基于图神经网络、图谱型数据的收益预测模型（附代码）

Original 全网Quant都在看 量化投资与机器学习 2021-07-26 17:14

收录于合集
#深度研读系列 21个 >

量化投资与机器学习公众号独家解读

量化投资与机器学公众号 ***QIML Insight——深度研读系列*** 是公众号今年全力打造的一档**深度、前沿、高水准**栏目。

公众号**遴选**了各大期刊前沿论文，按照理解和提炼的方式为读者呈现每篇论文最精华的部分。QIML希望大家能够读到可以成长的量化文章，愿与你共同进步！

第一期 | 第二期 | 第三期 | 第四期 | 第五期 | 第六期
第七期 | 第八期 | 第九期 | 第十期 | 第十一期

前言

传统的股价预测的时许模型，对于收益率的假设往往不切实际，而最近兴起的机器学习模型，特别是深度学习模型对于股价的预测也存在着明显的问题：

- **大多数文献中，都直接预测股价或者是收益率，并没有考虑股票之间的排序。**
- **每只股票的序列单独输入模型中，并不能考虑股票或公司间多维度的信息：比如供应关系和产业关系。**

为了解决以上问题，作者提出了一个新的框架：Relational Stock Ranking(RSR)。这个框架主要由两个创新：

- **损失函数新增了关系股票收益排序的惩罚项，使模型能够顾及股票收益间的排序（Rank）。**
- **结合了时间图神经网络，使模型能够结合股票间的关系型数据，如行业属性、上下游、股权信息等。**

RELATIONAL STOCK RANKING（RSR）

RSR总共包含三层，分别是基于序列数据应用序列神经网络模型的Sequential Embedding Layer, 基于关系型数据应用图神经网络的Relational Embedding Layer, 和最终给出收益预测结果的Prediction Layer。如下图所示：

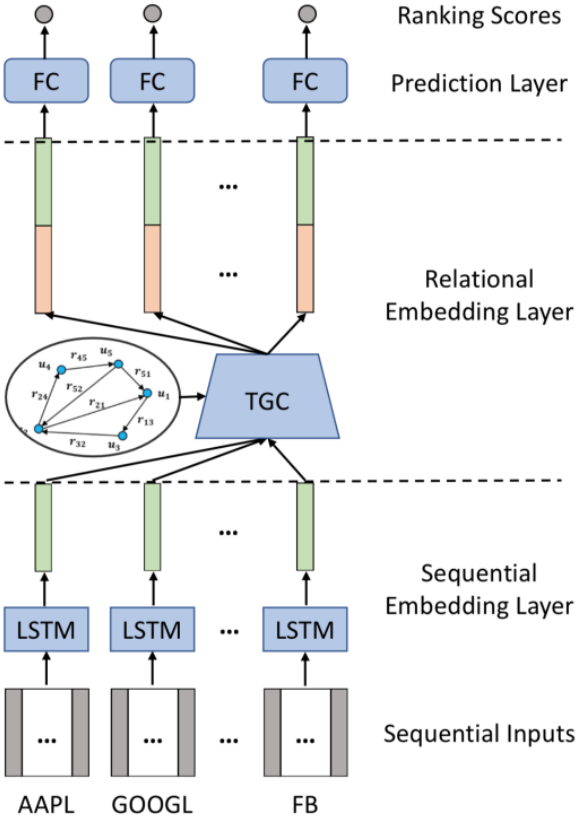


Fig. 1. Relational stock ranking framework. It should be noted that the LSTMs and FC units (Fully Connected layer) depicted in the same layer share the same parameters.

Table 2. Terms and notations.

Symbol	Definition
$\mathcal{X}^t \in \mathbb{R}^{N\times S\times D} = [\mathbf{X}_1^t, \cdots, \mathbf{X}_N^t]^T$	historical prices of N stocks on trading day t.
$\mathcal{A} \in \mathbb{R}^{N\times N\times K}$	binary encoding of stock relations.
$\mathbf{E}^t = [\mathbf{e}_1^t, \cdots, \mathbf{e}_N^t]^T \in \mathbb{R}^{N\times U}$	sequential embedding of N stocks learned from historical prices.
$\overline{\mathbf{E}^t} = [\mathbf{e}_1^t, \cdots, \mathbf{e}_N^t]^T \in \mathbb{R}^{N\times U}$	relational embedding of all stocks learned from $\mathbf{E}^t$ and $\mathcal{A}$.
$\mathbf{r}^{t+1}, \hat{\mathbf{r}}^{t+1} \in \mathbb{R}^N$	ground-truth and predicted ranking scores of N stocks.
$\mathbf{w}, b$	weights and bias to be learned.

Sequential Embedding Layer

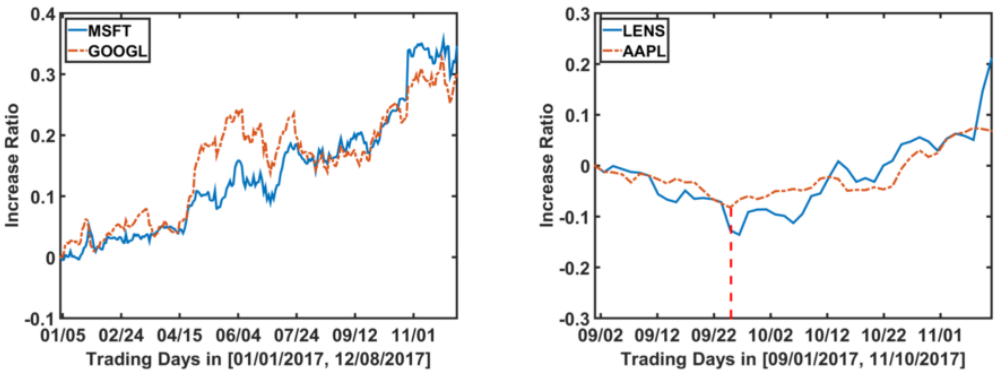
股票过去的价格变动对于未来的变化有明显的影响，所以整个框架的第一层采用序列模型去捕获股价序列间的依赖信息。RNN在最近的文献中，都有出色的表现。所以作者选用的RNN最为第一层的模型。更具体的，作者采用了LSTM，因为它能保留序列的长期记忆。并以LSTM的模型结果作为下一层的输入。（LSTM的输入为股价的历史序列X_t）

Relational Embedding Layer

这一层主要考虑股票之间的关系型数据，作者在模型中加入了两类关系型数据：

行业属性：两个公司是否属于同一个行业或板块，如果属于同一行业，那么两个公司之间的基本业务应该类似，股价的表现也应该有相似的趋势，如图2a中，MSFT和GOOGL的股价。

供应链关系：如果两家公司属于供应链的上下游，即一家公司是另一家的客户（或供应商），那么他们之间的股价的变动应该有传导效应。如图2b中，Apple的供应商LENS在AAPL发布iphone8之后，股价开始上涨。



(a) Sector-industry relation (b) Supplier-consumer relation

Fig. 2. Two examples of stock price history (normalized as increase ratio as compared to the first depicted trading day) to illustrate the impact of company relations on the stock price.

为了使RSR能够加入这些关系型的数据，作者采用了Temporal Graph Convolution(TGC)算法，将关系型的图谱数据与第一层的输出进行结合，作为第三层的输入。关于TGC，下一节会详细介绍。

Prediction Layer

最终将第二层的输入用于股票收益率的预测，这里的损失函数定义为：

$$l\left(\hat{\mathbf{r}}^{t+1}, \mathbf{r}^{t+1}\right) = \left\|\hat{\mathbf{r}}^{t+1} - \mathbf{r}^{t+1}\right\|^2 + \alpha \sum_{i=0}^{N} \sum_{j=0}^{N} \max\left(0, -\left(\hat{r}_i^{t+1} - \hat{r}_j^{t+1}\right)\left(r_i^{t+1} - r_j^{t+1}\right)\right)$$

其中 $\hat{r}^{t+1}, r^{t+1}$ 分别是股票的预测收益率和实际收益率。这个损失函数中，第一项为了是预测的收益误差越小越好理解。第二项，为了股票间的相对顺序与真实情况比，误差越小越好。

Temporal Graph Convolution

给定N个股票的序列特征（sequential embeddings）$E_t \in R^{N \times U}$ 和多维度的二元关系 $A \in R^{N \times N \times K}$，TGC的主要任务是基于二元关系，重新学习N个股票的序列特征。传统的图神经网络没有考虑关系的动态变化，对于所有时间中，每个点（Vertex）的影响性都用固定的方式计算，比如如下考虑关系重要性作为权重的特征计算：

$$\overline{\mathrm{e}_{\mathrm{i}}^{\mathrm{t}}} = \sum_{\{j|\mathrm{sum}(\mathrm{a_{ji}})>0\}} \frac{g(\mathrm{a_{ji}})}{d_j} \mathrm{e}_{\mathrm{j}}^{\mathrm{t}}$$

其中，$\bar{e}_i^t$ 为股票 i，使用网络关系，整合其他与之相关的股票，而重新生成的特征，d_j 为A关系张量中与股票i有关系的股票的数量，a_{ji} 为股票 j 与股票 i 的关系，函数 g 用于度量关系的强弱。可以看到，这里的函数 g 是固定不变的，并没有考虑不同时间的序列特征，作者做了如下改进，将t时刻的股票的序列特征也考虑在关系强度函数 g 中：

$$\overline{\mathrm{e}_{\mathrm{i}}^{\mathrm{t}}} = \sum_{\{j|\mathrm{sum}(\mathrm{a_{ji}})>0\}} \frac{g\left(\mathrm{a_{ji}}, \mathrm{e}_{\mathrm{i}}^{\mathrm{t}}, \mathrm{e}_{\mathrm{j}}^{\mathrm{t}}\right)}{d_j} \mathrm{e}_{\mathrm{j}}^{\mathrm{t}}$$

此时，关系强弱函数g的定义就关系到了TGC的具体表现，作者给出了显性模型和隐性模型两种定义：

显性，其中 ϕ 为激活函数， w, b 为参数。也就是说，关系的强弱，取决于第一项，两个股票当前时间的相似性，和第二项两者关系的重要。这两项的乘积决定了关系强弱。因为这两项都能清楚的解释，所以称为显性模型。

$$g\left(\mathbf{a_{ji}}, \mathbf{e_i^t}, \mathbf{e_j^t}\right) = \underbrace{\mathbf{e_i^{t^T}} \mathbf{e_j^t}}_{\text{similarity}} \times \underbrace{\phi\left(\mathbf{w}^T \mathbf{a_{ji}} + b\right)}_{\text{relation importance}}$$

隐性，如果把序列特征及关系都放到激活函数内部，则称为隐性模型。

$$g\left(\mathbf{a_{ji}}, \mathbf{e_i^t}, \mathbf{e_j^t}\right) = \phi\left(\mathbf{w}^T \left[\mathbf{e_i^{t^T}}, \mathbf{e_j^{t^T}}, \mathbf{a_{ji}^T}\right]^T + b\right)$$

实证

作者选取了NASDAQ和NYSE从2013年1月2日至2017年12月8日的数据，经过以下条件的过滤：

- 在此区间，98%的时间正常交易
- 股价从未低于5美元

分别在NASDAQ选取了1026只股票，在NYSE选取了1737只股票，获取了日度价格数据。并把测试时间分为了3个不同的阶段，如下表所示：

Table 3. Statistics of the sequential data.

Market	Stocks#	Training Days# 01/02/2013 12/31/2015	Validation Days# 01/04/2016 12/30/2016	Testing Days# 01/03/2017 12/08/2017
NASDAQ	1,026	756	252	237
NYSE	1,737	756	252	237

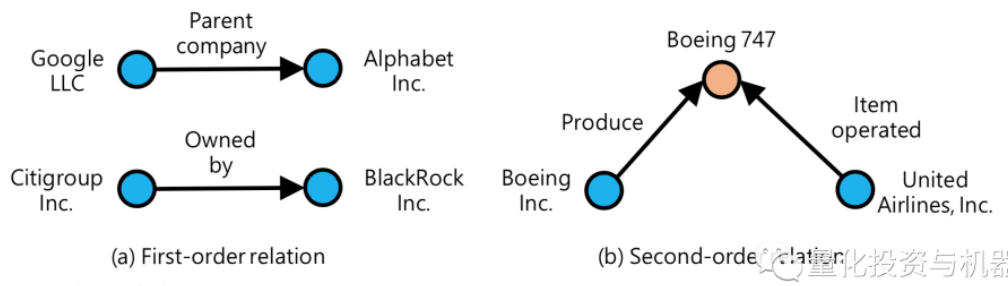
除了价格数据，还有股票的关系数据，包括行业属性数据及关系数据（总计42中关系），覆盖度如下：

Table 4. Statistics of sector-industry relation and Wiki relation data in the NASDAQ and NYSE datasets.

	Sector-Industry Relation		Wiki Relation	
	Relation Types#	Relation Ratio (Pairwise)	Relation Types#	Relation Ratio (Pairwise)
NASDAQ	112	5.00%	42	0.34%
NYSE	130	9.37%	32	0.30%

关系型数据，主要分为，如图4表示：

- 一阶关系：两者直接相关
- 二阶关系：两者分别和第三者相关



(a) First-order relation (b) Second-order relation

Fig. 4. Examples of the first-order and second-order company relations extracted from Wikidata.

回测设定

在2017/01/03至2017/12/08的测试区间，回测的设定如下：

- 买入预测收益最高的股票，日度换仓，收盘买卖
- 每次固定50,000美元的持仓规模
- 交易成本为0，假设全部成交

用于对比的其他模型

- SFM：首先将历史价格数据进行离散傅里叶变换，再输入到LSTM模型中进行预测。参考Liheng 2017；
- LSTM：基于历史收盘价及5、10、20和30均线的LSTM模型；
- RANK_LSTM：RSR中去除Relational Embedding Layer后的模型；
- Graph-base Ranking(GBR)：RANK_LSTM的损失函数增加图惩罚项；
- GCN：RSR的第二层采用GCN；
- RSR_E：RSR中的关系强弱函数g，用显性模型；
- RSR_I：RSR中的关系强弱函数g，用隐性模型。

衡量模型效果的指标有：

- Mean Square Error (MSE)，越小越好；
- Mean Reciprocal Rank(MRR)：是一个国际上通用的对搜索算法进行评价的机制，即第一个结果匹配，分数为1，第二个匹配分数为0.5，第n个匹配分数为1/n，如果没有匹配的句子分数为0。总之，越大越好。
- the cumulative investment return ratio (IRR)：越大越好。

最关心的问题

1、将股票的价格预测问题变成一个收益预测的任务，效果怎么样？相比当下热门的算法，RSR有没有优势？

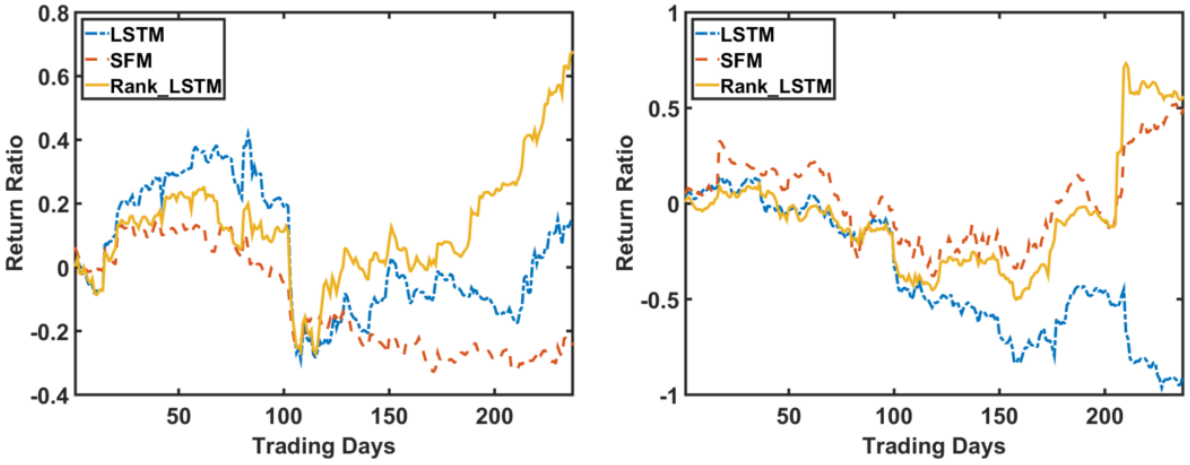
Table 5. Performance comparison between the solutions with regression formulation (**SFM** and **LSTM**) and ranking formulation (**Rank_LSTM**).

	NASDAQ			NYSE		
	MSE	MRR	IRR	MSE	MRR	IRR
SFM	5.20e-4±5.77e-5	2.33e-2±1.07e-2	-0.25±0.52	3.81e-4±9.30e-5	**4.82e-2±4.95e-3**	0.49±0.47
LSTM	3.81e-4±2.20e-6	3.64e-2±1.04e-2	0.13±0.62	2.31e-4±1.43e-6	4.59e-2±1.99e-2	0.99±0.73
Rank_LSTM	**3.79e-4±1.11e-6**	**4.17e-2±7.50e-3**	**0.68±0.60**	**2.28e-4±1.16e-6**	3.79e-2±8.82e-3	**0.56±0.68**

- 以IRR指标衡量，Rank_LSTM的效果比SFM和LSTM好很多，说明基于股票排序的学习比直接预测收益率，效果更优。这对我们研究更先进的排序学习算法(Learning-to-rank)有了信心。

- 但在MRR指标上，在NYSE市场，Rank_LSTM的表现差于SFM，可能是因为损失函数及考虑了绝对预测的准确性，又考虑了相对排序的准确性，从而降低了模型的稳定性。

- 图5给出了三个模型的累计收益曲线，可以看出仅仅买入一只股票的收益波动还是很大的，模型的表现不够稳定。



(a) NASDAQ (b) NYSE

Fig. 5. Performance comparison of **Rank_LSTM**, **SFM**, and **LSTM** regarding IRR.

2、股票间的关系数据能否提高神经网络模型的效果？文中提出的TGC与卷积图神经网络算法GCN相比有效性怎么样?

Table 6. Performance comparison among relational ranking methods with industry relations.

	NASDAQ			NYSE		
	MSE	MRR	IRR	MSE	MRR	IRR
Rank_LSTM	**3.79e-4±1.11e-6**	4.17e-2±7.50e-3	**0.68±0.60**	2.28e-4±1.16e-6	3.79e-2±8.82e-3	0.56±0.68
GBR	5.80e-3±1.20e-3	**4.46e-2±5.20e-3**	0.57±0.29	2.29e-4±2.02e-6	3.43e-2±6.26e-3	0.68±0.31
GCN	3.80e-4±2.24e-6	3.45e-2±8.36e-3	0.24±0.32	2.27e-4±1.30e-7	**5.01e-2±5.56e-3**	0.97±0.56
RSR_E	3.82e-4±2.69e-6	3.16e-2±3.45e-3	0.20±0.22	2.29e-4±2.77e-6	3.23e-2±6.18e-3	0.63±0.58
RSR_I	3.80e-4±7.90e-7	3.17e-2±5.09e-3	0.23±0.27	**2.26e-4±5.30e-7**	4.51e-2±2.41e-3	**1.06±0.27**

表6的测试中，关系型数据仅仅使用了行业属性数据：

- 加入行业属性的数据后，在NYSE的表现比NASDAQ的表现更好，可能是因为NASDAQ的股票波动更大，更受短期因素的影响；

- 在NYSE，所有加入图关系数据的模型的IRR都比Rank_LSTM来的好，说明关系型数据能增强模型的表现；
- RSR_E，RSR_I的表现优于GCN和GBR。说明，TGC相比GCN的效果更佳。

Table 7. Performance comparison among relational ranking methods with Wiki relations.

	NASDAQ			NYSE		
	MSE	MRR	IRR	MSE	MRR	IRR
Rank_LSTM	**3.79e-4±1.11e-6**	4.17e-2±7.50e-3	**0.68±0.60**	2.28e-4±1.16e-6	3.79e-2±8.82e-3	**0.56±0.68**
GBR	3.80e-4±2.40e-7	3.32e-2±4.50e-3	0.33±0.34	**2.26e-4±4.20e-7**	3.64e-2±5.35e-3	0.65±0.27
GCN	**3.79e-4±9.70e-7**	3.24e-2±3.21e-3	0.11±0.06	**2.26e-4±6.60e-7**	3.99e-2±1.03e-2	0.74±0.30
RSR_E	3.80e-4±7.20e-7	3.94e-2±8.15e-3	0.81±0.85	2.29e-4±2.77e-	2.3e-2±6.18e-3	**0.95±0.47**
RSR_I	**3.79e-4±6.60e-7**	**4.09e-2±5.18e-3**	**1.19±0.55**	**2.26e-4±1.37e-6**	4.58e-2±5.55e-3	0.79±0.34

如表7所示，在考虑Wiki关系型数据后，RSR_E和RSR_I在两个市场的IRR都是最高的。下表展示了，Wiki关系型数据中最重要的5个关系，**其中P1056_P1056表示两个公司是否生产同一种产品，这个关系是最重要的，也是公司的产业链关系，可以用产业图谱表示。**

Table 9. Impacts of different types of Wiki relation regarding the Relative Performance Decrease (**RPD**) of **RSR_I** on NASDAQ as removing the selected relation.

Relation	P1056_P1056	P463_P463	P452_P452	P361_P361	P1056_P452
ID in Table 14	46	R38	35	31	45
#Occurrences	130	58	506	1,1	19
RPD	-144.00%	-70.13%	-21.66%	-17.52%	-15.71%

总结

- 前段时间，JPM有很多文章介绍Learning-to-rank的算法，本篇文章虽然没有之间采用Learning-to-rank的算法，但在损失函数设计中巧妙的考虑了股票间的排序。
- 产业链、供应链等图谱型数据，日益成为大家关注的数据类型，但很多机构并没有想好怎么用这类数据？传统基于量价的深度学习模型，结合基于图谱类数据的图神经网络，给这类数据的应用指明了一条可探索的道路。

资源

作者在github上开源了论文的代码，心急的小伙伴尽快尝鲜：

https://github.com/fulifeng/Temporal_Relational_Stock_Ranking

收录于合集 #深度研读系列 21

< 上一篇
从『Man VS AI』到『Man + AI』

下一篇 >
基于Order Book的深度学习模型：预测多时间段收

益序列

People who liked this content also liked

北大满哥与奥迪的罗生门

量化投资与机器学习