



東南大學
SOUTHEAST UNIVERSITY

全流程的供应链需求自动预测与可信 调控

汇报人：刘志远教授、顾子渊副研究员、周臻
东南大学 交通学院





汇报大纲



一、自动预测技术



二、调控数据增强



三、可信调控

□ 三个核心技术创新点

- 自动预测技术
- 面向调控的数据增强
- 稀疏场景下的可信调控

□ 多个技术要点

- 模型可解释性落地化，可被一线工作人员理解
- 双层stacking，集成模型推荐与可解释性
- 基于增强数据改进损失函数，迁移先验调控约束
- 调控结果收集，建立匹配机制与供给需求弹性

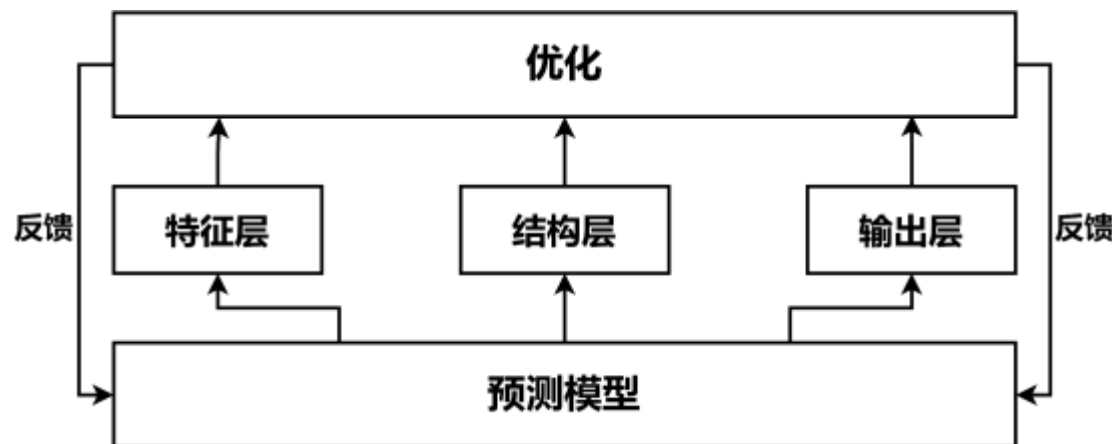
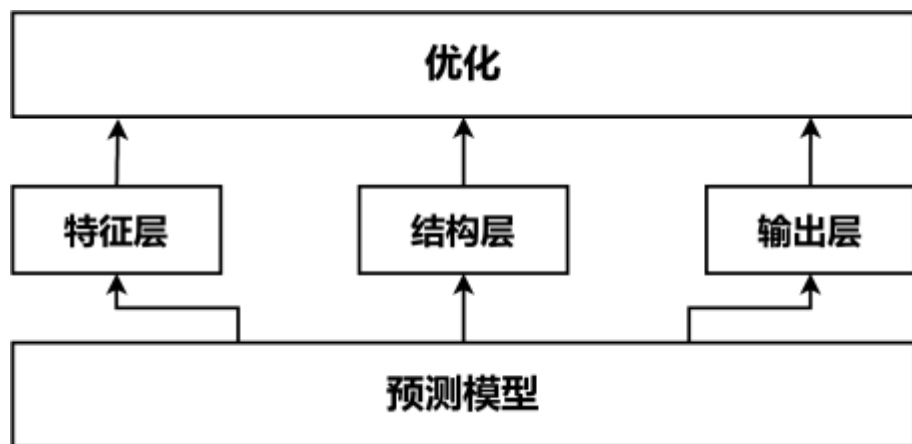


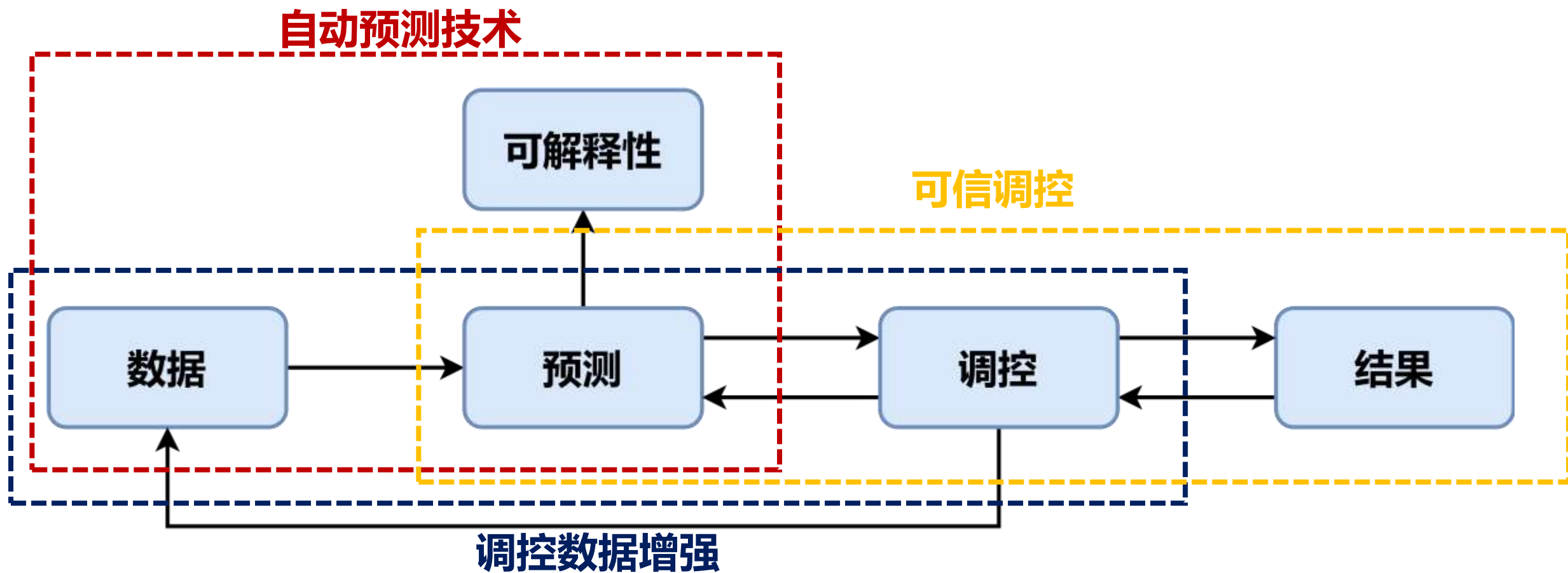
□ Predict Then Optimize

- 预测与优化调控串行连接
- 预测模型的特征层、结构层与输出层为优化调控提供业务因子的接口

□ Predict Joint Optimize

- 预测与优化调控一体化，对优化调控进行评价
- 业务部门上下游打通（当期分析与回溯分析）





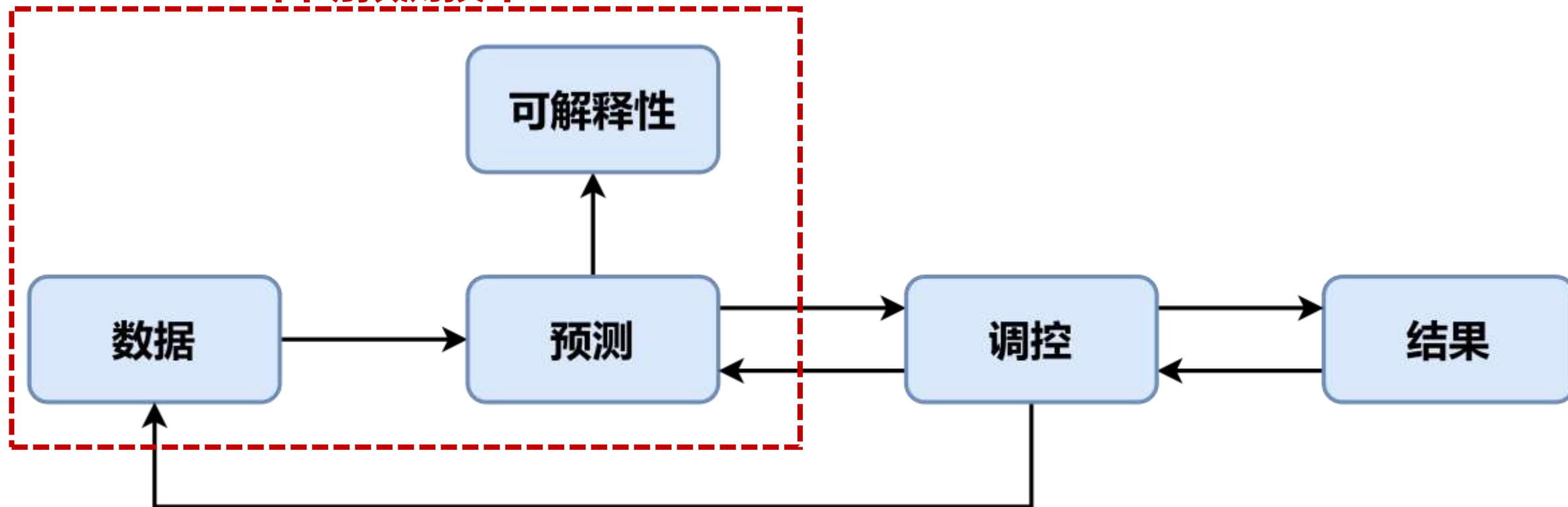
一、自动预测技术



東南大學
SOUTHEAST UNIVERSITY



自动预测技术



□ 自动预测技术

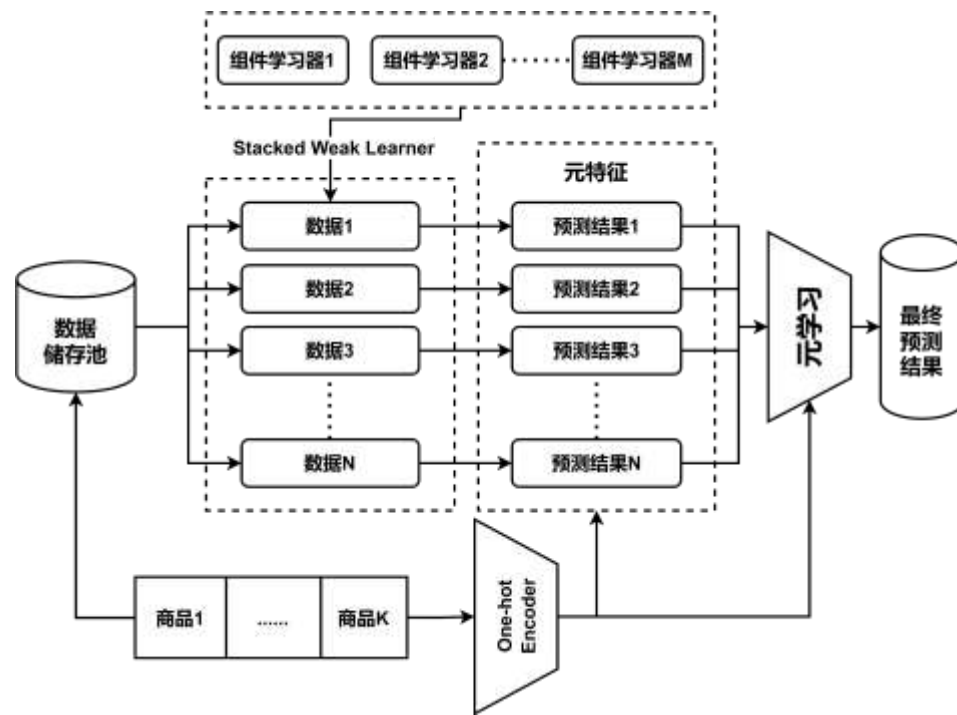
■ 自动预测与可解释技术

- 预测计算流程实现自动构建和自动执行
- 保证高精度的同时，对所有的输入特征，能够解析出对预测结果的影响与模型推荐

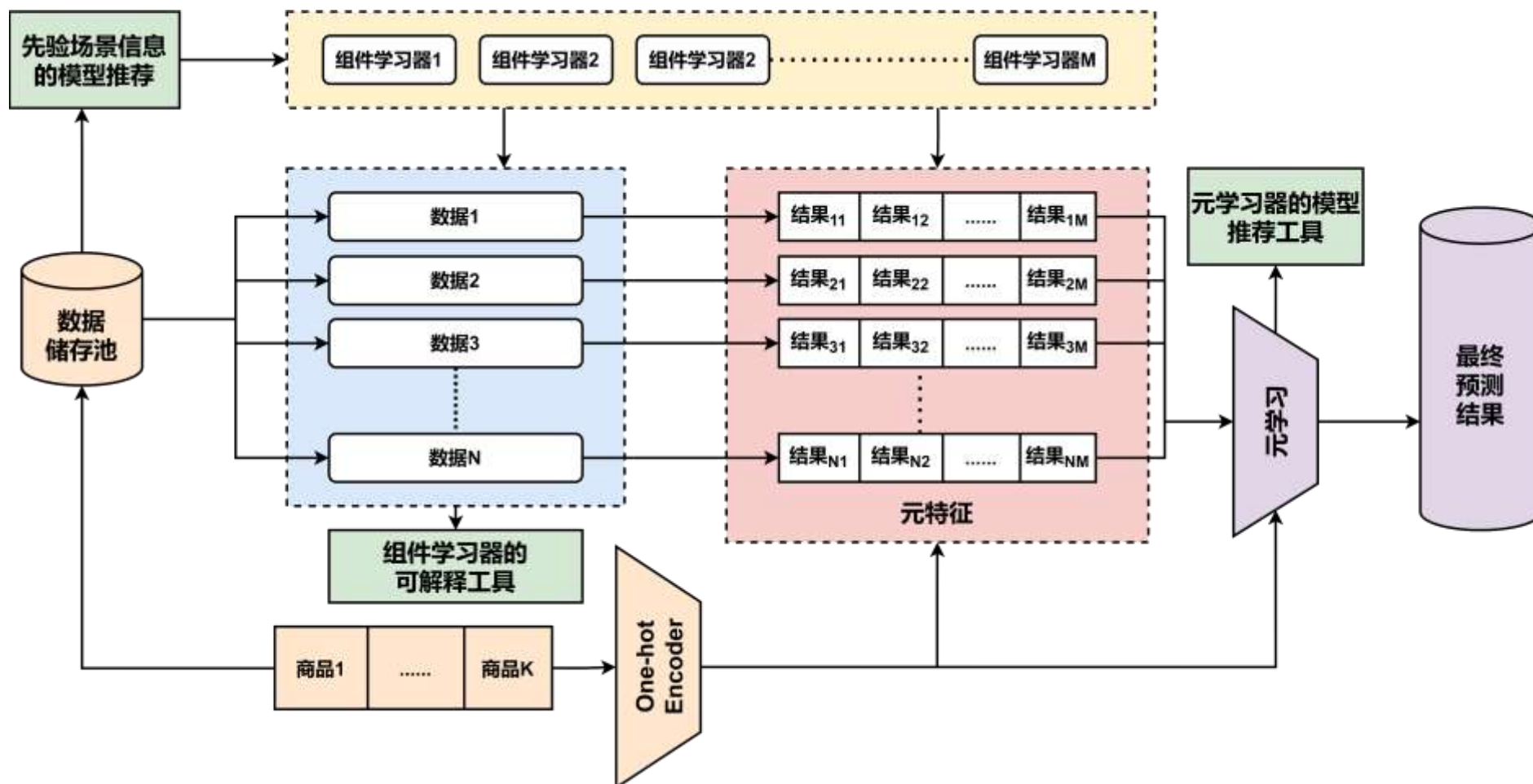
■ 双层Stacking（有监督的异质组件学习器+有监督的元学习器）

- 集成数据场景下的模型推荐，根据数据量、场景的不同对异质组件学习进行二次选择或分配权重
- 集成基于结果的模型推荐，根据预测结果，基于元学习器的可解释性框架进行模型推荐
- 集成鲁棒的变量可解释性，集成异质组件学习器的可解释性工具与有监督的元学习器的可解释性工具

目标：解决预测模型操作繁琐、使用门槛高、被一线工作人员不信任、需要后续二次调整的问题



□ 自动预测技术



Liu, Y., Lyu, C., Khadka, A., Zhang, W., and Liu, Z.*, 2019. Spatio-temporal Ensemble Method for Car-Hailing Demand Prediction. *IEEE Transactions on Intelligent Transport Systems*.

□ 边际效应与因果效应

- 离散变量的边际与因果效应需要输出在当前状态 $[T_i, W]$ 下控制变量 T_i 增加一个单位的变化对于因变量 Y 的影响程度，即：

$$CE(T_i) = E_s[Y(T_i = t_0 + 1, W = w) - Y(T_i = t_0, W = w)]$$

- 由于供应链数据集中大部分变量为连续变量，其边际与因果效应可以表示为

$$CE(T_i) = \lim_{h \rightarrow 0} E_s \left[\frac{Y(T_i = t_0 + h, W = w) - Y(T_i = t_0, W = w)}{h} \right]$$

变量名	描述
Y	因变量（结果变量）
T_i	自变量（控制变量）
W	混杂变量
$CE(T_i)$	边际效应（因果效应）
h	无穷小量
X_i	测试数据

变量名	描述	变量名	描述
$Freq_{lag1}$	上一周的供货频率	QTY	这一周的供货量
$Freq_{lag2}$	上两周的供货频率	QTY_{lag1}	上一周的供货量
$Freq_{lag3}$	上三周的供货频率	QTY_{lag2}	上两周的供货量
$Freq_{lag4}$	上四周的供货频率	QTY_{lag3}	上三周的供货量
$Freq_{lag5}$	上五周的供货频率	QTY_{lag4}	上四周的供货量

变量名	值	变量名	值
$Freq_{lag1}$	21	QTY	242
$Freq_{lag2}$	18	QTY_{lag1}	200
$Freq_{lag3}$	32	QTY_{lag2}	626
$Freq_{lag4}$	131	QTY_{lag3}	3747
$Freq_{lag5}$	263	QTY_{lag4}	7364

□ 边际效应-回归分析 (OLS)

- 使用简单的线性模型来拟合供应链数据
- 没有考虑控制变量的交叉项与多项式，计算得到的边际效应为固定值，无法满足复杂供应链场景的需求

变量	系数	方差	t	P值	CI下界	CI上界
$Freq_{lag1}$	17.6574	0.782	22.587	0.000	16.123	19.192
$Freq_{lag2}$	-5.1463	0.755	-6.815	0.000	-6.629	-3.664
$Freq_{lag3}$	-0.5313	1.235	-0.430	0.667	-2.956	1.893
$Freq_{lag4}$	-1.3648	1.128	-1.210	0.227	-3.579	0.850
$Freq_{lag5}$	-1.5475	0.831	-1.862	0.063	-3.179	0.084
QTY_{lag1}	0.5102	0.018	27.992	0.000	0.474	0.546
QTY_{lag2}	-0.1441	0.032	-4.511	0.000	-0.207	-0.081
QTY_{lag3}	0.1421	0.034	4.136	0.000	0.075	0.210
QTY_{lag4}	-0.0164	0.030	-0.552	0.581	-0.075	0.042

□ 边际效应-stacking

■ 双层stacking

变量名	边际效应	变量名	边际效应	变量名	边际效应
$Freq_{lag1}$	11.8559	$Freq_{lag4}$	0.0000	QTY_{lag2}	0.0000
$Freq_{lag2}$	0.5776	$Freq_{lag5}$	0.3883	QTY_{lag3}	0.0000
$Freq_{lag3}$	0.3344	QTY_{lag1}	2.0856	QTY_{lag3}	0.0000

□ 因果效应

- **为什么需要因果推断：** 混杂因素可能导致错误地将相关性误解为因果关系。如果未正确控制或考虑混杂因素，可能会错误地断定一个变量导致了另一个变量的变化，而实际上这种关系可能是由混杂因素引起的。

- 因果效应估计

$$CE(T_i) = E_s[Y_i^{(t_1)} - Y_i^{(t_2)} | W_i = w]$$

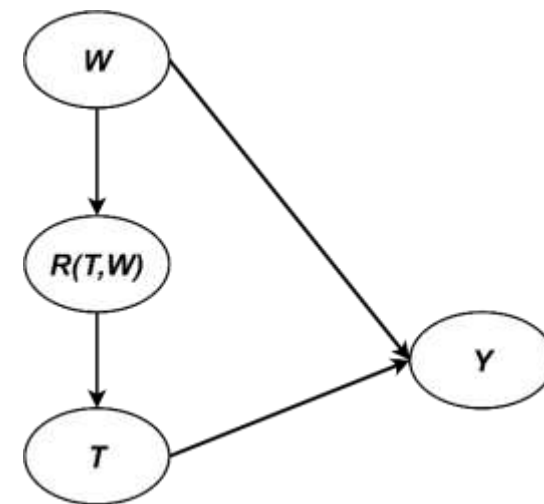
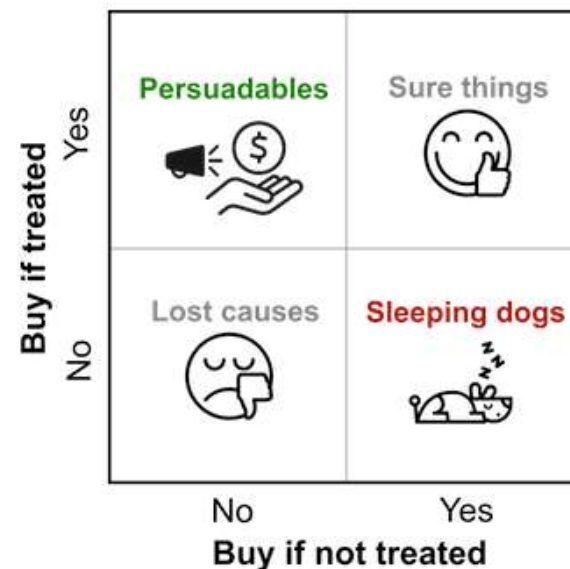
- 外生性假设

$$\{Y_{t_i}^{t_0}, Y_{t_i}^{t_1}\} \perp T_i | W_i$$

- 重叠性假设

$$\varepsilon < P[T = t_0 | W = w] < 1 - \varepsilon$$

$$CE(T_i) = \frac{1}{|\{i: T_i = t_0, W_i \in L\}|} \sum_{\{i: T_i = t_0, W_i \in L\}} Y_i - \frac{1}{|\{i: T_i = t_1, W_i \in L\}|} \sum_{\{i: T_i = t_1, W_i \in L\}} Y_i$$



□ 因果效应

- 局部因果效应
- 因果随机森林

变量	系数	方差	t	P值	CI下界	CI上界
$Freq_{lag1}$	25.27878	2.421724	10.43834	1.66E-25	20.53229	30.02527
$Freq_{lag2}$	-4.29437	2.95413	-1.45368	1.46E-01	-10.0844	1.495616
$Freq_{lag3}$	1.182548	3.428514	0.344916	7.30E-01	-5.53722	7.902312
$Freq_{lag4}$	-5.2583	2.682254	-1.96041	4.99E-02	-10.5154	-0.00118
$Freq_{lag5}$	1.320056	1.673834	0.788642	4.30E-01	-1.9606	4.600711
QTY_{lag1}	0.289844	0.109485	2.647337	8.11E-03	0.075257	0.50443
QTY_{lag2}	0.205848	0.108064	1.904877	5.68E-02	-0.00595	0.417649
QTY_{lag3}	0.230538	0.122192	1.886695	5.92E-02	-0.00895	0.470029
QTY_{lag4}	-0.13235	0.063359	-2.08897	3.67E-02	-0.25654	-0.00817

□ 自动预测技术-先验模型推荐

^	FREQ ÷	FREQ_lag1 ÷	FREQ_lag2 ÷	FREQ_lag3 ÷	FREQ_lag4 ÷	QTY_lag1 ÷	QTY_lag2 ÷	QTY_lag3 ÷	QTY_lag4 ÷
4	23	11.0	7.0	6.0	0.0	81.0	176.0	122.0	0.0
5	35	23.0	11.0	7.0	6.0	186.0	81.0	176.0	122.0
6	9	35.0	23.0	11.0	7.0	793.0	186.0	81.0	176.0
7	58	9.0	35.0	23.0	11.0	269.0	793.0	186.0	81.0
8	164	58.0	9.0	35.0	23.0	2666.0	269.0	793.0	186.0
9	31	164.0	58.0	9.0	35.0	4796.0	2666.0	269.0	793.0
10	40	31.0	164.0	58.0	9.0	1484.0	4796.0	2666.0	269.0
11	36	40.0	31.0	164.0	58.0	2282.0	1484.0	4796.0	2666.0
12	49	36.0	40.0	31.0	164.0	4000.0	2282.0	1484.0	4796.0
13	36	49.0	36.0	40.0	31.0	2300.0	4000.0	2282.0	1484.0
14	39	36.0	49.0	36.0	40.0	700.0	2300.0	4000.0	2282.0
15	67	39.0	36.0	49.0	36.0	966.0	700.0	2300.0	4000.0
16	51	67.0	39.0	36.0	49.0	744.0	966.0	700.0	2300.0
17	84	51.0	67.0	39.0	36.0	486.0	744.0	966.0	700.0
18	59	84.0	51.0	67.0	39.0	1702.0	486.0	744.0	966.0
19	85	59.0	84.0	51.0	67.0	1049.0	1702.0	486.0	744.0
20	160	85.0	59.0	84.0	51.0	1297.0	1049.0	1702.0	486.0
21	345	160.0	85.0	59.0	84.0	5562.0	1297.0	1049.0	1702.0
22	279	345.0	160.0	85.0	59.0	8083.0	5562.0	1297.0	1049.0
23	34	279.0	345.0	160.0	85.0	9300.0	8083.0	5562.0	1297.0
24	81	34.0	279.0	345.0	160.0	618.0	9300.0	8083.0	5562.0
25	65	81.0	34.0	279.0	345.0	1267.0	618.0	9300.0	8083.0
26	23	65.0	81.0	34.0	279.0	1721.0	1267.0	618.0	9300.0

□ 组件学习器

- XGBoost, Generalized Linear Model, Linear Model, Gradient Boost Regressor, Random Forest, KNN

□ 元学习器

- Xgboost

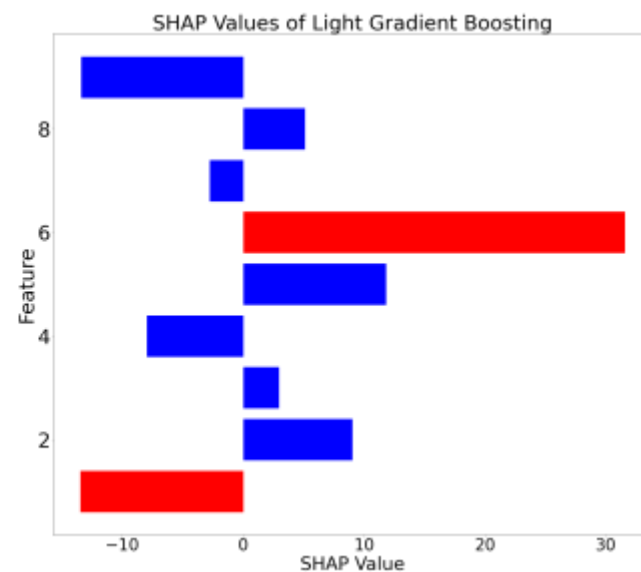
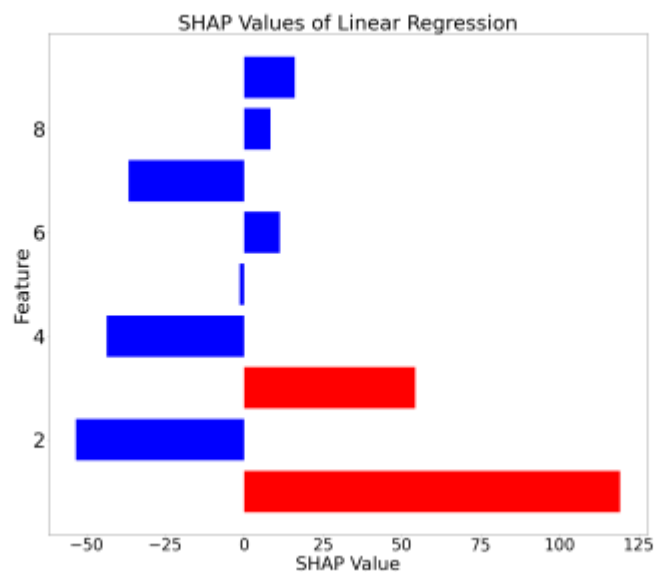
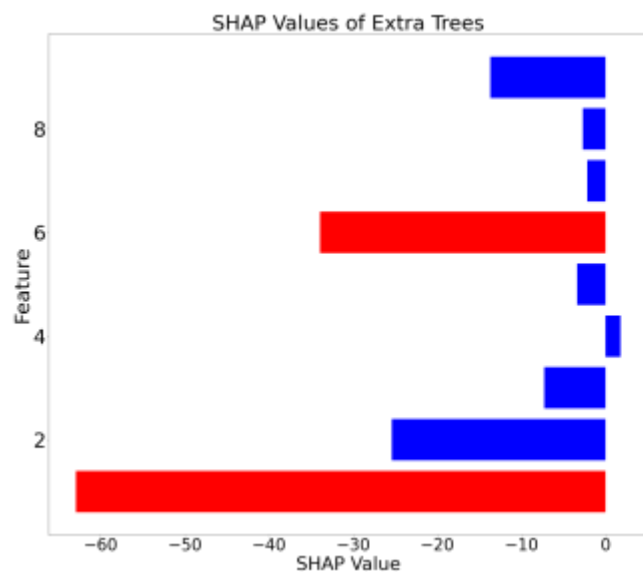
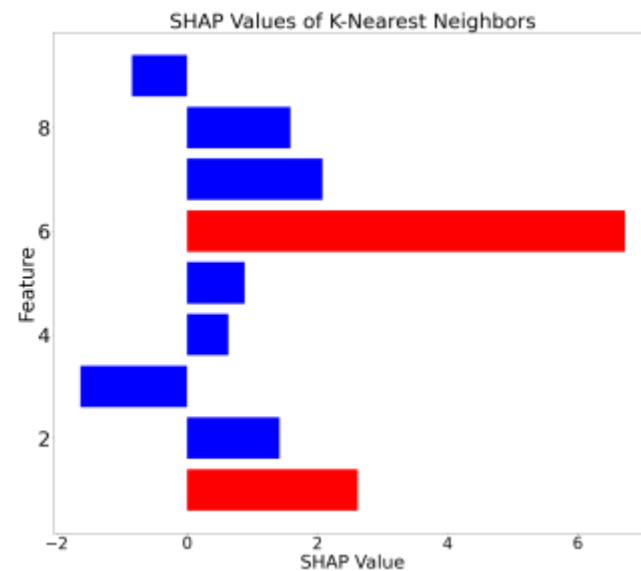
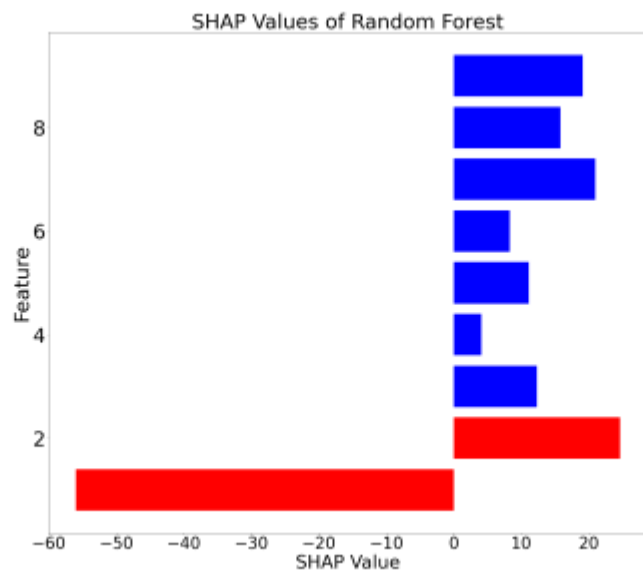
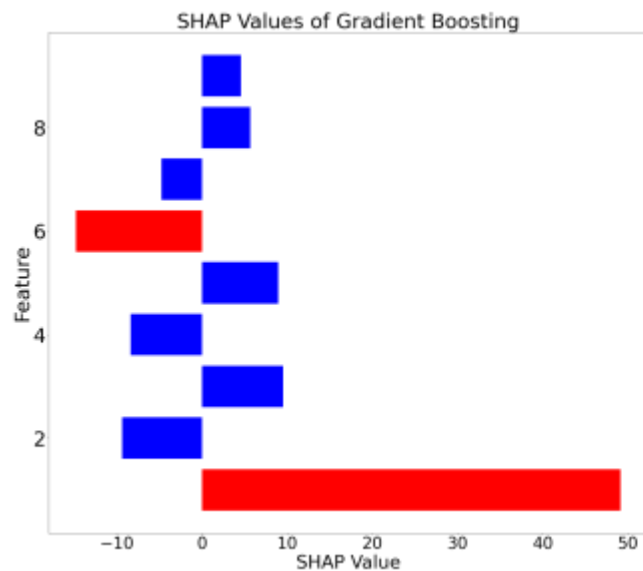
□ RMSE

- Stacking拥有最低的RMSE

Model	RMSE
XGBoost	561.7620582664131
GLM	960.2766749379653
LM	561.5574714078268
GBR	589.1003235432349
RF	486.2766519926351
KNN	713.664081885856
Stacking	477.9539265928434

Liu, Y., Liu, Z.*, and Jia, R., 2019. DeepPF: A Deep Learning Based Architecture for Metro Passenger Flow Prediction. *Transportation Research Part C*, 101, 18-34.

创新性-自动预测技术



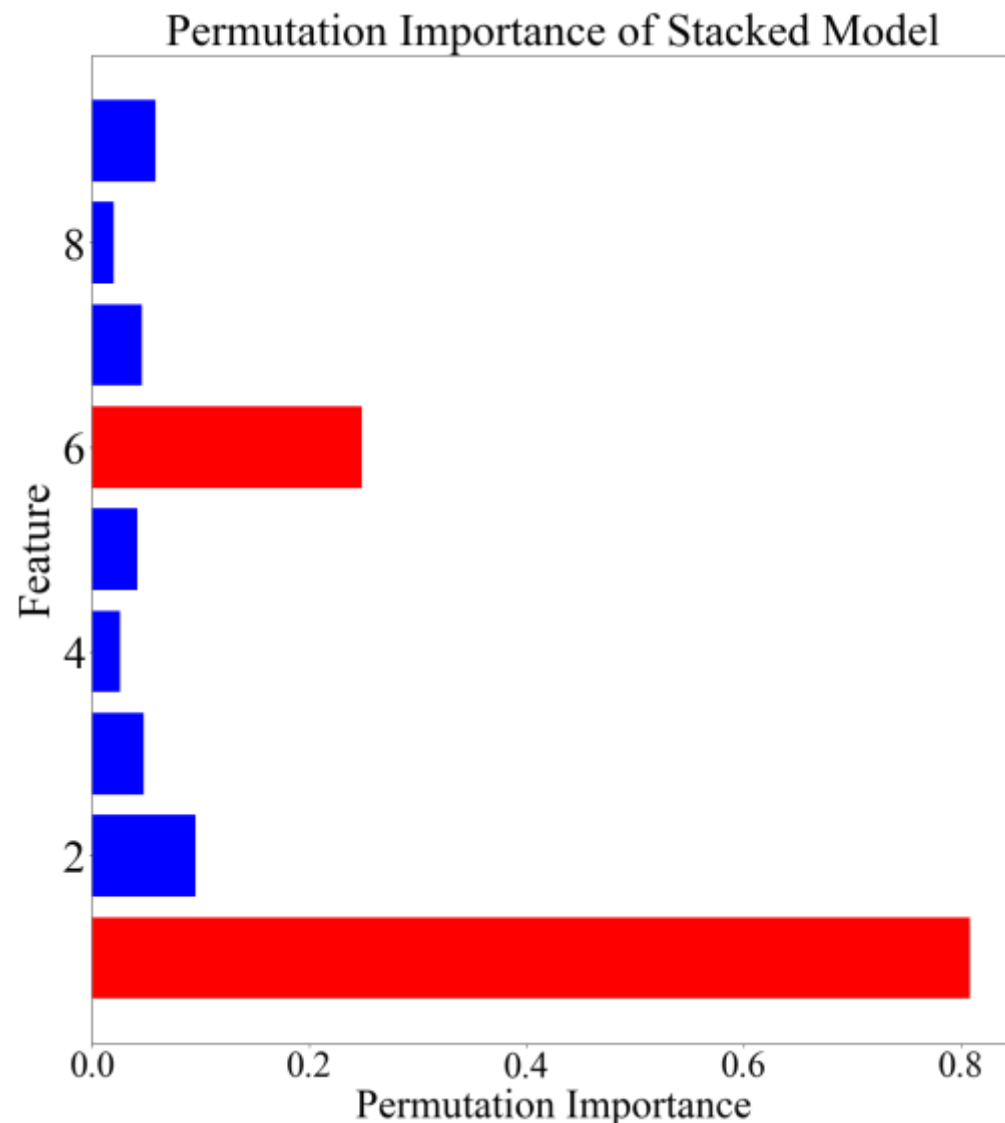
□ 自动预测技术-可解释性

- 基于**重抽样**融合组件学习器层的可解释性与元学习器层的可解释性，得到更加**鲁棒**的模型可解释性工具
- 后续创新方向：基于**贝叶斯理论**改进可解释性

Algorithm 2: Consensus average interpretation of stacked models

Initialization: $\{CL\}, ML, X_{tr}, Y_{tr}, F, L, n_{re}; CM_{per}$ and CM_{SHAP} with zeros of size (n_{re}, n_{re}, F) ; $St(\{CL\}, ML)$

```
(1) for loc in range(1, L + 1), do:
(2)   standardize  $X_{tr}^{(loc)}$ 
(3) end for
(4) for loc in range(1, L + 1), do:
(5)   get  $X_{tr}^{(loc)}$  and  $Y_{tr}^{(loc)}$ 
(6)   for i in range(1,  $n_{re}$ ), do:
(7)     resample  $X_{tr}^{(loc)}$  and  $Y_{tr}^{(loc)}$  to get  $X_{resample}$  and  $Y_{resample}$  of size  $n_{re}$ 
(8)     fit  $St(\{CL\}, ML)$  on  $X_{resample}$  and  $Y_{resample}$ 
(9)     compute  $Per_{fea}$  and  $SHAP_{fea}$  on  $X_{resample}$  and  $Y_{resample}$ 
(10)    store the mean of  $Per_{fea}$  and  $SHAP_{fea}$  in  $CM_{per}$  and  $CM_{SHAP}$ , respectively
(11)   end for
(12) end for
(13) compute the mean of  $CM_{per}$  and  $CM_{SHAP}$  against axis 0,1
(14) plot results
```



□ 自动预测技术-先验模型推荐

■ 模型自身特点

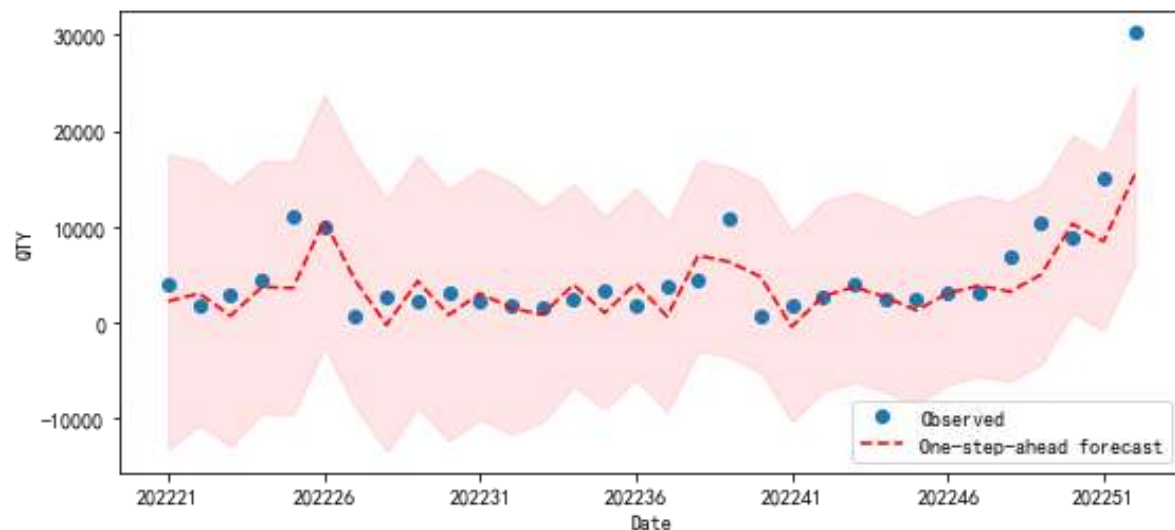
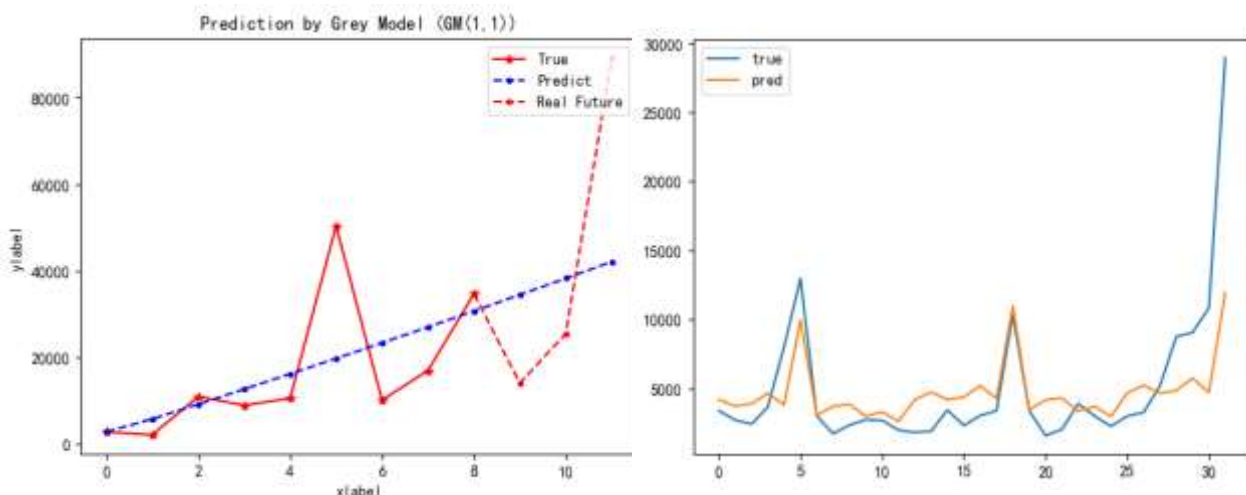
- 比如灰度模型适用于少数据量与短序列预测、Informer适用于长序列趋势预测但需要大量数据、ARIMA适用于短序列季节性预测、Prophet适用于趋势预测

■ 业务场景可解释性要求

- 针对不同的业务场景的模型需求进行推荐
- 有些场景需要模型具有可解释性，有些场景只需要模型输出的预测结果

□ 自动预测技术-元学习器模型推荐

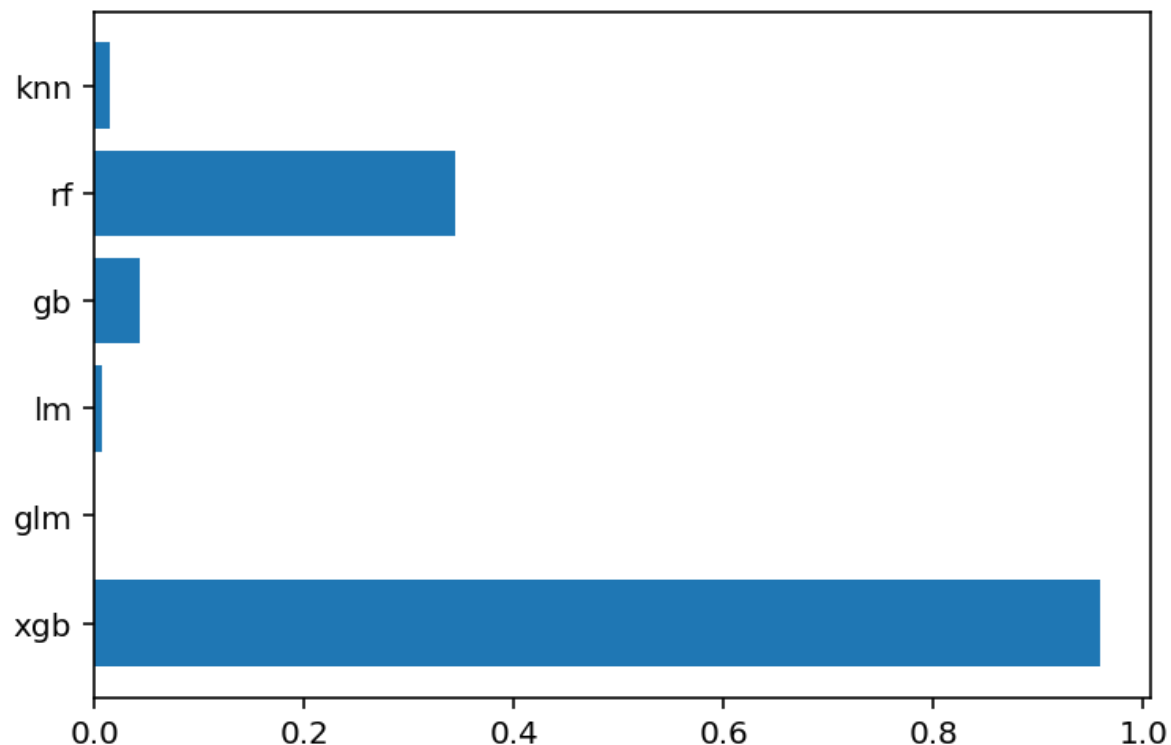
- 基于元学习器的可解释性工具
- 以结果为导向的模型推荐



□ 自动预测技术-可解释性

■ 从结果出发的模型推荐

■ **XGBoost = 0.959054**, Generalized Linear Model = 0, Linear Model = 0.006976, Gradient Boost Regressor = 0.043222, **Random Forest = 0.343530**, KNN = 0.015311

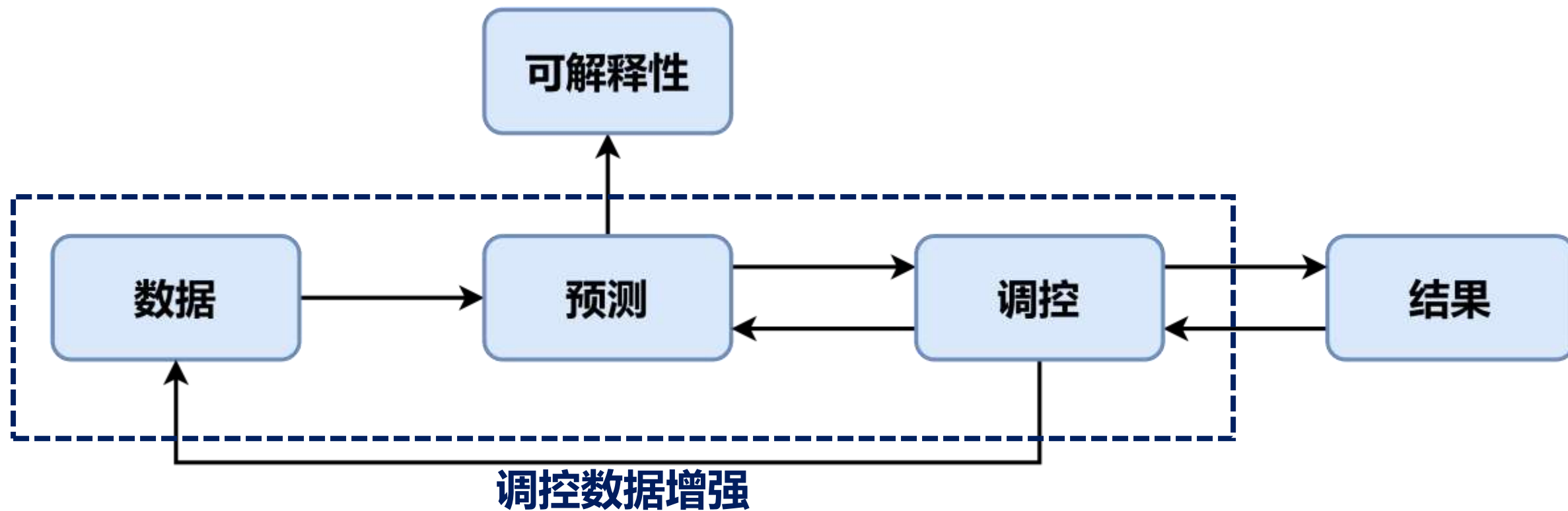


二、调控数据增强



東南大學
SOUTHEAST UNIVERSITY





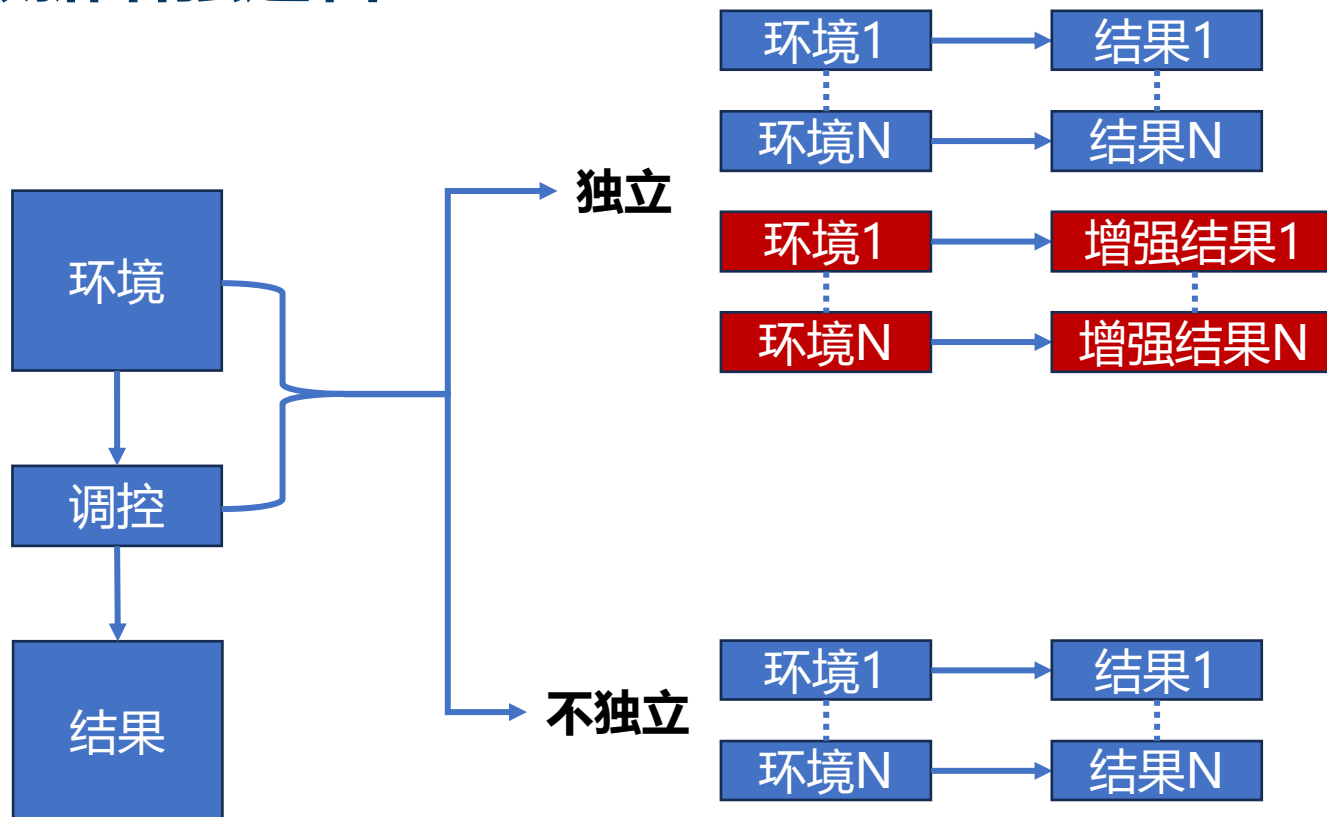
□ 为何需要调控数据增强？

- 预测的历史数据收集较容易，但调控是**稀疏的并且难以收集**，调控之后结果的收集有时间差，在这段时间差内，**调控没有任何先验信息，传统调控方法只能依靠人工**
- 人工直接按照约束对预测值进行调控，不可信
- 借助数据增强，可以使得**模型科学的、合理的自动生成调控值**

□ 调控的稀疏性指的是什么？

- 调控约束存在非常多的场景：调控节奏[10%、30%、20%、40%]; [20%、10%、20%、50%]。**每一种不同的调控约束都是一种场景，每个场景下收集到的调控样本会很少 (Few-shot)**
- **非常多的场景没有样本 (Zero-shot)**

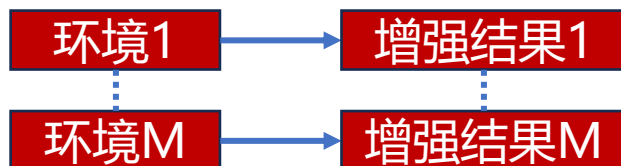
□ 数据增强逻辑



逻辑:

(1) 若调控与环境之间独立, 既调控的先验知识不来源于过去环境, 则对所有真实数据结果进行增强;

(2) 若调控与环境之间不独立, 既环境导致了调控, 则计算真实数据环境与调控样本环境的马氏距离, 根据距离阈值筛选需要增强的样本, 并加权增强 (噪声)。



□ 面向调控的数据增强

■ 基于调控场景的数据增强

- 例子：发货节奏调控，Q1 10%、Q2 30%、Q3 20%、Q4 40%
- 根据马氏距离选择真实数据进行模拟发货节奏，在训练集中构造符合既定发货节奏的数据，称为增强数据
- 标记增强数据，与真实数据混合进行训练

■ 损失函数

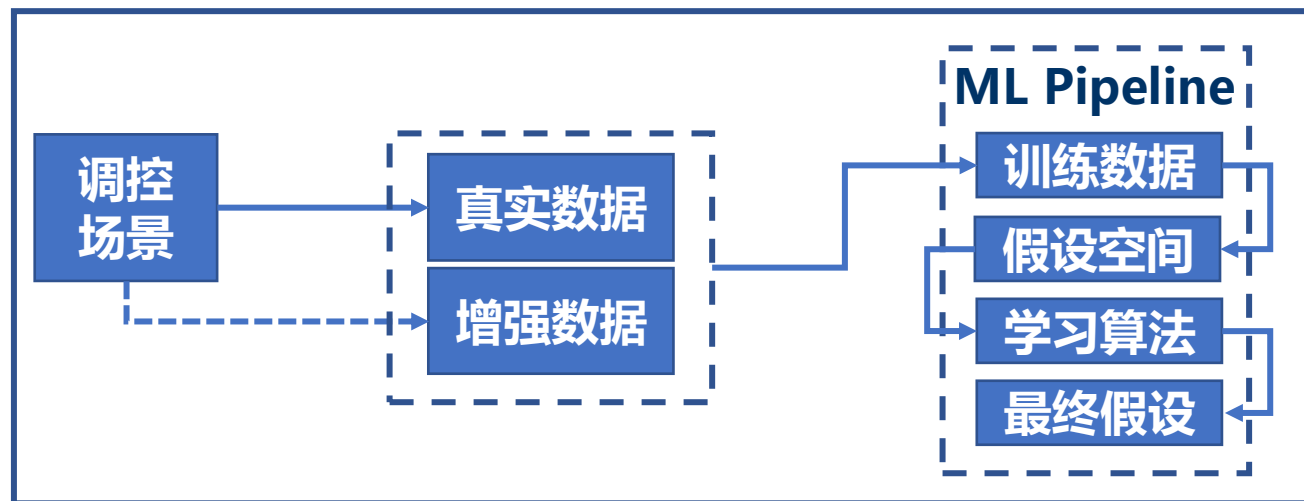
- 真实数据：
$$L_r = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

- 增强数据：
$$L_a = \frac{1}{m} \sum_{i=1}^m (Y_i^a - \hat{Y}_i)^2$$

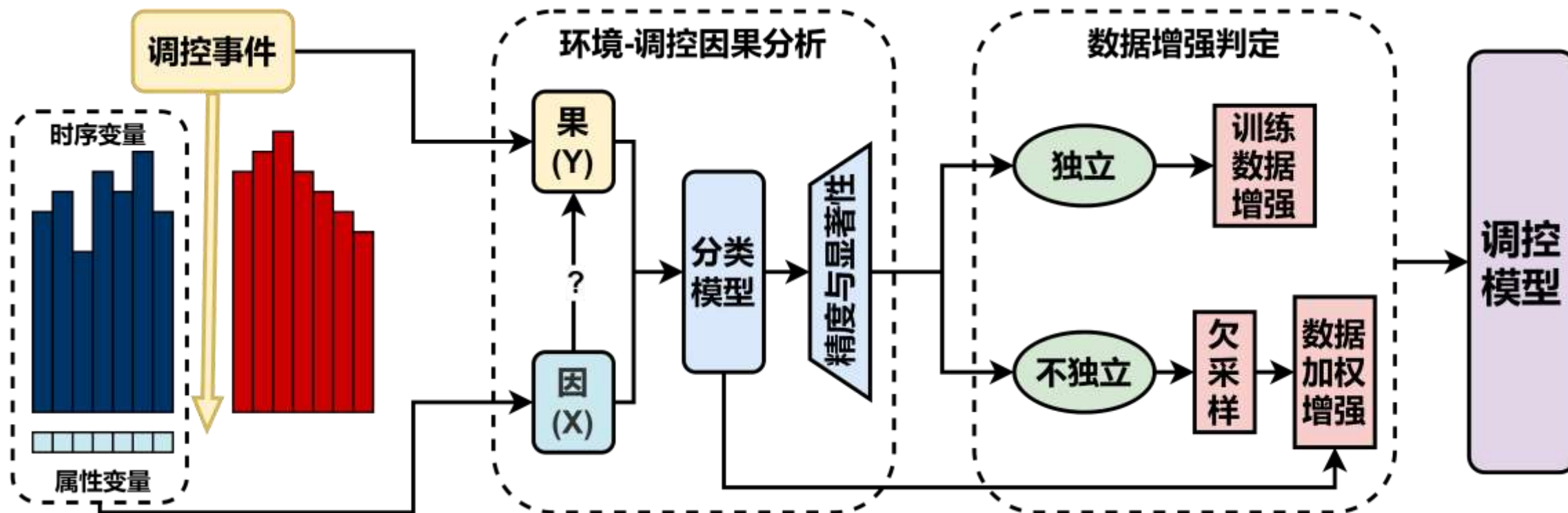
- 损失函数：

$$L = L_r + \alpha L_a$$

目标：解决调控场景复杂、调控约束难以融入预测模型的问题；帮助模型度过zero-shot的尴尬期



□ 数据增强逻辑



□ 面向调控的先验信息注入

■ 损失函数（以调控节奏为例）

➤ 真实数据:

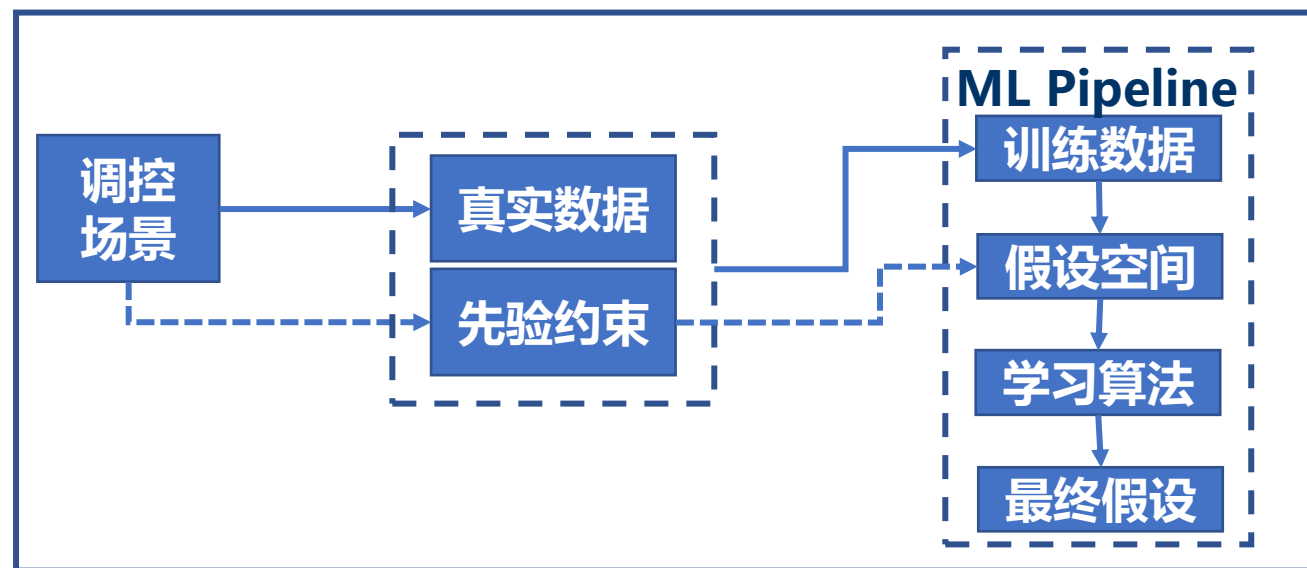
$$L_r = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

➤ 损失函数:

$$Loss = L_r + \beta \left\| \frac{\hat{Y}_{\parallel}}{\sum_{i=1}^n \hat{Y}_i} - b \right\|_2$$

➤ 约束信息

$$b = [0.1, 0.3, 0.2, 0.4]$$



□ 数据准备

- 数据维度为7904*17，需要预测的为下四个月的需求量
- 随机插入1000条版本节奏调控信息（10%、30%、20%、40%），调控信息为二元变量

FREQ :	FREQ_lag1 :	FREQ_lag2 :	FREQ_lag3 :	FREQ_lag4 :	QTY :	QTY_lag1 :	QTY_lag2 :	QTY_lag3 :	QTY_lag4 :	QTY_pre1 :	QTY_pre2 :	QTY_pre3 :	Control v
3	1.0	2.0	0.0	3.0	8	2.0	22.0	0.0	35.0	6.0	6.0	5.0	1
0	0.0	0.0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1
1	3.0	0.0	1.0	6.0	1	25.0	0.0	5.0	50.0	0.0	0.0	3.0	1
0	0.0	0.0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1
1	3.0	0.0	0.0	1.0	1	9.0	0.0	0.0	1.0	0.0	2.0	1.0	1
1	0.0	0.0	1.0	0.0	3	0.0	69.0	1.0	0.0	0.0	0.0	0.0	1
0	1.0	0.0	0.0	1.0	0	3.0	0.0	69.0	1.0	0.0	0.0	0.0	1
0	9.0	5.0	5.0	1.0	0	69.0	32.0	146.0	4.0	16.0	9.0	2.0	1
8	2.0	1.0	2.0	0.0	47	104.0	18.0	9.0	0.0	6.0	5.0	0.0	1
0	0.0	5.0	6.0	8.0	0	0.0	153.0	34.0	47.0	0.0	0.0	1.0	1
17	33.0	17.0	9.0	7.0	137	231.0	353.0	37.0	24.0	0.0	5.0	9.0	1
3	2.0	4.0	4.0	16.0	65	24.0	23.0	20.0	91.0	1.0	2.0	10.0	1
1	2.0	0.0	0.0	0.0	6	7.0	0.0	0.0	0.0	0.0	1.0	1.0	1
4	5.0	7.0	16.0	4.0	23	14.0	29.0	80.0	8.0	5.0	15.0	10.0	1
2	1.0	1.0	1.0	0.0	11	6.0	63.0	2.0	0.0	2.0	1.0	0.0	1
7	8.0	1.0	2.0	0.0	489	68.0	21.0	18.0	0.0	2.0	1.0	1.0	1
1	6.0	10.0	0.0	13.0	3	113.0	66.0	0.0	176.0	0.0	0.0	0.0	1
1	1.0	2.0	7.0	8.0	7	15.0	19.0	489.0	68.0	4.0	3.0	16.0	1

□ 独立性检验

■ 查看多元环境变量与二元调控变量的相关性

■ 逻辑回归的方差齐性检验F-test

➤ F value: -2.79

➤ P value: 1.00

➤ 接受原假设，所有回归参数都等于0，模型所有参数不显著异于0，模型不存在

```
LR F: -2.795352687619544
LR p: 1.0
```

	coef	std error	tval	pval	[lower alpha	upper alpha]
FREQ	0.004754	5.63779e-05	84.3198	0	0.00464327	0.00486427
FREQ_lag1	-0.001798	9.0708e-05	-19.7282	1.31706e-84	-0.00196728	-0.00161172
FREQ_lag2	-0.009041	9.21621e-05	-98.0952	0	-0.0092213	-0.00886003
FREQ_lag3	0.023674	9.26263e-05	255.582	0	0.0234921	0.0238552
FREQ_lag4	-0.026067	7.96881e-05	-327.108	0	-0.0262228	-0.0259104
QTY_lag1	-0.000005	2.47018e-06	-1.84868	0.0645416	-9.40805e-06	2.74895e-07
QTY_lag2	-0.000025	2.59872e-06	-9.52944	2.06079e-21	-2.98578e-05	-1.9671e-05
QTY_lag3	-0.000296	2.59163e-06	-114.284	0	-0.000301263	-0.000291104
QTY_lag4	0.000187	2.46949e-06	75.887	0	0.000182562	0.000192242

结论：

该调控与环境独立，环境中没有明确的信号导致生成该调控指令

□ 数据增强

- 调控的约束为节奏调整: [10%、30%、20%、40%]

1 rows × 10 columns pd.DataFrame										
	FREQ	FREQ_lag1	FREQ_lag2	FREQ_lag3	FREQ_lag4	QTY_lag1	QTY_lag2	QTY_lag3	QTY_lag4	Control
1126	23	95.0	22.0	38.0	47.0	3521.0	1284.0	913.0	4030.0	

- 将上周的FREQ变为100，原先的未来四周的使用stacking方法的预测值为：
[1055.730347,23.80345,37.492107,39.420593]
- 经过数据增强之后预测值为: [309.29, 824.23, 549.47, 1098.95]
- 真值为: [138, 31.0, 15.0, 38.0]

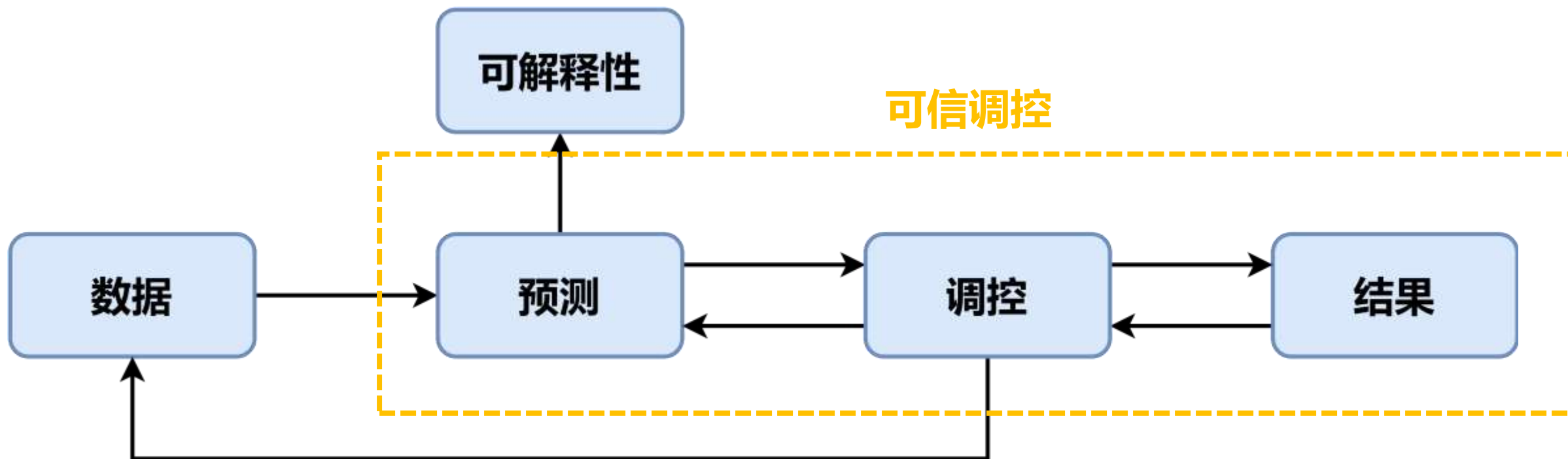
1 rows × 4 columns np.ndarray				
	0	1	2	3
0	309.298248	824.234802	549.479675	1098.958252

三、可信调控



東南大學
SOUTHEAST UNIVERSITY





□ 数据展示

■ 100条调控, 12个月

label	1	2	3	4	5	6	7	8	9	10	11	12	y	control
A03784	14317	47781	53464	46903	41924	123423	32345	56850	65150	68051	89562	194292	56317	45000
A03767	5884	16788	30260	17992	24796	231120	15024	18916	26284	45392	26568	84756	32300	32900
A04091			176	3822	2684	89487	13500	19247	26402	35800	43224	96080	23303	10000
A08445	4792	4601	15170	9887	12265	66638	14264	25652	55359	20447	35153	132804	19929	90000
A08446	1978	2646	7248	5908	5535	17928	9518	10158	25210	23561	27955	98029	33631	20000
A04421	2147	9010	11290	7082	8944	66435	5675	8571	21162	14515	18760	50654	18444	40000
A08419	300	15000	6098	21090	1220	2342	1735	11659	37279	53917	95284	241180	20554	50000
A08451	12229	5056	14466	27633	20764	65600	17955	14346	29840	15697	24488	84559	17696	40000
A08447	3180	10036	19321	16689	17996	40280	9230	12471	30019	18481	31758	102082	10845	21000
A04090	2183	8139	11379	7256	8956	69557	4377	8485	10640	14479	10778	36801	10348	100000
A03777	802	4746	2999	5691	4323	22544	9356	5534	25081	16304	31103	91807	9209	20000
A04314	1544	4985	7128	6508	4552	21892	9748	8176	26100	14156	28012	79052	7207	10000
A03796	2692	2106	6283	6788	5340	16866	6526	7321	13974	10172	16564	62709	11895	9000
A04419	72	2280	1245	5496	126	951	700	6908	17328	17536	36166	111132	8443	20000
A03781	142	588	3404	1914	1692	3980	2354	4126	5524	11220	6495	22682	28412	28000
A04422	2271	5125	6279	6178	5606	14409	4628	9333	13767	14899	13622	42102	12796	30000
A00452	676	3749	3765	2584	2445	8833	4449	4282	12214	7288	11138	32660	13293	60000
A03826	10033	244	3879	941	1630	23176	2254	4483	3877	4233	6534	19348	3530	10000

□ 稀疏场景下的可信调控

■ 稀疏调控场景的收集

- 场景是**Zero-shot**或**Few-shot**，导致调控的准确性受质疑
- **打通供应链下游，收集调控结果数据**，使得调控模型从Zero-shot到Few-shot

■ 可信调控

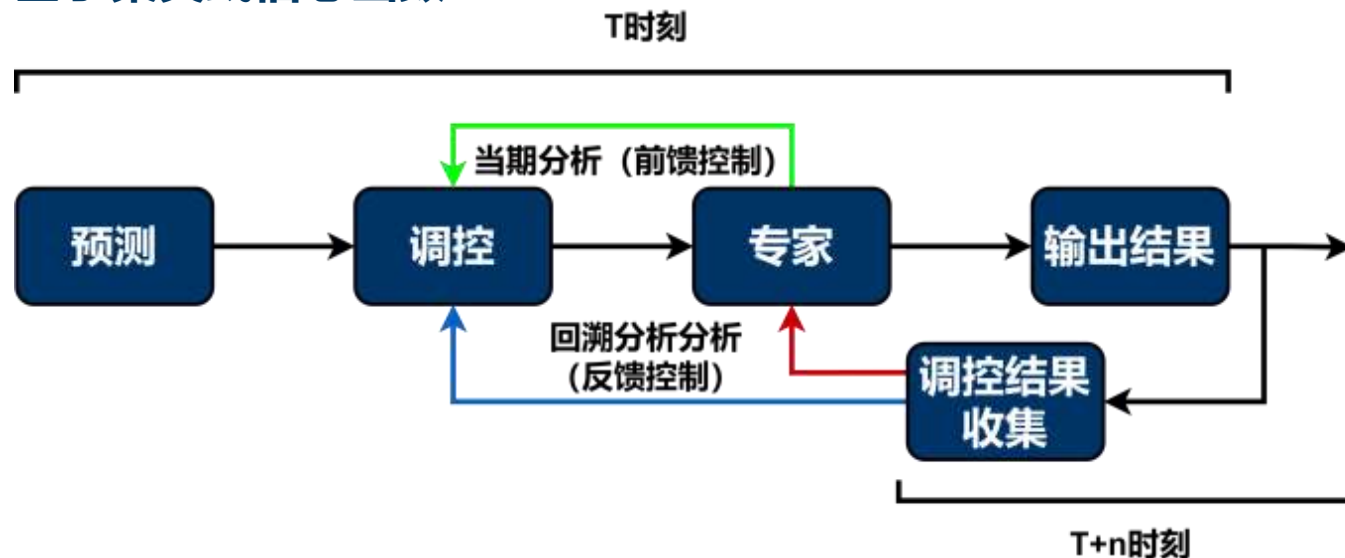
- 当期调控与回溯分析，融入了**控制理论**
- **前馈控制**(当前时间节点的专家意见，指导当前节点的调控值)
- **反馈控制**(在一段时间间隔后，模型收集到真实供需数据，反馈给调控模型)

目标：结合当期调控与回溯分析，解决模型调控的稀疏性与调控的可信度受一线工作人员质疑的问题

□ 稀疏场景下的可信调控

■ 反馈控制（专家）

- 弱监督学习-不准确监督，识别潜在的专家调控的误差，然后尝试进行修正
- 调控存在噪声（误差），**标签噪声容忍学习算法**，同时学习标签噪声模型和调控预测器，将数据生成过程的两个组件（专家的噪声与调控）解耦
- 基于概率、基于聚类或信念函数



□ 认知不确定性与偶然不确定性

- **认知不确定性**：用来描述模型中的不确定性（专家调控环境的稀疏性）

$$Var(y) = \sigma^2 + \frac{1}{T} \sum_t f(x_t)^T f(x_t) - E(y)^T E(y)$$

- **偶然不确定性**：用来描述观测中固有的噪声（专家调控自身的误差）

$$\sigma^2(x_i) = E[(f(x) - E(f(x_i)))^2]$$

□ 专家调控包含了两种不确定性与一种可信度

- 基于回溯分析得到的真值与专家调控值的残差，这种残差被假设与其他的变量相关，即可以用其他的变量预测
- 稀疏场景下不考虑拟合残差
- 专家调控的两种不确定性 $Var(y), \sigma^2(x_i)$ 来自拟合、专家调控的可信度 $C(x_i)$ 来自噪声
- 可信度与不确定性融合输入至专家调控修正环节

□ 可信度与不确定性

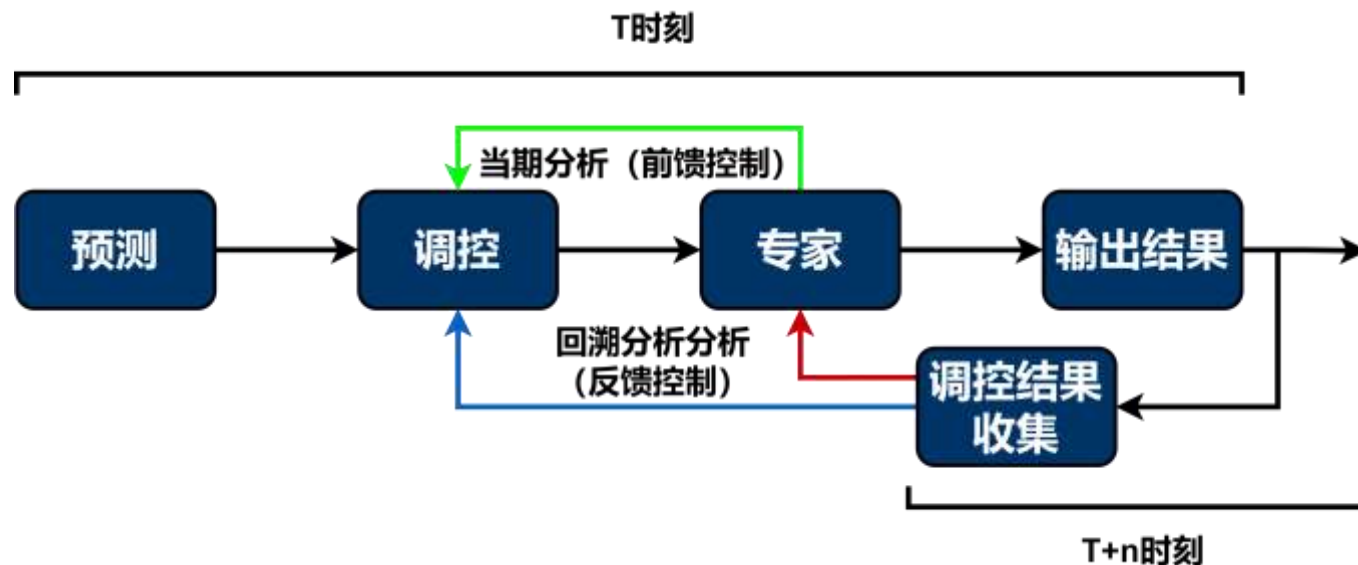
- 可信度系数值高于**0.8**，则说明可信度高
- 可信度系数介于**0.7~0.8**之间，则说明可信度较好
- 可信度系数介于**0.6~0.7**，则说明可信度可接受
- 可信度系数小于**0.6**，说明可信度不佳
- 可信度可以量化每一个样本的（每一次专家调控）可信程度
- 认知不确定性：5827.048120182586

	res	eu	confi	Norm_eu
0	11317	5272.236156	0.172644	0.495894
1	600	6293.840888	0.866558	0.593960
2	13303	9682.222510	0.271458	0.919216
3	70071	3657.178569	0.000424	0.340862
4	13631	8936.893454	0.432315	0.847671
5	21556	9490.835469	0.693197	0.900844
6	29446	6859.409775	0.113219	0.648250
7	22304	6413.618289	0.290077	0.605457
8	10155	8782.017544	0.836231	0.832804
9	89652	4931.161106	0.000133	0.463154
10	10791	9659.193851	0.266531	0.917005
11	2793	6867.103617	0.577893	0.648988
12	2895	7778.269012	0.623370	0.736452
13	11557	10398.276699	0.677084	0.987951
14	412	1230.380236	0.197947	0.107910
15	17204	5924.205149	0.173771	0.558478
16	46707	4504.351729	0.006983	0.422184
17	6470	4761.019435	0.525174	0.446822
18	10760	8924.738514	0.440316	0.846504
19	7253	6633.554293	0.585770	0.626569

□ 稀疏场景下的可信调控

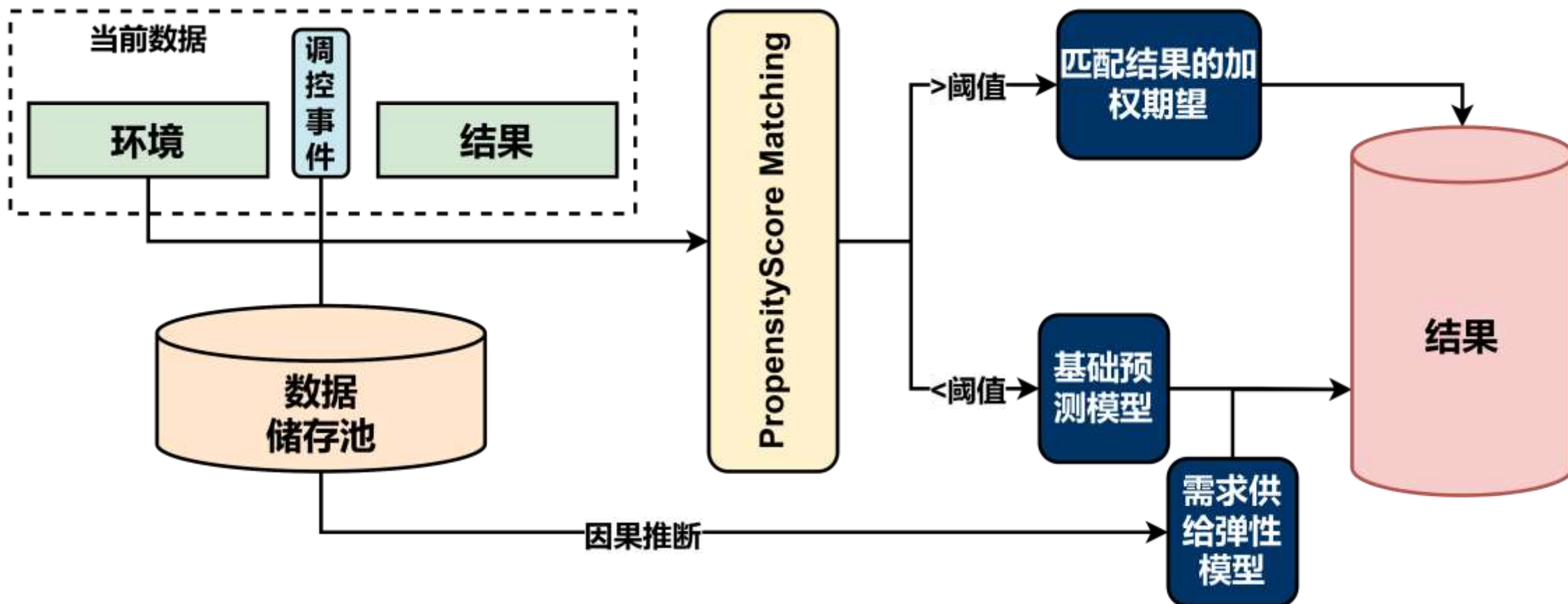
■ 反馈控制（调控）

- 计算调控场景（调控编码+环境）的**Propensity Score**
- 如果Propensity Score高于阈值，使用**匹配值的加权期望作为调控值**
- 如果Propensity Score低于阈值，使用**基础需求供给预测调控模型+需求供给弹性模型**
- 需求供给弹性模型：基于**因果推断**，预估需求供给对于环境的敏感度，作为调控值的参考



□ 稀疏场景下的可信调控

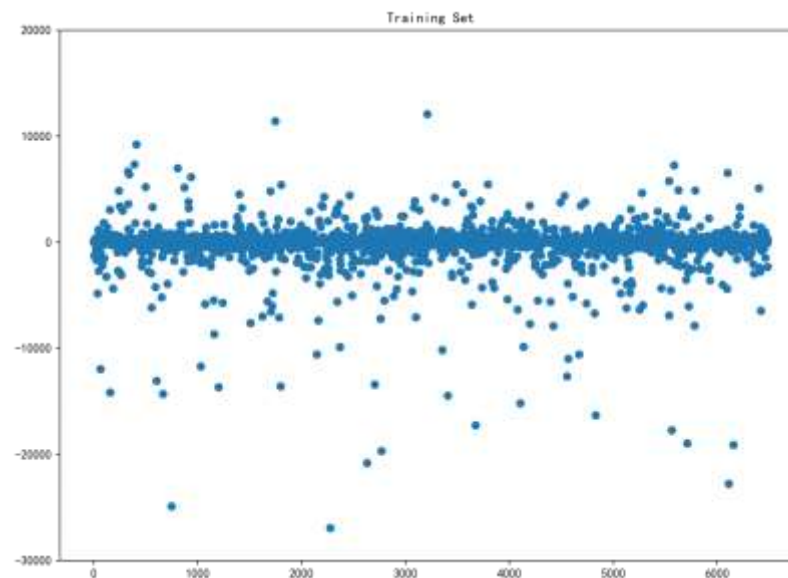
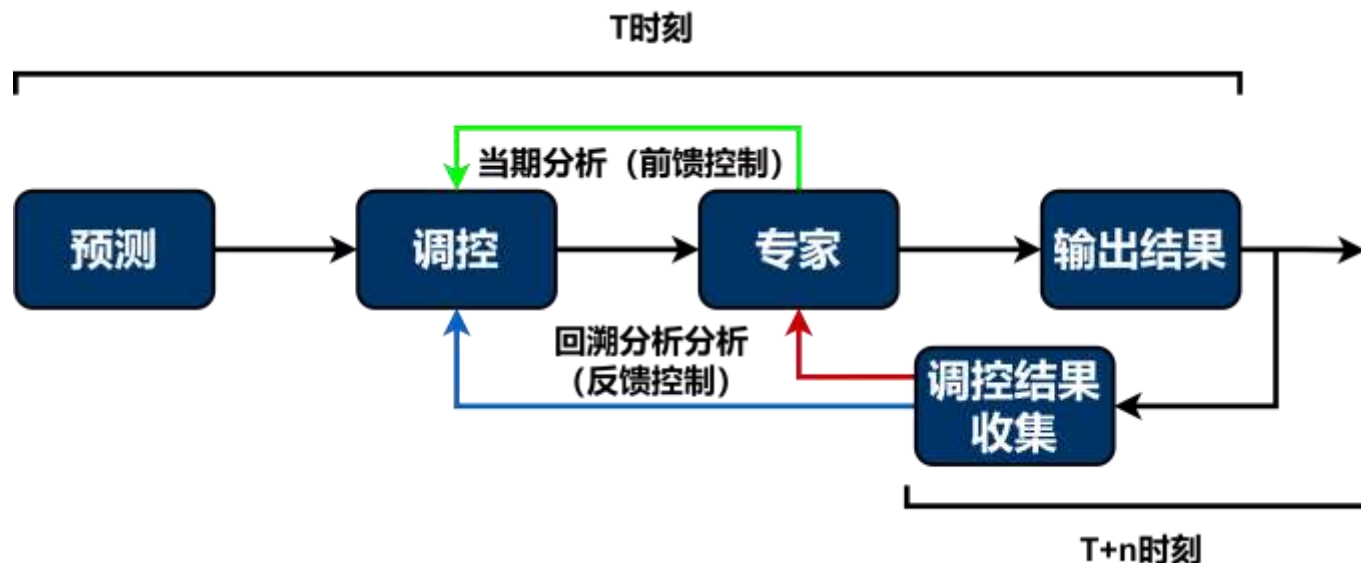
■ 反馈控制（调控）



□ 稀疏场景下的可信调控

■ 前馈控制（调控）

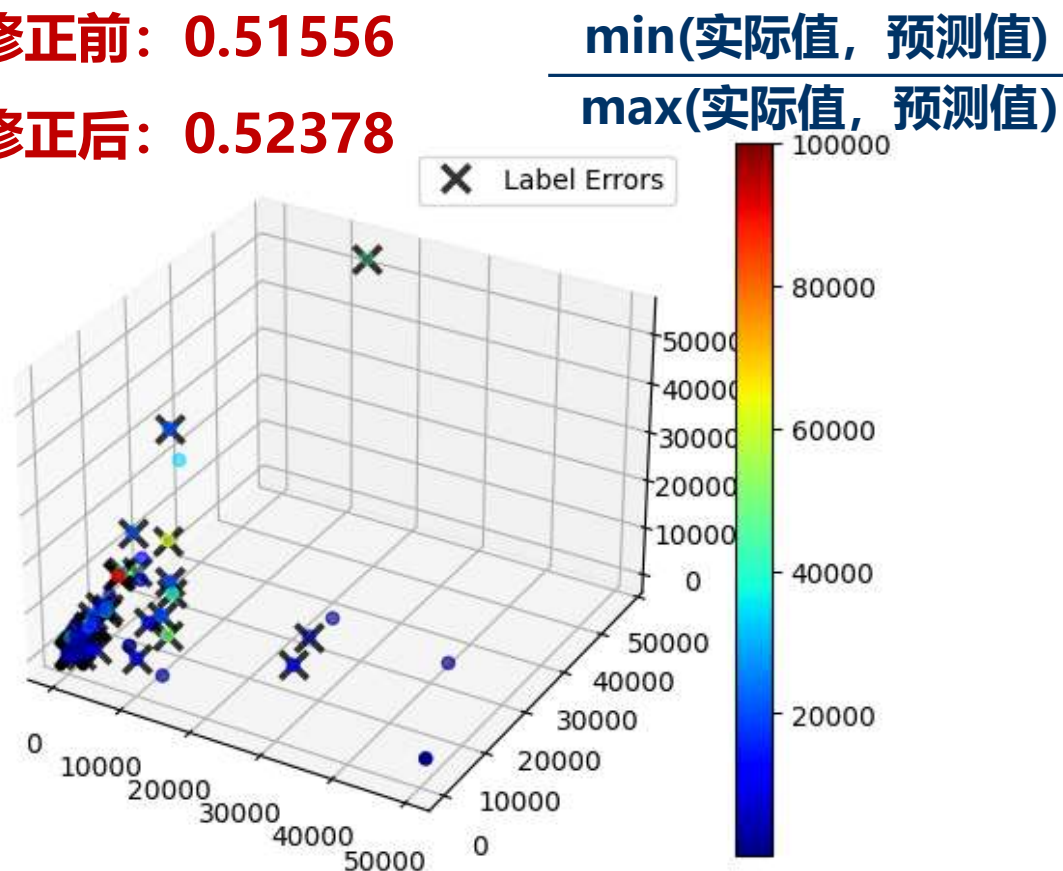
- 专家当期调控输入包括：当前的前置调控值、专家调控值、两个组件（专家的噪声与调控）解耦
- 以信度函数理论为例
- 使用信度质量函数（Belief Mass Function）量化当前专家调控值的信念
- 基于Dempster-Shafer（DS）理论，进行调控值的合成



□ 标签可信度与专家调控修正

■ 修正前: 0.51556

■ 修正后: 0.52378



Zhou, H., Mueller, J., Kumar, M., Wang, J. L., & Lei, J. (2023). Detecting Errors in Numerical Data via any Regression Model. *arXiv preprint arXiv:2305.16583*.

	is_label_issue	label_quality	given_label	predicted_label	True_label
0	False	0.845621	45000	51688	56317
1	False	0.514259	32900	62435	32300
2	False	0.569622	10000	37232	23303
3	True	0.082217	90000	13550	19929
4	False	0.978394	20000	20705	33631
5	False	0.870964	40000	33357	18444
6	False	0.544328	50000	75230	20554
7	False	0.760247	40000	32436	17696
8	False	0.728584	21000	13259	10845
9	True	0.227623	100000	37230	10348
10	False	0.940166	20000	18447	9209
11	False	0.616875	10000	21339	7207
12	False	0.759289	9000	19440	11895
13	False	0.808650	20000	27566	8443
14	True	0.399477	28000	8173	28412
15	False	0.603714	30000	17810	12796
16	True	0.147455	60000	14707	13293
17	False	0.783586	10000	1806	3530
18	False	0.993812	20000	20147	9240
19	True	0.296779	10000	75230	2747

□ 调控合成

■ 可信度最大合成

■ 专家调控修正前：
0.51556、修正后：
0.52378

■ 前置调控：0.45263

■ 合成后：0.53467

	True_label	expert_label	original_label	expert_label_quality	original_label_quality	merge_label	merge_label_maxmin
0	56317	25340	28908	0.038438	0.227039	28391.398520	28908
1	32300	27100	24947	0.145043	0.431915	25488.248491	24947
2	23303	33550	10325	0.536758	0.973145	18581.291621	10325
3	19929	38793	12551	0.293263	0.001025	38701.608879	38793
4	33631	31274	9272	0.577193	0.300318	23744.077938	31274
5	18444	3882	11218	0.349846	0.094062	5436.462340	3882
6	20554	29635	20483	0.004401	0.031028	21619.857799	20483
7	17696	33007	19104	0.364970	0.210068	27928.069618	33007
8	10845	27326	22309	0.153072	0.851869	23073.188593	22309
9	10348	4347	17936	0.483925	0.000249	4353.991232	4347
10	9209	32964	10603	0.006219	0.491970	10882.142562	10603
11	7207	32693	14046	0.030453	0.649781	14880.794317	14046
12	11895	16462	9560	0.213190	0.951029	10823.881555	9560
13	8443	31332	14272	0.165120	0.653078	17714.868944	14272
14	28412	2959	3618	0.038196	0.075156	3395.938362	3618
15	12796	8261	8659	0.000017	0.089467	8658.922236	8659
16	13293	6923	6754	0.114041	0.003820	6917.522967	6923
17	3530	3092	5140	0.993231	0.499527	3777.329721	3092
18	9240	4635	8649	0.053412	0.311533	8061.522032	8649
19	2747	22541	19574	0.331081	0.393579	20929.556133	19574

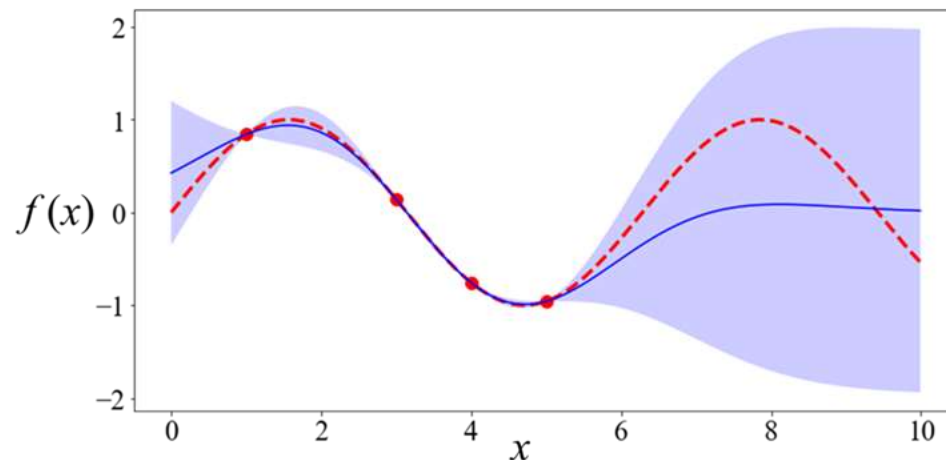
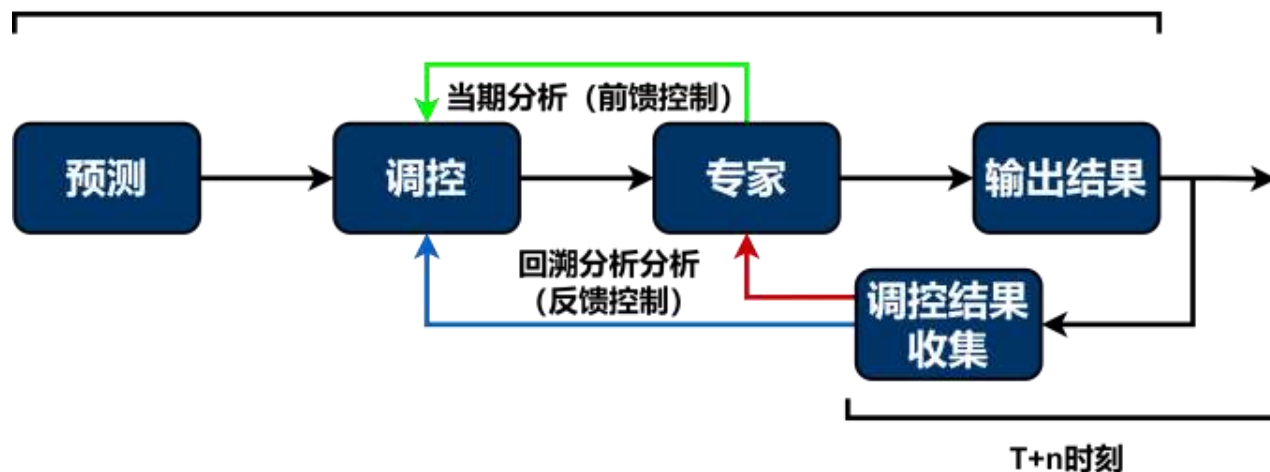
□ 稀疏场景下的可信调控

■ 可信调控输出

- 在时间节点T，输入的调控值包括：**专家当期调控值**、**回溯分析得到的调控值**
- 对于两个从专家与历史真值学习到的调控值进行**加权平均**，得到最终调控值

$$C = \sigma C_e + (1 - \sigma) C_t$$

- 通过构造代理模型，使用**贝叶斯优化框架**基于调控真值确定权重



Huo, J., Liu, Z.*, Chen, J., Cheng, Q., and Meng, Q., 2023. Bayesian Optimization for Congestion Pricing Problems: A General Framework and Its Instability, *Transportation Research Part B*, 169, 1-28.

□ 稀疏场景下的可信调控

- 考虑模型可信度的贝叶斯优化框架
- 专家模型在不同的环境下的可信度 $B_{m_1}(x_i)$ 是不一样的，基于贝叶斯优化融合当期与回溯分析的调控输出值，做到可信与精准调控

$$\min_{\sigma} E_{m_1, m_2} [l(\sigma, B_{m_1}(x), x, y_i, m)]$$

s.t.

$$0 \leq \sigma \leq 1$$



感谢您的倾听
敬请批评指正



東南大學
SOUTHEAST UNIVERSITY

