



中央财经大学

Central University of Finance and Economics

# 本科生毕业论文（设计）

分层时间序列方法在商品销量预测中的应用

学生姓名: 杨谨行

学 号: 2017312292

学 院: 统计与数学学院

专 业: 应用统计学

指导教师: 许欣怡

日 期: 2021 年 5 月 16 日

## 内容摘要

预测商品的销量可以帮助零售企业更好地管理库存和筹划营销。在每件商品的时间序列数据通过销售地区、商店、产品类别等维度汇总后，一家企业的商品销量数据常常以大规模、多层次的分层时间序列形式呈现。

针对销售型的分层时间序列（HTS）数据的特点，本文创新性地将新兴的时间序列预测方法与 HTS 模型校准技术相结合，提出两个代表性方法。第一种方法从时间序列方法出发，用大规模的 Prophet 模型结合 HTS 的自中向外法和最优结合法进行模型校准。第二种方法则着重处理用于分层的变量。使用滑窗、滞后阶等特征工程方法，提取各层的时间序列特征将序列转化为有监督的机器学习问题，并训练 LightGBM 模型使用自中向外法对模型校准。

在对这两种方法的实证中，本文使用了从美国加利福尼亚州、得克萨斯州和威斯康星州 10 家沃尔玛商店于 2012-2016 产生的 30970 条真实商品销量时间序列，以及基于底层序列加工的 11830 条按 11 个不同的层级组合加总所得的上层序列，数据量达到七千万行（约 4.7GB）。本文分析了多个这些序列受外生变量的影响下的统计规律，并说明了其有显著代表性的 HTS 特点。接着，本文讨论了加权均方根标度误差（WRMSSE）指标在评价 HTS 模型上的优越性，并比较了两种方案相较 ARIMA、LightGBM 单模型等基准方案的表现提升程度。实证发现，Prophet 方法的准确性和稳定性在验证集和测试集中表现更好，但 Prophet 和 LightGBM 方案对沃尔玛的销售额预测都基本符合趋势，也有很强的工业界落地的商业价值。

**关键词：**分层时间序列 prophet 模型 lightgbm 模型 商品销量预测

## ABSTRACT

Sales forecasting can empower retailers to efficiently manage inventory and plan marketing. Usually, all the time series data of each product can be summarized through dimensions such as sales area, store, and product category. In this case, a company's product sales data are often presented in the form of large-scale and multi-level hierarchical time series (HTS).

Aiming at the characteristics of the sales hierarchical time series data, this paper innovatively combines the emerging time series forecasting method with the HTS model

calibration technology and proposes two representative methods. The first method starts from the time series method, combining a large-scale Prophet model with HTS's Middle-out method and using the optimal combination method for model calibration. The second method concentrates on the stratification variables. Using feature engineering methods such as rolling window and lagging, this paper extracts the time series features of each layer to transform the time series forecasting task into a supervised machine learning problem and train a LightGBM model to calibrate it with Middle-out way.

In the empirical analysis part, this paper uses 30,970 real product sales time series published by 10 Walmart stores in California, Texas, and Wisconsin of the United States in 2012-2016, and 11,830 upper series obtained by summing up series in 11 different layers. After discussing the advantages of the Weighted Root Mean Square Standardized Effect (WRMSSE) metric in evaluating the HTS model, this paper compares the performance of two schemes with the benchmark models such as ARIMA and LightGBM single model in the context of WRMSSE. The empirical findings indicate that the Prophet outputs more accurate and stable predictions in the validation set and test-set, but the Prophet and LightGBM schemes both essentially model the trend for Walmart sales data, and also show promising commercial value in the industry.

**KEY WORDS:** hierarchical time series prophet lightgbm sales forecast

# 目录

<b>一、 绪论</b> .....	<b>1</b>
(一) 问题背景.....	1
(二) 国内外研究现状.....	3
(三) 研究内容与本文结构.....	6
<b>二、 典型商品销量分层时间序列数据的分析</b> .....	<b>6</b>
(一) 数据基本情况.....	6
(二) 数据描述性统计.....	9
<b>三、 针对分层时间序列预测问题的模型方案</b> .....	<b>16</b>
(一) 方法一：基于 HTS 调优的 Prophet 模型.....	16
(二) 方法二：利用特征工程转换数据的 LightGBM 模型.....	18
<b>四、 实证结果与分析</b> .....	<b>23</b>
(一) 模型评价指标.....	23
(二) 模型实证结果.....	24
<b>五、 结论与展望</b> .....	<b>28</b>
(一) 本文结论.....	28
(二) 本文创新性.....	28
(三) 不足之处与展望.....	28
<b>参考文献</b> .....	<b>29</b>

# 分层时间序列方法在商品销量预测中的应用

## 一、绪论

### (一) 问题背景

随大数据技术不断发展，大量以层次结构组织的时间序列数据（Hierarchical Time Series, HTS）开始出现在互联网、零售、生产制造等商业场景中。比如，一个手机软件每个用户日浏览时间的时间序列数据可根据性别、地理位置或产品功能等维度被分组（grouped）与加总（aggregated）从而组织成多层级的数据；一个厂家的整体销量，也可根据销售国家、商品种类、零售商等类别拆分（disaggregated），从而建立起对各个单一商品时间序列基节点几层的结构。一个可以被三次分为 A、B、C 三组的三级分层时间序列如图 1 所示：

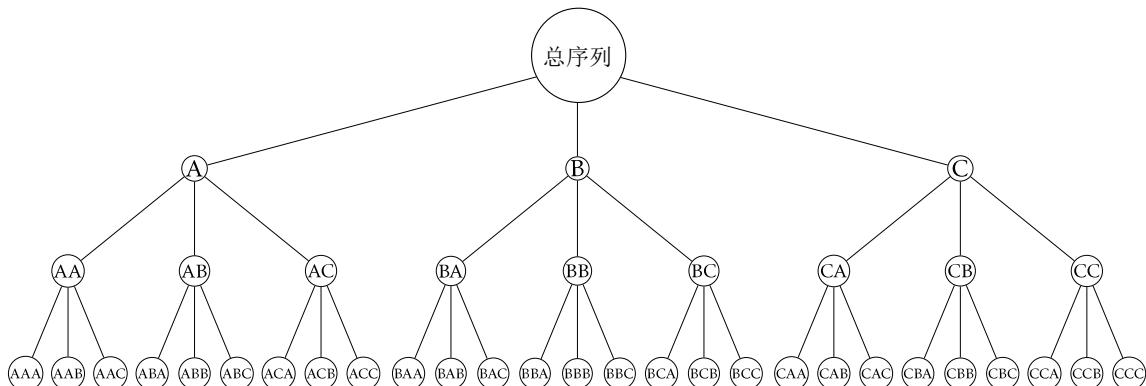


图 1 三级分层时间序列示意图

HTS 数据有其独特特点。第一，这些时间序列通过自然属性分组聚类后，其内部很可能是相互关联的，如一些地区的销量整体性地高于另一区域，即同层时间序列类中的各序列间不独立同分布[1]。第二，对 HTS 的预测通常要求通过对每个级别的预测相加而给出上一层级别的预测，即要求每组的预测值等于组成该组的各个系列的预测值。比如，一个公司通常会要求预测总销售额、全国销售额、地区销售额，直至单个门店的销售额在整个层级结构中适当相加可得到上层销量[2]。然而，也有一些 HTS 不必满足这种不同层次的等式约束。例如一个在点击多次后进入的浏览量可以不等于父级别上的网页浏览量。因为互联网用户可能直接访问子网页而不访问父网页，这会打破之前的假设(父级的观察值严格等于子级的观察值之和)，导致了层次结构中的不平等约束[3]。

此类序列不纳入本文的研究范围。第三，有时 HTS 数据对时间序列的分解方式是不唯一的。有些情况下每层都在较高层的节点内不能调换，例如，销售的时间序列按国家、州和地区进行地理上的分类。但不少情况中分解形成的层次结构不唯一。例如，销量时间序列可能涉及地理分组（按国家）和产品分组（按销售的产品类型）。虽然这些分组变量中的每一个都能导出唯一的层次结构，但如若同时使用两者，则由于分层的顺序不唯一（如可以先按地理分再按产品分，也可以先按产品分再按地理分）且这两种分层方式都有实际含义，因此仅选择其中一种层次结构是不合适的。HTS 领域多篇核心论文的作者（Rob J. Hyndman, 2015）将不可调换顺序的 HTS 区分为分组时间序列（Grouped Time Series, GTS）[2]，本文不讨论两者差异因此使用 HTS 代称两者不作区分。

作为较为新兴的学术研究话题，对 HTS 的建模应用目前研究主要仅落地在能源领域。电力需求数据通常可以用层次结构表示，例如，整个国家的用电量可以按州、城市和家庭进行分类。分层时间序列预测各层预测精度的保证及一致性可以帮助国际在发电和配电规划中进行更好的决策，包括电网[5]、太阳能发电[6]、能源运输[7]、和空气污染[8]等。然而，HTS 应该还有更广泛的应用前景。比如宏观经济预测中，国民经济账户分为生产、收入和支出以及资本交易。生产可以进一步分为农林牧渔等；收入和支出以及资本交易进一步分为个人、公司、公共公司等。在国民卫生预测中，婴儿死亡率可以按性别分列；在每个性别中，可以进一步按照不同的省区分，这些都可以将 HTS 校准方法嵌入已有的模型解决方案中。

作为一个典型的应用落地场景，本文将聚焦讨论 HTS 数据的预测方法对零售业领域的赋能可能性。当零售商不能准确测量市场需求，供应链效率就无法保证[9]。许多商品由零售店网络中散布各地的子公司销售，销量也可以分地区、分种类、分产品，非常符合 HTS 的特点。尽管数据挖掘技术已经在商业应用中迭代多轮，但 HTS 的引入目前只有一些探索性讨论。通过 HTS 对销售额预测，大型零售公司可以在保持库存和客户需求之间做出更好的数据驱动决策，从而最大化商店优化利润，而不是传统地仅仅依赖于历史销售记录统计趋势的外推。

然而，对 HTS 预测常常不得不在信息损失和丧失加总约束上做权衡。通常，最底层的时间序列会有较高的波动从而很难准确预测，而最顶层的时间序列虽然较为平滑但损失了底层的更多的信息[10]。不论是预测底层的序列并向上加总（Bottom-Up）还是只预测顶层往下拆解（Top- Down）都无法囊括所有信息。但如果对每一层都独立地建模预测看似合理，但在预测时会打破 HTS 的“加总约束条件”产生加总谬误。为了更好地权衡，前人研究了许多 HTS 的方法将在下部分详细介绍。

## (二) 国内外研究现状

截止到本文截稿，对分层时间序列的研究仅有一篇中文文献，且侧重于时间序列聚类角度[11]。近年国外对 HTS 的论文却是层出不穷，最早可以追溯到 Dalrymple 在 1987 年提出的三种管理者预测销量的方法，分别为预测总销量后用固定的权重分到每个商品、预测每个商品然后加总为总销量、直接预测每层的销量而无视加总一致性的约束条件[12]。随后 1990 年，第一种方法被 Charles 等人命名为分解方法（Disaggregation methods）[13]，开启了后人对针对 HTS 特点而建模的研究。

使用分层时间序列技术对已有模型校准的动机主要来源于两个。第一是不同层次时间序列有潜在的加总关系，如果对不同层级的序列分别建模会损失信息且产生“加总谬误”。第二是最底层的产品销量序列通常波动性较大，容易对上层预测产生噪音。总体而言，预测 HTS 的方法可以分为自底向上（Bottom-Up, BU）、自顶向下（Top-Down, TD）以及基于这两种方法的组合出的多种变形和综合应用，衍生出自中向外（Middle-Out）、最优结合（Optimal Combination）、最短路径（Trace Minimization, MinT）等方法，但国外学者对基础模型的选取和与实际问题的结合的重视程度并不高。

### 1. 自底向上方法

预测分组时间序列的常用方法之一是自底向上方法，Dangerfield 等人及 Zellner 等人分别在 1992 和 2000 等人对方法进行明确定义[14][15]。这种方法首先要为基层序列生成基本预测（base prediction），然后对上层序列依次聚合来实现上层的各种预测需要。

对图 1 的三层时序结构记总层级  $K = 3$ ，对任意层序列  $X$  可以用序列被分层次数的字母个数来进行标记，比如用 A 表示第一层的 A 系列、用 AC 表示第一层 A 系列的第二层 F 系列，以此类推。记在时间  $t = 1, 2, \dots, n$  下对序列 X 的观测为  $Y_{X,t}$ ，如  $Y_{ABC,t}$  表示第一层为 A、第二层为 B 的序列分解后的 C 序列在时间 t 下的观测，根据上一部分讨论的 HTS 加总性约束条件，可以用符号表达为：

$$Y_t = \sum_i Y_{i,t}, \quad Y_{i,t} = \sum_j Y_{ij,t}, \quad Y_{ij,t} = \sum_k Y_{ijk,t}, \quad Y_{ijk,t} = \sum_\ell Y_{ijkl,t}$$

进一步，记  $m_i$  为第  $i$  次分解上层序列后得到的第  $i$  层子序列个数，并记  $m = m_0 + m_1 + m_2 + \dots + m_K$ ，在图 1 序列中， $m_i = 3^i$ ， $m = 40$ 。

记第  $i$  层的  $m_i$  个观测为  $m_i$  维列向量  $\mathbf{Y}_{i,t}$ ，则所有观测可记为  $m$  维列向量  $\mathbf{Y}_t = [Y_t, \mathbf{Y}_{1,t}, \dots, \mathbf{Y}_{K,t}]^T$ ，本例中进一步展开后有

$$\mathbf{Y}_t = [Y_t, Y_{A,t}, Y_{B,t}, Y_{C,t}, Y_{AA,t}, Y_{AB,t}, \dots, Y_{CC,t}, Y_{AAA,t}, Y_{AAB,t}, \dots, Y_{CCC,t}]$$

则将加总性约束条件分解到最下层序列之后可以用矩阵表达为

$$\mathbf{Y}_t = \mathbf{S} \mathbf{Y}_{K,t}$$

其中，

$$\mathbf{S} = \begin{bmatrix} 11111111111111111111111111 \\ 11111111100000000000000000 \\ 000000001111111100000000 \\ 000000000000000001111111 \\ 111000000000000000000000 \\ 000011100000000000000000 \\ \vdots \\ 000000000000000000000000 \\ 100000000000000000000000 \\ 010000000000000000000000 \\ \vdots \\ 000000000000000000000001 \end{bmatrix}$$

其中是一个  $m \times m_K$ 、秩为  $m_K$  的矩阵，被称为加总矩阵（summing matrix），矩阵的第一行对应的等式为把所有的基层序列加在一起为总序列，进一步地，对未来  $h$  个时间点  $t = n + 1, n + 2, \dots, n + h$  的预测也就可以记为

$$\hat{\mathbf{Y}}_t = \mathbf{S} \hat{\mathbf{Y}}_{K,t}$$

许多实证研究证实了自底向上预测的有效性，如 Kinney (1971) 发现，按市场细分的分类收益数据自底向上比使用公司层面数据直接预测更准确[16]。Dunn 等人 (1976) 对电话需求的研究表明，从较低层次的建模中聚合的预测比自顶向下的方法更准确[17]。Zellner 和 Tobias (2000) 使用了来自 18 个国家的年度 GDP 增长率，并发现分类自底向上预测结果更好[15]。

但是，Bottom-Up 方法虽使预测结果满足加总约束，但由其对序列间关系的忽略和高噪音底层序列的存在，这种方法往往在多层分类数据上表现不佳。

## 2. 自顶向下方法

自顶向下的方法首先需在层次结构的顶部生成“总体”系列的基本预测，然后根据适当的比例分解预测。在这种情况下，预测也就满足了加总的约束。以图 1 序列为例，对各层的权重  $p_A, p_B, p_C, p_{AA}, p_{AB}, \dots, p_{CC}, p_{AAA}, p_{AAB}, p_{CCC}$  确定后，只需根据

$$Y_{i,t} = p_i \cdot Y_t, \quad Y_{ij,t} = p_{ij} \cdot Y_{i,t}, \quad Y_{ijk,t} = p_{ijk} \cdot Y_{ij,t}, \quad Y_{ijkl,t} = p_{ijkl} \cdot Y_{ijk,t}$$

对预测进一步分解即可，从而得到

$$\tilde{Y}_t = S p_j \hat{Y}_t$$

关于比例的计算方法，Gross 和 Sohl (1990) 讨论过几种可能的选择比例的方式[13] 如表 1 所示。

这种方法的好处是在顶层进行可靠的预测，但它可能会受到与单个序列动态相关的信息丢失的影响[5]。从而在层次结构的较低层次产生较不准确的预测，特别是对于具有

复杂分布的单个序列。事实上，Hyndman 等人（2011）发现在任何自顶向下的方法都会在预测中引入偏差，即使基础预测本身是无偏的[19]。

表 1 Gross-Sohl 自顶向下方法中的比例选择方法

对过去比例取平均 Average Historical Proportions	对过去的加权平均求比例 Proportions of the historical averages	预测权重法 Forecast Proportions
$p_j = \frac{1}{T} \sum_{t=1}^T \frac{Y_{j,t}}{Y_t}, j = 1, \dots, m_K$	$p_j = \frac{\sum_{t=1}^T \frac{Y_{j,t}}{T}}{\sum_{t=1}^T \frac{Y_t}{T}}, j = 1, \dots, m_K$	$p_j = \prod_{l=0}^{K-1} \frac{\hat{Y}_{j,t}^{(l)}}{\hat{S}_{j,t}^{(l+1)}}, j = 1, \dots, m_K$
$p_j$ 反映了历史的所有时间点 $t = 1, \dots, T$ 上最顶层序列 $Y_t$ 中 最底层序列的占比的平均值	$p_j$ 用最顶层序列 $Y_t$ 根据时间倒数加 权平均与最底层序列的根据时间倒 数加权平均的比值	使用 $h$ 天后的预测 $\hat{Y}_{j,h}^{(l)}$ 与在 $j$ 下面 1 层的预测加总 $\hat{S}_{j,t}^{(l)}$ 的比 重累乘得到比例

### 3. 对自顶向下和自底向上基本算法的优化方法

#### (1) 自中向外方法 (Middle-Out)

Middle-Out 方法是对自底向上和自顶向下方法最简单的结合，其首先选择某个“中间水平”后对该水平的所有序列生成预测，接着对该层以上的序列自底向上聚合形成预测，最后再对下级的序列自顶向下分解形成预测，从而保证了各层预测的连贯性。

#### (2) 最优结合法 (Optimal Combination method)

Hyndman 等人（2011）提出最优结合法使用了最小二乘的思路。首先，先为层次结构中的每个序列生成独立的基本预测[19]。第二步，将从 $t$ 时刻向 $t + h$ 时刻的预测转换为一个回归问题

$$\hat{Y}_{t+h|t} = \mathbf{S}\boldsymbol{\beta}_{t+h|t} + \boldsymbol{\varepsilon}_h$$

其中， $\hat{Y}_{t+h|t}$ 表示对所有层全部序列  $h$  个时间点外的预测，而  $\boldsymbol{\beta}_{t+h|t}$  表示第  $K$  层（即最底层）序列的预测在未来未知的均值，误差  $\boldsymbol{\varepsilon}_h$  有均值 0 和未知的方差  $\Sigma_h$ 。Hyndman 根据预测近似满足层次聚合结构（对于任何合理的预测集都应发生），提出了假设  $\boldsymbol{\varepsilon}_h \approx S \cdot \boldsymbol{\varepsilon}_{K,h}$ ，即误差也近似满足层次聚合结构，从而推导出对  $\boldsymbol{\beta}_{t+h|t}$  的最佳线性无偏估计为  $\hat{\boldsymbol{\beta}}_{t+h|t} = (\mathbf{S}^T \mathbf{S})^{-1} \mathbf{S}^T \hat{Y}_{K,t+h|t}$ ，从而有

$$\tilde{Y}_{t+h|t} = \mathbf{S}(\mathbf{S}'\mathbf{S})^{-1} \mathbf{S}^T \hat{Y}_{K,t+h|t}$$

其中， $\tilde{Y}_{t+h|t}$  表示校准过的各层预测，同时，Hyndman 还指出，所有的 HTS 预测方法可以视为

$$\tilde{Y}_{t+h|t} = \mathbf{S}\mathbf{P}\hat{Y}_{K,t+h|t}$$

其中，矩阵  $\mathbf{P}$  用于对基层的预测之间提取共同元素并校准，加总矩阵  $\mathbf{S}$  则用于将基层

预测映射到各层。

这种方法使用层次结构中可用的所有信息，对独立基础预测进行优化组合来生成一组尽可能接近单变量预测的修正预测，并允许在层次结构的每个层次上的序列之间进行关联和交互。它对任何级别的预测进行特殊调整，因此，如果基础预测是无偏的，它将生成无偏的调优预测。

然而，这个方法对置信区间的预测会依赖于位置的残差方差阵  $\Sigma_h$ ，因为其使用的 Generalized Least Squares (GLS) 方法推导出的  $\text{Var}(\hat{\mathbf{Y}}_{t+h|t}) = \mathbf{S}(\mathbf{S}^T \Sigma_h^{-1} \mathbf{S})^{-1} \mathbf{S}$ ，因此有不少学者认为这个矩阵由于可识别性 (identifiability) 的限制不可能被估计，并且 Hyndman, Lee, and Wang (2016) 后提出使用 Weighted least squares 并使用预测的误差方差来估计  $\Sigma_h$  的方法也没有解决这个问题[1]。

### (3) 其他优化方法

在 Hyndman 等人 2011 和 2016 年研究结果基础上，Shanika L 等人 (2019) 提出了最小路径调优法 (Minimum trace reconciliation, MinT) [20]，并证明其调优后所得的一致性预测保证至少与基本预测一样好且大大提高了计算效率，Tiago 等人 (2020) 研究电力行业案例时进一步提出对 GLS 的 W 矩阵做压缩估计 (Shrinkage) 的 Shrink MinT 方法。这些方法预计在将来还会被不断改进，但其本质上都基于最优结合法的模型形式[21]。

## (三) 研究内容与本文结构

本文创新性地将 HTS 技术分别与机器学习模型 LightGBM 和新兴的 fbprophet 时间序列模型结合，试图给零售提供一个训练多层级历史商品销量时间序列数据并能综合天气、节假日等因素的预测方案，从而帮助赋能零售企业进行库存的精准把控。

本文的第二部分将具体分析一个典型的商品销量分成时间序列数据并作为实证所用。第三部分将介绍所使用模型，第四部分将展示模型结果，第五部分做结论与展望并探讨研究后续可能的改进方向。

## 二、典型商品销量分层时间序列数据的分析

### (一) 数据基本情况

#### 1. 数据来源

本数据使用的零售巨头沃尔玛在美国 10 家店铺的销量、价格数据，提供方为尼科西亚大学 (University of Nicosia) 的 Makridakis 开放预测中心 (Makridakis Open Forecasting Center, MOFC)，数据于 2020 年 3 月发布于 Google 旗下的数据科学竞赛

平台 Kaggle，吸引了 5558 支队伍参赛使用该数据进行分析建模。

## 2. 数据时间跨度

首先，在时间维度上，数据涵盖 2011 年 1 月底至 2016 年 6 月中旬的销售数据，如图 2 所示其中前 1913 天为训练集、最后四周作为测试集（目前数据来源方尚未公布数据），测试集前四周数据就作为模型验证集（数据来源方已公布）。

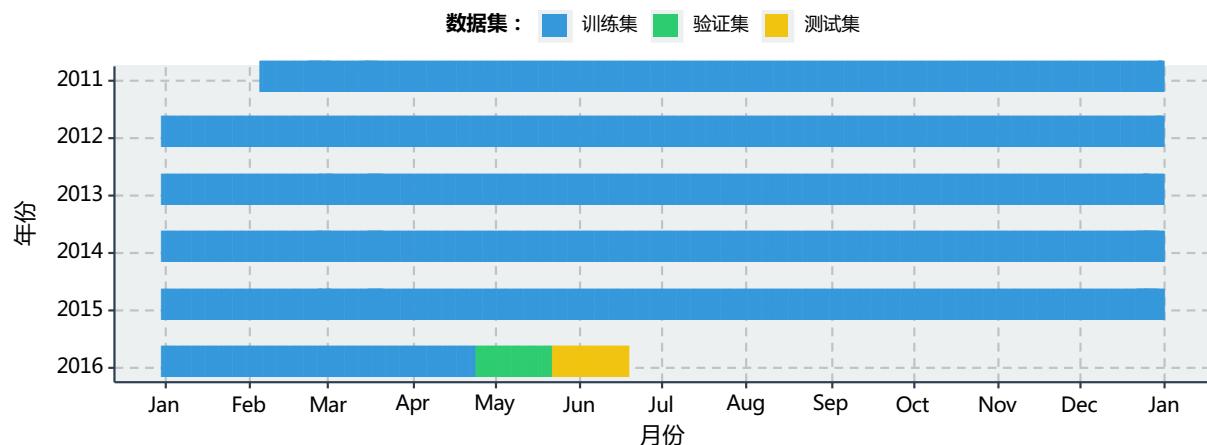


图 2 数据集划分示意图

## 3. 分层变量与分层结构

在该 HTS 数据的分层维度上，数据按分组顺序来自 3 个州中的 10 家商店、每家商店有分为 3 大类商品和 7 类细分商品的 3049 条商品时间序列，共有 30490 条商品销量时间序列。

表 2 分层变量释义及取值范围表

变量名	变量释义	取值范围
state_id	州编号	CA、WI、TX
store_id	商店编号	CA1、TX3 等 10 类
category_id	商品大类编号	Hobbies、Foods、Household
department_id	商品细分类别编号	Hobbies1、Household2 等 7 类
item_id	商品编号	1~3049
date	日期	2011 年 01 月 29 日~2016 年 6 月 19 日
Sales	商品在该日期的销量	0~572

若以州、商店为第一类分层维度、以商品大类、细分类、商品为第二组分层维度则

时间序列可以展开成图 3 所示结构<sup>1</sup>。

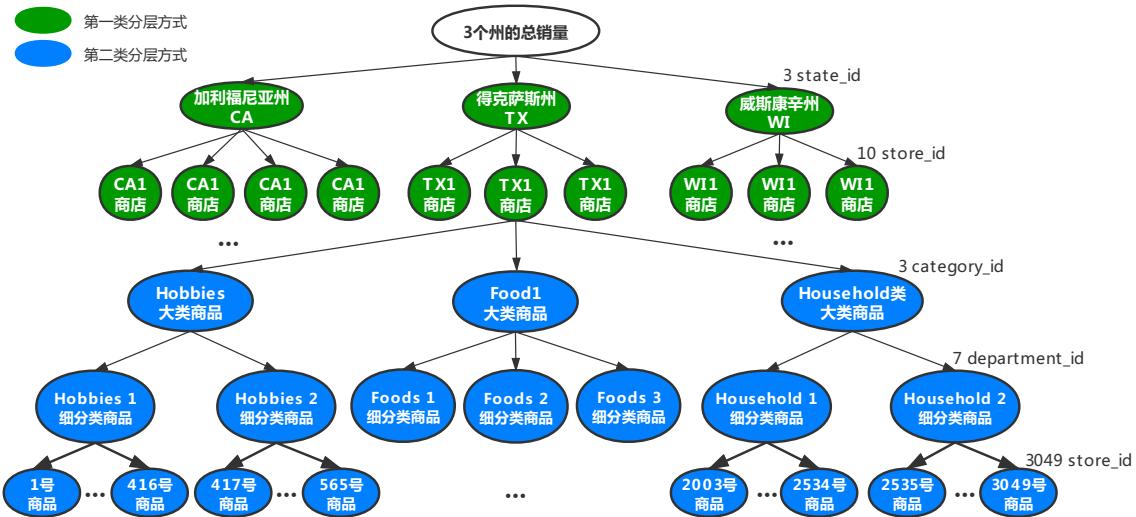


图 3 数据分层结构示意图

因此，最底层的 30490 条序列可按不同的分组组合形式向上层汇总，共有 11 种汇总方式共可形成 11830 条不同的序列，从而共产生 42840 条序列，具体如表 3 所示：

表 3 不同层级数据的加总方式及序列数量统计

加总层级	记号	加总方式	时间序列数量
第一层	L1	全部加总的顶层序列	1
第二层	L2	按不同的州加总	3
第三层	L3	按不同的商店加总	10
第四层	L4	按不同的大类加总	3
第五层	L5	按不同的细类加总	7
第六层	L6	按不同的州、大类加总	9
第七层	L7	按不同的州、细分类加总	21
第八层	L8	按不同的商店、大类加总	30
第九层	L9	按不同的商店、细分类加总	70
第十层	L10	按不同的商品加总	3,049
第十一层	L11	按不同的商品、州加总	9,147
第十二层	L12	完全分解的底层序列	30,490
总计			42,840

<sup>1</sup> 两组维度满足“分组时间序列的特点”可调换，但内部的顺序是严格的“分层时间序列”结构，不可交换

#### 4. 外生变量

除了时间、因变量以及用于分层的变量之外，数据还包含一些外生于时间序列的变量。他们刻画了影响商品销量的诸多其他因素，具体如表 4 所示。

表 4 外生变量释义及取值范围表

变量名	变量释义	取值范围
Weekday	周	1~7
Events	节日	如复活节、感恩节、超级碗等 21 类
Events_type	节日类型	文化、宗教、体育、全国
Price	商品定价	0.99~1999 (美元)
SNAP <sup>2</sup>	SNAP 券允许日	CA 允许、TX 允许等
item_id	商品编号	1~3049
date	日期	2011 年 01 月 29 日~2016 年 6 月 19 日
Sales	商品在该日期的销量	0~572 (件)

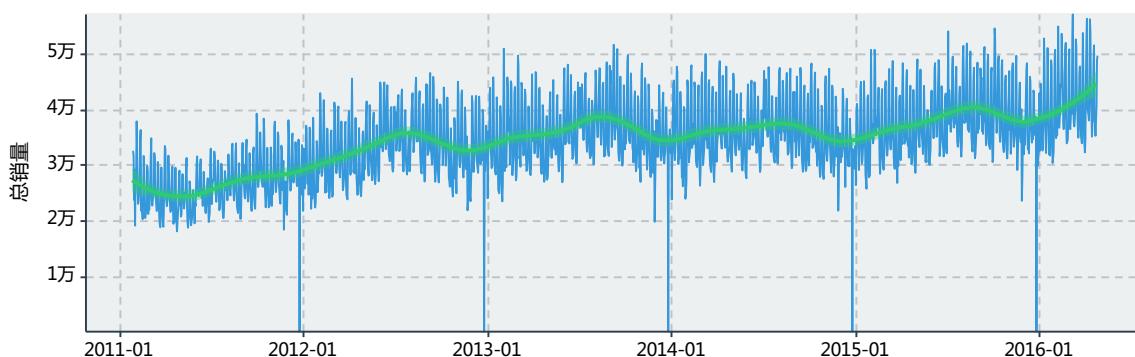
#### (二) 数据描述性统计

本部分继续说明多级分层变量和多个外生变量影响下商品销量的描述性统计规律。

##### 1. 分层变量与时间序列的关系分析

###### (1) 顶层序列

从图 4 可看出：首先，在 2011 到 2016 年间的销量整体呈增长趋势；第二，存在不同频率的季节性，每年在 10 月左右销量会达到峰值、且每月中的波动性也很显著；另外，注意到每年的 12 月 25 日因圣诞节歇业销量为 0，由于不在预测范围内因此预先进行剔除处理。

图 4 顶层 (L1) 序列时间序列及趋势拟合图<sup>3</sup>

<sup>2</sup> SNAP (Supplemental Nutrition Assistance Program) 是美国农业部发放给困难家庭的食物券，旨在帮助困难家庭购买健康食品，在 Walmart 中，SNAP 券可抵扣部分商品价格。

<sup>3</sup> 平滑的趋势线使用局部加权多项式回归而成，下文同。

## (2) 中间层序列

底层序列可以向上加总成十一个不同的层级 (L1 为顶层、L12 为底层, 如表 3 所示), 而在不同层之间, 还可以对同层时间序列进行横向对比。比如由图 5 可看出, 在不同州 (L2)、不同商店 (L3) 层面对比加州的第三家商店销量整体较高、加州 1 号和 3 号商店的季节性较为明显, 威斯康辛州的一号和二号商店在 2012 年中和年末销量有较大波动。

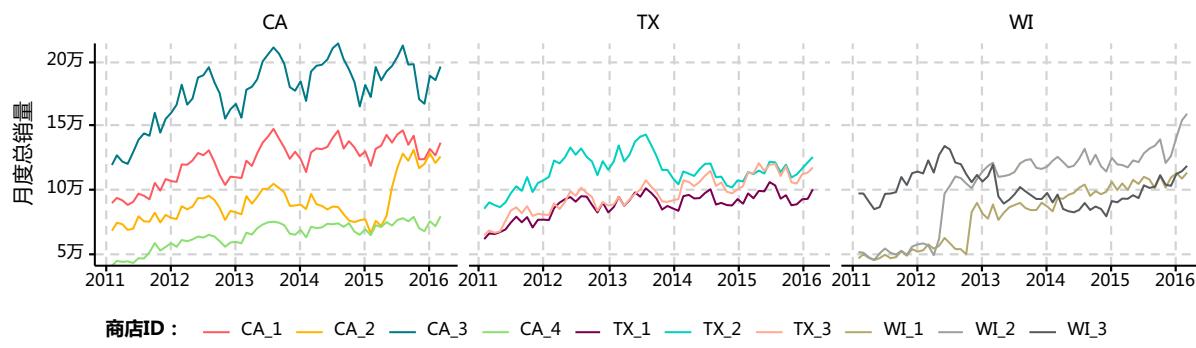


图 5 按不同商店分 (L3) 月度销量时序图

而从图 6 直接查看日频数据的分布和箱线图特征可以更明显地看出: 威斯康辛州两家店日销量在 5 万左右低位形成第二峰值。以及加州二号店在 13 万左右高位形成第二峰值, 其余店的日销量则大致满足正态分布形状。

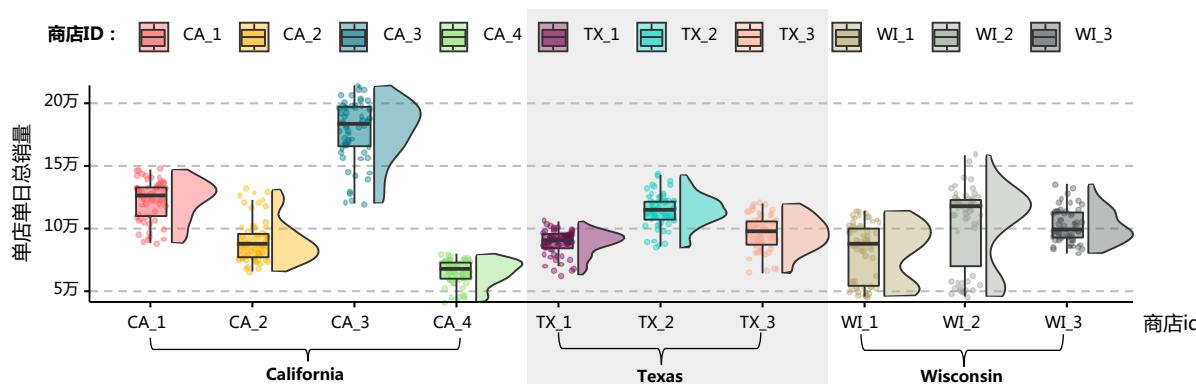
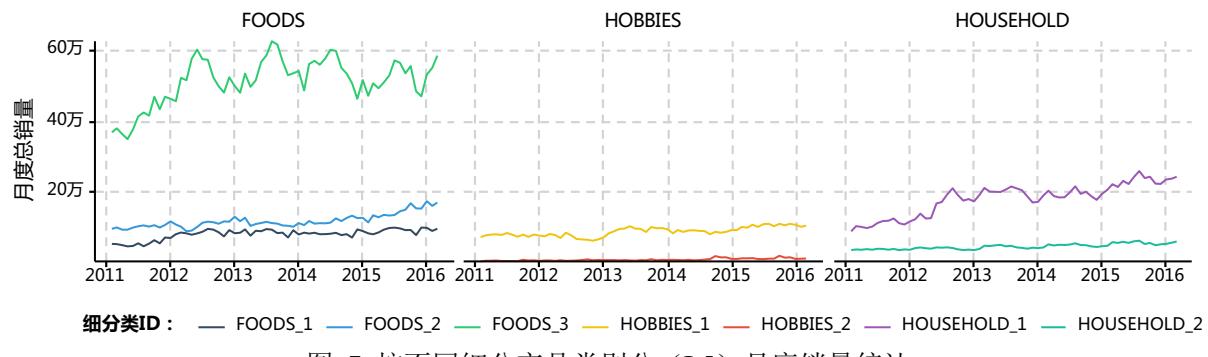


图 6 按不同商店分 (L3) 日度销量云雨图

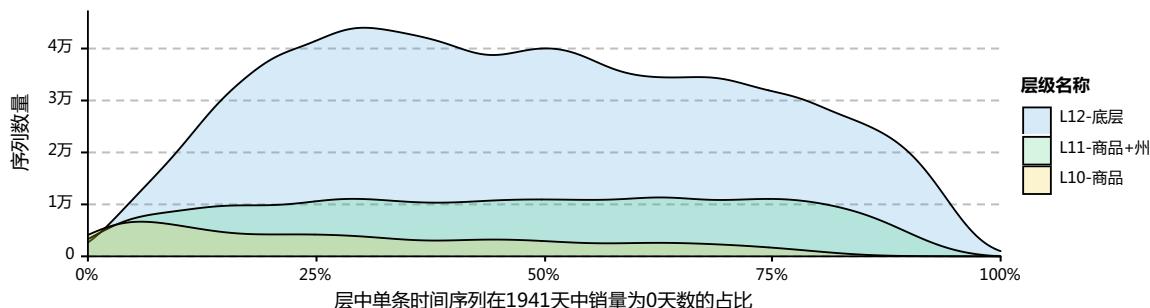
若根据不同商品的大类 (L4) 和细分类 (L5) 的横向对比, 则如图 7 所示第三类食物的销量明显高于其他且季节性很明显, 而 HOBBIES2 类产品的销量很低, 结合其较高的均价推测可能多为奢侈品。



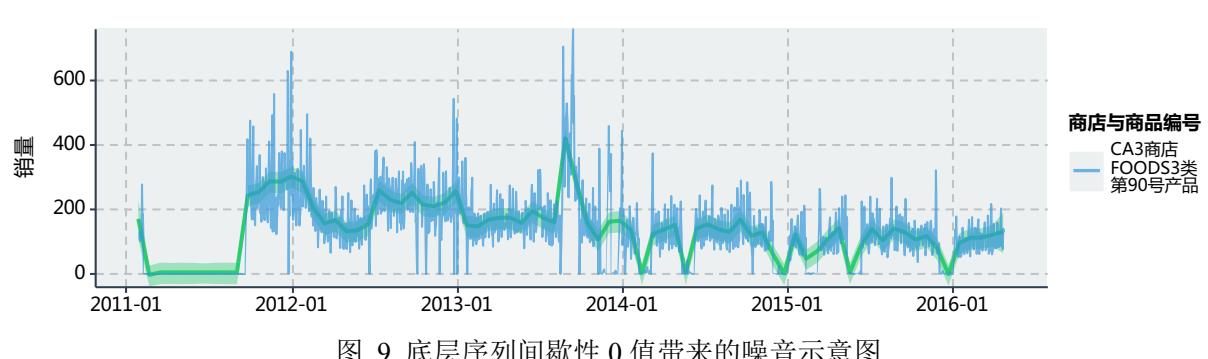
### (3) 底层序列

本文所用数据集的底层时间序列噪声非常大。

首先，噪声来自 0 销量的出现。如图 8 所示，底层 (L12) 中绝大部分序列在预处理后仍然有 25% 以上的 0 值，而按商品和州、商品向上加总两层后，可以看到 3079 条第十层序列仍然有不少高缺失序列，意味着一款商品 1 天里在 10 商店都没有卖出 1 件的情况时常发生。



其次，这最底层的 30490 条序列中的平均销量水平也各不相同，有些高价格“爱好”类产品每日的最大销量也不超过 5 个，而如图 9 所示，食品类产品通常日售百件以上。



再者，即使高价格的产品也会因为“断货”等因素出现间隔性的 0 销量。

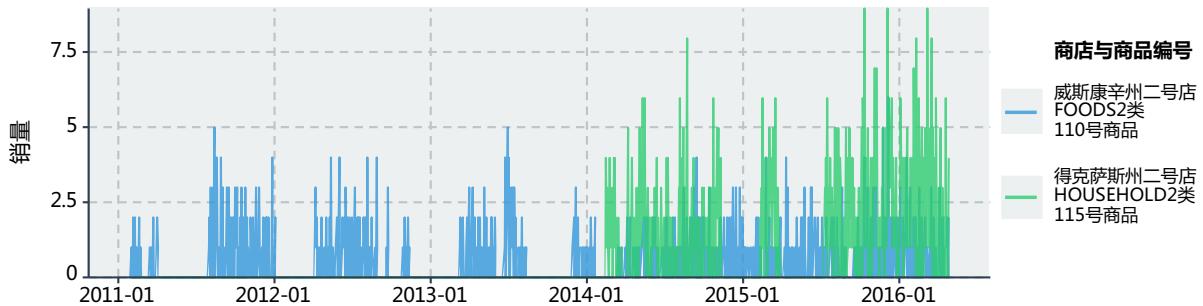


图 10 底层序列销量水平差异示意图

可以看出，Walmart 的销量数据充分反映了 HTS 底层序列噪声大这一典型特征，作为研究的实证数据具有一定的代表性。

此外，该数据的底层序列还会受价格、节假日等诸多外生变量的影响，从而能够将波动性向各级上层依次传导，导致加总局部的局部震荡或系统性的出现季节性特征。因此，本文将于下一部分讨论外生自变量对时间序列带来的影响。

## 2. 时间序列受外生变量的影响

### (1) 价格

首先看价格在各层的差异。从图 11 按州和大类 (L6) 分层后的 9 条序列的价格信息可以观测到：统一类商品在不同州的定价基本没有差异（只在食品大类中有细微差别）。而在不同商品类别中，食品平均比家庭用品便宜。与其他两种商品相比，HOBBIES 类商品的价格覆盖的范围更广、且其 HOBBIES 中的第 2 类商品有价格集中峰值。

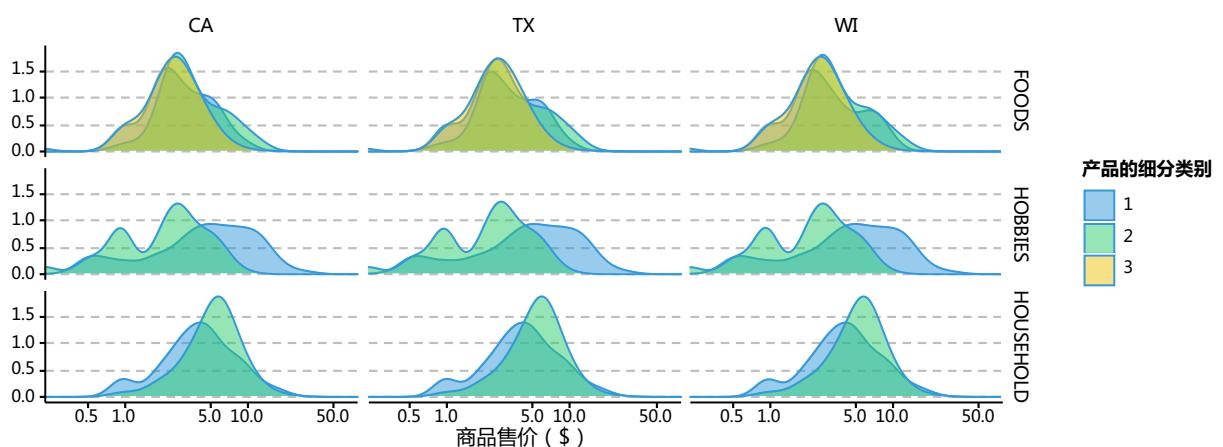


图 11 分层维度对价格影响示意图

其次看价格差异对底层序列的影响。价格的调整跟销量的关系在底层是动态互相影响的，从图 12 可以看出，商品在一段时间没有售出后，企业可能会采取降价的措施来刺激销量，待销量回暖之后可能会上调价格扩大利润但减少销量，结合第一部分所说的“缺货”问题，调价实际上更加剧了底层序列的不稳定性。

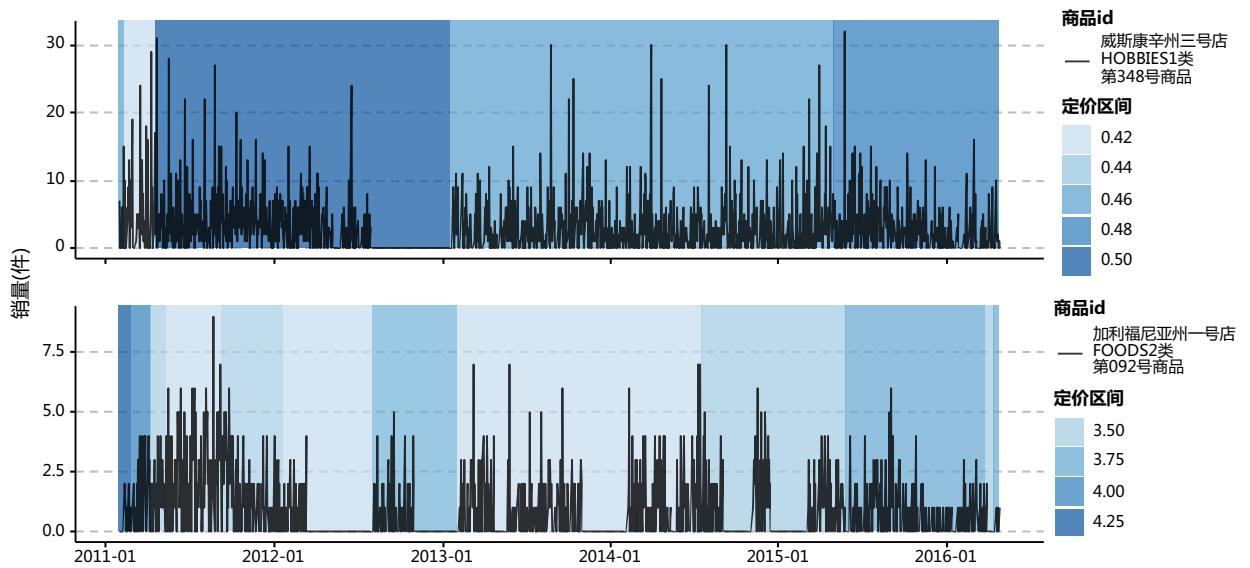


图 12 价格调整对底层序列的影响举例

## (2) 工作日与非工作日

销量受到工作日与非工作日的影响也很大，人们通常偏好在周末的时候购物因而周末的销量更高，但值得注意的是在七月份周五的销量也很高，此外周二到周四的销量在各月都低于平均。

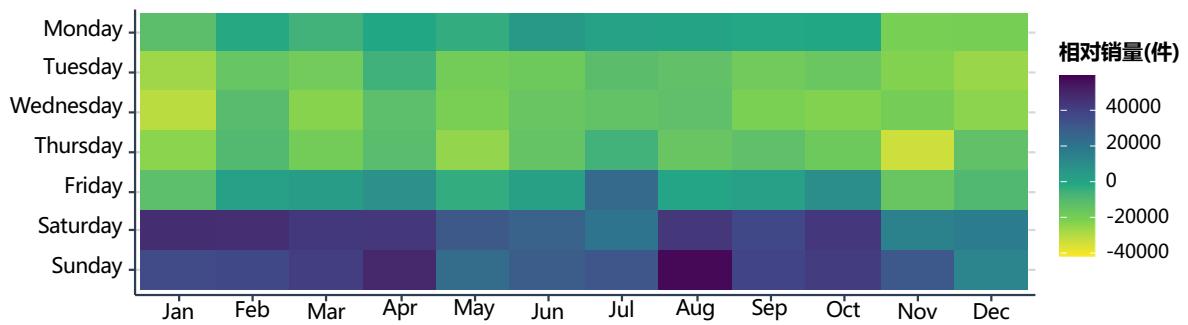


图 13 相对销量在周、月间日期热力图

如果将 10 家五年来的单日销量按周一到周日分组绘制分布图，可以看出周六日有近 50%以上的单店单日销量在 4000 件以上、约 30%以上的周末可达 5000 件，但工作日

则近 90%的日子一家店销量过不了 5000。

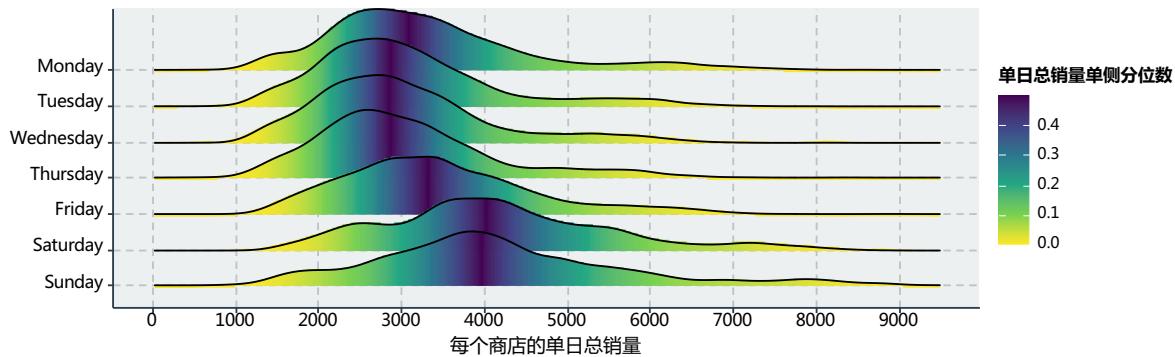


图 14 不同周次下各店单日销量分布峰峦图

因此，星期也是影响销量的重要变量，在后续建立模型时，基于特征工程的模型需要考虑 7 位单位的各阶滞后、基于时间序列的模型则需考虑 7 天为季节性长度。

### (3) 节假日

在销售额（销量\*价格）方面，从图 15 上图可以看到三件产品每日的总销量差别较大，食物和类的销售额最高。图中蓝线（正常日）和绿线（节假日）对比可以看出，不同商品对节日的效果不同，食品类对受节假日的影响不大，但其他两类商品价格都会降低。然而进一步考虑节假日类型，发现在“体育”和文化活动期间，食品的销量实际上更高，而“全国性”和“宗教性”活动都会导致销售量的相对下降与前面结论一致。

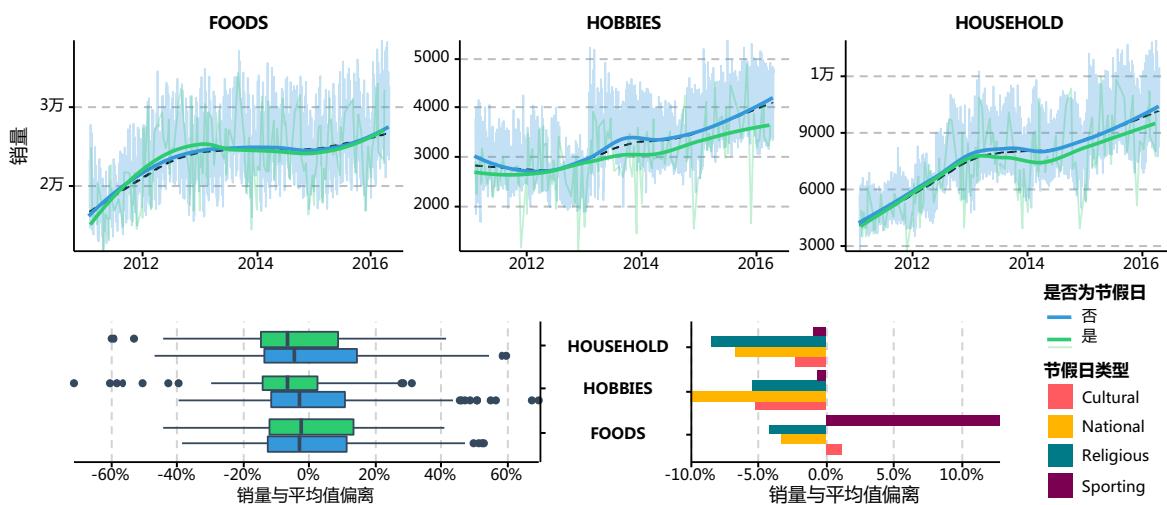


图 15 商品大类 (L4) 对节日因素影响示意图

而在不同州的对比中我们发现，2014 年前的德克萨斯州节假日对销量影响为正，而

威斯康星州 2014 以后，节假日对销量有正影响。不同节假日中，“全国性”节假日对其销量的影响的负面影响也很大。

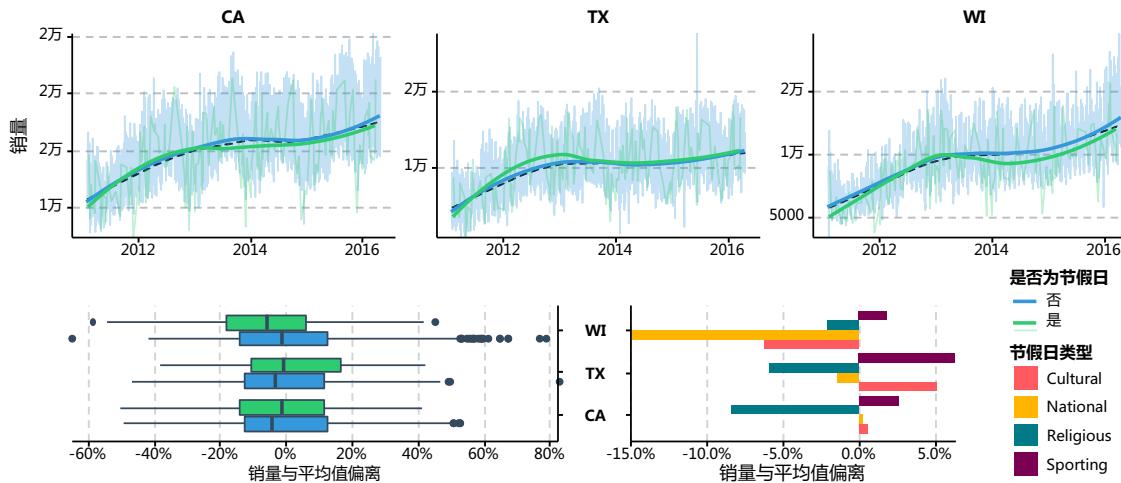


图 16 商品所处州 (L2) 对节日因素影响示意图

综合来看，不仅不同的层级加总后的同层序列之间有差异、其受外生变量影响的大小和方向也会有区别。这种非线性的关系可能是传统时序模型难以捕捉到的。因此本文将探索能高效利用 HTS 技术的底层时间序列模型。

#### (4) 补贴与折扣

在商品销量数据中，商家所应用的营销折扣或补贴也是一种可以事先确定的外生变量，在本数据集中，沃尔玛在不同的州会收取营养补助食品券（Supplemental Nutrition Assistance Program, SNAP），每个月的具体日期是固定的但不同州有差异，以 2012 年前 6 个月为如图 17 所示。

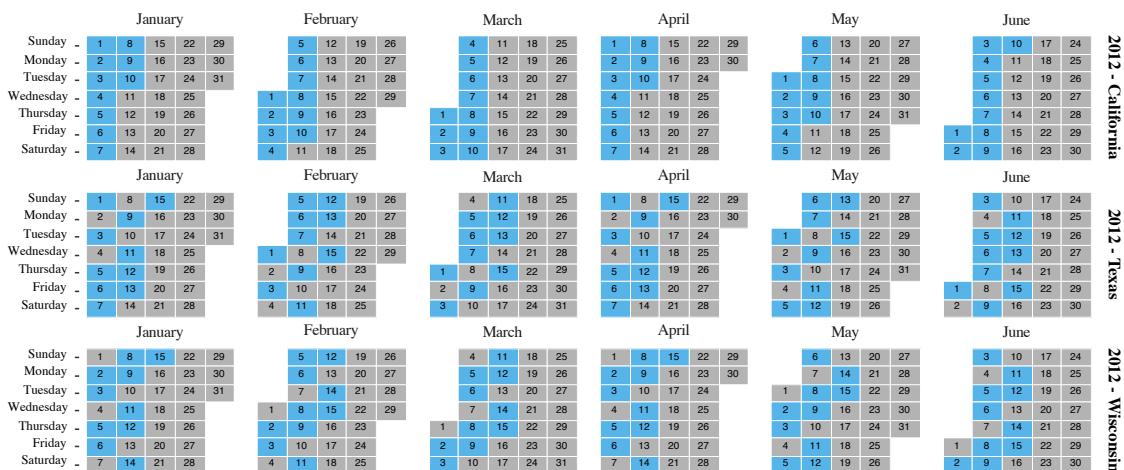


图 17 SNAP 日的日期分布示意图

而补贴对销量的影响可以清晰地在图 18 中看到，在每个周不管是相对销量还是绝对销量都有显著的提升效应。

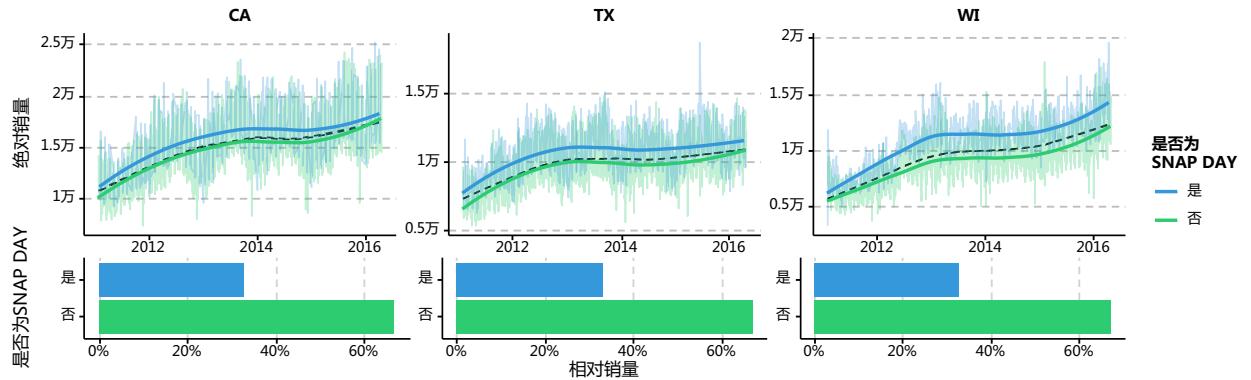


图 18 按不同州 (L2) 的 SNAP 日对销量影响

### 三、针对分层时间序列预测问题的模型方案

该部分将详细介绍基于 Prophet 和 LightGBM 模型构建的两个 HTS 校准预测方案，两个方案整体数据处理、特征工程、模型训练的数据流向示意图如图 19 所示。

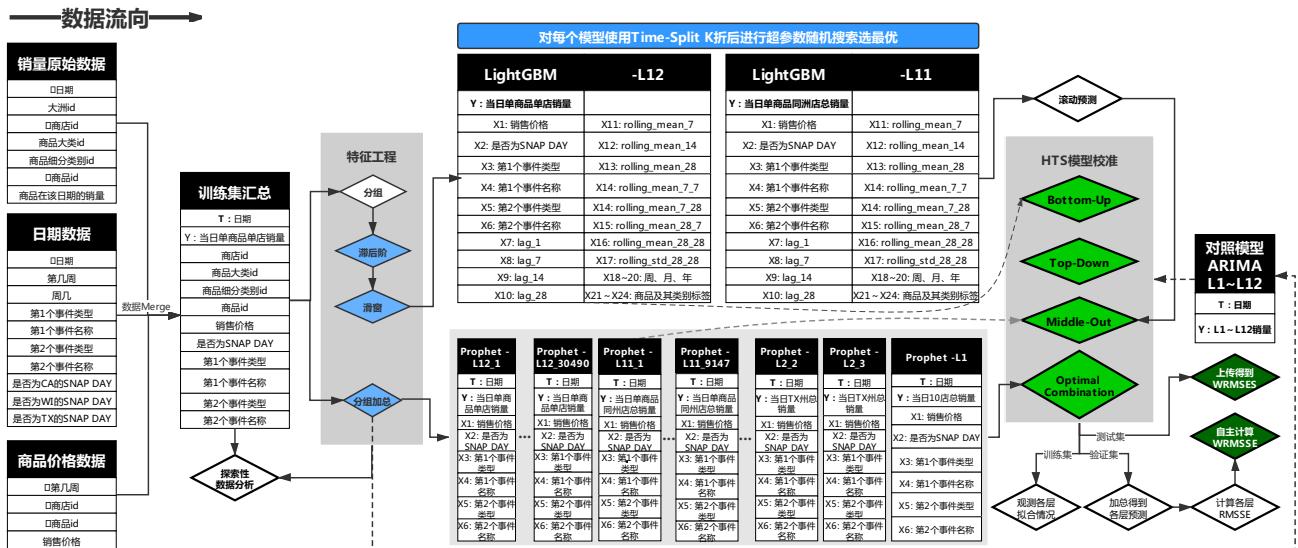


图 19 模型构建与预测数据流向示意图

#### (一) 方法一：基于 HTS 调优的 Prophet 模型

##### 1. Prophet 模型基本介绍

Prophet 模型全称 Fbprophet，是一个新型时间序列预测框架，由 Facebook 在 2017 年 2 月开源。自发布以来已有许多学者将其应用于金融[22]、宏观经济[23]、电子商务

[24] 和公共卫生领域[25]。该模型与 Hastie 等人 1987 年提出的广义叠加性模型 (Generalized Additive Model, GAM) 的想法一脉相承, Prophet 是一个曲线拟合 (Curve-fitting) 模型的生成模型 (Generative Model), 其与 ARIMA 等模型的本质区别在于他的模型设定不会显式地表现模型的结构 (dependence structure)。

选择 prophet 作为底层模型的原因有两方面。一方面针对单个时间序列, Prophet 模型在预测上比传统时序模型有明显优势。第一, 与 ARIMA 等模型不同, prophet 不需要序列有规律性的间隔, 也不需要手动处理异常值和填充缺失值, 从而无需大量的数据预处理, 适用于含有噪声的时间序列数据。第二, prophet 模型可以灵活地调整周期性如接受多个季节性, 其预算速度不会如 ARIMA 等模型一样大大降低。第三, Prophet 可以接收外生变量。最后, 模型高度封装、调参建议, 适合非专业科研人员使用。另一方面, 将 Prophet 模型应用于 HTS 方法时, Prophet 的好处还体现在其拟合速度很快, 尤其是现今经过多轮的性能优化之后, 因而适用于大规模的时间序列预测应用部署当中。

## 2. Prophet 模型应用原理

### (1) 单序列模型搭建

在对 Prophet 模型的应用中, 模型由商品销量的增长趋势项 $g(t)$ 、销量的周期性因素 $s(t)$ 、节假日项 $h(t)$ 和误差项 $e$ 四部分组成:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

其中:

- $g(t) = \frac{C}{1+e^{(-k(t-m))}}$  表示商品销量时序中非周期性变化的部分, 使用的是一个指数增长模型, C 为销售完全饱和时的销量常数, 实际应用中可以将 C 设置为 $C(t)$ , 基于销售人员对未来各个时点市场规模的判断, 也可以引入人口等变量进行估计。k 为增长率, m 为一个日期偏差抵消参数。
- $s(t) = \sum_{n=1}^N \left[ a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right) \right]$ 是一个标准的傅立叶级数, 表示销量每周、每月、每季、每年等周期性的季节因素 (Seasonality), t 表示当前所处的时间期, P 表示每个周期的时间长度, 该模型在给定 N 和 P 下需要借助傅里叶级数对  $2N$  个参数  $[a_1, b_1, \dots, a_N, b_N]^T$  进行估计, 比如  $N=10$  时, 模型会对  $s(t) = X(t)\beta, X(t) = [\cos\left(\frac{2\pi(1)t}{365.25}\right), \dots, \sin\left(\frac{2\pi(10)t}{365.25}\right)]$  进行 GLS 拟合, 而 N 则会通过 AIC 准则自动选取。
- $h(t) = Z(t)\kappa_i, \kappa_i \sim \text{Normal}(0, \sigma)$  ( $Z(t) = [1(t \in D_1), \dots, 1(t \in D_i), \dots, 1(t \in D_L)]$ ) 表示每个节假日 i 对应的所有未来和过去节假日集合  $D_i$  对时序列带来强度为  $\kappa_i$  的影响。

- $\epsilon_t$  是被假设为服从正态分布的误差项。

### (2) HTS 模型校准方法与 Prophet 结合

当前, 国内外学者主要将 prophet 模型应用于单一的时间序列, 尚未有人将其与 HTS 领域的模型校准方法结合使用。但事实上, 根据商品销量数据受强季节性、节日、价格等因素影响的特点, 选用 Prophet 作为 HTS 的底层模型具有很强的潜在实用价值。同时, Facebook 对于其应用于大规模时间序列预测的设计初衷也指导我们使用多模型的预测思路。针对 Prophet 模型本文应用了两种校准方法:

#### ①自中向外法

首先, 我们对按商品细分类、商店加总 (L11 层), 并对形成的 9147 条序列建立 HTS 模型, 然后对下层的序列按照其过去 6 个月的销量和占总销量的比来确定权重  $p_j = \frac{1}{180} \sum_{t=n-180}^n \frac{Y_{L11,t}}{Y_t}, j = 1, \dots, 3$  (WI 和 TX) 或 4(CA), 再使用  $p_j$  来向下分解 9147 个模型对未来 28 天的预测值。而对上层则使用 9147 条序列依次往上加总。

#### ②最优结合法

第二种方法则直接对所有层的 42840 个序列建立 42840 个模型并得到预测  $\tilde{Y}_{K,t+h|t}$  ( $K = L1, L2, \dots, L12$ ), 由于建模过程是互相独立的, 所以生成的序列之间会产生加总谬误, 如对一个商店七类商品分别的预测加起来可能不等于对商店的预测。

首先定义  $S$  是一个  $30490 \times 42840$ , 秩为 42840 的加总矩阵 (summing matrix), 如前文所述其根据  $\hat{Y}_t = S\hat{Y}_{K,t}$  将 L12 层序列向上加总得到  $K = L1, L2, \dots, L11$  层序列。

最终根据

$$\tilde{Y}_{t+h|t} = S(S'S)^{-1}S^T\hat{Y}_{K,t+h|t}$$

可以得到加总谬误被 GLS 最小化的校准预测值  $\tilde{Y}_{t+h|t}$ 。

## (二) 方法二: 利用特征工程转换数据的 LightGBM 模型

### 1. LightGBM 模型基本介绍

微软 2017 年开发的开源算法 LightGBM (Light Gradient Boosting Machine) 解决了梯度提升决策树 (Gradient Boosting Decision Tree, GBDT) 在特征维数高、数据量大的情况下, 由于需要对每个特征扫描所有数据实例来估计分割点的信息增效率而降低效率的问题[27]。

LightGBM 主要在 GBDT 基础上引入了三个改进: 基于梯度的单边采样 (GOSS)、互斥特征捆绑法 (EFB) 和直方图算法 (Histogram algorithm), 三个方法都大大加速了

训练过程和内存开销，并且实现决策树的并行计算，因此在工业界和数据科学竞赛中应用广泛，常被用于点击率预测、多分类、搜索、排序等。有研究统计近年 Kaggle 比赛有一半以上的金牌冠军方案都基于 GBDT[28]，而 LightGBM 对其的改进又大大拓宽了树模型的应用前景。

## 2. LightGBM 模型应用原理

目前，将 LightGBM 应用在时间序列领域的研究主要有水文[29]、风力[30]等，也有学者将其用于与其他模型叠加使用[31]，在工业应用中今年更是成为许多 Kaggle 数据集的冠军方案。对模型的应用需要首先通过特征工程将时间序列建模转化为一个有监督学习（supervised learning）问题。

在本文中，LightGBM 和 Prophet 方案一个很大区别在于模型的个数。Prophet 接收的是一整条时间序列，因此需要训练的模型数就等于输入的时序数，而 LightGBM 则在输入每条序列时需要将其提取特征形成多行数据（一条完整的 1913 天销量时序忽略开始 28 天后可生成 1818 行训练集），但在每一层只需训练一个模型，在每条序列都完整时该模型接收的数据量约为  $m_K \times 1818$ ，其中  $m_K$  为该层时序数量

### (1) 特征工程

首先，我们先对各个序列的时序特征通过特征提取转换为横截面数据，从而应用 LightGBM 模型进行有监督学习和预测。

表 5 LightGBM 模型特征工程后提取变量列表

变量类型	变量名	变量释义
因变量	sales	商品在本州商店（L11 层）中的总销量
日期类	day	以 2011 年 1 月 29 日开始的日期标记
	dayofweek	表示星期几
	month	第几个月
	year	哪一年
商品信息类	price	商品价格
	store_id	商品编号
	category_id	商品大类编号
	department_id	商品细分类别编号
商店信息类	state_id	商店所在大洲
销量滞后阶特征	lag_1	商品昨天的总销量
	lag_7	商品一周前的总销量
	lag_14	商品两周前的总销量
	lag_28	商品四周前的总销量
销量滑窗特征	rolling_mean_7	商品过去一周的总销量

销量滑窗滞后特征	rolling_mean_14	商品过去两周的总销量
	rolling_mean_28	商品过去四周的总销量
	rolling_mean_7_7	商品两周前到一周前的总销量
	rolling_mean_7_28	商品五周前到一周前的总销量
	rolling_mean_28_7	商品五周前到四周前的总销量
	rolling_mean_28_28	商品八周前到四周前的总销量
	rolling_std_28_28	商品八周前到四周前的总销量的标准差
其他外生日期变量	is_snap	是否为该州的 SNAP 日
	event_name_1	第一个节日的名称
	event_type_1	第一个节日的类型
	event_name_2	第二个节日的名称
	event_type_2	第二个节日的类型

注意到，特征工程可能还需要在模型预测环节进行。部分滞后阶和滑窗的特征需要结合未来预测的数据来生成测试集的自变量，因此需要使用滚动向前预测（recursively prediction）的方法，即在每得到新一天的数据点后利用该预测值、过去的真实值、已预测值重新进行一次特征提取（除商品八周前到四周前的总销量的标准差，其不含在测试集中因此不需要），才再继续进行预测下一天的销量值，该过程如图 20 所示。

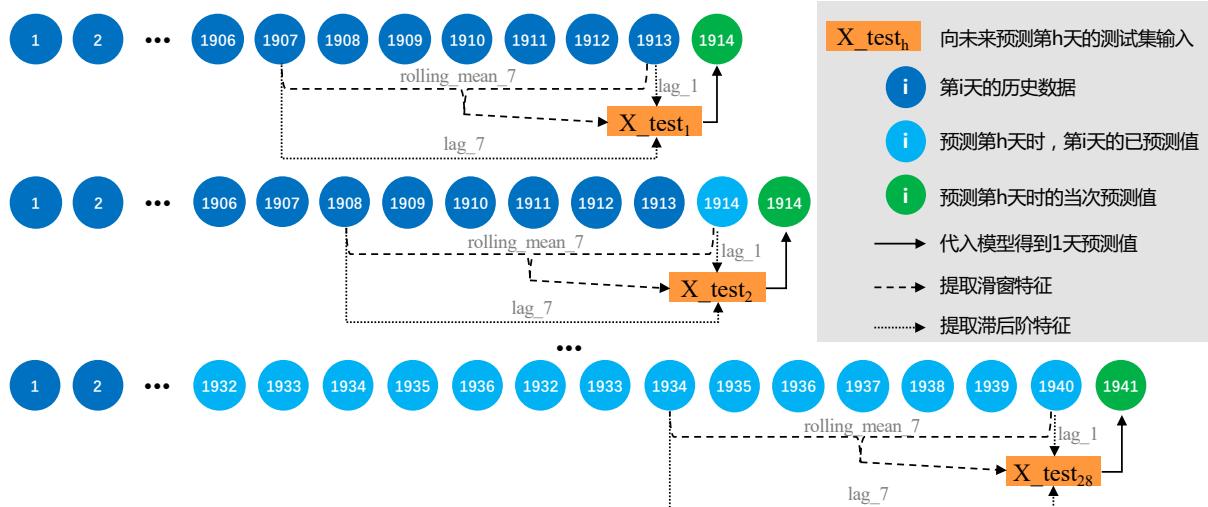


图 20 LightGBM 模型滚动预测法示意图（以 3 个特征为例）

## (2) 单序列的模型搭建

对已经转化为销量数据训练集  $X = \{(x_i, y_i)\}_{i=1}^n$ ，LightGBM 寻找对真实函数  $f^*(x)$  最小化损失  $L(y, f(x))$  的估计  $\hat{f}(x)$ ，问题可以写成：

$$\hat{f} = \arg \min_f L(y, f(x))$$

其中损失函数在本例中使用 Poisson 损失  $L(y_i, \hat{f}(x_i)) = \sum_{i=1}^n (y_i \hat{f}(x_i) - e^{\hat{f}(x_i)})$ , 而  $f(x)$  可用 T 的决策树加总而成:

$$f(x) := f_T(X) = \sum_{t=1}^T f_t(X)$$

因此对训练到第 t 步的 LightGBM, 新加入一个基决策树模型  $f_t(x_i)$  所面临的损失函数可写成:

$$L_t = \sum_{i=1}^n L(y_i, f_{t-1}(x_i) + f_t(x_i)), \hat{f}_0(x) = \arg \min_f L(y, f(x))$$

即新集成的决策树以上一组树拟合的残差为目标继续最小化 Loss, 使得每次迭代中的 Loss 都沿着负梯度方向减小。

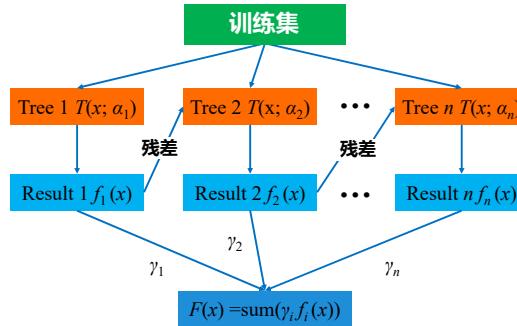


图 21 GBDT 模型的迭代拟合残差的一般方法示意图

传统的 GBDT 方法采用最速下降法, 只考虑损失函数的梯度  $-\left[\frac{\partial L(y_i, f_t(x_i))}{\partial f_t(x_i)}\right]_{f(x)=f_{t-1}(x_i)}$ 。而 LightGBM 中采用牛顿法更快地逼近目标函数, 对上式省略常数项后可以进一步写成

$$L_t \cong \sum_{i=1}^n \left( g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right)$$

其中,  $g_i$  和  $h_i$  分别表示  $L_i$  的一阶梯度和 V 二阶梯度:

$$\begin{aligned} g_i &= \partial_{f_{t-1}(x_i)} \Psi(y_i, f_{t-1}(x_i)) \\ h_i &= \partial_{f_{t-1}(x_i)}^2 \Psi(y_i, f_{t-1}(x_i)) \end{aligned}$$

LightGBM 在这里继续使用一个单边梯度采样 (Gradient-based One-Side Sampling, GOSS) 技术来排除大部分梯度小的样本 (梯度小的样本已被模型充分学习), 记  $I_j$  为 LightGBM 选择出的样本叶子 j, 对一个决策树的样本决策树, 记其样本权重  $w_{q(x)}$ ,  $q \in \{1, 2, \dots, J\}$ ,  $J$  为叶节点的总数,  $q(x)$  为决策树的决策准则, 则损失函数可写为:

$$L_t = \sum_{j=1}^J \left[ \left( \sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left( \sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right]$$

因此给定一个树结构  $q(x)$ , 每个树的最优化问题可以转化为一个二次函数的最优化, 每个叶节点的最佳权重得分  $w_j^*$  和损失函数的极小值  $L_T^*$  可以解为:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$$

$$L_T^* = -\frac{1}{2} \sum_{j=1}^J \frac{\left( \sum_{i \in I_j} g_i \right)^2}{\sum_{i \in I_j} h_i + \lambda}$$

其中,  $L_T^*$  可以看作是评估数结构  $q$  质量的得分方程, 因此目标函数在增加这一轮分割后的增益为

$$G = \frac{1}{2} \left( \frac{\left( \sum_{i \in I_L} g_i \right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left( \sum_{i \in I_R} g_i \right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{\left( \sum_{i \in I} g_i \right)^2}{\sum_{i \in I} h_i + \lambda} \right)$$

其中,  $I_L$  和  $I_R$  分别表示左支和右支。在 GBDT 算法中, 会对所有叶子的信息增益都进行计算并同时向下分裂, 但实际上, 许多信息增益较低的叶子不需要进行继续的搜索和分割。LightGBM 的策略则是只分裂在同一层上信息增益中最大的叶, 并通过超参数控制树木深度不要过高, 从而尽量避免过拟合。

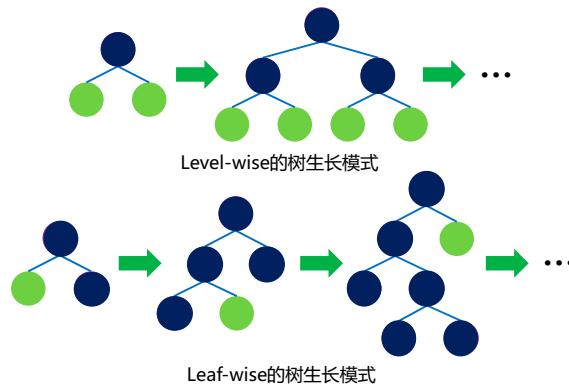


图 22 树模型按层与按叶生长模式对比图

此外对数据, LightGBM 还会预先将其处理成直方图形式, 把连续的浮点特征值离散化为分箱, 并对原先互斥的特征(如不同节日等)采用特征合并算法(Exclusive Feature Bundling, EFB), 将其捆绑。最后, 在分裂树分裂时若已经计算出一个节点后, 另一节点直接用父节点和兄弟节点直方图做差得到, 这样显著降低了存储和计算的成本。

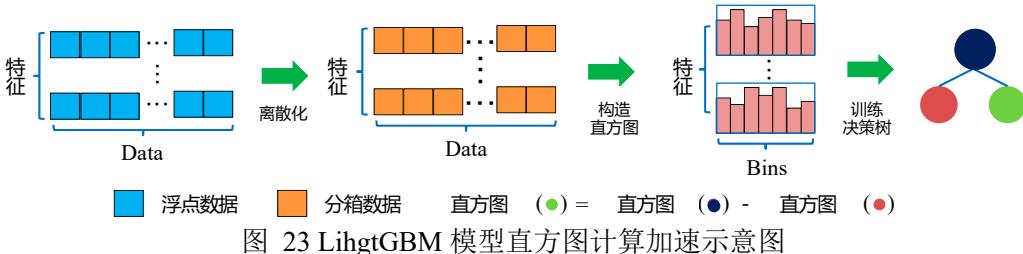


图 23 LightGBM 模型直方图计算加速示意图

### (3) HTS 模型校准方法与 LightGBM 结合

类似地 Prophet 的第一种方法<sup>4</sup>, 对模型的校准采用 Middle-Out: 对预测出 9147 条 L11 层的一个商品在一个州内各店总销量后, 再使用  $\frac{1}{180} \sum_{t=n-180}^n \frac{Y_{L11,t}}{y_t}, j = 1, \dots, 3$ (WI 和 TX)或 4(CA)确定权重  $p_j$  向下分解得到 L12, 再对 L1~L9 进行 Bottom-Up 加总以满足加总一致性约束。

## 四、实证结果与分析

### (一) 模型评价指标

#### 1. 单序列的点估计指标

首先, 对于验证集和测试集  $h = 28$  天的 42840 条时间序列, 先分别计算预测值  $\hat{y}_t$  与真实值  $y_t (t = 1, 2, \dots, n)$  的差距, 这个差距使用均方根标准预测误差 (Root Mean Squared Scaled Error, RMSSE) 来衡量:

$$RMSSE = \sqrt{\frac{1}{h} \frac{\sum_{t=n+1}^{n+h} (y_t - \hat{y}_t)^2}{\frac{1}{n-1} \sum_{t=2}^n (y_t - y_{t-1})^2}}$$

使用 RMSSE 来对销售型分层时间序列评估的原因有几个。首先, 与均方根误差  $RMSE = \sqrt{\frac{\sum_{t=1}^h (y_t - \hat{y}_t)^2}{h}}$  相比, 由于不同商品的日常销量的均值和波动性本质上就有区别 RMSE 并不能很好地衡量, 而 RMSSE 分母则相当于商品过去  $n$  日的每日变化量的均方根平方和, 或是可以认为是“用今天预测明天”这一策略在历史数据中的 RMSE 指标。而 RMSSE 比 RMSE 多的这个分母让不同量级的商品销量预测准确率之间更加可比。第

<sup>4</sup> 注意到, Prophet 方案使用了自中向外和最优结合但本文 LightGBM 方案只使用了前者, 原因在于: 每条序列在 1913 的历史数据中就会生成  $1913-28=1885$  个数据, 而 LightGBM+最优结合需要对 12 层中的每个序列训练 12 个 LightGBM 模型并调整参数, 笔者没有足够的计算资源进行, 但该方法在原理上可行。

二，跟准确率  $\sqrt{\frac{\sum_{t=1}^h (\hat{y}_t/y_t)^2}{h}}$  等指标相比，RMSSE 的计算无需考虑分母为 0（真实预测为 0）的情况，稳定性较高；第三，RMSSE 指标对过大或过小预测的惩罚是对称的。

## 2. 所有序列的加权平均

对每一层的 42840 个序列，使用其最后 28 天的总销售金额（预测出来的销售量乘以事前确定的价格）归一化后作为权重，再对 RMSSE 加权得到 WRMMSE。

$$WRMSSE = \sum_{i=1}^{42,840} w_i * RMSSE$$

这样加权的原因主要有两个。第一，42840 条序列在不同层级中，越上层的序列预测量级越大，因而出现大的偏差，得到的惩罚也应该更大。第二，对 30790 条基时间序列，对一件商品预测出现的偏差，会对上面十一层序列的评价指标都有负面影响，更符合实际情况（对基层销售者和顶层的供应管理者都不利）。第三，该指标对整体偏差的模型施加更大的惩罚。假设 A 模型对 30790 条序列预测整体偏小 1 件，而 B 模型则一半偏小 1 一半偏大 1，那么在加总层面中 A 模型因为整体低估了销量分数会比正负一定程度抵消的 B 模型更差，B 模型因对上层的供应管理者而言影响小因此分数更高合理。

## (二) 模型实证结果

### 1. 模型拟合结果

#### (1) LightGBM

在 L11 层的 LightGBM 模型训练完成后，测算出的重要性里商品、商品价格、星期对模型影响最大，此外滑窗和滞后的特征重要性也比较高，而 7 天滞后特征在其中占比最大也符合探索性数据分析中发现的结论。

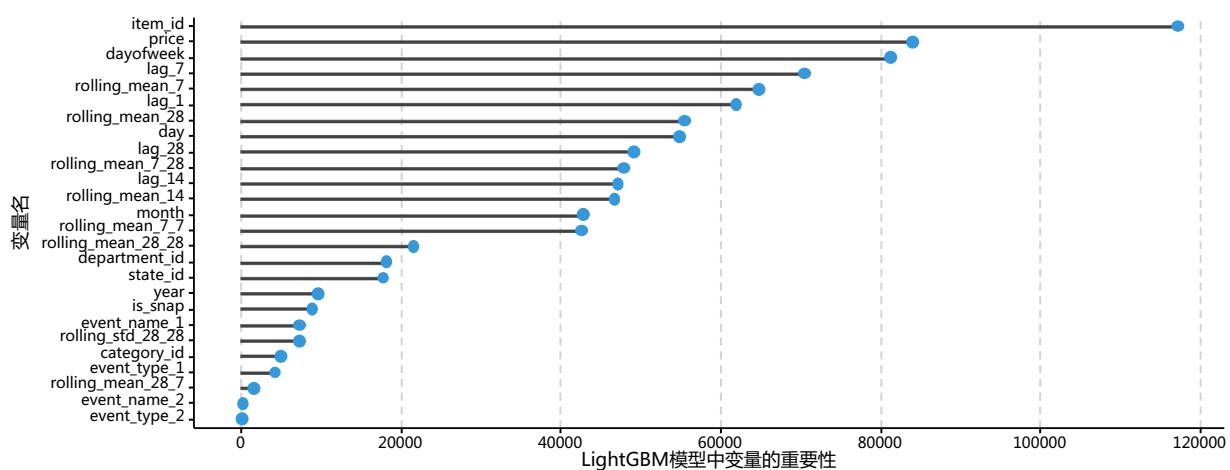


图 24 LightGBM 模型中各特征的重要性图

## (2) Prophet

以顶层的 Prophet 模型拟合结果（图 25）为例，该模型较好地拟合了整体的趋势，从最后 180 天来看，大部分数据落于置信区间中且对未来的预测良好地延续了月度趋势。

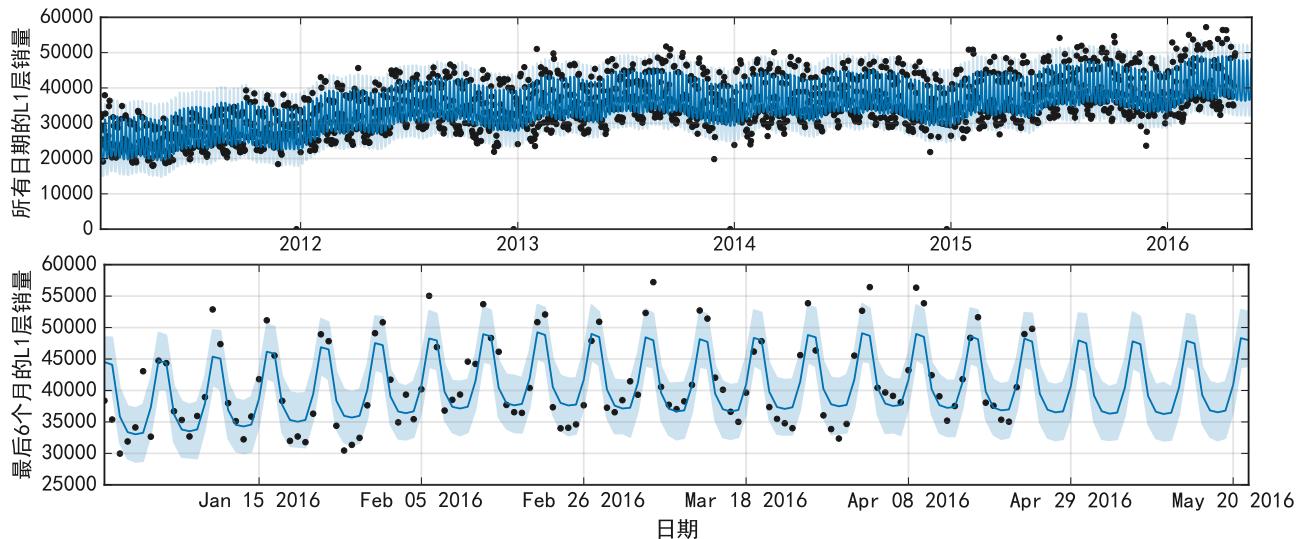


图 25 Prophet-L1 模型拟合与预测结果（上图为完整时序、下图为最后 6 个月时序）<sup>5</sup>

而从模型的分解图 26 来看，趋势项刻画了 Walmart 销量先上涨、后持平再上涨的过程，周频季节性刻画了周末高、周中低的特点，而年频则良好体现了夏天旺季、秋冬圣诞节前后淡季的特征。

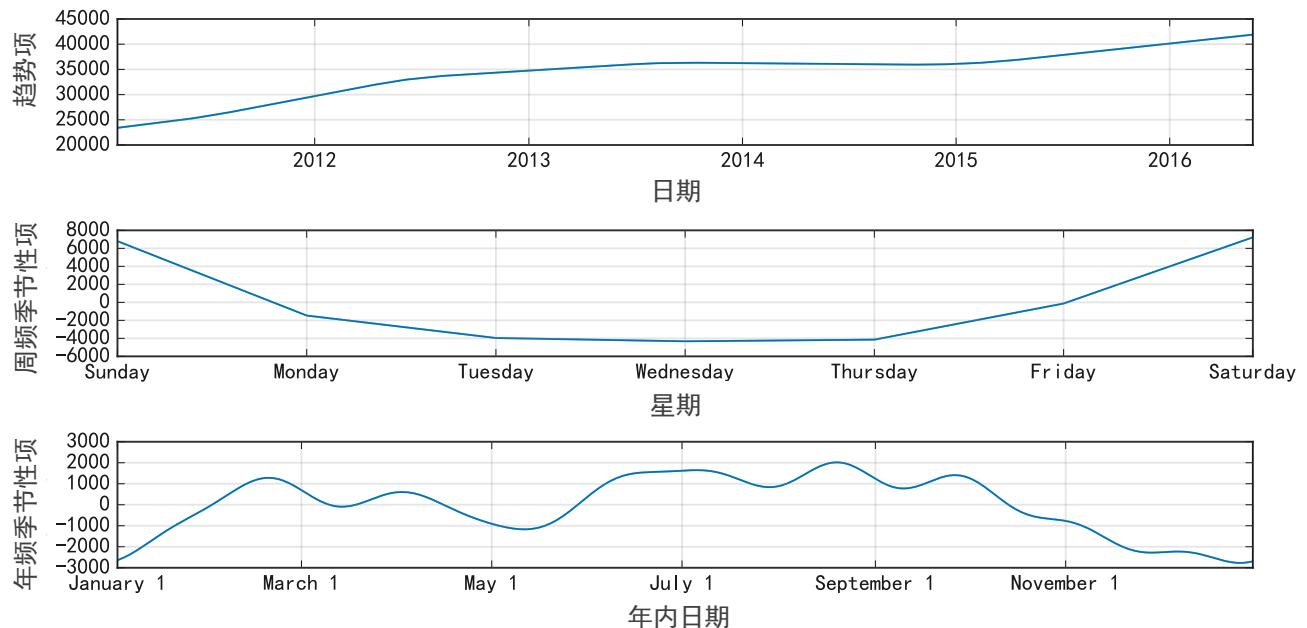


图 26 Prophet-L1 模型对原始序列的分解

<sup>5</sup> 黑色点表示真实数据，蓝色线表示拟合值，浅蓝色区间代表上下 95% 置信区间内值的范围

## 2. 模型指标评价

将用于实验的 LightGBM 方案和两个 Prophet 方案在十二层的验证集和测试集的 RMSSE 和 WRMSSE 预测表现结果如下：

表 6 模型验证集和测试集各层 RMSSE 结果与 WRMSSE 结果统计表

评价指标	验证集指标												测试集指标			
	WRMSSE	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	WRMSSE	排名	
对照模型	ARIMA 自底向上	0.961	0.911	0.926	0.935	0.918	0.975	0.949	0.994	0.948	0.988	1.068	0.997	0.928	0.908	2095
	ARIMA 自顶向下	0.847	0.829	0.673	0.753	0.656	0.768	0.725	0.810	0.785	0.856	1.027	0.969	0.910	0.796	1182
	ARIMA Optimal 校准	0.745	0.697	0.592	0.693	0.571	0.688	0.661	0.755	0.738	0.819	1.026	0.970	0.912	0.691	520
	LightGBM 单模型 <sup>6</sup>	0.786	0.743	0.733	0.733	0.732	0.760	0.739	0.762	0.775	0.800	0.895	0.872	0.853	0.829	1912
实验模型	LightGBM 自中向外	0.542	0.345	0.396	0.396	0.369	0.419	0.440	0.497	0.524	0.596	0.808	0.819	0.825	0.644	279
	Prophet 自中向外	0.694	0.423	0.466	0.466	0.486	0.550	0.540	0.632	0.682	0.755	1.080	1.061	1.040	0.615	154
	Prophet Optimal 校准	0.634	0.483	0.495	0.495	0.521	0.591	0.543	0.615	0.601	0.680	0.848	0.845	0.843	0.579	57

在图 27 对验证集 RMSSE 的分解中：首先发现对照模型中效果最差的是自顶向上的 ARIMA，是因为底层的序列噪声过大且 ARIMA 没有充分利用季节性原因导致、ARIMA 自顶向下则需要直接从最顶层出发需要预测下层 30490 个序列的比例表现也较差，而最优结合法对其有显著的改进。第二，采用自中向外法的 Prophet 在底层意外出现了很差的预测，与模型开始的层数过高有关，相比之下从 L9 出发的 LightGBM 底层得分更低。第三，对比 Prophet 的最优结合校准与 Prophet 自中向外，可以看到在调整了每一层的预测之后虽然对 L1-L5 的分层结果变差，但底层的预测结果显著改善因此 WRMSSE 得分提高。

而在图 28 测试集结果的比较中可以看到 LightGBM 模型在测试集的表现会低于验证集，而 ARIMA 三个基准模型的表现都变好了，Prophet 的表现则更为稳定，其中 Prophet 最优结合模型更是与全球 5558 支提交预测结果的队伍中排到了 57 名（约 1%）。

## 3. 预测结果

图 29 展示三个模型在 L1 层的预测结果，可以看到 L1 层的季节性三个模型拟合结果良好，对样本外的真实曲线贴合良好，并对未来的预测重复体现了前文讨论的周频季节性。

<sup>6</sup> 单模型 LightGBM 不使用任何 HTS 模型校准方法，而只是对所有层数据不做区别地提取特征后输入模型，而从与使用 Middle-Out 方法的 LightGBM 形成实验对照组。

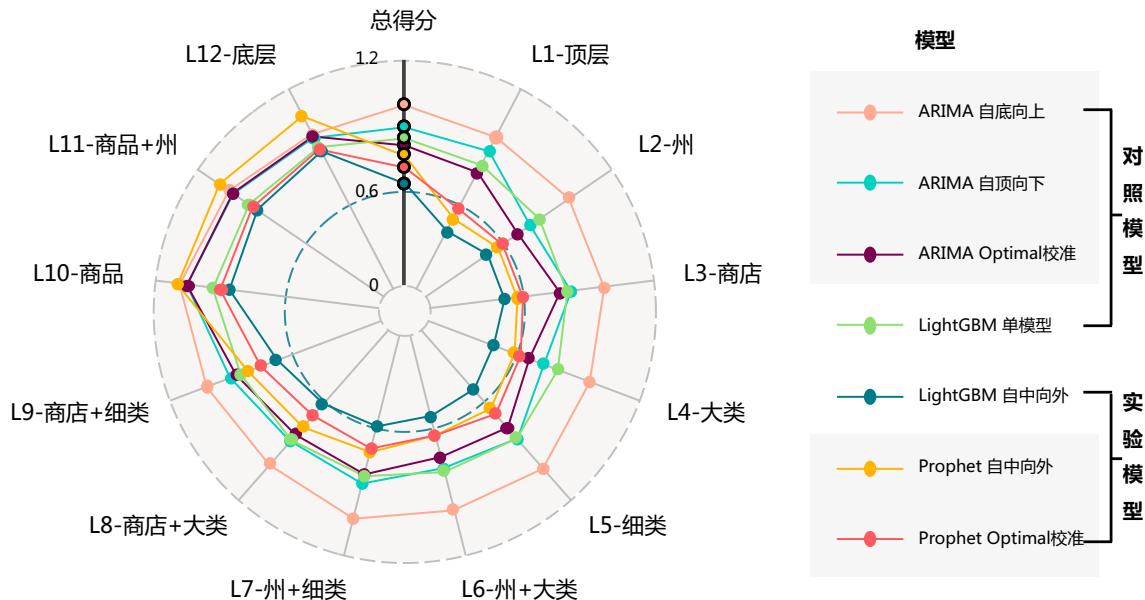
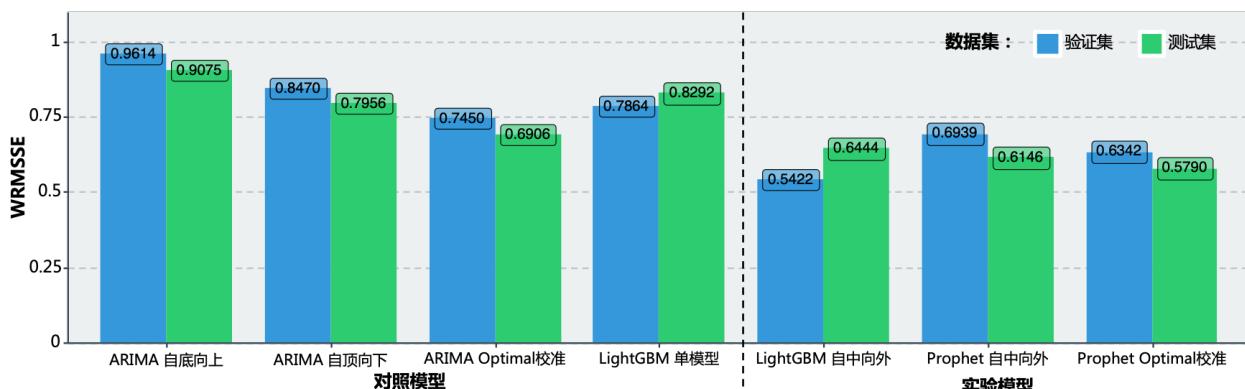

 图 27 模型在验证集中 RMSSE 指标雷达图<sup>7</sup>


图 28 模型在验证集和测试集测算的 WRMSSE 指标

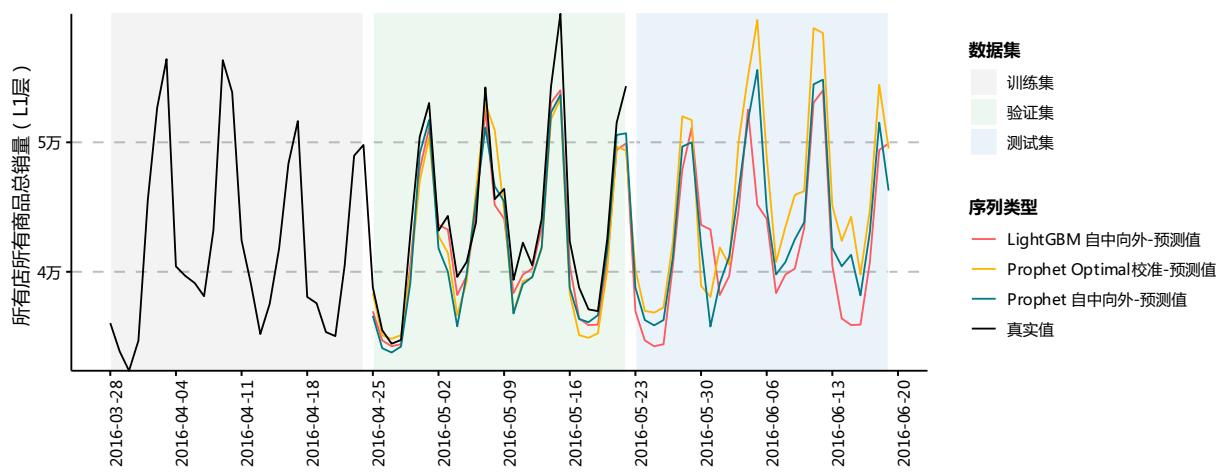


图 29 实验模型在不同数据集中的预测结果

<sup>7</sup> 衡量指标为 loss 型的 WRMSSE，因此越贴近圆心模型表现越好

## 五、结论与展望

### (一) 本文结论

本文讨论了时间序列领域近年新兴的分层时间序列模型校准技术并将其应用于 LightGBM 和 prophet 模型中给出了较有工业应用价值的商品销量预测方案。

总体来看，采用 fbprophet 和 LightGBM 模型与分层时间序列模型校准技术可以从三个方面提升大规模分层时间序列预测的准确性。一方面来自于底层模型对趋势、季节性、离群值和噪声等复杂模式下更好的拟合，也有效整合了节日、商品价格等外生变量；另一方面来自于 HTS 技术对模型结果进行的针对性校准，使其在各层中最小化了加总谬误问题，即独立预测向上层加总时不等于上层序列的预测值。最后，两个模型也充分控制了底层序列噪音带来的影响。

### (二) 本文创新性

学术方面，本文提供了第一篇介绍 HTS 的中文文献、第一篇将分层时间序列模型校准方法与 LightGBM 及 Prophet 结合的文献；应用方面，本文提出的 HTS 校准与前沿时间序列预测方法的结合，可为大规模分层时间序列预测的智能零售场景提供解决方案。

### (三) 不足之处与展望

在本文针对 Walmart 商品销量数据所建立的模型上受到了一定的计算预算限制，模型还有多个可能的改进空间：

- 不同层的模型可以各自网格搜索最优的参数，同时如第二部分 1.2 小节，可以针对不同层的模型采用特殊的数据清洗方案来达到模型的最佳；
- 可以采用滚动 K 折的方法逐月囊括新数据并重新训练模型来计算 WRMSSE 指标，这样的交叉验证更能使结果贴合实际使用场景
- 可以选取接受季节性和外生变量的 SARIMAX 模型作为基准模型（Benchmark）来代替 ARIMA，从而更好地反应底层模型改进对于 HTS 问题的帮助
- 可以通过神经网络对多个模型进行融合，使预测的不稳定性被平滑。

在应用场景中，两个方案在工业应用于大规模数据中具有可行性。笔者使用的 NVIDIA GeForce RTX 2080 Ti GPU 服务器（CPU 型号：Intel (R) Xeon (R) CPU E5-2678 v3 @ 2.50GHz；内核数：6 cores；内存：62 GB）对本案例七千万行（约 4.7GB）的时间序列数据的模型训练成本可行，LightGBM+Middle-Out、Prophet+Middle-Out、Prophet+Optimal 的完整训练分别约需 56、28、40 小时。实践中可装载更多的服务器周期训练模型，落地更优良的分层时间序列预测方案，推动智慧零售的数字化转型进程。

## 参考文献

- [1] Hyndman R J, Lee A J, Wang E. Fast computation of reconciled forecasts for hierarchical and grouped time series[J]. Computational Statistics & Data Analysis, 2016, 97:16-32.
- [2] Wickramasuriya S L, Athanasopoulos G, Hyndman R J. Forecasting hierarchical and grouped time series through trace minimization[J]. Monash Econometrics & Business Statistics Working Papers, 2015.
- [3] Liu, Zitao, Yan Yan, and Milos Hauskrecht. A flexible forecasting framework for hierarchical time series with seasonal patterns: A case study of web traffic[J]. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018.
- [4] Elsalamony, H. A. Bank direct marketing analysis of data mining techniques[J]. International Journal of Computer Applications, 2014. 85(7), 12-22.
- [5] Almeida, Vânia, Rita Ribeiro, and João Gama. Hierarchical time series forecast in electrical grids[J]. Information Science and Applications (ICISA) 2016. Springer, Singapore, 2016. 995-1005.
- [6] Panamtash H, Zhou Q. Coherent Probabilistic Solar Power Forecasting[C]. IEEE International Conference on Probabilistic Methods Applied to Power Systems, 2020.
- [7] Abouaraghoub W, Nomikos N K, Petropoulos F. On reconciling macro and micro energy transport forecasts for strategic decision making in the tanker industry[J]. Transportation Research Part E: Logistics and Transportation Review, 2018.
- [8] Rydlewski J, Kosiorowski D, Mielczarek D. Forecasting of a Hierarchical Functional Time Series on Example of Macromodel for Day and Night Air Pollution in Silesia Region: A Critical Overview[J]. Papers, 2017.
- [9] Harsoor A S, Patil A. Forecast of Sales of Walmart Store Using Big Data Applications[J]. International Journal of Research in Engineering and Technology, 2015 (04): 51-59.
- [10] Hyndman R J, Lee A J, Wang E. Fast computation of reconciled forecasts for hierarchical and grouped time series[J]. Computational Statistics & Data Analysis, 2016 (97):16-32.
- [11] 段江娇, 薛永生, 林子雨, et al. 一种新的基于隐 Markov 模型的分层时间序列聚类算法[J]. 计算机研究与发展, 2006, 43(1):61-67.
- [12] Dalrymple, Douglas J. Sales forecasting practices: Results from a United States survey[J].

- International journal of Forecasting 3.3-4 (1987): 379-391.
- [13] Gross, Charles W., and Jeffrey E. Sohl. Disaggregation methods to expedite product line forecasting[J]. Journal of forecasting 9.3 (1990): 233-254.
- [14] Dangerfield B J, Morris J S. Top-down or bottom-up: Aggregate versus disaggregate extrapolations[J]. International Journal of Forecasting, 1992, 8(2):233-241.
- [15] Arnold, Zellner, and, et al. A note on aggregation, disaggregation and forecasting performance[J]. Journal of Forecasting, 2000.
- [16] Kinney, William R. Predicting earnings: entity versus subentity data[J]. Journal of Accounting Research, 1971: 127-136.
- [17] Dunn, Douglas M., William H. Williams, and T. L. DeChaine. Aggregate versus subaggregate models in local area forecasting[J]. Journal of the American Statistical Association, 1976: 68-71.
- [18] Fisher W D. A Note on Aggregation and Disaggregation[C]. Northwestern University, Center for Mathematical Studies in Economics and Management Science, 1977.
- [19] Hyndman R J, Ahmed R A, Athanasopoulos G, et al. Optimal combination forecasts for hierarchical time series[J]. Computational Statistics & Data Analysis, 2011, 55(9):2579-2589.
- [20] Wickramasuriya, Shanika L., George Athanasopoulos, and Rob J. Hyndman. Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization[J]. Journal of the American Statistical Association, 2019: 804-819.
- [21] Silveira Gontijo, Tiago, and Marcelo Azevedo Costa. Forecasting hierarchical time series in power generation[J], 2020: 3722.
- [22] 康孟海, 于建军. 基于 fbprophet 框架的期末余额预测方法[J]. 科研信息化技术与应用: 中英文, 2019, 010(003):P.13-20.
- [23] 马跃祎. 基于 Prophet 模型的原木与锯材市场风险预警分析[J]. 农业与技术, 2020, v.40;No.364(23):170-173.
- [24] Navratil M, Kolkova A. Decomposition and Forecasting Time Series in the Business Economy Using Prophet Forecasting Model[J]. Central European Business Review, 2019.
- [25] Kumar, Naresh, and Seba Susan. COVID-19 Pandemic Prediction using Time Series Forecasting Models[C]. 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2020.

- [26] Battineni G, Chintalapudi N, F Amenta. Forecasting of COVID-19 epidemic size in four high hitting nations (USA, Brazil, India and Russia) by Fb-Prophet machine learning model[J]. Applied Computing and Informatics, 2020, 6(1):1-10.
- [27] Ke, Guolin, et al. "Lightgbm: A highly efficient gradient boosting decision tree." Advances in neural information processing systems 30 (2017): 3146-3154.
- [28] Bojer, Casper Solheim, and Jens Peder Meldgaard. Kaggle forecasting competitions: An overlooked learning opportunity[J]. International Journal of Forecasting 37.2 (2021): 587-603.
- [29] 许国艳, 周星熠, 司存友,等. 基于GRU和LightGBM特征选择的水位时间序列预测模型[J]. 计算机应用与软件, 2020, 037(002):25-31,53.
- [30] Cao Y, Gui L. Multi-Step wind power forecasting model Using LSTM networks, Similar Time Series and LightGBM[C]. 2018 5th International Conference on Systems and Informatics (ICSAI), 2018.
- [31] Weng T, Liu W, Xiao J. Supply chain sales forecasting based on lightGBM and LSTM combination model[J]. Industrial Management & Data Systems, 2019.

### 中央财经大学本科毕业论文（设计）原创性声明

本人郑重声明：所提交的毕业论文（设计）《分层时间序列方法在商品销量预测中的应用》，是本人在指导老师的指导下独立进行研究工作所取得的成果。除文中已经注明引用的内容外，不含任何其他个人或集体已经发表或撰写过的作品成果，不存在购买、由他人代写、剽窃和伪造数据等作假行为。对本文研究/设计做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果，如违反有关规定或上述声明，愿意承担由此产生的一切后果。

作者签名: 

2021 年 5 月 10 日

## 致谢

大学四年的时光一晃而过，转眼间我已从初到学校遐想未来的十八少年到了毕业论文落笔之时。此时此刻，我如释重负，回首过去，我却又感慨万千。

首先，我希望诚挚地感谢我的论文指导老师许欣怡老师。她在忙碌的教学工作中挤出时间来跟进、审阅、修改我的论文并提出了非常多的建设性建议。

我院李丰老师、王思洋老师、马景义老师、杨钥含老师、刘苗老师、杨欣欣老师和加州大学伯克利分校的 Jared Fisher 教授，你们设计了相当精彩的核心专业课课程。在课堂中你们循循善诱的教导下我逐渐沉淀了扎实的统计与数据科学专业知识和技能，我毕业论文的完成离不开你们帮助我打下的学科基础。

还有所有教过我学科基础课程的老师们和学院的其他老师们，是你们严谨细致、一丝不苟的作风成为了同学们工作、学习的榜样，成就了统数学院优良的学习氛围。同时也要感谢四年中陪伴在我身边的舍友、同学、朋友们，有了你们的支持、鼓励和帮助，我才能不留遗憾，在转专业、北大经济双学位、伯克利交换四跨专业的迷茫后坚定了自己的方向，最后顺利录取哥伦比亚大学数据科学专业继续追逐梦想的道路。

另外，我还要感谢总能在我失意的时候接起电话倾听、陪伴我成长的含辛茹苦的父母，谢谢你们！

最后，再次对关心、帮助我的老师和同学表示衷心的感谢！

杨谨行

2021 年 5 月 10 日