

量化投资

上海鸣熙资产管理有限公司

合伙人： 吴比亚

内容

- ◆ 量化投资的概况
- ◆ 流程与方法论
- ◆ 组合优化研究
- ◆ 机器学习的应用
- ◆ 行业未来的思考
- ◆ 彩蛋



◆ 量化投资的概况

买方 (BUY SIDE) VS 卖方 (SELL SIDE)

买方

- 进行投资决策
- 主要机构:

公募基金: BlackRock、 Vanguard、 Fidelity

对冲基金: Renaissance Technologies、 Point 72、
Citadel

卖方

- 提供交易支持和投资建议
- 主要机构:

Goldman Sachs、 J.P. Morgan、 Morgan
Stanley, Citigroup

一些历史上的传奇投资者

Investor	Key Fund/Vehicle	Period	Annualized Returns*
Jim Simons	Medallion Fund	1988–2018	39.1%
George Soros	Quantum Fund	1969–2000	32%*
Steven Cohen	SAC	1992–2003	30%
Peter Lynch	Magellan Fund	1977–1990	29%
Warren Buffett	Berkshire Hathaway	1965–2018	20.5%*
Ray Dalio	Pure Alpha	1991–2018	12%



◆ 量化投资方法论

量化投资方法



数据收集与清洗

- 来源：量价数据、财务数据、另类数据
- 处理：缺失值、异常值等

Alpha信号

- 统计方法：回归分析、机器学习、人工智能
- 指标：夏普比率、信息比率

投资组合优化

- 框架：大多基于Markowitz portfolio theory
- 约束：风险、流动性、交易成本

交易与执行

- 算法：VWAP、TWAP
- 监控：滑点



◆ 组合优化研究

组合优化

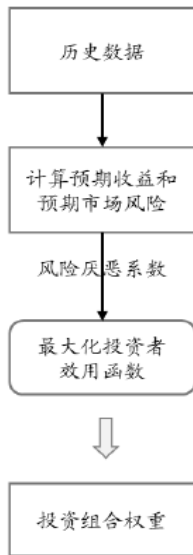
效用函数 1: $\max_w U = w^T \mu - \frac{\lambda}{2} w^T \Sigma w$

效用函数 2: $\max_w w^T \mu \quad s.t. \quad w^T \Sigma w \leq \sigma_{max}^2$

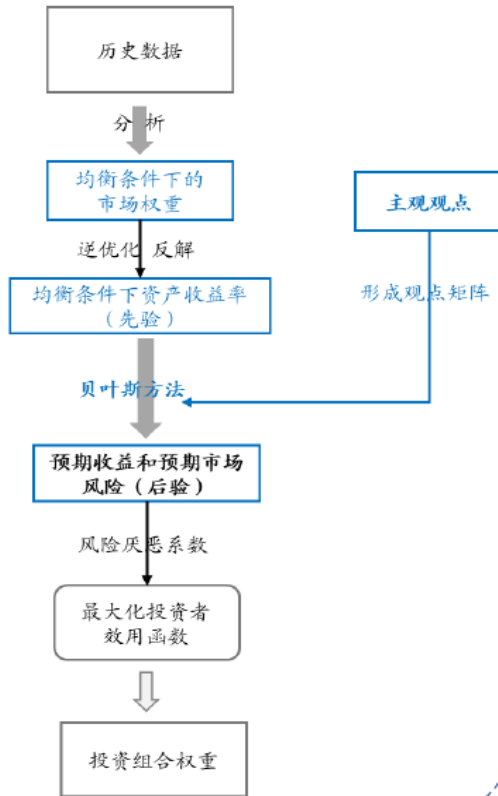
$$w^* = \frac{1}{\lambda} \Sigma^{-1} \mu = \left(\sqrt{\frac{\sigma_{max}^2}{\mu^T \Sigma^{-1} \mu}} \right) \Sigma^{-1} \mu, \quad \lambda = \left(\sqrt{\frac{\sigma_{max}^2}{\mu^T \Sigma^{-1} \mu}} \right)^{-1}$$

组合优化

均值-方差模型



Black-Litterman模型



组合优化

$$U = w^T \Pi - \frac{\lambda}{2} w^T \Sigma w$$

其中：

U: 投资者效用，也是投资组合优化的目标函数

w: 投资组合中每个资产的权重

Π : 市场均衡条件下每个资产的超额收益

λ : 市场的风险厌恶系数（代表市场的平均风险厌恶程度）

Σ : 各个资产间的协方差矩阵(超额收益协方差矩阵相同)

$$\Pi = \lambda \Sigma \omega$$

组合优化

BL模型将投资者关于资产收益率的主观观点作为输入变量，投资者的主观观点可以通过 P 、 Q 和 Ω 的形式进行设定。假定投资者有对 n 个资产具有 k 个观点，并且投资者的这些观点可以用如下方式表示：

- a) P ：资产观点矩阵。
- b) Q ：观点收益矩阵。
- c) Ω ：是观点误差的协方差矩阵。 Ω 是一个 $k \times k$ 矩阵。

$$P * \mu = Q + v$$

组合优化

主观观点举例说明。我们假定投资者在当期对于沪深 300、标普 500、恒生指数和国债指数四个资产有如下的三个观点：

未来一个月沪深 300 的收益率为-3%。观点误差水平为 ω_1 。

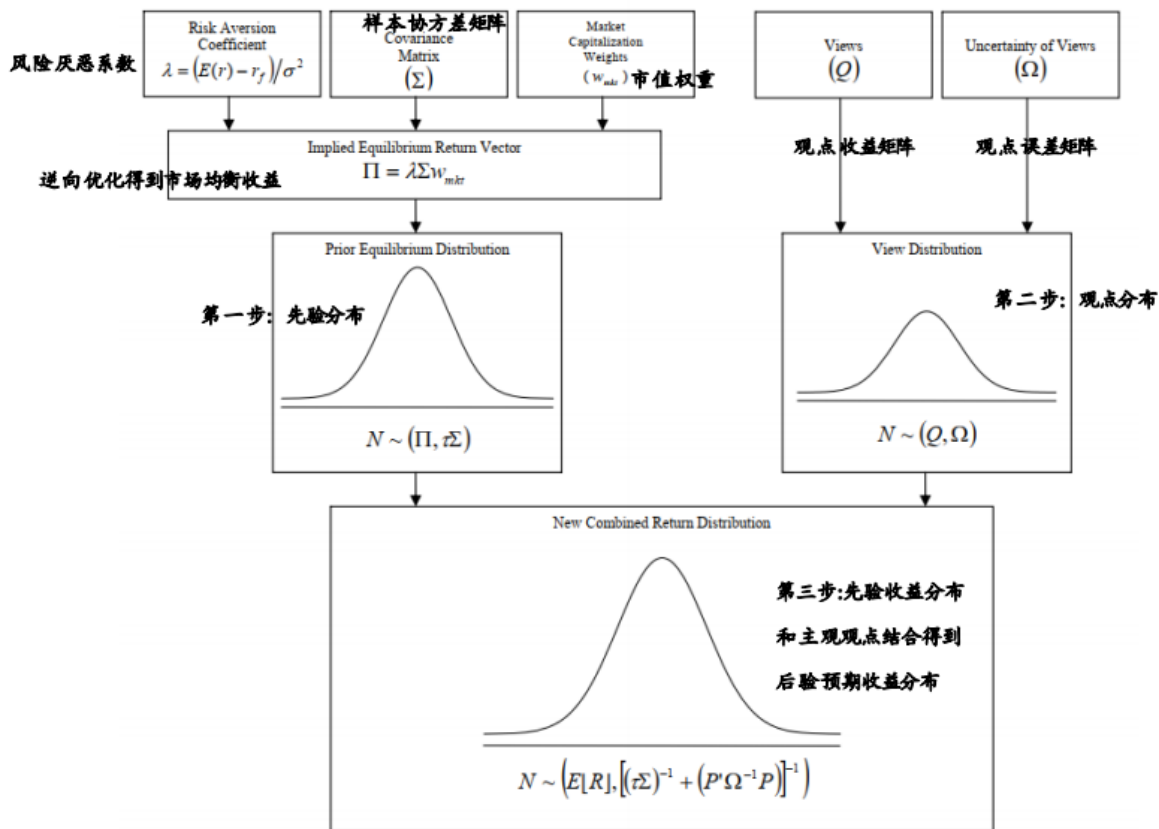
未来一个月沪深 300 表现优于美股 1%。观点误差水平为 ω_2

未来一个月国债指数收益率为 0.5%。观点误差水平为 ω_3 。

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Q = \begin{bmatrix} -0.03 \\ 1 \\ 0.005 \end{bmatrix}$$

$$\Omega = \begin{bmatrix} \omega_1 & 0 & 0 \\ 0 & \omega_2 & 0 \\ 0 & 0 & \omega_3 \end{bmatrix}$$



* The variance of the New Combined Return Distribution is derived in Satchell and Scowcroft (2000).

$$\hat{\Pi} = E(R) = [(\tau \Sigma)^{-1} \Pi + P^T \Omega^{-1} Q][(\tau \Sigma)^{-1} + P^T \Omega^{-1} P]^{-1}$$

BL模型的应用

由于原材料涨价带来的业绩增长预期，以及国家对新能源产业的大力支持，我们认为在2021年，基础化工、机械设备、电力设备这三个行业会分别跑赢大盘30%，20%，20%。

那么对应的，P的第一行中的1表示基础化工的股票，0代表机械设备、电力设备的股票，-1代表其余股票。第二行和第三行类似。

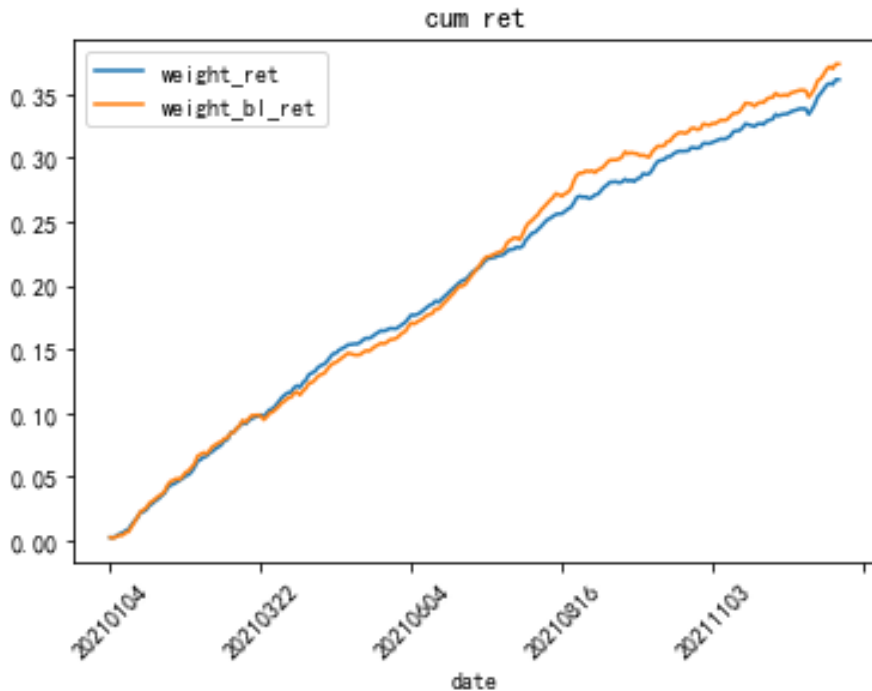
Q分别是我们认为能够跑赢的幅度。

Ω 是我们对这三个观点的不确定性。

$$P = \begin{bmatrix} 1, 1, \dots, 1, 0, 0, \dots, 0, 0, 0, \dots, 0, -1, -1, \dots, -1 \\ 0, 0, \dots, 0, 1, 1, \dots, 1, 0, 0, \dots, 0, -1, -1, \dots, -1 \\ 0, 0, \dots, 0, 0, 0, \dots, 0, 1, 1, \dots, 1, -1, -1, \dots, -1 \end{bmatrix}$$

$Q = [0.3, 0.2, 0.2]$

$\Omega = [0.01, 0.01, 0.01]$





◆机器学习的应用

图神经和神经网络概述

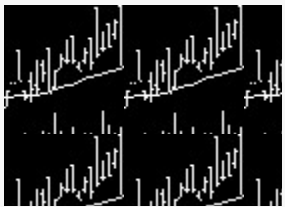
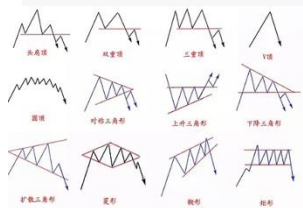
图神经网络的量化应用

图神经和神经网络概述

- ① 图分析概述
- ② Graph Attention Network
- ③ HIST
- ④ 图神经网络前沿
- ⑤ 附录

引言：为什么需要图分析

传统时序模型



时序模型忽视截面关联



传统时序模型很好，但是在纷繁复杂的金融市场，结合更多的信息才能更好的预测未来收益

图分析

行业中性化

在计算因子暴露度时，对每个行业内的股票进行调整，使得每个行业在特定因子上的平均暴露度为零。可以帮助投资者更准确地评估和利用因子效应，减少行业轮动对投资组合表现的影响，从而提高策略的稳定性和可复制性。

图神经网络

关联挖掘

因子增强

模式识别

风险评估

图聚类方法

通过分析资产间的相互作用和影响，图聚类可以帮助投资者理解不同资产之间的关联性，进而识别出潜在的投资机会或风险点，利用聚类结果，投资者可以更有针对性地选择资产，构建分散化程度更高的投资组合，降低整体风险。

图的定义

什么是图

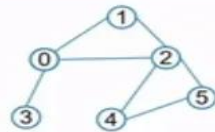


图的描述方式

图(graph)的定义

■ 图 $G = (V, E)$:

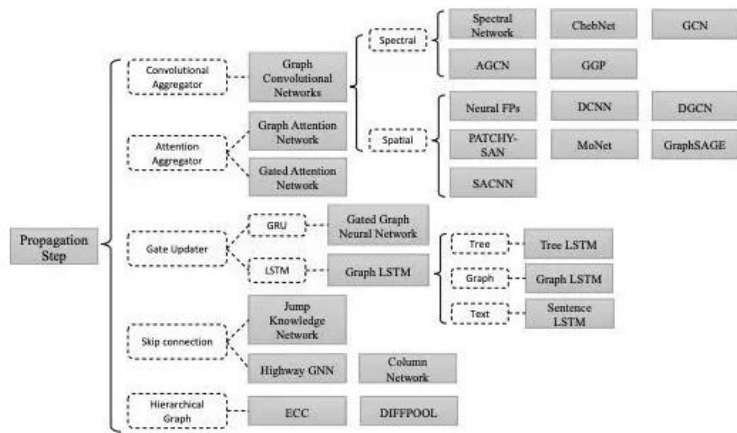
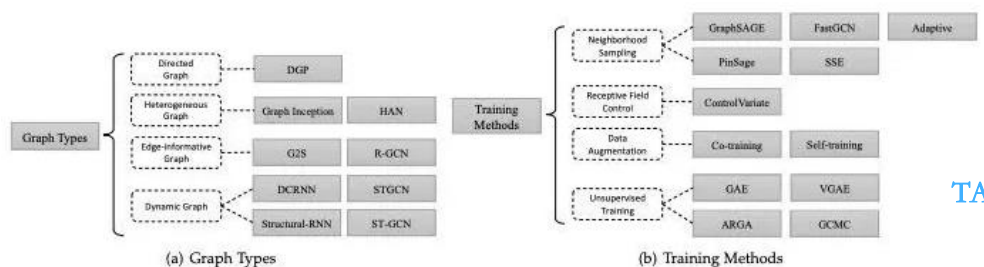
- 节点集 V ; 节点数量 $n = |V|$;
- 边集 E ; 边数量 $m = |E|$;
- 邻居矩阵 A ;
- 度矩阵 D ;
- 归一化邻接矩阵: $P = D^{-1/2}AD^{-1/2}$
- 归一化拉普拉斯矩阵: $L = I - D^{-1/2}AD^{-1/2}$
- 节点特征矩阵 $X \in \mathcal{R}^{n \times f}$, f 代表特征维度。



顶点



图神经网络



GraphConv

RelGraphConv

DNAConv

TAGConv

GatedGraphConv

SAGPooling

GATConv

GINConv

PointConv

APPNPConv

EdgeConv

GMMConv

MaxPooling

AvgPooling

图神经和神经网络概述

- ① 图分析概述
- ② **Graph Attention Network**
- ③ HIST
- ④ 图神经网络前沿
- ⑤ 附录

GAT

GCN

过度平滑



随着GCN层数的增加，节点特征可能会趋向于同质化，导致模型性能下降，这种现象被称为过度平滑

图结构依赖



GCN对图的结构比较敏感，如果图的结构信息不准确或不完整，可能会影响模型的性能。

难以并行



卷积基于全局计算，难以并行，使得原本简单的图卷积计算效率反而不如注意力机制

GAT

注意力机制



GAT引入了注意力机制，能够为图中的不同边分配不同的权重，这使得模型能够更加关注重要的邻居节点，提高了模型的表达能力和灵活性。

适应度高



GAT能够自适应地学习节点之间的关系，不需要预先知道图的结构，这使得它在处理动态图或结构不明确的图时表现出色。

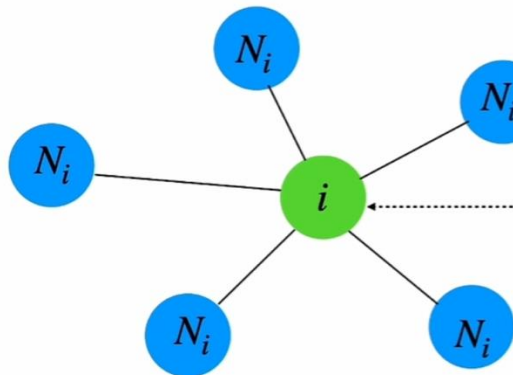
并行计算



由于注意力权重的计算是局部的，GAT可以有效地进行并行计算，提高了训练效率。

GAT原理介绍

GAT具体计算流程的公式解读



h_i →

← h'_i

• Step1

计算注意力系数

逐个计算节点 i (绿色) 与其相邻节点 (橘色节点) 的注意力系数;

$$e_{ij} = a([Wh_i] || Wh_j), j \in N_i$$

公式解读:

Wh_i 用共享参数 W 对 h_i 进行线性变换

$[. || .]$ 对节点 i 和 j 经过变换后的结果进行concat

a 把拼接后的高维特征映射到一个实数上

对注意力系数做归一化, 就是softmax

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(e_{ij}))}{\sum_{k \in N_i} \exp(\text{LeakyReLU}(e_{ik}))}$$

LeakyReLU 激活函数的一种

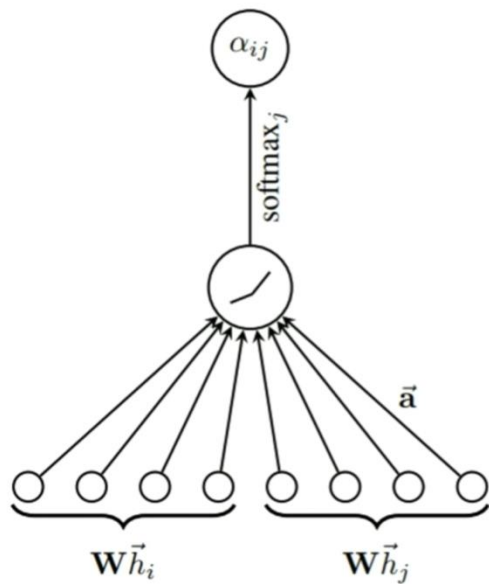
• Step2

加权求和

$$h'_i = \sigma(\sum_{j \in N_i} \alpha_{ij} Wh_j)$$

GAT原理介绍

GAT



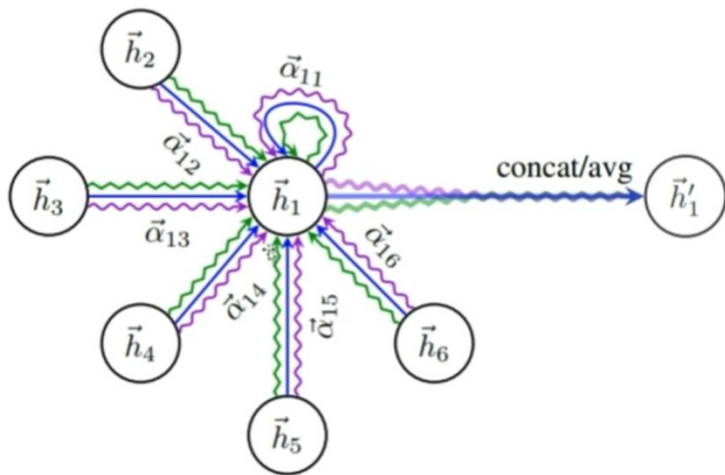
$$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$$

$$\alpha_{ij} = \underset{j \in \mathcal{N}_i}{\text{softmax}}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}.$$

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{a}^T[\mathbf{W}\vec{h}_i \parallel \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\text{LeakyReLU}\left(\vec{a}^T[\mathbf{W}\vec{h}_i \parallel \mathbf{W}\vec{h}_k]\right)\right)}$$

GAT原理介绍

GAT



$$\vec{h}'_i = \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \vec{h}_j \right).$$

$$\vec{h}'_i = \big\|_{k=1}^K \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \vec{h}_j \right)$$

$$\vec{h}'_i = \sigma \left(\frac{1}{K} \sum_{k=1}^K \sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \vec{h}_j \right)$$

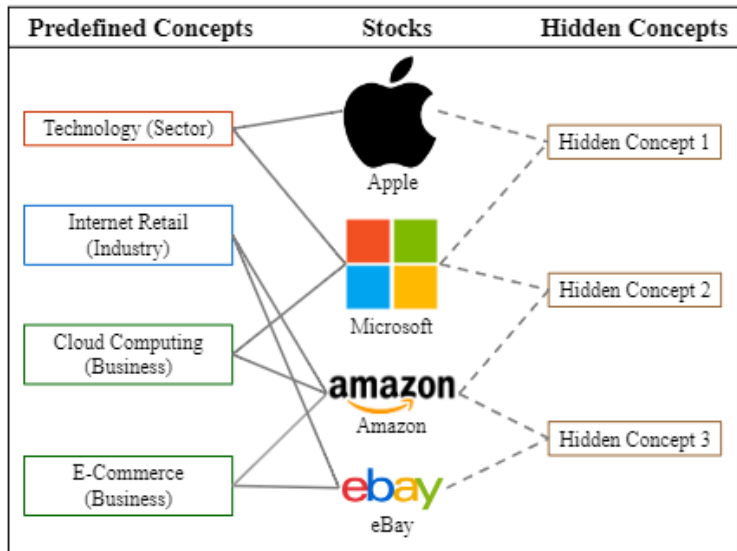
GAT vs GRU



图神经和神经网络概述

- ① 图分析概述
- ② Graph Attention Network
- ③ **HIST**
- ④ 图神经网络前沿
- ⑤ 附录

HIST: concept-oriented shared information for stock trend forecasting



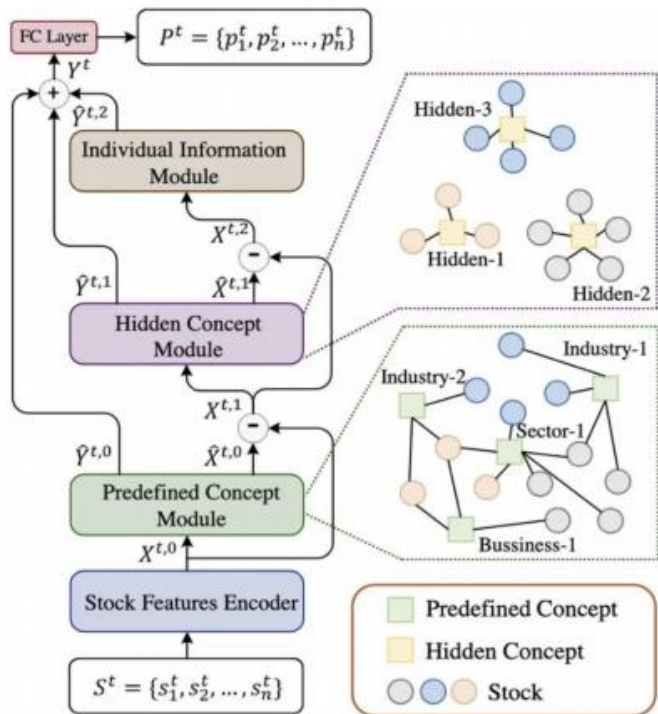
概念定义不完全

预定义信息过剩

概念内容动态变化



HIST



1. 股票特征编码器：首先，使用两层GRU网络对每个股票的特征进行编码，提取时序特征

$$x_i^{t,0} = \text{GRU}(s_i^t)$$

其中， s_i^t 是股票i在日期t的历史股票价格和成交量数据 $x_i^{t,0}$ 是股票i在日期t的初始嵌入。

2. 双重残差架构，三个信息处理模块：

在HIST的架构中为了解决预定义概念不完备的问题，用三个信息处理模块对概念信息进行加工

1. Predefined Concept Module：预定义模块
2. Hidden Concept Module：隐藏概念模块
3. Individual Information Module：个股信息模块

前两个模块的输出均包括两个部分：一部分用于最后对于股票收益率的预测，记为 $\hat{Y}^{t,0}$ 和 $\hat{Y}^{t,1}$ ；一部分用于得到融合概念信息后的股票向量表示，便于计算下一层模型的输入，记为 $\hat{X}^{t,0}$ 和 $\hat{X}^{t,1}$ 。最后一个模块的输出只有一部分，就是对于股票收益率的预测，记为 $\hat{Y}^{t,2}$ 。

第一个模块的输入为 step1 得到的股票向量表示 $X^{t,0}$ 。第二个模块的输入为初始股票向量表示 - 第一个模块输出的股票向量表示，即 $X^{t,1} = X^{t,0} - \hat{X}^{t,0}$ ，使得隐藏概念提取时不混杂预定义概念的信息。第三个模块的输入为初始股票向量表示 - 第一个模块输出的股票向量表示 - 第二个模块输出的股票向量表示，即 $X^{t,2} = X^{t,0} - \hat{X}^{t,0} - \hat{X}^{t,1}$ ，代表每只股票在预定义概念和隐藏概念之外的独立信息。

3. 聚合模块：将part2的信息累加，再通过一个全连接层，得到预测的个股收益率

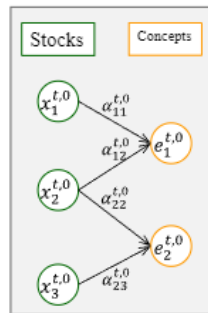
HIST: 预设概念模块

```
market_value_matrix = market_value.reshape(market_value.shape[0], 1).repeat(1, concept_matrix.shape[1])
stock_to_concept = concept_matrix * market_value_matrix
stock_to_concept_sum = torch.sum(stock_to_concept, 0).reshape(1, -1).repeat(stock_to_concept.shape[0], 1)
stock_to_concept_sum = stock_to_concept_sum.mul(concept_matrix)
stock_to_concept_sum = stock_to_concept_sum + (torch.ones(stock_to_concept.shape[0], stock_to_concept.shape[1]).to(device))
stock_to_concept = stock_to_concept / stock_to_concept_sum
```

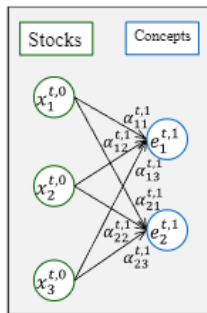
$$\alpha_{ki}^{t,0} = \frac{c_i^t}{\sum_{j \in N_k^t} c_j^t}$$

```
hidden = torch.t(stock_to_concept).mm(x_hidden)
hidden = hidden[hidden.sum(1)!=0]
stock_to_concept = x_hidden.mm(torch.t(hidden))
stock_to_concept = self.softmax_s2t(stock_to_concept)
hidden = torch.t(stock_to_concept).mm(x_hidden)
```

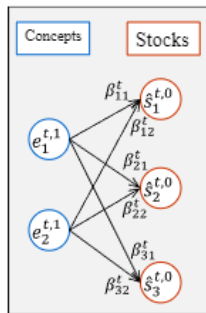
$$e_k^{t,0} = \sum_{i \in N_k^t} \alpha_{ki}^{t,0} x_i^{t,0}$$



(a)



(b)



(c)

```
concept_to_stock = cal_cos_similarity(x_hidden, hidden)
concept_to_stock = self.softmax_t2s(concept_to_stock)
p_shared_info = concept_to_stock.mm(hidden)
p_shared_info = self.fc_ps(p_shared_info)
p_shared_back = self.fc_ps_back(p_shared_info)
output_ps = self.fc_ps_fore(p_shared_info)
output_ps = self.leaky_relu(output_ps)
```

$$v_{ki}^{t,0} = \text{Cosine}(x_i^{t,0}, e_k^{t,0}) = \frac{x_i^{t,0} \cdot e_k^{t,0}}{\|x_i^{t,0}\| \cdot \|e_k^{t,0}\|}$$

$$\alpha_{ki}^{t,1} = \frac{\exp(v_{ki}^{t,0})}{\sum_{j \in S^t} \exp(v_{kj}^{t,0})}$$

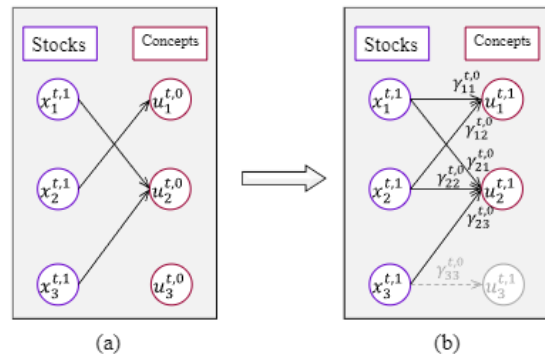
$$e_k^{t,1} = \text{LeakyReLU} \left(w_e \left(\sum_{i \in S^t} \alpha_{ki}^{t,1} x_i^{t,0} \right) + b_e \right)$$

HIST: 隐含概念模块

```
h_shared_info = x_hidden - p_shared_back
hidden = h_shared_info
h_stock_to_concept = cal_cos_similarity(h_shared_info, hidden)
```

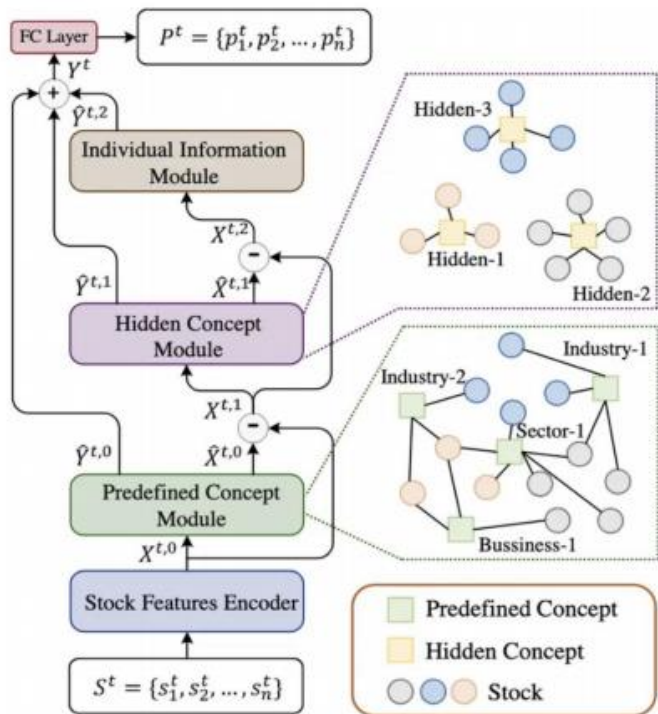
```
dim = h_stock_to_concept.shape[0]
diag = h_stock_to_concept.diagonal(0)
h_stock_to_concept = h_stock_to_concept * (torch.ones(dim, dim) - torch.eye(dim)).to(device)
row = torch.linspace(0, dim-1, dim).reshape([-1, 1]).repeat(1, self.K).reshape(1, -1).long().to(device)
column = torch.topk(h_stock_to_concept, self.K, dim = 1)[1].reshape(1, -1)
mask = torch.zeros([h_stock_to_concept.shape[0], h_stock_to_concept.shape[1]], device = h_stock_to_concept.device)
mask[row, column] = 1
h_stock_to_concept = h_stock_to_concept * mask
h_stock_to_concept = h_stock_to_concept + torch.diag_embed((h_stock_to_concept.sum(0) != 0).float() * diag)
hidden = torch.t(h_shared_info).mm(h_stock_to_concept).t()
hidden = hidden[hidden.sum(1) != 0]
```

```
h_concept_to_stock = cal_cos_similarity(h_shared_info, hidden)
h_concept_to_stock = self.softmax_t2s(h_concept_to_stock)
h_shared_info = h_concept_to_stock.mm(hidden)
h_shared_info = self.fc_hs(h_shared_info)
```



$$u_k^{t,1} = \text{LeakyReLU} \left(w_u \left(\sum_{i \in \mathcal{M}_k^t} \gamma_{ki}^{t,0} x_i^{t,1} \right) + b_u \right)$$

HIST: 个人信息模块



$$X^{t,2} = X^{t,0} - \hat{X}^{t,0} - \hat{X}^{t,1}$$

```
individual_info = x_hidden - p_shared_back - h_shared_back
output_indi = individual_info
output_indi = self.fc_indi(output_indi)
output_indi = self.leaky_relu(output_indi)
```

$$\hat{s}_i^{t,0} = \text{LeakyReLU} \left(w_s^0 \left(\sum_{k \in G^t} \beta_{ik}^t e_k^{t,1} \right) + b_s^0 \right).$$

图神经和神经网络概述

- ① 图分析概述
- ② Graph Attention Network
- ③ HIST
- ④ 图神经网络前沿
- ⑤ 附录

MDGNN: Multi-Relational Dynamic Graph Neural Network for Comprehensive and Dynamic Stock Investment Prediction

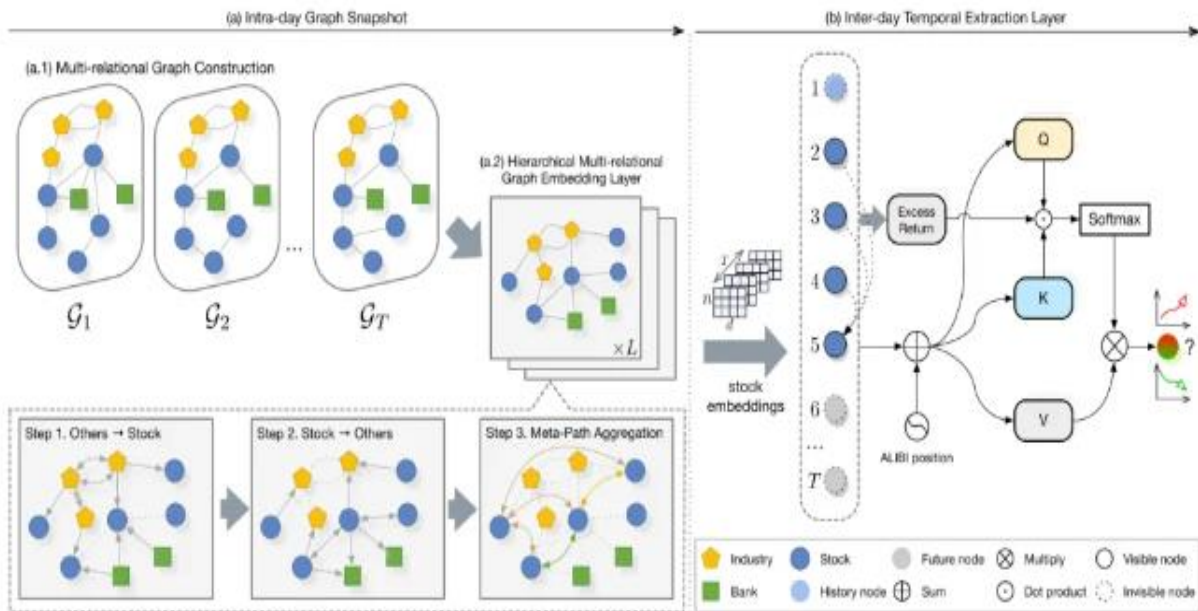


Figure 1: The overview architecture of the MDGNN Model.

模型框架

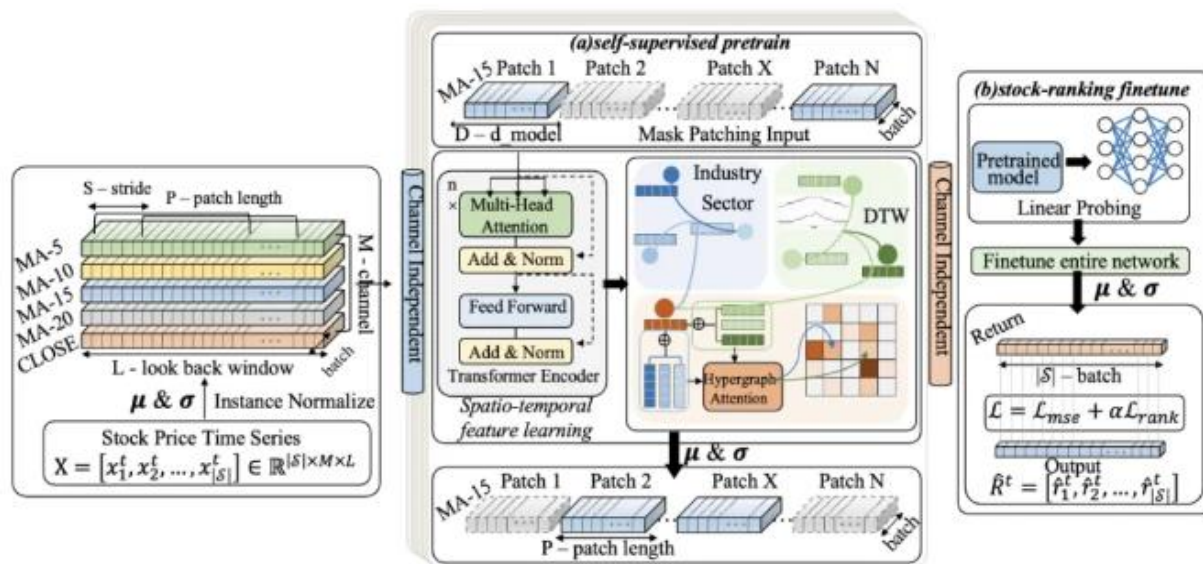


Figure 3: Overview of the proposed CI-STHPAN. A two-stage framework for stock selection, involving Transformer and HGAT based stock time series self-supervised pre-training and stock-ranking based downstream task fine-tuning.

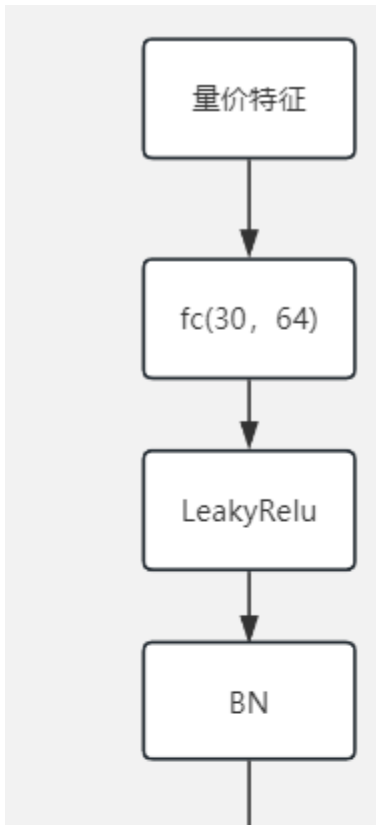
模型框架

图神经和神经网络概述

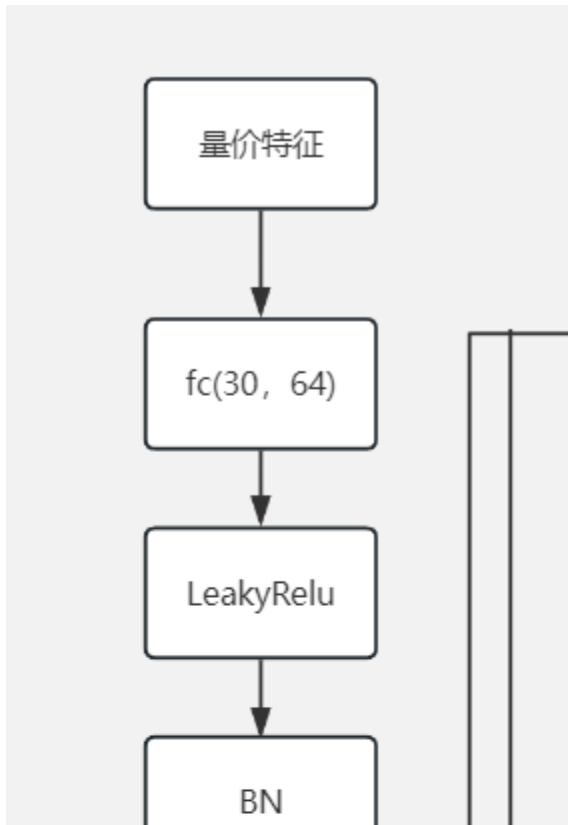
- ① 图分析概述
- ② Graph Attention Network
- ③ HIST
- ④ 图神经网络前沿
- ⑤ 附录

实验网络结构图

GRU:



GAT:





◆行业未来的思考（待补充）



信用卡



推荐书籍



- *The Intelligent Investor* , *Benjamin Graham & Jason Zweig*
- *Devil Take the Hindmost* , *Edward Chancellor*
- *Stock Wizards* , *Jack D Schwager*
- *Manias, Panics&Crashes: A History of Financial Crises* , *Charles Kindleberger*
- *The Alchemy of Finance* , *George Soros*
- *The Crash of 2008 & What It Means* , *George Soros*
- *Lords of Finance: The Bankers Who Broke the World* , *Liaquat Ahamed*
- *Expectations Investing* , *Mauboussin*
- *The Wisdom of Crowds* , *James Surowiecki*
- *Competition Demystified* , *Greenwald*

Q&A



招聘岗位

- 招聘岗位:

详见招聘海报



招聘海报



企业公众号

- Email: HR@mxzichan.com



THANKS FOR WATCHING

上海鸣熙资产管理有限公司

上海市虹口区东长治路359号双狮汇大厦B座1805室

021-61623028

