KDD 中的机器学习

Machine Learning in KDD

叶施仁 史忠植 吴 斌 宫秀军 游湘涛

(中科院计算所知识处理与人机交互实验室 北京100080)

Abstract Machine Learning (ML) Theory and ML's new development in KDD are introduced in this paper. We discuss its application backgrounds, computational complexity, classification, clustering, algorithm, methods of research, and trend in KDD

Keywords Machine learning, KDD, Machine discovery

1. 机器学习的发展

机器学习是研究获取新知识、新技巧,重组已经出现的知识的计算方法,是人工智能中的基本问题,其理论基础涉及人工智能、统计学、脑知模型、适应性控制理论、心理学模型、进化模型(如遗传算法)^[1]。机器学习的研究开始于50年代,经过40来年的发展,已经取得了很多的成果(表1),机器学习可以粗略地分为基于实例的学习、基于问题求解的学习、类比学习、机器发现等。

机器学习侧重于复杂的表示,模糊的问题以及基于搜索的方法。表示的方法包括决策树,规则,事例,簇和概念树。模糊的问题是指在缺少模型的对元组泛化,聚类,从反馈信息迟缓的大决策空间中学习。搜索方法包括贪婪搜索、遗传算法和一些前向搜索,带剪枝的广度优先搜索算法。

利用机器学习的某些技术可以建立知识库,自动的知识获取将有助于专家系统的维护和改进。甚至从现有的在线数据库中自动地发现新的知识,这些数据库的规模已经超出了人类手工的能力,早期的成功例子包括 GASOIL^[2], BMT^[3], ExpertEase^[4]等,1990年前至少有25个成功应用的报道^[5],其中 ExpertEase 的效益超过每年1000万美元。

2. 机器学习理论

机器学习是在没有很强的前提条件下进行泛化,这导致怎样衡量泛化成功程度的问题。实践表明,如果

缺少前提假设或者“偏好(bias)”,泛化是不可能的,纯粹的探索性学习是不存在的^[5]。归纳可以看成“知识的杠杆”,即使它是比演绎更高的手段,但如果没有“应用”的作用力,仍然没有意义。从某种程度说,不存在“通用的”学习方法,每一种方法的效果都基于它的前提假设,每一个问题需要具体分析。机器学习中一些常识(例如“假定越简单越正确”)实际上是一种猜测。对偏好和不同偏好组合的研究,既是应用要求的,也使我们对于偏好、知识的重要性有了新的认识,研究开发出一些利用知识的学习过程。

另一方面,机器学习的这种适应性使其很适合缺乏先验知识,或者相关知识表达困难的问题。学习的数据来源于真实世界,学习的正确性很难得到保证,这是问题的性质所决定的。机器学习可以利用计算机的强大计算能力反复验证其方法与结果。尽管问题本身很困难,随着机器学习理论的发展,近年来出现了推进算法(Boosting^[6]),筛选算法^[7]和支持向量机(SVM, Support Vector Machine^[8])等成功的方法。S. Anand 等提出的基于证据理论的数据发掘一般框架 EDM, G Shafer 的证据理论、Zadeh 的可信度理论和最近国内发展起来的未确知数学都是关于不完全信息的表达与处理的系统方法^[9]。这些方法大都在给定试验误差和衡量假说空间的有效大小的条件下,以学习器的泛化误差在边界上的形式(例如,有限空间中的模,无限空间中的 Vapnik-Chervonenkis 维)为理论基础。这些理论的最大贡献是,我们不必坚持象“误差控制在5%以内”这样的绝对标准,而是使用象“误差小于5%的概率至少是99%”这种基于概率统计的标准。

叶施仁、吴斌、宫秀军 博士生,主要研究方向:人工智能,数据采掘,史忠植 研究员,博士生导师,游湘涛 硕士生。

表1 机器学习的发展进程

	第一阶段	第二阶段	第三阶段	第四阶段
时间	50年代中~60年代中	60年代中~70年代中	70年代中~80年代中	80年代中~至今
研究内容	数值型编码问题,参数调整技术	符号表示,概念学习	ES中知识的获取,观察与发现学习,类比学习,解释学习	对大规模数据分类,聚类
特点	偏向模式识别,关心通用性,神经元模型	符号概念获取	机器发现,类比推理,辅助专家系统构造	数据库中的知识发现
典型系统算法	感知机(Rosenblatt, 1958)模拟进化过程(Friedberg, 1958),跳棋(Samuel, 1959)	积木世界学习(Winston, 1970), CLS(Hunt, 1963)	AQ11(Mitchaskr, 1980), ID3(Qurnlan, 1984), LEX(Mitchell, 1983)	CART(Stanford Univ), DBMiner(Han Jrauer, 1994), etc

1984年 Valiant 提出一种新的学习框架 PAC 学习理论^[10]。它仅要求与目标概念具有高概率的近似,而并不要求目标概念精确的辨识。设 c 是未知的概念, D 是样本空间的概率分布, 样本的大小为 m , ϵ 是 0 和 1 之间的常量, 学习算法 L 学习概念 c 是在随机试验下获得的。

▶ 在概率分布 D 下, 随机抽取表示概念 c 的大小为 m 的样本 S ;

▶ 设 h 是 L 以 S 作为训练样本下获得的假设(hypotheses);

▶ 设 $\text{error}(c, h)$ 是所有类别标记 c 与 h 不同的实例的概率, 即

$$\text{error}(c, h) = \sum_{x \in X, x \neq h(x)} \Pr[X],$$
 其中, $\Pr[x]$ 是实例 X 在 D 下的概率。

▶ 如果 $\text{error}(c, h) \leq \epsilon$, 报告成功; 否则, 报告失败。其中, error 是样本 S 在相同的概率分布下发生的。报告成功的概率反映了在概率分布 D 下, 学习算法 L 使用大小为 m 的样本学习概念 c 的成功率。

设 W_c^ϵ 是上述随机试验中报告成功的事件, 其概率为 $\Pr[W_c^\epsilon] = \sum_{S \in S_L, c} \Pr[S]$, 其中, S_L, c 是学习算法 L 得出与概念 c 接近的假设时, 大小为 m 的所有样本的集合。 $\Pr[S]$ 是抽取样本 S 的概率。

如果 $\Pr[W_c^\epsilon] \geq 1 - \delta$, ($0 < \delta < 1$), 就在给定的精确度 ϵ 和可信度 δ 下, L 从大小为 m 的样本中学习了概念 c 。因此, 评价学习算法包括使用的实例数目和计算复杂性两个方面, 它们受精确度 ϵ 和可信度 δ 的影响, 如果 ϵ 和 δ 的值越大, 计算的开销越大。学习时较多的实例具有高的成功概率和小的误差。同时, c 是概念集合 C 中的任意元, 它由概念的数目 n 和概念 c 的复杂性 s 决定。评价学习算法用样本复杂性和计算复杂性来描述, 这两个复杂性都与 n, s, ϵ 和 δ 有关。计算复杂性是最坏情况下以给定数目的样本产生假设所要求的计算时间。

在此基础上, Kearns、Li、Pitt 和 Valiant 对可表示为布尔公式的概念给出了一些新的结果。Haussler 应

用 Valiant 框架分析了变型空间和归纳偏置问题, 并给出了样本复杂性的计算公式。

计算效率和统计意义中的一个核心问题是搜索出来的模式的有效性。如果搜索过程产生的假设非常多, 评价假设的有效性和筛选有意义的部分是相当困难的。传统的有效性测试仅对一个假设进行。搜索有意义的模式, 控制搜索的进程可以避免陷入对噪音的采掘。机器学习既受有限的计算资源限制, 又受数据的质量(可学习性)的限制, 两者限定了它在应用中的效果。

研究表明, 强大的表示形式不一定导致好的结果, 适应性往往以稳定性为代价。通过统计偏好和变异能够获得这两者的平衡。初步的观察表明, 生成多个模型然后组合它们比单纯的搜索“最好的”模型要好。一些效果很好的方法, 包括打包(Bagging^[11])、堆叠(Stacking^[12])和误差校正输出码(Error-correcting Output Codes^[13]), 都是采用的这种思想。

3. 数据库中的机器发现

机器学习研究一般使用人工合成的数据, 这些数据是人类专家按照条件和结论事先组织好的, 实例是精心选择的, 反映一定的语义规则, 数目少, 无噪音, 通常覆盖整个状态空间; 而真实问题的数据是所给定的数据通常并未完全覆盖状态空间, 带有一定的噪音, 隐含的规则为人类专家事先并不知道, 样本和特征的数目以及噪音的频率导致传统的机器学习算法难以胜任。

KDD(Knowledge Discovery in Database, 数据库中的知识发现)是从数据集中抽取和精化新的模式的过程, KDD 的目标是识别出存在于数据库中的可信的、新颖的、有效的并能被人理解的模式^[14]。KDD 给机器学习的研究带来了巨大的机遇和诱人的前景。它使昔日关心的那些问题又重新变成研究的热点, 大量的真实数据使研究的成果得到论证。机器学习方法在当前的一些数据库工程中令人意外地取得了一些引人注目的结果, 这是别的方法难以获得的。另一方面, KDD 拓宽了机器学习的研究领域, 与数据库、统计学、

人机交互、可视化、信息检索、高性能计算的交叉发展。

机器学习关注计算效率和统计意义, KDD 中尤其如此。消极学习 (lazy learning) 与积极学习 (eager learning) 是计算开销的一种反映, 多数传统的方法都是积极的, 在学习的阶段花很大的代价对问题进行全面的泛化, 而如近邻算法这种消极方法, 所有的泛化和计算工作都在测试时进行, 两者结合起来不失一种好的方式, 例如, RISE 系统中采用自动确定规则和近邻的最佳组合^[15]。

机器学习和 KDD 的一个重要特征是学习的结果是可以理解的。然而, 可理解性是很难以表达的, 完全主观的。许多机器学习方法产生的模型, 使用 "if... then..." 的产生式, 可视化的方式表示, 对没有数学背景的人也是容易理解的。除了直接使用这些容易理解的方法外, 还可以将难以理解的模型 (如神经网络) 转化成容易理解的模型 (如决策树)^[16]。

采掘海量数据的算法必须具有线性的或者近似线性的时间复杂度, 因此大多数的机器学习算法必须改进。少数算法可以无损失地优化, 输出结果不会改变。多数的学习算法只能采用有损失的优化, 产生的结果未必有同样的精度, 但只需要耗费线性的时间。RIPPER^[17]、CWS^[18] 就是用这种方法对规则归纳算法的改进。

如果训练的数据超出了内存, 学习算法必须考虑怎样高效地从磁盘中读取。例如以顺序的方式访问数据, 而不是随机的方式; 尽可能地减少访问的次数。SLIQ、SPRINT 算法, 关联规则发现算法都体现和解决了这些问题^[19, 20]。理想的情形是, 学习算法仅需访问稳定的内存数据, 对全部数据的外部扫描只要一次, 同时, 通过抽样和综合两种途径对海量数据约简是不可避免的。例如, 将样本分成若干子集, 然后分别对它们进行学习, 将学习的结果再组合起来, 这就是一种抽样的方法。又如, 对样本分组, 然后反复选取那些能够提高学习效果例子加进来^[21]。综合的方法试图产生适合内存大小的数据, 然后对这些数据进行访问。综合的数据可以是统计出来的^[22, 23], 也可以是利用压缩技术得到的^[24]。抽样和综合可以互为补充, 两者结合起来使用。

在很多应用中, 可供使用的数据是很多的, 但对学习符合某种精度要求的概念并非必要。有时是很多的数据也未必得到期望的结果。如果实现能根据某些启发式信息, 确定多少数据是必需的, 这是很有用的。利用统计测试出来的学习曲线可以给我们有价值的线索。利用估计的出错率之类的贝叶斯参数, 一旦逼近某个值则停止学习。

将学习算法引入到 KDD 中, 需要注意 KDD 问题

自身的特点。例如, 机器学习中认为所有的错误代价都相同, 但实际的问题可能这样: 如果一个类包含很多对象, 高的正确性是容易获得的, 但价值不大; 对包含较少元组的类 (如潜在的客户) 错误划分的损失很大。

机器学习辅助过程的方法和形式化也是重要的研究方向。采掘数据的整合、清洗、预处理需要人工干预, 因此时间往往很长。这些步骤的自动化将大幅度地提高处理速度、降低成本并拓宽了应用范围。利用标准的机器学习构件重构可以快速生成可靠的、自足的学习系统, 便于非专业人士使用、有利于 KDD 系统大规模地铺开。这方面的工作尚不如人意^[25], 问题的描述和采用的技术还不成熟。尽管出现了一些不完善的向导和规则工具, 构件式的机器学习系统的结构、各种技术的集成, 这些在理论上和应用实验上都需要进一步的探索。

将机器学习应用到海量数据并非算法简单的比例缩放。传统方法是发现数据库中所反映的每一个模型, 它们是局部的模式或反模式 (异常)。与构造模型的方法相比较, 机器学习的理论还不够深入。目前, KDD 中一般都很关心效率。实际上, 泛化的合理性才是机器学习的核心, 围绕这个问题的工作十分有意义。如果数据库的规模超出了模型的能力, 即使可以使用统计或压缩的方法抽样, 选择合适的 "关注的焦点" 仍十分必要, 启发式信息和资源可以帮助我们获得问题的焦点。此外, 对多年积累的未经处理的数据库, 假定独立等同分布并不一定成立。考虑样本并非独立的, 过去、现在、将来的数据分布未必相同, 具有重要的意义。

在机器学习的研究中, 实例一般表示成特征向量, 每一个特征是一个数值或者符号, 所有的特征保存在一张表中。但大多数 KDD 应用中数据, 如 Web 页面由文本、HTML、图像和声音、动画等组成, 数据的结构和处理方式非常复杂, 有的还包含有时间和空间信息。即使是包含许多表和关系型数据库, 很少能够在不损失任何信息的情况下轻易地将它们关联成一张表。例如, 在许多公司的采掘应用中, 客户的行为是公司最关心的。而客户通常按照职业等特征层次性集聚在一起, 产品也是按层次性目录组织的。如果简单地把它们转换成一些向量来学习 (目前很多情况就是这样处理的), 完全可能忽略掉很多有意义的模式, 因此改进机器学习算法十分必要。

4. 分类

机器学习主要关心分类问题, 因为它是许多其它问题的基础和核心。

分类通过对具有类别标记的实例 (数据) 进行训练, 得出一个能够预测新实例类别的模型。设 $D_{train} =$

$\{d_1, \dots, d_m\}$ 为训练的实例集合, 每一个实例都有预先标记好的类别 $c_i, \{c_i\} = C$ 。通过对这些数据的有导师学习 (supervised learning), 产生一个称为分类器的模型, 它对不知道类别标记的 (测试) 实例预测其类别。预测的准确程度可以评价分类器的性能。常用的分类方法有基于贝叶斯网络的和基于决策树的。

设每一个实例包括一个决策属性和 n 个特征属性 $\{x_1, \dots, x_n\} = X$ 。贝叶斯网络由一个表示类别变量的节点 C 和若干个表示特征的节点 X_i 组成。利用贝叶斯网络可以计算 X 属于某一个类别 c_k 的概率 $P(C=c_k | X=x)$, 如果知道概率分布 $P(c_i | X)$, 通过选择概率最大的类别 c_k , 就可以得到最优的贝叶斯分类。当然, 真正的概率分布是不知道的, 我们只能根据训练数据得到近似的概率分布。

我们可以用多种方法构造贝叶斯分类器, 试验表明, 特征变量与类别变量直接关联的方法, 其分类性能较好, 朴素贝叶斯分类就是这种类型, 它是最简单的, 也是现在应用最广泛的。朴素贝叶斯对问题进行了简化, 限定对给定的类别变量、特征属性是不相关的, 即 $P(X|C) = \prod_i P(X_i|C)$ 。实际问题的特征属性往往并不具有这样的性质, 但即使这样, 朴素贝叶斯在某些情况下仍然具有很好的效果^[26]。文^[27]提出了对朴素贝叶斯分类的改进方法, 但仍有效果不理想的情形。

为了提高分类的准确性, 我们可以通过多种反映特征属性之间相关性的方式来提高朴素贝叶斯分类器的性能, 例如, 允许特征属性节点除了类别属性节点外还有其它父节点。但即使限定其父节点至多两个, 求最优的贝叶斯分类器仍然是 NP 问题^[28]。在最坏的情形, 任何构造最优的分类器算法的复杂性都是特征数的指数次方。为此, 文^[29]提出了 Tree Augmented Naive-Bayes (TAN) 算法, 它限定每个特征属性除了分类属性外至多有一个其它父节点, 算法的复杂性为特征数的平方。

决策树的方法包括 CART 算法^[30], ID3 算法, C4.5^[31], C4.5 采用贪婪和分治的方法构造决策树, 选取增益比最大的特征拆分当前节点:

$$GainRatio(X_i) = Gain(X_i) / splitInfo(X_i)$$

$$\begin{aligned} \text{其中, } Gain(X_i) &= - \sum_c P(c) \log P(c) + \\ &\quad \sum_{x_i} P(c|x_i) \log P(c|x_i) \\ &= H(C) - H(C|X_i) \\ &= MI(C, X_i) \end{aligned}$$

$$SplitInfo(X_i) = - \sum_{x_i} P(X_i) \log P(X_i) = H(X_i)$$

增益 $Gain(X_i)$ 是特征 X_i 对类别标记 C 的熵 (entropy H) 减少的程度, 即 X_i 是关于消除 C 的不确定性

的, 它与信息论中关于 C 与 X_i 的互信息是等价的。SplitInfo(X_i) 表示剔除 X_i 拆分当前节点后的熵。

尽管可以给出一些指导方针, 应用要取得成功仍然依赖机器学习专家的经验和技术。例如, 为了使用 ID3, 必须先将问题转换成实例的分类形式, 设计合适的实例描述形式, 其属性要能反映问题的本质, 类别标记要合适, 数据在开始时或许要变换重组过滤。这些过程可能还得反复多次, 直到找到可行的解决方案。

5. 聚类

与分类不同的是, 聚类是发现没有类别标记数据的类目, 因此是一种无监督归纳 (unsupervised induction)。类别标记需要由聚类学习算法来自动确定。除了机器学习外, 聚类还可以使用基于统计、神经网络和面向数据库技术的方法。设数据 $D = \{d_1, \dots, d_m\}$, 聚类的目标是将 D 划分为 k 个簇 $c_1, \dots, c_k$ 。这种划分可以是明确的, 将每个 d_i 指定唯一的簇; 也可以是含糊的, 指出其属于多个簇的概率。

在统计方法中, 聚类称聚类分析 (clustering analysis), 它是多元数据分析的三大方法之一 (其它两种是回归分析和判别分析)。主要研究基于几何距离的聚类。传统的统计聚类分析方法包括系统聚类法、分解法、加入法、动态聚类法、有序样品聚类、有重叠聚类和模糊聚类等。新的有名聚类方法是 AutoClass, 它使用了贝叶斯方法来聚类。

很多人工智能文献中, 聚类也称概念聚类 (concept clustering), 因为这里的距离不再是统计方法中的几何距离, 而是根据概念的描述来确定的。当聚类对象可以动态增加时, 概念聚类则称为概念形成 (concept formation)。典型的概念聚类方法有: COBWEB、OLOC 和基于列联表的方法。RUMMAGE 系统采用自顶向下的形式构造分类树, 每一个候选的簇采用最简单性和最小重叠的标准评价^[32], 其算法为:

- 1) 创建分类树的根节点;
- 2) 对每一个属性, 将对象按照其属性值排序;
- 3) 对属性的每个值, 根据它生成实例的概念描述;
- 4) 选择关于该属性的最好描述 (最简单的或者简单的);
- 5) 根据当前节点的每一个属性值从当前节点建立分支, 将实例分配到子节点中;
- 6) 递归地重复步骤 2~5, 直到生成的子树满足某种条件。

小结与讨论, 在最近和将来的 KDD 应用中, 机器学习系统应该能够连续运行, 从在线数据流中学习, 不断调整自身的行为, 并且必须保持可靠性和较少的“人工干预”。这一趋势在电子商务站点中能够反映出来。

系统自动地对新的用户根据其表现做出不同的响应;股票交易也需要这种技术;商业机构必须快速连续地对不断变化的市场做出反应,有人开始了在这方面的研究^[2],但一些问题必须考虑到。如怎样和新的在线数据链接,区分两种对学习模型进化的影响因素:由数据积累和学习过程不断进行带来;由环境变化体现的。

KDD 应用中,学习并非单独的过程,而是嵌入大型系统之中。学习算法与后台的数据库系统必须有一种高效的数据交互方式,与此相关的问题包括,怎样的抽样是最有效的,怎样使用它们,怎样转化到对数据库的顺序扫描。同时,机器学习与数据库的交互还包括用户和系统对元数据的使用。例如,字段的定义与对其值的约束,可以作为背景知识或者在学习过程中使用。

KDD 与数据仓库结合才能发挥其全部优势,但 KDD 与数据仓库集成是复杂的,不是一蹴而就,但机器学习本身可以使部分工作自动化。例如,经常遇到的判断关系型数据库中字段的相关性,Web 搜索结果的问题。它们可以形式化成学习的形式:给定目标模式 $\{X_1, X_2, \dots, X_n\}$ 及其训练实例,归纳出一般的规则,得出由 X_i 组成的泛化规则。对一张模式为 $\{Y_1, Y_2, \dots, Y_n\}$ 的表,目标是将其中的每一行 Y 划分到 X 中(可能没有这样的 X),这种映射是互相约束的。数据清洗是构造数据仓库的另一个关键,机器学习可以作为有效的手段。海量数据库中几乎不可避免地包含大量的噪音和缺值,只有机器学习算法能够处理存在噪音和缺值问题的数据时,机器学习才能从实验室搬到各种实际应用之中。模型系统中数据存在误差和缺值,怎样将其影响降到最低,是值得研究的。

大量的科学和商业用途使机器学习应用日趋广泛,渗透到各个领域,所起作用也并非简单的类别预测。KDD 中用户(包括专家和非专家)的交互是非常重要的。现在,各种 KDD 的应用,如金融、市场、分子生物学、异常甄别等,为机器学习的探索和测试提供了很好的环境。同时,领域的特殊性对算法的特殊要求也将促进其发展。

参考文献

- 1 Nilsson J. Introduction to Machine Learning. Stanford Univ., 1996
- 2 Slocombe S, Moore K, Zelouf M. Eng. expert system applications. In: Annual Conf. of the BCS Specialist Group on Expert Systems. 1986
- 3 Hayes-Michie J E. editor. Pragmatica; Bulletin of the Inductive Programming Special Interest Group, vol 1. Turing Institute Press, Glasgow, UK, 1990
- 4 Leech W J. A rule-based process control method with feedback. Advances in Instrumentation, 1986, 41: 169 ~ 175
- 5 Wolpert D. The lack of a priori between learning algorithms. Neural, 1996. 1341~1390
- 6 Freund Y, Schapire R E. Experiments with a new boosting algorithm. In: Proc. of the 13th Int'l Conf. on ML Learning. Bari, Italy: Morgan Kaufmann, 1996. 148~156
- 7 Littlestone N. Learning quickly when irrelevant attributes abound. A new linear-threshold algorithm. 1997. ML.285~318
- 8 Scholkopf B, Burges C, Smola A. Advances in kernel methods: Support vector machines. Cambridge, MA: MIT Press, 1998
- 9 史忠植. 高级人工智能. 科学出版社, 1998. 113~114
- 10 Valiant L. A theory of the learnable. CACM, 1984, 27(11): 1134~1142
- 11 Breiman L. Bagging predictors. ML, 1996. 123~140
- 12 Wolpert D. Stacked generalization. Neural, 1992. 241~259
- 13 Kong E B, Dietterich T G. Error-correcting output coding corrects bias and variance. In: Proc. of 12th Int. Conf. on Machine Learning. Tahoe City, CA, 1995. 313~321
- 14 Fayyad U M, Piatetsky-Shapiro G, Smyth P. From Data Mining to Knowledge Discovery: An Overview. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996
- 15 Domingos P. Unifying instance-based and rule-based induction. Machine. In: Proc. of 2nd Int'l Conf. on Knowledge Discovery & Data Mining. Portland, OR. AAAI, 1996. 141~168
- 16 Craven M W. Extracting comprehensible models from trained neural networks. Doctoral dissertation, Dept of Computer Sci., Univ. of Wisconsin-Madison, Madison, WI, 1996
- 17 Cohen W W. Fast effective rule induction. In: Proc. of the Twelfth Int'l Conf. on ML. Tahoe City, CA: Morgan Kaufmann, 1995. 115~123
- 18 Domingos P. Linear-time rule induction. In: Proc. of the Second Int'l Conf. on Knowledge Discovery and Data Mining. Portland, OR: AAAI Press, 1996. 96~101
- 19 Mehta M, Agrawal A, Rissanen J. SLIQ: A fast scalable classifier or data mining. In: Proc. of 5th Int'l Conf. on Extending Database Tech. Avignon, France: Springer, 1996. 18~32
- 20 Agrawal R, Mannila H, Srikant R, Toivonen H, Verkamo A I. Fast discovery of association rules. In: Advances in knowledge discovery and data mining Menlo Park, CA: AAAI, 1996. 307~328
- 21 Musick R, Catlett J, Russell S. Decision theoretic subsampling or induction on large databases. In: Proc. of the Tenth Int'l Conf. on ML. Amherst, MA: Morgan Kaufmann, 1993. 212~219

(下转第68页)

Bean 编程技术,运行环境是使用 Java 语言编写的 Web 服务器(起代理服务器的作用);(3)应用服务层: Java Servlet 编程技术、RMI 技术、JDBC 技术和 Java Bean 编程技术,运行环境是使用 Java 语言编写的 Web 服务器。

多层结构较两层结构和三层结构主要有伸缩性好、可管理性好、安全性高、可重用性好以及节省开发时间等诸多优点,然而,如何在理论和实践相结合的基础上高效地运用多层结构模型,其中还有一些问题需要我们去进行深入的研究和探讨,如 Web 技术、Java 技术以及分布式对象技术与数据库技术的有效集成问题。

总结与展望 我们在做国家“863”项目——基于 Web 的通用网络管理系统时,采用了基于 JDBC 的三层数据库访问模式。实践证明,基于 Java 语言的 JDBC 技术由于具备了 Java 语言的健壮性、安全性、平台无关性以及可自动下载等特点,因而具有其它数据库集成技术所不可比拟的优势。JDBC 技术和其它基于 Web 的数据库连接技术相比,主要有以下优点:

·JDBC 可以跨越不同的系统平台,它一次编译后,可以多次在不同的系统上执行,而无须另行开发。

·JDBC 可以将远端数据库中获得查询信息的 Java Applet 程序嵌入到 Web 页面中,简化了使用和操作。

·使用 JDBC 能够方便地向异种关系数据库发送 SQL 语句,获取存储于不同数据库管理系统上的数

据,且不影响继续使用已有的数据库。

·通过 Java 和 JDBC,可以在 Web 上经济有效地发布信息;同时,Java 语言的跨平台性缩短了开发时间,简化了软件安装和版本升级管理。

在 Internet/Intranet 上,以 Web 技术为核心,并结合 Java 技术,不仅可以构建一个与用户平台无关的统一而简单的交流方式,把已有的和未来的有关数据库、安全设施等各种技术纳入以 Web 为核心的 Browser/Server(浏览器/服务器)模式中,而且能够使人们突破传统工作模式,在更大范围内共享资源和协同工作。今天,计算机和网络技术正以飞快的速度向前发展,可以说网络技术的发展带动了数据库技术的发展,而数据库技术只有与飞速发展的网络技术特别是 Internet/Intranet 技术相适应,才会加快它的应用和发展速度。相信基于 Java 的 Web 数据库技术将成为 21 世纪数据库应用的发展方向。

参 考 文 献

- 1 王克宏. Java 语言 SQL 接口 JDBC 编程技术. 北京:清华大学出版社,1997
- 2 廖卫东,等. Java 程序设计使用指南. 北京:机械工业出版社,1996
- 3 王克宏. Java 技术及其应用. 北京:高等教育出版社,1999
- 4 Rubin A D, Gear D E Jr. Mobile Code Security. IEEE Internet Computer, 1998(Nov.-Dec.)
- 5 Li Gong. Security Java Class Loading. IEEE Internet Computer, 1998(Nov.-Dec.)

(上接第 100 页)

- 22 Moore A W, Lee M S. Cached sufficient statistics or efficient ML with large datasets. J. of AI, 1997, 67~91
- 23 Graefe G, Fayyad U, Chaudhuri S. On the efficient gathering of sufficient statistics or classification from large SQL databases. In: Proc. of the Fourth Int'l Conf. on Knowledge Discovery and Data Mining. New York, NY: AAAI, 1998. 204~208
- 24 Davies S, Moore A. Using Bayesian networks or lossless compression in data mining. In: Proc. of the Fifth Int'l Conf. on Knowledge Discovery and Data Mining. San Diego, CA: ACM, 1999
- 25 Kohavi R, Sommer F D, Dougherty J. Data mining using MLC++, a Machine learning library in C++. Int'l J. on AI, 1996, 537~566
- 26 Mitchell T M. Machine Learning. McGraw-Hill, 1997
- 27 Domingos P, Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 1997, 29: 103~130
- 28 Chickering D M. Learning Bayesian networks in NP-complete. In Lecture Notes in statistics, 1995
- 29 Friedman M, Goldszmidt T. Building classifiers using bayesian networks. In: Proc. of 13th National Conf. on AI, 1996
- 30 Breiman L, Friedman J H, et al. Classification and Regression Trees. Wadsworth Int'l Group, 1984
- 31 Quinlan J R. C4. 5: Programs for Machine Learning Morgan Kaufmann Pub. ,Inc. , Los Altos, CA, 1993
- 32 Fisher D, Langley P. Approached to Conceptual Clustering. In: Proc. of 9th Int'l Joint Conf. On AI, Los Angeles, CA. 1985. 691~697
- 33 Widmer G, Kubat M. Special issue on context sensitivity and concept drift. 1998, ML32(2)