

基于 XGBoost 模型的地铁列车运行状态仿真方法

门元昊¹ 吴亮² 刘晓双¹ 秦泉喃¹ 罗森林¹

(1. 北京理工大学信息系统及安全对抗实验中心, 100081, 北京;

2. 通号城市轨道交通技术有限公司, 100081, 北京//第一作者, 硕士研究生)

摘要 基于物理模型及列车性能参数的地铁列车运行状态仿真方法存在着列车适配性差、参数调整成本高等问题。为了给 ATC(列车自动控制)系统的研究提供更为准确、高效的模拟试验平台,提出一种基于 XGBoost(极端梯度提升)算法的列车运行状态仿真模型构建方法。该方法可从实际运行数据中学习列车的控制与运行特性,可针对不同的线路环境、不同列车车型实现更准确的列车运行状态仿真。在某地铁线路采用实车进行试验,结果表明:该方法建立的列车运行状态仿真模型准确、有效,可满足实际应用需求。

关键词 地铁列车; 运行状态仿真; 列车自动运行; 机器学习; XGBoost 算法

中图分类号 U29-39

DOI:10.16037/j.1007-869x.2022.03.022

A Metro Train Running State Simulation Method Based on XGBoost Model

MEN Yuanhao, WU Liang, LIU Xiaoshuang, QIN Xiaonan, LUO Senlin

Abstract The simulation method of metro train running state based on physical model and train performance parameters has the problems of poor train adaptability and high cost of parameter adjustment. In order to provide a more accurate and efficient simulation test platform for the research of ATC (automatic train control) system, a method for constructing simulation model of train running state based on XGBoost (extreme gradient boosting) is proposed. This method can acquire control and running characteristics of the train from actual operation data, and achieve more accurate running state simulation for different line environments and different types of train. Real vehicle is launched on certain metro line for experiments. The results show that the simulation model of train running state established by this method is accurate and effective, and meets the needs of practical application.

Key words metro train; running state simulation; ATO (automatic train operation); machine learning; XGBoost algorithm

First-author's address Information System and Security &

Countermeasures Experimental Center, Beijing Institute of Technology, 100081, Beijing, China

列车运行状态仿真即通过建立真实运行环境的近似模型,模拟地铁列车在 ATC(列车自动控制)系统下的运行表现,可结合电子地图实现列车位置、速度的估计反馈,是测试列车控制系统的重要工具。仿真的目的是为 ATC 系统的研究提供模拟试验平台。由于 ATC 在设计、开发、测试等各个环节所需投入的时间长且系统不断更新迭代,在其研制过程中持续地在真实的生产环境中进行开发与调试是不现实的,因此,通常需要先建立列车运行状态的仿真模型,在仿真模型上开展相关系统的研发工作。目前,针对列车运行状态的仿真仍以传统的物理模型或改进的物理模型为主^[1-3],这些方法均对列车的受力及运动学模型进行了简化与近似,且面对不同的列车与线路环境时,需对照真实的运行数据采用人工方式来调整模型参数,存在着列车适配性差、仿真误差大等问题,难以高效、准确地模拟 ATC 在真实环境中的实际表现,不利于系统的评估与改进。

针对上述问题,基于机器学习的理论与方法,本文通过挖掘列车在目标环境下的运行表现,结合特征工程提高算法对 ATC 系统延时等特性的学习能力,基于 XGBoost(极端梯度提升)算法建立具备更好适应性与准确性的列车运行状态仿真模型,为 ATC 系统的研发提供基础技术支撑。

1 现有的列车运行状态仿真方法概述

现有的列车运行状态仿真系统一般以物理模型为基础,结合人工或求解式的参数调整方法,实现列车控制输出与实际表现间函数关系的拟合,而对加速度、速度、位置等描述列车运行状态的属性进行估计与仿真,常采用包括单质点模型、多质

点模型、绳体模型等描述方法拟合列车的动力学表现。文献[4]提出了一种针对高速列车运行的仿真方法,使用绳体模型对列车进行动力学建模,能够较好地模拟出列车质量分布均衡、动力分布分散的特点,准确模拟列车运行状态,但由于不同列车的质量与动力分布特点差异较大等原因,该方法难以适配不同型号的列车和复杂运行的线路。

综上,随着仿真技术的不断发展,基于物理模型的方法对输出动力的作用效果进行估算的准确性在持续提升,但仍缺乏适应 ATC 系统输出延时、适配不同列车性能特点的能力。因此,引入机器学习理论,研发可自动化利用真实运行数据,构建可准确拟合控制命令与列车表现间复杂函数关系的仿真模型,以提高对列车运行状态的仿真能力。

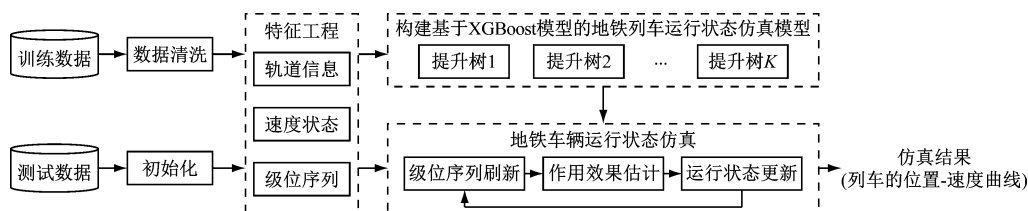


图1 基于 XGBoost 模型的地铁列车运行状态仿真方法原理

Fig.1 Principle of metro train running state simulation method based on XGBoost model

命令(包括历史命令信息)与实际加速度间的关系,从而实现当下1个采样时刻列车运行状态(借由速度、位置进行描述)的估计。由此建立可与 ATC 在电子地图辅助下进行交互反馈的列车运行状态仿真模型。

2.2 特征工程

训练数据的样本质量与特征空间是决定机器学习效果的关键要素之一,因此首先对原始数据中存在的完整数据(如采样遗漏等)、异常值(如日志错误等)进行清洗,提取出完整的站间运行数据,然后针对列车运行状态仿真任务,采用特征工程方法引导机器学习算法准确挖掘目标函数关系。

列车运行状态仿真模型的核心任务是:观察当前采样时刻的轨道坡度、列车速度、控制级位等数据,基于隐含了列车运动模型、控制延时特性、级位作用表现等信息的作用效果估计模型,预测列车在下1个采样时刻的速度表现。因此,特征空间应包含轨道信息、速度状态和级位序列,而训练标签可通过计算采样时刻间的列车速度变化获取。

综上,为建立可面向不同列车和不同线路环境、训练成本低、特征空间具有一定地铁环境数据

2 基于 XGBoost 算法的列车运行状态仿真方法

2.1 算法原理

本文提出一种基于 XGBoost 集成学习算法的列车运行状态仿真方法,其原理如图1所示。该方法基于真实的运行数据,首先针对列车运行过程中在不同速度、坡度及轨道状态等条件下施加相同的牵引/制动级位所产生不同加速度的作用效果,以及特定列车具有独特且伴有一定随机性的系统延时等问题,采用特征工程扩展优化原始数据维度,筛选并建立轨道信息、速度状态以及级位序列的特征子集,合并构成新的特征空间;在此基础上,利用 XGBoost 算法,学习任意采样时刻不同状态下控制

普适性的列车运行状态仿真模型,挖掘列车运动模型、控制延时特性、级位作用的表现特征,本文所述方法将当前时刻的坡度数据、速度信息、历史级位作为特征空间的主要组成。设 t 时刻列车的运行特征为 $x_t = \{e_t, v_t, l_t\}$, 其中: e_t 为 t 时刻列车所处位置的坡度; v_t 为 t 时刻列车运行速度; l_t 为 t 时刻列车输出的级位序列 $\{l_{t,0}, l_{t,1}, \dots, l_{t,k}\}$, $l_{t,k}$ 为 $t-k$ 时刻列车运行系统输出的级位。依据目标列车的经验最大延时周期数,将 k 设为 11, 历史时刻不足的采样点则实施补零操作。同时,将列车 $t+1$ 时刻与 t 时刻的速度差 a_t (即 t 时刻的加速度) 设为样本标签。最终构建出 13 维的特征向量及 1 维的样本标签,由此可依照有监督学习方法进行后续模型训练及验证。

2.3 模型构建

利用预处理后的数据集训练 XGBoost 模型,以实现采样点列车瞬时加速度的估计。XGBoost 算法是文献[5]提出的一种可扩展的端到端基于树的 Boosting(提升方法)家族算法,可通过构建多个弱学习器,将这些弱学习器组合形成强学习器。通常选取决策树等子模型作为此算法的弱分类器。XG-

Boost 算法因其优异的泛化性能被广泛应用于有监督学习领域,适合用于列车运行状态的仿真任务,相比 GBDT(梯度提升决策树)方法,XGBoost 算法在代价函数中加入了正则项,可有效控制模型的复杂度,同时支持多线程训练。采用 XGBoost 算法后模型的计算速度更快,可满足不同规模列车的运行数据仿真要求。

在模型训练中,XGBoost 算法利用坡度数据、速度信息、历史级位等参数自适应地挖掘列车行驶过程中的加速度表现和控制延时特点,以有效模拟列车在 ATC 系统下的真实运行表现。与基于列车输出级位和加速度间的对应关系表(以下简称“加速度表”)的传统物理模型相比,XGBoost 算法能够适应性地降低因环境因素干扰或列车实际参数与出厂参数不完全匹配等问题引发的仿真误差。XGBoost 算法的基本原理如下:

对于样本集 $N = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$,训练优化函数参数空间 F ,使得全局上目标函数 $O(F)$ 最小。目标函数可表示为:

$$O(F) = l(F) + \Omega(F) \quad (1)$$

式中:

$l(F)$ ——模型的损失函数,表示由样本的特征向量 \mathbf{x} 映射得到的模型预测结果 \hat{y} 与样本的真实标签 y 之间的误差;

$\Omega(F)$ ——正则化项,用以约束 F 的复杂性,控制模型的复杂度。

XGBoost 算法采用同 Boosting 算法家族一致的残差修正思想。针对每个样本,将现有估计与真实标签间的残差作为即将加入的新弱学习器的训练标签,由此可得到加入第 t 棵树后的估计结果为:

$$\hat{y}_i^{(t)} = \sum_{j=1}^{t-1} f_j(x_i) + f_t(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \quad (2)$$

式中:

x_i ——模型输入的第 i 条样本数据;

$f_i(x_i)$ 、 $f_t(x_i)$ ——分别为第 i 棵树和第 t 棵树的分类结果;

$\hat{y}_i^{(t-1)}$ 、 $\hat{y}_i^{(t)}$ ——分别为前 $t-1$ 棵和前 t 棵树的总分类结果。

将损失函数和正则化项代入式(1)目标函数中,可得第 t 次迭代时的目标函数为:

$$O^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (3)$$

式中:

n ——样本数量;

y_i ——第 i 条样本数据对应的真实标签;

$l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i))$ ——样本损失函数,表示 y_i 与加入第 t 棵树后总分类结果 $\hat{y}_i^{(t-1)} + f_t(x_i)$ 的误差;

$\Omega(f_t)$ ——正则化项,用以约束第 t 棵树 f_t 的复杂性,控制模型的复杂度。

对目标函数中的损失函数项进行二阶泰勒展开,并代入单个弱学习器的函数式与模型复杂度的函数式表达,能够得到展开的目标函数如下:

$$O^{(t)} \approx \sum_{j=1}^K \left[G_j w_j + \frac{1}{2} (H_j + \lambda) w_j^2 \right] + \gamma T \quad (4)$$

式中:

T ——叶子节点的个数;

w_j ——叶子节点权重;

K ——构建树的总棵数;

γ, λ ——均为调整参数,用以防止过拟合;

G_j ——损失函数一阶偏导的累加值之和;

H_j ——损失函数二阶偏导的累加值之和。

基于式(4)可计算每个叶子节点的最优权重和模型最小损失,从而在迭代过程中使用贪婪算法逐步训练 K 棵树,以组成具备强估计能力的集成模型。

2.4 状态仿真

地铁列车在运行过程中会依据地图信息和当前运行数据不断切换其行车状态,并输出相应的制动级位或牵引级位。在使用物理模型仿真时,常常会依据列车出厂时提供的加速度表和级位传输延时信息来预测计算当前时刻列车的加速度。具体的计算流程为:首先将列车系统输出的级位依次存到级位盒中,然后依据列车延时表获得当前作用在列车上的级位,接着根据加速度表和当前的列车速度找到对应的理论加速度,最后依据式(5)更新下一个周期的列车速度,并依据式(6)更新该周期内列车行驶距离。

$$v_{t+1} = v_t + a_t T_0 \quad (5)$$

$$s_t = \frac{v_{t+1}^2 - v_t^2}{2 a_t} \quad (6)$$

式中:

v_t —— t 时刻的列车速度;

T_0 ——列车运行的 1 个周期时长;

s_t —— t 到 $t+1$ 时刻内列车的运行距离。

由于列车型号的不一致使得在模拟仿真过程中需要不断地更换参数,所以难以用一套固定的模型参数实现准确的模拟仿真。此外,由于环境因素的干扰,使得列车在实际行驶过程出现与出厂参数不符的偏差。

综上,本文采用上述基于 XGBoost 算法构建的列车速度估计模型,替换传统物理模型的加速度计算模块。仿真时先对列车的线路信息及状态进行初始化,并进行与训练数据一致的特征工程处理;然后将采样点输入至 XGBoost 模型中,获得当前时刻的列车加速度,同理依据式(5)~(6)更新列车的速度和行驶距离;接着利用地图信息、ATC 系统输出更新的坡度信息和级位序列。上述步骤依次迭代进行,直到列车到站停车。最后,将列车在线路上的运行仿真数据进行记录,并分析其仿真效果。

3 基于 XGBoost 算法的列车运行状态仿真试验

3.1 试验目的

为验证基于 XGBoost 算法的列车运行状态仿真方法的有效性,本文对 3 列来自某地铁线路的真实列车数据进行学习及仿真测试,并与基于性能参数表的物理模型仿真方法进行比较。数据特征参数的具体说明如表 1 所示。

表 1 模型中采用的列车特征参数及其单位

Tab. 1 Train characteristics parameters and units used in model

属性特征	单位	属性特征	单位
速度	cm/s	坡度	cm/s ²
加速度	cm/s ²	采样周期	s
位置	cm	控制级位	级

注:在轨道电子地图的测绘过程中已将坡度转换为预计的列车运行加速度,故采用加速度的单位;一个采样周期为 0.2 s;控制级位为 -7 级~7 级。

3.2 评价方法

采用每个采样周期内加速度的均方根误差 R 和每站速度曲线误差百分比 E 两个指标,其计算式如式(7)、式(8)所示。 R 和 E 的值越小,表示模型的准确性越高,与实际列车运行情况的符合程度越高,即仿真效果越好。

$$R = \sqrt{\frac{1}{N} \sum_{i=1}^N (a_i - \hat{a}_i)^2}$$

(7)

$$E = \frac{1}{m} \sum_{j=1}^m \frac{|v_j - \hat{v}_j|}{v_j}$$

(8)

式中:

- N ——数据集中的数据总条数;
- a_i ——列车的实际加速度;
- \hat{a}_i ——列车的估计加速度;
- m ——线路上的数据采样数;
- v_j —— j 时刻列车的实际速度;
- \hat{v}_j ——采用模型计算得到的列车估计速度。

3.3 试验过程

首先对原始数据进行清洗,对加速度异常、缺失的数据点进行删减处理,然后依据算法原理中的特征工程方法逐条完成特征空间的建立。接着预留 1 个车站的数据(该站不参与模型训练)并用作单站列车运行速度曲线仿真,以验证模型的实用性。将其余站点的采样数据按照 7:2:1 的比例分别拆分至训练集、验证集和测试集内。

在进行列车运行状态仿真模型训练时,首先将训练集输入至 XGBoost 模型中,并用验证集进行超参数调整,最后使用测试集对仿真模型的性能进行分析评价。此外,将基于 XGBoost 模型的仿真结果与基于列车性能参数表物理模型得到的仿真结果进行对比。

基于验证集的表现,确定试验中采用的 XGBoost 模型超参数如下:最大树深度为 8,学习率为 0.6,基分类器数量为 500。在实用性检测时,将基于 XGBoost 算法的仿真模型应用到没参加模型训练的预留站上并进行模拟仿真,将预留站列车的实际运行速度、基于 XGBoost 模型得到的仿真速度、基于物理模型得到的仿真速度进行对比。

3.4 试验结果分析

参与试验的 3 列车的编号为 T1、T2、T3,数据量分别为 460 511 条、795 88 条、472 68 条。将基于物理模型得到的仿真结果与基于 XGBoost 模型得到的结果进行对比,如表 2 所示。从表 2 中可以看出,与基于物理模型得到的结果相比,基于 XGBoost 模型的列车运行状态仿真模型的 R 和 E 在多列车上的测试结果均较低,且在不同列车上的仿真性能表现更为稳定,这说明了基于 XGBoost 模型的仿真方法具有良好的环境适应性,能与不同的列车车型相匹配。

表 2 基于物理模型和基于 XGBoost 模型的仿真结果对比

Tab.2 Comparison of simulation results based on physical model and XGBoost model			
测试列车	仿真方法	R	E/%
T1	物理模型	2.774 1	39.65
	XGBoost 模型	1.045 2	6.75
T2	物理模型	2.926 0	42.31
	XGBoost 模型	0.872 8	5.09
T3	物理模型	2.786 7	41.41
	XGBoost 模型	1.000 7	6.47

将预留站列车的实际运行速度、基于 XGBoost 模型得到的仿真速度、基于物理模型得到的仿真速度进行对比,得到的结果如图 2 所示。从图 2 可以看出,基于物理模型得到的仿真速度在模拟控制级位变换较为频繁,且在复杂的目标速度调整(采样周期为 $(200 \sim 400) \times 0.2 \text{ s}$)及巡航速度控制(采样周期为 $(400 \sim 1\,300) \times 0.2 \text{ s}$)阶段均难以实现准确的速度估计,从而影响了整个车站的仿真效果。而基于 XGBoost 模型得到的仿真速度曲线则与列车的实际运行曲线更为贴合。

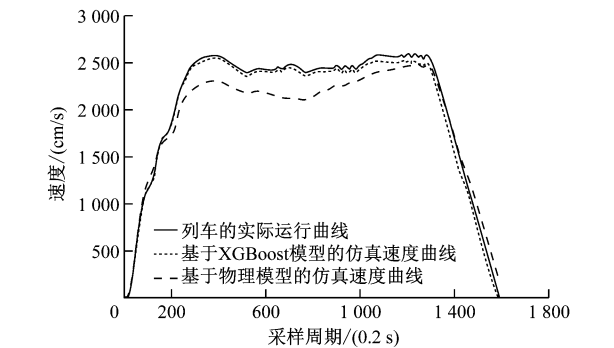


图 2 3 种不同测试工况下列车在单站运行时的速度曲线仿真对比

Fig.2 Speed curve simulation comparison of train entering single station under three different test conditions

综上所述,本文提出的基于 XGBoost 的列车运行速度仿真模型构建方法可从实际运行数据中有效挖掘列车运动模型、控制延时特性、级位作用表现等特征并加以利用,进而建立列车运行状态的仿真模型。该模型具有良好的环境适应性,能与不同的列车车型相匹配,可生成与实际情况接近的列车运行速度曲线,较为准确地实现对地铁列车运行状态的仿真。

4 结语

列车运行状态仿真是研发与优化 ATC 的重要工具,具备自动化适应能力与高精度仿真性能的状态仿真模型将助力轨道交通自动化技术的不断发展。本文提出的基于 XGBoost 模型的仿真方法可从实际运行数据中有效挖掘列车运动模型、控制延时特性、级位作用表现等特征,实现对 ATC 系统输出与实际作用效果间复杂函数关系的学习,进而建立具备一定适应性且模拟精度较高的列车运行状态仿真模型。试验采用真实数据验证了使用该方法所建立的仿真模型的有效性,且该方法适用于针对不同列车车型的运行状态仿真场景。未来应进一步结合其他先进的序列数据建模方法,不断改进、优化该模型的训练效果,为列车自动运行技术的发展提供更可靠、有效的技术支撑。

参考文献

[1] 何坤,肖壮,朱宇清,等. 基于模糊预测控制的城轨列车自动驾驶[J]. 现代计算机, 2017(35):28.
HE Kun, XIAO Zhuang, ZHU Yuqing, et al. Automatic train operation of metro train based on predictive fuzzy algorithm [J]. Modern Computer, 2017(35):28.

[2] 李坤阳,陈鸿辉,郭金松,等. 基于滑模自抗扰的高速列车自动驾驶算法研究[J]. 现代计算机, 2019(15):25.
LI Kunyang, CHEN Honghui, GUO Jinsong, et al. Research on automatic train operation algorithm for high-speed trains based on sliding mode active disturbance rejection control [J]. Modern Computer, 2019(15):25.

[3] 刘佳政,徐娟. 基于模糊预测控制的列车自动驾驶控制算法研究[J]. 控制与信息技术, 2018(1):7.
LIU Jiazheng, XU Juan. ATO controller research based on predictive fuzzy control algorithm[J]. Control and Information Technology, 2018(1):7.

[4] 唐金金,周磊山,佟路,等. 单列高速列车运行仿真模型与算法[J]. 中国铁道科学, 2012(3):111.
TANG Jinjin, ZHOU Leishan, TONG Lu, et al. Simulation model and algorithm for single high-speed train operation [J]. China Railway Science, 2012(3):111.

[5] CHEN T, GUESTRIN C. XGBoost: A scalable tree boosting system[C] // The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 13-17, 2016. San Francisco, California. New York: Association for Computing Machinery, 2016:785.

(收稿日期:2020-03-17)