

粒子群优化的广义T-S模糊模型参数学习方法

周欣然^{1,2}, 滕召胜¹, 易 钊¹

(1. 湖南大学电气与信息工程学院 长沙 410082; 2. 中南大学信息科学与工程学院 长沙 410075)

【摘要】提出了一种基于粒子群优化的广义T-S模糊模型参数学习方法。该方法用离散二进制微粒位置表示模型的结构参数,用普通微粒位置表示模型规则中模糊集隶属函数的参数;这两种微粒位置联合体构成一个模型完整的前件参数集。每一学习循环分两步,前一步用粒子群进化迭代调整所有前件参数,后一步用正交最小二乘法估计后件参数。该方法不需任何先验知识,运算量小,能产生紧凑的模糊模型。非线性动态系统模糊建模的数字仿真说明了该方法的有效性。

关键词 模糊建模; 广义T-S模糊模型; 正交最小二乘法; 粒子群优化
中图分类号 TP273 **文献标识码** A

Parameters Learning Approach for Generalized Takagi-Sugeno Fuzzy Model Using Particle Swarm Optimization

ZHOU Xin-ran^{1,2}, TENG Zhao-sheng¹, and YI Zhao¹

(1. College of Electrical and Information Engineering, Hunan University Changsha 410082;

2. School of Information Science and Engineering, Central South University Changsha 410075)

Abstract A parameters learning approach for generalized takagi-sugeno (T-S) fuzzy model is proposed in this paper on the base of analysis of generalized T-S Fuzzy model. The structural parameters of the approach are denoted by the position of discrete binary particles and the parameters of membership function in the approach are denoted by the position of ordinary particles. The combination of positions of the two kind of particles composes complete premise parameters set of a model. A learning cycle consists of two phases: first, all reasoning parameters are adjusted by evolutionary iteration of particle swarm; second, all consequent parameters are estimated through orthogonal least square error algorithm. The method requests scarcely any previous information about objects, take less calculating time, and is able to obtain compact fuzzy model. The simulation result shows the validity of the approach.

Key words fuzzy modeling; generalized takagi-sugeno (T-S) fuzzy model; orthogonal least Square error; particle swarm optimization (PSO)

文献[1]设计了一种参数化模糊集隶属函数,改变隶属函数中的一个参数,隶属函数便能自适应地近似三角型、梯型或高斯型等。采用这种参数化隶属函数的T-S模糊模型^[2](以下简称T-S模型)为广义T-S模糊模型(以下简称广义T-S模型),与T-S模型相比,该模型建模时可得到规则数较少的模糊模型^[3]。训练广义T-S模型可以采用基于梯度的各种优化方法,但用这类优化方法训练容易陷入目标函数的局部极值,而且要先用其他方法确定模型中的规则数。文献[3]提出了基于遗传算法的训练方法,模型前件和后件参数同时采用遗传算法优化,并成功训练出紧凑的模型。微粒群优化算法(PSO)是继遗传算法之后的一种进化算法,该算法概念简单,实现容易,

运算量较少;在解决一些典型函数优化问题时,能够取得比遗传算法更好的优化效果^[4]。自基本PSO^[5-6]和离散二进制PSO^[7]出现后,又出现了众多改进形式的PSO。

本文尝试用PSO训练广义T-S模型,并且考虑到在模型前件参数选定后,后件参数的确定是线性估计问题,因此将前、后件参数分开训练。用离散二进制PSO和普通PSO分别优化广义T-S模型的结构和模糊规则中模糊集隶属函数的参数;采用正交最小二乘法估计后件参数。前、后件参数分开训练是为了减少待PSO优化的目标函数中的变量个数,以提高训练模型的速度。本文采用简单的多次启动法^[8]提高PSO全局寻优概率,在获得的多个次优解中再

收稿日期: 2007-10-20; 修回日期: 2008-02-02

基金项目: 国家星火计划项目(2003EA770007); 湖南省杰出青年基金(01JZY2101)

作者简介: 周欣然(1976-), 男, 博士生, 主要从事智能检测、智能信息处理方面的研究。

挑选出一个相对好的解。

1 广义T-S模型的描述

对于具有 n 输入单输出、模糊规则数为 M 的T-S模型, 模糊规则形式为^[2]:

$$R_l: \text{if } x_1 \text{ is } F_1^l, x_2 \text{ is } F_2^l, \dots, x_n \text{ is } F_n^l \\ \text{then } y^l = c_0^l + c_1^l x_1 + \dots + c_n^l x_n \quad (1)$$

式中 $x_j (1 \leq j \leq n)$ 为输入变量; $1 \leq l \leq M$, M 为模型的规则数; F_j^l 为规则 l 中 x_j 所隶属的模糊集合; c_j^l 为后件参数; y^l 为模型根据此条规则得出的相应输出。对于输入向量 $\mathbf{X} = (x_1, x_2, \dots, x_n)$, 模糊推理时前件连接采用乘积、去模糊化采用重心法时, 模型的输出定义为:

$$y = \frac{\sum_{l=1}^M y^l \prod_{j=1}^n u_{F_j^l}(x_j)}{\sum_{l=1}^M \prod_{j=1}^n u_{F_j^l}(x_j)} \quad (2)$$

模型的模糊集合隶属函数 $u_F(x)$ 通常取三角形、梯形、高斯型或其他指数型, 这些隶属函数的特点是一旦给定为某种类型就不能改变其大致形状, 文献[1]设计了一种参数化隶属函数:

$$u_F(x) = \exp\left(-\left|\frac{x-\beta}{\alpha}\right|^\gamma\right) \quad \alpha > 0, \beta \in R, \gamma \geq 0 \quad (3)$$

通过改变参数 γ , $u_F(x)$ 便能自适应地近似以上所提到的各种隶属函数, 采用这种参数化隶属函数的模糊模型称为广义T-S模糊模型^[1], 其输出为:

$$y = \frac{\sum_{l=1}^M \left[y^l \prod_{j=1}^n u_{F_j^l}(x_j) \right]}{\sum_{l=1}^M \prod_{j=1}^n u_{F_j^l}(x_j)} = \frac{\sum_{l=1}^M \left[\left(c_0^l + \sum_{j=1}^n c_j^l x_j \right) \prod_{j=1}^n \exp\left(-\left|\frac{x_j - \beta_j^l}{\alpha_j^l}\right|^{\gamma_j^l}\right) \right]}{\sum_{l=1}^M \left[\prod_{j=1}^n \exp\left(-\left|\frac{x_j - \beta_j^l}{\alpha_j^l}\right|^{\gamma_j^l}\right) \right]} \quad (4)$$

2 微粒群算法

2.1 基本PSO算法

基本PSO^[5-6]算法将每个个体(意指鸟等)看作是D维连续搜索空间中的一个没有体积的微粒(点), 每个微粒在搜索空间中以一定的速度飞行, 这个速度

根据它本身的飞行经验以及同伴的飞行经验进行动态调整, 直至群体中的个体飞行到对环境适应度高的区域。

求目标函数 $f(\mathbf{X}) = f(x_1, x_2, \dots, x_d, \dots, x_D)$ 的最优值时, 将目标函数映射为个体对环境适应度计算式, 第 i 个微粒所处点的位置坐标 $\mathbf{X}_i = (x_{i1}, x_{i2}, \dots, x_{id}, \dots, x_{iD})$ 对应于一个可能解向量 \mathbf{X} 。微粒 i 的速度记为 $\mathbf{V}_i = (v_{i1}, v_{i2}, \dots, v_{id}, \dots, v_{iD})$; 迄今为止($0 \sim t$ 时刻之间)微粒 i 所经历过的最好位置记为: $\mathbf{P}_i^t = (p_{i1}^t, p_{i2}^t, \dots, p_{id}^t, \dots, p_{iD}^t)$, 或称为微粒 i 个体历史最好位置; 所有微粒经历过的最佳位置中的最好位置称为全局历史最好位置, 记为 $\mathbf{P}_g^t = (p_{g1}^t, p_{g2}^t, \dots, p_{gd}^t, \dots, p_{gD}^t)$; 微粒 i 的飞行速度和位置更新公式为:

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 r_{id}^t (p_{id}^t - x_{id}^t) + c_2 R_{id}^t (p_{gd}^t - x_{id}^t) \quad (5)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (6)$$

式中 $i=1, 2, \dots, s$, s 为群中微粒个数; t 为时刻或代龄; ω 为惯性权值; c_1 和 c_2 为加速系数, c_1 表示微粒个体的认知能力, c_2 表示微粒个体向社会(群体)学习的能力, 它们均取正常数; r_{id}^t 和 R_{id}^t 为 $[0,1]$ 中的随机数。

若求 $f(\mathbf{X})$ 最优值为求其最小值, 则微粒 i 经历过的最好位置更新公式为:

$$\mathbf{P}_i^{t+1} = \begin{cases} \mathbf{P}_i^t & f(\mathbf{X}_i^{t+1}) > f(\mathbf{P}_i^t) \\ \mathbf{X}_i^{t+1} & \text{其他} \end{cases} \quad (7)$$

整个微粒群迄今为止经历过的最好位置为:

$$\mathbf{P}_g^t = \underset{p}{\operatorname{argmin}} (f(\mathbf{P}_i^t)) \quad i=1, 2, \dots, s \quad (8)$$

如有必要, 可对微粒速度和位置做限幅处理。

2.2 离散二进制PSO算法

离散二进制PSO^[7]限制位置坐标分量只在 $\{0,1\}$ 中取值。该算法中, 微粒的速度先按式(5)更新, 再利用sigmoid函数如式(9), 先将速度压缩到 $(0, 1)$ 区间。

$$\operatorname{sig}(v_{id}^t) = \frac{1}{1 + \exp(-v_{id}^t)} \quad (9)$$

而位置则更新为:

$$x_{id}^t = \begin{cases} 0 & \rho_{id}^t \geq \operatorname{sig}(v_{id}^t) \\ 1 & \text{其他} \end{cases} \quad (10)$$

式中 ρ_{id}^t 为 $[0,1]$ 中的随机数。当 $v_{id}^t < -10$ 时, $\operatorname{sig}(v_{id}^t) \approx 0$, x_{id}^t 必然为0; 当 $v_{id}^t > 10$ 时, $\operatorname{sig}(v_{id}^t) \approx 1$, x_{id}^t 必然为1, 可见此时 x_{id}^t 的取值失去随机性。为了让 x_{id}^t 有一定的概率, 取0或1, 文献[9]建议限制 v_{id}^t 在 ± 4 之间。

2.3 PSO算法的参数

为了让PSO算法中的微粒在进化初始阶段有较大的速度惯性, 以便于微粒在较大范围寻优, 而让微粒在进化结尾阶段有较小的速度惯性, 以便于算法快速收敛, 本文将参数 ω 设计成指数函数:

$$\omega(t) = 1.5\exp(-5t / \text{maxiter}) \quad (11)$$

式中 maxiter 为设定的最大进化迭代代数。本文让 c_1 和 c_2 按文献[10]提出的策略变化:

$$\begin{cases} c_1(t) = (c_{1i} - c_{1f})\left(\frac{\text{maxiter} - t}{\text{maxiter}}\right) + c_{1f} \\ c_2(t) = (c_{2i} - c_{2f})\left(\frac{\text{maxiter} - t}{\text{maxiter}}\right) + c_{2f} \end{cases} \quad (12)$$

式中 c_{1i} 和 c_{1f} 分别为 c_1 始值和终值, 可分别取2.5和0.5; c_{2i} 和 c_{2f} 分别为 c_2 始值和终值, 可分别取0.5和2.5。 c_1 在进化初期有较大的值, 便于微粒个体各自独立寻优, 在进化后期有较小的值, 弱化微粒个体独立寻优功能; c_2 则线性递增, 各微粒在进化后期倾向于向最优微粒学习, 加速算法收敛。

$$\begin{bmatrix} \delta_i^1 & \alpha_{i,1}^1 & \alpha_{i,2}^1 & \cdots & \alpha_{i,j}^1 & \cdots & \alpha_{i,n}^1 & \beta_{i,1}^1 & \cdots & \beta_{i,j}^1 & \cdots & \beta_{i,n}^1 & \gamma_{i,1}^1 & \cdots & \gamma_{i,j}^1 & \cdots & \gamma_{i,n}^1 \\ \delta_i^2 & \alpha_{i,1}^2 & \alpha_{i,2}^2 & \cdots & \alpha_{i,j}^2 & \cdots & \alpha_{i,n}^2 & \beta_{i,1}^2 & \cdots & \beta_{i,j}^2 & \cdots & \beta_{i,n}^2 & \gamma_{i,1}^2 & \cdots & \gamma_{i,j}^2 & \cdots & \gamma_{i,n}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \delta_i^l & \alpha_{i,1}^l & \alpha_{i,2}^l & \cdots & \alpha_{i,j}^l & \cdots & \alpha_{i,n}^l & \beta_{i,1}^l & \cdots & \beta_{i,j}^l & \cdots & \beta_{i,n}^l & \gamma_{i,1}^l & \cdots & \gamma_{i,j}^l & \cdots & \gamma_{i,n}^l \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \delta_i^z & \alpha_{i,1}^z & \alpha_{i,2}^z & \cdots & \alpha_{i,j}^z & \cdots & \alpha_{i,n}^z & \beta_{i,1}^z & \cdots & \beta_{i,j}^z & \cdots & \beta_{i,n}^z & \gamma_{i,1}^z & \cdots & \gamma_{i,j}^z & \cdots & \gamma_{i,n}^z \end{bmatrix}$$

$\delta_i^l=1$ 表示第 i 个模型中第 l 条规则有效, 即上面矩阵的第 l 行起作用, 否则第 l 条规则无效。

显然, 一共有 s 个模型在进化。

3.2 微粒适应度函数设计

如上所述, P_1 、 P_2 的微粒位置联合体($P_1 \cdot X_i, P_2 \cdot X_i$)确定第 i 个广义T-S模型 T_i 。广义T-S模型的优劣可从精度和复杂度两方面衡量。为减少计算量, 模型训练精度用各样本的模型误差绝对值中最大者 e 刻画, e 越小则精度越高。设训练样本集为: $\{(X_q, \hat{y}_q)\}$, 其中 $X_q = (x_{1q}, \dots, x_{jq}, \dots, x_{nq})$, \hat{y}_q 是当 X_q 作用于被建模对象时, 对象的真实输出值, $q=1, 2, \dots, N$, 则模型 T 关于训练样本集的误差为:

$$e(T) = \max_{q=1}^N (|y_q - \hat{y}_q|) \quad (13)$$

y_q 是 X_q 代入式(4)的输出, 即后面的式(16)。模型复杂度由模型实际存在的(有效的)规则数 $M(T) =$

$\sum_{i=1}^s \delta_i^l$ 来反映, M 越小则复杂度越低。训练的目标就

3 基于微粒群算法的广义T-S模糊模型训练方案

模型结构采用二进制种群 P_1 进行优化, 隶属函数 $u_F(x)$ 中的参数 α 、 β 和 γ 采用实数编码种群 P_2 进行优化, 后件中的系数参数 c_j^l 采用正交最小二乘法估计。

3.1 种群微粒设计

种群 P_1 和 P_2 都含 s 个微粒, P_1 中第 i 个微粒的位置为二进制位串 $(\delta_i^1, \delta_i^2, \dots, \delta_i^l, \dots, \delta_i^z)$, 记为 $P_1 \cdot X_i$, z 是预先设定的最大模型规则数。 P_2 中对应的第 i 个微粒的位置为模型规则中的隶属函数参数排列后所构成的浮点数串: $(\alpha_{i,1}^1, \dots, \alpha_{i,j}^1, \dots, \alpha_{i,n}^1; \beta_{i,1}^1, \dots, \beta_{i,n}^1, \dots, \beta_{i,n}^1; \gamma_{i,1}^1, \dots, \gamma_{i,n}^1, \dots, \gamma_{i,n}^1)$, 共有 $3n \times z$ 个参数, 记为 $P_2 \cdot X_i$, 其中 $l=1, 2, \dots, z; j=1, 2, \dots, n$ 。这两个微粒位置联合体($P_1 \cdot X_i, P_2 \cdot X_i$)确定第 i 个广义T-S模糊模型 T_i , 为便于观察, 将($P_1 \cdot X_i, P_2 \cdot X_i$)排为矩阵:

是要得出 $e(T)$ 和 $M(T)$ 都较小的模型, 为便于处理, 将此双目标优化转化成单目标优化问题, 因此将对应于广义T-S模型 T_i 的微粒 $P_1 \cdot X_i$ 和 $P_2 \cdot X_i$ 的适应度定义为:

$$f_{B_i}(P_1 \cdot X_i) = f_{P_2}(P_2 \cdot X_i) = f((P_1 \cdot X_i, P_2 \cdot X_i)) = f(T_i) = w_e e(T_i) + w_M M \quad (14)$$

其中 w_e 和 w_M 表示权值。

3.3 广义T-S模型后件系数参数估计

当模型每次结构学习和前件参数学习完毕, 就要估计后件参数。后件参数估计是个线性估计问题, 常用的一次性估计方法有最小二乘法, 其缺点是当遇到奇异矩阵或近似奇异矩阵时该法将失效或估计值不可信。本文采用正交最小二乘法估计后件参数, 该方法计算量较小, 且数值计算稳定性好。步骤如下:

设模型通过结构学习和前件参数的学习, 产生了 M 条有效规则(依次重新编号为 $1, 2, \dots, M$), 对于样本输入部分 $X_q = (x_{1q}, \dots, x_{jq}, \dots, x_{nq})$, 记:

$$\begin{cases} u_q^l = \prod_{j=1}^n u_{F_j^l}(x_{jq}) \\ h_q^l = \frac{u_q^l}{\sum_{l'=1}^M u_q^{l'}} \end{cases} \quad (15)$$

此时广义T-S模型的输出为:

$$y_q = \sum_{l=1}^M h_q^l y_q^l = \sum_{l=1}^M h_q^l (c_0^l + c_1^l x_{1q} + \cdots + c_n^l x_{nq}) =$$

$$[h_q^1, h_q^2, \dots, h_q^M \quad h_q^1 x_{1q}, h_q^2 x_{1q}, \dots, h_q^M x_{1q} \quad \dots$$

$$h_q^1 x_{nq}, h_q^2 x_{nq}, \dots, h_q^M x_{nq}] \cdot [c_0^1, c_0^2, \dots, c_0^M$$

$$c_1^1, c_1^2, \dots, c_1^M \quad \dots \quad c_n^1, c_n^2, \dots, c_n^M]^T$$

$$q=1, 2, \dots, N \quad (16)$$

记:

$$\mathbf{H} = \begin{bmatrix} h_1^1 \cdots h_1^M & h_1^1 x_{11} \cdots h_1^M x_{11} & \cdots & h_1^1 x_{n1} \cdots h_1^M x_{n1} \\ h_2^1 \cdots h_2^M & h_2^1 x_{12} \cdots h_2^M x_{12} & \cdots & h_2^1 x_{n2} \cdots h_2^M x_{n2} \\ \vdots & \vdots & \vdots & \vdots \\ h_N^1 \cdots h_N^M & h_N^1 x_{1N} \cdots h_N^M x_{1N} & \cdots & h_N^1 x_{nN} \cdots h_N^M x_{nN} \end{bmatrix}$$

$$\mathbf{C} = [c_0^1, c_0^2, \dots, c_0^M \quad c_1^1, c_1^2, \dots, c_1^M \quad \dots \quad c_n^1, c_n^2, \dots, c_n^M]^T$$

$$\mathbf{Y} = [y_1, y_2, \dots, y_q, \dots, y_N]^T$$

$$\hat{\mathbf{Y}} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_q, \dots, \hat{y}_N]^T$$

式中 \mathbf{H} 是 $N \times M(n+1)$ 的矩阵; \mathbf{C} 就是后件要估计的参数。对于 N 个样本, 将式(16)写成矩阵形式:

$$\mathbf{Y} = \mathbf{H}\mathbf{C} \quad (17)$$

\mathbf{C} 的最小二乘法估计值为:

$$\hat{\mathbf{C}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \hat{\mathbf{Y}} \quad (18)$$

当因计算误差积累或传递造成 $(\mathbf{H}^T \mathbf{H})$ 是奇异矩阵或接近奇异矩阵时, 该法将失效或估计值不可信。正交最小二乘法估计步骤如下:

令 $\bar{\mathbf{H}} = [\mathbf{H} | -\hat{\mathbf{Y}}]$, $\bar{\mathbf{C}} = [\mathbf{C} | 1]^T$, 对 $\bar{\mathbf{H}}$ 左乘 \mathbf{Q}_1 、 \mathbf{Q}_2 、 \dots 、 $\mathbf{Q}_{M(n+1)+1}$ 等Householder变换阵将 $\bar{\mathbf{H}}$ 约化为上三角阵:

$$\mathbf{Q}_{N \times N} \bar{\mathbf{H}}_{N \times M(n+1)} = \begin{bmatrix} \mathbf{A} & | & -\mathbf{B} \\ \mathbf{0} & | & \mathbf{g} \\ \mathbf{0} & | & \mathbf{0} \end{bmatrix} \quad (19)$$

式中 $\mathbf{Q}_{N \times N} = \mathbf{Q}_{M(n+1)+1} \dots \mathbf{Q}_2 \mathbf{Q}_1$; \mathbf{A} 为 $M(n+1)$ 阶上三角方阵; \mathbf{B} 为 $M(n+1)$ 阶列向量; \mathbf{g} 为标量。定义残差矢量 $\mathbf{r} = \mathbf{H}\mathbf{C} - \hat{\mathbf{Y}}$, 利用正交变换保持矢量2-范数不变的性质得:

$$\|\mathbf{r}\|_2^2 = \|\mathbf{H}\mathbf{C} - \hat{\mathbf{Y}}\|_2^2 = \|\bar{\mathbf{H}}\bar{\mathbf{C}}\|_2^2 = \|\mathbf{Q}\bar{\mathbf{H}}\bar{\mathbf{C}}\|_2^2 =$$

$$(\mathbf{A}\mathbf{C} - \mathbf{B})^T (\mathbf{A}\mathbf{C} - \mathbf{B}) + \mathbf{g}^2 \geq \mathbf{g}^2 \quad (20)$$

\mathbf{g} 与 \mathbf{C} 无关, \mathbf{C} 的正交最小二乘估计值由方程:

$$\mathbf{A}\mathbf{C} = \mathbf{B} \quad (21)$$

给出, 因 \mathbf{A} 为上三角方阵, 故可用回代法解出 \mathbf{C} 。上面的 \mathbf{Q}_1 , \mathbf{Q}_2 , \dots , $\mathbf{Q}_{M(n+1)+1}$ 并不要算出, 而是直接对 \mathbf{H} 施加变换即可, 具体算法参见文献[11]。显然 $\|\mathbf{g}\|$ 就是此时的残差矢量范数。

3.4 广义T-S模型参数学习算法框架

1) 选择 P_1 和 P_2 的参数 s 、 ω 、 c_1 和 c_2 ; 模型评价权值 w_e 和 w_m ; 微粒位置采用3.1所述方式编码; 代龄 $t=0$; 设置进化代限 \maxiter 。

2) 种群初始化: 随机生成 $P_1 \cdot \mathbf{X}^t$ 、 $P_1 \cdot \mathbf{V}^t$ 、 $P_2 \cdot \mathbf{X}^t$ 和 $P_2 \cdot \mathbf{V}^t$; 限制 $P_1 \cdot \mathbf{V}^t$ 的分量在 ± 4 之间, 同时根据式(3)要求对 $P_2 \cdot \mathbf{X}^t$ 作限幅处理。显然 $P_1 \cdot \mathbf{P}_i^0 = P_1 \cdot \mathbf{X}_i^0$, $P_2 \cdot \mathbf{P}_i^0 = P_2 \cdot \mathbf{X}_i^0$ 。由式(14)和式(8)计算 $P_1 \cdot \mathbf{P}_g^0$ 和 $P_2 \cdot \mathbf{P}_g^0$, 其中后件参数由式(21)计算。

3) 训练一次模型:

(1) 种群进化一代 $t=t+1$; P_2 按式(5)和(6)迭代, 对 $P_2 \cdot \mathbf{X}^t$ 作限幅处理; P_1 按式(5)迭代, 限制 $P_1 \cdot \mathbf{V}^t$ 的分量在 ± 4 之间, 再用式(9)和式(10)处理。其中参数 ω 、 c_1 和 c_2 按式(11)和式(12)处理。

(2) 由式(14)和式(7)计算 $P_1 \cdot \mathbf{P}_i^t$ 、 $P_2 \cdot \mathbf{P}_i^t$, 其中后件参数由式(21)计算; 由式(8)计算 $P_1 \cdot \mathbf{P}_g^t$ 和 $P_2 \cdot \mathbf{P}_g^t$ 。

4) 训练终止判断: 满足下面3个终止条件之一可转第5)步, 否则转第3)步。

(1) t 达到设定的值 \maxiter ;

(2) P_1 或 P_2 的微粒全局最优位置的适应度小到设定的值。有:

$$f_{P_1}(P_1 \cdot \mathbf{P}_g^t) = f_{P_2}(P_2 \cdot \mathbf{P}_g^t) = f((P_1 \cdot \mathbf{P}_g^t, P_2 \cdot \mathbf{P}_g^t))$$

(3) 连续 m 代 P_1 或 P_2 的微粒全局最优位置的适应度变化量 $(\Delta f_{P_1}(P_1 \cdot \mathbf{P}_g^t) = f_{P_1}(P_1 \cdot \mathbf{P}_g^t) - f_{P_1}(P_1 \cdot \mathbf{P}_g^{t-m}))$ 足够小。

5) 解码生成广义T-S模型, 输出近似最优完整解, 算法终止。

4 仿真实验与结果分析

本文采用上述算法训练广义T-S模型来辨识如下非线性时不变仿真对象:

$$y(k) = \frac{y(k-1)y(k-2)u(k-1)[y(k-3)-1] + u(k-2)}{1 + y^2(k-1) + y^2(k-2)} \quad (22)$$

本文 $u(k)$ 取 $(0, 5)$ 中均匀分布的白色信号, 生成含200个元素的训练样本集 $\{y(k-1), y(k-2), y(k-3), u(k-1), u(k-2); y(k)\}$, $k=4, 5, \dots, 203$, 初始值 $y(1)$ 、 $y(2)$ 、 $y(3)$ 取0; 用同样方法生成测试样本集。

仿真实验中训练算法各参数选择如下: 种群大小 $s=40$, 惯性因子(首次值) $\omega=0.7$, 加速度系数(首次值) $c_1=2, c_2=2$, 预设最大规则数 $z=30$, 从式(14)可知, 固定取 $w_e=1$, 则 w_M 越大越有可能得出规则数少而训练精度低的模型, 逐步减少 w_M 并试探多次运行程序得出 $w_M=0.01$ 左右时训练出的模型对于样本的最大训练误差多在 $0\sim 0.3$ 之间, 规则数多在 $16\sim 19$ 之间, 以 $16\sim 17$ 条居多, 因此取 $w_M=0.01$, 进化代限 $\text{maxiter}=200$, 种群在 $(0, 1)$ 中初始化。

将式(3)中的 γ 固定取2, 便得到高斯型隶属函数的T-S模型, 也用本文方法(γ 不参入寻优)训练高斯型隶属函数的T-S模型, 为了让得出的T-S模型与广义T-S模型有相近的精度, 将此时的 w_M 调为 0.015 。采用广义T-S模型与高斯型隶属函数的T-S模型($\gamma=2$)的建模结果对比如表1所示。

表1 2种模型建模结果对照表

模型类型	模型预报值最大误差	预报值误差均值	模型的规则数
广义T-S模型	0.249 3	0.071 2	16
T-S模型	0.247 8	0.072 6	22

从表1可看出, 在两个模型的预报值最大误差和预报值误差均值相差不大时, 广义T-S模型的规则数要少6条。

另外, 本文还采用基于遗传算法的参数学习方法训练广义T-S模型, 两种训练方法训练出的结果模型有关指标如表2所示。

表2 2种训练方法训练广义T-S模型的建模结果对照表

训练方法	模型预报值最大误差	预报值误差均值	模型的规则数	200代内最好适应度函数达到最佳值代数
本文方法	0.249 3	0.071 2	16	152
基于遗传算法的方法	0.251 4	0.072 3	16	183

从表2可见, 用本文方法得出的模型规则数相同, 模型的泛化效果相近, 但本文方法在200代内寻到最佳模型时经历的代数要少。

仿真算例表明, 将广义T-S模型的前件参数映射为微粒的位置坐标, 综合运用粒子群优化方法和线性估计方法可训练出性能良好的模糊模型。

5 结束语

用广文T-S模型建模时, 模型前件参数的调整是非线性优化问题, 而后件参数的调整是线性估计问题, 针对此特征, 本文提出的学习方案可训练出精度较高、规则较少的次优模型, 而且不需要任何先验知识; 其不足是算法参数的选择靠经验和试凑摸索, 未能从理论分析各个参数对结果误差和算法收

敛性的影响。另外, 由离散二进制编码PSO与连续实数编码PSO构成的复合型PSO的优化机理还有待研究。

参 考 文 献

- [1] 王士同. 模糊系统、模糊神经网络及应用程序设计[M]. 上海: 上海科学技术文献出版社, 1998.
WANG Shi-tong. Fuzzy system, fuzzy neural network and design of applied program[M]. Shanghai: Shanghai Science and Technology Literature Press, 1998.
- [2] TAKAGI T, SUGENO M. Fuzzy identification of systems and its application to modeling and control[J]. IEEE Trans on Systems, Man and Cybernetics, 1985, 15 (1): 116-132.
- [3] 李合生, 毛剑琴, 代冀阳. 基于遗传算法的广义Takagi-Sugeno模糊逻辑系统最优参数辨识[J]. 自动化学报, 2002, 28(4): 581-586.
LI He-sheng, MAO Jian-qin, DAI Ji-yang. Generalized takagi-sugeno fuzzy logical system optimal parameter identification based on genetical gorithm[J]. ACTA Automatica Sinica, 2002, 28(4): 581-586.
- [4] 陈国初, 俞金寿. 微粒群优化算法[J]. 信息与控制, 2005, 34(3): 318-324.
CHEN Guo-chu, YU Jin-shou. Particle swarm optimization algorithm[J]. Information and Control, 2005, 34(3): 318-324.
- [5] KENNEDY J, EBERHART R C. Particle swarm optimization[C]//Proceeding of 1995 IEEE International Conference on Neural Networks. New York, NY, USA: IEEE, 1995.
- [6] EBERHART R C, KENNEDY J. A new optimizer using particle swarm theory[C]//Proceedings of the Sixth International Symposium on Micro Machine and Human Science. New York, NY, USA: IEEE, 1995.
- [7] KENNEDY J, EBERHART R C. A discrete binary version of the particle swarm algorithm[C]//Proceedings of the 1997 International Conference on Systems, Man, and Cybernetics. New York, NY, USA: IEEE, 1997.
- [8] VAN DEN BERGH F. An analysis of particle swarm optimizers[D]. Pretoria, South Africa: Department of Computer Science, University of Pretoria, 2002.
- [9] KENNEDY J, SPEARS W N. Matching algorithm to problem: an experimental test of the particle swarm and some genetic algorithms on multimodal problem generator[C]//Proceedings of the International Conference on Evolutionary Computation. Anchorage, Alaska, USA: [s.n.], 1998.
- [10] RATNAWEERA A, HALGAMUGE S K, WATSON H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficient[J]. IEEE Trans Evol Comput, 2004, 8(3): 240-255.
- [11] 顾启泰. 正交最小二乘算法及其应用[J]. 清华大学学报, 1996, 36(3): 106-112.
GU Qi-tai. Orthogonal least-square estimation and its applications[J]. Journal of Tsinghua University(Sci&Tech), 1996, 36(3): 106-112.

编辑 税红