计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
44(8): 1293~1300, 2007

基于频繁项集挖掘的贝叶斯分类算法

眭俊明　姜　远　周志华
(南京大学计算机软件新技术国家重点实验室　南京　210093)
(xujm@lamda.nju.edu.cn)

Bayesian Classifier Based on Frequent Item Sets Mining

Xu Junming, Jiang Yuan, and Zhou Zhihua
(*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210093)

Abstract　Naïve Bayesian classifier provides a simple and effective way to classifier learning, but its assumption on attribute independence is often violated in real-world applications. To alleviate this assumption and improve the generalization ability of Naïve Bayesian classifier, many works have been done cy researchers. AODE ensembles some one-dependence Bayesian classifiers and LB selects and combines long item sets providing new evidence to compute the class probability. Both of them achieve good performance, but higher order dependence relations may contain useful information for classification and limiting the number of item sets used in classifier may restricts the benefit of item sets. For this consideration, a frequent item sets mining-based Bayesian classifier, FISC (frequent item sets classifier), is proposed. At the training stage, FISC finds all the frequent item sets satisfying the minimum support threshold *min-sup* and computes all the probabilities that may be used at the classification time. At the test stage, FISC constructs a classifier for each frequent item set contained in the test instance, and then classifies the instance by ensembling all these classifiers. Experiments validate the effectiveness of FISC and show how the performance of FISC varies with different *min-sup*. Based on the experiment result, an experiential selection for *min-sup* is suggested.

Key words　machine learning; Bayesian classification; semi-naïve Bayesian classification; frequent item sets mining; ensemble learning

摘　要　朴素贝叶斯分类器是一种简单而且高效的分类学习算法,但是它所要求的属性独立性假设在真实世界应用中经常难以满足. 为了放松属性独立性约束以提高朴素贝叶斯分类器的泛化能力,研究人员进行了大量的工作. 提出了一种基于频繁项集挖掘技术的贝叶斯分类学习算法FISC (frequent item sets classifier). 在训练阶段,FISC找到所有频繁项集并计算可能用到的概率估值. 在测试阶段,FISC对于测试样本包含的每个项集构造一个分类器,通过集成这些分类器来给出预测结果. 实验结果验证了FISC的有效性.

关键词　机器学习;贝叶斯分类;半朴素贝叶斯分类;频繁项集挖掘;集成学习

中图法分类号　TP181

朴素贝叶斯分类器(naïve Bayesian classifier)是一种简单高效的分类器,在很多情况下能够取得和一些相对复杂的分类器相当的分类精度[1]. 但由于其所依赖的属性独立性假设在真实问题中往往并不

收稿日期:2007-03-14;修回日期:2007-05-08
基金项目:国家自然科学基金项目(60635030);江苏省自然科学基金项目(BK2005412)

成立,围绕着如何放松独立性假设,使得贝叶斯方法在属性间依赖性很强的情况下仍然能够取得良好的性能,研究人员做了大量的工作.

在现有的放松独立性假设的扩展方法中,AODE[2]和LB[3]算法取得了很好的效果. AODE通过集成若干1-依赖分类器来放松独立性假设,同时通过对1-依赖分类器的选择加以限制来避免模型选择带来的不利影响;LB算法在训练阶段结合频繁项集挖掘算法挖掘出高阶的并且对分类提供新的信息的频繁项集,在分类的过程中,从这些项集中选择一部分来改进朴素贝叶斯分类器.

在真实问题中,属性间的相关性可能同时存在于多个属性之间,仅使用1-依赖分类器难以很好地反映高阶相关性;而LB算法在分类阶段仅使用了一部分频繁项集,没有充分利用频繁项集所包含的信息. 基于上述考虑,本文提出了一种基于频繁项集挖掘的贝叶斯分类算法FISC(frequent item sets classifier),在训练阶段通过改进的Apriori[4-6]频繁项集挖掘算法得到频繁项集,计算并保存可能使用的概率估值;在分类阶段,通过集成若干基于频繁项集的分类器来放松属性独立性假设.

1 研究背景

1.1 朴素贝叶斯分类器及其改进

假设数据有 n 个属性,每个样本表示为 $\boldsymbol{X}=\langle x_1, x_2, \cdots, x_n\rangle$,其中 x_i $(1\leqslant i\leqslant n)$是样本 $\boldsymbol{X}$ 在第 i 个属性 A_i 上的取值. 样本可能属于 c 个类别 $y_1, y_2, \cdots, y_c$ 中的一个. 下文中用 $y\in\{y_1, y_2, \cdots, y_c\}$ 来指代 c 个类别中的任意一个类. 根据贝叶斯定理,样本 $\boldsymbol{X}$ 属于类别 y 的概率 $P(y|\boldsymbol{X})$ 可表示为

$$P(y\ |\boldsymbol{X})=\frac{P(y)P(\boldsymbol{X}\ |y)}{P(\boldsymbol{X})}. \tag{1}$$

在分类器训练过程中,$P(y)$, $P(\boldsymbol{X}|y)$必须通过训练数据来估计. 在实际应用中,训练数据的数目并不足以精确地估计这些后验概率. 朴素贝叶斯分类器假设在给定类别的情况下,所有属性之间是独立的,这样就可以通过下面的式子来估计 $P(\boldsymbol{X}|y)$:

$$P(\boldsymbol{X}\ |y)=\prod_{i=1}^{n}P(x_i\ |y). \tag{2}$$

Domingos和Pazzani认为,只要朴素贝叶斯分类器对于给定样本属于各个类别概率的序是正确的,即使在属性独立性假设不成立的条件下,朴素贝叶斯仍然可能有很好的分类效果[1]. 但是,有很多研究表明,在属性独立性假设不成立时,朴素贝叶斯分类器的表现可能会比较差.

为了放松属性独立性假设,研究人员进行了很多工作,大致有以下思路:1)对贝叶斯分类器使用的属性集进行操作,如Langley等人采用前向选择来产生一个属性子集用于构建朴素贝叶斯分类器[7],Pazzani使用后向选择来产生属性子集[8];2)修正贝叶斯分类器所使用的概率估计[9];3)将样本空间划分为若干满足属性独立性假设的子空间,如RBC方法[10]使用树结构递归地将样本空间划分为若干个满足属性独立性假设的子空间,NBTree[11]将朴素贝叶斯分类器和决策树相结合;4)使用贝叶斯网络考虑属性间的相关性,如Friedman和Goldszmidt等人提出的TAN[12](tree augmented naïve Bayes)模型;5)将若干贝叶斯分类器集成起来,如石洪波等人提出了TAN的不稳定构造算法,然后采用Boosting的方法提高其分类性能[13].

下面具体介绍两种性能出色的改进贝叶斯分类器——AODE(aggregating one-dependence estimators)[2]和LB(large Bayes)[3].

1.1.1 AODE

在Sahami[14]提出的KDB(k-dependence Bayesian classifier)中,每个属性依赖于类属性和最多 k 个其他属性. 为了提高效率,AODE[2]仅使用1-依赖分类器,只需要一张3维的表来保存概率值,减少了时间和空间的开销.

对于每个属性选择它所依赖的属性会带来额外的计算开销,也会增加分类器的易变性(variance). 为了避免模型选择,AODE选择那些被其他所有属性依赖的属性,然后用这些属性构造若干1-依赖分类器,并将这些分类器对每个类别的预测集成起来. 同时,为了避免由于样本少而导致概率估值不准确,只选择那些测试样本对应的属性取值在训练样本中出现次数大于某个阈值的1-依赖分类器.

贝叶斯分类器按照式(3)给出对于 $\boldsymbol{X}$ 的预测:

$$\operatorname*{argmax}_{y}P(y, \boldsymbol{X}). \tag{3}$$

$P(y, \boldsymbol{X})$可以写成下面的形式:

$$P(y, \boldsymbol{X})=P(y, x_i)P(\boldsymbol{X}\ |y, x_i). \tag{4}$$

因为式(4)对每个 x_i 都成立,它对于各组属性值的平均值也成立,即

$$P(y, x)=\frac{\sum\limits_{i:1\leqslant i\leqslant n\wedge F(x_i)\geqslant m}P(y, x_i)P(\boldsymbol{X}\ |y, x_i)}{|\{i:1\leqslant i\leqslant n\wedge F(x_i)\geqslant m\}|}, \tag{5}$$

式(5)中 $F(x_i)$ 是属性 A_i 取值为 x_i 的训练样本的数目, m 是一个阈值, 用来控制条件概率估计的可靠性

由式(5)导出的分类器就是 AODE. 由于分母对各类别都相同, 因此可以根据式(6)来进行分类:

$$\operatorname{argmax}_y \left(\sum_{i:\, 1\leqslant i\leqslant n \wedge F(x_i)\geqslant m} \hat{P}(y, x_i) \prod_{j=1}^{n} \hat{P}(x_j \mid y, x_i) \right), \tag{6}$$

其中, $\hat{P}(y, x_i)$ 和 $\hat{P}(x_j \mid y, x_i)$ 是 $P(y, x_i)$ 和 $P(x_j \mid y, x_i)$ 在训练数据集上的估计. 如果不存在 i 满足 $1\leqslant i\leqslant n \wedge F(x_i)\geqslant m$, AODE 就使用朴素贝叶斯分类器对样本分类

1.1.2 LB(large Bayes)

文献[15]指出, 概率 $P(y \mid \boldsymbol{X})$ 可以采用不同的概率估值的乘积组合来近似, 每一种不同的组合代表了不同的独立性假设. 例如, $P(x_1, x_2, x_3)P(x_4, x_5 \mid x_1)$ 和 $P(x_1, x_2, x_3)P(x_4 \mid x_1)P(x_5 \mid x_1, x_2)$ 可以看做是 $P(x_1, x_2, x_3, x_4, x_5)$ 的两种不同的近似估计. 因此, 选择包含更多信息的条件概率的乘积来近似 $P(\boldsymbol{X} \mid y)$, 能够一定程度地放松属性独立性假设. 条件概率中的条件属性是几个属性取值的组合, 在频繁项集挖掘中被称做一个项集. 这样每个条件概率对应了一个项集. LB[3] 就是通过在训练阶段选取一些满足条件的项集, 来为分类时选择更好的概率乘积组合.

为了保证项集对应的概率估计是可信的, 该项集在数据集中出现的次数必须足够多. LB 利用类似 Apriori 的频繁项集挖掘算法找出训练集中的频繁项集. 由于 LB 仅使用有限项乘积来估计概率, 因此必须优先选择那些能为分类提供新信息的项集. 为此, LB 为每个项集定义了兴趣度. 兴趣度用项集所有低一阶的子集来近似估计该项集的偏差来定义, 偏差越大代表该项集包含的信息越多. 在频繁项集挖掘过程中, 加入了兴趣度约束来保证得到的项集包含对分类有用的信息; 在分类阶段, 同样也优先选择兴趣度高的项集参与分类.

1.2 频繁项集挖掘算法

频繁项集挖掘是挖掘关联规则、因果关系、时序序列、多维模式、最大模式等很多重要任务的基础. 目前常用的算法有 Apriori[4-6] 和 FP-Tree[16] 等. Apriori 使用了关于频繁项集性质的先验知识, 即所有频繁项集的非空子集一定也是频繁项集. 该算法采用层次式搜索, 利用频繁 k-项集来生成频繁 $(k+1)$-项集. 首先, 从数据库中找到频繁 1-项集, 记做 $\boldsymbol{L}_1$. $\boldsymbol{L}_1$ 用来生成 $\boldsymbol{L}_2$, 即频繁 2-项集. 然后用 $\boldsymbol{L}_2$ 生成 $\boldsymbol{L}_3$, 直到不存在更多的频繁 k-项集. 每一次生成 $\boldsymbol{L}_k$ 都需要扫描数据库一遍. Apriori 采用的先验知识可以有效地减小搜索空间, 从而提高了生成频繁项集的效率.

2 FISC 算法

在真实问题中, 属性间的关联性有可能是在多个属性之间存在的, AODE 难以很好地反映这种高阶的相关性. 适当放松这一限制, 在效率和分类精度之间取得更好的折中, 或者根据具体的需要来选择合适的平衡, 将带来更大的灵活性. 从频繁项集的角度看, AODE 中选择的属性可以看做是使用了一阶频繁项集. 使用更高阶的频繁项集将更多地考虑属性间的相关性, 更大程度地放松属性独立性假设. LB 算法使用了频繁项集挖掘算法来产生高阶频繁项集, 但是它仅使用一个分类器, 只使用了有限的部分频繁项集. 而且 LB 引入了兴趣度度量, 还有一系列的规则从众多频繁项集中选择一部分来对测试样本进行分类. 这种模型选择加大了计算开销, 同时浪费了许多对于分类有用的频繁项集.

综合上面的考虑, 为了尽可能多地利用频繁项集中包含的对于分类有用的信息, 本文提出了基于频繁项集的贝叶斯分类算法 FISC 算法. FISC 使用 Apriori 算法来挖掘训练数据集中的频繁项集, 对于每个项集构造一个 k-依赖分类器, 集成多个分类器的结果对测试样本进行分类.

作为贝叶斯分类器, FISC 同样通过下面的公式来对样本 X 进行分类:

$$\operatorname{argmax}_y \hat{P}(y \mid \boldsymbol{X}) \propto \operatorname{argmax}_y \hat{P}(y, \boldsymbol{X}). \tag{7}$$

设 $\mathscr{P}(\boldsymbol{X})$ 为集合 $\{x_1, x_2, \cdots, x_n\}$ 的超集, S 为频繁项集挖掘算法作用在训练集上得到的频繁项集的集合, 则其中的 $P(y, \boldsymbol{X})$ 通过式(8)估计:

$$\hat{P}(y, \boldsymbol{X}) = \frac{\sum_{\chi \in \mathscr{P}(\boldsymbol{X}) \cap S} \hat{P}(y, \chi, \boldsymbol{X})}{\sum_{y'=y_1}^{y_c} \left(\sum_{\chi \in \mathscr{P}(X) \cap S} \hat{P}(y', \chi, \boldsymbol{X}) \right)}, \tag{8}$$

其中, χ 为 $\boldsymbol{X}$ 所包含的频繁项集

$$\hat{P}(y, \chi, \boldsymbol{X}) = \hat{P}(y, \chi)\hat{P}(\boldsymbol{X} \mid y, \chi) = \hat{P}(y, \chi) \prod_{i=1}^{n} \hat{P}(x_i \mid y, \chi) = \frac{\prod_{i=1}^{n} \hat{P}(x_i, y, \chi)}{\hat{P}^{n-1}(y, \chi)}. \tag{9}$$

在训练阶段, FISC 首先通过频繁项挖掘算法(这里使用 Apriori 算法)从训练集中找到频繁项集. 为了提高分类阶段的效率, 分类阶段可能用到的所有概率值都在训练阶段统计并保存下来. 这样, 就需要大量存储空间来存储这些数据. 限制频繁项集的最大数目能够保证所有的概率值都保存在限定容量的内存中. 设限定的存储容量为 $MaxMemoryVolume$, 则频繁项集的最大数目 $MaxNumPatterns$ 可以通过式(10)估计:

$$MaxNumPatterns = \frac{MaxMemoryVolume}{C + \sharp classes \times \sharp values}, \tag{10}$$

其中 $\sharp classes$ 为数据集中的类别数, $\sharp values$ 为所有属性不同取值数目的总和, C 为表示一个频繁项集所用的数据结构占用存储的容量, 由采用的表示频繁项集的数据结构决定. 在 Apriori 每次迭代结束时, 加入另一个终止条件, 判断当前已经产生的频繁项集数目是否已经超过 $MaxNumPatterns$. 如果超过, 则只将本轮迭代先产生的没有超过限制的那部分频繁项集加入其中, 并且终止迭代; 如果没有超过, 则算法进入下一次迭代.

对于得到的每个频繁项集 χ, 需要计算并存储与它相关的一些概率估计值, 包括 $P(\chi, y)$, $P(\chi, x_i, y)$. 其中 $P(\chi, y)$是该频繁项集与类别 y 的联合概率, $P(\chi, x_i, y)$为该频繁项集与类别 y 和其他所有属性某个取值的联合概率. 需要对每个类别和每个属性的每个取值求出这样的概率. 由于高阶的联合概率所能用到的样本相对较少, 采用 Laplace 修正来估计这些概率值:

$$P(\chi, y) = \frac{F(\chi, y) + 1}{F(\chi) + c}, \tag{11}$$

$$P(x_i, \chi, y) = \frac{F(x_i, \chi, y) + 1}{F(\chi, y) + v_i}, \tag{12}$$

其中, $F(\cdot)$表示在训练集中出现的次数, c 表示类别数, v_i 表示属性A_i 不同取值的数目.

为了增加算法的稳定性, $\chi = \varnothing$对于每个测试样本都看做是频繁项集, 也就是将朴素贝叶斯分类器作为若干分类器中的一个. 这样当 $\boldsymbol{S} = \varnothing$时, FISC 退化为朴素贝叶斯分类器.

在分类阶段, 对于测试样本 $\boldsymbol{X}$, 只需要找出 $\boldsymbol{X}$ 所包含的频繁项集, 并从相关数据结构中读出所需要的概率值, 就可以根据式(8)来计算它属于各类的概率, 然后根据式(7)给出该样本的类别. 算法的具体描述见图 1 所示:

Algorithm: FISC.
Input: Dataset, D; Minimum support threshold, min_sup; Test instance $\boldsymbol{X}$;
Output: the label y predicted by FISC;
Data structure: when χ is a frequent pattern, table $\chi.py$ records $P(\chi, y)$ for $y \in \{y_1, y_2, \cdots, y_c\}$, and table $\chi.pxy$ records $P(\chi, x_i, y)$ for every possible x_i and y.
Training Process:
① $S \leftarrow$ frequent item sets in D w. r. t. min_sup;
//using a method like Apriori;
② for each frequent item set $\chi \in \boldsymbol{S}$
③ for each class label $y \in \{y_1, y_2, \cdots, y_c\}$
④ calculate and record $\mathscr{P}(\chi, y)$ according to Eq. (11)
⑤ for each attribute A_i
⑥ for each value x_i of A_i
⑦ calculate and record $P(\chi, x_i, y)$ according to Eq. (12)
Test Process:
① $n \leftarrow$ number of attributes
② $M \leftarrow$ initial probability matrix
③ for each frequent item set $\chi \in \boldsymbol{S} \cap \mathscr{P}(\boldsymbol{X})$
④ for each class label $y \in \{y_1, y_2, \cdots, y_c\}$
⑤ $v \leftarrow 1$
⑥ for $i = 1$ to n
⑦ $v = v * P(\chi, x_i, y)$
⑧ $v = v / (P(\chi, y))^{n-1}$
⑨ $\boldsymbol{M}_y = \boldsymbol{M}_y + v$
⑩ normalize elements on each column of $\boldsymbol{M}$
output $\hat{y} = \underset{y}{argmax}\ \boldsymbol{M}_y$

Fig. 1 The FISC algorithm.
图 1 FISC 算法

算法惟一的参数 min_sup 使得算法具有一定的灵活性, 能够适应不同的应用. 可以从两个角度来理解参数 min_sup 对于 FISC 算法的影响. 首先, min_sup 的取值对构造集成的分类器的数量有影响. 当 min_sup 的值比较大时, 训练阶段产生的频繁项集就较少, 从而使得参与投票的分类器较少, 但是由于每个分类器所对应的频繁项集的支持度都比较高, 每个分类器所使用的概率估计可能会相对准确; 相反, 如果 min_sup 的值比较小, 产生的频繁项集会比较多, 但是有些分类器所使用的概率估计可能不是很可靠. 其次, min_sup 可以用来适应属性相关性对分类的影响. 例如当数据集属性间的独立性很强时, 可以认为满足朴素贝叶斯分类器的假设, 这时朴素贝叶斯分类器能够产生很好的结果, min_sup 应当取得比较大; 相反, 只有当很多属性联合起来才对分类有影响时, 选取较小的 min_sup, 从而能够更多地将这种相关性考虑进来.

3 实验测试

3.1 FISC与其他算法的比较

本文基于weka[17]环境对FISC和NB[18-19], AODE[2], LBR[20], TAN[12], LB[3]进行了比较. NB, AODE, LBR, TAN使用了weka[17]中的实现, 参数为weka中的默认参数, AODE中的*min-sup*根据文献[2]设置为30. LB按照文献[3]中描述的算法实现, 其参数也根据文献[3]进行设置. 由于NB, AODE等几个算法只能处理离散属性, 因而对于连续属性采用MDL[21]的方法事先进行了离散化处理. 同时, 几个算法都能对含有缺失值的数据集进行正常处理, 因此不对含有缺失值的样本进行特殊处理. 本节实验中, FISC的参数*min-sup*始终设置为训练样本数目的15%.

实验使用了21个来自UCI机器学习数据库[22]的数据集, 在每个数据集上都进行10次10倍交叉验证, 报告的实验结果为10次实验的平均结果. 表1中给出了10次10倍交叉验证得到的测试错误率的平均值和标准差. FISC在21个数据集中的4个取得了最低的平均测试错误率. 其中在House数据集上显著优于其他算法. 在其他一些数据集上, FISC的平均测试错误率也比较接近其他算法所达到的最低的平均测试错误率, 如Crx数据集, LB的平均测试错误率为12.70, FISC为12.75.

Table 1 Error Rates of Different Algorithms (in form of error±stdev)

表1 不同算法的分类误差率(表示为均值± 标准差) %

Dataset	NB	TAN	LBR	AODE	LB	FISC
Balance	28.03±1.06	28.83±1.16	28.35±0.77	30.24±0.88	**27.94±1.11**	30.61±0.73
Bcw	2.85±0.13	3.23±0.17	2.85±0.11	2.96±0.17	2.79±0.18	**2.75±0.22**
Bupa	**36.81±0.00**	**36.81±0.00**	**36.81±0.00**	**36.81±0.00**	**36.81±0.00**	**36.81±0.00**
Cleveland	16.37±0.39	17.62±0.45	16.30±0.42	16.90±0.58	17.00±0.42	17.76±0.81
Crx	13.54±0.24	14.54±0.54	13.61±0.29	13.28±0.23	**12.70±0.33**	12.75±0.26
Echocardiogram	23.78±1.82	24.19±1.49	24.32±2.21	22.43±1.82	**21.76±0.43**	23.51±1.71
German	24.78±0.39	25.63±0.49	25.02±0.55	**23.35±0.35**	24.32±0.36	24.36±0.37
Glass	25.93±0.55	**22.90±0.82**	25.84±0.76	24.16±0.80	25.98±1.43	23.93±0.65
Heart	16.44±0.43	17.59±0.53	**16.41±0.46**	16.52±0.74	17.11±0.49	17.48±0.55
Hepatitis	14.39±0.53	14.58±1.46	14.26±0.64	13.87±0.34	**11.16±1.14**	12.26±0.61
Horse	20.19±0.46	17.47±0.57	17.53±0.69	17.23±0.47	17.36±0.81	**16.98±0.47**
House	9.93±0.26	5.40±0.50	5.70±0.28	5.70±0.15	5.93±0.28	**4.39±0.20**
Hungarian	15.75±0.23	15.78±0.99	**15.37±0.22**	15.71±0.35	15.37±0.50	15.68±0.57
Iris	5.60±0.34	6.47±0.71	5.60±0.34	6.73±0.58	**5.47±0.28**	6.13±0.53
Labor	7.37±1.11	**5.44±1.54**	6.32±1.23	7.37±0.74	12.63±1.81	5.96±1.23
mfeat-mor	30.47±0.26	**27.99±0.40**	28.76±0.60	30.23±0.51	30.21±0.26	30.86±0.21
Postoperative	31.67±1.31	36.67±3.70	**31.22±1.33**	31.56±1.67	36.22±1.07	31.56±1.59
Segment	8.25±0.16	4.37±0.22	5.47±0.18	**4.28±0.12**	6.13±0.20	5.32±0.10
Tic-Tac-Toe	30.37±0.34	24.09±0.60	**14.42±0.84**	26.00±0.58	30.08±0.32	16.93±0.54
Vehicle	37.58±0.57	**26.09±0.66**	28.82±0.68	28.09±0.51	28.74±0.43	27.39±0.42
Wine	1.12±0.00	1.57±0.52	1.12±0.00	1.69±0.00	**0.67±0.24**	1.12±0.00

另外, 可以发现在许多数据集上, 几种算法的平均测试错误率非常接近, 仅使用平均测试错误率不足以衡量各个算法的表现, 因而表2给出了FISC算法同其他算法进行显著程度为95%的双尾成对*t*检验的结果, 表2中win代表FISC显著优于对比算法, tie表示两种算法没有显著差别, loss表示FISC显著逊于对比算法.

从表2可以看出, 21个数据集中, FISC在9个数据集上显著优于NB, 在8个数据集显著优于TAN, 在8个数据集上显著优于LBR, 在9个数据集上显著优于AODE, 在7个数据集上显著优于LB. 总的来说, FISC的性能与这些出色的算法相

比,性能相当甚至略优

Table 2 FISC wins/ties/loses Other Algorithms under Pairwise Two-Tailed *t*-tests at 0.05 Significance Level

表 2 FISC 与其他算法 0.05 显著性 *t* 检验比较

Dataset	NB	TAN	LBR	AODE	LB
Balance	Loss	Loss	Loss	Loss	Loss
Bcw	Tie	Win	Tie	Win	Tie
Bupa	Tie	Tie	Tie	Tie	Tie
Cleveland	Loss	Tie	Loss	Loss	Loss
Crx	Win	Win	Win	Win	Tie
Echocardiogram	Tie	Tie	Tie	Loss	Loss
German	Tie	Win	Win	Loss	Tie
Glass	Win	Loss	Win	Tie	Win
Heart	Loss	Tie	Loss	Loss	Tie
Hepatitis	Win	Win	Win	Win	Loss
Horse	Win	Tie	Win	Tie	Tie
House	Win	Win	Win	Win	Win
Hungarian	Tie	Tie	Tie	Tie	Tie
Iris	Loss	Tie	Loss	Win	Loss
Labor	Win	Tie	Tie	Win	Win
mfeat-mor	Loss	Loss	Loss	Loss	Loss
Postoperative	Tie	Win	Tie	Tie	Win
Segment	Win	Loss	Win	Loss	Win
Tic-Tac-Toe	Win	Win	Loss	Win	Win
Vehicle	Win	Loss	Win	Win	Win
Wine	Tie	Win	Tie	Win	Loss
win/tie/loss	9/7/5	8/8/5	8/7/6	9/5/7	7/7/7

3.2 *min-sup* 的影响

FISC 算法中惟一需要设置的参数是最小支持度 *min-sup*. 为了测试 *min-sup* 对 FISC 算法分类错误率的影响,本节比较了在给定数据集上选取不同的参数值得到的分类错误率. 因为各个数据集的样本数和属性数差距都比较大,因此本文采用数据集样本数的百分比

$$t = \frac{min\text{-}sup}{numberOfInstances} \tag{13}$$

作为参数. 在每个数据集上, t 从 0.05 开始,以 0.05 为步长逐渐增大到 0.50,在不同 t 值上的测试错误率.

实验结果表明在 21 个实验数据集上,测试错误率最低的情况所对应的 t 值并不相同. 为了直观地显示分类误差同 t 的关系,图 2 绘制出了在 4 个具有一定代表性的数据集上,FISC 在不同 t 值时的测试错误率. 图 2 显示出不同数据集上 FISC 的最低测试错误率所对应的 t 值不相同,而且测试错误率随 t 值变化的趋势也未必相同. 图 2(a)和(b)从总体走势上可以近似看做单调函数,但是它们一个为递增函数,一个为递减函数;图 2(c)中的曲线不具有单调性,最佳的 t 值出现在中间. 另外,图 2(a)对应的 Tic-Tac-Toe 数据集,当 t 从 0.05 变化到 0.5 时,误差率从 7.89%上升到 30.37%,变化非常显著,而图 2(d)所对应的 Bupa 数据集则对参数 t 不敏感

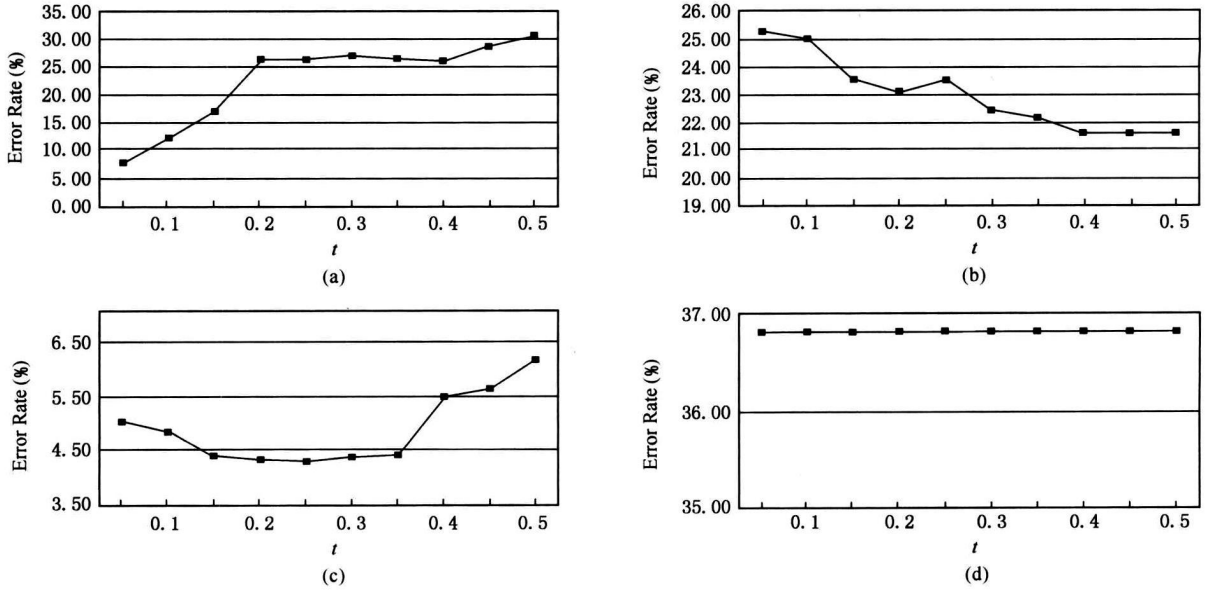

Fig. 2 The error rates of FISC under different t values on four data sets. (a) Tic-Tac-Toe; (b) Echocardiogram; (c) House; and (d) bupa

图 2 4 个数据集上 FISC 在不同 t 参数设置下的分类误差率. (a) Tic-Tac-Toe; (b) Echocardiogram; (c) House; and (d) Bupa

但是, 可以通过少数几次交叉验证得到一个较好的 t 的设置. 实验结果表明, 在 21 个数据集中, 有 9 个数据集当 t 取 5%~10%时取得最低的分类误差率, 另有 9 个数据集当 t 取 40%~50%时取得最低分类误差率, 只有 3 个数据集对应的最优的 t 值为 20%~25%. 即大部分数据集的最好参数设置 t=10%, 25%, 45%附近. 因此, 可以通过交叉验证技术对 FISC 在 t=10%, 25%, 45%等 3 种情况下的性能进行比较, 从而选择出比较合适的 t.

4 结束语

本文提出了一种基于频繁项集挖掘的贝叶斯分类学习算法 FISC, 该算法使用高阶频繁项集来辅助放松朴素贝叶斯分类器的属性独立性假设, 并采用集成学习技术降低模型选择所带来的不利影响. 实验验证了 FISC 的有效性.

FISC 算法的参数 *min-sup* 对算法性能有较大影响. 目前除了交叉验证外, 尚无好的技术来确定合适的 t 值. 为 FISC 设计计算开销小、效果好的 t 值估计技术, 将在今后的工作中做进一步研究.

参 考 文 献

[1] P Domingos, M Pazzani. Beyond independence: Conditions for the optimality of the simple Bayesian classifier[C]. The 13th Int'l Conf on Machine Learning, San Francisco, CA, 1996

[2] G I Webb, J R Boughton, Z J Wang. Not so naive Bayes: Aggregating one-dependence estimators[J]. Machine Learning, 2005, 58(1): 5—24

[3] D Meretakis, B Wuthrich. Extending naïve Bayes classification using long itemsets[C]. The 5th ACM SIGKDD Int'l Conf on Knowledge Discovery and Data Mining, San Diego, CA, 1999

[4] R Agrawal, M Srikant. Fast algorithms for mining association rules in large databases[R]. IBM Almaden Research Center, Tech Rep: RJ9839, 1994

[5] R Agrawal, M Srikant. Fast algorithms for mining association rules[C]. The 20th Int'l Conf on Very Large Data Bases, Santiago, Chile, 1994

[6] H Mannila, H Toivonen, A I Verkamo. Efficient algorithms for discovering association rules[C]. The AAAI'94 Workshop on Knowledge Discovery in Database, Seattle, WA, 1994

[7] P Langley, S Sage. Induction of selective Bayesian classifiers[C]. The 10th Conf on Uncertainty in Artificial Intelligence, Seattle, WA, 1994

[8] M J Pazzani. Constructive induction of Cartesian product attributes[C]. The Conf on Information, Statistics and Induction in Science'96, Singapore, 1996

[9] G I Webb, M J Pazzani. Adjusted probability naive Bayesian induction[C]. The 11th Australian Joint Conf on Artificial Intelligence, Brisbane, Australia, 1998

[10] P Langley. Induction of recursive Bayesian classifiers[C]. The 6th European Conf on Machine Learning, Vienna, Austria, 1993

[11] R Kohavi. Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid[C]. The 2nd ACM SIGKDD Int'l Conf on Knowledge Discovery and Data Mining, Portland, OR, 1996

[12] N Friedman, D Geiger, M Goldszmidt. Bayesian network classifiers[J]. Machine Learning, 1997, 29(2): 131—163

[13] Shi Hongbo, Huang Houkuan, Wang Zhihai. Boosting-based TAN combination classifier[J]. Journal of Computer Research and Development, 2004, 41(2): 340—345 (in Chinese)
(石洪波, 黄厚宽, 王志海. 基于 Boosting 的 TAN 组合分类器[J]. 计算机研究与发展, 2004, 41(2): 340—345)

[14] M Sahami. Learning limited dependence Bayesian classifiers[C]. The 2nd Int'l Conf on Knowledge Discovery and Data Mining, Portland, OR, 1996

[15] P M Lewis. Approximating probability distributions to reduce storage requirements[J]. Information and Control, 1959, 2(3): 214—225

[16] J W Han, J Pei, Y W Yin, *et al*. Mining frequent patterns without candidate generation: A frequent-pattern tree approach[J]. Data Mining and Knowledge Discovery, 2004, 8(1): 53—87

[17] I H Witten, E Frank. Data Mining: Practical Machine Learning Tools with Java Implementations[M]. San Francisco, CA: Morgan Kaufmann, 1999

[18] P Langley, W F Iba, K Thompson. An analysis of Bayesian classifiers[C]. The 10th National Conf on Artificial Intelligence, San Jose, CA, 1992

[19] I Kononenko. Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition[G]. In: B Wielinga, J Boose, B Gaines, eds. Current Trends in Knowledge Acquisition. Amsterdam, Netherlands: IOS Press, 1990. 190—197

[20] Z Zheng, G I Webb. Lazy learning of Bayesian rules[J]. Machine Learning, 2000, 41(1): 53—84

[21] M Fayyad, B Irani. Multi-interval discretization of continuous-valued attributes for classification learning[C]. The 13th Int'l Joint Conf on Artificial Intelligence, Chambéry, France, 1993

[22] C L Blake, C J Merz. UCI repository of machine learning databases[OL]. http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998

Xu Junming, born in 1984. M. S. candidate of Nanjing University. His main research interests include machine learning and data mining.

眭俊明, 1984 年生, 硕士研究生, 主要研究方向为机器学习、数据挖掘等.

Jiang Yuan, born in 1976. Received her Ph. D. degree in computer science from Nanjing University in 2004. She is associate professor at the Department of Computer Science & Technology, Nanjing University. Her main research interests include machine learning, information retrieval and data mining.

姜 远, 1976 年生, 博士, 副教授, 主要研究方向为机器学习、信息检索、数据挖掘等(jiangy@lamda.nju.edu.cn).

Zhou Zhihua, born in 1973. He is professor and Ph. D. supervisor at the Department of Computer Science & Technology, Nanjing University. Senior member of China Computer Federation. His main research interests mainly include artificial intelligence, machine learning, data mining, information retrieval, pattern recognition, evolutionary computation and neural computation.

周志华, 1973 年生, 博士, 教授, 博士生导师, 中国计算机学会高级会员, 主要研究方向为人工智能、机器学习、数据挖掘、信息检索、模式识别、演化计算、神经计算等(zhouzh@lamda.nju.edu.cn).

Research Background

Naïve Bayesian classifier is simple and effective, but suffers from its assumption on attribute independence which is often violated in real-world applications. Many works have been done to relax this assumption and improve the generalization ability of naïve Bayesian classifier. Frequent item set mining is an important field of data mining, in which many algorithms have been developed. This paper proposes a frequent item sets mining-based Bayesian classifier, the FISC (frequent item sets classifier). At the training stage, FISC finds all the frequent item sets by frequent item sets mining techniques and computes all the probabilities that may be used at the classification time. At the test stage, FISC constructs a classifier for each frequent item set contained in the test instance, and then classifies the instance by ensembling all these classifiers. Experiments validate the effectiveness of FISC.

Seventeenth International World Wide Web Conference (WWW2008) April 21~25, 2008

The International World Wide Web Conferences Steering Committee (IW3C2) cordially invites you to participate in the 17th International World Wide Web Conference (WWW2008), to be held on April 21~25, 2008 in Beijing, China. The conference series has become the premier venue for academics and industry to present, demonstrate, and discuss the latest ideas about the Web. The technical program for the five-day conference will include refereed paper presentations, plenary sessions, panels, and poster sessions. The WWW2008 program will also include Tutorials and Workshops, a W3C track, a Developers track, a WWW in China track, and Exhibitions.

IMPORTANT DATES

Submission Deadlines:

Workshop Proposals: October 1, 2007

Refereed Papers: November 1, 2007 (HARD deadline; no extensions will be granted)

Tutorial Proposals: November 1, 2007

Posters: January 25, 2008 (estimated)

Acceptance Notification: Refereed Papers—January 15, 2008 (tentative)

Conference dates: April 21~25, 2008

REFEREED PAPERS

WWW2008 seeks original papers describing research in all areas of the Web. Topics include but are not limited to:

- Browsers and User Interfaces
- Data Mining
- Industrial Practice and Experience
- Internet Monetization
- Mobility
- Performance and Scalability
- Rich Media Search
- Search
- Security and Privacy
- Semantic Web
- Social Networks and Web 2.0
- Technology for Developing Regions
- Web Engineering
- XML and Web Data

General queries regarding WWW2008 submissions can be sent to submissions@www2008.org. Other inquiries about the conference can be sent to info@www2008.org.

The full call for papers, including detailed information about the scope of each track, formatting and submission requirements will be available soon from http://www2008.org.