

GP 算法中适应度函数的光滑拟合与调整参数方法研究¹⁾

王四春^{1,2} 张泰山¹ 殷志云² 张楚文²

¹⁾(中南大学信息科学与工程学院 长沙 410083)

²⁾(湖南商学院信息科学研究所 长沙 410205)

(E-mail: wscabcbank@sina.com)

摘要 研究了遗传程序设计 (GP) 算法中适应度函数的光滑拟合问题, 结合 LAMs (Linear association memories) 方法和 HJ (Hook 和 Jeevs) 方法两种方法, 估计 GP 树数值权值, 以减少 GP 树适应度值评价的计算代价. 光滑拟合的好坏关键取决于调整参数的选择. 提出了一种选择调整参数的新方法, 同时, 给出了两个数学例子, 并与广义交叉实验 B-样条函数仿真比较验证.

关键词 自遗传程序设计 (GP), 适应度函数, 光滑拟合, 调整参数
中图分类号 TP301

Research on Smooth Fitting and Regularization Parameter under the Fitness Function of Genetic Programming Algorithm

WANG Si-Chun^{1,2} ZHANG Tai-Shan¹ YIN Zhi-Yun² ZHANG Chu-Wen²

¹⁾(The Information Science and Engineering College of Central South University, Changsha 410083)

²⁾(The Information Science Institute of Hunan Business College, Changsha 410205)

(E-mail: wscabcbank@sina.com)

Abstract This paper deals with the smooth fitting problem under the genetic programming (GP) algorithm. To reduce the computational cost required for evaluating the fitness value of GP trees, numerical weights of GP trees are estimated by adopting both linear associative memories (LAM) and the Hook and Jeevs (HJ) method. The quality of smooth fitting is critically dependent on the choice of the regularization parameter. So, we present a novel method for choosing the regularization parameter. Two numerical examples are given with comparison to the generalized cross-validation (GCV) B-splines.

Key words Genetic programming, fitness function, smooth fitting, regularization parameter

1 引言

考虑一个单变量逼近函数式,

$$y(t) = \mu(t) + \varepsilon(t) \quad (1)$$

1) 国家自然科学基金 (60373083), 湖南省自然科学基金 (05JJ40118), 湖南省教育厅 (04C313) 资助
Supported by National Natural Science Foundation of P. R. China (60373083), the Natural Science Foundation of Hunan Province (05JJ40118), Hunan Education Office (04C313)

收稿日期 2004-7-4 收修改稿日期 2006-1-9

Received July 4, 2004; in revised form January 9, 2006

其中 $\mu(t)$ 为未知函数, $\varepsilon(t)$ 为噪音函数, 数学期望 $E(\varepsilon(t)) = 0$, 方差是 σ^2 . 观测函数 $y(t)$, $t = t_0, t_1, \dots, t_n$ ($t_0 < t_1 < \dots < t_n$), 构造学习集 $L\{(t_i, y_i) | y_i \equiv y(t_i)\}_{i=0, \dots, n}$, 搜索 $\mu(t)$ 的估计量 $\bar{\mu}(t)$. 目前, 有许多 $\bar{\mu}(t)$ 估计方法, 如插值多项式法、局部加权回归法、样条插值和神经网络法等. 本文试图使用遗传程序设计^[1,2]的方法, 搜索 $\mu(t)$ 的最优估计值. 为此, 首先必须解决两个问题. 首先, GP 树上的结点数数值权值能够以有效的计算方式被估计, 选择一个最优 GP 树, 通过树结点数数值参数或权值进行评价. 尽管一些 GP 树是 $\mu(t)$ 的一些好的候选解, 但它们的数值权值愈低, 获得的级别愈低, 将在下一次进化过程中被淘汰. 原始的 GP 算法主要集中在动态修改 GP 树的结构, 缺乏数值权值估计技术. 目前, 国外主要采用 Sharman K C, Watson A H, Gray G J, Bettenhausen K D, McKay B, Weinbrenner T, Yang Y S. 等人研究的非线性优化评价方法, 由于种群通常是由数百或数千个 GP 树构成的, 要计算每个 GP 树的数值权值非常耗时. 本文采用 LAMs 和 HJ 搜索法的混合法, 将极大地减少计算代价.

其次, 由于 L 受到噪音的影响, GP 算法适应度评价函数应该含有调整参数项, 适应度函数应该包含两项, 一项用来考虑 GP 树对 L 的适应性, 另一项代表光滑适应度的调节项. 这里, 最重要的任务是选择一个合适的参数, 使之在给定的 L 数据附近获得一个解, 同时, 尽可能地光滑. 如参数太小, 解结果不稳定, 如参数太大, 解结果过分光滑. 目前, 国内外在 GP 算法数值权值估计中, 有许多流行的参数选择方法, 但难度均较大^[3~7]. 本文设计了一种新颖的简单启发式方法. 该方法能够有效地计算并选择优良的 GP 树. 文献报道情况, 在 GP 算法中, 采用调整参数方法研究光滑拟合问题, 国内外尚未见到.

本文讨论了调整适应度函数和 GP 树数值权值估计的有效方式; 提出了选择调整参数的新方法, 描述了 GP 算法中的光滑拟合问题概况; 给出了一个数学实例.

2 GP 树适应度函数的调整

遗传程序设计算法是在遗传算法的基础上发展起来的, 组成群体的个体采用一种动态树状结构表示计算机程序. 本文的候选模型为 $\mu(t)$, GP 树的结点由终结点 (Terminals)、原始函数 (Primitive functions) 与运算符组成. 其中终结点也称叶结点, 它是将问题分级为子问题后最基本的解的成分. 这种树状的层与结点均是可变化的. 终结点是问题的原始变量, 根结点和中间结点统称为内部结点, 它们则是组合这些原始变量的函数. 在进化过程中, GP 树结构通过遗传算子动态变化. 遗传操作包括复制、交叉和变异, 每个算子应用于被选择的个体, 复制是根据它们的适应度值复制个体, 交叉是从当前群体中, 根据适应度值挑选两个个体的不同部件 (如子树、子程序、子路径、子公式等) 重新组合产生新个体. 变异是由程序随机产生一棵新的子树, 以代替被变异概率选中结点以下的原有子树部分.

GP 在求解函数逼近时, 是一种非常有用的工具. GP 树结构在进化过程中, 动态变化以获得最优的函数形式. 为了改善 GP 树结构的质量, 连接在 GP 树结点上的数值权值估计总是向着进一步极小化适应度函数方向发展. 如果学习样本受到噪音破坏, 问题将变得难于解决. GP 树应该约束学习样本同它的光滑函数逐渐逼近. 构造光滑函数的最有效方法是采用调整适应度评价函数中的调整项 (光滑约束条件). 适应度函数采用 $f = E + \lambda\theta$ 形式, E 代表 GP 树适应逼近度, θ 是光滑估计量, λ 是调整参数. 这里只是尽可能地通过使用适当的调整参数的近似数值权值估计来促使 GP 树作为一个光滑函数与它的期望相符. 事实上, 仅仅使用遗传算子很难期望产生这样一个 GP 树. 本节讨论调整适应度函数、选择函数集和数值权值的问题.

2.1 适应度函数的调整方法

我们有

$$f = f_A + \lambda\theta \quad (2)$$

(2) 给出适应度函数和它的调整项, 其中 $f_A = \frac{1}{n+1} \sum_{i=0}^n (y(t_i) - \bar{u}(t_i))^2$, θ 是 GP 树函数的

光滑估计量, λ 是调整参数. 如果选择几何向量作为 θ , 称为零 - 状态 Tikhonov 调整^[4], 则 GP 树在进化过程中变成一个高度非线性函数, 有时 GP 树的值呈现非预期的行为, 在 t_i 到 t_{i+1} 之间的某一点上突然变得很大或很小. 这种行为不能在 $t_i (i = 0, \dots, n)$ 某一点上测定到, 因此, 零 - 状态 Tikhonov 调整或基于算子的微分调整不能充分有效地消除多余的震动或在 t_i 到 t_{i+1} 之间的大峰现象^[3,4]. 采用式 (3) 的光滑函数测定 θ ,

$$\theta = \frac{1}{n+1} \left\{ \sum_{i=1}^n \bar{\mu}(t_i)^2 \right\} (|\bar{\mu}'(t_0)| + |\bar{\mu}'(t_n)| + \bar{\mu}'_L) \quad (3)$$

其中 $\bar{\mu}'_L$ 是 $\bar{\mu}'(t) = \frac{d\mu}{dt}$ 在 t_0 到 t_n 之间的曲线长度, 能够非常有效地在 $t_i \sim t_{i+1}$ 之间获得无任何大峰的光滑 GP 树.

为使进化过程不断进行, 应用式 (2) 中给出的适应度函数, 通过优化计算, 测定 λ 和估计权值, 计算种群中的所有 GP 树. 这种处理方法计算相当费时. 所以, 我们使用一种简单的方式, 用 f_A 的值代替 GP 树的真实适应度值, 当使用遗传算子时, 不同时考虑光滑处理. 求 f_A 的值, LAMs 方法^[1~3] 是一种非常有效的计算方法. 从群体中随机选择的 GP 树, 只有个别好的 GP 树具有 f_A 的极小值, 然后采用 HJ 搜索方法^[4,5] 选择 λ 参数, 求适应度函数 f 的极小值.

2.2 终点集和函数集

在本文中, 终点集只含有一个变量 t , 随机大小小于 1. t 有一个数值权值, t 的加权值代表结点的实际输出. 经过仔细选择的函数使得 GP 树产生稳定的解. 选择函数集 F

$$F = \{g_1, g_2, g_3, g_4, +, -, *, \div\} \quad (4)$$

式中, $g_1 = \frac{a_1}{1+e-b_1(x-c_1)^2}$, $g_2 = \frac{a_2}{e[(x-b_2)/c_2]^2}$, $g_3 = \frac{a_3}{b_3} \sqrt{b_3^2 + (x-c_3)^2}$, $g_4 = a_4(b_4x-c_4)^2$, g_i 通常是一个前向神经网络中使用的 S 形函数, g_2 是高斯核心函数, g_3 是径向基函数 (RBF), g_i 含有三个受 ‘+’, ‘-’, 和 ‘*’ 算术算子决定的数值权值 (a_i, b_i 和 c_i), 同样, 它们中的每一个都含有一个输出结点的加权数值权值. 此外, 本文使用的函数集中, 可以用到其它反映特定问题的函数, 如: 周期函数 $\mu(t)$, 三角函数等.

2.3 GP 树的适应权值

计算 GP 树权值过程由两个步骤组成. Step 1: 采用 LAMs 方法, 使用仅含 f_A 的适应度函数粗略计算种群中的 GP 树的估计权值^[2,3]. Step 2: 从种群中选择几个具有 f_A 最小值的 GP 树, 用包含 f_A 和的适应度函数 f 评价它们的权值, 采用 HJ 搜索方法, 通过极小化 f 搜索最优权值. HJ 搜索方法与其它方法相比, 快速而且有效, HJ 搜索方法主要特点是采用局部模式而不是采用空间搜索^[4,5]. 值得注意的是, 当 HJ 搜索方法完成时, GP 树的实际适应度值又变成 f_A , 而遗传算子又被下一代使用.

3 调整参数的选择方法

目前, 有许多选择的方法, 包括误差法 (DP)、交叉验证法 (CV)、复合余数和光滑算子法 (CRESO)^[5]、L-曲线法^[6,7]、零-交叉法 (ZC)^[8]. DP 方法需要了解方差 σ^2 知识, 这对实际应用不利. CV 方法: 虽不需要掌握方差 σ^2 知识, 但如学习样本不大, CV 误差估计较大. CRESO 方法: 采用的最小绝对值作调整参数, 产生函数 $\theta(\lambda) + 2\lambda \frac{d}{d\lambda} \theta(\lambda)$ 的相对极大

值. 由于 HJ 方法不能产生 $\theta(\lambda)$ 的连续值, 所以几乎不可能计算 $\frac{d}{d\lambda}\theta(\lambda)$, 同时我们也很难找到这样一个 λ 值, 因为 λ 值的微小变化, 会引起 HJ 方法产生不同的 $\theta(\lambda)$ 值. L- 曲线方法: 调整参数的标准是 λ 的选择, 相当于在 L- 曲线的“拐点”有极大曲率, 即条件上需要在 f_A 和 θ 的拐点同时获得一个折中合理的数据拟合和光滑, L- 曲线方法存在许多弊端, 一是曲线的不连续性, 二是 λ 值的轻微变化产生的结果差异较大. ZC 方法: 一种基于实验证据的启发式方法, 即找到一个使 η 最小或为零的最佳参数 λ , 即 $(\lambda) = |f_A(\lambda) - \lambda\theta(\lambda)|$. ZC 方法的优点是容易计算 η , 但通过我们的实验观测, ZC 方法的最大缺点是在许多情况下, 容易使 GP 树过光滑.

为了弥补目前调整参数选择方法的不足, 本文采用了一种基于实验结果的简单启发式方法, λ 的选择通过一个简单的经验公式 (5) 计算, 即

$$\lambda = \frac{1}{s\theta}\tau^\alpha \quad (5)$$

式中, $\tau = \frac{1}{2n} \sum_{i=1}^n \{y(t_{i+1}) - y(t_i) - (\bar{\mu}(t_{i+1}) - \bar{\mu}(t_i))\}^2$, $s_\theta = \theta$, α 通过式 (6) 计算.

$$\alpha = \begin{cases} \max\{100(\max(f_A, \tau) - \min(f_A, \tau)), \tau\} + \xi, & \text{if } f_A > \tau \\ \xi, & \text{其它} \end{cases} \quad (6)$$

每个 τ , s_θ 和 α 值在 HJ 方法开始估计权值前被确定. 计算 λ 同计算 GP 树权值的 HJ 方法描述如下:

HJ 方法形式描述: $HJ(w^{-*}, w^{-0}, r, f)$, 开始点为 w^{-0} , 最佳点 w^{-*} 通过求最小参数值函数 f 寻找, 直到 r 达到最大迭代数.

1) 所有 GP 树的权值通过 LAMs 方法粗略计算. 从群体中选择最好的 GP 树, 对它们中的每一个, 计算 τ^1, s_θ^1 , 符号上标代表迭代数. 设 $\xi = 0.95$, 计算 α^1 , 使用 $\tau^1, s_\theta^1, \alpha^1$ 通过公式 (4) 计算 λ^1 . 这样, GP 树的权值向量 w^{-0} 就通过 LAMs 方法找到. 接着开始 $HJ(w^{-1}, w^{-0}, r, f^1)$, ($f^1 = f_A + \lambda^1\theta$), 获得权值向量 w^{-1} .

2) 按照步骤 1), 计算 $\tau^2, s_\theta^2, \alpha^2$ 和 λ^2 , 除 $\xi = 0.8$ 外, 计算 α^2 , 开始 $HJ(w^{-2}, w^{-1}, f, f^2)$.

3) 直到当前迭代数 k 达到最大允许的数 h , 重复步骤 2), 这里, 假如 k 是奇数, 设 $\xi = 0.95$, 假如 k 是偶数, 设 $\xi = 0.8$. 如 h 是偶数, 且 $|f_A^{-h} - f_A^{-h-1}|/f_A^{-h} > 0.1$, 步骤 (2) 用 $\xi = 0.95$ 重复多次. 这里, 当 $HJ(w^{-h}, w^{-h-1}, r, f^h)$ 时, f_A^{-h} 相当 f_A 的值.

本文设计 $r = 1000$, $h = 4$, 因为 r 太小, GP 树的权值不能充分估计.

4 GP 算法的光滑拟合

GP 算法中的光滑拟合研究步骤与方法算法描述如下.

算法 1.

1) 随机产生一个种群或通过遗传算子从前代产生一个新的种群, 所有树的初始权值按 Koza 方法设为标准的 GP 树, 值为 1. 群体中每个树, 增加随机数, 大小小于 0.5, 整个树的权值, 使用 LAMs 方法进行估计. LAMs 方法设法求 f_A 的最小值, 即 f_A 的值代表整个树的适应度值.

2) 选择几个好的 GP 树, $\{\bar{\mu}_i\}_{i=1, \dots, \theta}$, 获得较小的 f_A 值的树. θ 代表选择树的数目, 本文选择 5 个好树, 每个被选择的 GP 树, 其权值 $\vec{w}_i^{LAM} = (w_{i,j}^{LAM})_{j=0, \dots, p_i}$ 采用 HJ 方法, 通过调整参数 λ 的选择得到. 这里, \vec{w}_i^{LAM} 是 π - 空间向量. 由于 HJ 方法对开始点非常敏感, 轻微的不同开始点可能引起不同的结果, 为避免这个问题, 需要选择几个不同的

开始点, 以得到好的结果. 本文通过增加随机数到初始权值 \vec{w}_i^{LAM} , LAMs 方法已经发现 $\vec{w}_i^{LAM} = (w_{i,j}^{LAM} + \delta_{i,j})_{j=0,\dots,p_i}$, $\delta_{i,j}$ 是随机数, 大小小于 $0.5|w_{i,j}^{LAM}|$. 从 \vec{w}_i^{LAM} 中, HJ 搜索开始, 重复 ρ 次, 本文为了减少计算代价, 设为 5 次. 由于通过 f 最小估计权值, 不容易产生最好的 GP 树 μ 估计, 所以我们设计一种选择适当权值的启发式方法, 算法描述如下.

算法 2.

对每个 $\vec{\mu}'$, 权值向量估计从第 k 个点开始, $\vec{w}_{i:k}^{HJ}$ ($k = 1, \dots, \rho$), 为简便起见, 定义两个符号, 即 $f_{A(i:k)}$ 和 $\theta_{i:k}$, $f_{A(i:k)}$ 代表 $\vec{w}_{i:k}^{HJ}$ 同 $\vec{\mu}_i$ 的 f_A , $\theta_{i:k}$ 代表 $\vec{\mu}_i$ 的 θ 同 $\vec{w}_{i:k}^{HJ}$.

1) 初始 $\vec{w}_{i:*}^{HJ}$ 为 $\vec{w}_{i:1}^{HJ}$, $\vec{w}_{i:*}^{HJ}$ 代表最佳权值.

2) 假如 $f_{A(i:*)} + \zeta\theta_{i:*} > f_{A(i:k)} + \zeta\theta_{i:k}$ ($k \geq 2$), $\zeta = \frac{\{\frac{1}{2}(f_{A(i:*)} + f_{A(i:k)})\}^{1.3}}{\frac{1}{2}(\theta_{i:*} + \theta_{i:k})}$, 那么 $\vec{w}_{i:1}^{HJ}$ 变

成 $\vec{w}_{i:*}^{HJ}$.

3) 重复 2) $\rho - 1$ 次, $k = k + 1$, 得到最后 $\vec{w}_{i:*}^{HJ}$.

$\vec{\mu}_i$ 的适应度值变成 $f_{A(i:*)}$, 遗传操作算子基于这个值在下一代被应用. 从 $\{\vec{\mu}_i\}_{i=1,\dots,\theta}$ 中选择最好的树 $\vec{\mu}_*$, 每个树的权值向量为 $\vec{w}_{i:*}^{HJ}$, 定义两个符号 $f_{A:i}$ 和 θ_i , $f_{A:i}$ 代表 $\vec{w}_{i:k}^{HJ}$ 同 $\vec{\mu}_i$ 的 f_A , θ_i 代表 $\vec{\mu}_i$ 的 θ 同 $\vec{w}_{i:k}^{HJ}$.

4) 选择 $\vec{\mu}_1$ 作为 $\vec{\mu}_*$.

5) 如果 $f_{A:i} + \zeta\theta_i > f_{A:i} + \zeta\theta_i$ ($i \geq 2$), $\zeta = \frac{\{\frac{1}{2}(f_{A:i} + f_{A:i})\}^{1.3}}{\frac{1}{2}(\theta_* + \theta_i)}$ 那么, $\vec{\mu}_*$ 变成 $\vec{\mu}_i$.

6) 重复步骤 5) $\theta - 1$ 次, $i = i + 1$, 找到最后好树 $\vec{\mu}_*$.

3) 通过遗传算子产生一个新的种群, 重复步骤 1) 和步骤 2), 通过步骤 4)~6) 选择当前代最好的树 $\vec{\mu}_C$ 和上一代最好的树 $\vec{\mu}_P$, 计算 $\vec{\mu}_C$ 是否比 $\vec{\mu}_P$ 更好.

如果 $f_{A:P} + \zeta\theta_P > f_{A:C} + \zeta\theta_C$ ($k \geq 2$), 那么 $\vec{\mu}_C$ 被选择, 否则 $\vec{\mu}_P$ 被保留, $\vec{\mu}_C$ 被淘汰.

这里, $\zeta = \frac{\{\frac{1}{2}(f_{A:P} + f_{A:C})\}^{1.2}}{\frac{1}{2}(\theta_P + \theta_C)}$.

4) 重复步骤 3), 直到达到允许的代数.

影响好的 GP 树选择的重要因素是 ρ 和 θ . 当它们的值增加时, 找到好的 GP 树的可能性愈大, 但却增加计算代价.

5 数值仿真及结果分析

在本节中, 研究一个铃形函数. 为便于比较, 引进 GCV B- 样条函数结果. 本文还采用了最著名的 GCVSPL 软件计算包^[9]. 学习集设为 $\{(t_i, \mu(t_i) + \varepsilon_i) | t_i = a + i(b-a)/n\}_{i=0,\dots,n}$, a 和 b 分别为开始点和结束点, $n = 50$, ε_i 为接近于 0 高斯噪音, 方差 σ^2 . 准备学习集, 考虑两种情况, $\sigma = 0.1$ 和 $\sigma = 0.05$. 尽管高斯噪音的方差几乎相同, 但由于噪音有时严重干扰原函数的特征, 造成 GP 结果或 GCV B- 样条函数结果却很不相同, 所以, 每种情况准备 5 个不同的学习集数据.

一个学习集的拟合度好坏由 $MSE_L^{\vec{\mu}} = \frac{1}{n+1} \sum_{i=1}^n \{y_i - \vec{\mu}(t_i)\}^2$ 决定, $\vec{\mu}(t)$ 的好坏被定义

为 $MSE_T^{\vec{\mu}} = \frac{1}{n+1} \sum_{i=1}^n \{\mu(t_i) - \vec{\mu}(t_i)\}^2$, 来自噪音学习样本 L 的导数函数 $\mu'(t)$ 的好坏被定义

为 $MSE_T^{\vec{\mu}'} = \frac{1}{n+1} \sum_{i=1}^n \{\mu'(t_i) - \vec{\mu}'(t_i)\}^2$.

铃形函数如式 (7) 定义,

$$\mu(t) = 1.5\{\exp(-30(t-0.25)^2) + \sin^2(\pi(t-0.2))\}, \quad 0 \leq t \leq 1 \quad (7)$$

由于铃形函数学习样本的噪音在 $t = 0$ 附近可能破坏原函数的特征, 所以容易导致 $\mu'(t)$ 的估计产生很差的结果. 这将严重影响 GCV B-样条函数的行为特征. 实验结果如表 1 所示: 一方面, GCV B-样条的值在许多情况下很大且差异悬殊, 另一方面, 有些情况下, $\bar{\mu}_{SP1}(t)$ 的结果比 $\bar{\mu}_{GP}(t)$ 略显好, 如第 1 和第 3 种情况 ($\sigma = 0.05$)、第 2 和第 5 种情况 ($\sigma = 0.1$). 对 $\sigma = 0.1$, 图①表示最好的 $\bar{\mu}_{GP}(t)$ 和它相应的 $\bar{\mu}'_{GP}(t)$ (情况 4), 图②表示最差的情况(情况 2). 观察两个图形, 在靠近 $t = 0$ (在最差条件下 $t = 0.25$) 区域结果显示同原函数比较图形有些区别, 这是因为该区域学习样本受到噪音干扰所致.

表 1 铃形函数 GP 和 GCV B-样条函数结果
Table 1 The results of GP and GCV B-spline for the bell-shaped function

Mean	σ	GCV B-spline							# 结点	
		GCV Error	M_L^{μ}	M_T^{μ}	$M_T^{\mu'}$	M_L^{μ}	M_T^{μ}	$M_T^{\mu'}$		
Noise: $\varepsilon \sim N(0, 0.05^2)$										
1	4.41E-3	5.10E-2	3.26E-3	2.17E-3	3.64E-4	2.20E-2	2.29E-3	2.92E-4	1.01E-2	23
2	-3.01E-3	5.02E-2	2.93E-3	1.64E-3	5.76E-4	5.96E-2	2.29E-3	6.73E-4	1.90E-2	23
3	4.41E-4	5.08E-2	3.23E-3	2.13E-3	4.40E-4	3.50E-2	2.56E-3	4.00E-4	1.63E-2	23
4	3.63E-3	4.98E-2	3.17E-3	9.24E-4	1.11E-3	1.86E-1	2.18E-3	6.37E-4	1.04E-3	23
5	6.11E-3	4.98E-2	2.53E-3	1.08E-3	1.11E-3	9.62E-2	1.80E-3	8.06E-4	3.19E-2	23
Noise: $\varepsilon \sim N(0, 0.05^2)$										
2	-4.58E-3	1.11E-1	1.43E-2	8.41E-3	2.34E-3	1.23E-1	1.07E-2	7.93E-4	2.52E-2	23
3	-9.93E-3	1.07E-1	1.48E-2	1.04E-2	1.22E-3	5.05E-2	1.14E-2	1.69E-3	1.75E-2	23
4	1.15E-2	1.11E-1	1.47E-2	1.04E-2	2.88E-3	7.11E-2	1.00E-2	1.02E-3	2.91E-2	26
5	-3.83E-3	1.01E-1	1.24E-2	8.56E-3	1.27E-3	7.06E-2	9.72E-3	1.01E-3	2.08E-2	23
6	9.62E-3	1.04E-1	1.33E-2	9.53E-3	2.19E-3	8.25E-2	9.99E-3	1.93E-3	1.78E-2	9

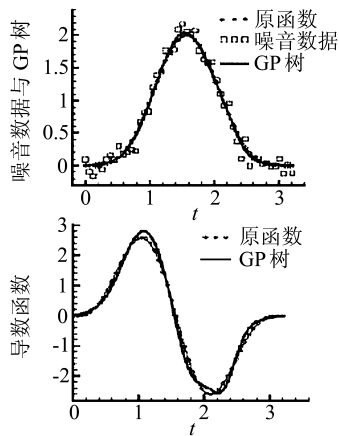


图 1 铃形函数最好的 GP 树 $\bar{\mu}_{GP}(t)$ 和它相应的 $\bar{\mu}'_{GP}(t)$ 结果, 当 $\sigma = 0.1$, MSE_T^{μ} 和 $MSE_T^{\mu'}$ 的值分别是 $7.93E-4$ 和 $2.52E-2$

Fig. 1 The results of the best GP tree and $\bar{\mu}_{GP}(t)$ its correspondent $\bar{\mu}'_{GP}(t)$ for the bell-shaped function, when $\sigma = 0.1$. The values of MSE_T^{μ} and $MSE_T^{\mu'}$ are $7.93E-4$ and $2.52E-2$, respectively

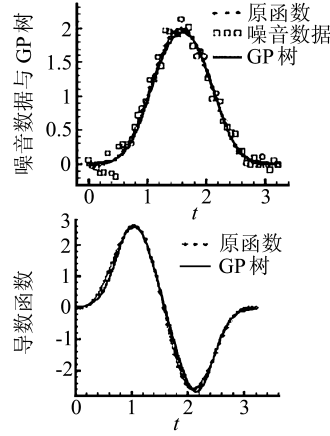


图 2 铃形函数最差的 GP 树 $\bar{\mu}_{GP}(t)$ 和它相应的 $\bar{\mu}'_{GP}(t)$ 结果, 当 $\sigma = 0.1$, MSE_T^{μ} 和 $MSE_T^{\mu'}$ 的值分别是 $1.93E-3$ 和 $1.78E-2$

Fig. 2 The results of the worst GP tree $\bar{\mu}_{GP}(t)$ and its correspondent $\bar{\mu}'_{GP}(t)$ for the bell-shaped function, when $\sigma = 0.1$. The values of MSE_T^{μ} and $MSE_T^{\mu'}$ are $1.93E-3$ and $1.78E-2$, respectively

6 结论

本文提出了基于 GP 的光滑拟合方法. 该方法成功的关键是通过调整参数的选择快速估计 GP 树的权值. 为了估计权值, 我们引入了 LAMs 方法首先粗略估计种群整个树的权

值,以减少计算代价,然后采用 HJ 方法通过调整参数选择搜索潜在的 $\bar{\mu}'(t)$ 好的候选解,精确计算树的权值. 调整参数选择依赖启发式方法,这样克服了传统的方法如 CV 和 L-曲线方法,不能快速被应用的不足. 同样,通过调整参数的选择可以获得较好的 GP 树.

数值仿真对比了 GP 同 B-样条的性能,结果显示,在许多情况下,估计 $\mu(t)$ 和 $\bar{\mu}'(t)$, GP 胜过 B-样条.

该方法的不足之处在于,调整参数的选择和好的 GP 树选择标准是一种启发式的,就象 CRESO 和 ZC 方法,没有严格的数学基础. 数值仿真结果也显示,在某些情况下,选择的调整参数不是最佳的,只是可以接受,参数值的细微变化能够得到比较好的结果,这有待于进一步研究.

References

- 1 Lin J C, Durand D M. Weighted linear associative memory approach to nonlinear parameter estimation. *Journal of Optimization Theory and Applications*, 1996, **90**(1): 139~159
- 2 Yeun Y S, Lee K H, Yang Y S. Function approximations by coupling neural networks and genetic programming trees with oblique decision trees. *Artificial Intelligence in Engineering*, 1999, **13**(2): 223~239
- 3 Yeun Y S, Su J C, Yang Y S. Function approximation by superimposing genetic programming trees with application to engineering problems. *Information Sciences*, 2000, **122**(2,4): 128~131
- 4 Craven P, Wahba G. Smoothing noisy data with spline functions. *Numerische Mathematik*, 2001, **31**(2): 377~403
- 5 Colli-Franzone P, Guerri L, Taccardi B, Viganotti C. Finite element approximation of regularized solutions of the inverse potential problem of electrocardiography and applications to experimental data. *Alcolo*, 2001, XXII(1): 78~81
- 6 Hansen P C. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Review*, 2002, **34**(4): 561~580
- 7 Hansen and P C, O'Leary D P. The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM Journal of Science Statistical Computing*, 2002, **14**(1): 1487~1503
- 8 Peter R J, Ramesh M G. A new method for regularization parameter determination in the inverse problem of electrocardiography. *IEEE Transactions on Biomedical Engineering*, 2002, **441**(8): 19~39
- 9 Woltring H. A Fortran package for generalized cross-validation spline smoothing and differentiation. *Advantage Engineering Software*, 2003, **8**(2): 104~113

王四春 副教授, 在读博士研究生, 研究方向: 计算机人工智能.

(WANG Si-Chun Associate professor and Ph. D. candidate. His research interests include artificial intelligence and automatic programming.)

张泰山 教授, 博士生导师, 研究方向: 智能控制、人工生命科学及先进控制理论.

(ZHANG Tai-Shan Professor. His research interests include intelligent control, artificial lifetime science, and advancing control theory.)

殷志云 教授, 博士生导师, 研究方向为进化计算理论和计算机代数.

(YIN Zhi-Yun Professor. His research interests include evolutionary computation theory and computer algebra.)

张楚文 副教授, 研究方向: 系统工程和智能决策.

(ZHANG Chu-Wen Associate professor. Her research interests include system engineering and intelligent decision.)