ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.2, February 2008, pp.351−357
DOI: 10.3724/SP.J.1001.2008.00351

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

属性构造原则与时序计数算子的研究*

邵 华+, 赵 宏

(东北大学 软件中心,辽宁 沈阳 110004)

Study on the Attribute Construction Rules and Time-Serial Count Operators

SHAO Hua+, ZHAO Hong

(Software Center, Northeastern University, Shenyang 110004, China)

+ Corresponding author: Phn: +86-24-22945273, Fax: +86-24-22955451, E-mail: shaohcn@hotmail.com, http://www.neu.edu.cn

Shao H, Zhao H. Study on the attribute construction rules and time-serial count operators. *Journal of Software*, 2008,19(2):351−357. http://www.jos.org.cn/1000-9825/19/351.htm

Abstract: Although count operator was used effectively in the process of data preprocessing, abusive use would cause the inconsistent problem of attribute relationship. To solve that problem, after proposing three attribute construction rules, time-serial count operator, a new algorithm for time-serial correlative model without inconsistent problem of attribute relationship is proposed. The time-serial increment count operator can remarkably reduce the high computing cost of time-serial count operator if the assumption is satisfied. The results of experiments prove the above conclusion.

Key words: attribute construction; count operator; attribute relationship consistency rule; preprocessing of time-serial data

摘 要: 包括计数算子在内的属性构造技术往往能够提高数据挖掘模型的预测精度,但不加条件地使用会导致属性关系不一致问题.为解决此问题,在提出了属性关系一致等 3 个属性构造原则后,给出了在时序相关模型下避免属性关系不一致问题的新算法——时序计数算子.时序增量计数算子在满足其假设条件下,可以较小的代价显著地降低时序计数算子的高计算成本.实验结果验证了上述结论.

关键词: 属性构造;计数算子;属性关系一致原则;时序数据预处理

中图法分类号: TP311 **文献标识码**: A

在许多数据挖掘应用中,实际数据往往由于数据量大、数据属性多、数据不均衡等问题而导致其比 UC Irvine 的实验数据复杂得多.尽管许多学习算法提供了选择和抽取属性或者构造属性的方法,但理论分析和实验表明,许多算法在分析具有无关或冗余的多属性数据时,算法的可扩展性很差.为了提高预测模型的精度,精心地进行数据预处理是一种很好的办法.[1]

属性构造可以通过变换产生新的属性,从而弥补原有数据表示空间的不足.它和属性选择都是通过数据预处理来提高模型精度的关键技术.作为一种属性构造技术,计数算子在提高模型精度的过程中作用显著.

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2002AA113020 (国家高技术研究发展计划(863))

Received 2004-04-09; Accepted 2006-11-03

Bloedorn 和 Michalski 在文献[2]中介绍的数据驱动的构造归纳系统 AQ17-DCI,对复杂的 Monk 数据,通过计数算子等属性构造方法,可以让人惊讶地产生预测精度为 100%的模型.而该文在引入计数算子后,使得系统对不均衡数据中的弱类别的预测精度从不到 1%提高到 6%~10%[3].另外,由于计数算子可以作为构造 profile[4]的技术手段,很方便在实际应用中使用,因此对计数算子的研究有着显著的实际应用价值.

一些相关文献[2,5]描述了计数算子等属性构造方法会取得很好的应用效果,所以,对属性构造的进一步研究是有意义的.本文针对属性构造提出了 3 条基本原则,它们描述了属性构造的约束条件.这对于研究或者使用类似 AQ17-DCI 的属性自动构造系统具有重要意义.此外,根据这些原则,本文提出了针对时序数据的计数算子和改进算法.通过实验结果分析不同算法对精度的影响,进而证实本文所提出的结论.

1 属性构造原则

在现有的属性自动构造系统中,不再由专家人为地设计新属性,而是系统自动产生.而无限制的属性构造会产生导致模型应用失效的欺骗性数据(参考第 2.3 节中的例 1),因此,对属性构造原则的研究就显得十分重要.这里提出的属性构造原则包括属性关系一致原则、目标排除原则和时序相关有序原则.

定义 1(属性关系一致原则). 在数据挖掘预测模型里,训练数据中各属性之间的逻辑关系与测试、验证数据中各属性之间的逻辑关系一致.

定义 2(属性关系不一致问题). 违反属性关系一致原则的问题为属性关系不一致问题.

定义 3(目标排除原则). 数据挖掘预测模型里的新属性不能利用当前记录的目标属性的结果来构造.

定义 4(时序相关模型). 数据挖掘时序相关模型是指该模型的数据存在非空的描述交易时间或者次序的属性 *Time*,并且该数据中属性 *A* 的取值与属性 *B* 的历史值存在依赖关系,这种模型称为时序相关模型.这里表示时间或者次序的属性 *Time* 称为时序属性.时序相关模型数据可以根据相应记录的时序属性值进行时序的比较,时序值小的记录值会影响时序值大的记录值.

定义 5(时序相关有序原则). 在数据挖掘时序相关模型中进行属性构造时,当前记录的值不能利用比当前时序值更大的记录值来构造.

目标属性排除原则和时序相关有序原则是属性关系一致原则的两个特例,因为在真实数据里目标属性值和未来数据都是未知的,利用未知数据来构造属性值是没有任何实际应用意义的.遵守属性关系一致原则的目的有二:一是保证属性构造不会破坏现有数据模型中各属性的内在逻辑约束,这样才能保证包括新属性产生模型的有效性;二是保证数据模型在将来实际预测时使用的数据能够根据历史数据正确构成新属性.遵守属性关系一致原则,避免属性关系不一致问题,这是数据预处理阶段的基本守则.

2 计数算子和时序计数算子

2.1 计数算子

本文令样本数据集由 $D=\{d^r|r=1,\ldots,N\}$表示,$d^r=(x_{1,r},\ldots,x_{i,r},\ldots,x_{t,r},x_{0,r})$代表数据集 D 的第 r 条记录,$x_{i,r}$ 表示第 r 个实例的解释属性 X_i 的值,$x_{0,r}$ 表示第 r 个实例的目标属性 X_0 的值,$i=1,\ldots,t,t$ 表示解释属性的数目,N 表示记录总数.

许多数据库管理系统都支持计数函数.计数函数用来统计满足条件 C 下样本数据集 $D=\{d^r|r=1,\ldots,N\}$中被关注的属性集合 $S=\{X_i|i=1,\ldots,t\}$中的属性出现的次数,表示为 $Count(D,S,C)$.这里,S 称为计数目标集合,C 称为计数条件,集合 S 包含的属性称为计数目标属性,简称计数属性.例如,$Count(D,\{x_1,x_2\},\text{“}x_1=0\wedge x_2=0\text{”})=2$ 表示样本集 D 中属性 x_1,x_2 中共有两次同时取零值.

定义 6(计数算子). 计数算子构造属性 *CT*,其值为样本数据 D 中每条记录计算计数属性 X_c 取当前值的数目,表示为 $AttrCount(D,X_c,CT)$.注意,计数算子返回的不是一个数据,而是样本数据 D 中的一列数据,并被作为新属性的属性值.计数算子可以记为

$$AttrCount(D,X_c,CT)=\{x_{CT,r}|x_{CT,r}=Count(D,\{X_c\},\text{“}X_c=x_{c,r}\text{”}),r=1,\ldots,N\}.$$

计数算子的算法如下:

输入:样本数据 *D*、属性集合 *S* 和新属性 *CT*.

输出:*D* 中一列属性名为 *CT* 的数值型数据.

1. create table *T*(*S*,*CT*,*flag*); //建立一个临时表 *T*,表中属性包含 *S*,数值型属性 *CT* 和作为标记使用的 *flag*.
2. insert into *T* select *S*, *count*(*S*) from *D* group by *S*;
3. alter table *T* add *CT* INT NULL;
4. insert into *CT* select *T.CT* from *T*, *D* where *T.S*=*D.S*; //连接 *T* 和 *D*
5. drop TABLE *T*;

上述算法是计数算子的一种形式化实现算法.注意,在特定的数据操作环境中有一些特殊的高效语句,例如,SAS 就提供 SUMMARY 过程,可以直接实现整个过程.

2.2 时序计数算子

对于时序数据,计数算子是一种很好的属性构造方法.这里有一个十分重要的细节问题,就是在构造衍生属性的值时,是否可以使用原来的样本数据 *D*.如果采用同一样本数据(也是通常的做法),一方面可以批量、快速地产生衍生属性值;另一方面,却有可能违背属性关系一致原则,产生属性关系不一致问题,导致预测模型过度拟合问题.如果训练集的数据交易时间跨度相对较长,这个问题就应该考虑.下面我们通过计数算子的时序算法和例 1 来进一步说明这种属性关系不一致问题.

定义 7(时序计数算子). 时序计数算子构造属性 *CT*,其值为样本数据 *D* 中每条记录计算属性集合 *S* 中的属性在历史数据中取当前值的数目,表示为 *AttrTimeCount*(*D*,*S*,*Time*,*Key*,*CT*),这里,参数 *Time* 为表示样本时序的属性,称为时序属性.属性 *Key* 为数据 *D* 的关键字属性.

设计时序计数算子的目的是遵守时序相关有序原则,我们定义算法如下:

输入:样本数据 *D*、属性集合 *S*、关键字 *Key* 和新属性名 *CT*.

输出:*D* 中一列属性名为 *CT* 的数值型数据.

1. create table *History*(*S*,*Time*,*Key*,*CT*); //建立历史数据表.
2. *CountTime*=Min(*Time*) from *D*; //*CountTime* 为 *D* 中 *Time* 属性的最小值
3. insert into *History* select *S*, *Key* from *D* where *Time*=*CountTime*; //初始化历史数据
4. AttrCount(*History*,*S*,*CT*); //调用计数算子
5. Updata *D* set *CT*=*History*.*CT* where *History*.*Time*=*CountTime* AND *D*.*Time*=*CountTime* AND *D*.*Key*=*History*.*Key*;
6. *CountTime*1=Min(*Time*) from *D* where *Time*>*CountTime*; //*CountTime*1 为 *Time* 属性的下一个最小值
7. if *CountTime*1>*CountTime* then {*CountTime*=*CountTime*1; goto Step 3};

2.3 解释计数算子的问题

例 1(GB 问题):表 1 的前 3 列为原始交易数据,其中 *Result* 属性值只有 *G* 和 *B* 两种.最后两列是分别通过计数算子和时序计数算子构造的衍生变量:

Ratio:=*AttrCount*(*D*,{#*Cunstomer*,*Result*})/*AttrCount*(*D*,{#*Cunstomer*}),

Time-serial Ratio:=*AttrTimeCount*(*D*,{#*Cunstomer*,*Result*},#*Transaction*,#*Transaction*)/

AttrTimeCount(*D*,{#*Cunstomer*},#*Transaction*,#*Transaction*).

从 *Ratio*,*Time-serial Ratio* 和 *Result* 在表 1 中的数据可以看出,*Time-serial Ratio* 存在“.”值,表示无法被计算出来.此外,由计数算子构造的 *Ratio* 与 *Result* 为 *G* 的可能性完全一致,即 *Ratio* 为 100,*Result* 为 *G* 的可能性为 100%;*Ratio* 为 50,*Result* 为 *G* 的可能性为 50%;*Ratio* 为 0,*Result* 为 *G* 的可能性为 0%等.而时序计数算子的预测结果(见表 2)不仅没有出现类似的情况,而且从中可以发现一个特殊的规则:*Time-serial Ratio* 为 50,*Result* 为 *G* 的可能性为 100%.

Table 1 Training data of GB problem

表 1 GB 问题的训练数据

#Transaction	*#Cunstomer*	*Result*	*Ratio*	*Time-Serial ratio*
1	01	*G*	50	.
2	02	*B*	80	.
3	03	*G*	66.7	.
4	01	*G*	50	100
5	04	*B*	66.7	.
6	02	*G*	80	0
7	05	*B*	0	.
8	03	*B*	66.7	100
9	06	*G*	100	.
10	02	*G*	80	50
11	04	*G*	66.7	0
12	02	*G*	80	66.7
13	01	*B*	50	100
14	07	*G*	100	.
15	01	*B*	50	66.7
16	03	*G*	66.7	50
17	06	*G*	100	100
18	05	*B*	0	0
19	02	*G*	80	75
20	04	*G*	66.7	50

Table 2 Result of GB problem computed by time-serial count operator

表 2 时序计数算子计算 GB 问题的结果

Time-Serial ratio	*Transactions*	*G-Results*	*Ratio of G-result* (%)
.	7	4	57
0	3	2	67
50	3	3	100
66.7	2	1	50
75	1	1	100
100	4	2	50

GB 问题的测试数据为表 3 的前 3 列,第 4 列 G-Ratio 表示 *Result* 为 *G* 概率的百分数,只有 100 和 0 两种值.表 3 的后两列分布为计数算子和时序计数算子预测结果为 *G* 的精度(百分数).可以发现,计数算子居然会出现无法预测的值".",而且其预测误差要高于时序计数算子的误差.

Table 3 Validating data of GB problem

表 3 GB 问题的验证数据

#Transaction	*#Cunstomer*	*Result*	*G-Ratio*	*Count accuracy*	*TimeCount accuracy*
21	07	*B*	0	100	50
22	08	*B*	0	.	57
23	06	*G*	100	100	50

3 增量时序计数算子

由于时序计数算子对每个交易时间都要进行连接,其计算代价很高,时间复杂度为 *O*(交易时间次数).对海量数据而言,计算时间尤其是无法忍受的.

定义 8(交易周期). 对于给定的概率 *p*,定义交易周期 *K*,满足 $P\{Count(D,\{T,I\},"T>t\wedge T<t+K\wedge I=i")\}<1-p$.这里,$Count(D,\{T,I\},"T>t\wedge T<t+K\wedge I=i")$表示数据集 *D* 中计数目标 *i* 在 *t* 到 *t*+*K* 时间内的交易次数.

定义 9(增量时序计数算子). 增量时序计数算子构造属性 *CT*,其值为样本数据 *D* 中每条记录增量计算属性集合 *S* 中的属性在历史数据中取当前值的数目,表示为 *AttrIncTimeCount*(*D*,*S*,*Time*,*K*,*Key*,*CT*),这里,参数 *Time* 为表示样本时序的属性,*K* 为交易周期.

我们定义的增量时序计数算子算法如下:

输入:样本数据 *D*、属性集合 *S*、时序属性 *Time*、交易周期 *K* 和新属性名 *CT*.

输出:*D* 中一列属性名为 *CT* 的数值型数据.

1. create table *History*(*S*,*Time*,*Key*); //建立历史数据表
2. *BeginTime*=Min(*Time*) from *D*; //*BeginTime* 为 *D* 中 *Time* 属性的最小值
3. *EndTime*=Max(*Time*) from *D*; //*EndTime* 为 *D* 中 *Time* 属性的最大值
4. insert into *History* select *S*, *Key* from *D* where *Time*<*BeginTime*+*K* AND *Time*>=*BeginTime*; //初始化历史数据
5. Updata *D* set *CT*=*History*.*CT* where *History*.*Time*>=*BeginTime* AND *D*.*Time*<*BeginTime*+*K* AND *D*.*Time*>=*BeginTime* AND *D*.*Key*=*History*.*Key*;
6. *AttrCount*(*History*,*S*,*CT*); //调用计数算子
7. if *BeginTime*+*K*>*EndTime* then {*BeginTime*=*BeginTime*+*K*; goto Step 4} else delete *T*;

考虑到在许多实际问题中,交易主体的交易时间有间隔,即具有一个交易间隔时间 *T*.如果在交易间隔时间里只进行一次计算,那么即使 *K*<*T*,增量算法也应该与原算法的结果差别不大.此外,如果交易系统本身可以提供增量交易数据,并且增量交易数据的周期本身就符合算法的要求,那么增量时序算子就会完全利用增量交易数据作为增量的训练数据,这时,单次的增量算法有可能低于计数算子的计算量.有些交易系统不能提供增量交易数据,甚至交易数据的内容在半年后还会有改动,这时需要利用增量算法生成训练数据.

4 实验及结果

时序计数算子在解决海关报关欺诈问题[4]时已体现出其应用价值.为了进一步准确描述其理论价值,需要用公共数据集的实验来证明.下面的实验数据来自 UCI 公布的电影文件(Gio's movie files)中 MAIN 数据[6]里 Hitchcock,E.S.Porter,Griffith,DeMille,Seitz,Lubitsch 和 Walsh 这 7 位作者的信息(共计 286 条记录),并修正了其中几处明显的拼写错误.我们把实验数据中的第 1 列重命名为 *ID*,第 3 列重命名为 *Directors*,第 4 列重命名为 *Year* 且作为时序属性.实验环境为 SAS,并利用 SAS 提供 SUMMARY 过程实现计数算子.

为了深入分析数据集自身特性对实验结果的影响,通过定义不同的计数目标和时间段在原始实验数据的基础上又构造了 6 个实验.表 4 为每个实验中的数据特点,*Records* 表示样本数据集 *D* 中包含记录的数目,*#Combination* 表示计数目标属性集 *S* 在 *D* 中出现的独特组合的总数,*#Time-serial* 表示时序属性在 *D* 中具有独特值的数目.*AttrCount* 表示计数算子算法,*AtimeC* 表示时序计数算法,*AincTC*3,*AincTC*5 和 *AincTC*10 表示 *K* 分别取 3,5 和 10 的增量时序计数算法.图 1 为各实验中不同算法所使用的时间比较,时间单位为毫秒.表 5 为各实验中不同算法导致的错误率的比较.这里,错误率定义为:与 *AttrCount* 算法计算结果相比有差异的 *Records* 除以该实验所使用数据集的记录总数(百分数).表 6 为各实验中不同算法导致的错误度的比较.这里,错误度表示该算法 *Q* 相对于时序计数算法 *AtimeC* 结果的一种度量,定义为 *sum*(*abs*(*DQ*.*ct*−*DAtimeC*.*ct*)),*DQ* 表示算法 *Q* 的结果数据集,*DAtimeC* 表示时序计数算法的结果数据集.

Table 4 Features of experiments

表 4 实验特性

#Experiment	*Records*	*Count attribute*	*Years*	*#Combination*	*#Time-serial*
1	62	Directors	1911–1920	5	10
2	171	Prc	1911–1941	7	31
3	286	Directors	all	7	64
4	62	Prc	1911–1920	6	10
5	171	Directors	1911–1941	9	31
6	286	Prc	all	18	64

从表 4 可以看出,实验 1 和实验 4 中的"*#Time-serial*"都为 10,这使得表 5 和表 6 里相应的实验中 *AincTC*10 和 *AttrCount* 的结果相同.这进一步说明,如果"*#Time-serial*"不大于交易周期值,就会使时序增量计数算子失效.图 1 说明了实验中时序计数算法计算的时间明显要比其他算法高出数倍.而表 5 和表 6 说明,增量计数算子的错误率和错误度要比计数算子低很多,可以不同程度地避免时序不一致问题.对于增量计数算子的不同交易周期,可以看出,只要"*#Time-serial*"充分大,错误度之间的比例就接近交易周期的比例.对于具体应用问题,可以根

据部分数据的实验结果,选择适当交易周期的增量计数算子进行属性构造.

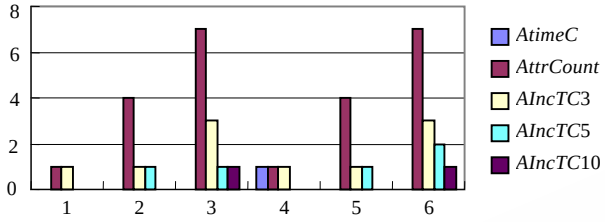

Fig.1 Time used in experiment (ms)

图 1 实验用时的比较(ms)

Table 5 Error ratio comparison in each experiment

表 5 实验结果错误率比较

Error ratio (%) / Method	Experiment					
	1	2	3	4	5	6
AttrCount	100.00	93.57	100.00	93.55	95.32	96.15
*AIncTC*3	80.65	81.29	86.71	80.65	82.46	77.62
*AIncTC*5	95.16	85.96	95.80	90.32	92.40	84.62
*AIncTC*10	100.00	90.06	98.95	93.55	95.32	91.26

Table 6 Error degree comparison in each experiment

表 6 实验结果错误度比较

Error degree / Method	Experiment					
	1	2	3	4	5	6
AttrCount	682	5 946	15 656	1 620	4 028	13 408
*AIncTC*3	260	862	1 250	270	524	1 472
*AIncTC*5	426	1 662	2 222	730	902	2 908
*AIncTC*10	682	3 670	4 208	1 620	1 616	6 048

5 相关工作比较

研究时序数据预处理方法的相关工作有很多,基本上可以分为两类.一类是引用非时序的数据挖掘预测模型或分类模型的相关方案,对原始数据进行一定的加工处理,如去除噪音、填补缺失数值或重新采样,使之能够为后继数据挖掘方法所用的过程.此外,属性构造[1,2,5]或属性选择[7]等方法也取得了不错的效果,另一类是以离群指数[8]、傅里叶变换[9]和小波变换[10]等为代表的数据预处理研究工作,这些工作是以研究离群数据挖掘等非分类模型为目的的.

尽管都是研究时序数据的数据预处理方法,但本文是以解决时序数据中属性关系不一致的问题为目的来研究属性构造方法,提出了 3 条属性构造原则和时序计数算子.与以上介绍的研究内容相比,本文的研究成果具有一定的独特性.

6 结束语

计数算子在数据挖掘预处理阶段十分重要,可以构造质量更高的属性.但是,如果在使用时存在属性关系不一致问题,就会导致计数算子产生的模型在实际应用中失效.本文提出的属性关系一致等原则规范了数据挖掘应用中属性构造的使用条件,并且据此修改的时序计数算子可以避免属性关系不一致问题,但依然存在算法计算代价过大的问题.增量时序计数算子不仅解决了时序计数算子时间复杂度高的问题,而且具有很好的应用前景.

References:

[1] Liu H, Motoda H. Feature transformation and subset selection. IEEE Intelligent Systems, 1998,13(2):26−28.

[2] Bloedorn E, Michalski RS. Data-Driven constructive induction. IEEE Intelligent Systems, 1998,13(2):30−37.

[3] Shao H, Zhao H, Chang GR. Applying data mining to detect fraud behavior in customs declaration. In: Proc. of the 2002 Int'l Conf. on Machine Learning and Cybernetics. Beijing: IEEE Computer Society, 2002. 1241−1247.

[4] Adomavicius G, Tuzhilin A. Using data mining methods to build customer profiles. IEEE Computer, 2001,34(2):74−82.

[5] Lavrac N, Gamberger D, Turney P. A relevancy filter for constructive induction. IEEE Intelligent Systems, 1998,13(2):50−56.

[6] Movies main file. The UCI KDD archive. http://kdd.ics.uci.edu/databases/movies/data/main.html

[7] Yan XB, Lu T, Li YJ, Cui GB. Research on event prediction in time-series data. In: Proc. of the 2004 Int'l Conf. on Machine Learning and Cybernetics. Shanghai: IEEE Computer Society, 2004. 2874−2878.

[8] Zheng BX, Xi YG, Du XH. Outlier mining for time series data based on outlier index. Acta Automatica Sinica, 2004,30(1):70−77 (in Chinese with English abstract).

[9] Zheng BX, Du XH, Xi YG. A new algorithm of outlier mining in time series data. Control and Decision, 2002,17(3):34−37 (in Chinese with English abstract).

[10] Peng H. Time series similar mining based on wavelet transformation. Journal of Xihua University (Natural Science), 2005,24(1): 89−91 (in Chinese with English abstract).

附中文参考文献:

[8] 郑斌祥,席裕庚,杜秀华.基于离群指数的时序数据离群挖掘.自动化学报,2004,30(1):70−77.

[9] 郑斌祥,杜秀华,席裕庚.一种时序数据的离群数据挖掘新算法.控制与决策,2002,17(3):34−37.

[10] 彭宏.基于小波变换的时序数据相似性挖掘.西华大学学报(自然科学版),2005,24(1):89−91.

邵华(1973－),男,辽宁沈阳人,博士,讲师,CCF 会员,主要研究领域为数据挖掘.

赵宏(1954－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机应用,计算机网络,多媒体技术.