

基于 L1 正则化的贝叶斯网络分类器

王影 王浩 俞奎 姚宏亮

(合肥工业大学计算机与信息学院 合肥 230009)

摘要 目前基于节点排序的贝叶斯网络分类器忽略了节点序列中已选变量和类标签之间的信息,导致分类器的准确率很难进一步提高。针对这个问题,提出了一种简单高效的贝叶斯网络分类器的学习算法:L1 正则化的贝叶斯网络分类器(L1-BNC)。通过调整 Lasso 方法中的约束值,充分利用回归残差的信息,结合点序列中已选变量和类标签的信息,形成一条优秀的有序变量拓扑序列(L1 正则化路径);基于该序列,利用 K2 算法生成优良的贝叶斯网络分类器。实验表明,L1-BNC 在分类精度上优于已有的贝叶斯网络分类器。L1-BNC 也与 SVM,KNN 和 J48 分类算法进行了比较,在大部分数据集上,L1-BNC 优于这些算法。

关键词 贝叶斯网络分类器(BNC),Lasso 方法,K2 算法,L1 正则化

Bayesian Network Classifier Based on L1 Regularization

WANG Ying WANG Hao YU Kui YAO Hong-liang

(School of Computer & Information, Hefei University of Technology, Hefei 230009, China)

Abstract Variable order-based Bayesian network classifiers ignore the information of the selected variables in their sequence and their class label, which significantly hurts the classification accuracy. To address this problem, we proposed a simple and efficient L1 regularized Bayesian network classifier (L1-BNC). Through adjusting the constraint value of Lasso and fully taking advantage of the regression residuals of the information, L1-BNC takes the information of the sequence of selected variables and the class label into account, and then generates an excellent variable ordering sequence (L1 regularization path) for constructing a good Bayesian network classifier by the K2 algorithm. Experimental results show that L1-BNC outperforms existing state-of-the-art Bayesian network classifiers. In addition, in comparison with SVM, Knn and J48 classification algorithms, L1-BNC is also superior to those algorithms on most datasets.

Keywords Bayesian network classifier, Lasso method, K2 algorithm, L1 regularization

1 引言

近年来,贝叶斯网络分类器(Bayesian Network Classifiers, BNC)作为一种直观、高效、可靠的数据挖掘分类工具已经得到了广泛研究与应用^[1,2]。为了提高分类精度,已经提出两类贝叶斯网络分类器:限制性贝叶斯网络分类器和无限制的贝叶斯网络分类器。限制性贝叶斯网络分类器是将分类节点作为其它属性节点的父节点,而无限制的贝叶斯网络分类器则将分类节点看作普通节点进行处理。

目前存在的限制性贝叶斯网络分类器主要有朴素贝叶斯网络分类器(Naive Bayesian Classifiers, NBC)及其扩展模型(TAN, AODE, HNB)。从网络的拓扑结构来说,朴素贝叶斯分类器假设所有属性都是相互条件独立的,每个属性仅仅依赖类标签变量(Class Label)^[3]。然而,其假设性条件在实际应用中很少能够满足。在大多数实际情况下,该假设的违背会显著地降低预测精确度^[4]。为此, Friedman 等人提出了树扩张型贝叶斯网络分类器(Tree-Augmented Bayesian Classi-

fiers, TAN)以提高分类准确率。此外, Webb 借鉴了集成学习的思想,研究并提出了 AODE 算法^[5], Zhang 提出了 HNB 算法(Hidden Naive Bayes)^[6,7]。总的来说, NBC, TAN, AODE, HNB 这些算法都是将分类节点作为其它属性节点的父节点,对贝叶斯网络都有一定的限制。

为进一步放松对限制性贝叶斯网络分类器的限制条件, Heckerman 等人提出了无任何限制条件的贝叶斯网络分类器(General Bayesian Networks Classifier, GBN)^[8], GBN 把分类节点作为网络结构中的普通结点,属性之间的依赖关系不受限制^[9,10]。最近的研究成果表明,GBN 要优于有限制的贝叶斯网络分类器^[11]。

已提出许多启发式搜索方法,用来学习贝叶斯网络分类器。K2 算法由于简单高效性,是目前构造贝叶斯网络分类器的主要算法^[12,13]。该算法的一个关键问题是需要预先定义一个全局的变量拓扑序列。现存的获得变量拓扑序列的方法主要有 Estevam 等人在特征排序(Feature Ranking)算法的基础上提出了利用 χ^2 (记为 CHD)或互信息(记为 MI)的方法对

到稿日期:2011-02-27 返修日期:2011-04-21 本文受国家自然科学基金(610701131),973 前期预研项目(2009CB326203)和合肥工业大学“千人计划”团队专项基金项目(2010 hgj0717)资助。

王影(1987-),女,硕士生,主要研究方向为人工智能与数据挖掘,E-mail:wangying182@163.com;王浩(1962-),男,博士,教授,硕士生导师,主要研究方向人工智能、数据挖掘、面向对象技术。

节点进行排序^[12,14]。还有一种使用条件互信息来获得节点顺序^[13]的方法。但这些方法通常都存在一个缺点:进行变量排序时,只考虑了单变量的信息,不能对已选入点序列的变量进行动态调整,忽略了已选变量的信息。

在统计学中,Lasso^[15,16]作为一种系数压缩和变量选择的方法,已经成为处理高维数据的新特征选择技术^[17]。Lasso在进行了特征选择时,使用L1正则化技术压缩回归系数,使一些变量的系数精确为0。该方法在进行变量选择的过程中,考虑了序列中已选变量以及类标签变量的信息,即每次选择与回归残差最相关的变量进入已选变量集合,当新变量与残差的相关度高于序列中已选变量与残差的相关度时,已选变量将从变量集合中移除,等待下一次选入。当所有变量都进入序列时,可以得到一个有序的变量拓扑序列,即L1正则化路径^[15]。L1正则化路径的形成充分考虑了已选变量和类标签变量的信息,使学习的拓扑序列更加准确。因此Lasso方法在变量选择过程定义了一个更好的变量拓扑序列。本文中,将Lasso方法与K2算法相融合,通过调整Lasso方法的约束值,提出了L1正则化贝叶斯网络分类器(L1-BNC)。L1-BNC通过引入Lasso方法获得一条包含所有变量的L1正则化路径,即变量拓扑序列。在此序列的基础上,通过K2算法构造贝叶斯分类器。

为了验证L1-BNC的有效性,本文从UCI数据库中获得了18个数据集进行实验^[18]。实验表明,L1-BNC在分类精度上优于已有的贝叶斯分类器。同时,将L1-BNC与SVM,KNN和J48分类算法进行了比较。实验表明,在大部分数据集上,L1-BNC优于这些算法。

2 研究背景

2.1 K2算法

K2算法:

输入: n 个节点的序列,父节点的上限约束值 u ,包含 m 个实例的数据集 D

输出: 每个节点的父节点集

Begin;

$\pi_i := \{\}$;

For $i := 1$ to n do

While ($|\pi_i| < u$)

z 为 x_i 前面的节点,计算评分函数 g ,如果 g 增加,则将 z 加入到 x_i 的父节点集 $\pi_i := \pi_i \cup z$; 否则, π_i 不变;

EndWhile

EndFor

End

图1 K2算法

K2算法作为一种简单高效的贝叶斯网络分类器构造算法,要求在算法开始时定义一个所有变量的拓扑序列^[19]。K2算法使用一个贪婪搜索的方法来构造网络结构,它首先初始化每个节点的父节点为空集,然后从序列中的第二个节点开始测试可能的父节点,计算评分函数(评分函数可以为BIC、MDL或Bayesian评分),将具有最大评分函数的节点加到父节点集中;搜索直至所有节点的父节点搜索结束或达到最大父节点数,算法结束。在这个过程中,如果评分函数为Baye-

sian评分,则计算评分函数时主要是最大化,如式(1):

$$P(B_s, D) = c \prod_{i=1}^n \max_{\pi_j} \left[\prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \right] \quad (1)$$

式中,每个属性 $x_i (i=1, \dots, n)$ 有 r_i 个可能的值 $(v_{i_1}, v_{i_2}, \dots, v_{i_{r_i}})$, D 是一个数据集, B_s 是包含所有属性的网络结构。 π_i 是在网络中节点 x_i 的父节点。 N_{ijk} 是数据集 D 中变量 x_i 取值为 v_{i_k} 、其父节点 π_i 取其第 j 个取值时的实例数, $N_{ij} = \sum N_{ijk}$, c 是每个网络结构 B_s 的先验概率 $P(B_s)$ 。其算法过程如图1所示。

2.2 Lasso方法

R. Tibshirani提出了一种有偏估计的方法,用来处理高维数据的特征选择问题,该方法称为Lasso^[16]。Lasso主要是通过L1惩罚回归来求得式(2)、式(3)的最优解。

$$\arg \min_{\beta} \left\{ \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 \right\} \quad (2)$$

$$\text{subject to } \sum_{j=1}^p |\beta_j| \leq s \quad (3)$$

式中, β_j 为第 j 个变量的回归系数; $s \geq 0$ 是约束值,它是对回归系数 β_j 的L1惩罚,其取值可以从0到无穷大。当 s 的值比较小时,某些相关度低的变量系数就被压缩为0,从而将这些变量删除,以达到特征选择的目的;当 s 取得足够大时,将不再具有约束的作用,此时所有的属性将被选择并且形成一个变量拓扑序列,即L1正则化路径。

目前解决Lasso问题的一种经典算法是最小角回归算法(Least Angle Regression, LARS),该算法快速高效^[15]。LARS寻找L1正则化路径时只需 n 步(n 为变量数),主要由于它是一种残差拟合的过程,每次在回归残差的基础上选择新的变量。该过程是一个残差不断减小的过程^[16]。

回归残差综合了类标签变量与已选变量的信息,通过这种方法可以有效地找到Lasso中方程的最优解,即L1正则化路径。LARS算法描述如图2所示。根据上面的算法,可以得到一个L1正则化路径。当约束值 s 足够大时,路径中将包含所有的变量,这时得到一个排好序的变量序列。图3为LARS对数据集diabetes的L1正则化路径分析^[15]。

LARS算法过程:

- (1) 算法开始时,先把所有的系数 β_j 设置为0。
- (2) 找到与响应变量 y 最相关的一个变量 x_j 。
- (3) 把 β_j 朝着和 y 最相关的方向增大,然后沿着这个方向计算残差 $r = y - \hat{y}$,直到另一个使得 r 和 x_j 最相关的变量 x_k 被选中为止。
- (4) 朝着 (β_j, β_k) 联合的最小均方方向增大,该方向与 (β_j, β_k) 的残差有相等的相关系数,直到另外一个变量 x_m 出现以满足 r 和 x_m 最相关。
- (5) 直到所有的变量被选择,算法结束。

图2 LARS算法

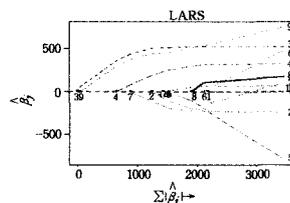


图3 LARS对数据集diabetes的L1正则化路径分析

从图3中可以看到: s 的值减小时,可以从路径中移除一

些不重要的变量,从而达到特征选择的目的。最重要的是,随着 s 值的不断增大,如 $s \geq 3000$ 时,可以得到一个包含所有变量的 L1 正则化路径 $\{3, 9, 4, 7, 2, 10, 5, 8, 6, 1\}$ 。

3 L1 正则化的贝叶斯网络分类器(L1-BNC)

Lasso 方法中的 L1 正则化技术近年来一直是人们研究的热点^[20]。K2 算法作为一种贝叶斯网络分类器的构造方法,一直被广泛应用^[12,13]。其学习网络的质量最主要的一个因素取决于变量的拓扑序列:拓扑序列越准确,学习的网络质量越好。因此,能够准确学习变量拓扑序列成为整个算法的关键。L1-BNC 正是将这两种技术融合在一起进行贝叶斯网络分类器的学习。

LARS 算法作为解决 Lasso 问题的一种方法,在学习变量序列时利用回归的方法进行变量选择,每次选择一个因变量进入激活集(active set),该变量在迭代过程中与回归残差的相关度最高。当变量被选入激活集时,首先会与其它已被选入变量进行比较。如果该变量与残差的相关度大于已选变量的相关度,LARS 会对变量序列进行动态调整,然后根据约束值大小决定是否将已选入变量从集合中移除^[15,16],从而保证激活集中变量序列与分类变量的相关度是递减排列的。激活集中的变量序列在统计学中称为 L1 正则化路径。

根据上述 LARS 算法的思想,将 LARS 算法获得 L1 正则化路径的方法转变为获得变量拓扑序列的一种途径,将该方法应用于贝叶斯网络分类器的学习。在 L1 正则化路径中,变量位置越靠前,其与类标签变量的相关度越高,整条路径的变量与类标签变量的相关性按递减方式排列。由于该路径是应用于分类领域,因此类标签变量很重要。在得到 L1 正则化路径之后,将分类变量放在路径的最前面作为所有变量的父节点,从而得到一个包含类标签变量的完整变量拓扑序列,进行贝叶斯分类器的学习。图 4 为 L1-BNC 算法的具体描述。

L1-BNC 算法:

输入:数据集 train_data, test_data;

输出:贝叶斯网络分类准确率 rate

Begin:

第一阶段:将数据集进行预处理后,调用 LARS 算法;在这个过程中,需要调整上述方程(3)的约束值,直到所有变量都可以被选入变量拓扑序列,即 LARS(train_data);通过调整 LARS 算法的约束因子,获得一条包含所有变量的变量拓扑序列(L1 正则化路径)order。

第二阶段:获得变量拓扑序列 order 后,结合 K2 算法,进行贝叶斯分类器的学习

For 每个变量拓扑序列 order DO

lars_k2_classification (order, train_data, test_data);

或者 lars_k2_classification (order, train_data, k);

返回分类器的准确度;

EndFor;

train_data, test_data 分别为训练和测试数据集;算法使用 k 折交叉验证方法,计算每次分类的准确率 ratio。

其中 k 为交叉验证的折数,本文中 k 设置为 10。

End

图 4 L1-BNC 算法流程

4 实验结果

4.1 实验数据

为了测试 L1-BNC 算法的有效性,本文从 UCI 数据库中选取了 18 个数据集进行实验,数据集的详细描述如表 1 所列^[18]。实验中,L1-BNC 在上述 18 个数据集上与 9 个数据挖掘常用分类算法进行比较,这 9 个常用分类算法是 NB, TAN, AODE, HNB, K2CHI, K2MI, K 近邻, J48, SVM。为了准确测试 L1-BNC 算法与其它算法的分类效果,设计实验时将上述分类算法分成两组。第一组为 L1-BNC 与 NB, TAN, AODE, HNB, K2CHI, K2MI 在分类准确度上的比较,其中 NB, TAN, AODE, HNB 为有限制贝叶斯分类算法。K2CHI, K2MI 为无限制(general)贝叶斯分类算法,并且这两种算法都是基于节点次序的分类算法,L1-BNC 与其进行比较更具针对性。第二组为 L1-BNC 与其它分类器 KNN, J48, SVM 在分类精度上的比较^[21]。

表 1 数据集描述

No	Train set	Sample	Features	Test set
1	spect	80	23	test set
2	iris	150	5	CV-10
3	wdbc	569	31	CV-10
4	ionosphere	200	35	test set
5	spectf	200	35	CV-10
6	diabetes	768	9	CV-10
7	machine	209	8	CV-10
8	promoters	106	58	CV-10
9	segment	2310	20	CV-10
10	glass	214	10	CV-10
11	iono	351	35	CV-10
12	breast	3146	10	CV-10
13	tictac	4312	10	CV-10
14	wine	178	14	CV-10
15	cleve	2727	12	CV-10
16	austra	690	15	CV-10
17	lymph	1332	19	CV-10
18	vehicle	3809	19	CV-10

实验数据集 spect 和数据集 ionosphere 采用的是 UCI 测试集,其它数据集全部采用十折交叉验证(CV-10)方法测试分类准确率。此外,实验所用的贝叶斯分类器工具包是 BNT-SLP 软件包^[22]。AODE, HNB, KNN, J48, SVM 5 种分类算法全部来自 WEKA 软件包^[23]。对于实验中的连续属性,调用 Causal Explore 包中的离散数据函数进行离散化处理^[24]。AODE 和 HNB 实验时,采用的 NumericToNominal 过滤器将 Numeric 属性转换成 Nominal 属性。

4.2 L1-BNC 与贝叶斯分类算法的比较

4.2.1 L1-BNC 与有限制贝叶斯分类算法的比较

本节首先把 L1-BNC 算法与 NB, TAN, AODE, HNB 4 个有限制贝叶斯分类算法在 18 个数据集上进行比较,其中 AODE, HNB 是最近提出的两种对朴素贝叶斯分类算法的扩展^[5,6]。图 5 给出了实验结果(实验中所有结果图采用 $x=y$ 对角线表示法,即落在对角线下面的点表示 L1-BNC 算法优于其它算法)。

图 5 中纵坐标表示 L1-BNC 的分类准确度,横坐标分别表示其它 4 种贝叶斯分类器的分类准确度。如果点落在对角线之上,则说明 L1-BNC 的分类器精度比较好;反之,则说明其它分类器的分类精度比较好。如果越靠近对角线,则说明两类算法的分类精度相当。从图中可以观察到 L1-BNC 优于上述 4 种朴素贝叶斯分类器算法,其主要原因在于这 4 种算

法都是有限制的贝叶斯分类算法,在分类时,有一部分数据信息将会丢失,而我们提出的 L1-BNC 是一种对网络结构无限制的分类算法,因此其准确度高于上述 4 种分类算法。

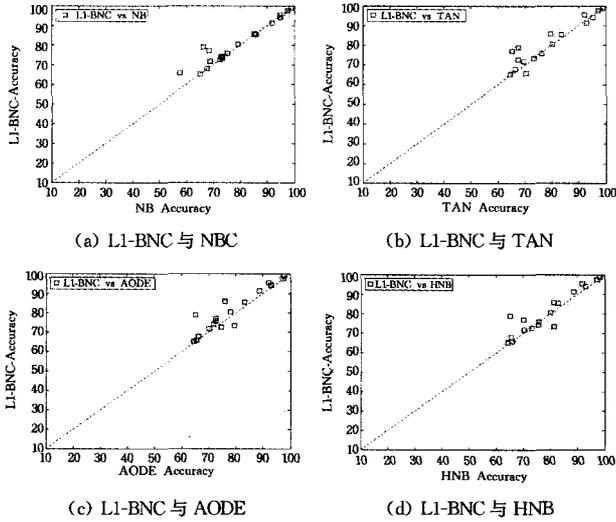


图 5 L1-BNC 与 NB, TAN, AODE, HNB 分类算法准确度比较

4.2.2 L1-BNC 与无限制(general)贝叶斯分类算法的比较

本节将 L1-BNC 与两种最新研究的无限制贝叶斯分类器(即 K2CHI 和 K2MI^[12,13])进行比较。这两种算法分别通过比较节点与类标签之间的 CHI 与 MI(互信息)来获得节点序列,其实验结果如图 6 所示。

在图 6(a)、图 6(b)中分别比较了 L1-BNC 与 K2CHI 和 K2MI 的分类准确度。可以看出大部分数据集的点落在了对角线上或对角线上面,少部分数据集点落在对角线下面,说明 L1-BNC 的分类准确度优于 K2CHI 和 K2MI。但是这种优势不如图 5(a),图 5(b)的优势明显,说明 K2CHI, K2MI 的分类准确度高于 NB 和 TAN。这个结果与 R. Estevam 等人和 Franz Pernkopf 等人的实验结果(K2CHI, K2MI 优于 NB 和 TAN)一致^[12,13]。

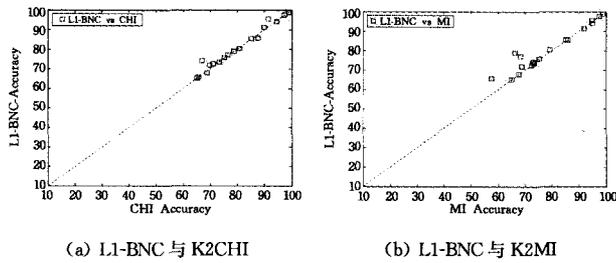


图 6 L1-BNC 与 K2CHI, K2MI 分类准确度比较

4.3 L1-BNC 与其它分类算法的比较

前面实验比较了 L1-BNC 与贝叶斯分类算法的分类准确度。从图 5、图 6 可以看出 L1-BNC 算法优于贝叶斯分类算法,这促使我们进一步来验证 L1-BNC 算法的分类准确度。本部分实验选取了 3 种数据挖掘中分类效果较好且常用的分类算法(KNN, J48, SVM)在上述数据集上测试分类准确度^[21]。最后将测试结果与 L1-BNC 分类结果进行比较,其中对于 KNN 我们选取了 K=1 和 K=3 时的分类准确度进行比较。实验结果如图 7 所示。

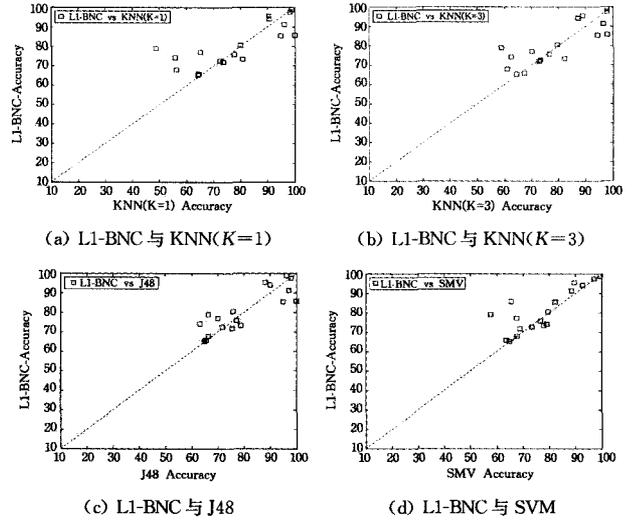


图 7 L1-BNC 与 KNN, J48 和 SVM 分类准确度比较

从图 7 可以看出大部分数据集点落在了对角线之上,少部分数据集点落在对角线附近。尤其是 L1-BNC 与 SVM 分类器相比时,很少有数据落在对角线下面。这说明 L1-BNC 的分类精度与上述算法的分类精度相比仍然处于优势。

为了更好地理解不同分类算法在所有数据集的获胜情况,将上述所有算法的比较结果以表格的形式列出,如表 2 所列。在表中使用 win/tic/loss 的形式来表示算法之间在分类精度上的比较结果。Win 表示获胜, tie 表示相等, loss 表示失败。表中每个单元 $w/t/l$ 表示 L1-BNC 与其它分类算法相比在 w 个数据集上获胜,在 t 个数据集上与其它算法不分胜负,在 l 个数据集上输于其它算法。表中粗体表示 L1-BNC 与其它 9 种算法的比较结果。从表 2 可以清楚地看出, L1-BNC 与上述 9 个分类算法相比,分别在 17, 11, 16, 13, 8, 10, 10, 9, 13, 11 个数据集上获胜,这说明 L1-BNC 算法在大部分数据集上优于其它分类算法。

表 2 不同分类算法在所有数据集上的获胜情况

	NB	TAN	AODE	HNB	CHI	MI	1NN	3NN	SVM	J48
TAN	13/1/4									
AODE	14/0/4	6/0/12								
HNB	13/0/5	6/0/12	8/3/7							
CHI	16/1/1	11/2/5	13/0/5	11/0/7						
MI	15/1/2	8/2/8	13/0/5	10/0/8	7/8/3					
KNN(K=1)	12/0/6	10/0/8	9/0/9	7/1/10	9/0/9	10/0/8				
KNN(K=3)	12/0/6	10/0/8	10/1/7	9/2/7	9/0/9	11/0/7	9/0/9			
J48	15/0/3	11/1/6	9/1/8	9/1/8	8/0/10	9/1/8	8/1/9	10/1/7		
SVM	13/0/5	8/0/10	6/0/12	7/0/11	5/0/13	8/0/10	8/0/10	8/1/9	7/1/10	
L1-BNC	17/1/0	11/2/5	16/0/2	13/0/5	8/5/5	10/5/3	10/0/8	9/0/9	13/0/5	11/0/7

结束语 本文提出了一种新的贝叶斯分类器构造算法 L1-BNC。它融合了 LARS 算法和 K2 算法。选用了 Lasso 的

一种经典解决方法 LARS 来获得变量的拓扑序列。通过使用 K2 算法,将获得的序列应用于构造贝叶斯网络分类器。

为了验证算法的有效性,从 UCI 数据库获得了 18 个数据集进行实验,将 L1-BNC 与数据挖掘中 9 种常用分类算法在分类准确度上进行了比较。实验中将选取的 9 种分类算法分成两组进行比较,即贝叶斯分类算法和其它分类算法。实验表明,L1-BNC 在分类精度上优于已有的贝叶斯分类器分类算法。为了更进一步验证 L1-BNC 的有效性,将 L1-BNC 与其它分类算法(如 SVM,KNN,J48)进行了比较。实验表明,在大部分数据集上,L1-BNC 优于这些算法。从实验结果可以看出,L1-BNC 是一种简单有效的分类算法。

参 考 文 献

[1] Cheng J, Greiner R, Kelly J, et al. Learning Bayesian networks from data: An information-theory based approach[J]. *Artificial Intelligence*, 2002, 137: 43-90

[2] Kojima K, Perrier E, Imoto S, Satoru Miyano Optimal Search on Clustered Structural Constraint for Learning Bayesian Network Structure[J]. *Journal of Machine Learning Research*, 2010, 11: 285-310

[3] Friedman N, Geiger D, Goldszmidt M. Bayesian Network Classifiers[J]. *Machine Learning*, 1997, 29(2/3): 131-163

[4] Domingos P, Pazzani M. Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier[C]// *Proceedings of the Thirteenth International Conference on Machine Learning*. San Francisco, CA: Morgan Kaufmann Publishers, Inc., 1996: 105-112

[5] Webb J I. Not so Naïve: Aggregating one Dependence Estimator [J]. *Machine Learning*, 2005, 58: 5-24

[6] Zhang H, Jiang L, Su J. Hidden Naive Bayes[C]// *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05)*. AAAI Press, 2005: 919-924

[7] Zhang H, Jiang L, Su J. Augmenting Naive Bayes for Ranking [C]// *Proceedings of the 22nd International Conference on Machine Learning (ICML)*. ACM Press, 2005: 1025-1032

[8] Geiger D, Heckerman D. Knowledge representation and inference in similarity networks and Bayesian multinets[J]. *Artificial Intelligence*, 1996, 82: 45-74

[9] Cheng J, Greiner R. Comparing Bayesian network classifiers[C]// *UAI-99*. 1999

[10] Cheng Jie, Greiner R. Learning Bayesian Belief Network Classi-

fiers, Algorithms and System[C]// *AI 2001, LNAI 2056*. 2001: 141-151

[11] Madden M G. On the classification performance of TAN and general Bayesian networks [J]. *Knowledge-Based Systems*, 2009, 22: 489-495

[12] Hruschka E R J, Ebecken N F F. Towards efficient variable ordering for Bayesian networks classifier[J]. *Data & Knowledge Engineering*, 2007: 258-269

[13] Pernkopf F, Bilmes J. Order-based Discriminative Structure Learning for Bayesian Network Classifiers[C]// *ISAIM*. 2008

[14] Liu Feng, Tian Feng-zhan, Zhu Qi-liang. A Novel Ordering-based Greedy Bayesian Network Learning Algorithm on Limited Data[C]// *AI 2007, LNAI 4830*. 2007: 80-89

[15] Efron B, Hastie T, Johnstone I, et al. Least Angle Regression [J]. *Annals of Statistics (with discussion)*, 2004, 32(2): 407-499

[16] Tibshirani R. Regression shrinkage and selection via the lasso [J]. *J. Royal. Statist. Soc B*, 1996, 58(1): 267-288

[17] Vidaurre D, Bielza C, Larrañaga P. Learning an L1-regularized Gaussian Bayesian Network in the Equivalence Class Space[J]. *IEEE Transactions of Systems*, 2010: 1083-4419

[18] Asuncion A, Newman D J. UCI Machine Learning Repository [EB/OL]. <http://www.ics.uci.edu/~mlern/MLRepository.html>, 2007

[19] Cooper G, Herskovitz E. A Bayesian method for the induction of probabilistic networks from data[J]. *Machine Learning*, 1992, 9: 309-347

[20] idaurre D, Bielza C, Larrañaga P. Learning an L1-regularized Gaussian Bayesian Network in the Equivalence Class Space[J]. *IEEE Transactions on Systems*, 2010: 1231-1242

[21] Wu X, Kumar V, Ghosh R J, et al. Top 10 algorithms in data mining[J]. *Knowledge and Information Systems*, 2008, 14(1): 1-37

[22] BNT-SLP. BNT Structure Learning Package[EB/OL]. <http://bnt.insa-rouen.fr/SL.html>, 2010

[23] Weka Wiki. <http://weka.sourceforge.net/wiki/>. 40

[24] Aliferis C F, Tsamardinos I, Stanikov A R, et al. Causal Explorer: A Causal Probabilistic Network Learning Toolkit for Biomedical Discovery[C]// *Proceedings of the International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences*. 2003: 371-376

(上接第 174 页)

[20] Segata N, Blanzieri E. Fast and Scalable Local Kernel Machines [J]. *Journal of Machine Learning Research*, 2010, 11 (2010): 1883-1926

[21] Segata N. Local Approaches for Fast, Scalable and Accurate Learning with Kernels[D]. Trento, University of Trento, 2009

[22] Chen L. New analysis of the sphere covering problems and optimal polytope approximation of convex bodies[J]. *Journal of Approximation Theory*, 2005, 133(1): 134-145

[23] Yang Xu-bing, Chen Song-can, Chen Bin, et al. Proximal support vector machine using local information [J]. *Neurocomputing*, 2009, 73(1-3): 1227-1234

[24] Ye Qiaolin, Zhao Chunxia, Ye Ning, et al. Localized twin SVM via convex minimization[J]. *Neurocomputing*, 2011, 74(4): 580-587

[25] Tian Sheng-feng, Mu Shao-min, Yin Chuan-huan. Cooperative clustering for training SVMs[C]// *Proc. of 3rd International Symposium on Neural Networks*. LNCS 3971, Chengdu, 2006: 962-967

[26] MacQueen J B. Some Methods for classification and Analysis of Multivariate Observations[C]// *Proc. of the 5th Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, University of California Press, 1967: 281-297

[27] 牟少敏. 核方法的研究及其应用[D]. 北京: 北京交通大学, 2008