基于数据库抽样的海量数据分类算法研究

李雪婵
(广东广播电视大学 广州 510091)

摘 要 本文对目前比较优秀的各种分类方法进行了介绍、分析和比较。在此基础上,借鉴决策树方法的快速分类特性,提出了一种基于数据库抽样的海量数据分类算法,给出了算法的设计思想和实现原理,并对多处理环境下的优化进行了探讨。实验研究表明,该算法可以明显提高海量数据库的分类效率。
关键词 分类,算法,海量数据,数据库

Research on Classification Calculation Way of a Great Amount of Data According to the Database Sampling

LI Xue-chan
(Guangdong Radio & TV University, Guangzhou 510091, China)

Abstract In this paper, some excellent classifying methods are introduced and analyzed first. Then the quick classifying character of decision tree method is used for reference, and a mass data classification algorithm is proposed based on database sampling. Both the designing thoughts and implementation principle of the algorithm are given. The optimization of the algorithm is also discussed in multi-processor environment. An example shows that, this classifying algorithm can improve the efficiency of classification in mass database.
Keywords Classification, Algorithm, Mass data, Database

分类是基于数据库的知识发现中重要的研究领域之一。分类的目的是根据数据集的特点构造一个分类函数或分类模型(也可称作分类器),该模型能把未知类别的样本映射到给定类别中的某一个。分类可用于预测,其目的是从历史数据记录中自动推导出对给定数据的推广描述,从而能对未来数据进行预测。和回归方法不同的是,分类的输出是离散的类别值,而回归的输出是连续或有序值。

构造模型的过程一般分为训练和测试两个阶段。在构造模型之前,要求将数据集随机地分为训练数据集和测试数据集。在训练阶段,使用训练数据集,通过分析由属性描述的数据库元组来构造模型,假定每个元组属于一个预定义的类,由一个称作类标号的属性来确定。由于提供了每个训练样本的类标号,该阶段也称为有指导的学习。通常,模型用分类规则、判定树或数学公式的形式提供。在测试阶段,使用测试数据集来评估模型的分类准确率,如果认为模型的准确率可以接受,就可以用该模型对其它数据元组进行分类。

1 现有分类方法简述

数据分类实际上就是从数据库中发现数据对象的共性,并将数据对象分成不同类别的一个过程。目前已经提出很多分类算法,但归纳起来一般可分为决策树方法、统计方法、机器学习方法、神经网络方法等几种。

(1)决策树方法

决策树是以实例为基础的归纳学习方法。它从一组无次序、无规则的元组中推理出决策树表示形式的分类规则。它采用自顶向下的递归方式,在决策树的内部节点进行属性值的比较,并根据不同的属性值从该节点向下分支,叶节点是要学习划分的类。从根节点到叶节点的一条路径就对应着一条合取规则,整个决策树就对应着一组析取表达式规则。SPRINT算法[5]是其中最具代表性的一种方法。

(2)贝叶斯分类方法

贝叶斯(Bayes)分类是统计学分类方法,它利用概率统计知识进行分类。该方法能运用到大型数据库中,且方法简单、分类准确率高、速度快。它主要利用贝叶斯定理来预测一个未知类别的样本属于各个类别的可能性,选择其中可能性最大的一个类别作为该样本的类别。由于贝叶斯定理假设一个属性值对给定类的影响独立于其它属性的值,而此假设在实际情况中经常是不成立的,因而其分类准确率可能会下降。

(3)基于关联规则的分类方法

此类方法一般分两个步骤构造分类器。第一步:发现所有右部为类别属性值的类别关联规则。第二步:从已发现的规则中选择高置信度的规则来覆盖训练集。也就是说,如果有多条关联规则的左部相同,而右部为不同的类,则选择具有最高置信度的规则作为可能规则。该算法对于发现隐藏于大量事务记录中的关联规则来说是比较有效的。但也存在一些缺陷,如为了防止漏掉某些规则,最小支持度经常被设为0,此时该算法就发挥不了优化作用,结果是产生的频繁项目集有时多得在内存无法容纳,从而使得程序无法继续运行。

(4)基于数据库技术的分类方法

虽然数据挖掘的创始人主要是数据库领域的研究人员,然而至今为止提出的大多数算法均没有利用数据库的相关技术,数据挖掘应用也很难与数据库系统集成,此问题已成为该领域研究的关键问题之一。在基于数据库技术的分类算法中,比较有代表性的有 MIND 和 GAC-RDB。MIND 算法是采用数据库中用户定义的函数(user-defined function,简称 UDF)发现分类规则的算法。该算法的优点是通过采用 UDF 实现决策树的构造过程,使得分类算法易于与数据库系统集成。其缺点是算法用 UDF 完成主要的计算任务,而 UDF 一般是由用户利用高级语言实现的,无法使用数据库系统提供的查询处理机制,无法利用查询优化方法,且 UDF 的编写和维护相当复杂。GAC-RDB 算法是一种利用 SQL 语句实现的分类算法。该算法采用一种基于分组记数的方法统计训练数据集中各个属性取值组合的类别分布信息,通过最小置信度和最小支持度两个阈值找出有意义的分类规则。该算法使用

李雪婵 讲师,工程硕士,主要研究方向为计算机应用和数据库。

关系数据库系统提供的聚集运算功能，利用SQL语句完成主要的计算任务，使得应用程序易于与数据库系统集成。

除了上述描述的多种分类算法之外，还有一些其他分类算法，如后向传播分类、k-最近邻分类、基于案例的推理、遗传算法、粗糙集和模糊集方法等。

2 一种基于数据库抽样的海量数据分类算法

在众多的分类算法中，决策树分类以其分类速度快、模型简单、便于理解等优点得到了广泛的应用。在对海量数据进行分类时，由于要对每一个属性值进行评估，计算量大，花费的时间长。通常，决策树分类方法可以有较快的执行速度，但以降低精度作为代价。本文在决策树分类方法的基础上，提出一种基于数据库抽样的高效分类算法，该算法能较好地对海量数据进行分类，同时保证较高的精确度。

2.1 设计思想

本算法采用宽度优先策略构造决策树。首先，从原始数据库中抽取样本数据，通过分析样本数据为每个属性寻找一个最优分割点，然后在整个数据集上使用某个分割指数，在这些划分中找到最优的划分属性。在本算法中采用的分割指数仍然是SPRINT算法中使用的gini参数。假设一个训练集 S 有 n 条记录，它们分别属于 m 个不同的类，则集合 S 的gini定义如下：

$$\mathrm{gini}(S)=1-\sum_{i=1}^{m}p_i^2$$

其中 P_i 是类 i 在 S 中出现的相对频率。

如果集合 S 按照某个分割点被划分成 S_1 和 S_2，则划分后的gini按下式计算：

$$\mathrm{gini}_s(S)=(n_1/n)*\mathrm{gini}(S_1)+(n_2/n)*\mathrm{gini}(S_2)$$

其中 n,n_1,n_2 分别为 S,S_1,S_2 的记录数。$\mathrm{gini}_s(S)$越小，表明分割规则越好。

对数值型属性而言，候选分割点为训练集中属性按值排序后的相邻属性值的中点。若某属性 A 的某个中点值为 V，可以将训练集分成 $A<V$ 和 $A>V$ 两个部分。若某数值型属性有 M 个互不相同的取值，则候选分割点为 $M-1$ 个。对于非数值型属性，若某属性有 K 个互不相同的取值，划分的实质是将 K 个值分成两个集合。

说明：①本算法在处理数值型属性时，按下列方法寻找该属性的最优分割点。设 A 是原始数据库 D 中的某一个数值属性，D 中包含 n 条记录，通常这 n 条记录在 A 属性上有 n 个不同的取值。随机从 D 中抽取 S 条记录作为样本数据 S，按属性 A 排序，得到属性值序列。然后，从该序列中均匀地抽取 $M-1$ 个点，利用这些点将属性 A 的取值分割成 M 个区间。在对该属性计算最优分割点时，只对 M 个区间的边界值计算gini参数，明显减少了计算量。

②本算法将样本数据按照最优划分原则进行分割的同时，也将原始数据集一同划分。

③当把某一节点上的样本数据 S 划分形成 S_1 和 S_2 后，它们将作为2个子节点的样本数据，这样避免了每次计算时都重新抽样和排序。

通常，样本数据量越大，抽样算法的精度就越高。反之，则精度降低。

2.2 算法描述

在划分抽样数据之后，已经得到了划分空间。此时若再对原始数据库进行划分，只需对数据库扫描一遍，就能在划分的同时把类出现的次数一起记录下来。区间上每类的记录数称为类分布。该算法构造决策树时，将原始数据进行分类。

本算法描述如下：

(1)当树的层次 $L=1$ 时：

①扫描整个数据库，得到样本数据；

②对样本数据中每个数值型属性执行：对数值型属性进行排序；从该序列中均匀地抽取 $M-1$ 个值作为划分区间的边界值；扫描原始数据库，得到 M 个区间的类分布计数。

(2)当 $L>1$ 时：

对决策树 L 层的每个节点执行：

①如果该节点上的数据属于同一类，停止对该节点的操作；

②在样本数据上对所有的数值型属性计算gini参数，从而分别找到它们的最优分割点；

③在所有属性的最优分割点中，找到最小的gini参数值，从而得到本次划分的最优分割点；

④根据最优分割点把当前节点的样本数据 S 分成2个子集，得到新的划分区间 S_1 和 S_2；

⑤扫描当前节点上的原始数据，更新类分布计数；

(3) $L=L+1$，重复执行(2)。

2.3 算法在多处理器计算机上的实现

对于海量数据库而言，要对数以亿计的数据进行分类，其对硬件设施的要求是非常高的，一个处理器一般很难完成，即便能够完成，所耗费的时间也是不可接受的。因此，对于海量数据库的分类操作一般都在有多个处理器的小型机或中型机上完成。在这种情况下，我们需要对上述算法进行并行化，即将分类任务合理地分配到多个处理器上，使这些处理器并行运算、协同工作，以最大限度地提高运算效率。

在并行算法中，主要问题仍然是寻找最优分割点以及如何划分数据。与SPRINT算法的并行化方法相似，原始数据均匀分布在各个处理器上。也就是说，将原始数据库 D 大致均匀地划分成多个小数据库 D_i，让每个处理器正好存放一个小数据库。每个处理器并行地抽取样本数据。在每个处理器 P_i 保存全部的样本数据，每个处理器可以单独地决定每个属性上的 M 个划分区间。在选取划分区间时，处理器之间不需要进行通信。在对决策树 L 层上的某个节点进行处理时，需用一次通信，要求处理器之间交换本地各个区间点的类分布情况。此后，可以独立地计算gini参数值，得到最优的划分。实际上，这里是用冗余的存储减少通信量，以提高执行速度。具体描述如下：

STEP 1 处理器 P_i 在 D_i 上抽取样本数据 S_i，对其上的属性列表进行排序；

STEP 2 处理器之间交换属性列表和样本数据。在每个处理器上，对每个属性进行排序，最终得到所有属性的有序列表。此时，每个处理器上都保存有序的样本数据，记为 S；

STEP 3 P_i 根据 S 独立地确定划分区间的边界值，并在本地数据上计算每个区间上的本地类分布情况；

STEP 4 各处理器交换本地信息，得到全局类分布情况；

STEP 5 在每个处理器上，处理第 L 层的每个节点：

①如果该节点上的数据属于同一类，算法返回；

②否则，P_i 开始计算，求得最优的分割点；

③各处理器独立地划分样本数据和本地的原始数据；

④样本数据划分成子集 S_1 和 S_2，作为决策树 $L+1$ 层节点的样本数据，同时计算本地类分布情况；

STEP 6 重复执行STEP 4和STEP 5。

在实现过程中，每个处理器对本地属性列表的排序使用堆排序的方法，在STEP 2中可以使用二路归并的方法完成并行排序。由于在有序的样本数据上获得的 $M-1$ 个区间边界值也是有序的，而且每个处理器上顺序相同，因此只需按相同的顺序进行交换，不需要额外的信息。该方法使得通信限定在最小的范围内。当内存空间较大时，该算法效率较高。

此外，该方法也可以进行改进，不必让每台处理器都存放全部的样本数据，只用一台处理器负责对样本的控制，由它决定区间的划分边界，每次由该处理器把划分区间的边界值广播给其它处理器。这样，每划分一个节点，增加一次广播划分区间边界值的操作。由于区间的个数相对较少，每次的通信量不大，既保证了算法的性能，也节省了内存空间。

3 实验结果及对比

我们用 Matlab 语言实现了该算法，并在计算机上进行了实验，与 SPRINT 算法进行了比较。实验数据通过 QUEST 实验数据生成器产生。限于实验环境，该算法的样本数据为原始数据的 10%，数据量从 10k 到 100k 之间变化，区间个数从 10 到 100 之间变化，实验结果如表 1 所示。

表 1 实验数据及结果对比

数据量	SPRINT 算法执行时间(单位:秒)	新算法在不同区间数目的执行时间(单位:秒)		
		100	50	10
10k	87	8	5	2
30k	425	10	10	7
50k	1003	14	12	9
80k	太长，中断	25	17	14
100k	太长，中断	36	23	20

实验表明：该算法的执行速度明显优于 SPRINT 算法，数据量越大，该算法的优势越明显。从实验可以看出，该算法是一种有效的基于数据库抽样的数据分类方法。该算法串行运行速度明显比 SPRINT 算法快，而且可以很方便地用 SQL 语句实现。在样本数据和原始数据量足够大时，可以近似地认为样本数据真实地反映出原始数据属性的分布情况。因此，该算法是解决海量数据分类的有效方法。可以预见，在多处理器并行运行时，随着参与的处理器的数目的增加，该算法的执行效率将进一步提高。

结束语 分类是基于数据库的知识发现中一项重要的研究课题。在数据量急剧增长的时代，算法的执行速度、可伸缩性以及输出结果的可理解性等特性显得尤为重要。此外，由于分类的效果一般和数据的特点有关，有的数据噪声大，有的缺值，有的分布稀疏，有的属性间相关性强，有的属性值是离散的，而有的是连续的或混合的，这些都会不同程度地影响分类效果，目前还不存在能适合各种不同数据的优良分类方法。本文所采用的算法还是基于决策树框架，其它方法是否存在更有效的算法，有待于以后进一步探索。

参 考 文 献

[1] 邵峰晶，于忠清. 数据挖掘原理与算法. 北京：中国水利出版社，2003
[2] 刘红岩，陆宏钧，陈剑. 利用数据库技术实现的可扩展的分类算法. 软件学报，2002(6)：1076-1081
[3] 胡侃，夏绍玮. 基于大型数据仓库的数据采掘：研究综述. 软件学报，1998(1)：53-62
[4] 王实，高文. 数据挖掘中的聚类方法. 计算机科学，2000(4)：42-45
[5] Shafer J C, Agrawal R, Mehta M. SPRINT: A scalable parallel classifier for data mining // Proceedings of 1996 International Conference on Very Large Databases. Bombay, India, 1996: 544-555
[6] Guha S, Rastogi R, Shim K. Cure: An efficient clustering algorithm for large database // Proceedings of 1998 ACM-SIGMOD International Conference on Management of Data. Seattle, WA, 1998: 73-84

(上接第 296 页)

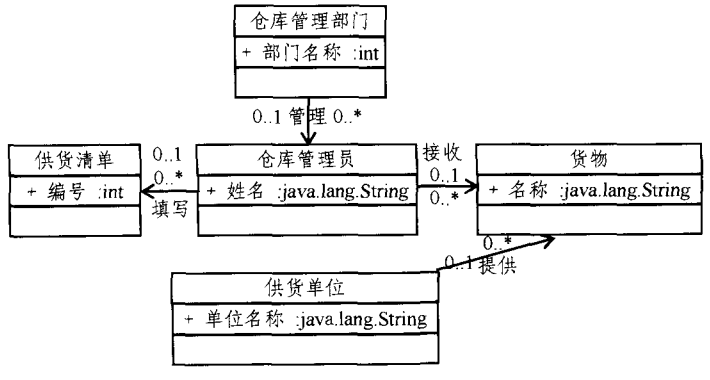

图 3 仓库管理系统类图

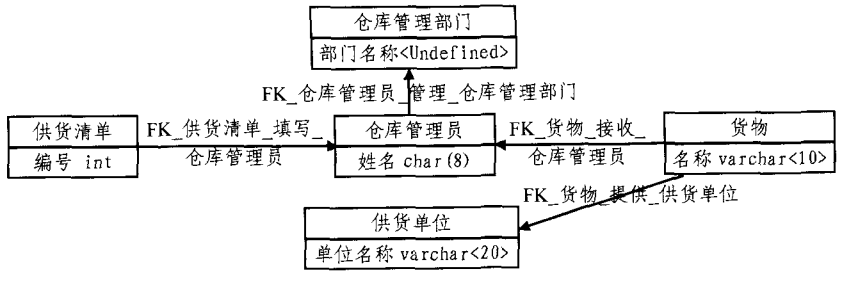

图 4 通过类图转换的物理模型

参 考 文 献

[1] 何玉洁. 数据库原理与应用教程[M]. 北京：机械工业出版社，2003
[2] 吴季，金贺中. UML 面向对象分析(第一版)[M]. 北京：北京航空航天大学出版社，2002
[3] 李兰涛，王忠民. 基于 UML 的软件产品线建模方法研究[J]. 微计算机信息，2006，22(10-3)：204-206
[4] Schmuller J. UML 基础、案例与应用[M]. 北京：人民邮电出版社，2005
[5] 白尚旺. Power Designer 软件分析设计技术[M]. 北京：电子工业出版社，2004
[6] 姜江，等. Power Designer 数据库系统分析设计与应用[M]. 北京：电子工业出版社，2004