

Temporal Relational Ranking for Stock Prediction

Abstract

- 주식예측?
 - a. 올바른 투자 결정을 내릴 수 있도록 주식의 미래 추세를 예측하는것
 - b. 과거에는 시계열 데이터로 예측, 최근엔 딥러닝을 통해서 예측하고 있음
- 문제점
 - a. 딥러닝은 주식의 **Classification**(트렌드 예측), **Regression**(주가예측)으로 한정함
 - b. 수많은 주식을 독립적으로만 처리하고 있음(주식간의 관계의 부재)
 - c. 두 주식간의 종속관계, 혹은 같은 **SCM** 체인은 고려되지 않음
 - i. 구글과 알파벳 관계 등
- 제안
 - a. 주식 예측을 위해 **Relational Stock Ranking (RSR)**이라는 새로운 방법 제안
 - i. 주식의 순위를 딥러닝으로 조절
 - ii. 시간에 민감한 주식들의 관계를 캡처링
 - b. 이것을 하기위한 딥러닝 아키텍처 **Temporal Graph Convolution**를 새로 만듦

Intro

Table 1. An intuitive example that one method predicting the price change of stocks more accurately (*i.e.*, smaller MSE) leads to a less profitable stock selection (*i.e.*, smaller profit). Method 1 selects stock A (30) while Method 2 selects stock B (10).

Ground Truth			Method 1					Method 2				
			Prediction			Performance		Prediction			Performance	
A	B	C	A	B	C	MSE	Profit	A	B	C	MSE	Profit
+30	+10	-50	+50	-10	-50	266	30	+20	+30	-40	200	10

A, B, C denote three stocks; numbers (+20) are the true/predicted price change of stocks; values in bold correspond to suggested selections.

- 보다 정확한 주식 예측을 위하여 Mse가 적은 모델을 선택하는데, mse가 작은 방법을 선택 했을때 적은 Profit을 얻을 수도 있다.

Intro

- 주식 예측 과거 방법
 - a. 주식 예측의 과거방법은 Kalman Filter, Autoregressive model같은 시계열 분석 모델 기반
- 과거 문제점
 - a. 지표(index)에 크게 의존함, Finance 전문가가 아니면 최적화 하기 힘들
 - b. 통계 기반인데, Real Market이 꼭 통계적으로 움직이지 않음
 - c. 그렇다고 지표를 늘리면 모델이 복잡해져서 작동안함
- 최근 방법과 문제점 1
 - a. 딥러닝 RNN등을 활용하여 최근은 트렌드나 가격을 비교적 정확히 맞추고 있음
 - b. SFM(State Frequency Memory)은 현재 A state-of-the-art, 하루 50달러 이하의 mse를 나타냄
 - i. 논문 보고 싶다~~~
 - c. 그러나 앞에서 보았듯이 mse가 낮은 모델이라고 최대이익을 보장해주진 않음

Intro

- 최근 방법과 문제점2

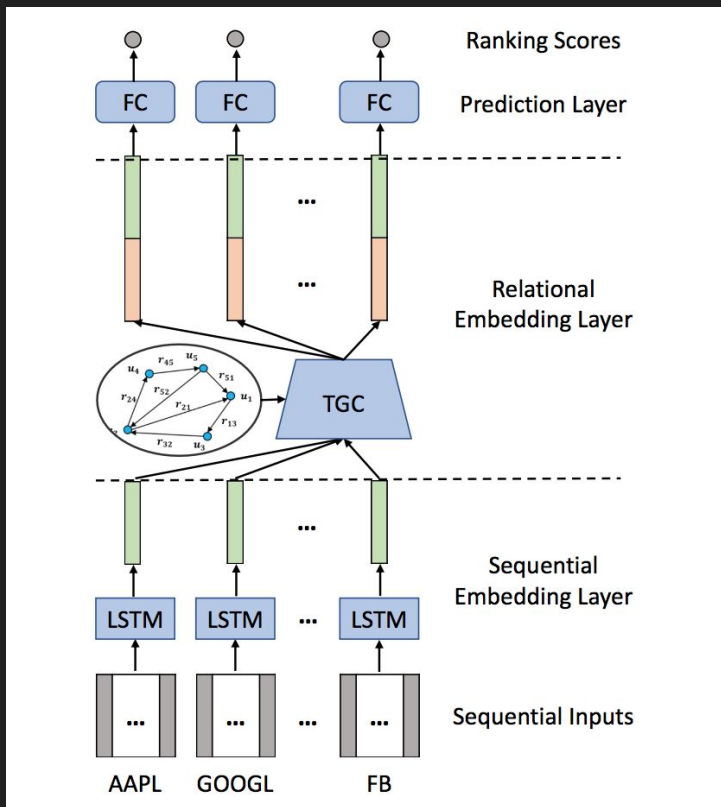
- a. 딥러닝은 기본적으로 데이터는 **independence** 한다는 가정으로 시작
- b. 주식은 서로간의 관계가 중요한 단서가 될수 있으나 훈련 못함
 - i. 구글과 페이스북은 같은 산업군으로 유사한 장기 추세를 가질수 있음
 - ii. scm에서 공급업체가 문제가 생기면 관련된 회사들은 같이 주식이 떨어질수 있음

- 제안 방법

- a. 주식 간의 관계를 그래프로 그리고, 그래프 기반으로 주식 예측을 정규화
- b. 그러나 Convolution은 시간이 고정되어 있어 시장의 변화를 감지할수 없음
- c. 그래서 주식을 순위를 예측하는 것으로 바꾸고, 순위의 기준은 수익률 같은 우리가 정하는 것으로 설정
- d. 그런다음 순위 문제를 해결하기 위해 Relational Stock Ranking(RSR)을 제안함

Intro

Relational Stock Ranking(RSR)



- RSR설명

- Sequential Emb Layer : 과거 주식 데이터를 데이터를 LSTM으로 훈련
- Relational Emb Layer : 새로 고안된 Temporal Graph Convolution (TGC)에 시간에 민감한 주식간의 관계를 설명하는 방식으로 emdding을 훈련(?)
- 합쳐서 FC로 랭킹스코어 구함
- d.

이상태로 했더니 FSM보다
115% 많이 나옴

PRELIMINARIES

- Long Short-Term Memory
- Graph-based Learning
 - a. 주식 간의 관계를 그래프로 그리고, 그래프 기반으로 주식 예측을 정규화
 - b. 그러나 Convolution은 시간이 고정되어 있어 시장의 변화를 감지할수 없음
- Graph Convolutional Networks
 - a. Graph Convolutional Networks (GCN)

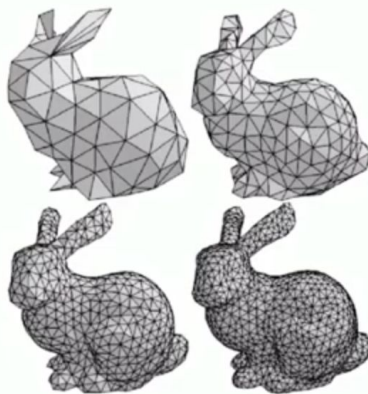
Graph Convolutional Networks?

graph?

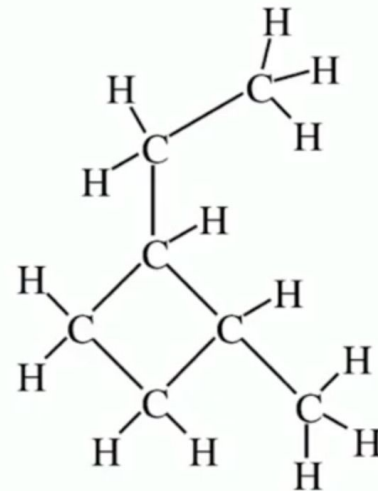
Data Representation – Graph



Social Graph



3D Mesh

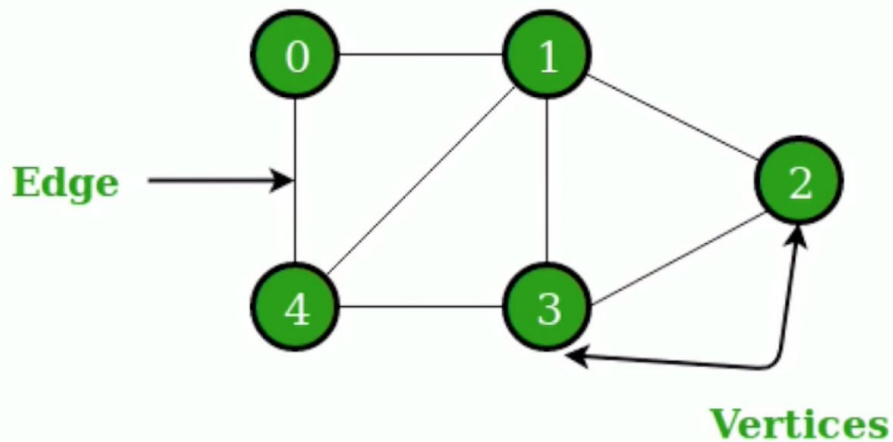


Molecular Graph

Graph Convolutional Networks?

컴퓨터공학에서 표현하는 graph?

Graph Structure



Vertex (Node)

Edge

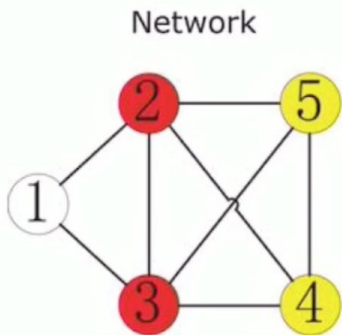
Graph Convolutional Networks?

컴퓨터공학에서 표현하는 graph?

Graph Structure

Vertex (Node)
Node Feature Matrix

Edge
Adjacency Matrix



연결정보

Adjacency matrix A

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

피쳐정보

Feature matrix $A+I$

	feature 1	feature 2	feature 3	feature 4	feature 5
node 1	1	1	1	0	0
node 2	1	1	1	1	1
node 3	1	1	1	1	1
node 4	0	1	1	1	1
node 5	0	1	1	1	1

Graph Convolutional Networks?

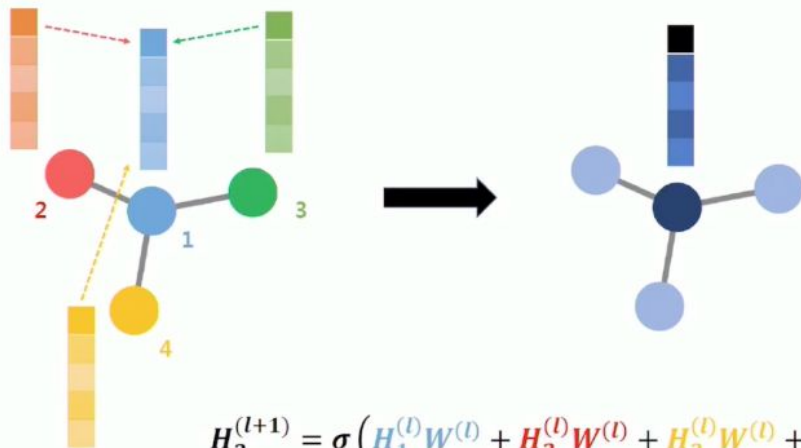
- CNN은?
 - a. Convolution 연산을 사용해서 이미지에서 피처를 추출
 - b. 웨이트 셰어링(장점 : 파라미터 수가 작아지고, 계산량도 줄고, 오버피팅도 잘 안되고)
 - c. 로컬피처, 리셉티브 필드를 배우게 된다.(Local feature)
 - d. 픽셀이 달라진다 하더라도 잘 인식한다. (Translation invariance)

- Graph Convolution Network도 같은 성질을 가져야함
 - a. 어떻게 하면 될까?

Graph Convolutional Networks?

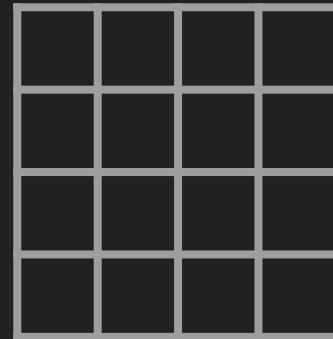
컴퓨터공학에서 표현하는 graph?

How to update hidden states in GCN



$$H_2^{(l+1)} = \sigma \left(H_1^{(l)} W^{(l)} + H_2^{(l)} W^{(l)} + H_3^{(l)} W^{(l)} + H_4^{(l)} W^{(l)} + b^{(l)} \right)$$

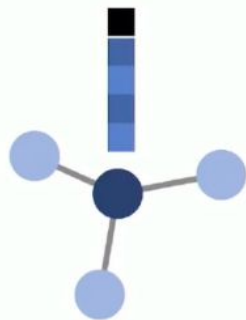
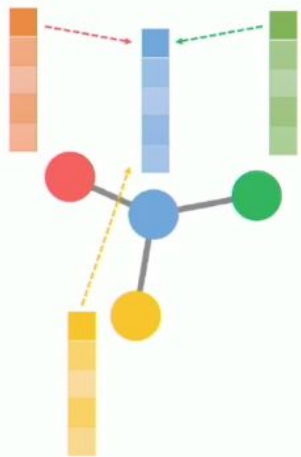
$$\Rightarrow H_i^{(l+1)} = \sigma \left(\sum_{j \in N(i)} H_j^{(l)} W^{(l)} + b^{(l)} \right)$$



- 노드 피쳐 M_a
- 주변 노드 피쳐와 웨이트를 곱해서 더해서 시그모이드
- CNN의 local feature와 weight sharing 과 같음

Graph Convolutional Networks?

How to update hidden states in GCN

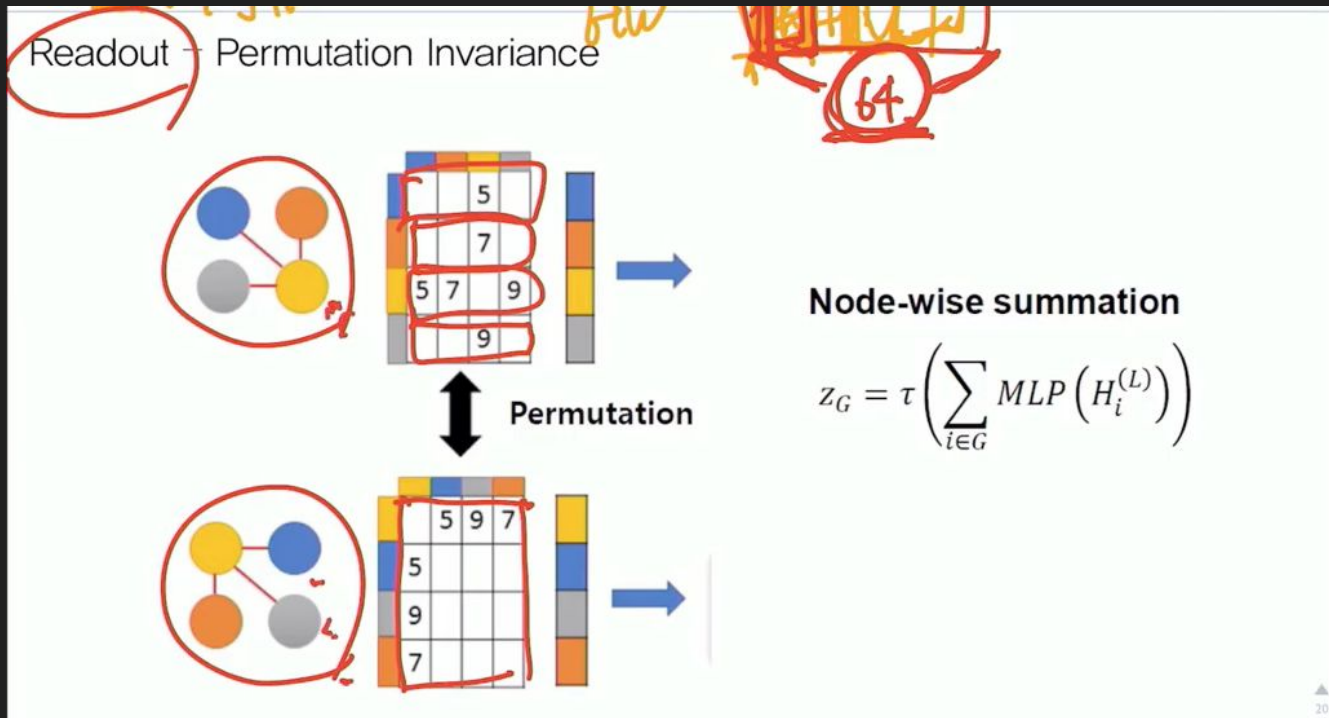


$$H^{(l+1)} = \sigma(AH^{(l)}W^{(l)} + b^{(l)})$$

learnable parameters are shared

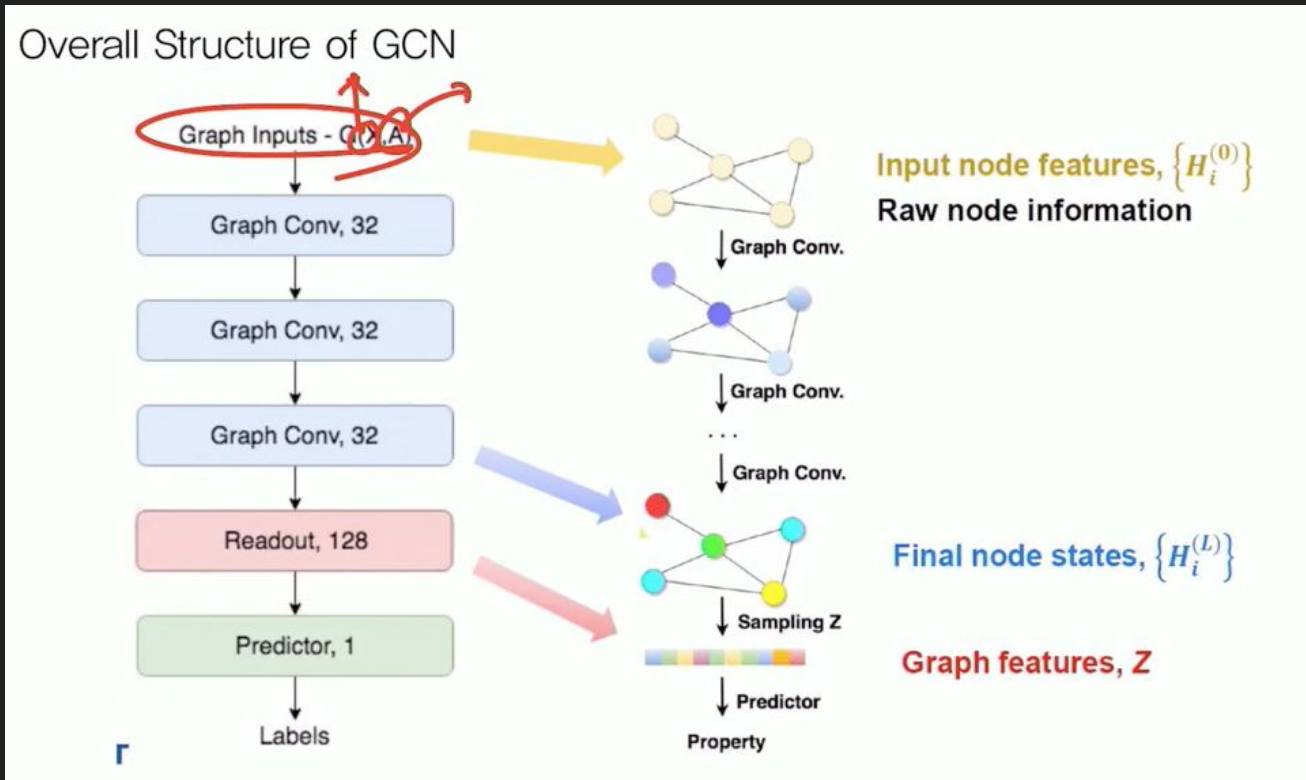
- 연결 정보인 adjacency matrix를 곱해서 식을 완성함
- 매트릭스 연산이기 때문에 빠르고 gpu사용 가능

Graph Convolutional Networks?



- Translation invariance하기 위해서 readout layer 추가

Graph Convolutional Networks?



- 그래프의 특성을 뽑아서 어떤 특성을 가진 그래프 인지 예측할수 있음

RELATIONAL STOCK RANKING

- RSR

- a. 주식은 보통 시계열로 가격을 예측하지만
- b. 여기서 해야 하는것은 수익에 따른 Rank를 예측할 것
- c. 또한 주식간의 명시적 관계를 넣어(Graph) 잠재적 영향을 반영할것임

- Framework

- a. Sequential Embedding Layer.

- i. 주식시장의 과거 추세로 미래를 예측할수 있다는것은 직관적이고, 그래서 순차적으로 데이터를 넣음
 1. 예를들어 주식이 장기간 떨어지면 평균선으로 회귀하려는 성질을 보임
- ii. 또한 RNN은 시계열 처리에 좋고, Long Term을 보기위해 LSTM을 사용함

RELATIONAL STOCK RANKING

- Framework

- b. Relational Embedding Layer

- i. 두개의 다른 주식의 관계를 모델링 하는 방법
 - ii. 순차 데이터에 명시적 도메인 지식(예: 주식관계)을 삽입하는 방법을 고안함 injection of explicit domain knowledge (i.e., stock relations)
 - iii. 명시적 도메인 지식(주식관계의 예)
 - 1. 두 회사가 산업군이 같으면 추세가 비슷
 - 2. 같은 scm에 있고 호재면 가격이 오름(아이폰 렌즈회사가 아이폰 때문에 가격 오름)
 - iv. Temporal Graph Convolution
 - 1. 주식시장에서 이러한 관계를 찾아내기 위해서 Temporal Graph Convolution를 고안하였고 시간에 민감한 관계 embedding을 만들어 낸다.

RELATIONAL STOCK RANKING

- Framework

- c. Prediction Layer

- i. 마지막으로 LSTM에서 나온 embedding과 relation embedding을 합쳐서 FC에 넣고 순위 점수를 예측한다.
 - ii. 나온 점수로 순위 리스트를 만든다.

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

- iii. 첫번째는 그냥 GT의 rank와 예측의 rank의 비교
 - iv. 두번째는 Pair단위 랭크 차의 곱에서 마이너스 한것 중 0과 비교하여 큰것(max margin loss)
 - v. 알파는 하이퍼 파라미터

RELATIONAL STOCK RANKING

- Temporal Graph Convolution

- LSTM에서 나온 embedding과 GCN의 feature vector를 E_t 기준으로
- 주식관계를 매트릭스로 만든 매트릭스 adjacency matrix a_{ij} (i 와 j 의 관계)로 graph-based learning methods를 실행한다..

A. Uniform Embedding Propagation

- i. 관계 매트릭스에서 a_{ij} 가 연관이 있는 것만 뽑아서 더함

$$\bar{\mathbf{e}}_i^t = \sum_{\{j | \text{sum}(a_{ji}) > 0\}} \frac{1}{d_j} \mathbf{e}_j^t,$$

RELATIONAL STOCK RANKING

- Temporal Graph Convolution.

- B. Weighted Embedding Propagation

- i. 두 주식간의 관계가 adjacency matrix 말고, 다른 변수에 의해 가격에 영향을 끼칠수 있으므로 두 주식관계의 영향관계를 배우는 관계감도함수 g 를 설정한다.
 - ii. 그냥 웨이트이고 이것을 학습함

$$\bar{\mathbf{e}}_i^t = \sum_{\{j | \text{sum}(\mathbf{a}_{ji}) > 0\}} \frac{g(\mathbf{a}_{ji})}{d_j} \mathbf{e}_j^t,$$

RELATIONAL STOCK RANKING

- Temporal Graph Convolution.

- C. Time-aware Embedding Propagation

- i. 관계뿐만 아니라 주식시장은 시간관계도 중요하므로 관계감도함수에 a_{ij} (adjacency matrix, lstm embedding1, lstmembedding2)를 넣어서 구한다,

$$\bar{\mathbf{e}}_i^t = \sum_{\{j | \text{sum}(a_{ji}) > 0\}} \frac{g(a_{ji}, \mathbf{e}_i^t, \mathbf{e}_j^t)}{d_j} \mathbf{e}_j^t,$$

RELATIONAL STOCK RANKING

- Temporal Graph Convolution.

- C. Time-aware Embedding Propagation

- i. 시간을 고려한 관계 감도 함수 2가지를 구하는 방법

- 1. Explicit Modeling

- a. 두 주식간의 유사성 과 관계의 강도의 곱을 학습

$$g(\mathbf{a}_{ji}, \mathbf{e}_i^t, \mathbf{e}_j^t) = \underbrace{\mathbf{e}_i^{tT} \mathbf{e}_j^t}_{\text{similarity}} \times \underbrace{\phi(\mathbf{w}^T \mathbf{a}_{ji} + b)}_{\text{relation importance}},$$

- 2. Implicit Modeling

- a. 그냥 Fully connect에 다 넣어서 학습함. 더 많은 비선형성을 확보함

$$g(\mathbf{a}_{ji}, \mathbf{e}_i^t, \mathbf{e}_j^t) = \phi(\mathbf{w}^T [\mathbf{e}_i^{tT}, \mathbf{e}_j^{tT}, \mathbf{a}_{ji}^T]^T + b),$$

RELATIONAL STOCK RANKING

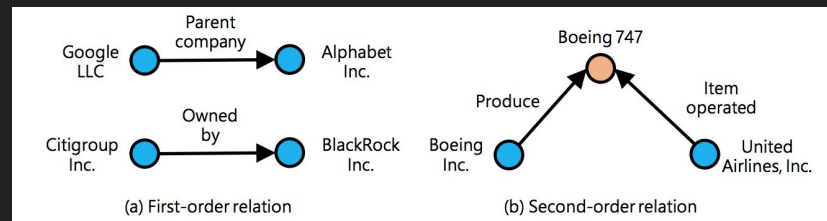
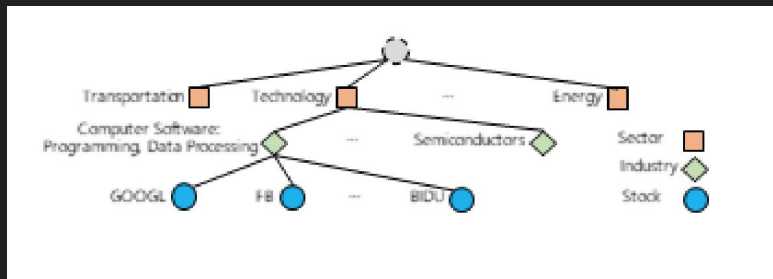
- Temporal Graph Convolution.

- D. Connection with Graph-based Learning

- i. GCN은 시간을 고려하지 못하는데 위에서 말함 TGC는 시간을 고려 했으므로 GCN의 시간패턴을 추가하여 일반화 한것으로 볼수 있으므로 Temporal Graph Convolution이라고 볼수 있음

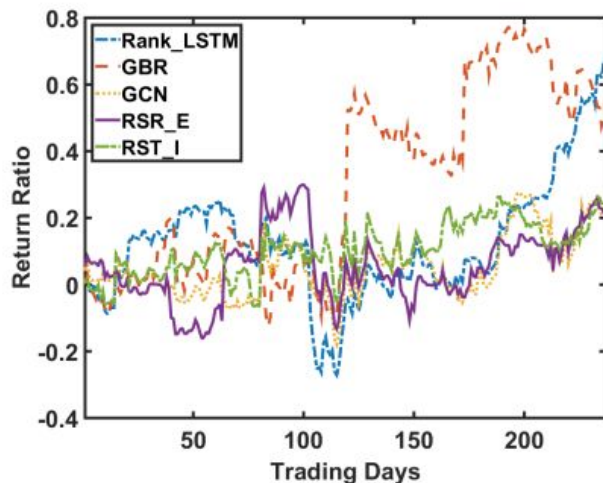
Data Collection

- Sequential Data
 - A. 1 day 증가 수익률로 ranking을 계산함
 - B. 전체 데이터 셋의 max값으로 노말라이징 함
 - C. 5, 10, 20 , 30의 이동평균선을 계산함
- Stock Relation Data
 - A. Sector-Industry relations
 - B. Wiki Company-based Relations

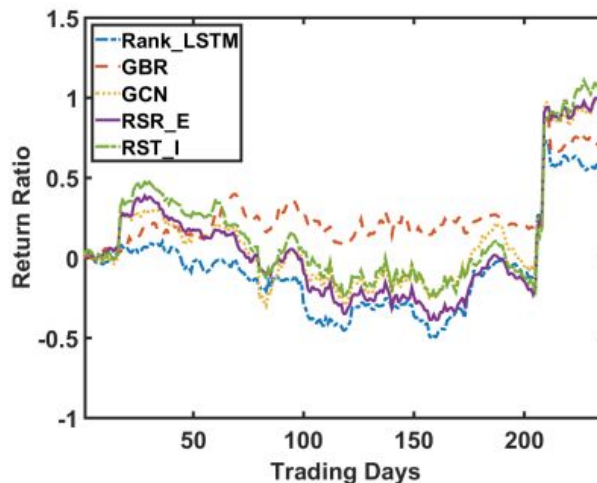


Experiments

- Sector-Industry relations
 - rank_LSTM : relation 없는것
 - GBR : 보통 graph learning 방식
 - GCN : SOTA graph learning
 - RSR_E : explicit 제안방법
 - RSR_I : impicit 제안방법



(a) NASDAQ



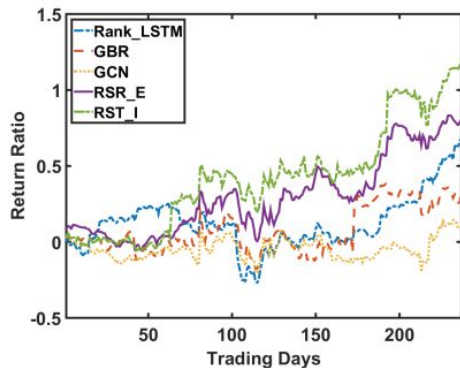
(b) NYSE

Experiments

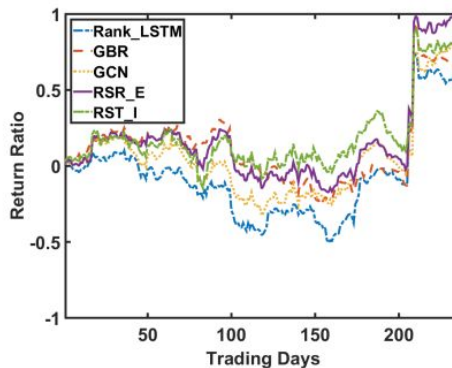
- Wiki Company-based Relations

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-6	4.28e-2±6.18e-3	0.96±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



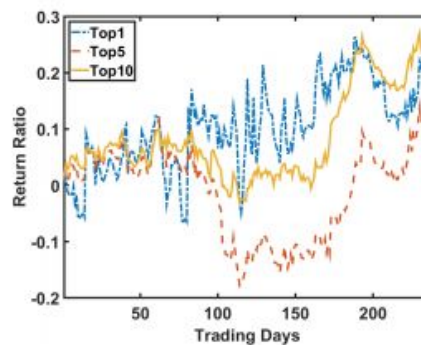
(a) NASDAQ



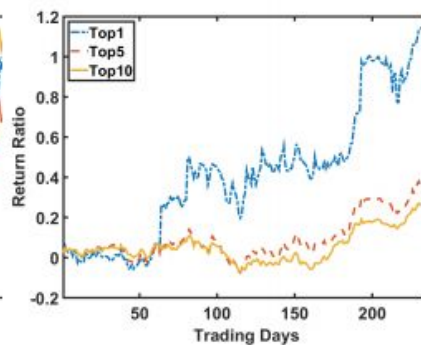
(b) NYSE

Experiments

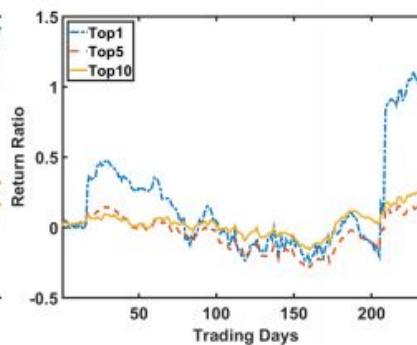
- Relations



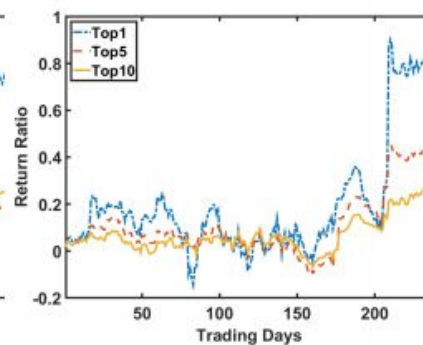
(a) NASDAQ-Industry



(b) NASDAQ-Wiki



(c) NYSE-Industry



(d) NYSE-Wiki