

改进果蝇优化 LSSVM 超声波萃取产物 浓度软测量

廖建庆¹, 王 涵¹, 王咸鹏²

(1. 宜春学院物理科学与工程技术学院, 江西宜春 336000;
2. 海南大学南海海洋资源利用国家重点实验室, 海南海口 570228)

摘要: 针对超声波天然产物萃取过程中产物浓度难以在线检测的问题, 提出了一种改进果蝇优化最小二乘支持向量机(Least Squares Support Vector Machine, LSSVM)的超声波萃取产物浓度软测量建模方法。首先将混沌优化与迭代步长动态调节方法相融合, 提出了一种混沌动态步长改进果蝇优化算法(Chaos Dynamic Step Fluit Fly Optimization Algorithm, CDSFOA), 该算法引入动态调节因子对步长动态更新, 并利用混沌优化实现各变量之间映射等操作, 能够有效提高果蝇优化算法的收敛精度和收敛速度, 然后利用 CDSFOA 对 LSSVM 进行参数寻优, 构建最优 CDSFOA-LSSVM 软测量模型, 最后利用超声波斛皮素萃取实验数据进行验证。结果表明, 提出的模型不仅有较好的学习和泛化能力, 而且具有良好的预测精度, 可为超声波天然产物萃取工艺优化提供理论指导。

关键词: 超声波; 萃取; 最小二乘支持向量机; 软测量

中图分类号: O426

文献标识码: A

文章编号: 1000-3630(2020)-02-0169-07

DOI 编码: 10.16300/j.cnki.1000-3630.2020.02.008

Soft measurement of ultrasonic extracted product concentration by LSSVM of improved fruit fly optimization algorithm

LIAO Jianqing¹, WANG Han¹, WANG Xianpeng²

(1. College of Physical Science and Engineering, Yichun University, Yichun 336000, Jiangxi, China;

2. State Key Laboratory of Marine Resources Utilization of the South China Sea, Hainan University, Haikou 570228, Hainan, China)

Abstract: Aiming at the problem that the product concentration in ultrasonic natural product extraction is difficult to detect online, a soft measurement method using the least squares support vector machine (LSSVM), which is optimized by an improved fruit fly optimization algorithm (FOA), is proposed. Firstly, the improved FOA with chaotic dynamic step size (named as CDS-FOA) is obtained by combing chaos optimization and iterative step-size dynamic adjustment. This algorithm introduces the dynamic adjustment factor to update the step size dynamically, and uses chaos optimization to realize the mapping between different variables, which can effectively improve the convergence precision and convergence speed of the FOA. Then, the CDS-FOA is used to optimize the parameters of LSSVM to construct the optimal CDSFOA-LSSVM soft measurement model. Finally, the experimental data from ultrasonic quercetin extraction are used to verify the effectiveness. Results show that the proposed model not only has better learning and generalization ability, but also has good prediction accuracy, which can provide guidance for the optimization of ultrasonic natural product extraction processes.

Key words: ultrasonic; extraction; least squares support vector machine (LSSVM); soft measurement

0 引 言

在超声波天然产物萃取领域, 通常需要对萃取产物溶液浓度进行实时监测, 以便及时了解和掌握萃取过程的工作状况和参数变化情况。产物溶液浓

度作为萃取过程控制的关键参数, 目前主要通过间断取样、实验室滴定的方法获得, 这样不可避免地存在一定的滞后性, 不能及时调整不合理的萃取参数和状态, 而且有时对腐蚀性强的液体检测还存在一定的困难^[1]。因此, 建立准确、可靠的超声波萃取过程产物浓度的软测量模型, 对提高超声波萃取质量、提升超声波天然产物萃取过程的自控水平具有重要意义。

针对诸多化学工业过程的优化问题, 国内外学者已提出了大量基于数据驱动的软测量方法, 由于最小二乘支持向量机(Least Squares Support Vector

收稿日期: 2019-01-12; 修回日期: 2019-03-16

基金项目: 国家自然科学基金项目(61701144)、江西省教育厅科技项目(GJJ170894)、江西省高等学校教改课题(JXJG-18-15-5)

作者简介: 廖建庆(1977-), 男, 江西吉安人, 博士, 副教授, 研究方向为超声波天然物提取过程建模与优化。

通讯作者: 廖建庆, E-mail: jndxljqs@126.com

Machine, LSSVM)采用等式约束与平方和的误差成本函数, 只要求解一个线性方程组, 这样使得运算效率得到极大提高^[2]。因此, 在发酵过程建模^[3]、超声波萃取过程建模^[4]、熔解指数预测^[5]等非线性化学工业建模领域得到了广泛应用。研究表明, 最小二乘支持向量机的精度和泛化能力主要受核函数和惩罚系统这两个参数的影响, 目前已有许多国内外研究者采用果蝇优化算法(Fruit Fly Optimization Algorithm, FOA)^[6]及其改进算法对最小二乘支持向量机的两个主要参数进行优化, 并取得了较好效果。

果蝇优化算法是台湾学者潘文超首次提出的一种群智能算法^[6]。该算法是一种基于果蝇觅食行为而推演出的全局优化算法, 已成功应用于(Proportion Integration Differentiation, PID)控制参数优化^[7]、支持向量机参数优化^[8]、多目标优化^[9]等领域。相较于其他群智能算法, 果蝇优化算法具有结构和计算简单、运行时间短和可调参数少等优点^[6]。当然, 由于算法自身的特点, 果蝇优化算法与其他智能算法一样, 也存在收敛速度慢、精度低等问题。对此, Liao 等^[4]在果蝇优化算法的基础上做了改进, 同时对 LSSVM 参数寻优, 得到了超声波萃取过程软测量模型。但由于算法种群单一且缺乏遍历性, 使得模型存在精度不高且运算速度较慢等问题。

本文提出了一种混沌动态步长改进果蝇优化算法。一方面, 通过利用混沌特性来提高果蝇种群的多样性和搜索的遍历性, 引入混沌扰动变量来提高算法的持续搜索能力; 另一方面, 通过引入步长动态调节因子对果蝇搜索步长进行随机动态调节, 使得算法的收敛速度得到进一步提高, 改进算法能有效地提高种群的多样性和搜索的遍历性, 进而提高算法精度, 避免陷入局部最优。然后利用该改进果蝇优化算法对 LSSVM 参数进行寻优, 并将该建模方法应用于超声波天然产物(卫矛茎干)萃取斛皮素溶液浓度的软测量中。

1 最小二乘支持向量机

设数据样本集为 $\{x_i, y_i\}, i=1, 2, \dots, n$, 其中 $x_i \in \mathbf{R}^n$ 为输入量, $y_i \in \mathbf{R}$ 为对应于 x_i 的输出量, i 为样本数目, 则问题的目标为确定决策函数:

$$y(x) = \text{sgn}(f(x)) \quad (1)$$

其中, 函数 $f(x)$ 表达形式为

$$f(x) = \boldsymbol{\omega}^T \cdot \boldsymbol{\phi}(x) + \mathbf{b} \quad (2)$$

其中: $\boldsymbol{\phi}(\cdot)$ 为输入空间映射到高维特征空间的非线性映射, 待求量 $\boldsymbol{\omega}$ 为系数向量, \mathbf{b} 为偏置项。未知量可通过式(3)的最优化问题来确定:

$$\begin{cases} \min_{\boldsymbol{\omega}, \mathbf{b}, \boldsymbol{\xi}} L(\boldsymbol{\omega}, \mathbf{b}, \boldsymbol{\xi}) = \frac{1}{2} \boldsymbol{\omega}^T \boldsymbol{\omega} + \frac{1}{2} \gamma \sum_{i=1}^n \xi_i^2 \\ \text{s.t. } y_i (\boldsymbol{\omega}^T \cdot \boldsymbol{\phi}(x_i) + \mathbf{b}) = 1 - \xi_i, \quad i=1, 2, \dots, n \end{cases} \quad (3)$$

其中: γ 为惩罚系数。为了求解该优化问题, 利用拉格朗日(Lagrange)函数求解式(3)的优化问题, 根据卡罗需-库恩-塔克(Karush-Kuhn-Tucker, KKT)优化条件, 得到线性方程组为

$$\begin{bmatrix} 0 & -\mathbf{y}^T \\ \mathbf{y} & \boldsymbol{\Omega} + \gamma \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{b} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \quad (4)$$

其中: $\boldsymbol{\Omega}$ 是方阵, 其第 j 行 i 列的元素可表示为 $\Omega_{ij} = y_i y_j \boldsymbol{\phi}(x_i)^T \cdot \boldsymbol{\phi}(x_j) = y_i y_j K(x_i, x_j)$; $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$; $\mathbf{Q} = [y_1 \boldsymbol{\phi}(x_1), y_2 \boldsymbol{\phi}(x_2), \dots, y_n \boldsymbol{\phi}(x_n)]^T$; \mathbf{I} 为单位矩阵; $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_n]^T$, $i, j=1, 2, \dots, n$ 。

求解式(4), 可得最小二乘支持向量机的输出为

$$f(x) = \sum_{i=1}^n \alpha_i K(x, x_i) + \mathbf{b} \quad (5)$$

其中: $K(x, x_i)$ 为核函数, 本文用径向基函数(Radial Basis Function, RBF)核, $K(x_i, x) = e^{-\|x_i - x\|^2 / 2\sigma^2}$, σ 为核宽度。

2 果蝇优化算法的改进

2.1 初始果蝇优化算法

初始果蝇优化算法步骤如下:

(1) 初始化 给定种群规模为 N_{Sizepop} , 最大迭代次数为 M , 随机初始化果蝇群体位置为 $(X_{\text{axist}}, Y_{\text{axist}})$;

(2) 分配各果蝇个体利用嗅觉搜寻食物的方向与距离, 产生随机参数 $V_R \in [-1, 1]$ (又称迭代步长), 结合初始位置, 产生果蝇个体下一步的新位置为

$$\begin{cases} X_i = X_{\text{axist}} + V_R \\ Y_i = Y_{\text{axist}} + V_R \end{cases} \quad (6)$$

(3) 估计果蝇个体与原点间的距离 D_i , 计算味道浓度判定值 C_{Si} ;

$$D_i = \sqrt{X_i^2 + Y_i^2} \quad (7)$$

$$C_{Si} = \frac{1}{D_i} \quad (8)$$

(4) 将各味道浓度判定值 C_{Si} 分别代入味道浓度判定函数(又称目标函数), 求出各果蝇个体位置的味道浓度 C_{Smelli} ;

$$C_{\text{Smelli}} = \text{Function}(C_{Si}) \quad (9)$$

(5) 对果蝇个体味道浓度值进行排序, 找出当前味道浓度最大的果蝇个体 $C_{\text{bestSmell}}$;

$$[C_{\text{bestSmell}} \quad I_{\text{best}}] = \max(C_{\text{Smelli}}) \quad (10)$$

式(10)中, $C_{\text{bestSmell}}$ 表示味道浓度最佳的果蝇个体, I_{best} 表示最佳果蝇的序号;

(6) 根据记录的最佳果蝇序号和最佳的果蝇位置(最优个体), 此时所有果蝇个体利用视觉向最佳位置飞去;

$$C_{\text{Smellbest}} = C_{\text{bestSmell}} \quad (11)$$

$$\begin{cases} X_{\text{axist}} = X(I_{\text{best}}) \\ Y_{\text{axist}} = Y(I_{\text{best}}) \end{cases} \quad (12)$$

其中, $C_{\text{Smellbest}}$ 表示最佳浓度。

(7) 最后进行迭代过程, 重复步骤(2)~(5), 在寻优过程中判断当前迭代的最佳味道浓度是否优于上一迭代的最佳味道浓度, 同时, 判断当前迭代次数是否小于最大迭代次数 M ; 若是则执行步骤(6), 否则结束。

2.2 步长调节策略

对于初始 FOA, 迭代步长的选择直接影响最佳浓度位置, 进而影响算法的收敛速度和精度。本文借鉴文献[10]的思想, 引入一种步长调节因子来动态地调节搜索步长, 即对方向与距离产生的随机搜索步长(V_R)分配不同的步长调节因子 $\omega(n)$ 以实现步长自适应地动态调节, 表达式见式(13)和(14)。

$$\omega(n) = \omega_0 \cdot \exp\left[-20\left(\frac{n}{M}\right)^a\right] \quad (13)$$

$$V_R = \omega(n) \times (2r-1) \quad (14)$$

其中: n 为迭代次数; $\omega(n)$ 为动态步长调节因子; a 为调节参数, 可视具体情况取 $[1, 30]$ 内的整数; $r \in [0, 1]$, 为随机函数。

果蝇个体利用嗅觉搜寻食物的随机方向与距离式(6)调整为

$$\begin{cases} X_i = X_{\text{axist}} + \omega(n) \times (2r-1) \\ Y_i = Y_{\text{axist}} + \omega(n) \times (2r-1) \end{cases} \quad (15)$$

$\omega(n)$ 随迭代次数的动态变化见图 1。结合式(13)~(15)和图 1 可见, 在算法迭代初期, $\omega(n)$ 随迭代次数的增加衰减较慢, 即保持一个较大的步长

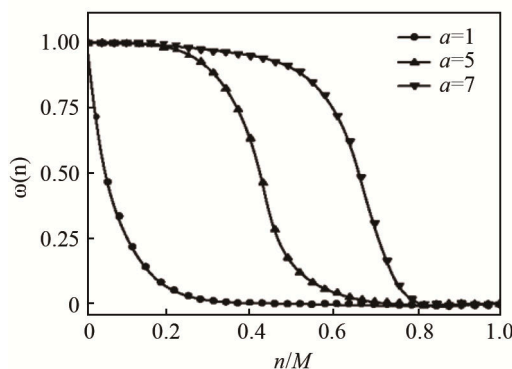


图1 步长因子的变化曲线

Fig.1 Variation curves of the dynamic step factor

寻优, 可以有效地避免算法过早陷入局部最优。到了迭代后期, $\omega(n)$ 随迭代次数的增加而衰减到一个很小的值, 使得算法在后期的收敛速度加快, 从能克服果蝇算法在寻优后期收敛速度慢、易陷入局部最优等缺点。

2.3 混沌优化

混沌优化^[11]是一种利用混沌运动特性进行优化搜索方法, 由于混沌变量映射通常是由 Logistic 方程得到, 而 Logistic 映射是混沌系统中最著名及应用最多的系统模型之一, 因此本文引入 Logistic 映射, 其数学模型为

$$x(k+1) = \mu x(k)[1-x(k)] \quad (16)$$

其中: k 为迭代次数; $x(k) \in [0, 1]$; μ 为控制变量。研究表明, 当 $x(k) \neq 0, 1$ 时, Logistic 映射处于混沌状态^[12]。依据式(16), 混沌变量 $C_{x_j}(k)$ 的一种变换式为

$$C_{x_j}(k+1) = 4C_{x_j}(k)(1-C_{x_j}(k)) \quad (17)$$

其中: C_{x_j} 表示第 j 个混沌变量, 显然当 $C_{x_j}(0) \neq 0, 1$ 时, 将产生混沌现象。式(17)中的优化变量 $x_j(k) \in [x_{L,j}, x_{U,j}]$, 利用式(18)和(19)与混沌变量 $C_{x_j}(k+1)$ 进行往返映射。

$$C_{x_j}(k) = [x_j(k) - x_{L,j}] / (x_{U,j} - x_{L,j}) \quad (18)$$

$$x'_j(k) = x_{L,j} + C_{x_j}(k)(x_{U,j} - x_{L,j}) \quad (19)$$

式(19)中, $x_{L,j}$ 和 $x_{U,j}$ ($j=1, 2, \dots, n$) 分别为第 j 维变量的搜索上界和下界; $x'_j(k)$ 为经混沌映射后的第 j 个混沌变量 $C_{x_j}(k)$ 转为优化变量获得的值。

2.4 算法验证

选取 6 个非线性函数进行测试, 各函数形式、搜索范围、理论最优值和峰值如表 1 所示。其中 f_1 和 f_3 是单峰函数, f_2 和 f_4-f_6 为多峰函数。为了验证本文提出的 CDSFOA 算法性能, 选择经典的粒子群算法(Particle Swarm Optimization, PSO)、蝙蝠算法(Bat Algorithm, BA)和 FOA 3 种算法进行比较, 算法性能采用下列两种方法进行评估。

将测试函数迭代次数固定设为 800 次, 分别采用 PSO、BA、FOA 和 CDSFOA 算法经过 20 次独立运行, 并将运行得到的最优 $C_{\text{Smellbest}}$ 值用最优值(C_B)、最差值(C_W)、优化均值(C_M)、标准方差(S_{id})以及平均运行时间来表征, 结果如表 2 所示。

由表 2 可见, 与 PSO、BA、FOA 算法相比, CDSFOA 算法对各测试函数都能找到或更接近理论的最优值, 即求解得到的最优值、最差值、优化均值和标准方差均优于 POS、BA 和 FOA 算法; 另外在搜索时间方面, CDSFOA 算法相比 PSO 和 BA

表 1 测试函数
Table 1 Benchmark functions

函数	表达式	搜寻范围	最优值	峰值
球曲面函数	$f_1(x) = \sum_{i=1}^n x_i^2$	[-100, 100]	0	单峰
格里旺克函数	$f_2(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600, 600]	0	多峰
罗森布鲁克函数	$f_3(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[30, 30]	0	单峰
Rastrigin 函数	$f_4(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5.12, 5.12]	0	多峰
阿克莱函数	$f_5(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i) + 20 + e$	[-100, 100]	0	多峰
沙弗函数	$f_6(x) = \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} + 0.5$	[-100, 100]	0	多峰

表 2 各算法的性能比较
Table 2 Performance comparison between different algorithms

函数	维数	算法	C_B	C_W	C_M	S_{id}	时间/s
球曲面函数	30	PSO	0.037 6	0.098 8	0.057 6	1.300 1	2.778 9
		BA	0.212 6	1.401 9	1.117 8	2.124 8	2.299 8
		FOA	0.003 5	0.102 5	0.010 6	0.003 5	0.396 5
		CDSFOA	0	5.6632×10^{-15}	8.9981×10^{-15}	7.632×10^{-6}	0.498 7
格里旺克函数	30	PSO	3.0098×10^{-3}	4.0065×10^{-2}	8.9982×10^{-1}	5.003×10^{-1}	6.332 4
		BA	3.1265×10^{-2}	8.9954×10^{-1}	0.9954×10^{-2}	2.0036×10^{-1}	4.335 8
		FOA	0.9986×10^{-2}	7.0069×10^{-2}	2.9635×10^{-2}	0.3356×10^{-1}	0.965 4
		CDSFOA	0	0.0001	0.000 4	0.002 2	0.918 6
罗森布鲁克函数	30	PSO	18.395 4	39.265 4	20.069 8	26.332 8	6.221 4
		BA	21.952 4	61.023 8	33.002 4	25.482 1	6.008 7
		FOA	17.215 6	37.215 9	20.135 9	8.952 4	0.482 5
		CDSFOA	11.0235 7	16.215 8	12.036 9	2.6514 7	0.789 1
Rastrigin 函数	30	PSO	10.985 2	16.325 8	10.985 4	4.325 8	8.025 8
		BA	25.369 4	44.365 2	29.215 8	10.258 7	7.002 5
		FOA	13.258 6	12.325 1	11.658 7	0.235 8	1.069 8
		CDSFOA	0	$0.235 6 \times 10^{-11}$	$3.215 7 \times 10^{-12}$	$2.248 9 \times 10^{-4}$	1.302 5
阿克莱函数	30	PSO	$9.856 4 \times 10^{-6}$	$8.002 3 \times 10^{-5}$	$5.326 9 \times 10^{-6}$	$9.321 5 \times 10^{-4}$	2.014 6
		BA	$0.564 2 \times 10^{-4}$	$6.023 5 \times 10^{-2}$	$0.885 41 \times 10^{-3}$	$7.902 1 \times 10^{-1}$	2.913 5
		FOA	$6.325 8 \times 10^{-8}$	$2.014 5 \times 10^{-6}$	$3.302 1 \times 10^{-7}$	$0.663 5 \times 10^{-3}$	0.802 4
		CDSFOA	$1.235 4 \times 10^{-10}$	$2.001 4 \times 10^{-8}$	$4.021 8 \times 10^{-9}$	$1.587 4 \times 10^{-2}$	0.889 1
沙弗函数	30	PSO	$6.874 1 \times 10^{-4}$	$7.325 8 \times 10^{-3}$	$2.951 4 \times 10^{-4}$	$5.021 8 \times 10^{-2}$	5.229 8
		BA	$2.147 8 \times 10^{-3}$	$8.021 6 \times 10^{-1}$	$8.250 3 \times 10^{-3}$	$2.018 4 \times 10^{-1}$	4.987 2
		FOA	$5.023 6 \times 10^{-4}$	$6.230 1 \times 10^{-2}$	$9.014 7 \times 10^{-3}$	$7.025 8 \times 10^{-1}$	1.223 7
		CDSFOA	$1.025 8 \times 10^{-5}$	$7.021 5 \times 10^{-4}$	$1.025 8 \times 10^{-5}$	$2.987 4 \times 10^{-2}$	1.503 6

两种算法所需时间都要短，但也发现 CDSFOA 算法相比 FOA 算法所需时间稍长，可能的原因为 CDSFOA 算法对果蝇群体做了动态步长的调节操作和引入了混沌扰动操作，从而增加了运算时间。同时也发现，改进算法在寻优精度方面要远优于 FOA 算法，表明本算法具有更高的精度和更好的稳定性。

3 CDSFOA 优化 LSSVM 软测量模型

3.1 基于 CDSFOA 算法的 LSSVM 模型参数优化

LSSVM 模型参数优化的思想是通过智能优化算法搜索一组最优的参数(γ 和 σ)。本文利用改进的果蝇优化算法对 LSSVM 模型参数进行寻优，即

构建 CDSFOA-LSSVM 软测量模型。CDSFOA 算法优化 LSSVM 模型参数 γ 和 σ 的流程如下：

- (1) 参数初始化；
- (2) 利用式(13)和(14)更新果蝇的搜寻步长，利用式(15)更新果蝇的位置，产生该果蝇个体下一步的新位置 (X_i, Y_i) ；
- (3) 估计与各果蝇个体与原点之间的距离 D_i ，计算味道浓度判定值 C_{Si} ；
- (4) 计算果蝇群体的平均味道浓度(平均适应度) $C_{Smell,A}$ 和味道浓度方差 δ^2 ；
- (5) 当遍历次数 $M > 0$ 时，判断 δ^2 是否小于味道浓度方差阈值 H ，若是，将更新后的步长果蝇种群位置 (X_i, Y_i) 进行混沌优化和映射操作，最后将混沌位置变量 (X'_i, Y'_i) 返回到搜索空间内的果蝇种群新位置， $M = M - 1$ ；否则，转步骤(9)；
- (6) 分别估计新位置 (X'_i, Y'_i) 与原点之间的距离 D'_i 和味道浓度判定值 C'_{Si} ；
- (7) 求解果蝇个体位置味道浓度 $C'_{Smelli,1}$ 和 $C'_{Smelli,2}$ ；
- (8) 判断味道浓度 $C'_{Smelli,1}$ 、 $C'_{Smelli,2}$ 是否优于最佳味道浓度 $C_{Smellbest}$ ，若不是，则 $C_{Smellbest} = \{C'_{Smelli,1}, C'_{Smelli,2}\}$ ， $(X_{axist}, Y_{axist}) = (X'_i, Y'_i)$ ，然后转步骤(3)，否则执行步骤(10)；
- (9) 经若干次迭代搜索后，判断 $C_{Smellbest}$ 是否保持不变，若是则引入混沌扰动，然后转步骤(2)，否则执行步骤(10)；
- (10) 将得到的优化解作为参数 γ 和 σ 值，建立最优 CDSFOA-LSSVM 软测量模型。

3.2 CDSFOA-LSSVM 建模步骤

基于 CDSFOA-LSSVM 的超声波提取过程建模流程如图 2 所示，其建模实现步骤如下：

- (1) 确定模型的输入变量和输出变量，同时对样本预处理；
- (2) 确定模型最优参数组合 (γ, σ) ；
- (3) 构建模型的核矩阵；
- (4) 采用二次规划算法解出最优 α_i 值；
- (5) 求解偏置项 b 的值、最优向量和支撑向量，构造 CDSFOA-LSSVM 的输出模型；
- (6) 利用 CDSFOA-LSSVM 模型对超声波提取过程进行训练和预测。

4 斛皮素浓度预测模型

超声波天然产物萃取槽简易模型如图 3 所示。为了构建超声波天然物提取过程的软测量预测模

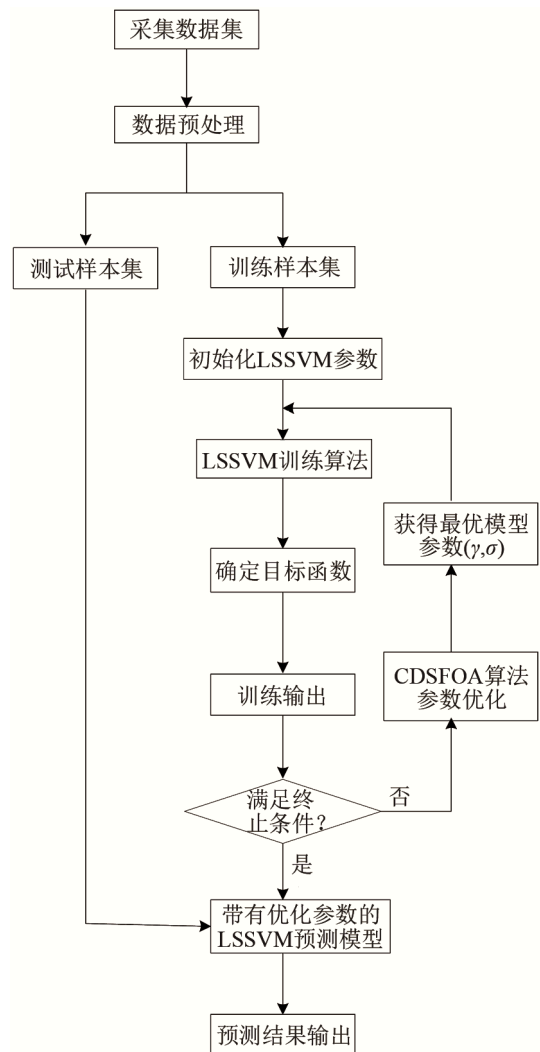


图 2 基于 CDSFOA-LSSVM 的超声波提取过程建模流程图
Fig.2 Modeling flow chart of ultrasonic extraction process based on CDSFOA-LSSVM

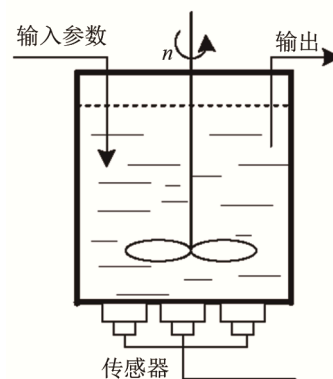


图 3 超声波萃取槽示意图
Fig.3 Schematic diagram of ultrasound extraction tank

型，本文以天然产物——卫矛茎干为萃取对象，通过主成分分析法确定萃取温度、萃取时间、超声频率、超声功率、溶剂浓度和液料比等 6 个萃取变量作为模型的输入，将萃取产物斛皮素溶液浓度

作为模型的输出。以超声波卫矛茎干萃取过程实验数据作为建模数据样本, 样本随机取自不同的实验批次。

依据上述建模方法, 利用 CDSFOA-LSSVM 的超声波萃取过程的建模步骤, 首先, 对样本做归一化预处理, 然后将样本分为训练样本集、验证样本集和推广样本集。然后利用训练样本集训练 LSSVM 模型, 接着再利用 CDSFOA 对 LSSVM 两个参数进行寻优, 得到优化后的 CDSFOA-LSSVM 软测量模型, 最后利用验证样本集对软测量模型预测效果进行检验。预测误差如表 3 所示。

表 3 模型对验证样本集的预测误差
Table 3 Prediction errors of different models for the verification sample set

模型	E_{MA}	$E_{MR}/\%$	E_{RMS}	S_{id}
LSSVM	0.073 0	3.02	0.123 6	0.105 5
FOA-LSSVM	0.069 5	2.25	0.100 4	0.089 1
CDSFOA-LSSVM	0.057 6	1.88	0.089 2	0.070 6

由表 3 可见, 改进果蝇优化后的 LSSVM 模型相对初始 LSSVM 模型的平均绝对误差(E_{MA})、平均相对误差(E_{MR})、均方根误差(E_{RMS})和标准差(S_{id})分别减少了 0.015 4、1.14%、0.034 4 和 0.034 9, 说明 CDSFOA-LSSVM 软测量模型对验证样本集的预测误差明显小于初始 LSSVM, 而 FOA 算法优化的 LSSVM 模型介于中间。

为了进一步验证所提模型的推广性能, 初始 LSSVM、FOA-LSSVM 和 CDSFOA-LSSVM 预测模型对推广样本集的预测误差结果如表 4 所示。

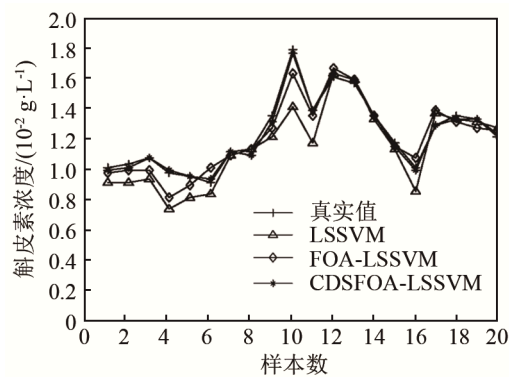
表 4 模型对推广样本集的预测误差
Table 4 Prediction errors of different models for the generalized sample set

模型	E_{MA}	$E_{MR}/\%$	E_{RMS}	S_{id}
LSSVM	0.0790	3.65	0.1013	0.0995
FOA-LSSVM	0.0602	2.10	0.0660	0.0714
CDSFOA-LSSVM	0.0457	1.63	0.0442	0.0507

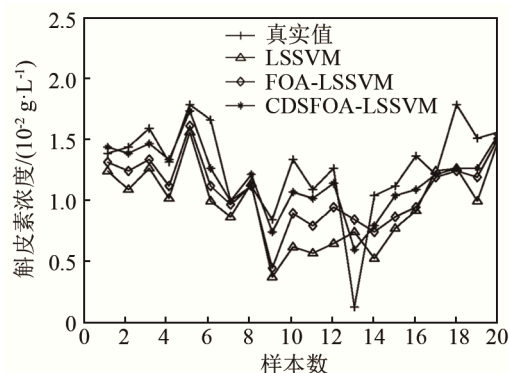
从表 4 中四个误差指标可见, 优化后的 LSSVM 模型比初始 LSSVM 模型得到的预测结果更接近真实值, 而改进后的果蝇优化算法 CDSFOA 相比未改进的果蝇优化算法 FOA 优化效果更好。其中, CDSFOA-LSSVM 的 E_{MA} 相对初始 LSSVM 的 E_{MA} 减少了 0.033 3, E_{MR} 减少了 2.02%, E_{RMS} 减少了 0.057 1, S_{id} 减少了 0.048 8。

优化前后 LSSVM 预测模型对验证样本和推广样本的预测跟踪结果如图 4 所示。由图 4 可以看出, 改进的果蝇优化 LSSVM 预测模型相比未改进的果

蝇优化 LSSVM 以及初始 LSSVM 预测模型的跟踪性能得到了明显提高, 更加接近真实值。另外, 优化后的 LSSVM 模型的泛化能力比初始 LSSVM 更好, 如图 4(b)所示。



(a) 验证样本集



(b) 推广样本集

图 4 模型对(a)验证样本集, (b)推广样本集的预测结果
Fig.4 Prediction results of different models for (a) verification sample set, (b) generation sample set

5 结 论

(1) 在初始果蝇算法的基础上, 引入步长动态调节因子和混沌优化算法, 提出了一种性能更加优越的改进果蝇优化算法(CDSFOA)。

(2) 为验证模型的有效性, 通过 6 个测试函数的仿真实验, 表明该算法相比其他对比算法在收敛精度和速度方面有明显的优势。

(3) 通过超声波萃取斛皮素浓度过程建模实例, 比较初始 LSSVM、FOA 优化的 LSSVM 以及 CDSFOA 优化的 LSSVM 预测模型, 表明 CDSFOA 优化算法提高了模型的预测精度和泛化能力。

参 考 文 献

- [1] 刘镇清. 化工生产液浓度的超声波在线检测[J]. 现代化工, 1995, 15(8): 43-44.
LIU Zhenqing. Ultrasonic online detection of chemical production

- liquid concentration[J]. *Modern Chemical Industry*, 1995, **15**(8): 43-44.
- [2] 石洋, 胡长青. 基于粒子群最小二乘支持向量机的前视声呐目标识别[J]. *声学技术*, 2018, **37**(2): 122-128.
SHI Yang, HU Changqing. Forward-looking sonar target recognition based on particle swarm and least squares support vector machine[J]. *Technical Acoustics*, 2018, **37**(2): 122-128.
- [3] NIU D P, ZHANG L, WANG F L. Modeling and parameter updating for nosiheptide fed-batch fermentation process[J]. *Industrial & Engineering Chemistry Research*, 2016, **55**(30): 8395-8402.
- [4] LIAO J Q, QU B D, ZHENG N. Effects of process parameters on the extraction of quercetin and rutin from the stalks of euonymus alatus (thumb.) sieb and predictive model based on least squares support vector machine optimized by an improved fruit fly optimization algorithm[J]. *Applied Sciences*, 2016, **6**(11): 340-359.
- [5] WANG W C, LIU X G. Melt index prediction by least squares support vector machines with an adaptive mutation fruit fly optimization algorithm[J]. *Chemometrics and Intelligent Laboratory Systems*, 2015, **141**: 79-87.
- [6] PAN W T. A new fruit fly optimization algorithm: taking the financial distress model as an example[J]. *Knowledge-Based Systems*, 2012, **26**: 69-74.
- [7] ZHANG X Y, CHEN G, JIA S M. Parameters optimization of PID controller based on improved fruit fly optimization algorithm[C]//*International Conference on Swarm Intelligence*. Springer, Cham, 2018: 421-431.
- [8] ZHANG Q T, FANG L Q, MA L L, et al. Research on parameters optimization of SVM based on improved fruit fly optimization algorithm[J]. *International Journal of Computer Theory and Engineering*, 2016, **8**(6): 500-505.
- [9] RIZK-ALLAH R M, EL-SEHIEMY R A, DEB S, et al. A novel fruit fly framework for multi-objective shape design of tubular linear synchronous motor[J]. *The Journal of Supercomputing*, 2017, **73**(3): 1235-1256.
- [10] YU Y C, LI Y, LI J C, et al. Self-adaptive step fruit fly algorithm optimized support vector regression model for dynamic response prediction of magnetorheological elastomer base isolator[J]. *Neurocomputing*, 2016, **211**: 41-52.
- [11] 廖建庆, 王涵, 王威鹏. 混沌动态步长果蝇优化算法[J]. *传感器与微系统*, 2019, **38**(8): 139-142.
LIAO Jianqing, WANG Han, WANG Xianpeng. Health state assessment of a rolling bearing based on CFOA-MKHSVM method[J]. *Chinese Journal of Scientific Instrument*, 2019, **38**(8): 139-142.
- [12] VAIDYANATHAN S. Chaos in neurons and adaptive control of Birkhoff-Shaw strange chaotic attractor [J]. *International Journal of PharmTech Research*, 2015, **8**(5): 956-963.
- [13] YUAN X F, DAI X S, WU L H. A mutative-scale pseudo-parallel chaos optimization algorithm[J]. *Soft Computing*, 2015, **19**(5): 1215-1227.