

# 大规模时间序列的分层预测



光华管理学院

Guanghua School of Management

**Feng Li**

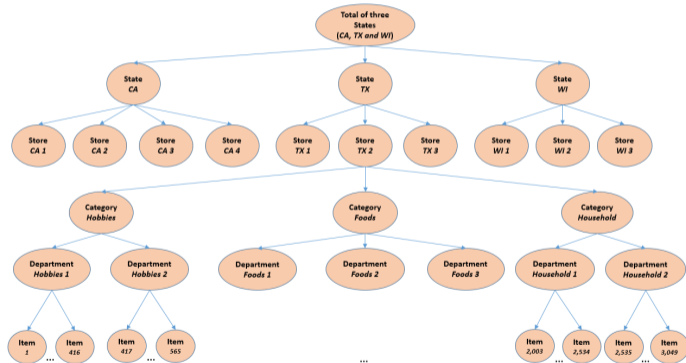
[feng.li@gsm.pku.edu.cn](mailto:feng.li@gsm.pku.edu.cn)

<https://feng.li/>



- 1 基于自上而下对齐方法的层级时间序列预测
- 2 Walmart M5 竞赛的第二名解决方案
- 3 带有不可变约束的预测调和
- 4 中国电商数据的应用

- 层级时间序列是一类特殊的网络型数据。
- 在层级结构中，各层级的预测结果应当与其聚合结构保持一致性（coherence）。





- 多变量时间序列模型（如 VAR）难以直接应用，因为底层序列通常噪声较大或呈间歇性。
- Copula 模型难以扩展到大规模层级。
- 纯深度学习模型往往无法满足层级一致性约束（**coherent constant**）。



## 层级时间序列预测

### ↪ 如何实现预测一致性?

- 在所有节点上生成基础预测 (base forecasts)。
- 自下而上 (**Bottom Up**) 方法 (Schwarzkopf et al., 1988) 忽略了上层节点, 因为底层噪声会在聚合时传播到高层。
- 自上而下 (**Top Down**) 方法 (Gross & Sohl, 1990) 忽略底层节点, 丢失了部分信息, 即使基础预测无偏, 也可能引入系统性偏差。



- 层级底部通常表现出明显的间歇性模式。
- 层级上部包含可预测的成分，如趋势或季节性，这些成分来源于底层序列的聚合。
- 典型业务场景：
  - 总部管理者或投资者通常并不关心某个具体门店或产品的预测，而更关注上层层级（如总产品线）的预测结果，因为它直接影响总体营收 (Anderer & Li, 2022, IJF)。
  - 政府或机构希望通过保持若干上层地区（如六个省份）的预测不变，从而稳定整体经济表现 (Zhang, Kang, Panagiotelis & Li, 2023, EJOR)。



## 自上而下对齐方法

### ↳ 改进的 N-BEATS 集成模型用于上层预测

- N-BEATS 模型 (Oreshkin et al., 2020) 是一种纯深度学习方法，其神经网络架构基于前向与反向残差连接。
- N-BEATS 模型的性能依赖于集成 (ensembling) 步骤，即通过预测组合技术整合不同模型的预测结果。
- 学习精度对学习率十分敏感，且收敛速度较慢。为此，我们将 **Lookahead** 优化器 (Zhang et al., 2019) 引入 N-BEATS 训练器，以进一步提升训练精度与稳定性。



## 自上而下对齐方法

### ↪ 用于底层的偏差校正 LightGBM 模型

- 在层级底部，我们采用偏差校正的 **LightGBM** 模型来刻画间歇性时间序列。
- LightGBM 使用均方根误差 (RMSE) 作为损失函数，定义如下：

$$RMSE = \sqrt{\frac{1}{h} \sum_{t=n+1}^{n+h} (Y_t - \hat{Y}_t)^2}.$$

- 我们进一步定义 RMSE 的自定义梯度如下：

$$\text{gradient} = \begin{cases} -2e_t & e_t < 0, \\ -2\lambda e_t & e_t \geq 0 \end{cases}$$

其中  $\lambda > 0$  为调节参数，用于引入非对称损失函数。



## 自上而下对齐方法

### ↪ 对齐上层与聚合的底层预测结果

- 给定来自稳定 N-BEATS 模型的上层独立预测结果，最后一步是确定由 LightGBM 生成的最优底层预测。
- 我们调整 LightGBM 模型中的损失系数  $\lambda$ ，使得：
- 优化目标为通过层级对齐最小化上层预测与聚合后的底层预测之间的 RMSE：

$$\arg \min_{\lambda} \left\{ \sqrt{\frac{1}{h} \sum_{t=n+1}^{n+h} \left( \hat{Y}_t^{(\text{top})} - \text{Agg}(\hat{Y}_t^{(\text{bottom})}(\lambda)) \right)^2} \right\},$$

其中  $\text{Agg}(\cdot)$  表示用于底层聚合的算子。



## 自上而下对齐方法

### ↳ M5 竞赛数据的层级结构

- 该数据集包含了层级结构的每日销售数据，总计 42,840 条时间序列，覆盖 1,941 天。

层级级别	描述	序列数量
1	所有产品、所有商店、所有州	1
2	各州的所有产品	3
3	各商店的所有产品	10
4	各产品类别的总销售	3
5	各部门的总销售	7
6	各州与类别汇总的所有产品销售	9
7	各州与部门汇总的所有产品销售	21
8	各商店与类别汇总的所有产品销售	30
9	各商店与部门汇总的所有产品销售	70
10	产品 $x$ 的总销售（跨所有商店/州）	3,049
11	产品 $x$ 在各州的销售	9,147
12	产品 $x$ 在各商店的销售	30,490
总计		42,840



# 自上而下对齐方法

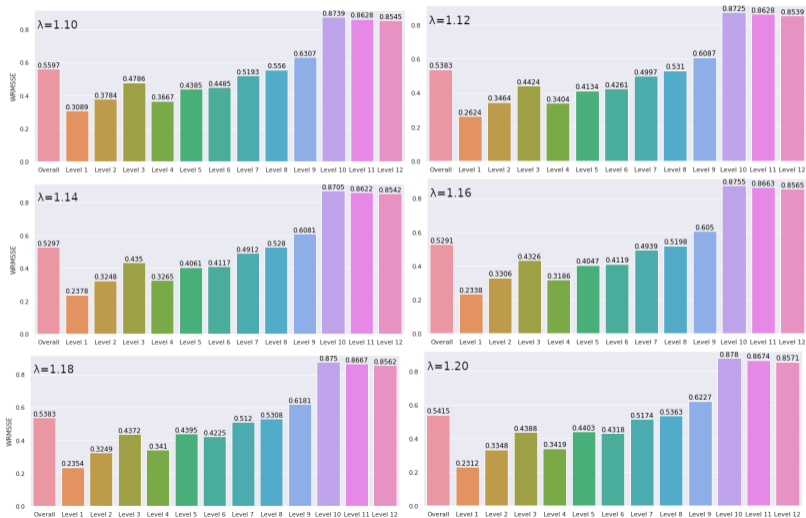
## ↳ 用于底层间歇性时间序列的特征

特征名称	描述
sell_price	商品在给定日期的销售价格。
event_type	108 类事件类型，例如体育、文化、宗教活动等。
event_name	对应 event_type 的 157 个事件名称，例如超级碗、情人节、总统日等。
event_name_2	竞赛数据中提供的第二事件名称特征。
event_type_2	竞赛数据中提供的第二事件类型特征。
snap_CA, TX, WI	加利福尼亚、德克萨斯和威斯康星州的 SNAP（二元补助信息）指示变量。
release	商品在商店中的上架周次。
price_max, min	训练数据中商品在商店的最高价与最低价。
price_mean, std, norm	训练数据中商品在商店的价格均值、标准差与归一化价格。
item, price_nunique	商店中唯一商品数与唯一价格数。
price_diff_w	商品在商店的周度价格变化。
price_diff_m	商品在商店的价格相对于月均值的变化。
price_diff_y	商品在商店的价格相对于年均值的变化。
tm_d	月中的日期（天）。
tm_w	一年中的周次。
tm_m	一年中的月份。
tm_y	训练数据中的年份索引。
tm_wm	一月中的周次。
tm_dw	一周中的星期几。
tm_w_end	周末指示变量。



# 自上而下对齐方法

## ↪ 预测误差与偏差乘子分析

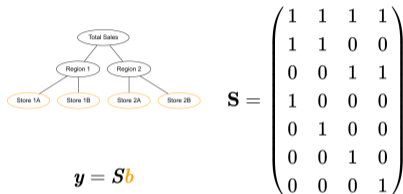




- 本研究聚焦于点预测问题。
- N-BEATS 依赖高效的集成方法，而 LightGBM 则需要合理的特征工程。
- 使用本框架进行概率预测是可行的，只需替换为相应的概率型损失函数。
- 欢迎试用我们的 Python 实现：  
[github.com/matthiasanderer/m5-accuracy-competition](https://github.com/matthiasanderer/m5-accuracy-competition)

## 什么是 Reconciliation?

- 我们可以通过包含 0 和 1 的求和矩阵 ( $S$ ) 来构建一致性约束:



- 若将基础预测堆叠为一个  $n$  维向量  $\hat{y}$ , 则调和后的预测  $\tilde{y}$  表示为:

$$\tilde{y} = SG\hat{y}$$

- 不同的  $G$  选择会产生不同的调和方法。例如, MinT 方法通过最小化  $E[(y - \tilde{y})'(y - \tilde{y})]$  来获得最优解 (Wickramasuriya et al., 2019, JASA)。



- 调和的目标是找到  $\tilde{\mathbf{y}}$  以最小化：

$$(\tilde{\mathbf{y}} - \hat{\mathbf{y}})' \mathbf{W} (\tilde{\mathbf{y}} - \hat{\mathbf{y}})$$

- 传统的调和方法要求  $\tilde{\mathbf{y}}$  满足一致性约束 (**Coherence**) 以及非负性约束 (**Non-negativity**)。
- 对于不能出现负值的数据，优化问题需满足  $\tilde{y}_i \geq 0$ ，对所有  $i$  均成立。
- 该问题可通过二次规划 (**Quadratic Programming**) 求解。



- 研究动机：在预测调和后，部分变量的预测精度可能会变差。
- 我们希望保持部分变量的预测结果不可变 (**Immutable**)。
- 我们的研究目标：求解  $\tilde{\mathbf{y}}$  以最小化

$$(\tilde{\mathbf{y}} - \hat{\mathbf{y}})' \mathbf{W} (\tilde{\mathbf{y}} - \hat{\mathbf{y}})$$

其约束条件包括：

- ① 一致性约束 (Coherence constraints)
- ② 不可变约束 (Immutability constraints)
- ③ 非负性约束 (Non-negativity constraints)



- 我们将求和矩阵划分如下:

$$\mathbf{S} = \begin{bmatrix} [I] \mathbf{S}_1 & \mathbf{S}_2 \\ \mathbf{I}_{m-k} & \mathbf{0}_{(m-k) \times k} \\ \mathbf{0}_{k \times (m-k)} & \mathbf{I}_{k \times k} \end{bmatrix},$$

- 我们的方法给出以下优化问题的解:

$$\begin{aligned} \min_{\tilde{\mathbf{y}}} (\hat{\mathbf{y}} - \tilde{\mathbf{y}})' \mathbf{W}^{-1} (\hat{\mathbf{y}} - \tilde{\mathbf{y}}) \quad \text{s.t.} \\ \tilde{\mathbf{u}} = \hat{\mathbf{u}}, \\ \mathbf{S}_1 \tilde{\mathbf{v}} + \mathbf{S}_2 \tilde{\mathbf{u}} = \tilde{\mathbf{w}}, \end{aligned}$$



符号	记号	含义说明
完整层级结构	$y = (w', v', u)'$	全部序列。
不可变部分	$u$	调和后保持不变的序列。
可变部分	$(w', v)'$	调和后会被调整的序列。
基础层 (Basis)	$b = (v', u)'$	用于定义层级的序列，不一定与底层相同。
非基础层 (Non-basis)	$w$	当基础层数值与约束已知时可推导出的序列。



## Theorem

若基础预测向量  $\hat{y}$  无偏，则根据我们方法得到的调和预测  $S\hat{G}\hat{y}$  也是无偏的。

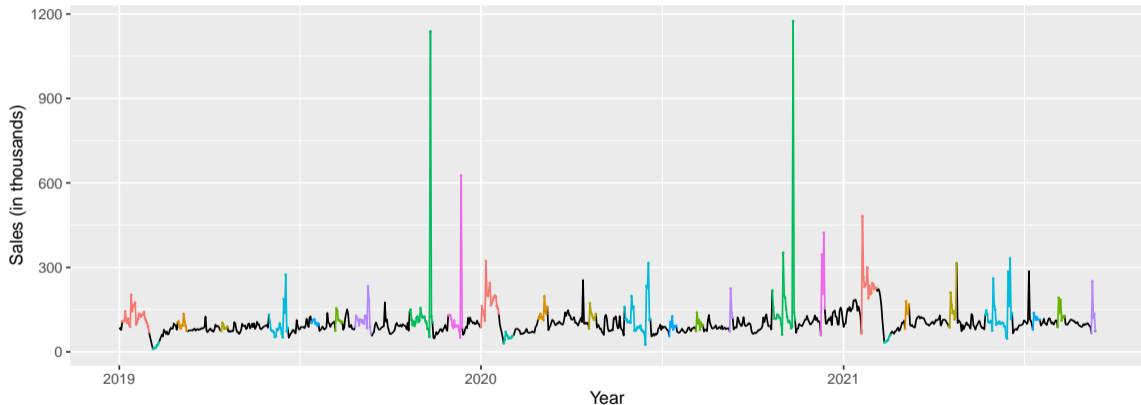
- 基础层中的任意子集都可以设为不可变，但对于给定层级结构，基础层的选择并非任意。

## Theorem

设  $y = Sb$  为层级的任意有效表示（例如  $b$  可为底层序列）。令  $b^* = (y_{j_1}, \dots, y_{j_m})'$  为候选的基础序列集合，其中  $j_1, \dots, j_m$  为  $y$  中的索引。令  $S_{\{j\}}$  为由  $S$  中对应这些索引行组成的方阵。若  $S_{\{j\}}$  可逆，则候选基础序列集合  $b^*$  为有效集合。



- 来自中国某大型电商平台的销售数据;
- 将“食品 (Food)”类销售作为顶层;
- 共有 40 个“中层”类别、1905 个“底层”类别;
- 许多序列表现出间歇性特征, 且促销活动对销售影响显著。





- 对于间歇性序列（零值超过 60%），使用简单指数平滑（SES）模型。
- 其他序列采用带 ARIMA 误差项和 Box-Cox 变换的回归模型（自变量为促销强度）。
- 在顶层层级中，若序列历史较长（超过 1 年），则将其定义为不可变序列（**Immutable Series**）。
- 此外，可以使用交叉验证等程序来自动化选择不可变序列集合。



- “C” 与 “C+NN” 表示带有不可变约束的调和预测，在未加与加上非负约束时的预测精度。
- “U” 与 “U+NN” 表示无约束调和预测的精度，分别对应未加与加上非负约束的情形。
- 对于 “C” 与 “C+NN” 列，顶层预测以及间歇性序列（Bottom-2）与长历史序列（Bottom-3）保持不变；中层与其余底层（Bottom-1）序列可调整。

层级	Base	OLS				WLS <sub>s</sub>				WLS <sub>v</sub>			
		C	C+NN	U	U+NN	C	C+NN	U	U+NN	C	C+NN	U	U+NN
Top	2.94	2.94	2.94	2.93	2.92	2.94	2.94	2.72	2.72	2.94	2.94	2.75	2.77
Middle	2.66	9.31	4.94	272.83	48.84	6.41	4.83	16.09	6.50	2.43	2.47	2.39	2.40
Bottom-1	2.04	8.98	4.31	3.98	2.70	7.19	3.71	2.96	2.32	1.97	1.88	1.86	1.83
Bottom-2	0.11	0.11	0.11	42.66	15.43	0.11	0.11	26.99	8.34	0.11	0.11	1.52	1.52
Bottom-3	1.08	1.08	1.08	1.64	1.48	1.08	1.08	1.36	1.25	1.08	1.08	1.58	1.19










- 施加不可变约束后，预测精度仍优于基础预测。
- 不可变约束在间歇性序列的预测中能有效稳定性能。
- 但不可变约束并不保证所有序列的精度都提高。
- 加入非负性约束通常能进一步提升模型表现。
- 可试用我们的 R 与 Python 实现：
  - R 示例：<https://github.com/AngelPone/chf>
  - Python 层级预测库：<https://github.com/AngelPone/pyhts>



# Thank you!

<https://feng.li>  
[feng.li@gsm.pku.edu.cn](mailto:feng.li@gsm.pku.edu.cn)

-  Zhang, B., Kang, Y., Panagiotelis, A. & Li, F. (2023). “Optimal Reconciliation with Immutable Forecasts”. *European Journal of Operational Research* 308.(1), pp. 650–660.
-  Anderer, M. & Li, F. (2022). “Hierarchical Forecasting with a Top-down Alignment of Independent-Level Forecasts”. *International Journal of Forecasting* 38.(4), pp. 1405–1414.
-  Oreshkin, B. N., Carпов, D., Chapados, N. & Bengio, Y. (2020). “N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting”. In: *Proceedings of The 2020 International Conference on Learning Representations*.
-  Wickramasuriya, S. L., Athanasopoulos, G. & Hyndman, R. J. (2019). “Optimal Forecast Reconciliation for Hierarchical and Grouped Time Series through Trace Minimization”. *Journal of the American Statistical Association* 114.(526), pp. 804–819.
-  Zhang, M., Lucas, J., Ba, J. & Hinton, G. E. (2019). “Lookahead Optimizer: k Steps Forward, 1 Step Back”. In: *Advances in Neural Information Processing Systems*. Vol. 32.
-  Gross, C. W. & Sohl, J. E. (1990). “Disaggregation Methods to Expedite Product Line Forecasting”. *Journal of Forecasting* 9.(3), pp. 233–254.
-  Schwarzkopf, A. B., Tersine, R. J. & Morris, J. S. (1988). “Top-down versus Bottom-up Forecasting Strategies”. *The International Journal Of Production Research* 26.(11), pp. 1833–1843.