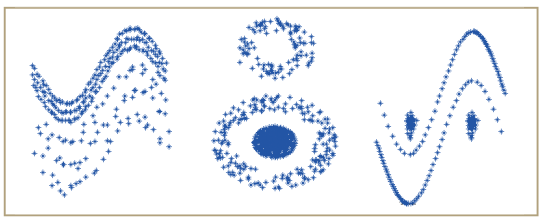

图5 非同质类

相切类 也有两种定义。密度定义是指数据集内至少存在两个类，其类内样本的密度不高于类间样本的密度，如图6的左图和中图所示。相似度定义是指数据集内至少存在两个类，其类内样本间的相似度不高于类间样本间的相似度，如图6右图所示。

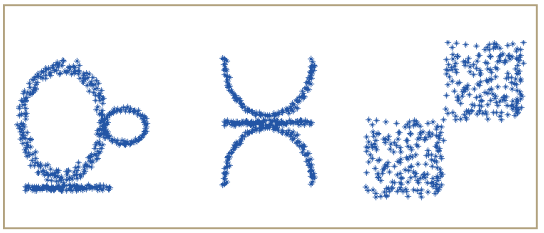

图6 相切类

重叠类 指的是数据集内至少存在一个样本同时属于两个或者多个类，如图7所示。这种情况非常类似于多标记学习，但是用做聚类分析的数据集没有类别的标记。

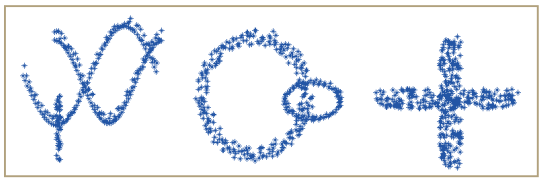

图7 重叠类

显然，非同质类、相切类和重叠类都背离了聚类分析的传统定义。如果数据集在理论上符合聚类的传统定义，则聚类分析可以将数据正确标定。但是，在实际应用中，上面说到的四种情形都可能碰到，甚至存在更加复杂的情形，如在非同质相切类中，同时存在相切类和重叠类等，如图8所示。

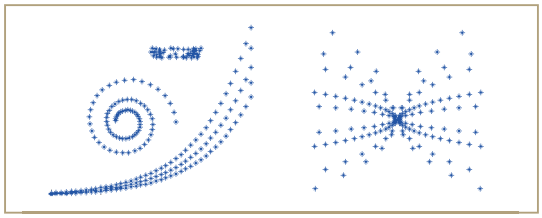

图8 复杂类

如果需要处理相切类、非同质类或者重叠类，在文献中提到两种方法可以使用：一种是适用于对类的形状加以限制并且知道类原型的情况，比如C-均值算法、k-平面聚类和模糊核聚类；另一种是在不对类的形状加以限制的情形下，需要另外的先验信息，比如文献中的半监督聚类算法（Semi-Supervised Clustering）和约束聚类算法（Constrained Clustering）。对于重叠类，有人提出了较少先验信息的聚类算法，如可加性聚类算法，但目前的效果尚不能令人满意。对于更加复杂的情形，现在的方法是构造映射使数据在新的空间里符合聚类假设，在缺少先验知识的条件下这是一项极其艰巨的任务。需要指出的是，集成聚类和谱聚类可以看作是体现这种思想的两种典型算法。谱聚类算法是通过提取与相似矩阵的有关矩阵的特征向量来形成新的特征空间，而集成聚类算法（Ensemble Clustering）是希望从不同的聚类结果中提取出更好的聚类结果，基本想法是通过共识函数来形成一个新的聚类问题。

聚类应用的特殊要求

除了类定义是非常复杂的问题之外，目前的应用环境也对聚类算法提出了许多挑战性的问题，如：

可伸缩性 现在的数据量通常十分巨大，常用的手段有数据抽样、数据网格（Grid）和特征提取等；

数据噪音 聚类的数据中常常含有噪音，如何进行数据清理是一个值得研究的问题；

加密数据聚类结果的保真性 在一些应用场合，需要对加密模糊后的数据进行聚类，如何保持加密模糊数据的聚类结果，以及反映加密模糊前数据的聚类结构也是实际应用需要解决的问题；

特殊应用领域的特殊问题 在图像分割里，所有像素点的位置是有序排列的，因此相似度计算（特别是近邻点），复杂度会大幅降低；在社区发现里，人们希望在一个连通图内，类间的顶点具有较少的边连接，而类内的顶点具有较多的边连接，这就需要定义新的相似度或者新的类定义；在文本分析里，人们希望同时发现典型文档和典型词，即所谓的异质聚类。

此外，聚类技术还存在一些其他问题，如算法的参数选择等等。有兴趣的读者可以参看相关的文献。严格说来，聚类技术发展至今，涉及领域众多，相关文献汗牛充栋。限于笔者眼力和精力，可能有更好的工作和问题本文未能予以介绍，难免留有遗珠之憾。■

注： 文中未被标注的概念或算法，或者在文献[2～3]中出现，或者已经给出英文原名。

于 剑

中国计算机学会理事。北京交通大学计算机学院教授。主要研究方向为机器学习、数据挖掘、模式识别。jianyu@bjtu.edu.cn

肖 宇

中国计算机学会学生会员。北京交通大学计算机学院博士生。主要研究方向为机器学习、数据挖掘。06120567@bjtu.edu.cn

参考文献

[1] Watanabe, Satosi . Knowing and Guessing: A Quantitative Study of Inference and Information. New York: Wiley. 1969: 376 ~ 377

[2] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Third edition, 2006, Elsivier, USA

[3] Jiawei Han, M. Kamber, Data Mining Concepts and Techniques, Morgan Kaufmann,2001

[4] Jian Yu, General c-means clustering model , IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(8): 1197 ~ 1211

[5] Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science , 2007, 315(5814) : 972 ~ 976

[6] 王 颖 陈松灿 张道强 杨绪兵, 模糊k-平面聚类算法, 模式识别与人工智能, 2007, 20(5): 704 ~ 710

[7] Dhillon, I. S., Guan, Y., & Kulis, B.. Weighted graph cuts without eigenvectors: a multilevel approach. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2007, 29: 1944 ~ 1957

[8] Lance Parsons, E. Haque, Huan Liu, Subspace clustering for high dimensional data: a review, Sigkdd Explorations, 2004, 6(1): 90 ~ 105

[9] Heyer, L.J., Kruglyak, S. and Yooseph, S., Exploring Expression Data: Identification and Analysis of Coexpressed Genes, Genome Research 9:1106 ~ 1115

[10] Girvan M. and Newman M. E. J., Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99, 2002: 7821 ~ 7826

[11] Chris Ding, Xiaofeng He, Horst D. Simon ,On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering, SDM 04

[12] B. Long, Z. Zhang and P. S. Yu, Co-clustering by block value decomposition, Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2005: 635 ~ 640

[13] Naftali Tishby, Noam Slonim, Data clustering by Markovian relaxation and the Information Bottleneck Method, NIPS 2000

[14] I.S. Dhillon, S.Mallela, and D. S. Modha, Information-theoretic co-clustering, Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2003: 89 ~ 98

[15] Zahn C. Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Transactions on Computers, 1971: 68 ~ 86