

# 动态模糊二叉决策树构造方法

谢 琳

(江苏教育学院 苏州学前教育分院 江苏 苏州 215008)

【摘要】 动态模糊数据分析是海量数据处理的核心课题之一. 讨论了动态模糊决策树的属性算法, 通过动态模糊二叉决策树的介绍, 给出了动态模糊决策树中各结点以及各层对实例集划分之间的关系. 由于划分格是对论域的划分, 进一步定义了动态模糊划分格, 给出了关于动态模糊决策树各层对实例集划分组成的集合的定理, 并且证明了动态模糊决策树的各层对实例集的划分组成的集合既是一个线性有序集也是一个动态模糊划分格等.

【关键词】 动态模糊格, 动态模糊决策树, 动态模糊二叉决策树

【中图分类号】 TP181 【文献标志码】 A 【文章编号】 1672-1292(2011) 04-0057-06

## A Method of Contruction for Dynamic Fuzzy Binary Decision Tree

Xie Lin

( College of Preschool of Suzhou Jiangsu Institute of Education Suzhou 215008 ,China)

**Abstract:** The dynamic fuzzy data analysis is one of the key topics for mass data. Now many researchers use fuzzy logic for analysis. This article discusses the properties algorithm of dynamic fuzzy decision tree. It gives the dynamic fuzzy binary decision tree and the relation between the nodes and the layers of the set of instances. The partition lattice is for the domain of discourse. It defines the dynamic fuzzy partition Lattice, and gives the theorems of Layers for the set of instances on the dynamic fuzzy decision tree. It proves that the set is a linearly ordered and that it is also dynamic fuzzy partition lattice.

**Key words:** dynamic fuzzy lattice, dynamic fuzzy decision tree, dynamic fuzzy binary decision tree

在动态模糊决策树(Dynamic Fuzzy Decision Tree, DFDT)学习中<sup>[1-4]</sup>, 属性 $(\vec{a}_i, \vec{\mu}_i)$ 的值域表示为 $(\vec{V}_{a_i}, \vec{V}_{\mu_i}) = \{(\vec{v}_1, \vec{\mu}_1), (\vec{v}_2, \vec{\mu}_2), \dots, (\vec{v}_k, \vec{\mu}_k)\}$ , 则 $|\vec{V}_{a_i}, \vec{V}_{\mu_i}| = k$ 表示属性 $(\vec{a}_i, \vec{\mu}_i)$ 值的个数.

根据属性 $(\vec{a}_i, \vec{\mu}_i)$ 的取值不同, 可得到属性 $(\vec{a}_i, \vec{\mu}_i)$ 对实例集 $U$ 的划分, 即 $\text{Div}_{(\vec{a}_i, \vec{\mu}_i)}(\vec{U}, \vec{U}) = \{(\vec{S}_1, \vec{S}_1), (\vec{S}_2, \vec{S}_2), \dots, (\vec{S}_n, \vec{S}_n) \mid (\vec{S}_i, \vec{S}_i) = \text{Par}_{(\vec{a}_i, \vec{\mu}_i)}^{(\vec{v}_k, \vec{\mu}_k)}(\vec{U}, \vec{U})\}$ , 其中 $n \leq |\vec{V}_{a_i}, \vec{V}_{\mu_i}|$ .

由于实例集中对应 $n = |\vec{V}_{a_i}, \vec{V}_{\mu_i}|$ 个类别, 条件属性 $(\vec{A}_i, \vec{\mu}_i)$ 对每个类别都有一个划分的确定度, 则条件属性 $(\vec{A}_i, \vec{\mu}_i)$ 对于整个实例集的划分确定度表示为:

$$\text{Cer}_{(\vec{A}_i, \vec{\mu}_i)}(\vec{U}, \vec{U}) = \sum_{k=1}^n \text{pro}_k \text{Cer}_{(\vec{A}_i, \vec{\mu}_i)}^k(\vec{U}, \vec{U}),$$

其中 $\text{pro}_k = |\text{Par}_{(\vec{A}_i, \vec{\mu}_i)}^{(\vec{v}_k, \vec{\mu}_k)}(\vec{U}, \vec{U})| / |\vec{U}, \vec{U}|$ 表示实例集 $(\vec{U}, \vec{U})$ 中属于第 $(\vec{v}_k, \vec{\mu}_k)$ 类的实例个数占整个实例集的比例.

用 $\Gamma_{(\vec{A}_i, \vec{\mu}_i)}(\vec{U}, \vec{U}) = 1 - \text{Cer}_{(\vec{A}_i, \vec{\mu}_i)}(\vec{U}, \vec{U}) = 1 - \sum_{k=1}^n \text{pro}_k \text{Cer}_{(\vec{A}_i, \vec{\mu}_i)}^k(\vec{U}, \vec{U})$ 作为决策树分类中属性选择条件,  $\Gamma_{(\vec{A}_i, \vec{\mu}_i)}(\vec{U}, \vec{U})$ 的取值范围为 $[0, 1]$ .

若 $\Gamma_{(\vec{A}_i, \vec{\mu}_i)}(\vec{U}, \vec{U}) = 0$ , 则条件属性 $(\vec{A}_i, \vec{\mu}_i)$ 对整个实例集的划分确定度为 1, 表示条件属性 $(\vec{A}_i, \vec{\mu}_i)$ 可以确定的划分实例集, 其中不包含不确定因素; 若 $\Gamma_{(\vec{A}_i, \vec{\mu}_i)}(\vec{U}, \vec{U}) = 1$ , 表示在条件属性 $(\vec{A}_i, \vec{\mu}_i)$ 上为实例集

收稿日期: 2011-08-10.

基金项目: 国家自然科学基金(60775045, 61033013).

通讯联系人: 谢琳, 讲师, 研究方向: 人工智能、机器学习、动态模糊逻辑. E-mail: xie\_lin\_2007@126.com

的一个最大不确定的划分. 因此  $\Gamma_{(\tilde{A}_i, \tilde{A}_j)}(\tilde{U}, \tilde{D})$  的值越小, 表示属性对实例集的划分越确定. 在属性选择算法中, 选择能使  $\Gamma_{(\tilde{A}_i, \tilde{A}_j)}(\tilde{U}, \tilde{D})$  取得最小值的条件属性  $(\tilde{A}_i, \tilde{A}_j)$  作为分支属性.

在构建决策树的过程中<sup>[5]</sup>, 属性  $(\tilde{A}_i, \tilde{A}_j)$  作为分支属性, 应该使分支能够涵盖尽可能多的实例, 这样就使待分类的实例数目减少, 从而使整个树的分支相对减少, 以达到简化的目的. 满足  $\min(\Gamma_{(\tilde{A}_i, \tilde{A}_j)}(\tilde{U}, \tilde{D}))$  的属性符合以上条件, 且通过使用  $\text{pro}_k$  更全面地考虑决策属性每个划分对整个决策实例集分类的贡献. 通过在属性的可用集中选取最好的属性建立分支, 已被用于进行分支的属性不再进行选取.

算法的基本描述为:

$$(1) \text{ 对于每个条件属性, 计算 } \text{Div}_{(\tilde{A}_i, \tilde{A}_j)}(\tilde{U}, \tilde{D}), \text{Div}_{(\tilde{A}_i, \tilde{A}_j)}(\tilde{U}, \tilde{D}) = \left\{ \bigcup_{k=1}^{I(\tilde{V}_{\tilde{A}_i}, \tilde{V}_{\tilde{A}_j})} \text{Par}_{(\tilde{A}_i, \tilde{A}_j)}^{(\tilde{v}_k, \tilde{v}_k)}(\tilde{U}, \tilde{D}) \right\};$$

(2) 计算条件属性  $(\tilde{A}_i, \tilde{A}_j)$  将实例划分为  $(\tilde{v}_j, \tilde{v}_j)$  类的确定度  $\text{Cer}_{(\tilde{A}_i, \tilde{A}_j)}(\tilde{U}, \tilde{D})$ , 以及决策属性值为  $(\tilde{v}_j, \tilde{v}_j)$  的实例个数占实例集总个数的比例  $\text{pro}_j$ ;

(3) 计算  $\Gamma_{(\tilde{A}_i, \tilde{A}_j)}(\tilde{U}, \tilde{D})$ , 选取满足  $\min(\Gamma_{(\tilde{A}_i, \tilde{A}_j)}(\tilde{U}, \tilde{D}))$  的属性作为分支属性, 此时将实例集进行了划分:  $\{\text{Par}_{(\tilde{A}_i, \tilde{A}_j)}^{(\tilde{v}_1, \tilde{v}_1)}(\tilde{U}, \tilde{D}), \text{Par}_{(\tilde{A}_i, \tilde{A}_j)}^{(\tilde{v}_2, \tilde{v}_2)}(\tilde{U}, \tilde{D}), \dots, \text{Par}_{(\tilde{A}_i, \tilde{A}_j)}^{(\tilde{v}_k, \tilde{v}_k)}(\tilde{U}, \tilde{D})\}$ ;

(4) 判断  $\text{Par}_{(\tilde{A}_i, \tilde{A}_j)}^{(\tilde{v}_k, \tilde{v}_k)}(\tilde{U}, \tilde{D})$  中包含的实例是否属于同一类别: 若属于同一类别, 则其对应的结点为叶结点; 否则, 转 (1) 重复前 3 步; 已经用于分支不再进行选择.

本文在以上概念的基础上进一步给出动态模糊二叉决策树的构造方法.

## 1 动态模糊二叉决策树

测试属性有多个属性值可能使得每个内部结点有两个或多个分支. 把每个内部结点只有两个分支的动态模糊决策树称为动态模糊二叉决策树 (Dynamic Fuzzy Binary Decision Tree, DFBDT). 当前有很多的算法可以建立二叉决策树, 如 CRAT 算法, 使用熵来作为选择最佳分裂属性和标准的度量, 且在每个子类产生子结点的地方只产生两个子结点<sup>[6]</sup>. 文献 [7] 给出了基于属性—值对的信息增益优化的二叉决策树生成算法, 并给出了多值属性转化为二值属性的方法.

动态模糊二叉决策树是 DFBDT 的一种特殊形式, 在树的结构上, 二者存在一定的差异. 在决策树的动态模糊化部分, 通过引入  $\geq_g$  关系, 分析决策树的各结点对应的实例子集之间的关系, 同时决策树各层也相应地对实例集进行了划分, 并且这些划分满足一定的关系. DFBDT 作为 DFBDT 的特殊形式, 这两者之间的关系如图 1 所示.

在 DFBDT 中落入上层结点与直接下层结点的实例组成的集合之间存在偏序关系, 在 DFBDT 中, 如 Attribute  $B = V_B$  与其两个子结点所包含的实例组成的集合  $\{\{1, 4, 5, 6, 7, 8, 9, 10, 11\}, \{1, 4, 6, 8, 9, 10\}, \{5, 7, 11\}\}$ , 其

元素之间不仅满足偏序的关系, 且  $\{1, 4, 6, 8, 9, 10\}$  与  $\{5, 7, 11\}$  的并等于  $\{1, 4, 5, 6, 7, 8, 9, 10, 11\}$ , 即满足半格的定义. 跟 DFBDT 一样, 分支属性对整个实例集也进行了划分. 为了研究对象域的各种划分, 文献 [8, 9] 提出了划分格的概念, 将其引入到决策树中, 来研究分支属性对实例集的划分.

一个动态模糊决策树可表示为:  $(\tilde{T}, \tilde{D}) = \langle \tilde{U}, \tilde{D} \rangle, \langle \tilde{C}, \tilde{D} \rangle, \langle \tilde{D}, \tilde{D} \rangle, \langle \tilde{V}, \tilde{V} \rangle, \langle \tilde{F}, \tilde{F} \rangle$ . 其中  $\langle \tilde{U}, \tilde{D} \rangle$  表示实例集  $\langle \tilde{C}, \tilde{D} \rangle \cup \langle \tilde{D}, \tilde{D} \rangle = (\tilde{A}, \tilde{A})$  是属性集合, 子集  $\langle \tilde{C}, \tilde{D} \rangle$  和  $\langle \tilde{D}, \tilde{D} \rangle$  分别称为条件属性和决策 (结果) 属性  $\langle \tilde{V}, \tilde{V} \rangle = \bigcup_{(\tilde{v}, \tilde{v}) \in (\tilde{A}, \tilde{A})} (\tilde{V}_i, \tilde{V}_i)$  表示属性  $(\tilde{A}, \tilde{A})$  的值域  $\langle \tilde{F}, \tilde{F} \rangle: (\tilde{U}, \tilde{D}) \times (\tilde{A}, \tilde{A}) \rightarrow (\tilde{V}, \tilde{V})$  为一个映射,

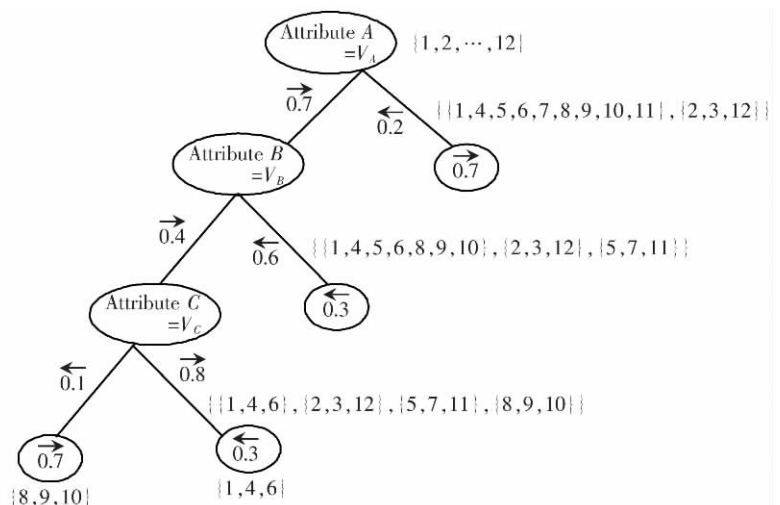


图 1 DFBDT 示例

Fig.1 The example of DFBDT

它指定了实例集 $(\vec{U}, \vec{D})$ 中每一个对象的属性值, 且 $(\vec{F}, \vec{F}) \subseteq [0, 1] \times [\leftarrow, \rightarrow]$ .

**定义 1** 给定一个有限论域  $U$  称  $\pi = \{X_i \mid 1 \leq i \leq m\}$  为  $U$  的一个划分. 若  $\pi$  满足以下条件:

- (1)  $X_i$  非空;
- (2)  $X_i \cap X_j = \emptyset \quad i \neq j$ ;
- (3)  $\cup \{X_i \mid 1 \leq i \leq m\} = U$ .

可以定义偏序关系来表示划分之间的粗细程度. 设  $\pi_1, \pi_2$  为论域  $U$  的两个划分. 若  $\pi_1 \leq \pi_2$  则  $\pi_1$  的任一划分块都包含于  $\pi_2$  的某一划分块. 即  $\pi_1, \pi_2$  为论域  $U$  的划分  $\pi_1 \leq \pi_2$  iff  $\forall X_i \in \pi_1, \exists X_j \in \pi_2$  使得  $X_i \subseteq X_j$ . “ $\leq$ ” 确定了论域  $U$  划分上的一个偏序关系 ( $\leq$  为自反、反对称和传递的). 如果  $\pi_1 \leq \pi_2$  称划分  $\pi_1$  细于  $\pi_2$ .

以划分为运算对象, 定义并、交运算:

$\pi_1 \wedge \pi_2$  为比  $\pi_1, \pi_2$  都细的最粗的划分;

$\pi_1 \vee \pi_2$  为比  $\pi_1, \pi_2$  都粗的最细的划分.

由于划分的并运算相当于求上确界运算, 交运算相当于求下确界运算, 同时定义了划分之间的粗细关系, 这样就得到了一个格, 称为划分格.

由于建立决策树的过程是使用分支属性对实例集进行划分的过程, 所以将划分格引入到 DFDT 中, 给出 DFDT 实例集的划分的定义并期望得到动态模糊划分格.

**定义 2** 设  $(\vec{U}, \vec{D})$  为 DFDT 训练实例集. 则称  $(\vec{\pi}, \vec{\sigma}) = \{(\vec{X}_i, \vec{X}_i) \mid 1 \leq i \leq m\}$  为  $(\vec{U}, \vec{D})$  的一个划分. 若  $(\vec{\pi}, \vec{\sigma})$  满足以下条件:

- (1)  $(\vec{X}_i, \vec{X}_i)$  非空;
- (2)  $(\vec{X}_i, \vec{X}_i) \cap (\vec{X}_j, \vec{X}_j) = \emptyset \quad i \neq j$ ;
- (3)  $\cup \{(\vec{X}_i, \vec{X}_i) \mid 1 \leq i \leq m\} = (\vec{U}, \vec{D})$ .

定义划分间的偏序关系和并、交运算如下:

$(\vec{\pi}_1, \vec{\sigma}_1), (\vec{\pi}_2, \vec{\sigma}_2)$  为实例集  $(\vec{U}, \vec{D})$  的划分  $(\vec{\pi}_1, \vec{\sigma}_1) \leq (\vec{\pi}_2, \vec{\sigma}_2)$  成立, 当且仅当  $\forall (\vec{X}_i, \vec{X}_i) \in (\vec{\pi}_1, \vec{\sigma}_1), \exists (\vec{X}_j, \vec{X}_j) \in (\vec{\pi}_2, \vec{\sigma}_2)$  使得  $(\vec{X}_i, \vec{X}_i) \subseteq (\vec{X}_j, \vec{X}_j)$ . 称为  $(\vec{\pi}_1, \vec{\sigma}_1)$  严格细于  $(\vec{\pi}_2, \vec{\sigma}_2)$ , 记作  $(\vec{\pi}_1, \vec{\sigma}_1) < (\vec{\pi}_2, \vec{\sigma}_2)$ .

同时, 定义  $(\vec{\pi}_1, \vec{\sigma}_1), (\vec{\pi}_2, \vec{\sigma}_2)$  的交、并运算如下:

$(\vec{\pi}_1, \vec{\sigma}_1) \wedge (\vec{\pi}_2, \vec{\sigma}_2) = \{\sup((\vec{X}_i, \vec{X}_i) \wedge (\vec{X}_j, \vec{X}_j)) \mid (\vec{X}_i, \vec{X}_i) \in (\vec{\pi}_1, \vec{\sigma}_1), (\vec{X}_j, \vec{X}_j) \in (\vec{\pi}_2, \vec{\sigma}_2)\}$ ,  
 $(\vec{\pi}_1, \vec{\sigma}_1) \vee (\vec{\pi}_2, \vec{\sigma}_2) = \{\inf((\vec{X}_i, \vec{X}_i) \vee (\vec{X}_j, \vec{X}_j)) \mid (\vec{X}_i, \vec{X}_i) \in (\vec{\pi}_1, \vec{\sigma}_1), (\vec{X}_j, \vec{X}_j) \in (\vec{\pi}_2, \vec{\sigma}_2)\}.$

通过分析得知, 在 DFDT 中, 落入各结点的实例组成的集合之间构成偏序关系; 在 DFBDT 中, 落入相邻层各结点的实例组成的集合之间构成了半格. 在给出了动态模糊划分之后, 对于这两种类型的决策树各层对实例集的划分之间的关系, 有以下的定理:

**定理 1** 动态模糊决策树的各层对实例集的划分组成的集合是一个线性有序集.

**证明** DFDT 的构建过程是使用分支属性对实例集进行划分的过程, 所以下层对实例集的划分细于上层对实例集的划分  $(\vec{\pi}_j, \vec{\sigma}_j) < (\vec{\pi}_i, \vec{\sigma}_i)$  ( $j$  在 DFDT 中位于  $i$  的下层) 总是成立的, 即划分是线性的, 同时  $<$  构成了偏序关系, 所以各层对实例集的划分是一个线性有序集. 证毕.

**定理 2** 动态模糊决策树的各层对实例集的划分组成的集合是一个动态模糊划分格.

**证明** 设  $(\vec{U}_\pi, \vec{D}_\pi)$  为动态模糊决策树的各层对实例集的划分组成的集合, 由  $(\vec{U}_\pi, \vec{D}_\pi)$  偏序集知.

设任意  $(\vec{\pi}_i, \vec{\sigma}_i), (\vec{\pi}_j, \vec{\sigma}_j)$  分别为 DFDT 的第  $i, j$  ( $j < i$ ) 层确定的对实例集的划分, 且分别有  $(\vec{\pi}_i, \vec{\sigma}_i) = \{(\vec{X}_i, \vec{X}_i) \mid 1 \leq i \leq m\}$ ,  $(\vec{\pi}_j, \vec{\sigma}_j) = \{(\vec{X}_j, \vec{X}_j) \mid 1 \leq j \leq m\}$ , 则分为两种情况:

- (1)  $j$  为  $i$  的直接下层. 则对于  $\forall (\vec{X}_i, \vec{X}_i) \in (\vec{\pi}_i, \vec{\sigma}_i)$  有两种情况:

如果  $(\vec{X}_i, \vec{X}_i)$  为叶结点所确定, 那么构建 DFDT 过程中停止对其划分, 则  $\exists (\vec{X}_j, \vec{X}_j) \in \vec{\pi}_j, \vec{\sigma}_j$  使得  $(\vec{X}_i, \vec{X}_i) = (\vec{X}_j, \vec{X}_j)$ , 即  $(\vec{X}_i, \vec{X}_i) \supseteq (\vec{X}_j, \vec{X}_j)$ .

如果  $(\vec{X}_i, \vec{X}_i)$  为非终端结点, 那么在建树过程中, 通过选择属性对其进行了划分,  $(\vec{X}_i, \vec{X}_i) = (\vec{X}_j^1, \vec{X}_j^1) \cup (\vec{X}_j^2, \vec{X}_j^2) \cup \dots \cup (\vec{X}_j^{1(V_A, \vec{V}_A)}, \vec{X}_j^{1(V_A, \vec{V}_A)})$ , 其中  $(\vec{V}_A, \vec{V}_A)$  为分支属性  $(\vec{A}, \vec{A})$  的值域. 由于  $j$  为  $i$  的直接下层, 则选择属性  $(\vec{A}, \vec{A})$  对  $(\vec{X}_i, \vec{X}_i)$  的划分都落入  $(\vec{\pi}_j, \vec{\sigma}_j)$  中, 即  $\vec{X}_j^k, \vec{X}_j^k \in (\vec{\pi}_j, \vec{\sigma}_j)$ , 其中  $k = 1, 2, \dots$ ,

$|\tilde{V}_{\tilde{A}} \tilde{V}_{\tilde{A}}|$ .

由以上知 $(\tilde{\pi}_j \tilde{\sigma}_j)$  由叶结点所确定的划分和直接上层通过选择属性的划分所组成,有 $(\tilde{\pi}_j \tilde{\sigma}_j) < (\tilde{\pi}_i \tilde{\sigma}_i)$  成立,即 $(\tilde{\pi}_j \tilde{\sigma}_j) \wedge (\tilde{\pi}_i \tilde{\sigma}_i) = (\tilde{\pi}_j \tilde{\sigma}_j) \in (\tilde{U}_\pi \tilde{J}_\pi)$ ,  $(\tilde{\pi}_j \tilde{\sigma}_j) \vee (\tilde{\pi}_i \tilde{\sigma}_i) = (\tilde{\pi}_i \tilde{\sigma}_i) \in (\tilde{U}_\pi \tilde{J}_\pi)$ .

(2)  $j$  为  $i$  的非直接下层:

$j$  为  $i$  的非直接下层,则  $i$  与  $j$  之间组成了一个序列  $i, i+1, \dots, k, k+1, \dots, j$ , 由  $i, i+1, \dots, k, k+1, \dots, j$  为线性有序有:  $(\tilde{\pi}_j \tilde{\sigma}_j) < (\tilde{\pi}_{j-1} \tilde{\sigma}_{j-1}) < \dots < (\tilde{\pi}_{k+1} \tilde{\sigma}_{k+1}) < (\tilde{\pi}_k \tilde{\sigma}_k) < \dots < (\tilde{\pi}_{i+1} \tilde{\sigma}_{i+1}) < (\tilde{\pi}_i \tilde{\sigma}_i)$  成立,即 $(\tilde{\pi}_j \tilde{\sigma}_j) \wedge (\tilde{\pi}_i \tilde{\sigma}_i) = (\tilde{\pi}_j \tilde{\sigma}_j) \in (\tilde{U}_\pi \tilde{J}_\pi)$ ,  $(\tilde{\pi}_j \tilde{\sigma}_j) \vee (\tilde{\pi}_i \tilde{\sigma}_i) = (\tilde{\pi}_i \tilde{\sigma}_i) \in (\tilde{U}_\pi \tilde{J}_\pi)$ .

由(1)(2)知,对  $\forall (\tilde{\pi}_i \tilde{\sigma}_i), (\tilde{\pi}_j \tilde{\sigma}_j) \in (\tilde{U}_\pi \tilde{J}_\pi)$ , 有 $(\tilde{\pi}_i \tilde{\sigma}_i) \wedge (\tilde{\pi}_j \tilde{\sigma}_j) \in (\tilde{U}_\pi \tilde{J}_\pi)$ ,  $(\tilde{\pi}_i \tilde{\sigma}_i) \vee (\tilde{\pi}_j \tilde{\sigma}_j) \in (\tilde{U}_\pi \tilde{J}_\pi)$  成立,则 $(\tilde{U}_\pi \tilde{J}_\pi)$  为格,称为动态模糊划分格,记为  $\Pi(U)$ .

## 2 基于动态模糊划分格的离散化方法

动态模糊划分格处理的是对实例集这个有限确定区域的划分,而数值型属性的值域,也是一个确定的区域,而且对这个区域的划分也是有限的.

设动态模糊决策树 $(\tilde{T}, \tilde{T}) = \langle \langle \tilde{U}, \tilde{J} \rangle, \langle \tilde{C}, \tilde{C} \rangle, \langle \tilde{D}, \tilde{D} \rangle, \langle \tilde{V}, \tilde{V} \rangle, \langle \tilde{F}, \tilde{F} \rangle \rangle$ , 其中 $|\langle \tilde{U}, \tilde{J} \rangle| = n$ ,  $(\tilde{A}_i, \tilde{A}_i)$  为数值型属性. 要将属性 $(\tilde{A}_i, \tilde{A}_i)$  离散化,需要先对其进行排序操作,将 $(\tilde{A}_i, \tilde{A}_i)$  的取值按从小到大的顺序排列,在这个数据序列的间隔处插入分隔点,得到对 $(\tilde{A}_i, \tilde{A}_i)$  值域的划分 $\{[(\tilde{v}_1, \tilde{p}_1), (\tilde{v}_2, \tilde{p}_2)], [(\tilde{v}_2, \tilde{p}_2), (\tilde{v}_3, \tilde{p}_3)], \dots, [(\tilde{v}_{k-1}, \tilde{p}_{k-1}), (\tilde{v}_k, \tilde{p}_k)]\}$ . 对得到的每个区间用一个动态模糊数 $(\tilde{\alpha}, \tilde{\alpha})$  代替,这样就完成了对 $(\tilde{A}_i, \tilde{A}_i)$  的一次离散化. 对 $(\tilde{A}_i, \tilde{A}_i)$  值域的一个划分称为一个离散化方案,如果两个离散化方案得到的对 $(\tilde{A}_i, \tilde{A}_i)$  的值域划分完全相同,则称这两个离散化方案等价.

由于 $|\langle \tilde{U}, \tilde{J} \rangle| = n$ , 所以 $\max(|\tilde{V}_{\tilde{A}_i} \tilde{V}_{\tilde{A}_i}|) = n$ , 由于同一个取值间隔可以存在多个分隔点,同时得到的离散化方案是等价的,所以规定一个取值间隔只允许存在一个分隔点,则分隔点的数目可能为  $0, 1, 2, \dots, n-1$ . 当分隔点数为  $j$  时,设置分隔点的方法有  $C_{n-1}^j$  种,则属性 $(\tilde{A}_i, \tilde{A}_i)$  不等价的离散化方案有  $\sum_{j=0}^{n-1} C_{n-1}^j$ , 如果在实例集中共有  $m$  个数值型属性,则对应的离散化后的实例集最多有  $2^{(n-1)m}$  种. 对于给定的实例集是有限的,不论采用哪种方法进行离散化,所产生的离散化后的实例集都在一个有限的范围内,即对实例集的离散化集合是一个有限域.

设实例集的条件属性集为 $(\tilde{C}, \tilde{C}) = \{(\tilde{A}_1, \tilde{A}_1), (\tilde{A}_2, \tilde{A}_2), \dots, (\tilde{A}_n, \tilde{A}_n)\}$ , 属性 $(\tilde{A}_i, \tilde{A}_i) \in (\tilde{C}, \tilde{C})$  的值域为 $(\tilde{V}_{\tilde{A}_i} \tilde{V}_{\tilde{A}_i})$ , 对单个数值型属性 $(\tilde{A}_i, \tilde{A}_i)$  的一个离散化方案是 $(\tilde{V}_{\tilde{A}_i} \tilde{V}_{\tilde{A}_i})$  的一个划分 $(\tilde{\pi}_i \tilde{\sigma}_i)$ .

把经过属性过滤后的实例集 $(\tilde{U}, \tilde{J})$  记为 $(\tilde{U}_N \tilde{J}_N)$ ,  $(\tilde{A}_i, \tilde{A}_i)$  为 $(\tilde{U}_N \tilde{J}_N)$  的任一条件属性,且 $(\tilde{A}_i, \tilde{A}_i)$  为数值型的,则对 $(\tilde{U}_N \tilde{J}_N)$  的离散化是多属性的离散化.

设 $(\tilde{\pi}_1 \tilde{\sigma}_1) = \{(\tilde{\pi}_1^1 \tilde{\sigma}_1^1), (\tilde{\pi}_1^2 \tilde{\sigma}_1^2), \dots, (\tilde{\pi}_1^n \tilde{\sigma}_1^n)\}$ ,  $(\tilde{\pi}_2 \tilde{\sigma}_2) = \{(\tilde{\pi}_2^1 \tilde{\sigma}_2^1), (\tilde{\pi}_2^2 \tilde{\sigma}_2^2), \dots, (\tilde{\pi}_2^n \tilde{\sigma}_2^n)\}$  是对实例集 $(\tilde{U}_N \tilde{J}_N)$  的两种离散化方案,其中 $(\tilde{\pi}_i^1 \tilde{\sigma}_i^1)$  是 $(\tilde{\pi}_1 \tilde{\sigma}_1)$  中对属性 $(\tilde{A}_i, \tilde{A}_i)$  的离散化方案,  $(\tilde{\pi}_j^2 \tilde{\sigma}_j^2)$  是 $(\tilde{\pi}_2 \tilde{\sigma}_2)$  中对属性 $(\tilde{A}_j, \tilde{A}_j)$  的离散化方案.

由于 $(\tilde{\pi}_j^i \tilde{\sigma}_j^i)$  是对属性 $(\tilde{A}_j, \tilde{A}_j)$  的离散化方案,所以在离散化中的偏序关系定义为:称离散化方案 $(\tilde{\pi}_i \tilde{\sigma}_i) \leq (\tilde{\pi}_j \tilde{\sigma}_j)$ , 当且仅当对实例集 $(\tilde{U}_N \tilde{J}_N)$  中的每个属性 $(\tilde{A}_k, \tilde{A}_k)$  都满足 $(\tilde{\pi}_k^i \tilde{\sigma}_k^i) \leq (\tilde{\pi}_k^j \tilde{\sigma}_k^j)$ ,  $k = 1, 2, \dots, n$ . 其中 $(\tilde{\pi}_k^i \tilde{\sigma}_k^i)$  为 $(\tilde{\pi}_i \tilde{\sigma}_i)$  中对 $(\tilde{V}_{\tilde{A}_k} \tilde{V}_{\tilde{A}_k})$  的划分.  $\leq$  定义了离散化方案之间的粗细程度,为偏序关系.

$(\tilde{\pi}_k^i \tilde{\sigma}_k^i) \leq (\tilde{\pi}_k^j \tilde{\sigma}_k^j)$  说明 $(\tilde{\pi}_i \tilde{\sigma}_i)$  中将属性 $(\tilde{A}_k, \tilde{A}_k)$  的值划分到了一个区间,则在 $(\tilde{\pi}_j \tilde{\sigma}_j)$  中对属性 $(\tilde{A}_k, \tilde{A}_k)$  值的划分一定属于同一个区间,而且在 $(\tilde{\pi}_i \tilde{\sigma}_i)$  中对属性 $(\tilde{A}_k, \tilde{A}_k)$  的值的两个或多个相邻分割区间合并为 $(\tilde{\pi}_j \tilde{\sigma}_j)$  中对属性 $(\tilde{A}_k, \tilde{A}_k)$  值的一个划分区间. 由于 $(\tilde{\pi}_i \tilde{\sigma}_i) \leq (\tilde{\pi}_j \tilde{\sigma}_j)$ , 则称 $(\tilde{\pi}_i \tilde{\sigma}_i)$  为 $(\tilde{\pi}_j \tilde{\sigma}_j)$  的细化.

对实例集 $(\tilde{U}_N \tilde{J}_N)$  的离散化方案的交、运算为: $(\tilde{\pi}_i \tilde{\sigma}_i) \cap (\tilde{\pi}_j \tilde{\sigma}_j) = \{(\tilde{\pi}_1^i \tilde{\sigma}_1^i) \wedge (\tilde{\pi}_1^j \tilde{\sigma}_1^j), (\tilde{\pi}_2^i \tilde{\sigma}_2^i) \wedge (\tilde{\pi}_2^j \tilde{\sigma}_2^j), \dots, (\tilde{\pi}_n^i \tilde{\sigma}_n^i) \wedge (\tilde{\pi}_n^j \tilde{\sigma}_n^j)\}$ ,  $(\tilde{\pi}_i \tilde{\sigma}_i) \cup (\tilde{\pi}_j \tilde{\sigma}_j) = \{(\tilde{\pi}_1^i \tilde{\sigma}_1^i) \vee (\tilde{\pi}_1^j \tilde{\sigma}_1^j), (\tilde{\pi}_2^i \tilde{\sigma}_2^i) \vee (\tilde{\pi}_2^j \tilde{\sigma}_2^j), \dots, (\tilde{\pi}_n^i \tilde{\sigma}_n^i) \vee (\tilde{\pi}_n^j \tilde{\sigma}_n^j)\}$ .

$(\vec{\pi}_i, \vec{\pi}_i) \cap (\vec{\pi}_j, \vec{\pi}_j)$  对实例集  $(\vec{U}_N, \vec{U}_N)$  的划分为比  $(\vec{\pi}_i, \vec{\pi}_i)$ 、 $(\vec{\pi}_j, \vec{\pi}_j)$  都细的最粗的划分  $(\vec{\pi}_i, \vec{\pi}_i) \cup (\vec{\pi}_j, \vec{\pi}_j)$  为比  $(\vec{\pi}_i, \vec{\pi}_i)$ 、 $(\vec{\pi}_j, \vec{\pi}_j)$  都粗的最细的划分。

定义了偏序关系  $\leq$ , 离散化方案的交、并运算, 就得到了以对实例集  $(\vec{U}_N, \vec{U}_N)$  的各种离散化方案为元素的格结构, 称为动态模糊离散格<sup>[10]</sup>。

由于各种离散化方案都是动态模糊离散格中的元素, 因此对实例集  $(\vec{U}_N, \vec{U}_N)$  的离散化问题就转化为对动态模糊离散格的搜索问题。在对属性  $(\vec{A}_i, \vec{A}_i)$  离散化时, 在  $(\vec{V}_{A_i}, \vec{V}_{A_i})$  中有很多的分隔点, 其中有的分隔点将其删除以后并不影响离散化的结果, 则此结点为冗余分隔点。

**定义 3** 对属性  $(\vec{A}_i, \vec{A}_i)$ , 有分隔点集合  $(\vec{S}, \vec{S}) = \{(\vec{S}_1, \vec{S}_1), (\vec{S}_2, \vec{S}_2), \dots, (\vec{S}_k, \vec{S}_k)\}$ 。如果将  $(\vec{S}_i, \vec{S}_i) \in (\vec{S}, \vec{S})$  删除, 将其分开的两个区间合并起来, 两个区间内的属性值用同一个动态模糊数  $(\vec{\alpha}, \vec{\alpha})$  来代替, 得到的实例集是一致的, 则把分隔点  $(\vec{S}_i, \vec{S}_i) \in (\vec{S}, \vec{S})$  称为冗余分隔点。

**定义 4** 将一个属性  $(\vec{A}_i, \vec{A}_i)$  的离散化方案中冗余分隔点数称为  $(\vec{A}_i, \vec{A}_i)$  的分隔点级别。

显然, 分隔点级别越高, 则对此属性的离散化中对应越多的冗余分隔点。

对属性的离散化的目标是在保持离散化结果一致性的前提下, 使冗余分隔点数降到最低<sup>[11]</sup>。基本的步骤为: 先计算每个属性的分隔点级别, 将各个属性按分隔点级别从小到大排序, 得到一个要删除冗余分隔点的属性序列, 然后逐轮删除冗余分隔点, 每一次操作只删除一个冗余分隔点。其算法描述为:

- (1) 求每个离散化属性的分隔点级别以及可以删除的分隔点集合;
- (2) 将各个属性按分隔点级别从小到大排序, 得到一个属性序列;
- (3) 依次处理属性序列中的每一个属性, 如果排在第  $i$  位的属性有冗余分隔点, 则合并此分隔点相邻的区间, 判断此时实例集的一致性。如果一致, 则第  $i$  位属性的冗余分隔点数减 1, 更新实例集; 否则保留该分隔点。转 (3) 进行下一轮的冗余分隔点删除操作;
- (4) 如果属性序列中各个属性都没有冗余分隔点, 算法结束, 得到离散化的实例集。

从算法中可以看出, 删除冗余分隔点的操作是对离散化方案不断加粗的过程, 得到一个分隔点数不断减少的离散化方案序列, 序列中相邻两个离散化方案满足偏序关系  $\leq$ , 序列中的最后一个方案即为实例集  $(\vec{U}_N, \vec{U}_N)$  的离散化方案。

在基于动态模糊划分格的离散化中, 首先考虑的是实例集的一致性, 保持一致性能限定离散化对原实例集的近似程度, 同时避免了基于信息熵的离散化方法所带来的数据冲突问题, 而且只根据属性值来进行操作, 没有人为地设置算法结束的准则, 减少了在实例集离散化过程中人为因素的影响。

### 3 DFDT 中缺失值的分类处理方法

在 DFDT 中, 对于属性值的缺少, 应分为以下的 3 种情况来分别处理:

- (1) 属性值对于分类没有信息贡献的属性;
- (2) 属性值对于分类存在部分的信息贡献;
- (3) 属性值为重要信息来源, 对于分类起着重要的作用。

下面以例子来说明 DFDT 对这 3 种类型缺失值属性的处理方法。假设在“判定流感发病率高低”的数据集中有以下属性: 姓名, 性别, 地域, 年龄, 发病时间, 温度, 等。其中“姓名”属性与分类结果没有必然的联系, 即对于此分类来说“姓名”属性没有任何的信息贡献。对于此类属性有两种处理方法: (1) 直接从数据集中删除此属性; 这种处理方法是由处理不完备数据集的第一个方法而来, 同时它也继承了“删除元组”方法的缺点。(2) 将此类方法进行类似离散化操作, 如给所有的实例的“姓名”属性赋一个动态模糊数, 如“姓名” =  $0.5$ , 这种处理方法的优点是保持了数据集的完整性。

“时间”属性是对分类有部分信息贡献的, 比如从“2005-10-10”属性值中可能推知 10 月份是流感的高发期, 对于此类的属性首先在数据集中简化属性值, 只保留对于分类有用的信息。经过简化后的属性, 其属性值对于分类有着完全的贡献, 即属性从对分类有部分贡献转化为重要属性。当经过简化后的属性值缺失时, 对其处理方式与对于重要属性的处理方式是相同的。

对于重要属性的处理又分为在训练实例集中的处理和测试集中的处理两种情况<sup>[12, 13]</sup>。在训练集中的处理方法有以下几种:

(1) 忽略包含有缺失值属性的实例;

(2) 应用其他实例中的信息给属性赋值: 假设某实例  $A$  属性的值未知, 此实例属于类别  $C$ , 根据属于  $C$  类别的其他实例中  $A$  属性的值来给其赋值;

(3) 使用决策树方法得到属性值: 假设训练集为  $S$ , 其中属性  $A$  的值部分未知, 则从  $S$  中取出属性  $A$  的值已知的实例构成  $S$  的子集  $S'$ , 在  $S'$  中进行构建决策树的操作, 令原来的分类属性作为一般的属性处理, 令  $A$  作为  $S'$  中的分类属性, 生成决策树, 产生规则, 利用规则能够给  $S-S'$  中的属性  $A$  赋值;

(4) 给属性赋值 “unknown”.

这些处理的方法各有优点和缺点. 第一种方法虽然简单易行, 但是由于忽略了部分实例, 势必造成信息的丢失, 破坏信息的完备. 第三种方法步骤复杂, 而且只适用于属性值缺少比较少和数据集中缺失值属性较少的情况. 第四种方法将 “unknown” 作为一个属性值来处理, 直接改变了数据集中属性的值域. 第二种方法虽然可能受到主观方面的影响, 而且实例与实例之间的相关度的标准比较难以确定, 但是此方法被证明具有很高的实用性. 由于另外 3 种方法的缺点比较明显, 在 DFDT 中采用第二种方法对训练集中的重要缺失值属性进行处理.

对训练集中的缺失值属性处理完后, 即可建立决策树, 并对其进行测试. 使用测试实例集, 并对测试实例集中的实例进行分类, 若实例中含有属性值缺少的情况, 那么分类立即变为不可能, 这就需要对测试集中的属性值缺少的情况进行处理. 对于测试集中缺失值属性的处理方法有很多种, 如代理属性分割、动态路径生成、懒惰决策树法等. 在 DFDT 中, 采取的操作类似于在训练实例集中的操作, 应用其他实例中的信息来给属性赋值.

## 4 结论

综上所述, 本文讨论了动态模糊决策树的属性算法. 通过动态模糊二叉决策树的介绍引出了动态模糊决策树中各结点以及各层对实例集划分之间的关系. 由于划分格是对论域的划分, 在此引入划分格, 通过其定义了动态模糊划分格, 给出了关于动态模糊决策树各层对实例集的划分组成的集合的定理.

## [参考文献] (References)

- [1] Li Fanzhang. Dynamic Fuzzy Logic and Its Applications[M]. New York: Nova Science Publishers, 2008.
- [2] 蔡晨. 动态模糊决策树模型及应用研究[D]. 苏州: 苏州大学计算机科学与技术学院, 2008.  
Cai Chen. Research and application on the dynamic fuzzy decision tree learning[D]. Suzhou: School of Computer Science and Technology, Soochow University, 2008. (in Chinese)
- [3] 尹阿东, 谢霖铨, 龙誉, 等. 动态决策树算法研究[J]. 计算机工程与应用, 2004(33): 103-105, 132.  
Yin Adong, Xie Linquan, Long Yu, et al. Researches on dynamic algorithm of decision trees[J]. Computer Engineering and Applications, 2004(33): 103-105, 132. (in Chinese)
- [4] Kazuaki Aoki, Toshiharu Watanabe, Mineichi Kudo. Design of decision tree using class-dependent features subsets[J]. Systems and Computers in Japan, 2005, 36(4): 37-47.
- [5] 李凡长, 刘贵全, 余玉梅. 动态模糊逻辑引论[M]. 昆明: 云南科技出版社, 2005.  
Li Fanzhang, Liu Guiquan, She Yumei. An Introduction to Dynamic Fuzzy Logic[M]. Kunming: Yunnan Science and Technology Press, 2005. (in Chinese)
- [6] Margaret H Dunham. 数据挖掘教程[M]. 郭崇慧, 田凤占, 靳晓明, 译. 北京: 清华大学出版社, 2005.  
Margaret H Dunham. Data Mining Tutorial[M]. Guo Conghui, Tian Fengzhan, Jin Xiaoming, Translated. Beijing: Tsinghua University Press, 2005. (in Chinese)
- [7] 孙超利, 张继福. 基于属性-值对的信息增益优化算法[J]. 太原科技大学学报, 2005, 26(3): 199-202.  
Sun Chaoli, Zhang Jifu. The information gain algorithm based on attribute-value on pairs[J]. Journal of Taiyuan University of Science and Technology, 2005, 26(3): 199-202. (in Chinese)
- [8] Yao Y Y, Yao J T. Granular computing as a basic for consistent classification problems[J]. Communications of Institute of Information and Computing Machinery, 2002, 5(2): 101-106.

(下转第 69 页)

- ary Computation[M]. New York: IEEE Press, 2006: 1 670-1 677.
- [7] Michael Geis, Martin Middendorf. A particle swarm optimizer for finding minimum free energy RNA secondary structures [C]// Swarm Intelligence Symposium 2007. New York: IEEE Press, 2007: 1-8.
- [8] 陈自郁,何中市,何静媛. 预测 RNA 二级结构离散粒子群优化算法[J]. 深圳大学学报: 理工版, 2009, 26(3): 272-277.  
Chen Ziyu, He Zhongshi, He Jingyuan. Discrete particle swarm optimization algorithm for RNA secondary structures [J]. Journal of Shenzhen University: Science and Engineering Edition, 2009, 26(3): 272-277. (in Chinese)
- [9] 胡桂武,彭宏. 基于免疫粒子群集成的 RNA 二级结构预测算法[J]. 计算机工程与应用 2007, 43(3): 26-29.  
Hu Guiwu, Peng Hong. Algorithm based on immune PSO ensemble for predicting RNA secondary structure [J]. Computer Engineering and Applications, 2007, 43(3): 26-29. (in Chinese)
- [10] 何静媛,邹东升,何中市. RNA 二级结构预测的自适应鱼群算法模型[J]. 系统仿真学报 2010, 22(6): 1 370-1 374.  
He Jingyuan, Zou Dongshen, He Zhongshi. Self-adaptive artificial fish swarm algorithm for RNA secondary structure prediction [J]. Journal of System Simulation, 2010, 22(6): 1 370-1 374. (in Chinese)
- [11] Jun Yu, Changhai Zhang, Yuanning Liu, et al. Simulating the folding pathway of RNA secondary structure using the modified ant colony algorithm[J]. Bionic Engineering, 2001(7): 382-389.
- [12] Muzaffar M, Eusuff, Kevin E Lansey. Optimization of water distribution network design using the shuffled frog leaping algorithm[J]. Journal of Water Resources Planning and Management, 2003, 129(3): 210-225.
- [13] 韩毅,蔡建湖,周根贵,等. 随机蛙跳算法的研究进展[J]. 计算机科学, 2010, 37(7): 16-19.  
Han Yi, Cai Jianhu, Zhou Gengui, et al. Advances in shuffled frog leaping algorithm [J]. Computer Science, 2010, 37(7): 16-19. (in Chinese)
- [14] 王翼飞,史定华. 生物信息学-智能优化算法及其应用[M]. 北京: 化学工业出版社, 2006: 178-210.  
Wang Yifei, Shi Dinghua. Bioinformatics-Intelligent Optimization Algorithms and Its Application [M]. Beijing: Chemical Industry Press, 2006: 178-210. (in Chinese)
- [15] Muzaffar Eusuff, Kevin Lansey, Fayzul Pasha. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization[J]. Engineering Optimization, 2006, 8(2): 129-154.
- [16] Elbeltagi E, Hegazy T, Grierson D. Comparison among five evolutionary-based optimization algorithms[J]. Advanced Engineering Informatics, 2005, 19(1): 43-53.
- [17] Muker M, Mathews D H, Turner D H. Algorithms and thermodynamics for RNA secondary structure prediction: a practical guide in RNA biochemistry and biotechnology [M]// Barciszewski J, Clark B F C, eds. NATO ASI Series. Dordrecht, NL: Kluwer Academic Publishers, 1999: 11-43.
- [18] Jens Reeder, Robert Giegerich. Design, implementation and evaluation of a practical pseudoknots folding algorithm based on thermodynamics [J]. BMC Bioinformatics, 2004(5): 104.

[责任编辑: 严海琳]

(上接第 62 页)

- [9] 李明仑, 李凡长. DFL 的格结构及其应用研究[J]. 计算机应用与软件 2007 24(9): 145-150.  
Li Minglun, Li Fanzhang. Research on lattice structure of DFL and its application [J]. Computer Applications and Software, 2007 24(9): 145-150. (in Chinese)
- [10] 李明仑. 基于动态模糊格的决策树理论及应用研究[D]. 苏州: 苏州大学计算机科学与技术学院 2008.  
Li Minglun. Research and application on theory of decision tree based on dynamic fuzzy lattice [D]. Suzhou: School of Computer Science and Technology, Soochow University, 2008. (in Chinese)
- [11] Iun H Witten, Eibe Frank. Data Mining Paractical Machine Learning Tools and Techniques with Java Implementations [M]. Beijing: China Machine Press, 2003.
- [12] Zoubin Ghahramani, Michael I Jordan. Supervised learning from incomplete data via an EM approach [J]. Advances in Neural Information Processing Systems, 1993(6): 120-127.
- [13] 刘鹏, 雷蕾, 张雪凤. 缺失数据处理方法的比较研究[J]. 计算机科学, 2004, 31(10): 155-156, 174.  
Liu Peng, Lei Lei, Zhang Xuefeng. A comparison study of missing value processing methods [J]. Computer Science, 2004, 31(10): 155-156, 174. (in Chinese)

[责任编辑: 严海琳]