文章编号:1001 - 9081(2011)05 - 01374 - 04
doi:10.3724/SP.J.1087.2011.01374

面向多维时间序列的过程决策树模型

刘　栋[1],宋国杰[2]
(1.河南师范大学 计算机与信息技术学院,河南 新乡 453007;
2.北京大学 信息科学技术学院,北京 100871)
(lnld429@126.com)

摘　要:为解决多维时间序列的分类并获取易于理解的分类规则,引入了时序熵的概念及构造时序熵的方法,基于属性选择和属性值划分两方面扩展了决策树模型。并给出了两种构造多维时间序列分类的决策树模型算法。最后,采用移动客户流失的真实数据,对过程决策树进行测试,展示了方法的可行性。

关键词:多维时间序列分类;熵;决策树;分类规则
中图分类号:TP311.13　　**文献标志码**:A

Process decision tree model based on multi-dimensional time series

LIU Dong[1], SONG Guo-jie[2]
(1. *College of Computer and Information Technology, Henan Normal University, Xinxiang Henan* 453007, *China*;
2. *College of Electronics Engineering and Computer Science, Peking University, Beijing* 100871, *China*)

Abstract: To solve the classification problem of multi-dimensional time series and obtain understandable classification rules, the concept of time series entropy and the method of structuring time series entropy were introduced. And the decision tree model was expanded based on both attribute selection and attribute value. Two algorithms for structuring decision tree model of multi-dimensional time series classification were presented. Finally, process decision tree was tested on mobile customer churn data, and the feasibility of the proposed method was demonstrated.

Key words: multi-dimensional time series classification; entropy; decision tree; classification rule

0　引言

分类是人工智能领域中一个重要的研究问题,已有大量的研究成果,如 K-近邻方法、人工神经网络、支持向量机、决策树方法等。传统方法的共同特点是可以很好地处理非过程性数据分类问题。然而,实际应用中的原始数据往往是在某一时间段内采样得到的,并具有连续过程特征的时间序列数据集。将序列数据所蕴含的过程特征融入分类任务,能够大幅度提高分类的精度。如在客户流失预测问题中[1],将移动用户的电话呼叫转移量明显增加和呼叫次数明显减少的过程特征提取出来用于流失预测任务,对分类精度的提高具有非常积极的意义。本文所研究的过程决策树模型就是针对这一问题而提出的。

本文尝试对传统决策树在属性选择和属性值划分两个方面进行扩展,建立一种适用于多维时间序列分类问题[2]的过程决策树模型。在属性选择上,本文给出时序熵的概念,代替传统决策树算法中的信息熵,度量属性值为时间序列的数据所包含的分类信息,从而选择合适的属性作为决策树的分支节点。对于时间序列属性的划分,本文采用基于动态时间弯曲(Dynamic Time Warping, DTW)[3]距离度量的模糊K-means方法[4]对时间序列属性进行聚类,以便在构造树的时候进行节点分支。

在以上两方面扩展的基础上,本文采用两种思路构造两种不同的过程决策树模型框架。本文尝试将其应用到移动通信的客户流失预测问题上,并与传统 ID3 算法[5]在这一问题上的应用进行比较。以验证该模型在时间序列分类问题中的效果。

1　相关工作

面向过程的多维时间序列分类的研究工作分为两类:第一类是与领域无关的方法。该类方法的分类模型采用 k-近邻搜索等基于距离的算法。而影响这一类方法效果的因素往往不是所采用的分类模型,而是对时间序列的索引方法和时间序列的距离度量[6]。第二类时间序列研究集中在领域相关的分类方法。这类方法主要是利用领域知识,对每条时间序列提取等长的特征向量,然后利用一般分类算法进行分类[7]。

多维时间序列的分类问题给基于距离的分类方法带来了难题,L_p 距离[8]和动态时间归整(Dynamic Time Warping, DTW)等距离度量方法很难扩展到多维时间序列的度量上。另外,由于多维时间序列在时间维度上的特点,很难用传统的主成分分析方法对其进行降维。因此实际应用中往往采用领域相关的分类方法。利用领域知识,提取特征,消除数据在时间上的纬度,转化为普通的非过程的多维数据在进行分类。然而,这类方法过于依赖领域知识,往往容易忽略时间序列数据带有的过程特性。因此,在领域知识或相关经验缺乏的情况下,这类方法很难取得较好的效果。

在许多实际应用中,过程神经元网络方法[9]可以较好地解决多维时间序列的分类问题。但遗憾的是,过程神经元网络与其他的人工神经元网络一样,有着典型的黑箱性。训练好的神经元网络可以用来对新的应用实例进行分类预测,但

收稿日期:2010 - 10 - 26;**修回日期**:2011 - 01 - 13。　**基金项目**:国家自然科学基金资助项目(60703066)。
作者简介:刘栋(1976 -),男,河南原阳人,讲师,博士研究生,主要研究方向:决策支持系统、机器学习;　宋国杰(1975 -),男,河南原阳人,副教授,博士,主要研究方向:数据挖掘、机器学习。

是网络结构本身对于使用者来讲是没有意义的。而在许多领域中,分类的规则也同样重要,它可以给该领域的专家提供更多、更有意义的信息。

目前已有的决策树学习算法有许多种,现有的决策树算法很难直接应用到多维时间序列的分类问题中。原因在于现有的算法要求“属性-值”对中的值,为已经划分好的少数离散值(ID3 算法)或存在简洁有效的实数属性值(C4.5 算法)[10]。而当属性的取值是时间序列时,这些决策树算法就很难直接对其进行学习和分类。

2　过程决策树模型建模

2.1　时序熵及其计算

在决策树模型算法中,借助信息论中熵的概念,通过计算当前所有属性的信息增益,也就是每种属性对于最终分类提供的信息量,选择属性作为当前的分支节点[11]。下面给出信息增益的概念及计算方法[12]:

假设 D 表示一数据集(X,y),$X \in \mathbf{R}^n$,$y \in C_1, C_2, \cdots, C_m$。$D$ 中元组分类所需期望信息为:

$$Info(D) = -\sum_{i=1}^{m} p_i \operatorname{lb} p_i$$

设属性 A 可将 D 划分为 v 个子集$\{D_1, D_2, \cdots, D_v\}$,基于属性 A 划分对 D 的元组分类所需的期望信息为:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

信息增益为:

$$Gain(A) = Info(D) - Info_A(D)$$

上述信息增益的计算要求属性 A 可将集合 D 划分成有限的 m 个子集。也就是说,在实际应用中属性 A 的取值应该是少数离散的值。这也就限制了这种信息的计算方法无法应用到以时间序列为取值的属性选择上。下面通过对信息熵 $Info$ 进行改进和扩展,从而找到可以计算时间序列属性分类信息增益的方法。

对上面定义中属性 A 划分对 D 的元组分类所需的期望信息的计算公式进行变形推导:

$$\begin{aligned}
Info_A(D) &= \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j) = \\
&\sum_{j=1}^{v} \left[P(A(X) = j) \times \left(-\sum_{i=1}^{m} P(y = C_i \mid A(X) = j) \operatorname{lb} P(y = C_i \mid A(X) = j) \right) \right] = \\
&-\sum_{j=1}^{v} \sum_{i=1}^{m} \left[P(A(X) = j) \times P(y = C_i \mid A(X) = j) \operatorname{lb} P(y = C_i \mid A(X) = j) \right] = \\
&-\sum_{j=1}^{v} \sum_{i=1}^{m} \left[P(y = C_i, A(X) = j) \operatorname{lb} \frac{P(y = C_i \mid A(X) = j)}{P(A(X) = j)} \right] = \\
&-\sum_{j=1}^{v} \sum_{i=1}^{m} \left[P(A(X) = j \mid y = C_i) P(y = C_i) \operatorname{lb} \frac{P(A(X) = j \mid y = C_i) P(y = C_i)}{P(A(X) = j)} \right]
\end{aligned}$$

下面将 $A(X)$ 的离散化取值 $1,2,\cdots,v$ 扩展到 $\mathbf{R}$ 上的连续区间$[a, b]$,设 $x = A(X)$,$p(x)$ 为 x 在$[a, b]$上的概率密度函数,则有:

$$Info_A(D) = -\int_a^b \sum_{i=1}^{m} \left[p(x \mid y = C_i) P(y = C_i) \times \operatorname{lb} \frac{p(x \mid y = C_i) P(y = C_i)}{p(x)} \right] \mathrm{d}x$$

上面给出的信息熵计算公式已经可以计算值域为实数的属性划分 D 的到的期望信息。那么将上面的公式进一步扩展。将连续数值型属性的取值由实数 x 值扩展为时间函数 $x(t)$。则有:

$$p(x) \rightarrow p(x(t))$$

$$p(x \mid y = C_i) \rightarrow p(x(t) \mid y = C_i)$$

那么信息熵的计算公式就可以扩展为时间序列属性的期望信息概念。

$$Info_A(D) = \int_t \left\{ -\int_a^b \sum_{i=1}^{m} \left[p(x(t) \mid y = C_i) P(y = C_i) \times \operatorname{lb} \frac{p(x(t) \mid y = C_i) P(y = C_i)}{p(x(t))} \right] \mathrm{d}x(t) \right\}$$

将上式所表示的概念命名为时序熵。注意,上式给出的是时序熵的泛化表示,时域运算符号可以有多种选择,例如对时间 t 求积分,取时间段 $[0,T]$ 内的最大值等。

另外,概率 $p(x(t))$ 和 $p(x(t) \mid y = C_i)$ 也是数学上的泛化表达。实际应用中很难直接由数据得到这两个概率分布的解析表达,因此要采取一些统计方法对其值进行逼近。下面给出近似计算时序熵的方法。

首先,对每个时间片断上的 A 属性取值进行离散划分。统计每一个时间片断 t 上的 A 属性取值的概率分布;然后取时域运算符号 $\int_t$ 为累加后取算术平均运算 $\frac{1}{T}\sum$。则时序熵的计算公式如下:

$$Info_A(D) = -\frac{1}{T} \sum_{t=0}^{T} \sum_{j=1}^{v} \sum_{i=1}^{m} \left[p(x(t) = j \mid y = C_i) \times P(y = C_i) \operatorname{lb} \frac{p(x(t) = j \mid y = C_i) P(y = C_i)}{p(x(t) = j)} \right]$$

这里对时间片断 t 上的 A 属性取值的概率分布的近似是通过分桶统计进行离散近似,当然,还可以采用其他方法,对概率分布函数进行拟合。另外,时域运算符也可以采用其他形式。

2.2　过程决策树模型的构建

本节给出两种建造过程决策树的方法,两种方法的区别在于:方法 1 在建造树之前对属性进行划分,而方法 2 则在建造树的过程中对属性进行划分。在两种方法中都采用模糊 K-means方法对时间序列属性进行聚类划分,模糊K-means聚类的算法省略。

2.2.1　预属性划分构建算法

首先对多维时间序列的每一维通过模糊K-means方法进行聚类。对于每条实例中每一维的时间序列,将隶属度最大的簇标号标记为该属性取值。之后采用传统的 ID3 算法框架建树,所不同的是,树的每一个分支节点需要记录所有分支的中心序列。

算法 1　PDT_Build1。

输入 训练数据 *Train*，包含 *N* 条实例，每条实例有 *Dim* 维；记录每一维划分数的 *clusters_num*；记录每一维中每一个划分的中心序列的结构数组 *centers*；训练数据的分类标号 *targets*；允许分错的实例个数 *inc_node*。

输出 过程决策树的根节点 *Root*。

1） 计算分类标号 *targets* 的信息熵 I

2） 计算每一维划分后的信息熵 $Info(A_i)$

3） 计算每一维的信息增益 $Gain(A_i) = I - Info(A_i)$

4） 选择信息增益最大的属性 A_i 作为 Root 的决策属性，并记录 A_i 的所有簇的中心序列

5） for $j = 1:clusters\_num(A_i)$

6） *Root* 下加一个分支节点，对应测试属性 A_i 的数列是否接近 $centers(A_i)$

7） 另 $Train(A_i = j)$ 为 *train* 中 A_i 属性属于第 j 个簇的子集

8） if $Train(A_i = j)$ 为空

9） 在分支下加一个叶节点

10） 类标号为 *targets* 中最普遍的类标号

11） else 新分支下加一个子树 $PDT\_Build1(Train(A_i = j), clusters\_num, centers, targets, inc\_node)$

12） 返回 *Root*

这种方法将属性的划分与树的构造分割开来。属性的选择仍然使用信息增益的度量方法。算法的输入并非时间序列，但是学习过后的过程决策树在对新的实例进行分类时，输入是多维时间序列。

2.2.2 后属性划分构建算法

用时序熵代替离散信息熵，从而得到每种属性的信息增益。选择信息增益最大的属性作为分支标准，然后对该属性进行聚类划分，在进行分支的同时生成决策树。算法具体构建过程描述如下：

算法 2 PDT_Build。

输入 训练数据 *Train*，包含 *N* 条实例，每条实例有 *Dim* 维，每一维是一条时间序列；记录每训练数据的分类标号 *targets*；允许分错的实例个数 *inc_node*。

输出 过程决策树的根节点 *Root*。

1） 创建根节点 *Root*

2） 开始计算分类标号 *targets* 的信息熵 I

3） 计算每一维划分后的时序熵 $Info\_TS(A_i)$

4） 计算每一维的信息增益 $Gain(A_i) = I - Info\_TS(A_i)$

5） 选择信息增益最大的属性 A_i 作为 *Root* 的决策属性

6） 对 *Train* 的 A_i 维进行聚类 $Fuzzy\_Kmeans(Train(A_i))$

7） 记录 A_i 的所有簇的中心序列

8） for $j = 1:clusters\_num(A_i)$

9） *Root* 加一个分支节点，测试属性 A_i 是否接近 $centers(A_i)$

10） 另 $Train(A_i = j)$ 为 *train* 中 A_i 属性属于第 j 个簇的子集

11） if $Train(A_i = j)$ 为空

12） 加一叶节点，类标号为 *targets* 中最普遍的类标号

13） else 加一子树 $PDT\_Build1(Train(A_i = j), targets, inc\_node)$

14） 返回 *Root*

该方法首先选择属性，在对其进行划分。这样做的一个好处是可以减少每次聚类划分的样本数量，从而减少花费在聚类上的时间，提高效率。

2.3 基于过程决策树模型的分类

通过上述方法对训练样本进行学习。当遇到新的实例时，可以将这条实例（多维时间序列数据）直接作为决策树的输入进行分类。分类算法如下：

算法 3 Use_PDT。

输入 测试实例 *Test*，为 *Dim* 维时间序列；决策树根节点 *Root*。

输出 分类标号。

1） if *Root* 为叶子节点
返回当前类标号

2） else A 为当前节点用于划分的属性，K 为属性 A 的簇数

3） for $i = 1:K$,

4） 计算实例 A 属性序列到每一簇中心序列的距 $dtw(test(A).ts, centers(i,:))$；

5） 选择距离最小的簇，记录标号 j；

6） 从第 j 个分支进入，类标号 = $Use\_PDT(test(\sim A), Root.child(j))$；

7） 返回类标号

3 实验与结果分析

下面利用移动通信的真实数据来进行移动客户流失预测，建造决策树，并基于测试集进行效果测试。另外，还采用提取特征指标的方法，对历史数据求平均值，然后对每种属性的历史平均值进行聚类划分，再采用传统的 ID3 方法建树，并在测试集中测试。将 ID3 决策树，与过程决策树进行比较。

3.1 实验数据

采用的数据集如下。某市 2004 年 1 月到 4 月期间的移动客户通信数据。数据总量约为 22 万条，我们随机抽取2 000 条数据作为训练数据集，另外 10 000 条数据作为测试数据集。每条数据包含如下内容：1）该用户 1 到 3 月每天的通话时间，共 91 d，为一个时间序列属性；2）该用户 1 到 3 月每天的短信数量，共 91 d，为一个时间序列属性；3）该用户 1 到 3 月每天的交往圈大小，即和该用户通话的不同电话号码的数量，共 91 d，为一个时间序列属性；4）该用户 4 月份是否离网，是布尔值，0 表示没有离网，1 表示离网，为预测结果的对比观察值。数据的前 3 项看做数据的 3 个属性，每一种属性的取值为一条长度为 91 的序列。数据的最后一项是数据的期望输出，即分类标号。

3.2 预测准确率比较

实验中分别利用方法 1 和方法 2 对训练数据进行学习，建造决策树。另外，我们对每一个维长度为 91 的历史数据求平均值，然后对每种属性的历史平均值进行聚类划分，再采用传统的 ID3 方法建树。用学习好的三棵决策树，对 10 000 条测试数据进行分类。比较 3 种不同途径得到的决策树的分类准确率。

经过多次实验，给出 3 种方法在不同聚类划分簇数 K 下对测试数据分类的平均准确率，结果见图 1 ~2。

由图 1 可以看出，简单地用历史平均值作为特征，去掉时间维度，并通过 ID3 算法构造的决策树分类准确率较低，基本上在 60% 以下。原因是简单地取平均值会忽略时间序列的过程特性，如增长或减小的趋势等。而过程决策树的分类准确率更高，原因是其划分和属性选择方法保留了一定的过程信息。因此，在使用决策树对实例进行分类时，实例在决策树中的分类路径选择每一个分支的判断标准并非序列的某个特征值，而是整个序列。

图 2 中 3 种方法构造的决策树的最高准确率仍保持图 1 中平均准确率之间的相对关系。另外,比较图 1 和图 2 中发现于某些相同 K 值的最高准确率和平均准确率的在两图中的值差距较大。导致这一现象的原因是决策树结构分支划分都是严格切割。因此当训练数据发生微小变化,或者聚类划分的结果稍有变动就会导致决策树结构的变化。因此,采取较为稳定的聚类方法,或者使用较为模糊的划分边界,是在这方面提高效果的重要措施。

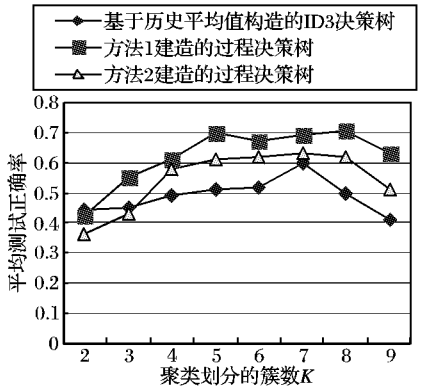

图 1　数据的平均分类准确率

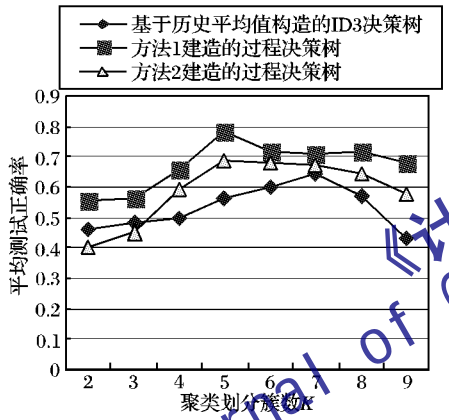

图 2　数据的最高分类准确率

3.3　建树效率比较

下面给出两种过程决策树构造方法,对于不同的 K 取值的建树时间。注意,这里方法 1 的建树时间包括对数据的聚类划分时间。实验结果如图 3。

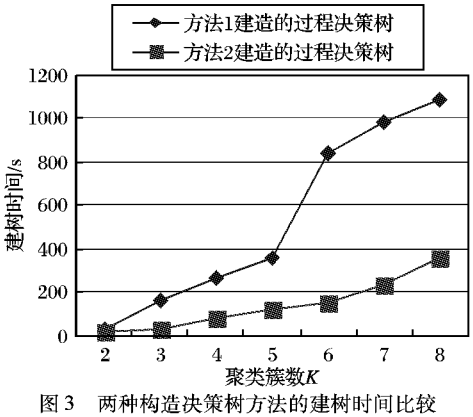

图 3　两种构造决策树方法的建树时间比较

由图 3 可以看出,方法 2 在时间效率上要明显优于方法 1。原因在于,过程决策树的构造时间大部分用于时间序列的聚类划分。而方法 1 在建树之前对数据进行聚类。这时聚类算法的数据包括所有输入的时间序列数据。相比之下,方法 2先对属性进行选择,然后进行聚类划分。这样除根节点外,其他的分支节点处的聚类操作针对的数据只是原始输入数据的一个子集。因此在聚类操作上,方法 2 比方法 1 节省了许多时间。

因此,虽然方法 2 的准确率低于方法 1,但是其效率却明显高于方法 1。这一特点在数据规模庞大的情况下显得尤为重要。因此,通过何种方法更好的计算时序熵,从而提高方法 2 的准确率,是一个重要问题。

4　结语

本文尝试对传统决策树算法进行扩展,构造一种适用于时间序列属性值的决策树模型,并将其应用到移动客户流失预测问题,取得了显著的效果。由于时序熵近似计算方法的局限性,方法 2 构造的过程决策树的分类准确率还不是十分理想。进一步的工作将集中寻找一种更为精确的时间熵运算方法,从而提高基于多维时间序列数据的分类效果。

参考文献:

[1]　金涛, 胡志改. 移动通信客户流失分析[J]. 移动通信, 2005, 29(2): 114 – 117.

[2]　杨一鸣, 潘嵘, 潘嘉林, 等. 时间序列分类问题的算法比较[J]. 计算机学报, 2007, 30(8): 1259 – 1266.

[3]　VAHDATPOUR A, SARAFZADEH M. Unsupervised discovery of abnormal activity occurrences in multi-dimensional time series, with applications in wearable systems [C]// Proceedings of SIAM International Conference on Data Mining. Columbus, Ohio, USA: SIAM, 2010: 641 – 652.

[4]　DING C, HE XIAOFENG. K-means clustering via principal component analysis [C]// Proceedings of the Twenty-first International Conference on Machine Learning. New York: ACM Press, 2004: 225 – 232.

[5]　DING RONGTAO, JI XINHAO, ZHU LINTING, *et al*. Study of the learning model based on improved ID3 algorithm [C]// First International Workshop on Knowledge Discovery and Data Mining. Washington, DC: IEEE, 2008: 391 – 395.

[6]　DING HUI, TRAJCEVSKI G, SCHEUERMANN P, *et al*. Querying and mining of time series data: Experimental comparison of representations and distance measures [J]. Proceedings of the VLDB Endowment, 2008, 1(2): 1542 – 1552.

[7]　POVINELLI J R, JOHNSON M T, LINDGREN A C, *et al*. Time series classification using Gaussian mixture models of reconstructed phase spaces [J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(6): 779 – 783.

[8]　LEE J A, VERLEYSEN M. Generalization of the Lp norm for time series and its application to self-organizing maps [C]// Proceedings of Workshop on Self-Organizing Maps. Paris: [s. n.], 2005: 733 – 740.

[9]　何新贵, 许少华. 过程神经元网络[M]. 科学出版社, 2007.

[10]　RUGGIERI. Efficient C4. 5[J]. IEEE Transactions on Knowledge and Data Engineering, 2002, 14(2), 438 – 444.

[11]　TSANG S, KAO B, YIP K Y, *et al*. Decision trees for uncertain data [C]// Proceedings of the 2009 IEEE International Conference on Data Engineering. Washington, DC: IEEE Computer Society, 2009: 441 – 444.

[12]　COVER T M, THOMAS J A. Elements of information theory[M]. Hoboken, NJ: John Wiley & Sons, 2006.