2022.5.5

LSTM与lightGBM基准预测

数据输入

训练集的时间跨度为756天（27*28），将其分成27个不重叠的时间窗。对于每个时间窗，时间窗自身用于计算预测误差，其对应的输入特征x如下：

- 价格特征：原始价格、相对同cat内的最大值的比例、相对同dept内的最大值的比例；
- 日历特征：日期在一年中的比例、星期几（可表示成在一周的比例）、距开售第一天的距离
- 销售特征：原始销量28期滞后、7期移动平均（28期滞后的）、28期移动平均（28期滞后的）
- 节假日（event name 31类，嵌入后6类；event type5类，嵌入后2类）
- SNAP
- 与间断相关的特征：上两个正销售额间的距离、距上一个正销售的距离
- id：store（10--4）、state（3--2）、cat（3--2）、dept（7--3）、item（3049--30）的id

模型设置

LSTM的网络设置与上次汇报相同，**嵌入层——Dropout——LSTM层——Dropout层——线性层**，输出负二项分布的两个参数，分位数预测通过计算分布的分位数得到（用所求分布抽取1000个数据，计算样本分位数）。

lightGBM使用python的 `lightgbm` 包实现，输入特征与LSTM相同，预测时逐期输入特征逐期预测。其支持对pinball loss进行优化。

基准预测结果与组合结果

基准预测

以下是各方法基准预测的对比（在元学习的test set上）：LSTM与LightGBM并没有经过调参处理，预测结果并不是很突出，没有太多的完全占优次数，在各个分位数预测预测的结果都属于较好，但不能好过nb_undamped 这种只依赖历史销量数据的统计方法结果。可能原因是：需要调参；使用数据的时间范围需要扩容（但是仅考虑近两年数据的原因主要在于个人电脑的内存限制）；网络结构等。但不排除就是nb_undamped在此数据上较好的可能性（因为自己不好实现M5排名靠前选手的方法）。

这两种基准方法是否还需要继续优化?

分位数	0.01	0.025	0.165	0.25	0.5	0.75	0.835	0.975	0.99	平均秩
quantGAM	0.0146	0.0368	0.2336	0.3378	0.5638	0.5554	0.4769	0.1592	0.0887	8.22
VZ	**0.0143**	**0.0356**	0.2275	0.3289	0.5374	0.5320	0.4551	0.1454	0.0785	4.83
WSS	**0.0143**	**0.0356**	0.2270	0.3322	0.5740	0.5672	0.4745	0.1520	0.0754	5.61
poisson_static	0.0409	0.0706	0.2692	0.3631	0.5448	0.5434	0.4786	0.2146	0.1491	10.22
poisson_damped	0.0243	0.0495	0.2347	0.3243	0.4974	0.4926	**0.4279**	0.1683	0.1084	6.44
poisson_undamped	0.0249	0.0497	0.2322	0.3201	**0.4915**	**0.4906**	0.4306	0.1880	0.1318	6.33
nb_static	0.0144	0.0358	0.2295	0.3382	0.5878	0.6205	0.5292	0.1498	**0.0759**	7.11
nb_damped	0.0150	0.0366	0.2206	0.3179	0.5172	0.5129	0.4428	0.1553	0.0839	5.33
nb_undamped	0.0144	**0.0356**	**0.2164**	**0.3105**	0.4958	0.4979	0.4359	0.1681	0.1043	**3.61**
LSTM	0.0144	0.0358	0.2206	0.3178	0.5172	0.5167	0.4426	**0.1444**	0.0760	3.72
lgb	0.0144	0.0361	0.2260	0.3174	0.5094	0.5167	0.4477	0.1507	0.0815	4.56

组合结果

根据之前的经验，使用固定效应惩罚系数为50的分位数回归进行回归组合，结果如下：

分位数	0.01	0.025	0.165	0.25	0.5	0.75	0.835	0.975	0.99
面板回归（11模型）	无	0.6167	**0.2150**	**0.3064**	**0.4890**	**0.4892**	**0.4204**	0.1340	0.0695
加权平均（11模型）	0.0155	0.0376	0.2210	0.3145	0.5035	0.5003	0.4269	0.1342	0.0696
简单平均（11模型）	0.0171	0.0389	0.2208	0.3147	0.5046	0.5010	0.4270	**0.1337**	**0.0694**
面板回归（9模型）	**0.0143**	**0.0356**	0.2168	0.3091	0.4915	0.4897	0.4217	0.1351	0.0702
面板回归（10模型）	**0.0143**	**0.0356**							

- 在0.975、0.99右侧极端分位数上简单平均最好，但差距并不大；
- 中间分位数组合方法最好；
- 左侧两个极端分位数的回归组合属于异常情况：0.025运行代码时报 singular matrix 警告，预测系数不正常；而0.01运行代码时触发 fatal error。可能原因是，新引入的两个方法在0.01分位数与0.025分位数的预测结果基本均是取0，引起了矩阵奇异的问题。

一个权宜处理方法：将0.025、0.01的分位数组合仅引入LSTM模型，结果改善。

组合系数结果如下：

分位数	0.01	0.025	0.165	0.25	0.5	0.75	0.835	0.975	0.99
截距	~	0.9716	-0.0008	0.0000	0.0002	0.0047	-0.0090	0.1329	0.1566
quantGAM	~	0.6030	0.0004	0.0002	0.0095	0.0409	0.0779	0.1231	0.1325
VZ	~	0.0811	0.0161	0.0111	0.0086	0.0321	0.1095	0.1280	0.0684
WSS	~	-0.0759	0.2143	0.1702	0.0291	0.0066	0.0075	0.0548	0.0668
poisson_damped	~	0.4835	-0.0241	-0.0116	-0.0087	-0.0358	-0.1015	-0.1819	-0.1920
poisson_undamped	~	-0.1566	-0.0012	0.0000	0.0750	0.1650	0.1873	0.1884	0.1930
poisson_static	~	1.2360	0.0631	0.2437	0.5476	0.5957	0.4526	0.1044	0.0721
nb_static	~	-1.8556	0.0976	0.0060	-0.0397	-0.0094	-0.0118	0.0291	0.1112
nb_damped	~	0.0530	0.0259	0.0182	0.0048	0.0007	0.0131	0.0438	0.0326
nb_undamped	~	-0.3129	0.3653	0.1993	0.0012	-0.0040	0.0032	0.0564	0.0604
lstm	~	1.2869	0.5046	0.3303	0.2165	0.0679	0.1104	0.0929	0.0675
lgb	~	0.0000	0.2697	0.2226	0.1164	0.1295	0.1502	0.2563	0.2676

只10模型的系数修改：

分位数	0.01	0.025	0.165	0.25	0.5	0.75	0.835	0.975	0.99
截距	0.0000	-0.0008							
quantGAM	0.0000	0.0001							
VZ	0.6147	0.0720							
WSS	0.5442	0.8208							
poisson_damped	0.0000	0.0000							
poisson_undamped	0.0000	-0.0005							
poisson_static	0.0000	0.0004							
nb_static	0.0000	0.0017							
nb_damped	0.0000	0.0018							
nb_undamped	0.0000	0.0864							
lstm	0.0000	0.6115							
lgb									

今后计划

尝试基于LSTM的时变分位数组合：

- 考虑时变权重与截距引入的问题；
- 使用特征的问题：自选特征或者参考 Ma & Flides (2021) 的设置，使用基于网络提取的特征。