第16卷 第8期
2006年8月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol.16 No.8
Aug. 2006

竞争选择分裂属性的决策树分类模型

房 立,黄泽宇
(北京交通大学 计算机与信息技术学院,北京 100044)

摘 要:构建决策树分类器关键是选择分裂属性。通过分析信息增益和增益比率、Gini索引、基于Goodman-Kruskal关联索引这三种选择分裂属性的标准,提出了一种改进经典决策树分类器C4.5算法的方法(竞争选择分裂属性的决策树分类模型),它综合三种选择分裂属性的标准,通过竞争机制选择最佳分裂属性。实验结果表明它在大多数情况下,使得不牺牲分类精确度而获得更小的决策树成为了可能。

关键词:决策树;信息增益;增益比率;Gini索引;Goodman-Kruskal关联索引
中图分类号:TP311.13 **文献标识码**:A **文章编号**:1673-629X(2006)08-0106-04

A Decision-Tree Classifier Model of Competition in Choosing Split Attribute

FANG Li, HUANG Ze-yu
(Department of Computer Science and Information Technology, Jiaotong University, Beijing 100044, China)

Abstract: The construction of decision-tree is centered on the selection algorithm of an attribute that generates a partition of the subsets of the training database that is located in the node about to be split. On the basis of analyzing three techniques for choosing the splitting attributes including the entropy gain and the gain ratio, the gini index and Goodman-Kruskal association index, propose a strategy to improve on classical decision-tree classifier C4.5 arithmetic(a decision-tree classifier model of competition in choosing split attribute). Experimental results show it is possible, in most cases, to obtain smaller decision trees without sacrificing accuracy.

Key words: decision-tree; entropy gain; gain ratio; gini index; Goodman-Kruskal association index

0 前 言

分类在数据挖掘中是一项非常重要的任务。分类的目的是学会一个分类函数或分类模型(也称分类器),该模型能把数据库中的数据项映射到给定类别中的某一个,以根据历史数据记录对未来数据进行预测[1,2]。

分类器的构造方法有统计方法、机器学习方法、神经网络方法等等。其中机器学习方法包括决策树法和规则归纳法。这些方法各自在不同的领域中起到了很重要的作用,如:在经济业和安全交易领域中可以建立不同的模型:预测债券价格的变化;决定交易的最佳时刻。航空公司可根据历史资料寻找乘客的旅行模式,改进航线的设置。

构建决策树分类器最重要的是选择分裂属性。文中仔细地分析了信息增益和增益比率、Gini索引、基于Goodman-Kruskal关联索引这三种选择分裂属性的标准,并根据经典决策树分类器C4.5算法提出了一种改进方法。这种方法综合了上述三种选择分裂属性的标准,通过竞争机制选择最佳分裂属性,并给出了具体算法,比较了它和C4.5算法的分类精确度、叶子节点数目及树的尺寸大小。

收稿日期:2005-11-30

作者简介:房 立(1980-),女,天津人,硕士研究生,主要研究领域为机器学习、数据挖掘。

1 决策树分类模型

决策树分类可描述为:输入数据即训练集是由一条条记录组成的。每条记录由若干条属性及一个特定的类标签组成,如$(a_1,\cdots,a_n,c)$,其中$a_i(i=1,\cdots,n)$表示属性,c表示类标。给定训练集$D=\{x_1,x_2,\cdots,x_n\}$,目标是确定一个映射函数$f:(A_1,A_2,\cdots,A_n)\rightarrow C$,使得对任意的未知类别的实例$x_i=(a_1,a_2,\cdots,a_n)$可标以适当的类标$C*$。

决策树方法利用信息增益寻找示例数据库中具有最大信息量的属性字段,建立决策树的一个节点,再根据该属性字段的不同取值建立树的分支;在每个分支子集中重复上述过程,直至节点中所有记录的类别都相同,再通过剪枝生成最终的决策树。

最典型的决策树学习算法ID3的算法核心是在决策树中各级节点上选择属性,用信息熵的增益率作为属性选择标准,使得在每一非叶子节点进行测试时,能获得关于被测试例子最大的类别信息,保证非叶子节点到达各后代叶节点平均路径最短,分类速度较快。

另一种是C4.5算法,是ID3算法的扩展[3,4],采用增

益比率的标准来选择分类属性。ID3 算法只能处理枚举型属性,而 C4.5 算法可将分类领域扩展到连续值属性,采用"两区间离散化"方法,先对属性值排序,根据信息增益度量算出阈值,并将属性值归类到两个区间,进行两路分裂[4,5]。采用决策树生成的分类器,可解释性好[6,7],既可以处理枚举型属性又可以处理数值型属性。但是决策树也存在一些缺陷,如:修剪策略复杂而且费时;而数值型属性则较难处理。

2　C4.5 算法

C4.5 算法构造决策树的核心思想是贪心算法,采用自上而下递归的各个击破方式构造决策树。开始时,所有属性都在根节点,这些属性若是连续的,将其离散化,把记录用所选属性递归地进行分裂,然后将该过程递归到每个子树,当每个节点上的数据都属于同一类别或没有属性可以用来分裂时停止,最后对树剪枝以处理过分适应(Over-fitting)的问题。

C4.5 分类器使用最大信息增益来选择连续型属性的阈值。设 S 是 s 个数据样本的集合,假定类标属性定义了 m 个不同类 $C_i(i = 1,2,\cdots,m)$。设 S_i 是类 C_i 中的样本数。对一个给定的样本分类所需的期望信息如公式(1):

$$I(S_1,S_2,\cdots,S_m) = -\sum_{i=1}^{m} P_i\log_2(P_i) \tag{1}$$

P_i 是任意样本,属于 C_i 概率,并用 S_i/S 估计。

设连续值属性 A 的阈值为 a, S 被 a 划分为 2 个子集:$S_1 = \{$样本 $\mid A$ 的取值 $< a\}$, $S_2 = \{$样本 $\mid A$ 的取值 $\geqslant a\}$。设 S_{ij} 是子集 S_j 中类 C_i 的样本数。由 a 划分成子集的熵或期望信息如公式(2):

$$E(A) = \sum_{j=1}^{2} \frac{S_{1j} + \cdots + S_{mj}}{S} I(S_{1j},\cdots,S_{mj}) \tag{2}$$

对于给定的子集 $S_j(j = 1, 2)$

$$I(S_{1j},\cdots,S_{mj}) = \sum_{i=1}^{m} p_{ij}\log_2(p_{ij}) \tag{3}$$

$P_{ij} = S_{ij}/S_j$ 是 S_j 中的样本属于类 C_i 的概率。则属性在 a 上的信息增益为:

$$\text{gain}(A_a) = I(S_1,\cdots,S_m) - E(A) \tag{4}$$

在现实数据集合里,经常出现属性值缺损。在 C4.5 算法中,当属性值有缺损时,按照分裂分支下的训练实例数的比率将有缺损值的实例分成一定比率的碎片[4,5]。

实际中的决策树可能很复杂,这会降低树的可理解性和可用性,使决策树对历史数据的依赖性增大,出现过度适应,因此必须通过比较子树的估计误差和修剪后的估计误差来决定是否修剪。在构造树的过程中,需要对数据集进行多次的顺序扫描和排序,因而导致算法的低效。

3　Gini 索引

使用 Gini 索引作为节点不纯度的度量。

数据训练集合 T 包含来自 N 个类的实例,Gini 指标定义为:

$$\text{Gini}(T) = 1 - \sum_{j=1}^{n} p_j^2 \tag{5}$$

p_j 是类 j 出现的频率。

如果一个划分将数据训练集合 T 分成两个子集 S_1 和 S_2,则分割后的 Ginisplit 是

$$\text{Gini}_{\text{split}}(T) = \frac{N_1}{N}\text{Gini}(S_1) + \frac{N_2}{N}\text{Gini}(S_2) \tag{6}$$

提供最小 Ginisplit 就被选择作为分割的标准。

4　基于 Goodman－Kruskal 关联索引

这是一种在有限集合的划分集合上进行数字度量的方法,它在建立决策树时定义了一种关于划分的度量。这种度量方法生成了一个源于 Goodman－Kruskal 关联索引[3]的系数 GK,它表明了这种度量能够成功应用于决策树的建立。

令 $\pi = \{B_1,\cdots,B_k\}$ 和 $\sigma = \{C_1,\cdots,C_l\}$ 是集合 S 的两个划分。π 和 σ 的 Goodman－Kruskal 系数为:

$$GK(\pi,\sigma) = 1 - \frac{1}{\mid S \mid}\sum_{i=1}^{k}\max_{1\leqslant j\leqslant i} \mid B_i \leqslant C_j \mid \tag{7}$$

在一个有限集合的划分集合上构造一个度量 d_{GK}。设 $T = (T,H,\rho)$ 是用于建立决策树 J 的数据集,令 ν 是 J 一个将被分割的节点,ρ_ν 是相对于 ν 的实例集合,将在数据集 ρ_ν 上的目标划分记为 θ_{ρ_ν}。一个属于属性集合 H 的属性 A_i 决定了关系 ρ 上的一个划分 π^A。

$$d_{GK}(\pi_R^A,\theta_R) = GK(\pi_{\rho_\nu}^{A_i},\theta_{\rho_\nu}) + GK(\theta_{\rho_\nu},\pi_{\rho_\nu}^{A_i}) \tag{8}$$

试验结果表明:用基于 Goodman－Kruskal 系数的度量来取代用于 C4.5 的普通分割标准,在大多数情况下,使得不牺牲精确度而获得更小的决策树成为了可能。基于 d_{GK} 的最小值,不能生成精确度好的决策树,但能够成功地用于选择分割属性,它所建立的决策树更小,具有可比较的精确度。

5　三重分裂属性选择标准

建树过程中选择分裂属性时,采用信息增益和增益比率[1,4]的标准,选择信息增益值最大的属性作为最佳分裂属性。这种方法不受属性独立性假设的约束。

用基于 Goodman－Kruskal 系数的度量标准选择分裂属性,这种方法所建立的决策树更小,叶子节点更少,且还具有可比较的精确度。

使用 Gini 索引也是一种选择分裂属性的标准。SLIQ 算法和 SPRINT 算法就是采用这种选择分裂属性标准。

经过仔细分析,综合这三种选择分裂属性的标准的特点,通过竞争机制,采用投票的方式多数胜少数的方法选择最佳分裂属性。

6　C4.5 算法的改进方法

文中提出的 Semi－LDtree 算法类似于 C4.5 算法,如

表 1 所示,采用分而治之自顶向下迭代递归的策略。对于连续值属性的阈值的产生,仍沿用 J48 提出的方法。先将数据集合按照连续值属性排序选择多个分裂点。对每一个分裂点计算信息增益,选择信息增益最大的分裂点作为阈值点。

文中采用三重分裂属性选择标准,用投票的方式多数胜少数的方法选取最佳的分裂属性,这样使得可以不牺牲分类精确度而获得更小的决策树成为了可能。

表 1 C4.5 算法的改进方法

学习算法:

输入:带有类标的训练数据集合;

输出:决策树;

1. 计算当前节点每个候选属性的信息增益和增益比率值,选择值最大的属性作为竞争分裂属性 gainAttr;

2. 计算当前节点每个候选属性的 Gini 索引值 gini,选择值最大的属性作为竞争分裂属性 giniAttr;

3. 计算当前节点每个候选属性的基于 Goodman - Kruskal 关联索引系数,选择值最大的属性作为竞争分裂属性 dgkAttr;

4. 如果 gainAttr = giniAttr = dgkAttr,则最佳分裂属性 = gainAttr;

5. 否则 gainAttr! = giniAttr = dgkAttr,则最佳分裂属性 = giniAttr;

6. 否则 giniAttr! = gainAttr = dgkAttr,则最佳分裂属性 = gainAttr;

7. 否则 dgkAttr! = gainAttr = giniAttr,则最佳分裂属性 = gainAttr;

8. 否则 dgkAttr! = gainAttr! = giniAttr,则最佳分裂属性 = gainAttr;

按最佳分裂属性分裂当前节点,生成子节点,在每个子节点上递归建树。

6.1 实验数据

为了将改进的 C4.5 算法和 C4.5 算法进行比较,2 个算法采用的实验数据选自 UCI[8]资源库[9]。

实验数据描述如表 2 所示,列出了每个数据集合的实例个数、类值个数、属性个数。评估分类器的性能采用十重交叉验证的方法。

表 2 实验数据描述

序	数据集合	实例数目	类值数目	属性数目
1	Audio	226	24	69
2	Zoo	101	7	16
3	Solarflare	1389	2	9
4	Lung-Cancer	32	3	56
5	Led	1000	10	24
6	House-Votes-84	435	2	16
7	Anneal	898	6	38
8	Balance-Scale	625	3	4
9	Echocardiogram	131	2	6
10	Glass	214	7	9
11	Pid	768	2	8
12	Wine	178	3	13
13	Vehicle	846	4	18
14	Syncon	600	6	60
15	Sonar	208	2	60
16	Chess	551	2	39
17	Tic-Tac-Toe	958	2	9
18	Iris	150	3	4
19	German	1000	2	20
20	Mushroom	8124	2	22
21	Solarflare-M	1389	6	10
22	Cleveland	302	2	13

6.2 实验结果分析

实验的主要目的是对改进的 C4.5 算法和 C4.5 算法在 75 个数据集合上比较分类精确度。每个分类器的分类精确度是在测试集合上成功预测的实例占总实例的百分比,采用 10 重交叉验证估计分类器的精确度。文中采用的 J48 是在 weka 实验平台下 C4.5 算法的具体实现程序。newc45 是在 weka 实验平台下改进的 C4.5 算法的实现程序。在运行 J48 和 newc45 的时候,均采用其默认参数。

表 3 2 种分类器的实验结果比较(精确度差<0)

数据集合	j48	叶子数目	树节点数目	CSA-tree	叶子数目	树节点数目	精确度差
anneal	90.980000%	53	78	86.637000%	53	78	-4.343000%
lung-cancer	50.000000%	19	25	46.875000%	19	25	-3.125000%
wine	93.820200%	5	9	91.011200%	10	19	-2.809000%
echo74	72.973000%	5	9	70.270300%	5	9	-2.702700%
hepatitis	86.250000%	8	15	83.750000%	7	13	-2.500000%
new-thyroid	92.093000%	9	17	89.767400%	9	17	-2.325600%
syncon	91.666700%	18	35	89.500000%	18	35	-2.166700%
ly	76.351400%	21	34	74.324300%	21	34	-2.027100%
ionosphere	91.453000%	18	35	90.598300%	15	29	-0.854700%
cmc	52.138500%	157	263	51.391700%	173	287	-0.746800%
dmplexer	72.900000%	151	301	72.200000%	151	301	-0.700000%
cleveland	76.567700%	34	57	75.907600%	32	54	-0.660100%
ptn	42.182900%	47	88	41.592900%	47	88	-0.590000%
pt	43.362800%	46	88	42.772900%	46	88	-0.589900%
glass3	75.700900%	18	35	75.233600%	18	35	-0.467300%
glass7	65.887900%	30	59	65.420600%	29	57	-0.467300%
ae	88.344500%	467	933	87.892800%	496	991	-0.451700%
bands	78.478700%	44	75	78.107600%	44	75	-0.371100%
heart	80.000000%	25	43	79.629600%	25	43	-0.370400%
segment	96.753200%	39	77	96.450200%	38	75	-0.303000%
sbl	91.508100%	61	93	91.215200%	58	89	-0.292900%
soybean	91.508100%	61	93	91.215200%	58	89	-0.292900%
letter-recog	87.885000%	1226	2451	87.700000%	1214	2427	-0.185000%
chess	92.196000%	22	43	92.014500%	22	43	-0.181500%
splice-c4.5	94.051000%	136	181	93.893600%	130	173	-0.157400%
pendigits	96.561100%	195	389	96.433800%	194	387	-0.127300%
adult	86.093900%	696	911	86.079600%	688	915	-0.014300%
平均值	79.915096%	133.74	238.40	78.81 552%	134.07	239.85	-1.104544%

表 4 2 种分类器的实验结果比较(精确度差=0)

数据集合	j48	叶子数目	树节点数目	CSA-tree	叶子数目	树节点数目	精确度差
bcw	94.420600%	28	31	94.420600%	28	31	0.000000%
bcwo	94.563700%	14	27	94.563700%	14	27	0.000000%
crx	86.087000%	30	42	86.087000%	20	30	0.000000%
echocardiogram	64.885500%	5	9	64.885500%	5	9	0.000000%
horse-colic2	83.967400%	7	11	83.967400%	7	11	0.000000%
horse-colic	83.967400%	7	11	83.967400%	7	11	0.000000%
house-votes-84	95.172400%	11	16	95.172400%	11	16	0.000000%
hypothyroid	99.241200%	7	13	99.241200%	7	13	0.000000%
iris2	96.000000%	5	9	96.000000%	5	9	0.000000%
iris	96.000000%	5	9	96.000000%	5	9	0.000000%
kr-vs-kp	99.436800%	31	59	99.436800%	31	59	0.000000%
labor	73.684200%	3	5	73.684200%	3	5	0.000000%
lym	76.351400%	21	31	76.351400%	21	31	0.000000%
post-operative	70.000000%	1	1	70.000000%	1	1	0.000000%
solarflare	83.153300%	15	20	83.153300%	15	20	0.000000%
solarflare-c	84.305300%	1	1	84.305300%	1	1	0.000000%
solarflare-m	95.104400%	1	1	95.104400%	1	1	0.000000%
solarflare-x	99.136100%	1	1	99.136100%	1	1	0.000000%
zoo	92.079200%	13	21	92.079200%	13	21	0.000000%
bc	74.125900%	6	11	74.125900%	6	11	0.000000%
bcn	74.475500%	4	6	74.475500%	4	6	0.000000%
contact-lenses	83.333300%	4	7	83.333300%	4	7	0.000000%
dnf	100.000000%	2	3	100.000000%	2	3	0.000000%
dnf11	90.000000%	8	15	90.000000%	8	15	0.000000%
haberman	71.895400%	3	5	71.895400%	3	5	0.000000%
mushroom	61.964500%	32	47	61.964500%	32	47	0.000000%
平均值	85.513481%	10.19	15.85	85.513481%	9.81	15.38	0.000000%

表 5 2 种分类器的实验结果比较(精确度差>0)

数据集合	j48	叶子数目	树节点数目	CSA-tree	叶子数目	树节点数目	精确度差
dis	99.125100%	12	23	99.151600%	12	23	0.026500%
sign	85.142700%	682	1363	85.174600%	659	1317	0.031900%
pid	73.828100%	20	39	73.958300%	20	39	0.130200%
balance-scale	76.640000%	52	103	76.800000%	52	103	0.160000%
led1	73.400000%	27	53	73.600000%	28	55	0.200000%
led2	73.400000%	27	53	73.600000%	28	55	0.200000%
led	73.400000%	27	53	73.600000%	28	55	0.200000%
soybean-large	83.387600%	52	78	83.713400%	51	76	0.325800%
hungarian	80.952400%	6	10	81.292500%	6	10	0.340100%
hungarian	80.952400%	6	10	81.292500%	6	10	0.340100%
german	70.500000%	103	140	70.900000%	89	127	0.400000%
audio	77.433600%	32	54	77.876100%	30	50	0.442500%
mfeat-mor	71.150000%	110	195	71.600000%	99	184	0.450000%
tit	85.073100%	95	142	85.595000%	109	163	0.521900%
aetrain	86.850700%	236	471	87.459100%	229	457	0.608400%
aetest	86.179000%	319	637	86.794400%	318	635	0.615400%
satellite	85.858600%	323	645	86.589000%	315	629	0.730400%
vehicle	71.985800%	98	195	72.813200%	92	183	0.827400%
glass7A	96.729000%	6	11	97.663600%	6	11	0.934600%
promoters	81.132100%	19	25	82.075500%	19	25	0.943400%
bupa	68.695700%	26	51	70.434800%	28	55	1.739100%
sonar	71.153800%	18	35	74.519200%	18	35	3.365400%
平均值	79.680441%	104 36	199 36	80 295582%	101 91	195 32	0.615141%

表3~5列出了C4.5,newc45在75个实验数据上分类精确度、叶子节点数目和树的尺寸大小的对比。

在这75个实验数据集合上,J48[10]的平均分类精确度为81.787037%;newc45的平均分类精确度为81.569843%;newc45与J48的平均分类精确度差为-0.217194%,非常小,可以看出newc45在绝大部分实验数据集上取得了与C4.5一样好的分类性能。在22个数据集合上,newc45的分类精确度比J48分类器的精确度高。在27个数据集合上,newc45的分类精确度比J48分类器的精确度低。在26个数据集合上,newc45的分类精确度与J48分类器的精确度一样高。

newc45在75个实验数据集上生成的决策树的平均节点数目为149,平均叶子节点数目为82;在J48上生成的决策树的平均节点数目为149.800000,平均叶子节点数目为82.293333。虽然它们的平均节点数目和平均叶子节点相差不大,但newc45在20个实验数据集上生成的决策树比C4.5生成的要小,在46个实验数据集上生成的决策树比C4.5生成的要大,而C4.5仅仅在9个实验数据集上生成的决策树比newc45生成的小。综上所述:从统计意义上讲,newc45在75个实验数据集上生成的决策树比C4.5生成的小。对于数据集合aetrain,aetest,vehicle,soybean-large,sign,satellite,mfeat-mor,german,audio,newc45的分类精确度比J48的高,而且newc45生成的决策树比C4.5生成的小。对于newc45的分类精确度比J48低的27个数据集合中,10个数据集合cleveland,glass7,hepatitis,ionosphere,letter-recog,pendigits,sbl,segment,soybean,splice-c4.5,newc45生成的决策树比C4.5生成的小。

7 结　论

综合信息增益和增益比率、Gini索引、基于Goodman-Kruskal关联索引这三种选择分裂属性的标准的特点,通过竞争机制,用投票的方式多数胜少数的方法选择最佳分裂属性,保留了经典决策树分类器C4.5一样好的分类精确度。实验结果表明它在大部分实验数据集上,可以生成更小的决策树。

关于是否还有其他更好的分裂标准来选择最佳的分裂属性,还有待下一步研究。

参考文献:

[1] Han Jiawei, Kamber M. Data mining concepts and techniques [M]. San Francisco: Morgan Kaufmann Publishers, 2001. 185-219.

[2] Mitchell T M. Machine learning[M]. New York, America: McGraw-Hill Companies, Inc, 1997.112-140.

[3] Simovici D A, Szymon J. A metric approach to building decision trees based on Goodman-Kruskal association index[A]. The Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining[C]. Sydney, Australia:[s.n.],2004.181-190.

[4] Witten I H, Frank E. Data mining: practical machine learning tools and techniques with java implementations[M]. Seattle: Morgan Kaufmann, 2000.

[5] Quinlan R. C4.5: Pragrams for machine learning[M]. California: Morgan Kaufmann Publishers, Inc, 1993.

[6] Quinlan J R. Induction of decision trees[J]. Machine Learning, 1986, 1 (1):81-106.

[7] Kohavi R. Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree Hybrid[A]. In: Simoudis E, Han J, Fayyad U M, eds. Proc of the 2nd Int'l Conf on Knowledge Discovery and Data mining[C]. Menlo Park: AAAI Press, 1996. 202-207.

[8] Blake C L, Merz C J. UCI Repository of machine learning databases[Z]. Irvine, CA: Department of Information and Computer Science, University of California, 1998.

[9] 黄厚宽,石洪波,王志海,等.一种限定性的双层贝叶斯分类模型[J].软件学报,2004,15(2):193-199.

[10] Friedman J H, Kohavi R, Yun Y. Lazy decision trees[A]. Thirteenth National Conference on Artificial Intelligence[C]. Menlo Park: AAAI Press,1996.717-724.

(上接第105页)

际应用表明,该系统具有很强的实用性和安全性。下一步的工作主要集中在进一步拓展系统支持的应用类型。

参考文献:

[1] Danseglio M, Dillard K, Maldonado J, et al. Windows Server 2003 Security Guide[EB/OL]. Microsoft Solutions for Security and Compliance group (MSSC). http://www.microsoft.com/technet/security/prodtech/windowsserver2003/w2003hg/sgch00.mspx,2005-12.

[2] Marsh K. Win32 Hooks[EB/OL]. Microsoft Developer Network Technology Group. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwui/html/msdnhooks32.asp,1994-02.

[3] TheWinPcap Team. WinPcap Documentation[EB/OL]. http://www.winpcap.org/docs/docs31/html/main.html,2005-12.

[4] Degioanni L. Development of an Architecture for Packet Capture and Network Traffic Analysis[D]. Turin, Italy: Politecnico Di Torino,2000.

[5] Risso F, Degioanni L. An Architecture for High Performance Network Analysis[A]. Proceedings of the 6th IEEE Symposium on Computers and Communications (ISCC 2001)[C]. Hammamet, Tunisia:[s.n.],2001.