

支持向量机在多类分类问题中的推广

刘志刚¹ 李德仁² 秦前清² 史文中³

¹(武汉大学遥感信息工程学院,武汉 430079).

²(武汉大学测绘遥感信息工程国家重点实验室,武汉 430079)

³(香港理工大学土地测量与地理咨询系地球信息研究中心,香港)

E-mail: weblzg@263.net

摘 要 支持向量机(SVMs)最初是用以解决两类分类问题,不能直接用于多类分类,如何有效地将其推广到多类分类问题是一个正在研究的问题。该文总结了现有主要的支持向量机多类分类算法,系统地比较了各算法的训练速度、分类速度和推广能力,并分析它们的不足和有待解决的问题。

关键词 支持向量机 多类分类 多类支持向量机

文章编号 1002-8331-(2004)07-0010-04 文献标识码 A 中图分类号 TP181

An Analytical Overview of Methods for Multi-category Support Vector Machines

Liu Zhigang¹ Li Deren² Qin Qianqing² Shi Wenzhong³

¹(School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079)

²(National Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079)

³(Advanced Research Centre for Spatial Information Technology, Department of Land Surveying and Geo-Informatics, the Hong Kong Polytechnic University, Hong Kong)

Abstract: Support Vector Machines (SVMs) is originally designed for binary classification. How to effectively extend it for multi-category classification is still an on-going research issue. This paper presents a general overview of existing representative methods for multi-category support vector machines and systematically compares their performances, including training speed, classification speed and generalization ability. The disadvantage and unsolved problem of these methods are also given.

Keywords: Support Vector Machines, Multi-category Classification, M-SVMs

1 前言

最初 SVM 是用以解决两类分类问题,不能直接用于多类分类,如何有效地将其推广到多类分类问题还是一个正在研究的问题。当前已经有许多算法将 SVMs 推广到多类分类问题,这些算法统称为“多类支持向量机”(Multi-category Support Vector Machines, M-SVMs)。它们可以大致分为两大类:

(1)通过某种方式构造一系列的两类分类器并将它们组合在一起来实现多类分类;

(2)将多个分类面的参数求解合并到一个最优化问题中,通过求解该最优化问题“一次性”地实现多类分类(如:文献[1][2])。

第二类方法尽管看起来简洁,但是在最优化问题求解过程中的变量远远多于第一类方法,训练速度不及第一类方法,而且在分类精度上也不占优^[9]。当训练样本数非常大时,这一问题更加突出。正因如此,第一类方法更为常用。该文仅对第一类方法进行讨论。

2 现有多类支持向量机算法

多类分类问题可以形式化地表述为:给定属于 k 类的 m 个训练样本 $(x_1, y_1), \dots, (x_m, y_m)$, 其中 $x_i \in R^n, i=1, \dots, m$, 且 $y_i \in \{1, \dots, k\}$, 要通过上述训练样本构造一个分类函数 f , 使对未知样本 x 进行分类时的错误概率(或者造成的损失)尽可能小。下面分别简单地介绍用 SVM 解决多类分类问题的现有主要方法。

2.1 1-v-r SVMs

用 SVM 解决多类分类问题最早的方法可能就是 1-v-R SVMs (One-versus-Rest) 算法^[4]。该方法依次用一个两类 SVM 分类器(后面简称分类器)将每一类与其它所有类别区分开来,得到 k 个分类函数。分类时将未知样本分类为具有最大分类函数数值的那类。

2.2 1-v-1 SVMs

该方法在每两类间训练一个分类器,因此对于一个 k 类问题,将有 $k(k-1)/2$ 个分类函数。当对一个未知样本进行分类时,每个分类器都对其类别进行判断,并为相应的类别“投上一

基金项目:利用空间数据挖掘技术进行新型影像目标识别和自动分类(编号:40023004);国家高新技术研究发展计划(编号:2001AA135081);香港 CRC 计划(编号:3_ZB40.1.34.9709)

作者简介:刘志刚(1976-),博士研究生,主要研究方向为模式识别与遥感影像分类。

10 2004.7 计算机工程与应用

票”,最后得票最多的类别即作为该未知样本的类别。这种策略称为“投票法”^[9]。采用上述方法的多类 SVMs, 简称为 1-v-1 SVMs(One-versus-One)算法^[9]。

2.3 有向无环图 SVMs

有向无环图多类 SVM 分类法(Directed Acyclic Graph SVMs, DAG-SVMs)^[10]在训练阶段和“1-1”投票一样,也要构造出每两类间的分类面,即有 $k(k-1)/2$ 个分类器。但是在分类阶段,该方法将所用分类器构造成为一种双向有向无环图(图 1): 包括 $k(k-1)/2$ 个节点和 k 个“叶”。其中每个节点为一个分类器,并与下一层的两个节点(或者叶)相连。当对一个未知样本进行分类时,首先从顶部的根节点开始,根据根节点的分类结果用下一层中的左节点或者右节点继续分类,直到达到底层某个叶为止,该叶所表示的类别即未知样本的类别。

表 1 8 类分类问题的 $L=5$ 纠错编码

类别号	码字				
	1	2	3	4	5
1	0	0	0	1	1
2	0	0	1	0	1
3	0	1	0	0	1
4	0	1	1	0	0
5	1	0	0	1	0
6	1	1	0	1	1
7	1	1	1	0	1
8	1	1	1	1	0

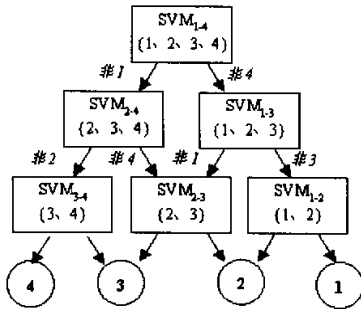


图 1 DAG-SVMs 分类

2.4 纠错编码 SVMs

Dietterich 提出对类别进行二进制编码将多类分类问题转化为多个两类分类问题^[9]。对于 k 类分类问题,给每个类别赋予一个长度为 L 的二进制编码,形成一个 k 行 L 列的码本(表 1)。对于其中第 $i(i=1, \dots, L)$ 列,将该位中码字为“0”的所有类别作为一类,其他码字为“1”的类别归为另一类,因此每个码位对应一个两类分类问题, k 类分类问题就转化为 L 个两类分类问题。采用具有纠错能力的编码对类别进行编码,并将 SVM 作为码位分类器。这种多类 SVM 分类方法被称为纠错编码支持向量机(ECC-SVM)^[10]。对于一个未知样本分类时, L 个 SVM 分类器的分类结果(0 或 1)构成一个编码 s ,计算码本内 k 个编码与 s 汉明距离,距离最小者所代表的类别即该测试样本所属类别。

对于多类分类问题一个好的纠错码应该满足如下条件^[9]:

- (1) 行 $r_i(i=1, \dots, k)$ 与行 $r_j(j=1, \dots, k; j \neq i)$ 之间不相关;
- (2) 列 $c_i(i=1, \dots, L)$ 与列 $c_j(j=1, \dots, L; j \neq i)$ 及其补 \bar{c}_j 之间不相关;

(3) 没有全为“0”或者全为“1”的列。

根据上述条件对于一个 k 类分类问题,纠错码长度 L 的取值范围是: $\lceil \log_{2k}, 2^{k-1}-1 \rceil$ 。

2.5 层次 SVMs

层次分类法首先将所有类别分成两个子类,再将子类进一步划分成两个次级子类,如此循环下去,直到得到一个单独的类别为止,这样就得到一个倒立的二叉分类树。该方法将原有的多类问题同样分解成了一系列的两类分类问题,其中两个子类间的分类函数采用 SVMs。文献[11,12]运用该方法将 SVMs 推广到多类分类问题,文献[12]称其为“基于决策树的多类支持向量机”。但本质而言,该方法是通过划分类别将原有问题转化成一列的两类问题,而通常所说的决策树(Decision Tree, DT)是对输入空间进行划分。为了避免产生理解上的混淆,这里将这类方法称为基于层次结构的多类支持向量机,简称层次支持向量机(Hierarchical SVMs, H-SVMs)。事实上在[13]中, SVMs 的思想已经被运用于 DT,但是由于该 DT 中每个节点只构造具有线性分类超平面,因此不能称为严格意义上的 SVMs 方法。

3 各种多类支持向量机算法的性能对比

3.1 训练速度

经过实验证明, SVM 的训练时间 T 大致满足如下关系^[14]:

$$T = cm^\gamma$$

其中 c 为一常数, γ 的大小与不同的分解算法有关。当采用 SMO 分解算法^[14]时, $\gamma \approx 2$ 。因此, SVM 的训练时间取决于参与训练的样本的数量多少。

由于 1-v-1 SVMs 和 ECC-SVMs 在训练其中每个分类函数时均需要所有的样本参与,因此它们的训练时间分别为 $T_{1-v-1} = kcm^\gamma$ 和 $T_{ECC} = Lcm^\gamma$, 其中 k 为类别数, L 为纠错编码的长度。

1-v-1 SVMs 和 DAG-SVMs 都是在每两个类别之间构造一个分类面,得到 $k(k-1)/2$ 个分类函数。假设每个类别的训练样本数相同,那么每个 SVM 的训练中将有 $2m/k$ 个训练样本参加。1-v-1 SVMs 和 DAG-SVMs 的整个训练时间为:

$$T_{1-v-1} = T_{DAG} = \frac{k(k-1)}{2} c \left(\frac{2m}{k} \right)^\gamma \approx 2^{k-1} ck^{2-\gamma} m^\gamma$$

当 $\gamma=2$ 时, 1-v-1 SVMs 和 DAG-SVMs 的训练时间将与类别数无关,仅相当于 1-v-1 SVMs 和 ECC-SVMs 中单个的分类函数训练时间的两倍。

H-SVMs 需要训练 $k-1$ 个分类函数。该方法的训练速度受层次结构的形态影响。首先考虑两种特殊的情况:

- (1) 从顶层开始,每一个包含多个类别的节点上的分类器只将一个类别与其它类别分开,这样的结构称为“偏态树”;
- (2) 从顶层开始,每一个包含多个类别的节点上的分类器都将其中类别均分成两类,这样的结构称为“正态树”。

假设每个类别的训练样本数相同,“偏态树”状的 H-SVMs 的训练总时间为:

$$T_H = \sum_{i=0}^{k-2} c \left(\frac{(k-i)m}{k} \right)^\gamma$$

如果 H-SVMs 的结构为“正态树”,且 $k=2^e$ (e 为大于 0 的正整数),其训练总时间为:

$$T_H = \sum_{i=0}^{e-1} 2^i c \left(\frac{m}{2^i} \right)^\gamma$$

显然“正态树”情况下的训练时间短于“偏态树”,具有其它

结构形状的 H-SVMs 的训练时间则处在两者之间。

为了更形象地对比上述几种算法的训练速度。假定:每类的训练样本数为 100, $\gamma=2, c=1, L=2*\log_2 k$, 分别计算出 $k=4, 8, 16, 32$ 时, 各种算法的相对训练时间。从图 2 可以看出, 在上述情况下, 1-v-r SVMs, DAG-SVMs 和正态树 H-SVMs 的训练速度相等且最快, 偏态树 H-SVMs 与 ECC-SVMs 的训练速度相当, 略慢于前三种算法。1-v-r SVMs 的训练速度随着类别数的增加急剧降低, 当类别数大于 16 时明显慢于其它算法。

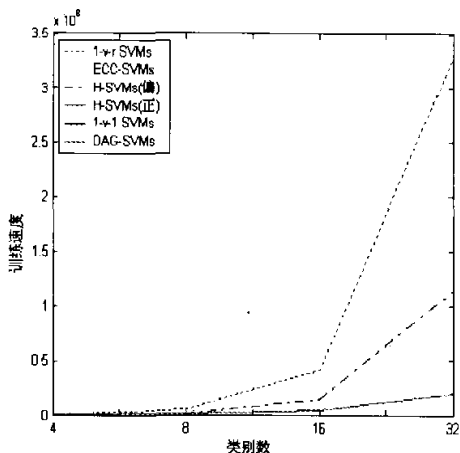


图2 各算法训练速度的比较

3.2 分类速度

假设某种 M-SVMs 算法在确定一个未知样本的类别时需要 b 个分类器 $\{f_i \in F\} (i=1, \dots, b)$, F 为该算法在训练阶段得到的分类器的总集。由于单个分类器所需的分类时间与其中支持向量的数量成正比, 因此 M-SVMs 算法对一个未知样本的分类

速度取决于所需分类器中支持向量的总数: $\sum_{i=1}^b S(f_i)$, 其中 $S(f_i)$ 表示第 i 个分类器中的支持向量数。由此可见, 一种 M-SVMs 算法的分类速度制约于两个因素: ①对单个未知样本分类所需分类器的数量; ②分类器其中支持向量的多少。

尽管 1-v-1 SVMs 和 DAG-SVMs 训练得到的两类分类器完全一致, 但是 DAG-SVMs 在对任何一个未知样本分类时只需运用 $k-1$ 个分类器, 1-v-1 SVMs 却需要所有的 $k(k-1)/2$ 个分类器。显然 DAG-SVMs 的分类速度快于 1-v-1 SVMs。

由于 DAG-SVMs 中每个分类器都只定义两个类别间的分类面, 1-v-r SVMs 中每个分类器需要定义某个类别与其他所有类别间的分类面, 因此总体而言后者的分类面较前者复杂, 当采用相同的核函数且其它训练条件相同的情况下, 后者的分类函数中的支持向量将多于前者。同时由于 1-v-r SVMs 分类时需要运用 k 个分类器, 较 DAG-SVMs 多 1, 因此一般而言 DAG-SVMs 的分类速度快于 1-v-r SVMs。

ECC-SVMs 分类过程中所需分类器的个数等于纠错编码的位数 L 。理论上说 ECC SVMs 的码位长只需 $\log_2 k$ 。但是此时码间最小汉明距离为 1, 编码将丧失纠错的功能。实际运用往往须取较长的码字以保持一定的最小码间汉明距, 通过编码的纠错能力得到较高的分类精度。从图 3 可以看出, 如果码字长取 $2\log_2 k$, 当类别数较多时, ECC-SVMs 分类时所需分类器的

数量明显少于 1-v-r SVMs, DAG-SVMs 和 1-v-1 SVMs, 但此时并不意味着 ECC-SVMs 的分类速度一定快于其它三种算法。因为 ECC-SVMs 中每个分类器是将对应码位为 0 的类同其它码位为 1 的类分开, 其中支持向量的数量与数据的可分性密切相关。因此对于某个具体问题, 码本设计恰当与否将直接影响到分类速度。

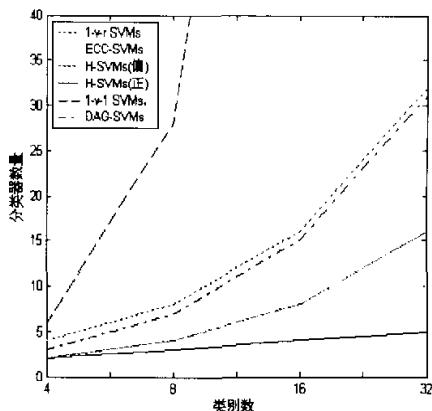


图3 各算法分类中分类器数量的比较

H-SVMs 的分类速度受其层次结构的形态影响。当结构形态为偏态树, 一个未知样本分类所需的分类器数量从 1 到 $k-1$ 不等, 平均 $k/2$, 少于 1-v-r SVMs (图 3)。此时, H-SVMs 与 1-v-r SVMs 类似。相同之处在于, 其中每个分类器都是将其一类同其它类分开; 不同之处在于, H-SVMs 中的分类器所处理的总类别数自上而下逐渐减少。因此分类器中的支持向量数一般少于 1-v-r SVMs。显然, 偏态树 H-SVMs 的分类速度将快于 1-v-r SVMs。

当 H-SVMs 的结构形态为正态树, 则对一个未知样本分类所需的分类器数量是确定的, 约 $\log_2 k$, 等于 ECC-SVMs 的理论最小值, 当类别数较多时, 远远少于其它算法。此时 H-SVMs 各分类器中支持向量的总量多少与其它算法的比较还有待进一步的研究。

3.3 结构选择与推广性能

由于 1-v-r SVMs 和 1-v-1 SVMs 对未知样本分类时需要使用所有的分类器, 且分类结果与分类器的使用顺序无关。因此它们的组合方式是唯一的。夏建涛将 1-v-r SVMs 作为 ECC-SVMs 的一种特例, 运用 fat-shattering 维和覆盖数 (Covering Number) 的概念给出了 1-v-r SVMs 的推广上界, 证明了 1-v-r SVMs 的推广上界与特征空间的维数无关, 可以通过增大其中所有 k 个 SVM 分类器的分类间隔来有效控制 1-v-r SVMs 的推广上界^[10]。然而当类别数较大时, 某一类的训练样本将大大少于其它类训练样本的总和, 这种训练样本间的不均衡将对精度产生影响^[11]。1-v-1 SVMs 分类时最大的缺点在于: 可能存在两个或多个类别得票数最多出现不可分的情况。另外, 尽管 1-v-1 SVMs 作为一种常用的多类分类方法在目前的许多运用中获得非常高的分类精度, 但至今还没有其推广性能理论描述。

k 类分类问题中, DAG-SVMs 中的有向无环图的组合方式有 $\prod_{i=0}^{k-3} [C_{k-i}^2]^i$ ($k \geq 3$) 种。但已有的实验证明组合的变化对分

类结果影响不大^[7]。因此可以认为 DAG-SVMs 中有向无环图的构建是任意的。文献[7]通过 VC 维的分析,给出了 DAG-SVMs 的推广上界,证明在高维空间中同样可以通过使 DAG-SVMs 中各分类器的分类间隙足够大来获得良好的推广性能。

ECC-SVMs 的推广上界被证明与特征空间维数无关,但同时取决于类别数、码长(L)、最小码间汉明距离(d)、编码的排列顺序以及分类间隙最宽的前 $[L-0.5*(d-1)]$ 个 SVM 的间隙宽度等多个影响因素^[10]。在具体问题中,如何确定码长、最小码间汉明距离生成编码,并寻找最优的排列方式以获取最佳的推广性能等问题依然有待解决。

对于一个 k 类问题, H-SVMs 所有可能的树状层次结构的数目为^[10]: $N_k = \prod_{i=1}^{k-1} 2^*i-1$, 其中 $k>1$ 。仅 N_{10} 就多达 34,459,425。

尽管根据前面对训练速度和分类速度的分析,笔者将倾向于使层次结构接近正态树,从而可以排除很多种组合方式,然而剩下的组合数依然是惊人的。那么不同的构造方式对分类精度有影响吗?下面以一种直观的方法来说明这个问题。假设有四类类别在特征空间中的分布如图 4 所示,采用两种正态树进行分类,并根据分类面的复杂程度,给出了 6 个分类器的分类精度(如图 4(b)、4(c))。假定某个测试集中四种类别各自有 100 个测试样本。第一种分类方案中一级 SVM_{1,2,3,4} 将 1,2 和 3,4 类样本分别准确地分到了左右次级节点, SVM_{1,2} 和 SVM_{3,4} 分别产生 10 个分类错误,因此总误差数为 20。第二种分类方案中一级 SVM_{1,3,2,4} 分别将 20 个 2,4 类样本错分到左侧次级节点, 20 个 1,3 类样本错分到右侧次级节点。尽管二级分类器 SVM_{1,2} 和 SVM_{3,4} 能够准确地将 1 类与 3 类样本分开, 2 类与 4 类样本分开,但是无法识别上级错分下来的样本。因此最终的总分类误差数为 40,高于第一种方案。该例说明:

- (1)不同的层次结构对 H-SVMs 的分类精度有影响;
- (2)H-SVMs 分类中存在自上而下的“误差累积”现象。

为了使 H-SVMs 具有最优的推广性能,必须使上层中两个子类之间的可分性尽可能强。为了构造合理的层次结构^[12],利用输入空间中各类样本间的欧氏距离和 Mahalanobis 距离逐层聚类(两类)。

这种方法的缺点在于:

- (1)对于高维数据基于距离的聚类的精度不够;
- (2)高维特征空间中数据的分布有别于输入空间,因此该方法不一定是最优的。

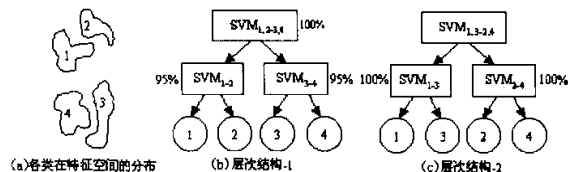


图 4 H-SVMs 分类示例

3.4 初步的结论

通过上述对比分析可以初步得出以下结论:

- (1)当需要对大量类别进行识别时,当前常用的 1-v-r SVMs 和 1-v-1 SVMs 的训练速度和分类速度将分别大大降低;
- (2)ECC-SVMs 的训练速度较 1-v-r SVMs 有明显改进,且当类别数大时仅需要少量的分类器。然而如何根据具体问题确定码本、选择排列顺序以达到最优的分类性能依然有待研究;

(3)DAG-SVMs 简单易行,具有理想的训练速度,分类速度略慢,对于一般规模的多类分类问题是一种有效的方法;

(4)H-SVMs 当层次结构的形态接近于正态树时,将具有理想的训练速度,同时分类时仅需要非常少量的分类器。如何根据具体问题构造合理的正态树,克服“误差累积问题”提高分类精度,减少分类器中支持向量的总量,提高分类速度,以实现最优的分类性能还有待研究。

4 结束语

许多的实际运用中(如:遥感影像分类,网页内容分类等)都需要解决大类别的分类问题,随着支持向量机在这些领域的深入运用,如何有效地将该方法推广到多类分类问题中必将引起人们广泛的兴趣。该文总结了现有主要的支持向量机多类分类算法,系统地比较了各算法的性能,并分析了它们的不足和有待解决的问题,希望对这方面的研究发挥一点承前启后的作用。(收稿日期:2003 年 12 月)

参考文献

- 1.Weston J, Watkins C. Multi-class Support Vector Machines[R]. Technical Report CSD-TR-98-04, Royal Holloway, University of London, Department of Computer Science, 1998
- 2.Crammer K, Singer Y. Ultraconservative Online Algorithms for Multi-class Problems[C]. In: Proc of the 14th Annual Conf on Computational Learning Theory, 2001
- 3.Hsu C-W, Lin C-JA. Comparison of Methods for Multi-Class Support Vector Machines[J]. IEEE Transactions on Neural Networks, 2002; (13): 415-425
- 4.Bottou L, Cortes C, Denker et al. Comparison of Classifier Methods: A Case Study in Handwriting Digit Recognition[C]. In: International Conference on Pattern Recognition, IEEE Computer Society Press, 1994; 77-87
- 5.Friedman J H. Another Approach to Polychotomous Classification[R]. Technical report, Stanford University, From: <http://www-stat.stanford.edu/reports/friedman/poly.ps.z>, 1996
- 6.Kreßel U. Pairwise Classification and Support Vector Machines[C]. In: B Scholkopf, C J C Burges, A J Smola eds. Advances in Kernel Methods: Support Vector Learning, The MIT Press, Cambridge, MA, 1999; 255-268
- 7.Platt J C, Cristianini N, Shawe-Taylor J. Large Margin DAGs for Multiclass Classification. In Advances in Neural Information Processing Systems, MIT Press, 2000; 12: 547-553
- 8.Dietterich T G, Bakiri G. Solving Multiclass Learning Problem via Error-Correcting Output Codes[J]. Journal of Artificial Intelligence Research, 1995; (2): 263-286
- 9.Jörg Kindermann, Edda Leopold, Gerhard Paass. Multi-class Classification with Error Correcting Codes[C]. In: E Leopold, M Kirsten eds. Treffen der GI-Fachgruppe 1.1.3, Maschinelles Lernen, 2000 GMD Report 114
- 10.夏建涛. 基于机器学习的高维多光谱数据分类[D]. 博士学位论文. 西北工业大学, 2002
- 11.Azimi-Sadjadi M R, Zekavat S A. Cloud Classification Using Support Vector Machines[C]. In: Proc of the 2000 IEEE Geoscience and Remote Sensing Symposium (IGRASS 2000), Honolulu, Hawaii, 2000-07; (2): 669-671
- 12.Fumitake Takahashi, Shigeo Abe. Decision-Tree-Based Multiclass (下转 65 页)

在 G1 调用了 G2.mtd,其最坏执行时间设为 30。若执行时间超过 30 后,迁移 t1 引发,开始调用副版本执行。若在 30 时间内执行完,则存在两种情况,一种是正常结束(normal),则需对执行结果进行可接受测试(Acceptance Test,简写 AT),如果结果是正确的(correct),则继续执行或到达目标位置(goalplace),如果不正确(incorrect),则开始调用副版本执行;另一种出现运行错误(runerror),提前退出,开始调用副版本执行。如果副版本正常结束(normal),则继续执行或到达目标位置(goalplace),如果运行错误(runerror)或超时(timeout),则向 errorplace 位置发送令牌,errorplace 将错误令牌返回给系统,通知运行错误,且无法恢复。

4 TCG-Net 模拟器

通过建立系统 TCG-Net 模型,将系统规格为多个可以执行的 TCG-Net 对象。对于单个 TCG-Net 对象来说,可以利用现有的 Petri 网的分析技术进行分析,而对整个应用来说,要验证时间约束是否能够满足,需要建立 TCG-Net 模拟器。如图 8 所示。

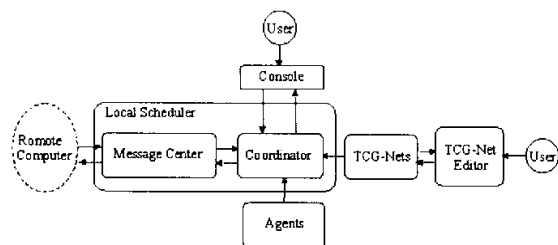


图 8 TCG-Net 模拟器结构

TCG-Net 模型可以在一个分布式环境上运行,在每个结点上有如图 8 所示的模拟器。用户(User)可以通过编辑器(TCG-Net Editor)来建立和修改 TCG-Net 对象;这些 TCG-Net 对象由局部调度器(Local scheduler)进行调用,当有新的 TCG-Net 执行时,调度器为其分配一个代理(Agent);用户可以通过控制台(Console)来控制 TCG-Net 的运行和控制结果的输出;TCG-Net 之间通过消息中心(Message Center)进行通信,或与本地 TCG-Net 进行通信,与其他位于远程计算机(Remote Computer)上的 TCG-Net 进行通信。通信即通过传送令牌来进行信息交互。每个代理的执行都是独立的,不受其他代理的控制。协调器(Coordinator)是局部调度器的主要模块,用来协调 TCG-Nets 的执行和同用户的交互。

5 结束语

Deng 等提出的 G-Net 模型被用于分布式信息系统的建模^[13],该模型采用模块化的设计,支持分布式执行,但是由于缺乏时间属性,而不适于实时系统的建模和分析。该文针对以珠

程为调度单位的分布式实时系统,对 G-Net 模型进行了拓展,引入了时间和容错机制,给出了 TCG-Net 模型。该模型提供了主/副版本容错机制,支持模块化的设计和分布式执行,从而有利于大型可靠分布式实时系统的构建。同时,通过 TCG-Net 模拟器,可以对所建模系统的规格进行分析。

(收稿日期:2003 年 4 月)

参考文献

1. TianLong Gu, Parisa A Bahri. A survey of petri net applications in batch processes[J]. Computers in Industry, 2002; 47: 99~111
2. Tasi J J P, Yang S J. Timing constraint petri nets and their application to schedulability analysis of real-time system specifications[J]. IEEE Trans Software Engineering, 1995; 21(1): 32~49
3. Verhoosel J P C, Hammer D K, Luit E Y et al. A model for scheduling object-based distributed real-time systems[J]. Real-Time Systems, 1995; 8(1): 5~34
4. 董荣胜, 赵岭忠, 蔡国水等. 基于对象的分布式实时系统调度模型研究[J]. 计算机研究与发展, 2002; 39(11): 1464~1470
5. 赵岭忠, 董荣胜, 蔡国水等. 基于对象的分布式实时系统约束的一致性研究[J]. 计算机工程与应用, 2002; 38(15): 60~62
6. 蔡国水, 占天龙, 董荣胜等. 实时对象模型及其可预测调度研究[J]. 计算机研究与发展, 2000; 37(1): 117~121
7. Yeong-Jia, Daniel Mosse, Shi-Kuo Chang. An object-based model for dependable real-time distributed systems[C]. In: The 2nd IEEE Workshop on Object-Oriented Real-Time Dependable Systems, Laguna Beach, California, 1996
8. I. Santhoshkumar, G. Manimaran, C. Siva ram Murthy. A pre-run-time scheduling algorithm for object-based distributed real-time systems[C]. In: The 1997 Joint Workshop on Parallel and Distributed Real-Time Systems, Geneva, Switzerland, 1997: 260~267
9. I. Santhoshkumar, G. Manimaran, C. Siva Ram Murthy. A pre-run-time scheduling algorithm for object-based distributed real-time systems[J]. Journal of Systems Architecture, 1999; 45(14): 1169~1188
10. Bastide R. Approaches in unifying Petri nets and the object-oriented approach[C]. In: The 1st workshop on object-oriented programming and models of Concurrency, Torino, Italy, 1995
11. Eugenio Battiston, Alfredo Chizzoni, Fiorella De Cindio. Modeling a cooperative development environment with CLOWN[C]. In: The 1st workshop on object-oriented programming and models of Concurrency, Torino, Italy, 1995
12. Charles A Lakos. A Cooperative editor for hierarchical diagrams; an object petri net model[C]. In: Workshop on object-oriented programming and models of Concurrency, Torino, Italy, 1995
13. Yi Deng, S K Chang, Jorge C A de Figueiredo et al. Integrating software engineering methods and petri nets for the specification and prototyping of complex information systems[C]. In: The 14th international conference on application and theory of petri nets, Chicago, 1993

(上接 13 页)

Support Vector Machines. From: <http://frenchblue.scitec.kobe-u.ac.jp/~abe/pdf/iconip02-takahashi.pdf>, 2002

13. Bennett K, Cristianini N, Shawe-Taylor J et al. Enlarging the Margin in Perceptron Decision Trees[J]. Machine Learning, 2000; (41): 295~313
14. Platt J. Fast Training of Support Vector Machines Using Sequential Minimal Optimization[C]. In: B Scholkopf, C J C Burges, A J Smola eds.

Advances in Kernel Methods; Support Vector Learning, MIT Press, 1999: 185~208

15. Nathalie Japkowicz, Shaju Stephen. The Class Imbalance Problem, A Systematic Study. From: <http://www.site.uottawa.ca/school/publications/techrep/2001/index.shtml>
16. Alex Bailey. Class-dependent features and multicategory classification[D]. PhD thesis. From navy.mil/csf/papers/baileyphd.pdf, 2001