表1是近几年来所公布的各主要处理器一览表[29]，从表中可看出，几乎所有的处理器都具有并行发送多条指令的能力，显然今后的处理器也将包含这一能力。另外，还可看到，提高处理器的性能有两条主要途径：一是依靠指令的动态调度和推测式执行来获取高性能；二是使用静态调度并竭力提高处理器的速度来获得较高性能。但目前还看不出那种途径具有更多的优越性。

四、指令级并行研究的展望

从目前的形势看，开发指令级并行性的处理器中，由于VLIW结构的处理器不具有目标代码的兼容性，超标量处理器就占绝对优势。今后，VLIW结构的处理器如果仍不能解决这一主要问题，也不会有多大起色。最近几年出现的TTA[30]，DS[31]，PEWs[32]，MISC[33]，Multiscalar[34]等结构，要么仅仅是一种概念型结构，要么缺乏目标代码的兼容性，因此也很难在通用处理器市场上与超标量处理器一争高低，最终的命运只能与VLIW相同。

然而，超标量处理器要想开发出更多的指令级并行性，获取更高的IPC(Instructions Per Cycle)，必需解决如下三个主要问题：及时提供指令，及时提供数据，增大程序中的并行性。

为了及时提供所需的指令，一个可能的途径是借助编译技术增大基本块，这等价于减少转移指令，使指令流被中断的机会减少。另一种途径是进一步优化转移处理策略，比如说转移预测的准确率。对超标量处理器性能影响最严重的是转移指令，由于5%的误预测率都会带来较高的性能损失，因此如何在允许的条件下提高转移预测率以及探索别的处理转移指令的方法，将是今后的主要方向[35,36]。

为了及时提供所需的操作数，必需有较好的数据预取策略[37]和更为有效且命中率更高的Cache结构和存储系统。

增大程序中并行性，最主要的是增大基本块的大小。途径之一是采用新的编译技术如超块技术[38,39]等来增大程序的基本块。途径之二是采取如多条件码域[40]和条件式执行等更为有效的转移处理策略尽量减少转移指令的数量。最后多条转移指令的同时预测也是一个非常有希望的方向。

除此之外，由于多存储体系的使用，处理器的通讯延迟有随处理器性能提高而增大的趋势，且INTERNET日益普及，使得与外界的交流，通讯越来越重要，因此，在提高处理器性能的同时，减少通讯延迟将是未来的发展方向。

（参考文献共39篇略）

（上接第35页）

$$F1(X)=\frac{X-1}{0-1}GG(1,1)+\frac{X-0}{1-0}GG(1,2)=-2(X-1)+3X=X+2$$

$$F2(X)=\frac{X-3}{2-3}GG(2,3)+\frac{X-2}{3-2}GG(2,4)=-3(X-3)+4(X-2)=X+1$$

所以 $F(X)=(X+2)+(X-0)(X-1)(X+1)=X^3+2$，即完全拟合。

四、讨论

由实例可见，本文的方法对多项式的拟合是完美的，因此对于能用多项式拟合的数值都可以较好地完成，而且有算法简单，易于实现的特点。但仍需要考虑以下问题：

- 可能需要依据某种技术从轴上对D进一步处理以确定更精确的函数式；
- 在简单情况下可用比较的方式确定更精确的函数式；
- 在不要求拟合函数的光滑性时，可直接采用区间划分所得到的线性分段函数作为目标函数。
- 在数值有误差的情况下是否可用比较的方式确定更精确的函数式？即该方法的鲁棒性。
- 递推过程的 Δ_i 应采用适当精度，该如何选取？
- 对本方法的误差考虑；
- 它对概率与统计学中的参数估计是否有参考坐标？

参考文献

[1]杨青云，《数据处理方法》，冶金工业出版社，1990

[2]徐萃薇，《计算方法引论》，高等教育出版社，1982

[3]李爱中，模型发现和智能决策支持系统工具的研究，哈工大博士论文，1991

[4]Lin, XiaoFen & Ungar, Lyle, Inventing Theoretical terms in Inductive Learning of Function Search and Constructive Methods, Methodologies for Intellingent Systems, 1989.4

计算机科学 1997 Vol. 24№. 4

具有鲁棒性的知识获取方法*)

A Robust Knowledge Acquirement Method

张师超 覃振兴

(广西师范大学数学与计算机系 桂林 541004)

摘 要 本文讨论知识获取的数学机理,建立了一种基于区间划分的不确定性数据的知识获取方法。

关键词 知识获取,区间划分,不确定推理,归纳学习

在科学研究中,归纳是获取知识的一种行之有效的最重要方法,用以从观察到的现象和数据中找出内在的规律。归纳通常基于传统的统计学的数值分析法。然而,统计学中的拟合意味着求解预定形式的函数中的参数,必须靠专家凭经验给出并经多次试探后选取一个相对较好的函数。

这种归纳技术对一些低维数或较有规律的数据进行处理是行之有效的,但对于一些比较复杂而维数较高的数据或显式函数不一定存在的数据,其处理能力将大大降低,只能较盲目地假定某个特殊函数。可见这种技术运用到知识获取中尚不成熟,需经过适当的修正和增强。不过,这并不影响统计技术作为归纳方法的基础。

国内也有一些学者在这方面做了大量的工作,李爱中等将一类启发信息和代数方程引入递归过程。但正如李爱中等指出的那样,递归归纳方法先天带有岐义性和不确定性。若采用回溯技术进行处理,则复杂度将大大增加,甚至出现组合爆炸。

另一方面,我们手头的资料和数据均由实际观察得到,难免具有一定的误差和不确定性,但递归归纳方法要求实验数据很准确,而且所发现的目标函数必须能递归定义,这是采用递归技术的一个致命的弱点。对于基数据的函数是特殊的或隐式的,或者实验数据含有误差,或者是不确定环境下的数据诸情况,难以施展递归技术。

本文建立一种新的知识获取机制,能确定任意一组给定数据的目标函数的线性逼近式。该学习模型基于区间划分对源数据进行预处理以确定关键点,同时用逐步增次法确定多项式次数,尽可能地减少误差数据对函数形式的影响,对领域专家的依赖也大大减少,所以具有一定的鲁棒性和自适应性。

一、知识获取模型

1. 区间划分

给定一个复变元集合 $V=\{y,x'_1,x'_2,\cdots,x'_{m0}\}$ 和关于这些变元的一组数据 $D=\{d_1,\cdots,d_n\}$,其中 $d_i=(y_i,x'_{i1},x'_{i2},\cdots,x'_{im0})$,它满足,当 $x'_{1j}=x'_{i1},x'_{2j}=x'_{i2},\cdots,x'_{m0j}=x'_{im0}$ 时,$y=y_i$。我们的目标就是发现一个函数 f,使得 $y=f(x'_1,\cdots,x'_{m0})$ 尽量拟合上述数据集合 D 中的整体规律。为简化问题的描述,不妨设 m0=1,即 $V=\{y,x'\}$,这样可以通过二维平面直观地理解本文中学习模型的实质。

先将 D 在 x' 方向上进行划分,得到 k 组数据集合 $D_1,D_2,\cdots,D_k$,它们满足:

i)对 $\forall(y_{ij},s_{ij})\in D_i, x'_{ij}\in[\underline{x}'_i,\overline{x}'_i]$,其中 $\underline{x}'_i$ 和 $\overline{x}'_i$ 分别为 D_i 中 x'_{ij} 的上确界和下确界,j=1,2,…,n_i。这里,n_i 为 D_i 中数据个数且 $n=\sum n_i$;

ii)若 D_i 与 D_j 相邻时,不妨设 j=i+1,则 $[x'_i,x'_i]\cap[x'_j,x'_j]$ 中仅含有点 $\underline{x}'_i=\overline{x}'_j$;

iii)(y,x') 在 $[\underline{x}'_i,\overline{x}'_i]$ 上线性相关,即存在 a_i,b_i 使得 $y_{ij}=a_i+b_ix'_{ij}, x'_{ij}\in[\underline{x}'_i,\overline{x}'_i]$。

为合理地划分数据集 D,我们采用统计技术中的相关系数 r 作为划分标准。即

$$r_i=\left(\sum_{j=1}^{s}(x'_{ij}-xx'_i)(y_{ij}-yy_i)\right)\Big/\sqrt{\left(\sum(x'_{ij}-xx'_i)^2\right)\left(\sum(y_{ij}-yy_i)^2\right)}$$

其中,s 为目前所涉及的数据个数,$xx'_i=(\sum x'_{ij})/$

*)本文得到国家自然科学基金和国家 863 计划的资助

$s, yy_i=(\sum y_y)/s$。

设 R 为一个阈值，若 $r_i \geqslant R$ 时，则认为 D_i 中的数据是线性相关的。

2. 选取代表点

经过这样的划分后，各组数据集内数据基本保持线性相关，而端点则为变化较大的转折点，选取它们作为代表点可以较好地体现整个数据集的特性。即取 $X=\bar{x}'_1, \bar{x}'_1, \bar{x}'_2, \cdots, \bar{x}'_{k-1}, x'_m$ 这 k+1 个点的数据作为代表点，不妨设 m=k+1，分别命名为 $x_1, x_2, \cdots, x_m$，它们的观测值分别命名为 G1(X1)，G1(X2)，…，$G1(X_m)$，得到一个抽样数据集合 U，写成如下形式：

X	x_1,	x_2,	…,	x_m
Y	G1(X1),	G1(X2),	…,	$G1(X_m)$

3. 构造拟合函数

不妨设得到的数据集为：$U=\{(g1(x), x) | x=x_1, x_2, \cdots, x_m\}$，要拟合得到的代数表达式为 Y=F(X)，N 为要拟合的步数。其拟合方法如下：

第一步，考虑数值：

X	x_1,	x_2,	…,	x_m
Y	G1(X1),	G1(X2),	…,	$G1(X_m)$

令 $F(X)=F1(X)+G2(X)(X-X_1)(X-X_2)$，其中

$$F1(X)=\frac{(X-X_2)}{(X_1-X_2)}G1(X1)+\frac{(X-X_1)}{(X_2-X_1)}G1(X2)$$

G2(X)被视为余项，对于已知点有：

$$G2(X_i)=\frac{G1(X_i)-F1(X_i)}{(X_i-X_1)(X_i-X_2)}, i=3,4,\cdots,m。$$

对于数值 G2(X3)，G2(X4)，…，$G2(X_m)$，若 $\Delta_i=\sum|G2(X_i)|<\varepsilon(\varepsilon\geqslant 0$ 为专家给定的较小的数值)，或数列长 $m-4\leqslant 0$ 则停止计算，且取 F(X)=F1(X)，余项 G2(X)可忽略不计；否则对 G2(X)进一步求精，方法是把 $G2(X_3), G2(X_4), \cdots, G2(X_m)$ 看作新的数值列进行拟合。即对如下数值进行处理：

X	x_3,	x_4,	…,	x_m
Y	G2(X3),	G2(X4),	…,	$G2(X_m)$

令 $G2(X)=F2(X)+G3(X)(X-X_3)(X-X_4)$，其中

$$F2(X)=\frac{(X-X_4)}{(X_3-X_4)}G2(X_3)+\frac{(X-X_3)}{(X_4-X_3)}G2(X_4)$$

G3(X)被视为余项，对于已知点有：

$$G3(X_i)=\frac{G2(X_i)-F2(X_i)}{(X_i-X_3)(X_i-X_4)}, i=5,6,\cdots,m。$$

若 $\Delta_i=\sum|G3(X_i)|<\varepsilon(\varepsilon\geqslant 0$ 为专家给出的较小的数值)，或数列长 $m-5\leqslant 0$ 则停止计算，且取 F(X)=F1(X)+G2(X)(X−X1)(X−X2)，余项 G3(X)可忽略不计；否则对 G3(X)进一步求精，即对如下数值进行类似的似合：

X	x_5,	x_6,	…,	x_m
Y	G3(X5),	G3(X6),	…,	G3(Xm)

如上依此类推，经过 $N(N\leqslant[m/2])$ 次求精即可停止，最后的 Δ_N 视为误差，$G_N(X)$忽略不计。于是：

$$\begin{aligned}F(X)&=F1(X)+G2(X)(X-X_1)(X-X_2)\\&=F1(X)+[F2(X)+G3(X)(X-X_3)(X-X_4)](X-X_1)(X-X_2)\\&\cdots\\&=F1(X)+\sum_{i=1}^{n}[F_{i+1}(X)(\sum_{j=1}^{n}(X-X_j))]\end{aligned}$$

即为所求的拟合函数，是一个次数不超过 2N+1 的多项式。

二、算法

假定源数据为 $V=\{x_i, y_i | i=1,2,\cdots,m\}$，设 m0=[m/2]+1。

1 定义一维数组 X[1-m]，Y[1-m]

//用于存放源数据//

二维数组 GG[1-m,1-m]，F[1-m,1-m]//用 FF[i,j]，GG[i,j]存放 $F_i(x_j)$和 $G_i(x_i)$的值//

2. N←1，X[i]←x_i；Y[i]←y_i，GG[i,j]←0；FF[i,j]←0；i,j=1,2,…,m

3 for j=1 to m do GG[1,j]←Y[j]

//赋原始数列初值//

4 若 2N+1≥m 则转 7

5 $FF[N,J]\leftarrow[\frac{X(j)-X(2N)}{X(2N-1)-X(2N)}GG(N,2N-1)+\frac{X(j)-X(2N-1)}{X(2N)-X(2N-1)}GG(N,2N)]$

$GG(N+1,J)\leftarrow\frac{GG(N,j)-FF(N,j)}{(X(j)-X(2N-1))(X(j)-X(2N))}$

其中，j=2N+1，…，m。

6 若 $\sum_{j=2N+1}^{m}|GG(N+1,J)|<\varepsilon$ 转 7，否则 N←N+1，转 5

7 输出结果

三、实例

下面，我们采用上述方法来拟合函数 $Y=X^5+2$ 在 X=0,1,…,10 点的 11 组数值，我们取阈值为 1，于是这些数值就划出了十个区间[0,1]，[1,2]，…，[9,10]。其拟合过程和结果如附录一所示(略)。显然有：

(下转第 24 页)