混合属性数据流的二重 k 近邻聚类算法

黄德才 沈仙桥 陆亿红
(浙江工业大学计算机科学与技术学院 杭州 310023)

摘 要 现有的数据流聚类算法大都只能处理单一数值属性的数据,不能应对同时包含数值属性与分类属性特征的数据,且已存在的混合属性数据流聚类算法在对数据的标准化处理和聚类上还有很大的改进之处,为此,提出二重 k 近邻混合属性数据流聚类算法。该算法采用 CluStream 算法的在线、离线框架,并提出了混合属性数据流下三步聚类的思想。算法先运用二重 k 近邻和改进的维度距离生成微聚类,然后利用动态标准化数据方法和基于均值的余弦模型生成初始宏聚类,最后利用基于均值的余弦模型和先验聚类结果进行宏聚类优化。实验结果表明,所提出的算法具有良好的聚类质量及可扩展性。

关键词 数据流,混合属性,聚类,二重 k 近邻
中图法分类号 TP18 **文献标识码** A

Double k-nearest Neighbors of Heterogeneous Data Stream Clustering Algorithm

HUANG De-cai SHEN Xian-qiao LU Yi-hong
(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract On the one hand, most of the existing data stream clustering algorithm can handle data with numerical attribute, but can not cope with the data containing both numeric and classification attributes. On the other hand, there is also a lot of room for heterogeneous data stream algorithms to improve standardization and clustering of data. So, double k-nearest neighbors of heterogeneous data stream clustering algorithm was proposed. The algorithm uses CluStream's online and offline framework with proposing three steps of clustering thought. Firstly, the algorithm uses double k-nearest neighbors and improved dimension distance to form micro clusters. Secondly, the algorithm uses dynamic standardization data method and cosine model based on mean value to form initial macro clusters. Thirdly, the algorithm uses cosine model based on mean value and priori clusters to do macro clustering optimization. Experimental results demonstrate that the proposed method improves clustering's accuracy and scalability.

Keywords Data stream, Heterogeneous, Clustering, Double k-nearest neighbors

1 引言

随着技术的发展,许多领域,如互联网数据传输、通信网络通话详细数据、大型零售业销售信息、网站访问日志等,每时每刻都在产生大量的数据。这些数据是连续、无界、不定速度的流式数据[1](也称为数据流,Data Stream),且大部分都包含丰富的数值属性和分类属性信息(称为混合属性数据流),其中数值属性是指属性的取值为连续数值,如长度、温度;分类属性的取值为有限的状态,如天气(晴、阴、雨)、颜色(红、橙、黄、绿)。如何有效地从这种混合属性的数据流中挖掘出具有价值的信息已显得极为重要。

聚类是数据挖掘领域中研究的热点之一,它将物理或抽象的对象集合中具有相似的对象聚集在同一个类中,属于无监督学习。目前,针对混合属性数据流的研究较少[2],且已有的大部分数据流聚类算法都局限于处理只包含数值属性的数据,少量的面向混合数据流的聚类算法也仅仅将分类属性进行简单的处理,其聚类结果都不十分理想。针对存在的问题,在对传统的维度距离公式进行改进的基础上,提出了混合属性数据流的二重 k 近邻聚类算法。二重 k 近邻利用数据点对象间的相似性来求得核心点的微簇集合,再通过动态标准化数据和基于均值的余弦模型进行初始宏聚类并经优化后得到最终聚类结果。

本文第1节简要介绍混合属性数据流产生的背景以及存在的问题;第2节是相关研究,通过分析和总结问题,提出混合属性数据流的二重 k 近邻聚类算法;第3节先总结现有算法的聚类模式,再提出新的分步聚类思想;第4节描述本文涉及的相关概念和算法计算步骤;第5节进行实验比较分析;最后总结全文。

2 相关研究

MacQueen 提出的 K-means[3]算法是基于欧式距离的经典聚类算法,但它只能对数值属性的对象集进行聚类,无法对

到稿日期:2013-01-09 返修日期:2013-04-14 本文受农村水电效益分析与增效关键技术研究与示范,水利部公益性行业科研专项(201001031)资助。

黄德才(1958—),男,博士,教授,博士生导师,CCF 高级会员,主要研究方向为数据仓库与数据挖掘、决策方法等,E-mail:hdc@zjut.edu.cn;沈仙桥(1987—),男,硕士生,主要研究方向为数据挖掘;陆亿红(1968—),女,硕士,副教授,主要研究方向为软件理论、决策方法、数据挖掘等。

分类属性或混合属性的数据集进行聚类。因此，Huang Z 等人在此基础之上先后提出了经典的 K-modes[4] 算法和 K-Prototypes[5] 算法。K-modes 算法用基于频率的方法使聚类代价函数达到最小，但其只能处理分类属性；K-Prototypes 算法统一了数值属性和分类属性的计算，但不管分类属性的类别状态个数有多少，都以数值 0 和 1 来衡量同一分类属性中不同取值之间的距离，使得聚类结果与实际类别有较大差异。

鉴于数据流聚类存在的问题，Aggarwal 等人[6] 提出了一种基于划分的 CluStream 算法。该算法由在线和离线两层组成，其中在线层执行 K-means 算法产生微簇，离线层根据用户的具体请求，利用在线层生成的微簇生成聚类结果，但该算法只能处理数值属性；随后，Aggarwal 等人[7] 提出了处理文本属性数据流聚类的算法，为分类数据设计了具有时间特性的半衰期概要元组。该算法可以快速地处理文本分类数据，但依然无法处理同时包含数值和分类的混合属性数据流。

杨春宇等人[8] 结合 CluStream 算法的两层框架，其提出了针对混合属性数据流的 HCluStream 算法，但由于该算法在处理分类属性时对每个属性的各种取值都进行匹配，因此，当分类属性很多或者取值范围很广时，算法将消耗大量的时间。

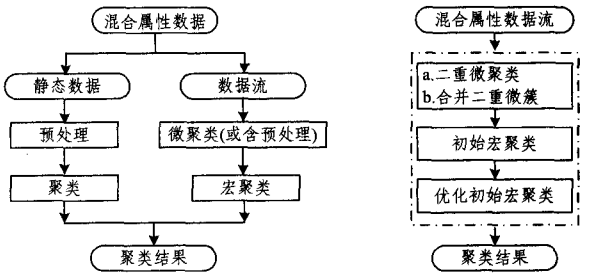
Hsu 等人[9] 针对混合属性数据流的增量聚类做了研究，算法中使用概念层次树来计算混合属性数据之间的相似度，但需要用户对数值、分类属性中的各属性取值范围给出大小关系并设置有效的差值，如果设置不当，其聚类结果与实际类别相比就会有较大误差。

由于目前关于混合属性数据流聚类的研究还比较少，也没有针对混合属性数据流聚类步骤的具体说明，本文提出先按混合属性数据流的属性特点进行微聚类，然后对混合属性数据流进行初始宏聚类，再对初始宏聚类进行优化的三步聚类算法，即混合属性数据流的二重 k 近邻聚类算法(Double k-nearest Neighbors of Heterogeneous Data Stream Clustering Algorithm，D_kHDSC 算法)，并形成一种通用的混合属性数据流的聚类框架，其有较强的可扩展性和良好的聚类效果。

3 混合属性数据流三步聚类

通过研究发现，目前大部分的数据流聚类算法都是针对只有数值属性的数据集，而已有的关于混合属性数据流的聚类算法没有注意到同一类中的数据在各维度上的相似性比与其它类中数据的相似性更大，并普遍采用如下方法直接进行聚类：①简单地丢弃分类属性部分，通过基于划分、网格、密度等方法对数据流进行聚类；②只关注数据流中的分类属性部分，而忽略数值属性部分，通过基于概率等方法对数据流进行聚类(如图 1(a)部分预处理环节)；③通过先分别处理数值属性和分类属性再结合的方法对数据流进行聚类，这也是目前研究的热点之处(如图 1(a)部分微聚类环节)。因此本文在总结以往算法经验的基础上，再结合本算法的创新过程，给出如下混合属性数据流的三步聚类算法(第②和③步都包含数据的标准化)：①接收混合属性数据流，生成二重 k 近邻微簇，经合并后得到最终微簇；②初始宏聚类；③优化初始宏聚类。

图 1 是对目前常用聚类算法的一个总结和本文改进的聚类层次，虚线框内即为本文改进部分。



(a)传统聚类层次　　(b)改进的聚类层次

图 1　传统与改进的聚类层次

新的聚类层次不仅细化了混合属性数据流聚类的内在关系，而且通过总结不同时期的聚类算法层次并进行对比，为混合属性数据流的发展提供了一种通用的聚类框架：①在线层接收数据时，以改进的维度距离公式作为新的相似性度量，生成二重 k 近邻微簇，经合并后得到高质量的微簇群；②先用数据标准化方法处理每一个微簇，然后通过初始宏聚类算法将具有相似特性的数据尽量集中到同一个簇中，为最后的聚类提供可靠的簇集合；③先对上一阶段的簇集合进行数据标准化，然后利用基于均值的余弦模型对标准化后的簇集合进行优化，从而得到聚类结果。本文算法由以下几部分组成：

(1)准备混合属性数据流环境，在接收数据流的同时进行微簇的生成，并将微簇存储到外存；

(2)读取指定时间窗口内的数据至内存，调用数据标准化和初始宏聚类算法；

(3)对初始宏聚类结果进行数据标准化，然后分析先验聚类结果并对其优化，最后将优化后的聚类结果呈现给用户。

4 相关概念及 D_kHDSC 算法

为了便于描述 D_kHDSC 算法的计算步骤，先约定一些符号含义。混合属性数据流集合表示为 $D=\{X_1, X_2, \cdots, X_i, \cdots, X_n\}$，集合 D 中每一个点 X_i 都有 m 个属性，即 $X_i=\{x_{i1}, x_{i2}, \cdots, x_{id}, x_{i(d+1)}, x_{i(d+2)}, \cdots, x_{im}\}$，其中，$\{x_{i1}, x_{i2}, \cdots, x_{id}\}$ 为数值属性，$\{x_{i(d+1)}, x_{i(d+2)}, \cdots, x_{im}\}$ 为分类属性。表 1 是一个示例数据集。

表 1　具有数值属性和分类属性的数据集

序号	温度	天气	颜色
X_1	35	晴	红
X_2	30	晴	红
X_3	15	阴	橙
X_4	25	晴	红
X_5	10	阴	黄
X_6	5	阴	黄
X_7	12	阴	橙

4.1 改进的维度距离

面向维度的距离[10] 是一种统筹考虑数值属性和分类属性差别的距离度量方法。这种距离度量方式与传统方法的不同之处在于：在多维属性情况下，任意两个对象在每一维上都接近比只在少数维上接近更有意义。设有点 $O(0,0,0)$、$A(4,4,-4)$、$B(6,0,0)$ 和阈值 $\varepsilon=5$，若采用欧几里德度量，则 $|OB|<|OA|$，即点 B 比点 A 更接近于点 O；若采用面向维度

的距离度量，则点 A 在每一维上与点 O 的距离都小于 ε，而点 B 的 X 轴与点 O 的距离大于 ε，所以点 A 比点 B 更接近于点 O。

在具体的聚类过程中，文献[10]设置了两步处理方法，即：①若 X_i、X_j 的第 p 维是数值属性，那么定义其距离为 $d(x_{ip},x_{jp})=|x_{ip}-x_{jp}|$；②若数值属性有 d 维，那么定义一个 d 维的相似度阈值向量 $\varepsilon=\{\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_d\}$；③若第 p 维分类属性有 n 个不同的取值，那么定义一个 $n\times n$ 维的相似度矩阵，以存储该维度不同取值之间的相似度。以上处理方法存在的问题是：相似度向量和矩阵是由用户给出的属性列各取值之间的度量值组成的，且参数较多。

本文根据同一类中的数据在各维度上比其它类中的数据更加相似的特点，做出如下改进(设数据集 $D=\{X_1,X_2,\cdots,X_i,\cdots,X_n\}$)：①将数值属性的距离公式推广到混合属性，即任意两点 X_i、X_j 在第 p 维的距离 $d(x_{ip},x_{jp})=|x_{ip}-x_{jp}|$。由于微聚类阶段的主要任务是将各维度取值相同的聚为一类，因此对于分类属性而言，公式采用二元化方法，即若 X_i、X_j 的属性值 x_{ip}、x_{jp} 相等，则 $d(x_{ip},x_{jp})$ 为 0，否则为 1；②提出了最优维度距离集合的概念。

设 D 中数据 X_i、X_j，则 X_i 与 X_j 的维度距离为：

$$Dm(X_i,X_j)=\sum_{p=r}^{s}d(x_{ip},x_{jp}),i\neq j \tag{1}$$

X_i 与其余点按式(1)计算之后，便得到 X_i 的维度距离集合 $DM(X_i)=\{Dm(X_i,X_1),Dm(X_i,X_2),\cdots,Dm(X_i,X_j),\cdots,Dm(X_i,X_n)\}(i\neq j)$。由 $DM(X_i)$ 可得 X_i 与其它点的 k 个最短距离，构成的集合称为 X_i 的最优维度距离集合，记作 $DM_{opt}(X_i)=\{Dm(X_i,X_1),Dm(X_i,X_2),\cdots,Dm(X_i,X_j),\cdots,Dm(X_i,X_k)\}$，并满足条件 $Dm(X_i,X_1)<\cdots<Dm(X_i,X_j)<\cdots<Dm(X_i,X_k)$。

当式(1)中的 $r=1,s=d$ 时，由 $DM_{opt}(X_i)$ 可得点 X_i 的最优数值属性维度距离集合，记作 $DMN_{opt}(X_i)$，相应的数值属性维度距离公式和集合分别记作 $Dmn(X_i,X_j)$ 和 $DMN(X_i)$；当式(1)中 $r=d+1,s=m$ 时，由 $DM_{opt}(X_i)$ 得到点 X_i 的最优分类属性维度距离集合，记作 $DMC_{opt}(X_i)$，相应的分类属性维度距离公式和集合分别记作 $Dmc(X_i,X_j)$ 和 $DMC(X_i)$。因此，本文改进的维度距离更好地利用了数值、分类属性的特性，可提高聚类精度。

4.2 二重 k 近邻的生成与合并

4.2.1 k 近邻

k 近邻[11]是传统数值属性聚类算法中很好的方法，学术界根据此方法提出了很多有代表性的算法。传统 k 近邻的概念为：设 D 为给定数据集，对象 X_i 的 k 近邻是 D 中距离 X_i 最近的 k 个数据点构成的集合；其中最经典的距离就是数值向量的欧式距离。因此，传统的 k 近邻只能处理全部是数值属性的数据，加之混合属性数据的维数较多、数值属性是定量的、分类属性是定性的，故难以直接用 k 近邻来完成聚类。

根据传统 k 近邻的不足和混合属性数据流既有数值属性又有分类属性的特点，本文提出了针对混合属性数据流的二重 k 近邻生成算法，即在一个微簇中，同时保存关于数值属性和分类属性的 k 近邻。如果数值属性个数不少于分类属性(即 $d\geq m-d$)，则第一个 k 近邻是关于数值属性的，第二个 k 近邻是关于分类属性的；反之(即 $d<m-d$)，第一个 k 近邻是关于分类属性的，第二个 k 近邻是关于数值属性的。以上两种情况都称第一个 k 近邻的优先级大于第二个 k 近邻。

4.2.2 混合属性数据流的二重 k 近邻

本文提出的二重 k 近邻概念是建立在改进的维度距离基础上的。对于给定数据集 $D=\{X_1,X_2,\cdots,X_i,\cdots,X_n\}$，混合属性数据流的二重 k 近邻的生成，就是求点 X_i 的 $DMN_{opt}(X_i)$ 与 $DMC_{opt}(X_i)$，最后将其合并的过程。为叙述方便，下面用 X 表示任意数据点 X_i。

注意到混合属性包括数值属性和分类属性。

①若 D 是数值属性占优(即 $d\geq m-d$)，则对于 X 的二重 k 近邻来说，X_i 若要排在 X_{i+1} 前面，则必须满足如下条件 r_1 或 r_2：

r_1：$Dmn(X,X_i)<Dmn(X,X_{i+1})$ 且 $Dmc(X,X_i)<Dmc(X,X_{i+1})$；

r_2：$Dmn(X,X_i)<Dmn(X,X_{i+1})$；

②若 D 是分类属性占优(即 $d<m-d$)，则以上条件 r_2 改为：

r_2：$Dmc(X,X_i)<Dmc(X,X_{i+1})$

假设数据集 D 按照限制条件 r_1、r_2 得到 X 的二重 k 近邻分别为 $N_k^{(1)}(X)$ 和 $N_k^{(2)}(X)$，那么 X 的 k 近邻微簇 $N_k(X)=N_k^{(1)}(X)\cap N_k^{(2)}(X)$，并称 X 为该微簇的核心点，简称核心点。

4.2.3 二重 k 近邻生成算法

给定数据集 $D=\{X_1,X_2,\cdots,X_i,\cdots,X_n\}$($n$ 足够大)以及近邻大小 k、内存缓冲区大小 $b(\geq k)$，则生成微簇 $N_k(X)$ 的算法如下。

算法 1 genOptK()

输入：D

输出：每个点 X_i 的微簇 $N_k(X_i)$

Step1 取一个数据点 X_i；

Step2 如 $i\leqslant b$，则 X_i 的第一个 k 近邻 $N_k^{(1)}(X_i)$(距离法则为 r_1)和第二个 k 近邻 $N_k^{(2)}(X_i)$(距离法则为 r_2)都为 $\{X_1,X_2,\cdots,X_i,\cdots,X_{i+b-1}\}$ 中距离 X_i 最近的 k 个数据点构成的子集合；转 Step3；

否则($i>b$)，X_i 的第一个 k 近邻 $N_k^{(1)}(X_i)$(距离法则为 r_1)和第二个 k 近邻 $N_k^{(2)}(X_i)$(距离法则为 r_2)都为 $\{X_{i-b+1},\cdots,X_{i+b-1}\}$ 中距离 X_i 最近的 k 个数据点构成的子集合；转 Step3；

Step3 输出 $N_k(X_i)=N_k^{(1)}(X_i)\cap N_k^{(2)}(X_i)$ 并结束。

对表 1 的数据，设参数 $k=2,b=4$：当数据点序号 i 小于等于缓冲区 $b(=4)$ 时，如 X_4 的二重 k 近邻分别为 $N_2^{(1)}(X_4)=\{X_1,X_2\}$、$N_2^{(2)}(X_4)=\{X_2\}$，所以 $N_2(X_4)=\{X_2\}$；当数据点序号 i 大于缓冲区大小 $b(=4)$ 时，如 X_5 的二重 k 近邻分别为 $N_2^{(1)}(X_5)=\{X_6,X_7\}$、$N_2^{(2)}(X_5)=\{X_6,X_7\}$，所以 $N_2(X_5)=\{X_6,X_7\}$。genOptK()算法产生任意数据点的微簇，一般表示为 $N_k(X)$。

4.3 聚类算法

下面将给出主要的聚类过程，即如何将各核心点 X 经 genOptK()生成的微簇 $N_k(X)$ 聚成最终类。具体过程为：先由算法 depthCluster()进行初始宏聚类，然后将其作为 optCluster()算法的输入，完成对初始宏聚类的优化，得到最终聚类结果。其中，聚类参数 $ratio$ 和 ω 为给定的阈值，不同时刻的 $ratio$ 和 ω 均为同一个值。在正式介绍聚类算法之前，先阐述与之相关的概念，即分类属性量化、动态数据标准化和余弦模型。

4.3.1 分类属性量化

由于 K-Prototypes 算法在整个聚类过程中以 0、1 的方法来处理各维度不同取值之间的距离，因此分类属性各维度取值之间的距离度量存在缺乏区分度和精度不高的问题，也就使得聚类结果与实际类别有较大差异。本文通过分析以往算法的优点和分类属性本身的特点，在初始宏聚类阶段先将分类属性按照实际情况排序，然后将位序赋值给原值。例如，表 1 数据集中的两个分类属性先按实际情况可排序为{晴，阴}和{红，橙，黄}两类，然后将相应位序赋值给它们，即分别对应{1,2}和{1,2,3}。

4.3.2 动态数据标准化

由于混合属性数据流中各维度的取值范围不同，不能直接参与运算，因此在初始宏聚类过程中，需先消除各属性的量纲。混合属性中的分类属性部分按上小节方法对取值进行量化。假设数据集 D 中以 X_i 为核心的微簇 $N_k(X_i)=\{X_1, X_2, \cdots, X_j, \cdots, X_k\}$，$D$ 中第 p 维上的最小值为 $\min[p]$，最大值为 $\max[p]$($p\in[1,m]$)，则对于 D 中以 X_i 为核心的微簇来说，核心点 X_i 的各维 x_{ip} 的标准化公式[12]如下：

$$x_{ip}'=\left|\frac{(x_{ip}+\sum_{j=1}^{k}x_{jp})/(k+1)-\min[p]}{\max[p]-\min[p]}\right| \tag{2}$$

传统的数据标准化处理的对象是每一个数据，从而使得误差较大。本文通过研究，采用对微簇或类进行标准化处理来替代传统方法，具有稳定性和降低误差的优点。经过式(2)的处理，X_i 中各值 x_{ip}' 的取值范围是[0,1]，算法如下。

算法 2 expStandardization()

输入：时间窗口，各核心点的微簇 $N_k(X_i)$，$\min[p]$和 $\max[p]$($p\in[1,m]$)

输出：标准化后的 $N_k(X_i)$

Step1 依次读取时间窗口内的数据；

Step2 将每一个数据 X_i 中 x_{ip}都经式(2)运算，并将新结果依然存在 X_i 内；

Step3 将 X_i 保存至外存中，转 Step1；完成，结束程序。

为便于描述，将微簇的标准化表示为 $E(N_k(X))$，其中，X 为核心点。以核心点 X_i 为例，$E(N_k(X_i))=\{x_{i1}', x_{i2}', \cdots, x_{id}', x_{i(d+1)}', x_{i(d+2)}', \cdots, x_{im}'\}$，其第 p 维标准化后的数据表示为 $E(N_k(X_i))_p'$，$p\in[1,m]$。

4.3.3 余弦模型

余弦是聚类中常用的相似度计算法[13]，针对混合属性聚类，比较常用的是基于 K-Prototypes 的处理方法，但其存在不足，前文已阐述，故本文对其进行了改进。假设 X_i、X_j 为标准化后的数据集 D 中两核心点，在此处均当作具有 m 维的向量，因此$|X_i|$、$|X_j|$为向量的模，并将 $div(|X_i|,|X_j|)$定义为模较小值除以模较大值，ω 为用户输入的阈值，那么 X_i 与 X_j 的相似度和模的满足条件如式(3)、式(4)所示，并称式(3)和式(4)为混合属性数据流之间相似的余弦模型。

$$\cos(X_i,X_j)=\frac{\sum_{p=1}^{m}x_{ip}'\cdot x_{jp}'}{\sqrt{\sum_{p=1}^{m}(x_{ip}')^2}\cdot\sqrt{\sum_{p=1}^{m}(x_{jp}')^2}} \tag{3}$$

$$div(|X_i|,|X_j|)\geqslant\omega \tag{4}$$

假设 $C_i=\{X_1, X_2, \cdots, X_i, \cdots, X_s\}$和 $C_j=\{X_{s+1}, X_{s+2}, \cdots, X_j, \cdots, X_n\}$是以 X_i、X_j 为初始点经初始宏聚类后形成的两个类，那么在宏聚类优化之前先标准化类簇，即对每一类用式(2)进行处理，最终可得到 m 维向量 $E(C_i)$，进而可得出两个类之间的相似度和模的满足条件式(5)、式(6)，其中 $E(C_i)$和 $E(C_j)$当作具有 m 维的向量，并将其称为混合属性数据流中基于均值的余弦模型。

$$\cos(E(C_i),E(C_j))=\frac{\sum_{p=1}^{m}E(C_i)_p'\cdot E(C_j)_p'}{\sqrt{\sum_{p=1}^{m}E(C_i)_p'^2}\cdot\sqrt{\sum_{p=1}^{m}E(C_j)_p'^2}} \tag{5}$$

$$div(|E(C_i)|,|E(C_j)|)\geqslant\omega \tag{6}$$

同理可以得到两微簇之间的相似度以及模的满足条件，表示为 $\cos(N_k(X_i),N_k(X_j))$ 和 $div(|E(N_k(X_i))|,|E(N_k(X_j))|)\geqslant\omega$。其中，$\omega$ 的意义在于控制两个向量的模长在聚类可接受的范围之内，即避免两向量在夹角很小的时候模长却相差很大的情况。基于均值的余弦模型不仅提高了混合属性中分类部分的精度，而且使数值和分类部分融合为一体，也加快了聚类的速度。

4.3.4 初始宏聚类

算法 3 depthCluster()

输入：聚类参数 ratio 和 ω，数据集 D 中每个标准化后的点 X_i 的微簇 $N_k(X_i)=\{X_1, X_2, X_3, \cdots, X_k\}$

输出：类簇及其编号：$\{C_1, C_2, \cdots, C_i, \cdots, C_j, \cdots, C_q\}$

Step1 新建类簇 C_q，并从 $D-\bigcup_{i=1}^{q-1}C_i$ 中选择一个未经处理过的微簇核心点 X_i，先将 X_i 放入簇中；如果处理完毕，转步骤 4；

Step2 遍历 X_i 微簇中元素，对于每一个元素 X_j($1\leqslant j\leqslant k$)转步骤 3；

Step3 若 $\cos(E(N_k(X_i)),E(N_k(X_j)))\geqslant$ratio 且 $div(|E(N_k(X_i))|,|E(N_k(X_j))|)\geqslant\omega$，则将 X_j 放入簇中；进入初始宏聚类，转步骤 2(此时步骤 2 中的元素 X_i 即为 X_j)；

否则 如果已进入 X_j 的递归，返回

否则 转步骤 2

如果处理完毕，转步骤 1

Step4 输出类簇及其编号并结束。

混合属性数据 X_i、X_j 越相似，$\cos(E(N_k(X_i)),E(N_k(X_j)))$越接近于 1，相反则越接近于 0，因此 *ratio* 的取值范围为[0,1]。算法 depthCluster()将相似性高的微簇聚成一体，然后再由算法 optCluster()通过分析现有类簇的各维度期望和类间的余弦模型，对其进行优化，得到最终的聚类结果。

4.3.5 宏聚类优化

算法 4 optCluster()

输入：聚类参数 ratio 和 ω，类簇及其编号

输出：宏聚类优化后的类簇及其编号

Step1 对每一个类调用标准化方法，并新建合并类集合，初始化为空；

Step2 选择一个类 C_i，计算其与另外所有的类 C_j($1\leqslant j\leqslant q, i\neq j$)之间的余弦模型；当没有足够的类可用于计算时，先将剩余的类放入合并类簇，再转 Step4；

Step3 如果满足 $\cos(E(C_i),E(C_j))\geqslant$ratio 和式(6)，那么先合并所有这些类，依然表示为 C_i；然后计算合并后的类 $E(C_i)$，并重新为所有类编号；为了算法描述方便，假设 C_i 的位置在新的类集合中不发生改变，集合表示为$\{C_1, C_2, \cdots, C_i, \cdots, C_j, \cdots, C_{q'}\}$，重新计算 C_i 与其余类 C_j 的余弦模型，重复这一步；
否则将 C_i 放入合并类集合并编号；再为剩下的类重新编号，为了算法描述方便，集合表示为$\{C_1, C_2, \cdots, C_i, \cdots, C_j, \cdots, C_{q''}\}$；转 Step2；

Step4 结束程序

至此，整个算法已描述完整。结合表1数据集，最终的聚类结果为：$\{X_1, X_2, X_4\}, \{X_3, X_5, X_6, X_7\}$。

5 性能分析和仿真实验

5.1 性能分析

(1)：实验1在线部分模拟的数据流速约为每秒25个数据点，其中：接收数据部分的时间复杂度为 $O(n)$；生成微簇部分由于涉及二重 k 近邻的创建和维护，因此这部分的时间复杂度为 $O(2k*n)$，总时间复杂度为 $O(n)+O(2k*n)$，其中 n 为处理的数据点数量。

(2)D_kHDSC算法按序从外存将数据读进内存的同时，对数据进行标准化处理，时间复杂度为 $O(n)$；初始宏聚类使用剪枝和Map映射，使得每一数据点只被处理一次，因此时间复杂度为 $O(n)$；假设混合属性的维度为 m，最终聚类阶段的合并次数为 v，那么时间复杂度为 $O(m*v)$，总时间复杂度为 $2*O(n)+O(m*v)$。

5.2 仿真实验

实验中的算法采用C++语言编写，编译环境为Microsoft Visual C++ 6.0，硬件环境为Intel(R) Core(TM) Duo 2.00GHz CPU、内存为2GB、硬盘为500GB，操作系统为Windows 7。实验1所用数据集为UCI公共数据库的zoo数据集，名称为zoo.arff。实验2为算法的时间性能对比。

实验1　zoo.arff为包含1个数值型、15个分类型的混合属性数据集，分为7个类。由于K-modes和K-Prototypes算法为混合属性的聚类提出了经典的解决方法，而且此后的很多研究工作也基于这两种算法展开，因此本文将它们作为比较算法，来测试D_kHDSC算法在聚类性能上所做的改进。因为已知的数据流聚类算法并不能对所有数据集进行完整的聚类，即存在聚好的类中数据丢失或者类的质量不纯(有噪声数据)，而被广泛采用的聚类熵可以对这些不足做出评价。表2是3种算法在zoo数据集上的聚类表现，聚类熵公式如下：

$$Entr(C_i)=-\frac{1}{\log(N)}\sum_{t\in T}\frac{N_{it}}{N_i}\log\left(\frac{N_{it}}{N_i}\right) \tag{7}$$

式中，N 是数据集的数量，N_i 是类簇 C_i 中数据的数量，N_{it} 是该簇中表示为类 t 的数量。$Entr(C_i)$的范围是[0,1]，1表示各个类簇的类是均匀分布的，0表示各类簇完全是由一个类组成的纯净簇，数值越低则聚类质量越高。

表2　聚类熵

对比	算法及相应值		
	K-modes	K-prototypes	D_kHDSC
zoo	0.161	0.15	0.108

对比3个算法的实验结果可知，D_kHDSC算法更好地完成了聚类任务。由于前两个算法在不同程度上丢失了分类属性的信息，再加上本算法利用基于维度距离的二重 k 近邻不断凝聚在各维度上都接近的数据，使得前两个算法的聚类效果没有本算法好。

实验2　表3显示的是D_kHDSC算法与K-prototypes算法的时间性能数据，实验数据集同实验1。由于K-prototypes算法的时间复杂度为 $O(n*k*t)$，具有处理大型数据集的能力，比现有混合属性数据流算法的时间性能好，且其在数据流环境中运用在线和宏聚类两个层次的聚类方法来聚类，因此本算法选择K-prototypes做对比实验，其中 n 是数据对象的数目，k 是类簇的数目，t 是迭代的次数。从表中可以看出，除了在线层时间相等之外，聚类层还具有一定的优势：一方面，K-prototypes算法需要随机选择 k 个中心点，这对聚类的精度和速度都有比较大的影响，且少量的噪声数据也可能严重影响该算法的运行速度；另一方面，本文算法提出的三步聚类法更适应混合属性数据流环境，提高了聚类的速度。

表3　算法运行时间(单位：ms)

算法时间对比	在线	微聚类	宏聚类
D_kHDSC	16	15	0.5
K-prototypes	16	24	

结束语　混合属性数据流的二重 k 近邻聚类算法在接收数据和生成微簇阶段先不断凝聚在各维度上都接近的数据，然后在初始宏聚类阶段通过基于均值的余弦模型形成最终类的雏形，再在宏聚类优化阶段通过分析类簇中各维度的期望做进一步聚类，以得到优质的聚类结果。由于本算法将混合属性数据流聚类的大部分工作移到了微簇生成和初始宏聚类阶段，因此下一步的研究工作是如何提出更出众的微簇算法以得到优质的微簇群，以及混合属性数据流中的分类属性量化方法。

参 考 文 献

[1] 屠莉，陈崚，绉凌君. 数据流的网格密度聚类算法[J]. 小型微型计算机系统，2009，30(7)：1376-1383

[2] 王述云，胡运发，范颖捷，等. 基于距离与熵的混合属性数据流聚类算法[J]. 小型微型计算机系统，2010，31(12)：2365-2372

[3] Marques J P. Pattern recognition concepts，methods and applications[M]. Beijing：Tsinghua University Press，2002：51-74

[4] Huang Z. Extensions to the K-means algorithm for clustering large datasets with categorical values[J]. Data Mining and Knowledge Discovery II，1998(2)：283-304

[5] Huang Z，Ma N G. Fuzzy K-modes algorithm for clustering categorical data[J]. IEEE Transactions on Fuzzy Systems，1999，7(4)：446-452

[6] Aggarwal C，Han J，Wang J，et al. A Framework for Clustering Evolving Data Streams[C]// Proceedings of 29th Very Large Data Bases Conference. 2003，81-92

[7] Aggarwal C C，Yu P S. A framework for clustering massive text and categorical data st reams[C]// Proc of the 6th SIAM Int Conf on Data Mining. Bethesda，2006：477-481

[8] 杨春宇，周杰. 一种混合属性数据流聚类算法[J]. 计算机学报，2007，30(8)：1364-1372

[9] Hsu C C，Huang Y. Incremental clustering of mixed data based on distance hierarchy[J]. Expert Systems with Applications，2008，35(3)：1177-1185

[10] 黄德才，吴天虹. 基于密度的混合属性数据流聚类算法[J]. 控制与决策，2010，25(3)：416-422

[11] 刘青宝，邓苏，张维明. 基于相对密度的聚类算法[J]. 计算机科学，2007，34(2)：192-196

[12] 李桃迎，陈燕，张金松，等. 基于聚类融合的混合属性数据增量聚类算法[J]. 控制与决策，2010，27(4)：603-609

[13] 周津，陈超，俞能海. 采用对象特征向量表示法的标签聚类算法[J]. 小型微型计算机系统，2012，33(3)：525-531