

基于 Bayes 的有噪训练集去噪方法研究^{*})

罗俊杰 孙江文 王崇骏 陈世福

(南京大学计算机软件新技术国家重点实验室 计算机科学与技术系 南京 210093)

摘要 利用有噪训练集训练分类器的过程中,去噪是基本的预处理步骤。传统的去噪工作只是简单地删除被标记为噪声的实例。显然,这样处理会清除噪声实例中的有用信息。本文提出一种基于 Bayes 的去噪方法,不但能辨识出噪声而且能纠正噪声实例的错误类标,从而保证其有效信息不会丢失。

关键词 噪声,噪声辨别,噪声纠正

Identifying and Correcting Mislabeled Training Instances Using Bayes

LUO Jun-jie SUN Jiang-wen WANG Chong-jun CHEN Shi-fu

(National Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China)

Abstract De-noising is a basic pretreatment in the process of training a classifier. Most traditional de-noising approaches only delete instances tagged as noise which obviously also eliminates the useful information in these instances. A new approach is presented with which we can not only identify noise but also correct it, so that the useful information will be reserved.

Keywords Noise, Noise identifying, Noise correcting

1 引言

属性噪声和类别噪声是有噪训练集中的两种典型噪声^[1],属性噪声是指实例的某些属性有错误,而类别噪声则是指属性正确但类标错误的实例^[2]。显然,这些噪声会很大程度上影响目标分类的精度和泛化能力。本文的工作集中在通过使用机器学习方法辨识和纠正带有类别噪声的错误标签的实例,从而提高训练数据集的质量,最后达到提高分类精度的目的。

常规有两种方法用来处理有噪训练集。第一种是对噪声进行容错处理,如切割树剪枝操作^[3,4],LEEWAY^[5]就是一种这样的算法。第二种通过使用机器学习方法来辨识和纠正噪声,从而提高训练集的质量,Kubica 和 Moore^[6]使用此方法来辨别噪声区域和使用残留的干净区域的数据来进行建模和分析。Schwarm 和 Wolfman^[7]在获取领域知识的基础上重建无噪声数据集的潜在分配概率。

以上两种方法对于类别噪声和属性噪声都适用。然而,Quinlan^[8]证明:对于高层次的噪声而言,如果从属性信息中清除噪声,然后训练产生的分类器使用在有同样噪声的数据处理时,其精确度会相应地降低。在误将带噪声的实例分类为相反类别的情况下,清理训练集将会使分类器预测的精度变高。John^[9], Brodley 和 Friedl^[10,11]将噪声清除工作简化为过滤操作。前者通过充分移除外枝来扩展 C4.5 的剪枝操作;后者使用一些学习算法来生成一个分类器,作为过滤器去辨别和更正错误标签的训练实例。Gamberger, Lavrac 和

Groselj^[12]使用一个分类过滤器,将训练数据被分割成 n 个子集,然后用对其中任意 $n-1$ 个子集进行训练得到的分类器 H_i 来对剩下的一个子集进行分类,其中被 H_i 正确分类的实例是可以被识别的。Zhu, Wu 和 Chen^[2]将数据集分割成不同的子集,并且从子集中归纳出一些好的规则来评估整个训练数据集,同时将一些被认为是错误标签概率较高的实例清除。简单地清除那些被认为带有噪声的实例将会丢失许多有用的信息,所以对这些实例进行更正应该是更好的选择。基于这种思想,本文基于使用 Bayes 分类器去噪的新方法来检测和更正错误标签的实例。

本文的组织结构如下:第 2 节简单介绍一下信息熵,本文提出的算法都是基于此展开的;第 3 节给出了本文提出的算法描述;第 4 节进行了相关实验并进行了比较和分析;最后对全文进行了总结。

2 信息熵

本文使用信息熵来评估一个实例属于某个类标签的典型性程度。本节简单介绍一下信息熵的相关内容以及使用信息熵的动机。

X 是一个在 χ 范围内的离散随机变量,概率集合函数 $\rho(x) = P\{X=x\}, x \in \chi$; 随机变量的熵 $H(X)$ 可以表达为

$$H(X) = - \sum_{x \in \chi} \rho(x) \log \rho(x) \quad (1)$$

熵可以作为随机变量不确定性的度量值,本文用它来评估一个实例属于某个类标签的典型性程度。在式(1)中,可以看到当所有的可能性都相等时, $H(X)$ 取到最大值,这符合完

^{*})国家自然科学基金资助项目(No. 60503021),江苏省自然科学基金(No. BK2005075),江苏省高技术研究发展计划资助项目(No. BG2006027, No. BG2007038)。罗俊杰 硕士研究生,主要研究领域为机器学习、智能系统应用;孙江文 硕士研究生,主要研究领域为机器学习、数据挖掘;王崇骏 博士,副教授,主要研究领域为机器学习、数据挖掘、智能化信息处理;陈世福 教授,博导,主要研究领域为人工智能与知识工程。

全不确定的情况。而当其中的一个可能概率取到 1, 而其他的值为 0 时, $H(X)$ 取到最小值。因此, 如果一个实例典型性属于某个类标, 这个实例可能性分布概率的熵值(用 Bayes 分类器进行评估)等于 0 或者接近 0(这同样也符合前面的理论)。当一个实例是典型性的, 但是它的原始标签跟对其进行分类的结果不同时, 我们认为这是个被错误标签的实例, 同时使用预测的标签来替代原始的标签。在本文中, 用阈值 T 来分辨典型实例, 如果实例的熵比 T 低, 则该实例为典型性实例, 否则为一个异常。

本文使用错误标签的实例来表示被错误分类的实例, 用异常来表示不典型的属于任何一个标签的实例。在使用 Bayes 分类器对错误标签的实例和异常进行分辨时, 我们假定两者属于类标签的可能性概率的分布是不同的。在前者得到的可能性分配中, 其中的一个值等于 1 或者接近 1, 而其他值等于 0 或者接近 0, 而且这个最大可能性的类标并不是原始给定的类标签。而在后者得到的可能性分配中, 不存在这样一个类标的可能性接近 1。

3 算法

3.1 算法思想

使用 Bayes 分类器来对一个实例进行分类时, 我们能得到某个实例属于每个类标可能性的可能分布情况。当类标其中的一个可能性等于 1 或者非常接近 1, 且其他的等于 0 或者非常接近 0 时, 我们可以说这个实例属于该类标签的概率等于 1 或者接近 1。相反, 当所有的可能性都几乎相等时, 不应该认为这个实例属于一个典型的类标签。一般而言, 辨别噪声的最大难度是按照一般规则将错误标签的噪声误差从异常中分辨出来。Bayes 分类器可以用来处理这类问题。在部分训练数据的基础上训练生成的 Bayes 分类器可以用来对训练集其他的实例进行分类。我们可以使用实例属于每个类标签的可能性概率分布来辨别噪声误差和异常误差。如果存在原始标签和预测标签不同的实例, 而且该实例属于预测标签的可能性概率为 1 或者接近 1, 那么该实例可以认为是噪声, 而且我们将使用预测标签来纠正该实例。当然, 以上论断成立的前提是确认存在一个好的 Bayes 分类器, 能够生成精确的模型对训练数据进行泛化。近几年, 出现了许多性能良好的 Bayes 分类器, 比如 AODE^[13], HBN^[14] 和 FBN^[15]。在 FBN 中, 一个完整的 Bayes Net 被用作一个结构体, 决策树从每个 CPT 中学习而得到。在一个特定的范围, FBN 能比较精确地估计底层数据的分布, 且训练 FBN 的时间代价比较小。本文使用 FBN 作为 Bayes 分类器来进行实验。

3.2 算法描述

算法的详细描述参见算法 1 和算法 2。从算法 1 中可以看到, 首先调用算法 2 中的函数来评估给定训练数据集 S 中每个实例的熵。在算法 2 中可以看到, 为了评估熵 E , 我们开始要随机打乱 S 中实例的次序, 且将其分割成 N 个子集。给定任何一个子集 P_i , 从 $S \setminus P_i$ (除去子集 P_i 的其他部分) 中训练得到的 Bayes 分类器 C_i 被用来评估 P_i 中的每个实例的典型性。对于每一个实例 I_k , 我们将得到其属于某个类标签的分布可能性 D_k , 然后从我们使用式(1)从 D_k 能够得到熵 E_k 。为了得到阈值 T , 所有熵的值按照升序排列, 在数据集中用参数 α 表示典型性实例所占的比例, 这样就可以计算出 T 来。为了决定 α (数据集中典型性实例的比例) 的值, 一般有两种方法: (1) 可变量。根据实际噪声设置一个值; (2) 固定值。系

统自动为 α 设置一个固定值, α 的可能值是由经验得到的。因为, 如果使用第一种方法的话, 那使用者应该知道数据集中的噪声水平, 所以我们使用第二种方法。如果 E_k 比 T 小, 而且 I_k 的原始标签预测值不同的话, 该实例被标记为一个错误标签的实例。然后我们使用预测值来替换错误标签实例的类标签。

算法 1 数据清理

```

输入: S(具有 n 个实例的训练集)
参数:  $\alpha$ (典型性实例的比例)
输出: A(干净训练集)
E = 估值(S)
indexArraySorted = 升序排序(E)
index =  $\alpha * n$ 
T = E[indexArraySorted[index]]
FOR k=1 to n do
  if  $E_k < T$  并且 预测的标签  $\neq$  原始标签 执行
    将  $I_k$  的类标设置为预测的类标
  end if
END FOR
A = S
返回 A

```

算法 2 估值

```

输入: S(具有 n 个实例的训练集)
输出: E(实例集对应的熵数组)
对任意实例  $I_k \in S, E_k = 0$ 
打乱 S 中实例的排序
将 S 分成 N 个子集
FOR i=0 to N do
  从  $S \setminus P_i$  中学习生成一个 Bayes 分类器  $C_i$ 
  对  $P_i$  中的每个实例  $I_k$  执行
     $D_k = C_i$ . distributionForInstance( $I_k$ )
   $E_k =$  计算熵值( $D_k$ )
END FOR
return E

```

3.3 多轮次的异常和停止准则

在停止准则得到满足前, 要使用多轮次而不是仅一次噪声处理来辨别和纠正错误标签的实例; 在 M 个持续的轮次中, 如果该数据集没有再改变, 噪声的清除工作将停止。多轮次的程序在算法 3 中给出。我们将 α 设定为一个较小的值, 这样被辨识为典型性实例的实例就应该是真正的典型性实例。在我们的系统中, 每一轮都是使用相同的阈值 T 。对于相同的 T 值, 我们不用每轮都重设 α 的值。

算法 3 多轮次数据清理

```

输入: S(具有 n 个实例的训练集)
参数:  $\alpha$ (典型性实例的比例)
输出: A(干净训练集)
E = 估值(S)
indexArraySorted = 按升序排序(E)
index =  $\alpha * n$ 
T = E[indexArraySorted[index]]
重复
  FOR k=1 to n do
    IF  $E_k < T$  和 predictLabel  $\neq$  originalLabel
      then 将  $I_k$  的类标设置为预测的类标
    END IF
  END FOR
  E = 估值(S)
直到满足停止条件
A = S
返回 A

```

4 实验

4.1 实验设定

如前所述, 本文使用 FBN^[2] 作为本方法中的 Bayes 分类器, 同样, 使用 FBN 作为评估分类准确的基本分类器, 并利用 UCI 数据集进行实验, 以评估本方法对噪声分辨和更正的性能。其中 Mushroom 有 8124 个实例, anneal 只有 898 个实例。用这两个数据集来描述两种类型的数据集; 一种数量较大, 一种较小。由于 FBN 不能处理连续属性值, 我们在 weka 中使用无人监督离散操作离散化 Anneal 中的数值变量。

为了增加噪声,我们采用了文献[2]中的方法。考虑到现实世界中只有确定类别的类才有可能被错误地标签,我们采用的方法如下:给定一对类(X,Y)和一个噪声水平 x ,其中标签为 X 的实例有 $x * 100\%$ 的,就会被错误地标签为 Y,实例 Y 也同样。使用这种方法,整个训练集被错误标签的比例会小于 $x * 100\%$,因为只是其中的一部分类被认为有疑问。本文只是将有较高级别的实例中的一对添加噪声。

在我们的实验中,在每个数据集上一个算法在十分交叉验证的基础上运行 5 次,得到该算法的平均精度和标准偏差。在每一层中,我们通过上述添加噪声的方式来使训练集变脏,然后使用测试集来评估系统的性能。下文中,原始数据意味着已经添加了噪声的训练集。

我们采用了文献[2]中的 3 个参数:ER1,ER2 和 NEP(噪声清除精度)。用 ER1 表示一个正确实例被标记为噪声的事件,ER2 表示一个噪声实例被标记为一个正确类标实例的事件,它们的方程定义如下:

$$P(ER_1) = \frac{|F \cap M|}{|G|}$$

$$P(ER_2) = \frac{|F \cap M|}{|M|}$$

$$NEP = \frac{|F \cap M|}{|F|}$$
(2)

$|X|$ 表示在集合 X 中的实例数量, F 是被辨别为噪声且已经被修正的实例集; f 是 F 的补集; G 是正常实例集, M 是噪声集。

4.2 参数 α 的选择

本文将对 α 的四个不同值(1/4, 1/2, 3/4 和 1)在噪声鉴定中所带来的性能变化进行比较。本文使用一种分层算法将原始数据分割成 10 个子集。噪声鉴定的结果见表 1 和表 2。由表 1 可以看到,当 α 取 1/4, 1/2, 3/4 三个设置时,ER1 值的性能比最后一个设置要好(一般有 3% 的提高);在所有的四个设置(除了当 $\alpha=1/4$ 且噪声比例为 0.1 和 0.15)中 ER2 的值相当,NEP 的性能在前面三个设定值远好过最后的一个设定(一般有 5%~40% 的提高)。如图 1 中所示,随着噪声辨别和修正的提高,前面三个设定的分类的精确度也比最后一个要好(大概有 2%~4% 的提高)。前面三个设定的性能比最后一个要好的原因是 anneal 的大小相对较小,但却有 23 个属性。如果添加噪声,每个实例的熵都提高了,实例变得不稳定了。所以,当 α 取值为 1 时,更多的错误生成了。同样如图 2 和表 2 所示,前面三个设定的对应噪声辨别和清除的性能仅比最后一个要好一点。在前面的三个设定中,当 α 的值取 1/2 时,噪声分辨和清除的性能和分类的准确性比其他两个

要好。所以在下文中,除非特殊情况下我们都将 α 的值设定为 1/2。

4.3 多轮次执行

当 α 取一个合适的值时,使用多轮次的执行能够预防将正常的实例标记为噪声,也能渐进式地分辨噪声。在我们的实验中,大多数的执行能够在少于 15 轮的时候就满足停止法则。我们把在不同噪声水平下每一轮噪声被清除的比例记录下来,结果示在图 3 中, x 轴表示的是执行的轮数, y 轴表示的是到每轮分辨噪声的百分比。图 3 显示在 anneal 数据集的情况下,在不同噪声水平下,首轮就能分辨出 37%~61% 的噪声,大多数噪声能够在 10 轮以内的执行中被发现。

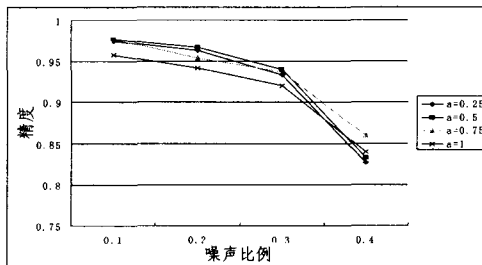


图 1 分类精度 (anneal)

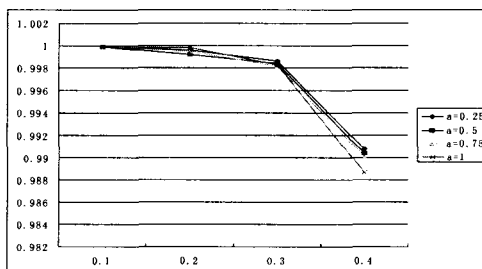


图 2 分类精度 (mushroom)

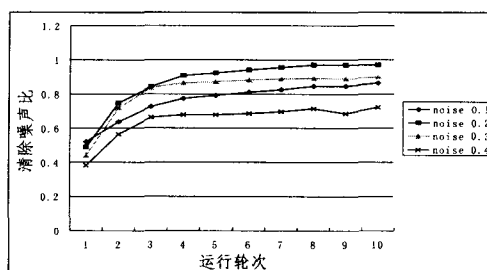


图 3 多轮次噪声清除

表 1 α 选择 (anneal)

NOISE (%)	$\alpha = 1/4$			$\alpha = 1/2$			$\alpha = 3/4$			$\alpha = 1$		
	P(ER1)	P(ER2)	NEP	P(ER1)	P(ER2)	NEP	P(ER1)	P(ER2)	NEP	P(ER1)	P(ER2)	NEP
10	0.0013 ± 0.0006	0.4722 ± 0.0409	0.9604 ± 0.0192	0.0042 ± 0.0013	0.1342 ± 0.0325	0.9105 ± 0.0278	0.0071 ± 0.0013	0.033 ± 0.0147	0.8653 ± 0.0167	0.03 ± 0.0039	0.0362 ± 0.0087	0.6035 ± 0.033
20	0.0161 ± 0.0393	0.2051 ± 0.0393	0.913 ± 0.0111	0.0203 ± 0.0024	0.0456 ± 0.0082	0.9104 ± 0.0095	0.024 ± 0.0027	0.0415 ± 0.0086	0.8959 ± 0.0101	0.0471 ± 0.0049	0.0677 ± 0.0082	0.813 ± 0.0139
30	0.0379 ± 0.0089	0.1243 ± 0.0247	0.8919 ± 0.0207	0.0384 ± 0.007	0.0806 ± 0.0138	0.8954 ± 0.0168	0.0433 ± 0.007	0.0793 ± 0.0144	0.8833 ± 0.0167	0.0643 ± 0.0105	0.1004 ± 0.0135	0.8339 ± 0.022
40	0.1037 ± 0.0079	0.2353 ± 0.0256	0.8026 ± 0.0214	0.1048 ± 0.014	0.1894 ± 0.0123	0.8088 ± 0.0176	0.0971 ± 0.005	0.1675 ± 0.0143	0.8251 ± 0.0145	0.1203 ± 0.017	0.1906 ± 0.0105	0.7884 ± 0.031

4.4 算法比较

上述实验表明,当 $\alpha=1$ 时,本方法与文献[12]中方法相

当。我们同时也运行了 Zhu, Wu 和 Chen^[2]中使用的分区过滤的方法(分成 5 个子集,噪声辨别的非客体方法和将阈值固

定设定为 $a=0.6$ 。最后的结果记录在表 3、表 4(噪声清除的性能)和表 5(分类精确度)中。第一个属性表示的是噪声

水平,其他属性表示的是从不同数据集中得到的结果,PF 表示的是分割过滤,EBF 表示我们用的基于熵的过滤。

表 2 a 选择 (mushroom)

NOISE (%)	$a = 1/4$			$a = 1/2$			$a = 3/4$			$a = 1$		
	P(ER1)	P(ER2)	NEP	P(ER1)	P(ER2)	NEP	P(ER1)	P(ER2)	NEP	P(ER1)	P(ER2)	NEP
10	0.0001 ± 0.0002	0.2599 ± 0.0316	0.9985 ± 0.0028	0 ± 0	0.0028 ± 0.0039	1 ± 0	0 ± 0	0 ± 0	1 ± 0	0 ± 0	0.0001 ± 0.0002	0.9997 ± 0.0004
20	0.0001 ± 0.0001	0.0765 ± 0.0316	0.9997 ± 0.0018	0.0001 ± 0.0002	0.0003 ± 0.0024	0.9995 ± 0.0004	0.0001 ± 0.0002	0.0002 ± 0.0003	0.9997 ± 0.0001	0 ± 0	0 ± 0	1 ± 0.0005
30	0.0012 ± 0.002	0.0182 ± 0.0041	0.9972 ± 0.0012	0.0009 ± 0.0003	0.0021 ± 0.0014	0.9978 ± 0.0023	0.0014 ± 0.0007	0.0027 ± 0.0001	0.9967 ± 0.0002	0.001 ± 0.0021	0.002 ± 0.0023	0.9977 ± 0.003
40	0.0052 ± 0.003	0.0101 ± 0.0051	0.992 ± 0.0023	0.0054 ± 0.0014	0.0098 ± 0.0011	0.9919 ± 0.0021	0.0059 ± 0.0003	0.0114 ± 0.0003	0.991 ± 0.0007	0.0071 ± 0.0026	0.0113 ± 0.0018	0.9893 ± 0.005

从表 3 和表 4 能看到,在小数据集中,比如 anneal,虽然 PF 算法在从 0.1 到 0.35 的低噪声水平情况下 NEP 能得到较好的性能(相比较大概有 2%~7%的提高),但是却将更多的噪声标记为正常实例(0%~78%)。也就是说,PF 算法中的 ER2 的值远大于 EBF 中的值。如表 5 所示,EBF 算法的分类精度比 PF 要好(大概有 0%~7%提高)。在噪声等级为 4 的情况下,ER2,NEP 和分类精度都比 PF 算法要好,只是 ER1 比 PF 稍差(大约 10%)。在大的数据集中,比如 mushroom,在噪声水平为 0~0.3 时,两种算法在噪声辨认和分类精度上都差不多。但是在噪声水平为 0.35~0.4 时,EBF 算法中的噪声辨别和分类精度上都比 PF 要好(相比较 ER1 有 40%~85%的提高,NEP 有 2%~7%的提高,分类精度有 2%~10%的提高)。这些结果表明,EBF 不但能很好地辨别噪声,而且能将其更正。

表 3 和 PF 的比较 (anneal)

NOISE (%)	EBF			PF		
	P(ER1)	P(ER2)	NEP	P(ER1)	P(ER2)	NEP
10	0.0042 ± 0.0013	0.1342 ± 0.0325	0.9105 ± 0.0278	0.0022 ± 0.0003	0.135 ± 0.0277	0.9746 ± 0.0029
20	0.0203 ± 0.0024	0.0456 ± 0.0082	0.9104 ± 0.0095	0.0025 ± 0.0004	0.2529 ± 0.0434	0.9837 ± 0.0031
30	0.0384 ± 0.007	0.0806 ± 0.0138	0.8954 ± 0.0168	0.0041 ± 0.0012	0.6776 ± 0.073	0.9589 ± 0.0116
40	0.1048 ± 0.014	0.1894 ± 0.0123	0.8088 ± 0.0176	0.0051 ± 0.0012	0.9549 ± 0.0055	0.7951 ± 0.045

表 4 和 PF 的比较 (mushroom)

NOISE (%)	EBF			PF		
	P(ER1)	P(ER2)	NEP	P(ER1)	P(ER2)	NEP
10	0 ± 0	0.0028 ± 0.0039	1 ± 0	0 ± 0	0.0329 ± 0.0061	1 ± 0
20	0.0001 ± 0.0002	0.0003 ± 0.0024	0.9995 ± 0.0004	0.0001 ± 0	0.1127 ± 0.0059	0.9997 ± 0.0002
30	0.0009 ± 0.0003	0.0021 ± 0.0014	0.9978 ± 0.0023	0.0017 ± 0.0002	0.2676 ± 0.0101	0.9944 ± 0.0007
40	0.0054 ± 0.0014	0.0098 ± 0.0011	0.9919 ± 0.0021	0.0079 ± 0.001	0.8569 ± 0.0249	0.9196 ± 0.0057

表 5 EBF 和 PF 的分类精度对比

NOISE (%)	ANNEAL		MOSHROOM	
	EBF	PF	EBF	PF
10	0.9785 ± 0.004	0.9833 ± 0.001	1 ± 0	1 ± 0.0001
20	0.9674 ± 0.005	0.9698 ± 0.004	0.9996 ± 0	0.9991 ± 0.001
30	0.9415 ± 0.008	0.9111 ± 0.012	0.9989 ± 0	0.9928 ± 0.001
40	0.8385 ± 0.007	0.7787 ± 0.015	0.991 ± 0.002	0.8864 ± 0.007

方法,使得在辨别噪声的同时能更正噪声的错误标签,从而保证其有效信息不会丢失。我们使用了一个 Bayes 分类器来模仿训练数据的潜在分布,同时使用该分类器得到每个实例属于某个类标签的概率。本方法从每个实例的分布概率得到的信息熵来评估该实例属于某个类标签的典型性。在每一轮中,熵值比阈值要小但错分类的实例被标记为噪声,这时用预测值来替代原始标签。实验结果表明,本方法在低噪声情况下,其性能跟 PF 相当或者略好;在高噪声的情况下,则一定比 PF 算法性能要好。

本文没有处理检测到的异常值,这应该是一件意义重大的事,我们将在以后的工作中进行这方面的研究。

参考文献

- [1] Mingers J. An empirical comparison of pruning methods for decision tree induction. *Maching Learning*, 1989, 4(2): 227-243
- [2] Quinlan J R. C4. 5: programs for machine learning. San Francisco, CA, USA: Morrigan Kaufmann Publishers Inc., 1993
- [3] Xia Y Y Y, Muntz R R. Learning naive bayes classifier from noisy data. Technical Report CSD-TR No. 030056. UCLA, 2003
- [4] Schwarm S, Wolfman S. Cleaning data with bayesian methods. 2000
- [5] Kubica J, Moore A. Probabilistic noise identification and data cleaning // The Third IEEE International Conference on Data Mining. IEEE Computer Society, 2003: 131-138
- [6] John G H. Robust decision tree: Removing outliers from databases. *Knowledge Discovery and Data Mining*, 1995: 174-179
- [7] Quinlan J R. Induction of decision trees. *Machine Learning*, 1986, 1(1): 81-106
- [8] Brodley C E, Friedl M A. Identifying and eliminating mislabeled training instances // AAA/IAAI, 1996, 1: 799-805
- [9] Brodley C E, Fredl M A. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 1999, 11: 131-167
- [10] Gamberger D, Lavarac N, Groselj C. Experiments with noise filtering in a medical domain // Proc. 16th International Conf. on Machine Learning. Morgan Kaufmann, San Francisco, CA, 1999: 143-151
- [11] Zhu X, Wu X, Chen Q. Eliminating class noise in large datasets // Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003). Washington DC, 2003: 920-927
- [12] Webb G I, Boughton J R, Wang Z. Not so naive bayes: Aggregating one-dependence estimators. *Machine Learning*, 2005, 58(1): 5-24
- [13] Zhang L J H, Su J. Hidden naive bayes // Proceedings of the 20th National Conference on Artificial Intelligencen. AAAI Press, 2005: 919-924
- [14] Su J, Zhang H. Full bayesian network classifiers // ICML'06: Proceedings of the 23rd International Conference on Machine Learning. New York, NY, USA, ACM Press, 2006: 897-904
- [15] Wu X. Knowledge acquisition from databse. Ablex Pulishing Corp., 1995

结束语 与传统的去噪方法不同,本文提出了一种新的