计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2019,28(10):226−232 [doi: 10.15888/j.cnki.csa.007091]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 XGBoost 和 LSTM 加权组合模型在销售预测的应用①

冯　晨[1,2], 陈志德[1,3]

[1](福建师范大学 数学与信息学院, 福州 350007)
[2](福建省网络安全与密码技术重点实验室 (福建师范大学) 福州 350007)
[3](福建省网络与信息安全行业技术开发基地, 福州 350007)
通讯作者: 冯　晨, E-mail: zhidechen@fjnu.edu.cn

摘　要: 针对多变量的商品销售预测问题, 为了提高预测的精度, 提出了一种 ARIMA-XGBoost-LSTM 加权组合方法, 对具有多个影响因素的商品销售序列进行预测, 本文采用 ARIMA 做单变量预测, 将预测值作为新变量同其他变量一起放入 XGBoost 模型中进行不同属性的挖掘, 并将 XGBoost 的预测值合并到多变量序列中, 然后通过将新的多维数据转换为监督学习序列后利用 LSTM 模型进行预测, 将 3 种模型预测结果进行加权组合, 通过多次实验得出最佳组合的权值, 以此计算出最终的预测值. 数据结果表明, 基于 XGBoost 和 LSTM 的加权组合的多变量预测方法比单一的预测方法所得到的预测值更为精准.

关键词: ARIMA; LSTM; XGBoost; 时间序列; 组合模型预测

引用格式: 冯晨,陈志德.基于 XGBoost 和 LSTM 加权组合模型在销售预测的应用.计算机系统应用,2019,28(10):226–232. http://www.c-s-a.org.cn/1003-3254/7091.html

Application of Weighted Combination Model Based on XGBoost and LSTM in Sales Forecasting

FENG Chen[1,2], CHEN Zhi-De[1,3]

[1](College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350007, China)
[2](Fujian Provincial Key Laboratory of Network Security and Cryptology (Fujian Normal University), Fuzhou 350007, China)
[3](Fujian Provincial Network and Information Security Technology Development Base, Fuzhou 350007, China)

Abstract: Aiming at the multi-variable commodity sales forecasting problem, in order to improve the accuracy of prediction, an ARIMA-XGBoost-Lstm weighted combination method is proposed to predict the sales sequence of commodities with multiple influencing factors, In this study, ARIMA is used for univariate prediction. The predicted value is used as a new variable together with other variables in the XGBoost model for mining different attributes, and the predicted values of XGBoost are merged into the multivariate sequence, and then the new multidimensional data is converted. In order to supervise the learning sequence and use the LSTM model for prediction, the three model prediction results are weighted and combined, and the best combination weights are obtained through multiple experiments to calculate the final prediction value. The data results show that the multivariate prediction method based on the weighted combination of XGBoost and LSTM is more accurate than the prediction obtained by a single prediction method.

Key words: ARIMA; LSTM; XGBoost; time series; combined model prediction

① 基金项目: 国家自然科学基金 (61841701); 福建省自然科学基金 (2016J01287, 2018J01781); 电子信息与控制福建省高校工程研究中心开放基金 (EIC 1703); 广东省自然科学基金 (2019B010137002)
Foundation item: National Natural Science Foundation of China (61841701); Natural Science Foundation of Fujian Province (2016J01287, 2018J01781); Open Fund of Engineering Research Center of Higher Education of Fujian Province for Electronic Information and Control (EIC1703); Natural Science Foundation of Guangdong Province (2019B010137002)
收稿时间: 2019-03-11; 修改时间: 2019-04-04; 采用时间: 2019-04-16; csa 在线出版时间: 2019-10-15

随着经济全球化的发展, 企业面临着生产成本不断增加, 市场销售疲软等考验. 企业要想赢得市场竞争就需要对市场具有敏锐的嗅觉以及精准的决策, 从而控制成本, 降低损耗. 这使得企业借助精准高效的销售预测, 进而做出可靠的决策, 成为现代企业成功的重要手段[1]. 而对时变数据建模是数据科学中的一个基本问题, 应用于医学, 金融, 经济学, 气象学和客户支持中心操作等各个领域. 使用在时间序列数据上训练的模型来预测未来值是一个值得充分研究的领域, 其中应用了诸如 ARIMA 之类的传统线性统计模型[2]. 最近较为流行的是 RNN 模型的应用, 特别是在处理时间序列预测的问题上, LSTM 模型的选择记忆功能有着独特的优势[3,4]. Xgboost 是“极端梯度上升”的简称, 该算法既具有线性模型求解器和树学习算法的能力, 同时也有着可以在单机上并行计算的能力, 能够自动利用 CPU 的多线程进行并行计算, 同时精度也得到提升, 相比于其他提升树方法更为优越[5]. 目前在商品销售预测上的方法很多, 如果只用传统的线性回归方法来预测, 就忽视了非线性因素的影响, 近年来, 随着深度学习的不断研究与发展, 深度学习的模型被用于时间序列的预测, 例如循环神经网络 (RNN) 将时序的概念引入到网络结构中, 但 RNN 模型也有着几点关键性的不足, 梯度的消失和梯度爆炸问题, 长期记忆能力不足等问题, 进而提出了长短期记忆网络模型[6], 基于此, 可以采用不同层次的模型进行组合来提高模型的优越性和泛化性[7].

本文针对商品销售的预测进行研究, 销售预测是一个分析和报告信息的过程, 它能够为管理者提供信息分析和市场研究数据, 在此基础上进行经营决策, 用来解决一些特定的市场问题, 在当代市场竞争中发挥着重要的作用. 中国市场野蛮生长的红利期已经过去, 经济已从速度向质量转变, 企业竞争回归到成本、效率的问题上来. 这更需要利用数据以实现精细化运营, 然而销售预测需以具体产品为预测对象, 在实际场景中更存在着诸多重要的影响销售的因素. 因此, 为提高预测精度, 本文将销售策略量, 天气, 节假日等加入特征变量中, 建立包含相关预测特征条件的多变量模型, 结合 ARIMA 对平稳序列的较好的预测能力以及 LSTM 对序列非线性部分的出色拟合性能, 采用基于 XGBoost 和 LSTM 的加权组合模型进行预测, 提高预测的精度.

1 相关理论

1.1 ARIMA 模型

ARMA 模型, 即差分自回归移动平均模型, 如下结构可以简记为 ARMA(p, q):

$$x_t = \alpha_0 + \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \cdots + \alpha_p x_{t-p} + \varepsilon_t + \beta_1 \varepsilon_{t-2} + \cdots + \beta_q \varepsilon_{t-q} \tag{1}$$

当$p = 0$时, 是 AR(p) 模型:

$$x_t = \alpha_0 + \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \cdots + \alpha_p x_{t-p} + \varepsilon_t \tag{2}$$

当$q = 0$时, 是 MA(q) 模型:

$$x_t = \mu t + \varepsilon_t + \beta_1 \varepsilon_{t-1} + \cdots + \beta_q \varepsilon_{t-q} \tag{3}$$

本文中 p, q 的取值原则是采用最小信息准则法. 通过该方法来识别平稳和可逆的 ARMA 过程[8]. 求解 p, q 在 m 以内的 BIC 矩阵, 寻找矩阵中 BIC 信息量最小的位置, 从而确定合适的 p, q 的值, 得到 ARIMA (p, n, q).

1.2 XGBoost 模型

XGBoost 是一种在梯度提升决策树算法的基础上进行改进而来的集成学习算法[9–11]. 其预测原理如下:

预测值为各样本与其权值乘积的累加和, 即:

$$\hat{y}_i = \sum\nolimits_j w_j x_{ij} \tag{4}$$

其中, 为 j 样本数, w_j为权值, x_{ij}为样本数据. XGBoost 在做回归时, 每棵树是依次加入模型中, 进而提升模型的效果, 这个集成可以表示为:

$$\begin{cases} \hat{y}_i^{(0)} = 0 \\ \hat{y}_i^{(t)} = \sum\limits_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \end{cases} \tag{5}$$

其中, $\hat{y}_i^{(t)}$为第 t 轮的模型预测, $\hat{y}_i^{(t-1)}$为保留的前 t 轮的模型预测, $f_t(x_i)$为新加入的函数, 加入节点过多会导致过拟合, 因此在目标函数中加入惩罚项来降低过拟合的风险, 惩罚项如下:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{6}$$

其中, γT为惩罚力度, $\frac{1}{2}\lambda \sum\limits_{j=1}^{T} w_j^2$为惩罚项. γ为叶子节点数 T 的系数. 目标函数由自身的损失函数和正则化惩罚项构成, 定义如下:

$$obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)} + f_t(x_i)\right) + \Omega(f_t) + constant \tag{7}$$

其中, obj为结构分数, 表示当选定一个树的结构后, 目标减少量的最大值.

1.3 LSTM 模型

LSTM Networks 是递归神经网络 (RNNs) 的一种. 后经过不断改进, 在处理和预测时间序列相关的数据时会比一般的 RNNs 表现的更好[12].

对于原始的$m\times n$维数据, 记作:

$$M_j=(T_{1j},T_{2j},\cdots,T_{nj}),j=(1,2\cdots,m) \tag{8}$$

对于多变量的数据需要转化为监督学习的序列, 即新的数据表示为如下:

$$\begin{cases}M'_j=(M_{ij-k},\cdots,M_{ij-1},M_{ij})\\ i=1,2\cdots,n,\ \ j=1,2,\cdots,n\end{cases} \tag{9}$$

LSTM 的模型中采用门的结构来解决了长期依赖问题[13], 其具体的神经网络的细胞结构如图 1 所示.

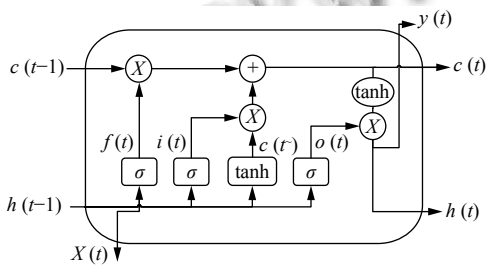

图 1 LSTM 记忆细胞结构

图 1 中每个 Sigmoid 层产生的数字在 0 和 1 的范围内. 每个 LSTM 通过 3 种类型的门[14]来控制每个单元的状态: 遗忘门决定了上一时刻的单元状态c_{t-1}有多少保留到当前时刻c_t; 输入门决定了当前时刻网络的输入x_t有多少保存到单元状态c_t. 输出门控制单元状态c_t有多少输出到 LSTM 的当前输出值h_t. 每一步的状态更新满足以下的步骤:

$$i_t=\sigma(W_{xi}x_t+W_{hi}h_{t-1}+b_i) \tag{10}$$

$$f_t=\sigma(W_{xf}x_t+W_{hf}h_{t-1}+b_f) \tag{11}$$

$$o_t=\sigma(W_{xo}x_t+W_{hf}h_{t-1}+b_o) \tag{12}$$

$$\tilde{c}_t=\tanh(W_{xc}x_t+W_{hc}h_{t-1}+b_c) \tag{13}$$

$$c_t=f_t\circ c_{t-1}+i_t\circ\tilde{c}_t \tag{14}$$

$$h_t=\tanh(o_t\circ c_t) \tag{15}$$

这里σ是逻辑 Sigmoid 函数, $\circ$表示按元素相乘; W_i,W_f,W_o,W_c是权重矩阵; b_i,b_f,b_o,b_c是偏移量.

2 基于 XGBoost 和 LSTM 的加权组合预测模型

对于实际销售数据中, 数据存在周期性, 季节性的变化, 通常为非平稳的时间序列. 通过差分平稳化后, 虽然在 ARIMA 模型中表现的较好, 但却丢失了周期性和季节性特征, 并且平稳的数据无法表现出现实销售增量变化, ARIMA 模型只依靠内生变量, 模型过于简单, 无法捕捉序列中的非线性因素. 神经网络算法在处理非线性问题具有独特的优势, LSTM 模型的加入可以解决以下几个问题, 首先是销售数据的连续性, 通过特殊的数据输入结构使得模型在预测时结合了历史的状态, 其次对比与传统的 RNNs 解决了输入变长的问题, 再者实际销量的影响因素较多, 而节假日, 策略量的变化会带来销售量的异常变化, 通过多变量的模型可以提升拟合的精度. 此时特征较多, 需要经过处理再放入神经网络的模型中, 本文中使用 XGBoost 算法进行特征的抓取, 充分利用多维变量中的潜在的特征. 在 ARIMA, XGBoost, LST 模型 3 种模型预测的传递过程会出现误差累计的问题, 对此本文对 3 种模型的预测测结果进行加权处理来减小误差累计对预测结果精度的影响. 具体实验过程如图 2 所示.

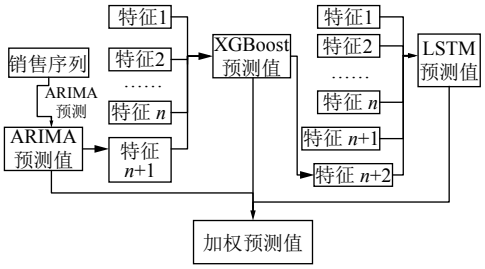

图 2 XGBoost 和 LSTM 加权组合模型预测结构

对于收集的原始销售序列$A=\{x_1,x_2,\cdots,x_n\}$进行可视化并观察数据的基本趋势, 然后采用增广迪基-福勒检验 (Augmented Dickey-Fuller test, ADF), 以及 JUNG-BOX 白噪声检验来检验时间序列的平稳性, 对于线性趋势的非平稳时间序列, 通过差分变换将其变为平稳序列. 收集影响销售的特征值, 特征值 1, 特征 2…特征值 n, 分别记为$T_1,T_2,\cdots,T_n$, 假设 A 经过 ARIMA 模型后所得到的预测序列T_{n+1}, 则有:

$$\begin{cases}T_{n+1}=T_{n+1}(m)+T_{n+1}(k)=\Phi(A(m))+\Phi(A(k))\\ m+k=n\end{cases} \tag{16}$$

其中, Φ表示 ARIAM 模型, $A(m)$为前 m 个销售数据,

$A(k)$为后 d 个数据.

将由 ARIAM 模型所得的预测序列T_{n+1}加入到特征序列中构建多维数组$M=\{A, T_1, \cdots, T_n, T_{n+1}\}$, 经过处理及标准化后放入到 XGBoost 模型中, 得到销量的预测序列T_{n+2}, 则有:

$$\begin{cases} T_{n+2}=T_{n+2}(m)+T_{n+2}(k)=\Theta(T_{n+1}(m))+\Theta(T_{n+1}(k)), \\ m+k=n \end{cases} \tag{17}$$

其中, Θ表示 XGBoost 模型, $T_i(m)$表示第 i 个特征序列的前 m 行, $T_i(k)$表示第 i 个特征序列的后 k 行.

此时再将新的销售预测序列T_{n+2}加入到多维数组 M 中得到新的数组$M'=\{A,T_1,\cdots,T_{n+1},T_{n+2}\}$, 然后加入到 LSTM 模型中得到销量预测序列$T_{n+3}$, 则有:

$$T_{n+3}=\psi(M') \tag{18}$$

其中, Ψ表示 LSTM 模型.

最后结合 3 个模型的商品销量预测量$T_{n+1}(k)$, $T_{n+2}(k), T_{n+3}(k)$, 给予相对应的权值, 通过多次实验得出最佳的组合权值, 进而得到最终的预测值 T, 记为:

$$\begin{cases} T=d_1T_{n+1}(k)+d_2T_{n+2}(k)+d_3T_{n+3}(k) \\ d_1+d_2+d_3=1 \end{cases} \tag{19}$$

其中, d_i为各个模型的的预测值的权值. 实验步骤如下:

步骤 1. ARIAM 预测: 单变量预测实验, 取出原数据中的实际销售数据列$X=\{X_1,X_2,\cdots,X_n\}$, 将$\{X_n\}$放入 ARIMA 的模型中, 对数据列进行 ADF 平稳性检验及 JUNG-BOX 白噪声值检测, 通过 ADF 检验可以得到单位根检验统计量对应的 p 值, 此值显著小于 0.05, 则该序列平稳, 通过不断实验后发现, p, q 的取值通常在 8 以内, 因此通过循环求解 8 以内的 BIC 矩阵, 找出矩阵中的最小信息量所对应的 p, q 的值为 1, 0, 得出预测模型 ARIMA (1, 0, 0), 为了使结果更加贴近现实情况, 采滚动预测, 每预测一周的销售数据后, 加入该周的实际的销售数据来预测下一周的销量, 最后整理预测值序列, 得到销售值序列的样本的预测序列T_8.

步骤 2. XGBoost 预测. 将T_8序列合并到特征序列中, 分成训练集和测试集两部分, 然后放入到 XGBoost 模型中作预测, 将预测值记作T_9.

步骤 3. 多维时间序列预处理. 将T_9合并到 M 中组成新的数组M_1, 接着对所有的特征归一化处理, 然后利用 *series_to_supervised* 函数对数据进行处理, 该函数将单变量或多变量时间序列转换为监督学习数据集, 使用 Pandas 的 Shift 函数, 将原始列向后移动 k 位后添加成新的列, 同时将当前时刻的除去销售值的特征移除.

步骤 4. LSTM 神经网搭建. LSTM 模型中, 搭建 2 个隐藏层, 第一隐藏层有 128 个神经元, 第二隐藏层有 256 个神经元, 输出层为 1 维的列向量, 即销售预测值, 输入变量是一个时间步 (t–1) 的特征, 损失函数采用 Mean Absolute Error (MAE), 优化算法采用 Adam, 激活函数采用 Sigmoid, 模型采用 500 个 *epochs* 并且每个 *batch* 的大小为 15.

步骤 5. 数据预测. 经过多次的实验, 发现当d_1 = 0.2, d_2 = 0.4, d_3 = 0.4时, 即

$$T=0.2\times T_{n+1}+0.4\times T_{n+2}+0.4\times T_{n+3} \tag{20}$$

其中, T 为最终的销量预测值, T_{n+1}为 ARIMA 模型的预测值, T_{n+2}为 XGBoost 模型的预测值, T_{n+3}为 LSTM 模型的预测值, 最后将最终的加权预测值作为后 13 周的销售预测值.

3 实验分析

3.1 实验环境

实验所使用计算机的配置如下: 处理器为英特尔酷睿 Duo CPU i5-6500, CPU 频率为 2.20 GHz; 内存为 8 GB; 操作系统为 Windows 10 (64 位); 基于 Python 3.6 编程; 集成开发环境为 PyCharm Community Edition 2016. LSTM 的实验使用的是 keras 深度学习框架.

3.2 实验数据

以某地的从 2018 年 4 月 2 号到 11 月 4 号的某商品销量数据为研究对象, 数据集中主要包含 7 个特征, 分别是节假日因素, 气温, 主要策略规格的策略量以及实订量等, 如表 1 所示.

表 1 数据特征

特征表示	特征含义	取值
vacation	假日	0-非节假日, 1-节假日
mainstrategy	主策略量的实订量	销售数据
XGBoost	XGBoost 的预测值	模型预测值
ARIMA	ARIMA 的预测值	模型预测值
(year, month, day)	(年, 月, 日)	日期数据
(amax, amin)	(气温最高值, 气温最低值)	取周气温的平均值

对于节假日因素, 本文采用虚拟变量来进行量化处理, 记作T_1, 取每周最高气温的平均值及最低气温的平均值, 分别记作T_2, T_3, 主要策略量的实订量记作

T_4, 年月日分别记作T_5, T_6, T_7, 然后构建多维数组 $M = (X, T_1, T_2, T_3, T_4, T_5, T_6, T_7)$, 划分数据. 将数据的前 0.67 作为训练集, 后 0.33 作为测试集, 对未来 13 周的销售量进行预测.

3.3 参数选择

在 LSTM 模型中, *n_1* 为 series_to_supervised 函数中设置的滞后观察数为 1, 即使用上一时刻的销售量来预测当前时刻的销售值, 设置两个隐藏层, 第一层的神经元个数设为 128, 第二层的神经元个数设置为 256, *dropout* 随机删除一些隐层神经元, 通过不断的调整这两个参数来解决预测过程中的过拟合的问题. 具体参数设置见表 2.

表 2 神经网络参数

参数	值
学习率	0.03
Dropout rate	0.2
n_in	1
epoch	500
Batch_size	15

在 XGBoost 模型中, subsimple 为训练的实例样本占整体实例样本的比例. max-depth 为每棵树的最大深度. *booster* 为设置需要使用的上升模型. *objective* 定义学习任务及相应的学习目标, 本文选的目标函数为 "*reg: linear*"-线性回归. 具体参数设置见表 3.

表 3 XGBoost 参数

参数	值
booster	*gbtree*
subsimple	0.67
maxdepth	6

3.4 实验结果及分析

为了更好的体现模型的优越性和实用性, 本文主要采用两个指标来进行模型的评估, 第一个是均方根误差 (Root Mean Square Error, *RMSE*), 以及一个平均准确率 (Mean Accuracy, *MA*), 定义如下:

$$RMSE = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{21}$$

$$MA = \frac{1}{N}\left(1 - \frac{|\hat{y}_i - y_i|}{\hat{y}_i}\right) \tag{22}$$

使用四种模型分别预测出后 13 周的销售值, 导出组合模型的预测值, 图 3 为组合模型预测值和实际销售值的对比, 销量变化的趋势得到较好的拟合, 预测平均准确率为 0.968, 预测误差很好的控制在了 0.05 以内, 预测相对准确.

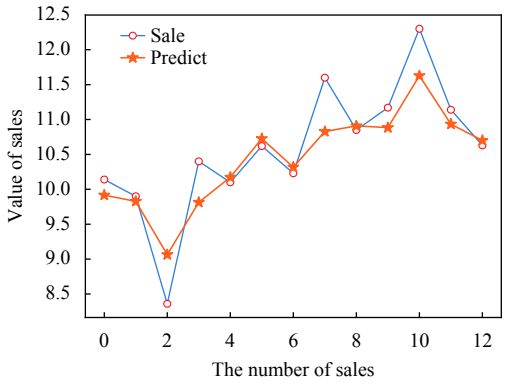

图 3 预测值与实际值对比

图 4 反映的是单个模型及组合在每周的预测的精度对比, 从图中可以看出, 不同模型预测时的表现的具有一定的差异性, 也存在个别异常时间点在单个模型预测中效果不佳. 首先对比 3 个模型的预测情况, ARIMA 模型预测效果是较弱的, 对于波动的灵敏度较低, XGBoost 和 LSTM 预测效果相对接近, 因此本文主要采用简单平均法来分配权重. 对比每周的预测结果, LSTM 比 XGBoost 模型在预测精度表现较好, 但稳定性相对较差, 因此在分配权重时既要保留 XGBoost 模型预测稳定的特点, 同时也能提高模型的精度, 因此对于这两种方法赋予相同的权值. 对于数据内生变量的影响通过分配 ARIAM 模型相对较小的比重进行模型的预测值的调整. 基于该想法进行预测结果的加权分析, 通过计算不同比重下预测结果的平均精度和 RMSE 的值, 发现在 XGBoost 和 LSTM 所占都为 0.4 时, ARIMA 占 0.2 时, 组合模型整体预测效果表现最佳, 对于非线性部分拟合的效果明显的体现除了加权组合的优势, 对比单个模型出现准确率在 0.8 附近的异常点, 组合模型的预测准确率都在 0.9 以上, 对比单独模型预测良好的部分, 组合模型保持了良好的预测精度, 预测结果的精度对比单一模型有很大的提高, 对于其他特征 (策略量, 温度, 节假日, 日期) 实行了一次复制, 即这些变量会在每个模型行的训练中出现两次, 而结果证明训练集误差几乎一致, 但是验证集误差更小, 表明通过特征的重复训练, 能够减小组合模型的过拟合程度, 模型的泛化性得以提升, 预测效果更佳.

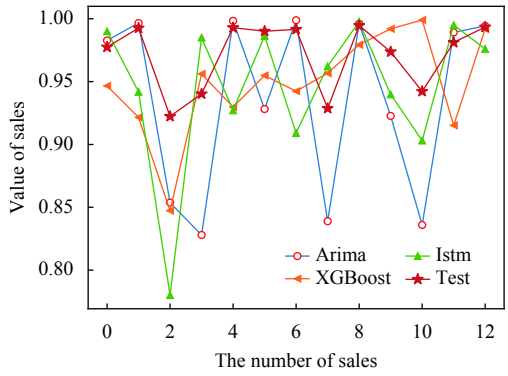

图 4 不同模型预测精度对比

观察不同模型的指标数值, 由导出的预测值和实际值求解出各个预测模型的 *RMSE* 指标值和 *MA* 指标值. 通过对比不模型的 *RMSE* 和 *MA* 的值, 来评估各个模型的整体的预测效果. 通过表 4 不同模型参数的对比, 组合模型的平均预测值达到了 96.84% 的水平, 相比于 ARIAM 预测提高了 3.28%, 相比于 XGBoost 的预测提高了 1.99%, 相比于 LSTM 模型提高了 2.28%, 综上所述, 组合模型在商品销售的预测中展现了优于其他单一模型的预测效果, 通过加权组合提高了模型的泛化性和有效性.

表 4 不同模型预测结果对比

预测模型	评价指标	
	RMSE	*MA*
ARIMA	0.927	0.935 642 95
XGBoost	0.572	0.948 649 69
LSTM	0.854	0.945 642 96
ARIMA-XGBoost-LSTM	0.428	0.968 398 46

研究主要策略量对于模型的影响, 在本文的实验中, 结合了主要规格的策略实订量这一特征值, 而在实验前通过观察不难发现, 这一主要策略量跟销售序列有着较为紧密的联系, 首先策略量序列总是小于对应的销售值, 其次, 通过数据可视化可以发现, 当销售增大时, 策略量也增大, 销售量减少时, 策略量也明显的减少, 在此基础上检验策略量的实订量的有无对预测的影响, 通过设计对比实验来验证. 实验结果如表 5 所示, 在加入了策略量后, 对比于未加入策略量实订量的实验结果相比, 预测的准确率提高了 4.05%, RMSE 降低了 0.64, 说明在其他条件不变的情况下, 策略量对于销售预测的影响较为显著, 可以通过加入该特征来辅助预测.

表 5 策略量对预测结果的影响对比

策略量对比	评价指标	
	RMSE	*MA*
未加策略量模型预测	0.492	0.927 888 05
加入策略量模型预测	0.428	0.968 398 46

4 结论与展望

本文提出的基于 XGBoost 和 LSTM 加权多变量的加权组合预测模型, 用来解决商品销售的预测, 通过加入天气, 主策略量实订量, 节假日, 日期 (年月日) 诸多因素来辅助预测, 提高预测的准确率, 提高模型的有效性, ARIMA 在处理线性模型效果较好, LSTM 神经网络可以通过学习来拟合非线性问题, XGBoost 模型则可以挖掘多维变量中不同维度的属性, 通过不同层面的模型的组合, 大大的提高了模型的泛化能力, 不仅针对商品销售的预测, 还可以应用于其他相关的多变量时间序列的预测领域, 提高预测的精度, 解决不同的实际问题. 但在对于该模型中的神经网络来说, 过多的属性存在过拟合的风险, 因此模型还可以在该方面优化.

参考文献

1 吴鹏. 神经网络在卷烟销售量预测的应用研究. 计算机仿真, 2012, 29(3): 227–230. [doi: 10.3969/j.issn.1006-9348.2012.03.056]

2 Zhang GP. Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing, 2003, 50: 159–175. [doi: 10.1016/S0925-2312(01)00702-0]

3 姚小强, 侯志森. 基于树结构长短期记忆神经网络的金融时间序列预测. 计算机应用, 2018, 38(11): 3336–3341. [doi: 10.11772/j.issn.1001-9081.2018040742]

4 Fischer T, Krauss C. Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 2017, 270(2): 654–669.

5 赵一安. 基于机器学习 Xgboost 模型解决商店商品销量预测的问题. 通讯世界, 2018, (11): 250–252. [doi: 10.3969/j.issn.1006-4222.2018.11.163]

6 Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735–1780. [doi: 10.1162/neco.1997.9.8.1735]

7 李俊, 何刚. 基于组合预测的商品销售量预测方法. 统计与决策, 2012, (8): 28–31.

8 吕忠伟. 单变量时间序列模型识别方法的实证研究. 统计与信息论坛, 2006, 21(3): 27–30. [doi: 10.3969/j.issn.1007-3116.2006.03.006]

9 Chen TQ, Guestrin C. XGBoost: A scalable tree boosting

system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, CA, USA. 2016. 785–794.

10 周鹏. 基于 CNN-XGBoost 的 PTA 平均粒径动态软测量模型. 信息技术与网络安全, 2018, 37(9): 61–64.

11 贾锐军, 冉祥来, 吴俊霖, 等. 基于 XGBoost 算法的机场旅客流量预测. 民航学报, 2018, 2(6): 34–37, 33. [doi: 10.3969/j.issn.2096-4994.2018.06.009]

12 于家斌, 尚方方, 王小艺, 等. 基于遗传算法改进的一阶滞后滤波和长短期记忆网络的蓝藻水华预测方法. 计算机应用, 2018, 38(7): 2119–2123, 2135. [doi: 10.11772/j.issn.1001-9081.2017122959]

13 Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks. eprint arXiv: 1211.5063, 2012.

14 Siama-Namini S, Namin AS. Forecasting economics and financial time series: ARIMA vs. eprint arXiv: 1803.06386, 2018.