信息系统学报
第2卷第1辑(总第2辑)：1－11

China Journal of Information Systems
Vol.2 No.1,May 2008：1－11

Graph-NB：一种高效准确的多关系朴素贝叶斯分类算法*

刘红岩[1]，陈海亮[1]，Han Jiawei[2]，Yin Xiaoxin[2]
（1 清华大学经济管理学院，北京 100084
2 美国伊利诺大学香槟分校计算机科学系，厄巴纳 61801）

摘 要 多关系分类是数据挖掘领域中的研究和应用热点。已有多关系朴素贝叶斯分类算法将所有与目标表相连的表都考虑在内，包括语义关系很弱的表。为此，本文提出一种新的分类算法—Graph-NB。它通过对表进行剪裁，达到优化语义关系图，从而一定程度上消除无关表对分类影响的目的。该算法实现了深度优先与广度优先两种遍历策略。实验结果表明，语义关系图的优化可以提高分类准确度和运行效率，相比于其他算法，该算法运行时间短，分类准确度高。

关键词 多关系分类，朴素贝叶斯分类，深度优先，广度优先
中图分类号 TP311，TP391

目前，绝大多数的信息存储在多关系数据库中，这些数据库一般具有多个相互存在联系的数据表。多关系数据挖掘致力于从多个表中直接挖掘信息和发现知识，而不是将数据库中的多个表合并成一个表后再进行挖掘。分类是数据挖掘领域中的一种重要技术，它根据数据集的特点构造出一个分类器（或分类模型），利用分类器对未知类别的样本赋予类别。

多关系分类算法大致可以分为两类：第一类是使用 propositionalization[1] 的传统分类算法。由于传统分类算法都是在单个表的基础上实现的，要将这些方法应用到多关系的背景下，需要使用 propositionalization 方法对数据库进行相应的转换，这种方法容易造成丢失信息，严格地说这类算法不属于多关系分类算法；第二类算法是直接以多个表为挖掘对象而构造分类器的方法[3-9,11-16]。这类算法常见的可分为两种，第一种是基于 Inductive Logic Programming[2] 的方法，如 FOIL[3]，TILDE[4]，ILP-R[6]，1BC[7]，1BC2[8]等；第二种将传统分类方法与多关系的相结合产生的方法。其中包括将朴素贝叶斯分类方法[17-19]应用到多关系背景下的方法，如 CrossMine[5] 和 Mr-SBC[9] 等。这些方法的共同特点是没有对数据库中的表进行选择，只要是直接或间接地与目标表（包含分类属性的表）相连接的表都考虑在内，这不仅会使算法效率降低，有时也会降低分类准确率。

为此，本文提出一种基于语义关系图的多关系朴素贝叶斯分类算法——Graph-NB 算法。Graph-NB 首先通过深度优先或广度优先策略遍历语义关系图中的所有表并收集相应的概率信息，然后通过应用多关系朴素贝叶斯分类算法，比较使用语义关系图中不同表时得到的训练集分类准确度来选取最优点对语义关系图进行裁剪简化，最后使用裁剪优化的语义关系图对测试集进行分类。该算法经过大量实验证明在效率和准确度方面均有提高。

* 基金项目：国家自然科学基金(70471006，70621061)。
通信作者：刘红岩，清华大学经济管理学院，副教授，e-mail：liuhy@sem.fsing.hua.edu.cn。

1 多关系朴素贝叶斯分类算法

贝叶斯分类是基于贝叶斯定理的统计学分类方法。基本原理是预测一个未知类别的样本属于各个类别的可能性,选择其中可能性最大的一个类别作为该样本的最终类别。

设样本用 $X=\{x_1,x_2,\cdots,x_n\}$ 表示,它可能属于 m 个类别 $C_1,C_2,\cdots,C_m$ 中的一个。$P(C_i|X)$ 表示样本 X 属于类别 C_i 的概率,根据贝叶斯定理可以得到

$$P(C_i \mid X)=\frac{P(x_1,x_2,\cdots,x_n \mid C_i)P(C_i)}{P(X)}=\frac{P(C_i)\prod_{j=1}^{n}P(x_j \mid C_j)}{P(X)} \tag{1}$$

由于各个类别的计算中 $P(X)$ 是不变的,则样本的类别由下面这个公式确定

$$C_{MAP}=\arg\max_{C_i\in C}P(C_i)\prod_{j=1}^{n}P(x_j \mid C_j) \tag{2}$$

现实世界中考虑到数据存储的效率问题,信息往往存在于一个关系数据库中的多个表中,将朴素贝叶斯分类算法应用到多关系的情况下需要做一些相应的转化。在多关系朴素贝叶斯分类算法中,假定目标表 t 有 n 个属性,目标表 t 中的记录可以用 $X=\{x_1,x_2,\cdots,x_n\}$ 表示,如果数据库中一个表 s 中有 p 个记录与表 t 中的 $X=\{x_1,x_2,\cdots,x_n\}$ 相连接,这 p 个记录分别是 $(y_{k1},y_{k2},\cdots,y_{kp})$,表 s 含有 r 个属性,那么每个记录可以用 $y_{ki}=(y_{ki1},y_{ki2},\cdots,y_{kir})$ 来表示。在属性相互独立的假设前提下,多关系朴素贝叶斯分类算法的公式为

$$\begin{aligned}C_{MAP}&=\arg\max_{c_j\in C}P(c_j \mid X)\\&=\arg\max_{c_j\in C}P(c_j)P(X \mid c_j)\\&=\arg\max_{c_j\in C}P(c_j)P(x_1,\cdots,x_n,y_{k11},\cdots,y_{k1r},\cdots,y_{kp1},\cdots,y_{kpr} \mid c_j)\\&=\arg\max_{c_j\in C}P(c_j)\prod_{i=1}^{n}P(x_i \mid c_j)\prod_{q=k1}^{kp}\prod_{t=1}^{r}P(y_{qt} \mid c_j)\end{aligned} \tag{3}$$

2 语义关系图

多关系朴素贝叶斯分类算法的关键是得到各个属性的概率分布,通过统计各个属性值的出现次数来计算条件概率值。要实现这个目的,必须明确数据库中各个表之间的连接关系。为此我们提出语义关系图的概念,用它来表达和概括数据库中各表与目标表之间的关系。

2.1 定义

语义关系图(Semantic Relationship Graph,SRG)是一种有向无环图。它包含三种元素:表结点、连接路径和连接属性。语义关系图是根据相应的连接属性通过连接路径将数据库中的所有表结点连接起来。目标表是语义关系图的起点。它与其他表结点通过直接或间接的连接路径相连。连接路径有两种:一种是主键到外键的连接,即连接路径末端的表中含有外键指向连接路径始端的表中的主键;另一种是外键到主键的连接,即连接路径始端的表中含有外键指向连接路径末端的表中的主键。可以说,语义关系图描述了一个数据库中数据表的所有相互关系,它的形状与 ER 图类似。

语义关系图之所以是有向的,一是因为有向图可以指定数据表之间的连接次序,二是因为数据库中两个表之间的关系往往不止一个。如果语义关系图是无向的,那么这种两个表之间的多关系可能

会造成表连接操作的循环进行。将其定义为有向图，就可以有效地避免这种循环情况的出现。

2.2 举例

图 1 给出了 PKDD CUP99 的一个金融数据库的语义关系图。图中每个结点代表数据库中的一个表，Loan 表是目标表，其他的表通过直接或间接的方式与 Loan 表相连。为了简便起见，图中省略了连接属性。从连接路径来看，图中 Disposition 表和 Car 表都被遍历两次，从分支的角度看，这个语义关系图总共有 4 个分支，其中有两个分支上都存在 Disposition 表和 Car 表。正因为是存在于不同的分支上，在实际遍历过程中，两者的概率分布是不一样的，需要区别对待。

图 1 中语义关系图的各表之间关系比较简单，各个表在语义关系图中都只出现一次。现实中有些数据库中的表之间可能存在多种关系，因而需要处理潜在的循环问题。

图 2 给出了 Inductive Logic Programming[2] 领域常用的测试数据集 Mutagenesis 数据库的关系图。由于 Bond 表通过外键与 Atom 表可以进行多次连接，所以两表之间形成一个环。为了在语义关系图中表达这种关系，需要将环拆成一条链，同时复制相应的 Atom 表和 Bond 表。可以根据实际情况决定复制次数，图 3 是复制一次的语义关系图。

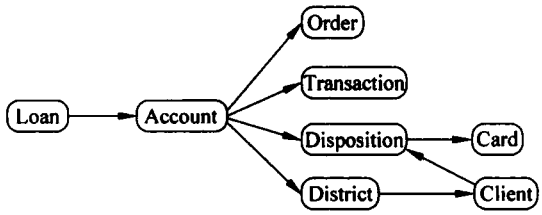

图 1 金融数据库的语义关系图

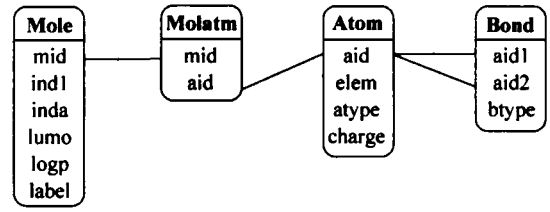

图 2 Mutagenesis 数据库的关系图

图 3 Mutagenesis 数据库的语义关系图

2.3 语义关系图的优化

语义关系图中并不是所有的表都与目标表密切相关，实际上很多表有可能对提高分类准确度是没有帮助的。在多关系分类中，过去很多的算法是将数据库中所有的表一起考虑来进行分类，这样有可能造成冗余，引入一些不相关的数据表，不但降低算法的运行效率，而且有可能会减少分类准确度。Graph-NB 算法通过裁剪优化语义关系图，去除与目标表不是很相关的表来提高分类准确度。具体做法是沿着连接路径，连接一个表后对训练集进行一次分类，在遍历完整个语义关系图后，再使训练集分类准确度最高的结点对语义关系图进行裁剪，得到优化的语义关系图。图 4 给出了利用广度优先策略对 PKDD CUP99 的金融数据库进行裁剪优化后的语义关系图。

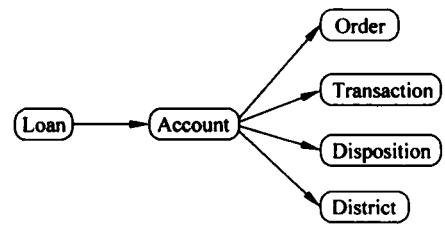

图 4 优化后的语义关系图示例

3 Graph-NB 算法

基于语义关系图的多关系朴素贝叶斯分类算法(Graph-NB 算法)与其他常见的分类算法一样，分

为训练和测试两个阶段。对于语义关系图中的表可以采用不同的遍历策略,如深度优先策略或广度优先策略。不同的遍历策略可能会产生不同的结果。因为 Graph-NB 算法计算的条件概率是累积的,也就是说历次遍历的表的概率信息在计算贝叶斯条件概率时是累积在一起的。那么不同的遍历顺序所使用的表信息也是不同的,因此可能会产生不同的分类准确度,选择裁剪优化的结点也会不一样。下面我们详细讨论深度优先和广度优先两种不同策略的实现方法。

3.1 深度优先策略

大部分的语义关系图含有多个分支,深度优先策略就是按照一定的顺序逐个遍历每个分支,而且是遍历完一个分支后才遍历下一个分支。在对训练集进行分类时,以前曾经遍历过的表,不管是当前分支的,还是其他分支的,都被用来进行贝叶斯分类。具体实现是采用递归的方法来遍历语义关系图中的表,训练阶段深度优先策略的 Graph-NB 算法如下。

算法 1 Graph-NB 深度优先——训练阶段

输入:目标表 tt,其他表集合 T 和语义关系图 G

输出:优化的语义关系图 OG,属性概率统计值集合 P

方法:

1. $maxAccuracy$= $CollectAndClassify$(tt)
2. Initialize $edgeNo$=0,$maxEdgeNo$=0;
3. **for each** out edge of target table tt **do**
   $edgeNo$++ ;
   NextEdge(tt)
   end
4. Cut off all of edges with number≥$maxEdgeNo$,and output remaining part of G as OG.

子过程 ***NextEdge*(*parentTable pt*)**

方法:

1. let t=table pointed to by edge with number $edgeNo$;
2. $propagate(pt,t)$;
3. $accuracy=CollectAndClassify(t)$;
4. **if** ($accuracy>maxAccuracy$) **then**
   $maxAccuracy=accuracy$;
   $maxEdgeNo=edgeNo$;
5. **for each** out edge of current table t **do**
   $edgeNo$++ ;
   NextEdge(t);
   end

算法首先调用 CollectAndClassify 函数收集目标表的统计信息并对其进行分类,得到的分类准确度返回给 maxAccuracy。第二步中初始化全局变量 edgeNo 和 maxEdgeNo,其中 edgeNo 代表语义关系图中各表的遍历顺序,同一个表可能有多个 edgeNo,因为该表可能存在于多个分支中。maxEdgeNo 记录分类准确度最高的结点。然后对于目标表的每个分支,算法都要调用 nextEdge 函数,最后,算法对连接路径上标号大于 maxEdgeNo 的表都进行裁剪,只剩下之前的表作为优化的语义关系图。

在子过程 nextEdge 中,算法调用 propagate 函数对连接路径末端的表进行 tuple ID propagation[5] 操

作，然后使用 CollectAndClassify 函数对其进行统计信息并分类，如果当前得到分类准确度比当前的 maxAccuracy 要大，则更新参数 maxAccuracy 和 maxEdgeNo 的值。之后，nextEdge 函数再次被调用来处理被连接的函数的所有分支。

训练阶段结束后，算法根据优化的语义关系图和各表属性的概率分布信息对未知类别的测试数据集进行分类测试。下面是深度优先策略下测试阶段的 Graph-NB 算法。

算法 2 Graph-NB 深度优先——测试阶段

输入：测试表 *st*，其他表集合 *T* 和优化的语义关系图 *OG*

输出：测试表 *st* 所有记录的类别

方法：

1. *computeProb*(*st*)；
2. **for each** table *t* pointed to by each out edge of target table *st* do
   Classify(*st*,*t*)
3. output class label of each tuple in table *st*.

子过程 *Classify*(*parentTab pt*,*currentTab t*)

方法：

1. *propagate*(*pt*,*t*)；
2. *computeProb*(*t*)；
3. **for each** table *ct* pointed to by each out edge of table *t* **do**
   Classify(*t*,*ct*)

可以看出，算法过程与训练阶段基本相似，遍历顺序也一样，只是在 maxEdgeNo 的地方停止继续遍历。

3.2 广度优先策略

基于广度优先策略的 Graph-NB 算法首先遍历与目标表相近的所有表，然后遍历这些表下一层的表，以此类推。下面是训练阶段基于广度优先策略的 Graph-NB 算法。由于测试阶段比较相似，就不再详细列出。

算法 3 Graph-NB 广度优先——训练阶段

输入：目标表 *tt*，其他表集合 *T* 和语义关系图 *G*

输出：优化的语义关系图 *OG*，属性概率统计值集合 *P*

声明：

<*left*,*right*>-one edge of SRG (<target relation,target relation> is the first edge of SRG)

Q-propagating routes stored in the form of queue

addQ(<*left*,*right*>)-add edge to the rear of Q

delQ()-return the front edge of Q and remove it

propagate (<*left*,*right*>)-join relation *right* and load its data using tuple ID propagation method

方法：

addQ(<*target relation*,*target relation*>)

maxAccuracy=0

while Q is not empty **do**

a. *propagate*(*delQ*( )),*edgeNo*++ ;

```
    b. add propagating routes (or join edges)
       for each edge out of current relation do
         addQ( <current relation, right r> )
       end
  c. accuracy= CollectAndClassify(current relation)
  d. if(accuray> maxAccuracy)then
         maxAccuracy=accuracy;
         maxEdgeNo=edgeNo;
     end
  end
```

该算法借用队列数据结构来实现。在初始化一个空的队列后,先将目标表的路径(即路径始末端都是目标表的虚拟路径)添加到队列中,然后从队列中取出一个连接路径,统计连接路径末端的表属性的概率信息,并将此表所包含的所有分支(连接路径)添加到队列的最后,在使用当前所获得到信息对训练集进行分类之后,再从队列的开头取出一个连接路径进行下一步的连接。除路径连接顺序不同外,其他操作与深度优先策略的算法是一致的。

4 实验结果与分析

我们通过在多组数据集上测试来比较 Graph-NB 和 CrossMine,FOIL,TILDE,1BC1,BC2,Mr-SBC 等算法的准确率与算法效率。另外,我们还设计实验比较深度优先与广度优先两种策略的优劣性。实验的运行环境是 IBM 笔记本 R40,Intel Pentium 4 2.2GHz CPU,512MB 内存,操作系统是 Windows 2000 Professional。Yin 的 CrossMine 算法[20]采用了抽样方法,其参数设置与文献[5]中设置相同,即 $MIN\ FOIL\ GAIN = 2.5$. $MAX\ RULE\ LENGTH = 6$. $NEG\ POS\ RATIO = 1$. $MAX\ NUM\ NEGATIVE = 600$。

4.1 Mutagenesis 数据库

第一组实验的实验数据是 ILP 领域广泛使用的 Mutagenesis 数据库。这个数据库包含 4 个表和 15218 条记录。目标表包含 188 条记录,目标属性有两种类别,分别对应 124 条和 64 条记录。表 1 列出了这个数据库的三种背景知识。

表 1 Mutagenesis 数据库背景知识

背 景	描 述
BK_0	每个分子包含原子,原子键,成键类型,原子类型以及原子上的部分电荷
BK_1	BK_0 中的定义以及 *mole* 表中的 *indl* 和 *inda* 属性
BK_2	BK_1 中的定义以及 *mole* 表中的 *logp* 和 *lumo* 属性

我们采用十倍交互验证方法(即 ten-fold cross-validation 方法)来进行测试。表 2、3 和 4 分别列出了各种算法在 Mutagenesis 数据库三种背景知识下的运行结果。其中,FOIL 和 TILDE 的结果参照文献[4],1BC 的实验结果从文献[10]获得,1BC2 的结果参照文献[9]。表 5 列出了这三种背景知识下的平均结果。

表 2 Mutagenesis 数据库 BK_0 运行结果

算 法	准确度/%	运行时间/秒
Graph-NB(without pruning)	77.5	1.1
Graph-NB(深度)	77	**1.2**
CrossMine	68.8	2.5
FOIL	61	4950
TILDE	75	93
1BC	**80.3**	—
1BC2	72.9	—
Mr-SBC	76.5	36

表 3 Mutagenesis 数据库 BK_1 运行结果

算 法	准确度/%	运行时间/秒
Graph-NB(without pruning)	77	1.1
Graph-NB(深度)	84.1	**1.1**
CrossMine	**88.2**	1.6
FOIL	61	9138
TILDE	79	355
1BC	—	—
1BC2	—	—
Mr-SBC	81	42

表 4 Mutagenesis 数据库 BK_2 运行结果

算 法	准确度/%	运行时间/秒
Graph-NB(without pruning)	78.1	1.1
Graph-NB(深度)	86.2	1.1
CrossMine	88.8	1.4
FOIL	83	**0.5**
TILDE	85	221
1BC	87.2	—
1BC2	72.9	—
Mr-SBC	**89.9**	48

表 5 Mutagenesis 数据库平均运行结果

算 法	准确度/%
Graph-NB(without pruning)	77.5
Graph-NB(深度)	82.3
CrossMine	81.9
FOIL	68.3
TILDE	79.7
Mr-SBC	**82.4**

从实验结果中可以看出，Graph-NB 算法运用剪裁策略后的准确度有较大提高。与其他算法相比，其运行时间比较短，执行效率较高。算法平均准确度位居第二，仅比第一名 Mr-SBC 的准确度低

0.1%,在三种背景下的准确度均处于较高的水平。

4.2 金融数据库

第二组实验所使用的数据是来自 PKDD CUP99 中的一个金融数据库。我们对数据库做了与文献[5]相同的修改,同样也使用了十倍交互验证方法。最终的数据库包括 8 个数据表和 75982 条记录。目标表 Loan 中含有 400 条记录,目标属性有两种类别,分别有 324 条和 76 条记录。表 6 给出了各种算法的平均运行结果,其中,TILDE 的结果参照文献[5]。

表 6 金融数据库平均运行结果

算 法	准确度/%	运行时间/秒
Graph-NB(深度)	85.25	2.94
Graph-NB(广度)	85.25	**2.91**
Graph-NB(without pruning)	79.2	1.9
CrossMine	**87.25**	13.4
FOIL	71.5	3479.3
TILDE	81.3	2429

从实验结果同样可以看到,Graph-NB 算法所采用的优化策略效果显著。与其他算法相比,其运行时间最短,准确度仅比最高的 CrossMine 低 2%,但时间快了一个数量级。深度与广度优先两种策略在这个数据集上的分类准确度一样,广度优先比深度优先所使用的平均时间略少,这很有可能是深度优先所采用的递归方法效率不是很高所导致的。

4.3 合成数据库

第三组实验用来比较深度优先与广度优先这两种策略的优劣性,它所使用的数据是用文献[5]中的数据生成器产生的。表 7 给出了数据生成器的详细参数指标。

表 7 数据生成器参数指标

名 称	描 述	定 义
$\|R\|$	# relation	x
T_{min}	Min # tuples in each relation	50
T	Expected # tuples in each relation	y
A_{min}	Min # attributes in each relation	2
A	Expected # attributes in each relation	5
V_{min}	Min # values in each relation	2
V	Expected # values in each relation	10
F_{min}	Min # foreign-keys in each relation	2
F	Expected # foreign-keys in each relation	z
$\|r\|$	# rules	10
L_{min}	Min # complex predicates in each rule	2
L_{max}	Max # complex predicates in each rule	6
f_A	Prob. of a predicate on active relation	0.25

为了测试算法在数据库表个数及记录数不同的情况下的准确度与运行速度,我们固定其他参数不变,只变动表个数与记录数两个参数来生成测试数据集 $RxTy$。其中 x 代表数据集中表的个数,y

代表生成表中期望的记录个数，数据集中表的外键个数设为 1。另外，生成的数据集中前 90%用作训练集，剩下的 10%作为测试集，所得到的分类准确度是测试集上的准确度。表 8 列出了两种策略在生成的 6 个数据库上的测试结果。

表 8 合成数据库运行结果(深度、广度优先)

数据集	深度优先		广度优先	
	准确度/%	运行时间/秒	准确度/%	运行时间/秒
R10T1000	81.0	16.0	81.0	4.4
R10T3000	87.3	3.8	**90.0**	3.7
R10T6000	96.2	4.1	96.2	3.9
R10T10000	70.5	23.4	70.5	24.9
R15T1000	69.0	4.8	69.0	3.0
R20T1000	87.0	12.0	87.0	3.7
Average	81.8	10.7	82.3	**7.2**

从以上实验的结果来看，广度优先策略比深度优先策略在算法执行效率上要高，平均运行时间要少。至于分类准确度，在广度优先策略下仅有 R10T3000 一个数据集上的分类准确度比深度优先策略的高，其他数据集的准确度都相等。这是因为广度优先策略最终所得到的语义关系图的表结点数一般要小于等于深度优先策略所得到的 SRG。广度优先策略先处理与目标表相近的表，然后再扩散到其他表，而深度优先策略要先处理完一个分支上所有表后才可以连接其他分支上的表。对于关系比较复杂或者 SRG 结点数比较多的数据库，广度优先策略能够利用数目较少的表来达到与深度优先策略相同或更高的分类准确度。整体上来看，可以说广度优先策略要优于深度优先策略。

5 总结

本文提出了一种新型的多关系朴素贝叶斯分类算法，Graph-NB，它将与目标表相连接的表进行选择，以便将关系弱或无关的表排除在分类模型之外。实验结果表明，语义关系图的优化对于准确度的提高效果显著。同时，Graph-NB 与 CrossMine 和 Mr-SBC 等多关系分类算法相比，Graph-NB 一般能够达到较高的分类准确度，运行效率高。另外，算法中广度优先策略得到的优化的语义关系图比深度优先策略的要精简，且广度优先策略需要的运行时间比深度优先策略要少，但广度优先策略却能够达到与深度优先策略相同或更高的分类准确度，所以整体而言，广度优先策略要优于深度优先策略。

参考文献

[1] Kramer S, N Lavrac, P. Flach. Propositionalization approaches to relational data mining[C]//S Dzeroski, N Lavrac. Relational Data Mining. Germany: Springer-Verlag, 2001: 262-291.

[2] Muggleton S. Inductive Logic Programming [M]. New York, NY: Academic Press, 1992.

[3] Quinlan J R, Cameron-Jones R M FOIL: A midterm report [C]//Proc 1993 European Conf Machine Learning. Vienna, Austria: Springer-Verlag, 1993: 3-20.

[4] Blockeel H, De Raedt L, Ramon J. Top-down induction of logical decision trees [J]. Artificial Intelligence 1998 (101)1-2: 285-297.

[5] Yin X, Han J, Yang J, Yu P S. CrossMine: Efficient Classification across Multiple Database Relations [C]// Ozsoyoglu M, Zdonik S. Proc 2004 Int Conf on Data Engineering (ICDE'04), Boston, MA: 2004: 399-410.

[6] Pompe U, Kononenko I. Naive Bayesian classifier within ILP-R[C]//L. De Raedt. Proc of the 5th Int Workshop on Inductive Logic Programming. Katholieke Universiteit Leuven, 1995: 417-436.

[7] Flach P, Lachiche N. 1BC: A first-order Bayesian classifier[C]//Proceedings of the 9th International Workshop on Inductive Logic Programming. London: Springer-verlag, 1999: 92-103.

[8] Lachiche, N., Flach P. 1BC2: a true first-order Bayesian Classifier [C]//Claude Sammut, Stan Matwin. Proceedings of the 12th International Conference on Inductive Logic Programming. Germany: Springer Berlin / Heidelberg, 2002: 133-148.

[9] Ceci M, Appice A, Malerba D. Mr-SBC: a Multi-Relational Naive Bayes Classifier[C]//N Lavrac, D Gamberger, L Todorovski, H Blockeel et al. Knowledge Discovery in Databases, Lecture Notes in Artificial Intelligence. 2838. Berlin: Springer, 2003: 95-106.

[10] Flach P , Lachiche N. First-order Bayesian Classification with 1BC. Submitted. [EB/OL]. http://hydria.u-strasbg.fr/~lachiche /1BC.ps.gz.

[11] Friedman N, Getoor L, Koller D, Pfeffer A. Learning Probabilistic Relational Models[C]//Thomas Dean. Proc of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI). San Francisco: Morgan Kaufmann, 1999: 259-266.

[12] Taskar B, Segal E, Koller D. Probabilistic Classification and Clustering in Relational Data[C]//Proc of 17th International Joint Conference on Artificial Intelligence. San Francisco: Morgan Kaufmann, 2001: 870-878.

[13] Emde W, Wettschereck D. Relational Instance-Based Learning. [C]//Saitta L. Proc 13th International Conference on Machine Learning. San Francisco: Morgan Kaufmann, 1996: 122-130.

[14] Neville J, Jensen D, Gallagher B, Fairgrieve R. Simple Estimators for Relational Bayesian Classifiers[C]// Proceedings of the third IEEE International Couference on Data Mining. Melbourne, FL, 2003: 609-612.

[15] Neville J, Jensen D, Friedland L, Hay M. Learning Relational Probability Trees. Technical Report 02-55, Department of Computer Science, University of Massachusetts Amherst, 2002.

[16] Macskassy S, Provost F A Simple Relational Classifier [C]//Proc the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2nd workshop on Multi-relational Data Mining. New York, NK: ACM Press, 2003.

[17] Duda R, Hart P. Pattern Classification and Scene Analysis [M]. New York: John Wiley & Sons, 1973.

[18] Join G H. Enhancements to the Data Mining Process[D]. California: Stanford University, 1997.

[19] Domingos P, Pazzani M. Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier [C]//Saittal. Proc 13th Intl Conf Machine Learning. San Francisco, CA: Morgan Kaufmann: 1996: 105-112.

[20] Yin X. CrossMine software. [EB/OL] http://www-sal.cs.uiuc.edu/~hanj/pubs/software.htm.

Graph-NB: an Efficient and Accurate Multi-relational Naive Bayesian Classifier

LIU Hongyan[1], CHEN Hailiang[1], HAN Jiawei[2] & YIN Xiaoxin[2]

(1 School of Economics and Management, Tsinghua University, Beijing 100084

2 Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA)

Abstract Multi-relational classification is one of the most popular research areas in data mining. Current multi-relational Bayesian classifiers take all the tables connected with the target table into consideration, including the weakly-linked ones. In this paper, we propose a new Classifier, Graph-NB. It optimizes the semantic relationship graph by

cutting off some tables, and reduces the adverse effect of those weakly-linked tables thereby. It is implemented by both the depth-first and width-first traverse strategy. Experimental study shows that the optimization of the semantic relationship graph is effective on the increase of accuracy. Comparing with other multi-relational naive Bayesian classifiers, it has short run time and relatively high classification accuracy.

Key words Multi-relational classification, Naive Bayesian classifier, Depth-first, Width-first