

基于改进核极限学习机和集成学习理论的目标机动轨迹预测*

寇英信, 奚之飞, 徐安, 李战武, 杨爱武
(空军工程大学航空工程学院, 陕西西安 710038)

摘要: 为了提高目标轨迹预测的精度以及预测模型的泛化能力, 提出基于改进蝙蝠算法优化的核极限学习机(Kernel Extreme Learning Machine, KELM)和集成学习理论目标机动轨迹预测模型。构建KELM模型, 并采用改进的蝙蝠算法对KELM的参数进行优化; 以优化后的KELM神经网络为弱预测器, 结合集成学习算法生成强预测器, 通过训练不断优化强预测的结构和参数, 得到一种基于集成学习理论的目标机动轨迹预测模型; 基于不同规模的样本, 将所得预测模型与逆传播神经网络、支持向量机和极限学习机等模型进行对比分析。仿真结果表明: 所提目标机动轨迹预测模型具有较好的预测精度和泛化能力。

关键词: 轨迹预测; 蝙蝠算法; 核极限学习机; 集成学习理论

中图分类号: TN95 文献标志码: A 文章编号: 1001-2486(2021)05-023-13

Maneuver trajectory prediction of target based on improved KELM and ensemble learning theory

KOU Yingxin, XI Zhifei, XU An, LI Zhanwu, YANG Aiwu

(Aeronautics Engineering College, Air Force Engineering University, Xi'an 710038, China)

Abstract: In order to improve the forecasting accuracy and generalization ability, a target maneuver trajectory forecasting approach based on ensemble learning theory and KELM (kernel extreme learning machine) optimized by the modified bat-inspired algorithm was proposed. A KELM model optimized by improved bat-inspired algorithm was constructed. Combined with the ensemble learning theory, the improved KELM neural network was regarded as weak predictor to generate strong predictor, the structure and parameters of the strong predictor were continuously optimized through training, and a target maneuver trajectory prediction model based on the ensemble learning theory was obtained. Based on samples of different sizes, the prediction performance of the model proposed in this paper was compared with BP (back propagation) neural network, support vector machine and extreme learning machine. The simulation results show that the generalization ability and prediction accuracy of the prediction model proposed is good.

Keywords: trajectory prediction; bat algorithm; kernel extreme learning machine; ensemble learning theory

轨迹预测是根据目标的历史运动轨迹, 学习和推理其包含的内在信息, 进而对目标未来的运动趋势做出合理推测的过程。在空战对抗过程中, 对敌方目标的未来运动轨迹做出合理的预测具有重要意义。现代空战获取胜利的关键在于先于敌方形成观察—判断—决策—行动(Observation Orientation Decision Action, OODA)循环, 从而达到先发制人的目的^[1]。而实现这一切的基础是能够对目标机动轨迹进行精确预测, 因此探索一种能够对目标机动轨迹实现准确预测的理论方法具有重要意义。

近几年, 对于目标机动轨迹预测方法的研究方向主要分为两种。

一种是参数法, 主要包括卡尔曼滤波、 α/β 滤波以及线性回归等传统预测方法。例如文献[2]针对目标运动模式不断变化、机动幅度较大的情况, 提出一种基于多项式卡尔曼滤波的运动轨迹预测算法; 文献[3]针对目标历史位置信息存在缺失的情况, 提出一种融合系统噪声估计的改进卡尔曼滤波算法; 文献[4]针对传统轨迹预测算法存在预测精度低以及算法实时性差的不足, 提出一种基于卡尔曼滤波的动态轨迹预测算法; 文献[5]针对精确目标运动模型存在高度非线性、数据处理难度大以及预测精度低等问题, 提出一种改进的交互式多模型轨迹预测算法。上述预测方法都只适用于目标运动特性相对简单的轨迹预

* 收稿日期: 2020-04-06

基金项目: 空军工程大学校长基金资助项目(XZJK2019040)

作者简介: 寇英信(1965—), 男, 陕西铜川人, 教授, 博士, 博士生导师, E-mail: kgykx@hotmail.com

测问题,但是在实际空战中,目标的运动往往是高度复杂的时序动态变化过程,目标的运动受到多种因素的影响,传统的预测算法无法充分学习目标的机动特性;同时,为了更加准确地描述目标的运动,在建模时一般模型的复杂度较高,导致预测算法的实时性和适应性较差,从而无法满足空战对抗的需求。

另一种是以神经网络为核心的非参数预测算法,主要基于目标历史运动数据对目标未来机动轨迹进行预测。例如:文献[6]基于广义回归神经网络(Generalized Regression Neural Network, GRNN)良好的非线性映射能力、高度的容错性以及鲁棒性,提出一种基于 GRNN 的高超声速飞行器轨迹预测方法;文献[7-8]针对逆传播(Back Propagation, BP)神经网络存在对初值敏感且全局搜索能力较差的不足,分别提出一种基于遗传算法和粒子群算法优化的 BP 神经网络轨迹预测模型。非参数法不需要建立准确的目标运动模型,也能实现对目标机动轨迹的精确预测,但是这些方法需要通过烦琐冗长的迭代训练才能达到精确预测的目的,并且神经网络在训练的过程中容易陷入局部最优解,因此在实际预测中很难达到最佳的预测效果。

无论是参数法还是非参数法,应用到目标机动轨迹预测中,在实时性、精确性和模型复杂性等方面都存在一定的不足。总体而言,参数方法的特点是预测性能与模型精度成正相关,主要存在模型学习能力不足、难以处理模型内在不确定性及目标运动数据突变影响预测精度等缺点。非参数方法则基于数据挖掘和神经网络^[9]等方法根据目标历史机动轨迹数据学习数据变化的内在规律,建立输入输出映射关系。其缺点是预测精度对历史数据的数量和质量依赖程度高,且模型参数确定难度较大。受空战对抗以及空战环境等不确定因素影响,实际目标机动轨迹数据具有较强的非线性、时变性和易受随机噪声影响等特征。现有目标机动轨迹预测建模方法存在模型参数确定困难、预测精度不高和泛化能力较差等不足,难以满足复杂空战环境下的目标机动轨迹预测的要求。此外,采用单一预测模型对目标机动轨迹进行预测时往往采用给定的全部训练样本进行全局建模,导致需要建立具有复杂结构的全局模型对大量样本支撑的复杂假设空间进行全局逼近^[10],从而进一步增加预测模型的优化难度。另外,全局预测模型容易出现对样本数据局部信息利用不充分的情况,导致难以完整描述训练样本支持的

假设空间,进而影响目标机动轨迹的预测精度。

为了克服参量法和非参量法以及单一全局建模存在的不足,本文提出一种基于改进的(Kernel Extreme Learning Machine, KELM)和集成学习理论的目标机动轨迹预测模型。以 AdaBoost. RT 算法为集成框架, KELM 神经网络为弱预测器,通过不断训练得到强预测器。为了进一步提高模型的预测性能,一方面针对 KELM 神经网络的预测性能容易受到惩罚系数和核参数的问题,利用改进的蝙蝠算法对这两个参数进行寻优,确定最佳的模型参数;另一方面,考虑到 AdaBoost. RT 算法存在对阈值敏感的不足,提出一种自适应阈值的 AdaBoost. RT 算法。

1 核极限学习机

极限学习机(Extreme Learning Machine, ELM)是一种新型性能优良的单隐层前向神经网络。与传统的神经网络相比, ELM 神经网络只需要一步即可确定网络的输出权值,极大地提高了网络的泛化能力和学习效率,具有良好的非线性拟合能力,算法的计算复杂度和搜索空间也大大减小^[11]。ELM 神经网络的结构如图 1 所示。

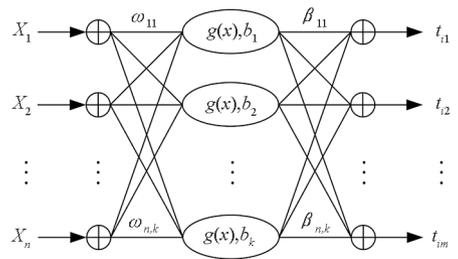


图 1 ELM 神经网络结构

Fig. 1 Structure of ELM neural network

假设有 N 个训练样本为 $\{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^N$, 样本输入 $\mathbf{x}_i = [x_{i1} \ x_{i2} \ \cdots \ x_{in}]^T \in \mathbb{R}^n$, 样本真实输出 $\mathbf{t}_i = [t_{i1} \ t_{i2} \ \cdots \ t_{im}]^T \in \mathbb{R}^m$ 。对于含有 K 个隐含层节点且激励函数为 $g_i(\mathbf{x}_i)$ 的 ELM 神经网络,其输出可以表示为:

$$y_j = \sum_{i=1}^K \beta_{ij} g_i(\boldsymbol{\omega}_i \cdot \mathbf{x}_j + b_i), \quad j = 1, 2, \dots, N \quad (1)$$

式中, $\boldsymbol{\omega}_i$ 为第 i 个隐含层节点与输入层节点之间的权值向量; b_i 为 ELM 的第 i 个隐含层节点的阈值; β_{ij} 为隐含层节点与输出层之间的权值; y_j 为 ELM 的输出值; $g_i(\boldsymbol{\omega}_i \cdot \mathbf{x}_j + b_i)$ 为第 i 个隐含层节点的激活函数。

通过样本来训练 ELM 神经网络,从而找到最

佳的网络参数 ω 和 β , 使得网络的预测输出与实际输出的误差趋近于零, 即 $\sum_{i=1}^N \|y_i - t_i\| = 0$ 。此时有:

$$t_j = \sum_{i=1}^K \beta_i g_i(\omega_i \cdot x_j + b_i), j = 1, 2, \dots, N \quad (2)$$

上述的 N 个方程可以表述为:

$$\begin{aligned} \mathbf{H}\boldsymbol{\beta} &= \mathbf{T} \quad (3) \\ \mathbf{H}(\omega_1, \dots, \omega_K, b_1, \dots, b_K, x_1, \dots, x_N) &= \\ \begin{bmatrix} g_1(\omega_1 \cdot x_1 + b_1) & g_1(\omega_2 \cdot x_1 + b_2) & \dots & g_1(\omega_K \cdot x_1 + b_K) \\ g_1(\omega_1 \cdot x_2 + b_1) & g_1(\omega_2 \cdot x_2 + b_2) & \dots & g_1(\omega_K \cdot x_2 + b_K) \\ \vdots & \vdots & \dots & \vdots \\ g_1(\omega_1 \cdot x_N + b_1) & g_1(\omega_2 \cdot x_N + b_2) & \dots & g_1(\omega_K \cdot x_N + b_K) \end{bmatrix} \end{aligned} \quad (4)$$

式中: \mathbf{H} 为 ELM 神经网络的隐含层输出矩阵; \mathbf{T} 为样本真实值构成的矩阵。

通过求解 $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}$ 的最小二乘解, 即可得到最优的权值 $\boldsymbol{\beta}^*$ 使得实际值与预测值之间的误差趋近于零。根据求解广义逆的相关理论, 最优权值可解算为:

$$\boldsymbol{\beta}^* = \mathbf{H}^+ \mathbf{T} \quad (5)$$

式中, \mathbf{H}^+ 为 ELM 隐含层输出矩阵的 Moore-Penrose 广义逆。其值可以通过正交投影法或者采用奇异值分解 (Singular Value Decomposition, SVD) 等方法解算得到^[12]。

在对 ELM 网络进行训练时, 如果 ELM 神经网络的隐含层节点数与样本数相等, ELM 能够零误差拟合所有训练样本。由于在实际应用中需要考虑算法的实时性, 故隐含层的节点数通常取值小于训练样本数目, 训练样本可能存在复共线性问题^[13], 从而导致 ELM 神经网络的稳定性和泛化能力都不太理想。

为了进一步增强 ELM 算法的泛化能力和稳定性, Huang 等^[14-15] 通过对比分析 ELM 和支持向量机 (Support Vector Machines, SVM) 的机理, 提出将核函数引入 ELM, 构造 KELM^[16]。

1) 基于 Mercer's 条件定义核矩阵:

$$\begin{cases} \boldsymbol{\Omega}_{\text{ELM}} = \mathbf{H}\mathbf{H}^T \\ \boldsymbol{\Omega}_{i,j} = h(x_i) \cdot h(x_j) = K(x_i, x_j) \end{cases} \quad (6)$$

式(6)用核矩阵 $\boldsymbol{\Omega}$ 来代替 ELM 中的随机矩阵 $\mathbf{H}\mathbf{H}^T$, 利用核函数将 ELM 网络的所有输入样本从低维度空间映射到高维度隐含层特征空间。在核函数的参数设定之后, 核矩阵的映射是定量。 $h(x)$ 为 ELM 网络隐含层节点的输出函数; 核函数 $K(\mu, v)$ 主要采用径向基函数 (Radial Basis

Function, RBF) 核函数、线性核函数和多项式核函数等, 通常设定为 RBF 核函数。

$$K(\mu, v) = \exp[-(\mu - v)^2 / \sigma] \quad (7)$$

2) 将参数 I/C 增加到单位对角矩阵 $\mathbf{H}\mathbf{H}^T$ 中的主对角线元素上, 使得单位对角矩阵的特征根不为零, 再基于此确定权值向量 $\boldsymbol{\beta}^*$ 。这样的处理使得 ELM 具有更好的稳定性和泛化能力。此时的 ELM 神经网络的输出权值可描述为:

$$\boldsymbol{\beta}^* = \mathbf{H}^T (\mathbf{I}/C + \mathbf{H}\mathbf{H}^T)^{-1} \mathbf{T} \quad (8)$$

式中: \mathbf{I} 为对角矩阵; C 为惩罚系数, 主要用作平衡结构风险和经验风险之间的比例; $\mathbf{H}\mathbf{H}^T$ 是通过核函数将输入样本进行映射得到的。

由以上公式可以得到 KELM 的输出为:

$$\begin{aligned} f(x) &= h(x) \mathbf{H}^T (\mathbf{I}/C + \mathbf{H}\mathbf{H}^T)^{-1} \mathbf{T} = \\ & \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix} (\mathbf{I}/C + \boldsymbol{\Omega}_{\text{ELM}})^{-1} \mathbf{T} \end{aligned} \quad (9)$$

式中, KELM 模型的输出权值可以描述为:

$$\boldsymbol{\beta} = (\mathbf{I}/C + \boldsymbol{\Omega}_{\text{ELM}})^{-1} \mathbf{T} \quad (10)$$

2 改进的集成学习算法

基于单一的 KELM 神经网络对目标机动轨迹进行预测时, 由于 KELM 神经网络的预测性能容易受到惩罚系数和核参数的影响, 为了提高目标机动轨迹预测的精度, 引入集成学习理论, 以改进的 AdaBoost. RT 算法为集成框架, 将 KELM 神经网络作为弱预测器, 提出一种基于 KELM-AdaBoost. RT 算法的目标机动轨迹预测模型。基本思想是预先设定一个阈值 φ , 利用样本数据对 KELM 神经网络进行训练, 然后根据 KELM 神经网络的预测结果来更新训练样本的权值, 保持各个弱预测器 KELM 的训练样本不变, 基于新的权重对弱预测器继续进行训练, 以此不断循环, 训练 M 轮得到 M 个弱预测器, 给每一个弱预测器的输出赋一个权值, 将所有弱预测器的输出加权得到最终预测结果。KELM-AdaBoost. RT 集成学习算法流程如图 2 所示。

由于 AdaBoost. RT 算法的性能受阈值的影响较大, 数值过大过小都会影响算法的性能, 因此阈值的确定难度较大。此外, 在样本权重进行更新时, 预测精度较低的样本在下一轮迭代中的权重保持不变, 导致样本之间的区分度不大。针对 AdaBoost. RT 算法存在的不足, 提出两点改进策略: 引入自适应阈值和增加预测误差大的样本权重。改进的 KELM-AdaBoost. RT 算法实施过程描述如下。

步骤 1: 确定样本数据和初始化 KELM 网络。

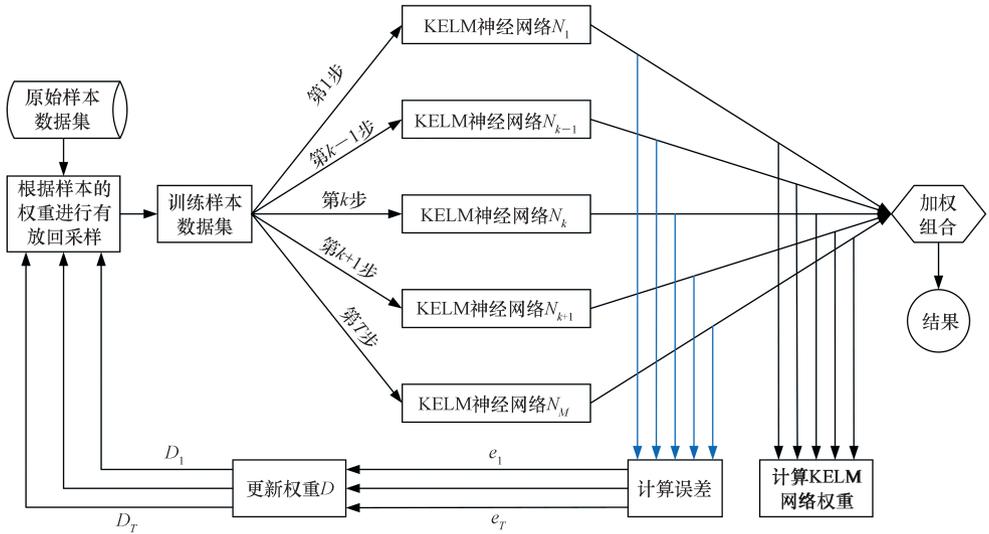


图 2 KELM-AdaBoost. RT 集成学习算法流程

Fig. 2 Flow chart of KELM-AdaBoost. RT integrated learning algorithm

从样本数据中随机选出 N 个样本数据 $\{(x_i, t_i)\}_{i=1}^N$, 根据构造的样本输入和输出维度确定 KELM 神经网络的结构。

步骤 2: 初始化训练样本的权值 $D_j(i) = 1/N$ ($i = 1, 2, \dots, N$), 设定算法的初始阈值 φ_j 和初始预测误差 e_j 。

步骤 3: 训练 KELM 弱预测器。利用样本数据训练第 k 个弱预测器, 计算每一个样本的误差 E_i^k 以及 KELM 神经网络的训练误差 ε_i 。

$$E_i^k = |h_j(x_i) - t_i| \quad (11)$$

$$\varepsilon_j = \sum D_j(i), i : E_i^j > \varphi_j \quad (12)$$

其中: $h_j(x_i)$ 为训练样本 i 第 j 次迭代计算的预测值; t_i 为真实值。

步骤 4: 更新预测误差阈值。根据 KELM 预测误差自适应调整阈值。基于 KELM 神经网络的训练误差的阈值自适应调整策略为:

$$\varphi_{j+1} = \begin{cases} \varphi_j - \lambda \cdot \max\left(\left|\frac{\varepsilon_j - \varepsilon_{j-1}}{\varepsilon_{j-1}}\right|, \delta\right), & \varepsilon_j < \varepsilon_{j-1} \\ \varphi_j + \lambda \cdot \max\left(\left|\frac{\varepsilon_j - \varepsilon_{j-1}}{\varepsilon_{j-1}}\right|, \delta\right), & \varepsilon_j \geq \varepsilon_{j-1} \end{cases} \quad (13)$$

步骤 5: 更新训练样本权值。计算第 $j+1$ 轮算法训练样本的权值, 其更新公式为:

$$D_{j+1}(i) = \begin{cases} [D_j(i)/B_j] \cdot \varepsilon_j^2, & E_i^j \leq \varphi_j \\ [D_j(i)/B_j] \cdot (1/\varepsilon_j^2), & E_i^j > \varphi_j \end{cases} \quad (14)$$

其中, B_j 为归一化因子。

步骤 6: 重复上述步骤 T 轮, 得到 T 个弱预测器 KELM, 每一个弱预测器的输出可以表示为

$f_t(x) (t = 1, 2, \dots, T)$, 加权得到强预测器输出:

$$f(x) = \frac{\sum_{t=1}^T [\ln(\frac{1}{e_t^2}) f_t(x)]}{\sum_{t=1}^T \ln(\frac{1}{e_t^2})} \quad (15)$$

3 改进的蝙蝠算法

由于 KELM 的性能与惩罚系数和核参数的选择密切相关, 为了提高 KELM 的预测性能, 引入改进的蝙蝠算法 (Improved Bat-inspired Algorithm, IBA) 对 KELM 的惩罚系数和核参数进行优化。蝙蝠算法 (Bat-inspired Algorithm, BA) 相对于经典的粒子群算法引入了局部搜索, 具有更好的搜索能力, 有助于算法跳出局部最优, 因此相对于传统的智能算法具有更好的优化能力。

蝙蝠算法是一种新的元启发式智能优化算法, 相较于粒子群算法和遗传算法^[17] 具有明显的优势。与粒子群算法相似, BA 随机初始化种群, 蝙蝠个体进行如下位置和速度更新:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta, \beta \in [0, 1] \quad (16)$$

$$V_i^t = V_i^{t-1} + (X_i^t - X^*)f_i \quad (17)$$

$$X_i^t = X_i^{t-1} + V_i^t \quad (18)$$

其中: X_i^t 为蝙蝠的位置; V_i^t 为蝙蝠的速度; β 为随机向量; X^* 为当前全局最优位置; f_i 为蝙蝠发出的脉冲频率; f_{\max} 和 f_{\min} 分别为最大、最小脉冲频率。

BA 的独特之处在于其结合了局部搜索。在进行局部搜索时, 蝙蝠个体将在最优个体附近采取随机搜索的方式进行局部寻优, 具体可以表示^[18]为:

$$X_{new} = X_{old} + \varepsilon A' \quad (19)$$

式中, ε 为取值在 0 和 1 之间的随机数; A' 为当前所有蝙蝠群体的平均响应。

随着算法进化程度的增加,蝙蝠个体的响应 A_i 和强度 r_i 也在不断变化,变化过程可以表述^[19]为:

$$A'_i = \alpha A_i^{-1} \quad (20)$$

$$r'_i = r_i^0 (1 - e^{-\gamma t}) \quad (21)$$

其中, α 和 γ 是常量。

每一次迭代计算之后都需要找到最佳位置 X^* ,直到满足算法的结束条件,输出全局最优位置。

蝙蝠算法与粒子群算法、遗传算法等群智能算法相比具有更好的优化性能,但是在收敛精度和收敛速度上还存在着不足。因此,为了提高蝙蝠算法的优化性能,避免算法陷入局部最优解,提高算法的收敛速度和精度,本文借鉴粒子群算法寻优过程^[20]。借鉴粒子群算法的粒子位置更新策略,以提高算法的局部搜索能力和优化精度^[21],对蝙蝠局部搜索策略做出以下改变:

$$X'_i = V'_i + \beta(X'_i - X'_k) + \rho(X'^{-1}_* - X'_i) \quad (22)$$

式中, β 和 ρ 为取值在 0 和 1 之间的随机数。

为了使算法的局部搜索能力和全局搜索能力达到平衡,将脉冲速率变化率 α 改进为动态变化的参数,借鉴模拟退火算法中冷却进程表中的冷却因素,对其进行如下改进^[22]:

$$\alpha_{new} = \left(\frac{1}{2}t\right)^{1/t} \alpha_{old} \quad (23)$$

4 基于改进的蝙蝠算法优化的 KELM-AdaBoost. RT 目标机动轨迹预测模型

针对目标机动轨迹预测问题,将改进的 AdaBoost. RT 算法与经过改进蝙蝠算法优化的 KELM 神经网络相结合,提出一种基于改进蝙蝠算法优化的 KELM-AdaBoost. RT 目标机动轨迹预测模型,流程如图 3 所示,具体实施步骤如下所示。

步骤 1: 输入训练样本数据,初始化蝙蝠算法以及 AdaBoost. RT 算法的参数。

步骤 2: 采用改进的蝙蝠算法优化 KELM 神经网络的参数。

步骤 3: 利用改进的蝙蝠算法优化的 KELM 神经网络进行训练,生成弱预测器,用训练好的弱预测器对所有训练样本进行预测并计算预测误差。基于弱预测器的预测误差对样本权重和算法

阈值进行调整,直至满足算法终止条件,得到基于若干个 KELM 弱预测器构成的强预测器。

步骤 4: 利用训练好的强预测器对目标未来机动轨迹进行预测,输出预测结果。

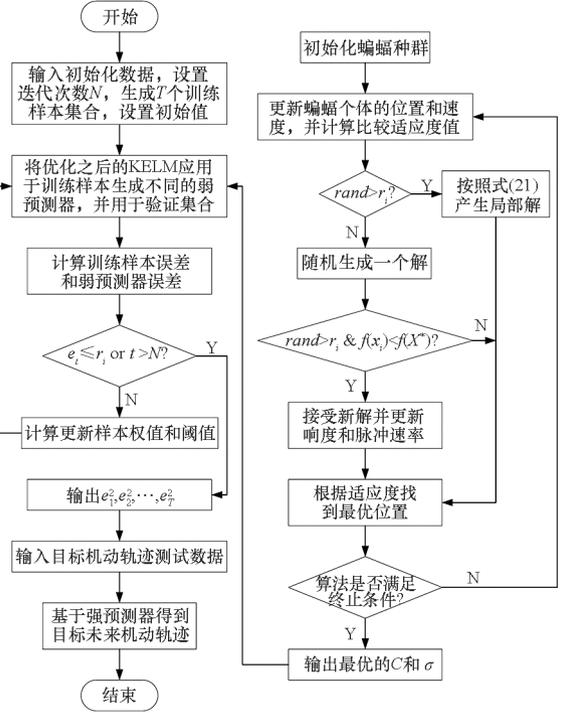


图 3 目标机动轨迹预测流程

Fig. 3 Flow diagram of the target maneuver trajectory forecasting

5 仿真分析

5.1 数据预处理

利用从空战对抗训练系统 (Air Combat Maneuvering Instrument, ACMI) 中提出的时间连续的 4 000 组飞行轨迹数据,数据之间的间隔为 0.25 s,每一个数据组包括目标的三维坐标。以对目标轨迹 x 坐标进行预测为例,假设利用目标 n 个历史轨迹数据,预测第 $n+1$ 时刻的目标轨迹 x 坐标。输入与输出之间的函数关系可表达为:

$$x_{1+n} = f(X_n) \quad (24)$$

式中: x_{1+n} 为第 $n+1$ 时刻模型的输出; X_n 为第 $n+1$ 时刻模型的输入。对于预测模型的输入数据,可以通过采用 C-C 法对历史目标机动轨迹 x 坐标时间序列进行相空间重构得到,可表示为:

$$X_n = [x_n \quad x_{n-\tau} \quad \cdots \quad x_{n-(m-1)\tau}] \quad (25)$$

式中: τ 为时间延迟; m 为嵌入维数。

5.2 实验设置

将目标机动三维坐标分离单独进行预测,以 x 坐标为例,采用 C-C 法^[23] 确定时间延迟和嵌

入维数,以确定的嵌入维数作为 KELM 神经网络的输入节点数。由于本文对目标机动轨迹进行单步预测,故输出节点数为 1。

各个算法的参数设置如下:KELM 神经网络的核函数采用 RBF 核函数,惩罚参数 C 和核参数 σ 通过改进的 IBA 算法寻优得到;IBA 的最大循环次数为 200,种群规模为 50,蝙蝠的最大和最小频率为 2 和 0,初始化响度为 1,脉冲速率为 0.5;KELM 弱预测器数目为 20,阈值初始化为 0.2。

将改进前后的 BA 性能进行了对比,证明了 IBA 的优越性;在此基础上构建 ELM 和 KELM 预测模型,结合 AdaBoost. RT 算法,分别建立基于 ELM-AdaBoost. RT 和 KELM-AdaBoost. RT 的预测模型,同时还构建基于 BP 和 SVM 的目标机动轨迹预测模型,仿真对比这六种预测模型性能优劣。

5.3 算法的优化性能对比

5.3.1 算法收敛准则

蝙蝠优化算法的收敛准则是基于整个群体收敛的,以最优群体为搜索目标,但是最终需要的是解空间中的一个最优个体,并不要求整个群体最优。为此基于算法中最优个体保存策略,可以观察到种群最优个体的进化情况。如果最优个体经过若干次进化没有被更新,或者变化的幅度较小,则可以认为算法已经收敛。可以将算法的收敛判断依据表示为:

$$|f(X_i^*) - f(X_{i-1}^*)| < \delta \quad (26)$$

式中: X_{i-1}^*, X_i^* 为算法相邻两次的最优个体; δ 为收敛阈值。

如果蝙蝠个体的全局最优位置的适应度值连续 ξ 次变化满足式(26),则认为算法已经收敛。

5.3.2 算法性能评价指标

为了更加客观准确地评价改进前后蝙蝠算法的性能差异,选取四个评价指标对算法性能进行评价^[24]。

1) 收敛率 (Converge Ratio, CR): 算法成功收敛次数与算法执行总次数的比值,反映算法的可收敛性。

2) 平均迭代次数 (Average Iteration Times, AIT): 对算法进行多次仿真实验,得到算法成功收敛时的平均迭代次数,反映算法的收敛速度。

3) 平均迭代运行时间 (Average Iteration Running Time, AIRT): 算法运行最大迭代次数所需的平均中央处理器 (Central Processing Unit,

CPU) 时间,反映算法的实时性。

4) 迭代次数标准偏差 (Iteration Times Standard Deviation, ITSD): 算法成功收敛时迭代次数的标准偏差,反映算法的收敛稳定性。

5.3.3 仿真实验与结果分析

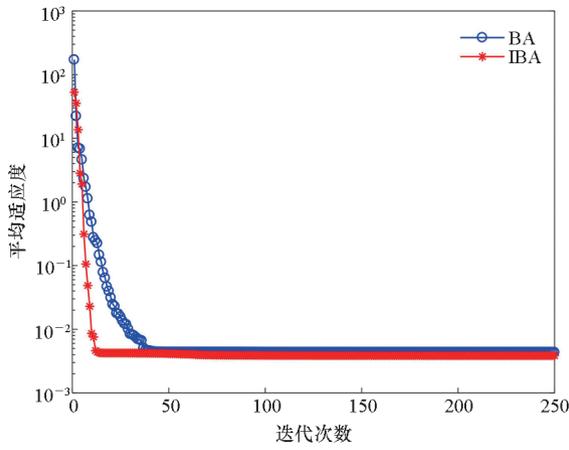
在对目标机动轨迹的三维坐标分别建立模型进行预测时,采用 IBA 对 KELM 神经网络进行优化。为了对比改进前后的蝙蝠算法性能变化,分别采用改进前后的蝙蝠对 KELM 神经网络进行 30 次优化仿真,算法性能对比见表 1。

表 1 算法性能对比

坐标	算法	CR/%	AIT	ITSD	AIRT/s
X	BA	83.3	186.80	847.07	15.86
	IBA	90.0	137.10	613.89	16.94
Y	BA	86.7	195.50	712.94	15.97
	IBA	96.7	125.60	262.49	16.91
Z	BA	93.3	187.10	1 178.00	16.13
	IBA	100.0	132.00	558.67	16.81

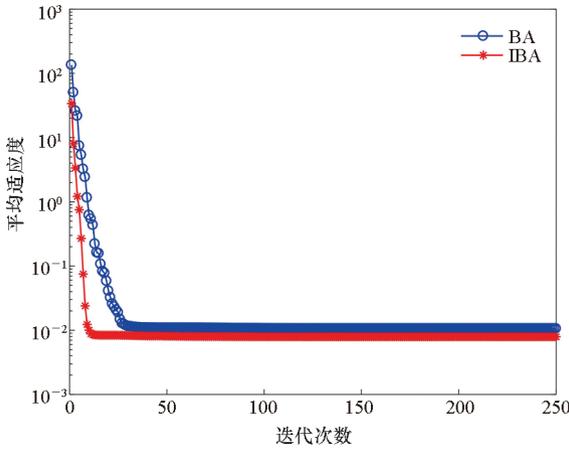
评价指标 CR 体现了算法的收敛性,CR 越大,这说明算法的收敛性越好,通过表 1 中 CR 的数值可以看出,改进策略能有效提高蝙蝠算法的收敛性;评价指标 AIT 体现了算法的收敛速度,AIT 越小,则算法的收敛速度越快,通过表中的 AIT 数值可知,改进之后的蝙蝠算法收敛速度明显加快;评价指标 ITSD 反映了算法性能的稳定性,通过表中数值可知,改进之后的蝙蝠算法优化同一个问题的性能更加稳定;评价指标 AIRT 反映了算法的实时性,通过表 1 可以看出,改进之后的算法实时性有一定程度的降低,这是由于改进之后的算法的复杂度变高了,从而影响了算法的实时性。

通过图 4 中算法的平均适应度下降曲线对算法的收敛过程做进一步分析。在图 4(a)中,BA 和 IBA 的适应度函数值达到稳定时的迭代次数分别是 87,72;在图 4(b)中,BA 和 IBA 的适应度函数值达到稳定时的迭代次数分别是 96,82;在图 4(c)中,BA 和 IBA 的适应度函数值达到稳定时的迭代次数分别是 113,68。这说明 IBA 收敛速度更快。通过图 4 可以看出,IBA 对 KELM 进行优化时的最终收敛数值都小于 BA,这说明 IBA 具有更佳的收敛精度,适用于优化 KELM 算法参数。



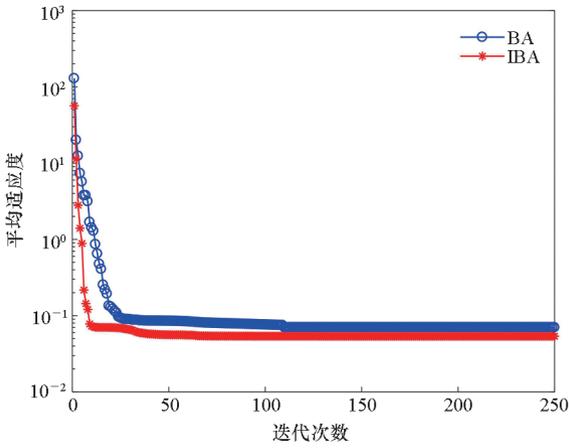
(a) X 坐标下平均收敛曲线对比

(a) Comparison of average convergence curve in X coordinate



(b) Y 坐标下平均收敛曲线对比

(b) Comparison of average convergence curve in Y coordinate



(c) Z 坐标下平均收敛曲线对比

(c) Comparison of average convergence curve in Z coordinate

图4 平均收敛曲线对比

Fig.4 Comparison of average convergence curve

5.4 预测模型仿真对比分析

利用 ACMI 记录的目标机动轨迹历史数据对 KELM 神经网络进行训练,使其具备预测目标未来机动轨迹的能力。不仅 KELM 神经网络的参数会对其学习和预测能力造成影响,KELM 神经网络的训练样本规模也会对其性能造成影响。因此,在确定 KELM 输入输出结构以及网络参数的基础上,研究和分析样本大小对预测误差的影响尤为重要。

为了更加客观地评价本文预测模型的有效性,从 ACMI 系统中提取两组数据作为预测模型训练和测试的样本数据,样本设置见表 2。

表2 仿真样本设置

Tab.2 Simulation sample settings

样本名称	样本属性	训练样本数量	测试样本数量
样本 1	小样本	1 499	1 000
	大样本	3 000	1 000
样本 2	小样本	300	1 000
	大样本	3 000	1 000

表 2 中大样本的训练数据包含了小样本的训练数据。根据六个预测模型的预测结果,绘制 BP, ELM, KELM, ELM-AdaBoost. RT, KELM-AdaBoost. RT 和 SVM 模型基于不同规模样本数据的预测结果对比图。仿真分为两个部分:第一个部分采用样本 1 中的大小样本,样本 1 中的大小样本的数量差异不大,通过该样本对比说明训练样本的变化是否会对模型的预测结果造成影响;第二个部分采用样本 2 进行仿真实验,样本 2 的大小样本的训练数据存在量级差异,通过该样本对比说明样本的量级差异将会对模型的预测结果产生一定的影响。

为了进一步检测目标机动轨迹预测模型的预测效果,用平均绝对误差 (Mean Absolute Error, MAE)、均方误差 (Mean Square Error, MSE)、归一化均方误差 (Normalized Mean Square Error, NMSE) 以及相关系数 (Correlation coefficient, Cor) 等指标对模型预测性能进行评价,定义如下所示:

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - \hat{x}_i| \quad (27)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2 \quad (28)$$

$$NMSE = \frac{\sum_{i=1}^n (x_i - \hat{x}_i)^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (29)$$

$$Cor = \frac{\sum_{i=1}^n |x_i - \bar{x}| \cdot |\hat{x}_i - \bar{\hat{x}}|}{n \cdot \sigma(x) \cdot \sigma(\hat{x})} \quad (30)$$

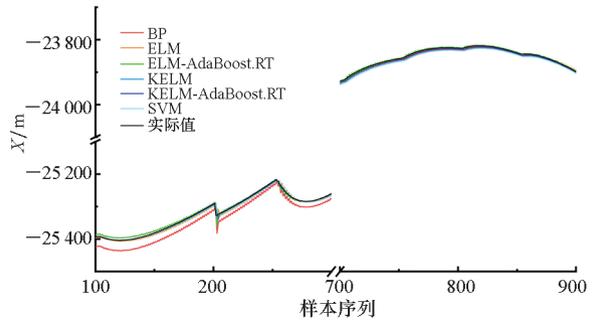
其中： \hat{x}_i 为模型的第 i 个预测值； x_i 为第 i 个目标真实轨迹数值； \bar{x} 为目标机动轨迹的平均值； $\bar{\hat{x}}$ 为目标机动轨迹预测值的平均值； $\sigma(\hat{x})$ 为目标机动轨迹预测值 \hat{x} 的标准差； $\sigma(x)$ 为目标机动轨迹 x 的标准差。显然可以看出，性能评价指标的数值越小，模型的预测精度越高。

5.4.1 样本数量差异仿真

图 5 ~ 7 为目标预测结果对比情况。通过图 5 ~ 7 中的 BP、SVM、ELM 以及 KELM 模型的预测结果以及误差对比可以看出，四种预测模型的预测值与目标机动轨迹的实际值的变化大体上保持一致，这说明四种基本模型可以对目标机动轨迹进行有效预测。但是从整体上可以看出，基本的 KELM 模型预测结果相较于其他模型更加接近真实值。

通过图 5 ~ 7 中的 ELM、ELM-AdaBoost. RT、KELM 以及 KELM-AdaBoost. RT 模型的预测结果以及误差对比可以看出，结合了 AdaBoost. RT 算法的 ELM 和 KELM 模型的预测结果相较于基本的 ELM 和 KELM 模型而言误差更小，且在目标机动轨迹波动时也能对其进行很好的预测，这些都说明 AdaBoost. RT 算法能够有效提高弱预测器的预测性能，并且具有更好的泛化性能。

通过图 5 ~ 7 中的图 (a) 和图 (b) 对比可以看出，训练样本较小差异将会对模型预测结果造成影响，同时也可以看出本文预测模型可以处理样本较少的情况。通过曲线对比可以初步看出，训练样本规模较大时，模型的预测结果更加精确，并且在数据波动时的预测效果也表现得更好一些。

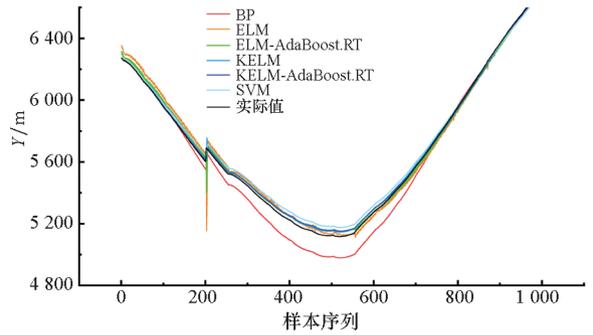


(b) 样本量为 3 998

(b) Sample size was 3 998

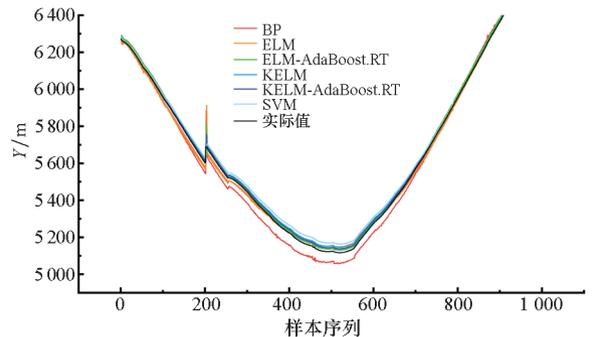
图 5 目标 X 坐标预测结果对比

Fig.5 Comparison of prediction results in target X coordinate



(a) 样本量为 2 499

(a) Sample size was 2 499

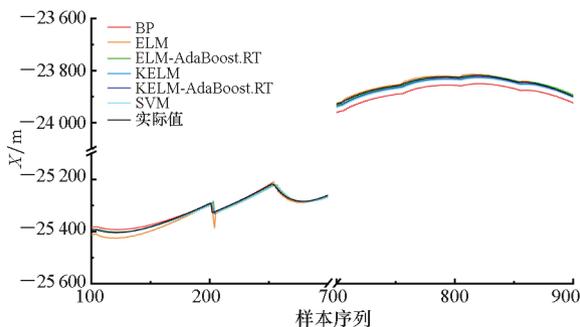


(b) 样本量为 3 998

(b) Sample size was 3 998

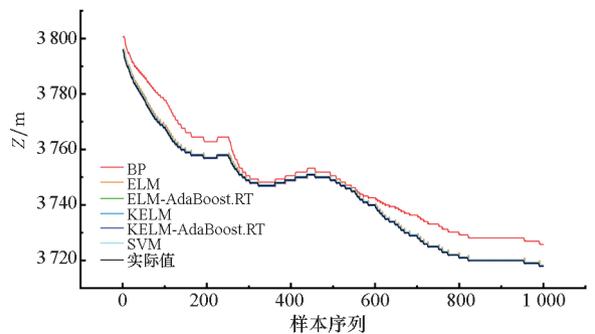
图 6 目标 Y 坐标预测结果对比

Fig.6 Comparison of prediction results in target Y coordinate



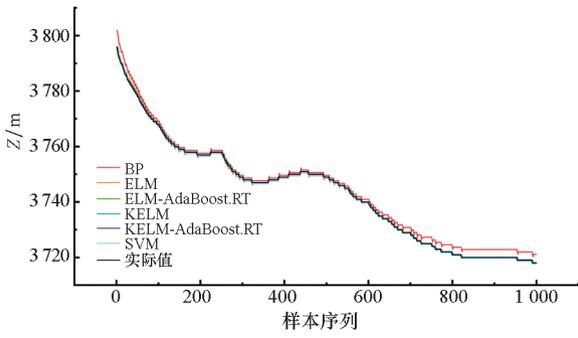
(a) 样本量为 2 499

(a) Sample size was 2 499



(a) 样本量为 2 499

(a) Sample size was 2 499



(b) 样本量为 3 998

(b) Sample size was 3 998

图 7 目标 Z 坐标预测结果对比

Fig. 7 Comparison of prediction results in target Z coordinate

为了进一步说明本文预测算法的预测效果,表 3 ~ 5 定量比较了 BP, ELM, KELM, ELM-AdaBoost. RT, ELM-AdaBoost. RT 和 SVM 模型基于不同规模样本数据的预测性能。

通过表 3 ~ 5 数据可以发现,与小样本的预测结果相比,较大的样本量将导致神经网络在预测过程中生成的 *MAE*、*MSE* 和 *NMSE* 略有减小,同时相关系数 *Cor* 也略有增加,这说明预测结果与真实值之间的接近程度高,可以很好地反映真实数据的情况。通过数据分析可知, *MAE*、*MSE*、*NMSE* 算法性能评价指标与样本量的大小呈负相关,样本容量越大,评价指标的数值越小;而相关系数 *Cor* 与样本量成正相关,这些都可以说明样本的大小将会对预测性能产生一定的影响。从算

表 3 不同预测模型对目标 X 坐标预测性能对比

Tab. 3 Performance comparison of different prediction models for target X coordinate prediction

样本规模	算法	MAE/m	MSE/m	NMSE/m	Cor
2 499	SVR	9.064 9	116.975 6	$3.035 8 \times 10^{-4}$	0.999 0
	ELM	7.918 7	101.255 4	$2.627 9 \times 10^{-4}$	0.998 9
	KELM	7.356 4	78.345 6	$2.033 3 \times 10^{-4}$	0.999 0
	ELM-AdaBoost. RT	6.492 2	62.988 3	$1.634 7 \times 10^{-4}$	0.998 9
	KELM-AdaBoost. RT	5.824 5	49.701 6	$1.289 8 \times 10^{-4}$	0.999 0
	BP	17.212 2	44.305 1	$1.000 0 \times 10^{-3}$	0.998 9
3 998	SVR	6.927 7	69.474 5	$1.803 1 \times 10^{-4}$	0.998 9
	ELM	5.511 3	55.840 9	$1.449 2 \times 10^{-4}$	0.998 9
	KELM	4.664 2	32.154 6	$8.345 0 \times 10^{-5}$	0.999 0
	ELM-AdaBoost. RT	5.364 1	44.659 8	$1.159 1 \times 10^{-4}$	0.998 9
	KELM-AdaBoost. RT	4.352 1	28.331 7	$7.352 9 \times 10^{-5}$	0.999 0
	BP	11.012 8	213.090 4	$5.530 3 \times 10^{-4}$	0.998 9

表 4 不同预测模型对目标 Y 坐标预测性能对比

Tab. 4 Performance comparison of different prediction models for target Y coordinate prediction

样本规模	算法	MAE/m	MSE/m	NMSE/m	Cor
2 499	SVR	0.435 1	0.301 1	$8.780 5 \times 10^{-4}$	0.998 8
	ELM	0.484 3	0.549 6	$1.600 0 \times 10^{-3}$	0.998 6
	KELM	0.392 7	0.232 2	$6.771 8 \times 10^{-4}$	0.998 9
	ELM-AdaBoost. RT	0.309 7	0.250 2	$7.294 9 \times 10^{-4}$	0.998 8
	KELM-AdaBoost. RT	0.274 6	0.108 0	$3.150 4 \times 10^{-4}$	0.998 9
	BP	5.580 5	39.957 6	$116.500 0 \times 10^{-3}$	0.9861
3 998	SVR	0.382 6	0.171 6	$5.002 6 \times 10^{-4}$	0.998 8
	ELM	0.365 4	0.243 1	$7.089 6 \times 10^{-4}$	0.998 7
	KELM	0.287 3	0.113 1	$3.298 0 \times 10^{-4}$	0.998 9
	ELM-AdaBoost. RT	0.263 0	0.179 8	$5.241 59 \times 10^{-4}$	0.998 8
	KELM-AdaBoost. RT	0.171 8	0.089 2	$2.602 0 \times 10^{-4}$	0.998 9
	BP	1.583 1	3.744 5	$10.900 0 \times 10^{-3}$	0.997 3

表 5 不同预测模型对目标 Z 坐标预测性能对比

Tab. 5 Performance comparison of different prediction models for target Z coordinate prediction

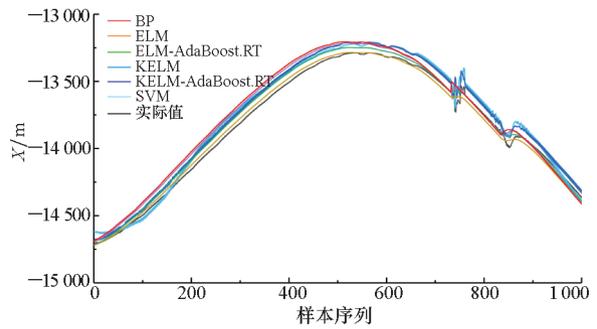
样本规模	算法	MAE/m	MSE/m	NMSE/m	Cor
2 499	SVR	32.235 0	$1.375 2 \times 10^3$	0.006 8	0.998 9
	ELM	31.581 3	$1.663 3 \times 10^3$	0.008 2	0.995 3
	KELM	29.028 9	$1.272 9 \times 10^3$	0.006 3	0.998 8
	ELM-AdaBoost. RT	20.208 0	$6.016 8 \times 10^3$	0.003 0	0.997 9
	KELM-AdaBoosta. RT	14.658 1	$3.304 1 \times 10^2$	0.001 6	0.998 8
	BP	60.975 8	$6.486 7 \times 10^3$	0.031 9	0.997 6
3 998	SVR	23.244 7	774.127 6	0.003 8	0.998 9
	ELM	15.314 0	390.457 7	0.001 9	0.998 2
	KELM	13.717 0	287.018 3	0.001 4	0.998 9
	ELM-AdaBoost. RT	11.818 1	220.449 6	0.001 1	0.998 7
	KELM-AdaBoost. RT	9.387 2	145.297 1	0.000 7	0.999 0
	BP	36.833 7	1924.900 0	0.009 5	0.998 3

法预测机理上对上述现象进行分析可知,较大的样本量可以更好地训练预测模型,使得模型可以更好地学习和了解目标历史机动轨迹的内在规律,从而可以实现对目标未来机动轨迹更精确的预测。因此,样本量越大,训练之后的预测模型中包含的目标运动的历史信息就越多,误差也就越小。

通过表 3~5 可以看出,结合了 AdaBoos. RT 算法的 ELM 和 KELM 模型的预测性能相较于基本的 ELM 和 KELM 模型而言更优,这些都充分说明 AdaBoos. RT 算法能够有效提高弱预测器的预测性能。

5.4.2 样本量级差异仿真

为了进一步说明训练样本的规模差异将对模型的预测结果造成一定程度的影响,利用样本 2 进行仿真验证。在进行仿真实验时,将大小样本的规模设置为 300 和 3 000,从而体现大小样本之间的量级差异对模型预测结果的影响。基于大小样本采用六种预测模型的预测结果对比如图 8~10 所示。



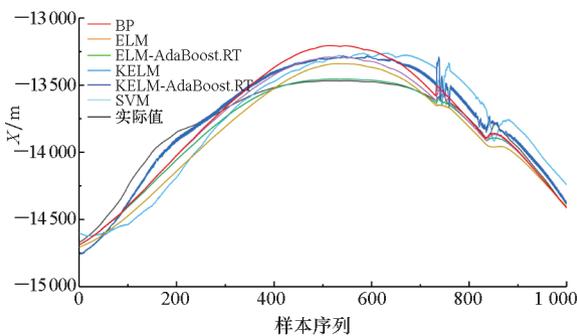
(b) 样本量为 3 000

(b) Sample size was 3 000

图 8 目标 X 坐标预测结果对比

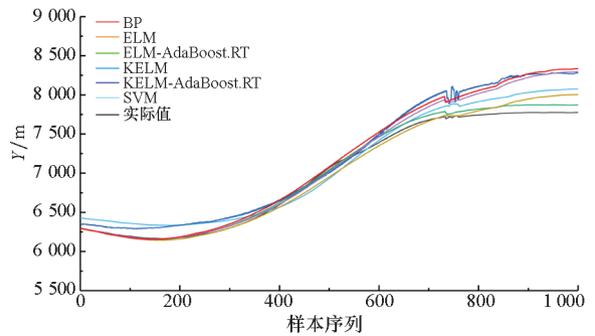
Fig. 8 Comparison of prediction results in target X coordinate

通过图 8~10 的对比结果,结合 5.4.1 节中的仿真结果可以看出,不论是样本规模存在量级差异,还是仅仅存在一定的数量差异,都会对模型的预测结果产生一定的影响。通过模型的预测性能的对比如分析,可以进一步分析样本规模的量级差异对模型预测效果的影响程度。表 6~8 给出了六种预测模型基于不同规模样本的预测性能。



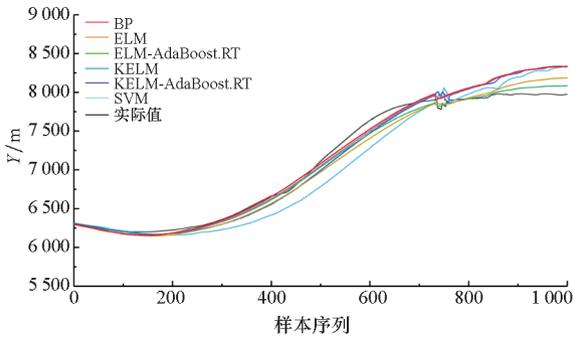
(a) 样本量为 300

(a) Sample size was 300



(a) 样本量为 300

(a) Sample size was 300

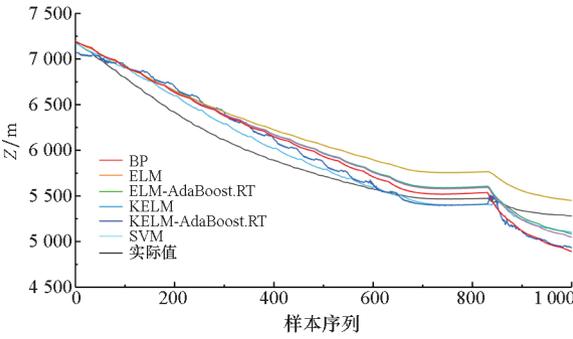


(b) 样本量为 3 000

(b) Sample size was 3 000

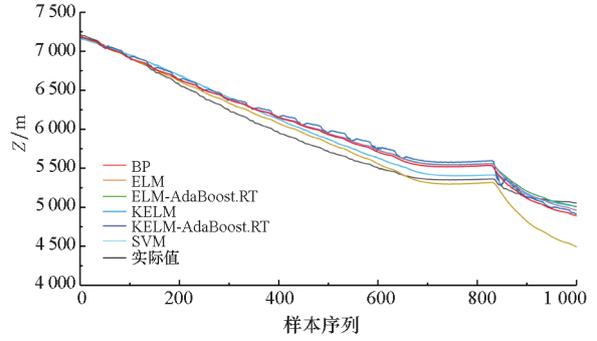
图 9 目标 Y 坐标预测结果对比

Fig. 9 Comparison of prediction results in target Y coordinate



(a) 样本量为 300

(a) Sample size was 300



(b) 样本量为 3 000

(b) Sample size was 3 000

图 10 目标 Z 坐标预测结果对比

Fig. 10 Comparison of prediction results in target Z coordinate

通过对比表 3~5 与表 6~8 中对应预测模型的性能指标,可以看出,样本规模差异越大,模型的预测性能差异也越明显。通过对比指标数据可以看出,样本的规模的不同导致了部分算法的 *MSE* 和 *NMSE* 指标出现量级差异,这都充分说明了训练样本的规模将会对模型的预测性能造成较大的影响。同时,也可以看出本文预测模型可以处理样本的较少的情况,模型的泛化能力较强。本质上,样本的大小反映了模型可以学习的历史信息的多少,历史信息越多,模型的预测结构越完整,泛化能力越强,从而模型的预测性能也越好。

表 6 不同预测模型对目标 X 坐标预测性能对比

Tab. 6 Performance comparison of different prediction models for target X coordinate prediction

样本规模	算法	MAE/m	MSE/m	NMSE/m	Cor
300	SVR	96.683 7	$1.139 8 \times 10^4$	0.005 9	0.995 9
	ELM	115.712 7	$1.685 4 \times 10^4$	0.087 6	0.956 1
	KELM	87.065 0	$1.499 5 \times 10^4$	0.078 0	0.992 7
	ELM-AdaBoost. RT	60.891 6	$4.797 2 \times 10^3$	0.024 9	0.987 8
	KELM-AdaBoost. RT	31.803 3	$1.998 8 \times 10^3$	0.010 4	0.998 5
	BP	111.768 3	$2.028 9 \times 10^4$	0.105 5	0.972 8
3 000	SVR	68.178 8	$5.691 5 \times 10^3$	0.029 6	0.996 5
	ELM	55.223 1	$4.248 5 \times 10^3$	0.022 1	0.988 9
	KELM	38.992 1	$2.046 2 \times 10^3$	0.010 6	0.996 9
	ELM-AdaBoost. RT	47.424 1	$2.896 8 \times 10^3$	0.015 1	0.991 6
	KELM-AdaBoost. RT	11.080 7	$1.774 5 \times 10^2$	0.000 9	0.998 8
	BP	78.942 1	$8.567 6 \times 10^3$	0.044 6	0.989 4

表 7 不同预测模型对目标 Y 坐标预测性能对比Tab. 7 Performance comparison of different prediction models for target Y coordinate prediction

样本规模	算法	MAE/m	MSE/m	$NMSE/m$	Cor
300	SVR	144.272 8	$3.319 1 \times 10^4$	0.050 4	0.998 7
	ELM	139.557 1	$2.303 7 \times 10^4$	0.035 0	0.994 3
	KELM	117.429 0	$3.468 1 \times 10^4$	0.052 6	0.993 5
	ELM-AdaBoost. RT	65.280 3	$6.074 4 \times 10^3$	0.009 2	0.996 2
	KELM-AdaBoost. RT	35.639 2	$1.665 6 \times 10^3$	0.002 5	0.998 7
	BP	157.940 5	$5.605 4 \times 10^4$	0.085 1	0.991 2
3 000	SVR	86.752 9	$1.028 8 \times 10^4$	0.015 6	0.998 6
	ELM	126.840 9	$2.539 8 \times 10^4$	0.038 6	0.989 7
	KELM	75.734 7	$1.156 6 \times 10^4$	0.017 6	0.998 0
	ELM-AdaBoost. RT	40.252 3	$2.591 1 \times 10^3$	0.003 9	0.997 7
	KELM-AdaBoost. RT	12.914 3	$2.618 9 \times 10^2$	0.000 4	0.998 9
	BP	90.630 0	$1.870 6 \times 10^4$	0.028 4	0.990 5

表 8 不同预测模型对目标 Z 坐标预测性能对比Tab. 8 Performance comparison of different prediction models for target Z coordinate prediction

样本规模	算法	MAE/m	MSE/m	$NMSE/m$	Cor
300	SVR	161.073 2	$5.018 9 \times 10^4$	0.126 9	0.991 9
	ELM	100.734 4	$1.246 8 \times 10^4$	0.031 5	0.987 7
	KELM	52.712 3	$5.506 1 \times 10^3$	0.013 9	0.998 3
	ELM-AdaBoost. RT	62.480 2	$5.695 6 \times 10^3$	0.014 4	0.994 8
	KELM-AdaBoost. RT	43.578 5	$3.745 5 \times 10^3$	0.009 5	0.998 5
	BP	176.258 2	$3.991 7 \times 10^4$	0.100 9	0.967 3
3 000	SVR	102.739 7	$1.913 1 \times 10^4$	0.048 4	0.997 6
	ELM	59.092 3	$4.717 2 \times 10^3$	0.011 9	0.994 0
	KELM	25.383 1	$1.676 2 \times 10^3$	0.004 2	0.998 6
	ELM-AdaBoost. RT	37.6401	$1.872 5 \times 10^3$	0.004 7	0.998 2
	KELM-AdaBoost. RT	20.080 0	$8.263 5 \times 10^2$	0.002 1	0.998 9
	BP	132.833 5	$2.955 0 \times 10^4$	0.074 7	0.998 1

6 结论

针对目标机动轨迹预测问题,借鉴集成学习理论,将集成学习理论与 KELM 神经网络相结合,构造一种基于 KELM 的强预测器模型。基于 ACMI 记录的目标历史机动轨迹数据,对模型进行训练学习,进而将其应用于目标机动轨迹预测。基于理论分析和仿真验证,可以得到以下结论:

1)从算法性能指标对比结果来看,KELM-AdaBoost. RT 模型以及 ELM-AdaBoost. RT 模型的预测结果相较于 KELM 预测模型和 ELM 预测模型好。这些说明基础学习理论可以有效提高弱预测器 KELM 的预测精度。

2)通过仿真充分验证了训练模型的样本大小将会对预测模型的性能产生较大的影响,训练样本越大,模型的预测性能越好。

3)多组仿真实验验证了基于改进 KELM 和集成学习理论的预测模型具有良好的泛化性能和预测性能,因此本文算法不仅可以应用于目标机动轨迹预测,还可以应用于其他空战问题研究。

参考文献 (References)

- [1] 寇英信,李战武,陈哨东,等.火控系统在航空作战中的作用:作战飞机之“魂”[J].电光与控制,2013,20(12):1-5.
KOU Yingxin, LI Zhanwu, CHEN Shaodong, et al. The important role of fire control system in air combat—soul of

- fighters[J]. *Electronics Optics & Control*, 2013, 20(12): 1-5. (in Chinese)
- [2] 姜佰辰, 关键, 周伟, 等. 基于多项式卡尔曼滤波的船舶航迹预测算法[J]. *信号处理*, 2019, 35(5): 741-746. JIANG Baichen, GUAN Jian, ZHOU Wei, et al. Vessel trajectory prediction algorithm based on polynomial fitting Kalman filtering [J]. *Journal of Signal Processing*, 2019, 35(5): 741-746. (in Chinese)
- [3] 赵帅兵, 唐诚, 梁山, 等. 基于改进卡尔曼滤波的控制河段船舶航迹预测[J]. *计算机应用*, 2012, 32(11): 3247-3250. ZHAO Shuaibing, TANG Cheng, LIANG Shan, et al. Track prediction of vessel in controlled waterway based on improved Kalman filter[J]. *Journal of Computer Applications*, 2012, 32(11): 3247-3250. (in Chinese)
- [4] 乔少杰, 韩楠, 朱新文, 等. 基于卡尔曼滤波的动态轨迹预测算法[J]. *电子学报*, 2018, 46(2): 418-423. QIAO Shaojie, HAN Nan, ZHU Xinwen, et al. A dynamic trajectory prediction algorithm based on Kalman filter[J]. *Acta Electronica Sinica*, 2018, 46(2): 418-423. (in Chinese)
- [5] 翟岱亮, 雷虎民, 李炯, 等. 基于自适应IMM的高超声速飞行器航迹预测[J]. *航空学报*, 2016, 37(11): 3466-3475. ZHAI Dailiang, LEI Humin, LI Jiong, et al. Trajectory prediction of hypersonic vehicle based on adaptive IMM[J]. *Acta Aeronautica et Astronautica Sinica*, 2016, 37(11): 3466-3475. (in Chinese)
- [6] 杨彬, 贺正洪. 一种GRNN神经网络的高超声速飞行器航迹预测方法[J]. *计算机应用与软件*, 2015, 32(7): 239-243. YANG Bin, HE Zhenghong. Hypersonic vehicle track prediction based on GRNN[J]. *Computer Applications and Software*, 2015, 32(7): 239-243. (in Chinese)
- [7] 谭伟, 陆百川, 黄美灵. 神经网络结合遗传算法用于航迹预测[J]. *重庆交通大学学报(自然科学版)*, 2010, 29(1): 147-150. TAN Wei, LU Baichuan, HUANG Meiling. Track prediction based on neural networks and genetic algorithm[J]. *Journal of Chongqing Jiaotong University (Natural Science)*, 2010, 29(1): 147-150. (in Chinese)
- [8] 甘旭升, 端木京顺, 孟月波, 等. 基于粒子群优化的WNN飞行数据气动力建模[J]. *航空学报*, 2012, 33(7): 1209-1217. GAN Xusheng, DUANMU Jingshun, MENG Yuebo, et al. Aerodynamic modeling from flight data based on WNN optimized by particle swarm [J]. *Acta Aeronautica et Astronautica Sinica*, 2012, 33(7): 1209-1217. (in Chinese)
- [9] HAQUE A U, MANDAL P, NEHRIR H M, et al. A hybrid intelligent framework for wind power forecasting engine[C]// *Proceedings of IEEE Electrical Power and Energy Conference*, 2014: 184-189.
- [10] 雷达. 基于智能学习模型的民航发动机健康状态预测研究[D]. 哈尔滨: 哈尔滨工业大学, 2013. LEI Da. Civil aircraft engine health condition prediction based on intelligent learning models[D]. Harbin: Harbin Institute of Technology, 2013. (in Chinese)
- [11] 杨锡运, 关文渊, 刘玉奇, 等. 基于粒子群优化的核极限学习机模型的风电功率区间预测方法[J]. *中国电机工程学报*, 2015, 35(增刊1): 146-153. YANG Xiyun, GUAN Wenyuan, LIU Yuqi, et al. Prediction intervals forecasts of wind power based on PSO-KELM[J]. *Proceedings of the CSEE*, 2015, 35(Suppl 1): 146-153. (in Chinese)
- [12] HUANG G B, ZHU Q Y, SIEW C K. Extreme learning machine: theory and applications [J]. *Neurocomputing*, 2006, 70(1/2/3): 489-501.
- [13] 裴飞, 陈雪振, 朱永利, 等. 粒子群优化核极限学习机的变压器故障诊断[J]. *计算机工程与设计*, 2015, 36(5): 1327-1331. PEI Fei, CHEN Xuezhen, ZHU Yongli, et al. Transformer fault diagnosis based on particle swarm optimization and kernel-based extreme learning machine [J]. *Computer Engineering and Design*, 2015, 36(5): 1327-1331. (in Chinese)
- [14] HUANG G B. An insight into extreme learning machines: random neurons, random features and kernels[J]. *Cognitive Computation*, 2014, 6(3): 376-390.
- [15] CHI C, TAY W P, HUANG G B. Extreme learning machines for intrusion detection[C]// *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2012: 1-8.
- [16] HUANG G B, ZHOU H M, DING X J, et al. Extreme learning machine for regression and multiclass classification[J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2012, 42(2): 513-529.
- [17] YANG X. A new metaheuristic bat-inspired algorithm[J]. *Computer Knowledge and Technology*, 2010(284): 65-74.
- [18] GANDOMI A H, YANG X S. Chaotic bat algorithm[J]. *Journal of Computational Science*, 2014, 5(2): 224-232.
- [19] NWULU N I, XIA X H. Optimal dispatch for a microgrid incorporating renewables and demand response [J]. *Renewable Energy*, 2017, 101: 16-28.
- [20] MENG A B, CHEN Y C, YIN H, et al. Crisscross optimization algorithm and its application [J]. *Knowledge-Based Systems*, 2014, 67: 218-229.
- [21] LI G S, CHOU W S. Path planning for mobile robot using self-adaptive learning particle swarm optimization[J]. *Science China Information Sciences*, 2017, 61(5): 1-18.
- [22] 胡梦月, 胡志坚, 仇梦林, 等. 基于改进AdaBoost. RT和KELM的风功率预测方法研究[J]. *电网技术*, 2017, 41(2): 536-542. HU Mengyue, HU Zhijian, ZHANG Menglin, et al. Research on wind power forecasting method based on improved AdaBoost. RT and KELM algorithm [J]. *Power System Technology*, 2017, 41(2): 536-542. (in Chinese)
- [23] 王石, 隋永新, 董琰, 等. 基于改进型小数据量法的局域网流量预测[J]. *吉林大学学报(工学版)*, 2016, 46(4): 1254-1260. WANG Shi, SUI Yongxin, DONG Yan, et al. LAN traffic flow prediction using improved small data method[J]. *Journal of Jilin University (Engineering and Technology Edition)*, 2016, 46(4): 1254-1260. (in Chinese)
- [24] 敖永才, 师奕兵, 张伟, 等. 自适应惯性权重的改进粒子群算法[J]. *电子科技大学学报*, 2014, 43(6): 874-880. AO Yongcai, SHI Yibing, ZHANG Wei, et al. Improved particle swarm optimization with adaptive inertia weight[J]. *Journal of University of Electronic Science and Technology of China*, 2014, 43(6): 874-880. (in Chinese)