Earnings announcement return Extrapolation

By Wensheng Li / Chenyu Wu

GAR-Earnings-Team-2

June 1st

Motivation

- **Ret1 and ts_delay(ret1, 1)** demonstrate strong predictive capabilities
- Earnings announcement return **extrapolation** Shows:
- There is a significant **positive correlation** between **historical** earnings announcement returns and **future** earnings announcement returns.

- **Supporting Theories from Literature**
- **Attention Theory**: Investors focus more on stocks on a few important days, impacting their future behavior a lot.
- **Gambling Theory**: Investors' tendency to gamble on stocks with high historical returns influences future performance.

1 Introduction

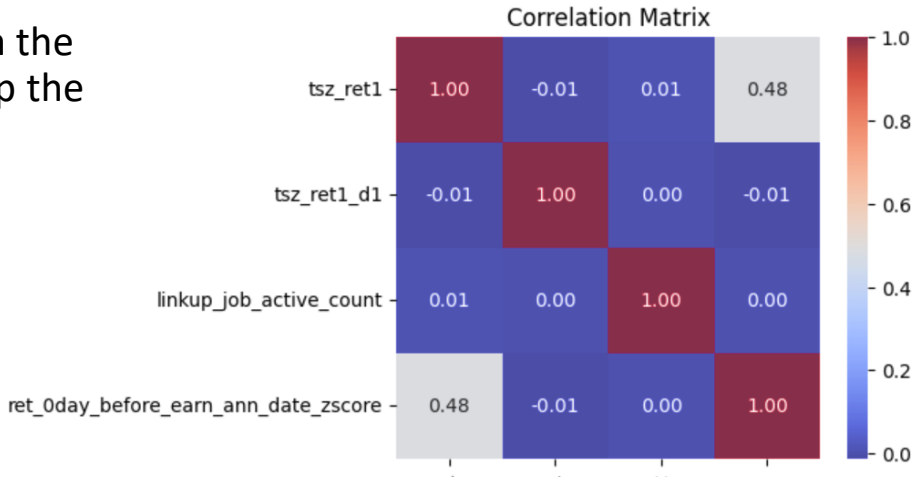
Earnings announcements (EAs) are a focal point for many investors and drive abnormal trading volume and stock returns (e.g., Beaver 1968; Shao et al. 2020). Moreover, there is significant evidence that EAs are salient predictable events upon which investors like to gamble (e.g., Liu et al. forthcoming). A common motivation for gambling is a belief about the gambling event.[1] If investors have extrapolative expectations with respect to the principal outcome of EAs—the EA *return*—then they will believe that stocks with a recent series of high EA returns will continue to have high EA returns. Gambling on these beliefs will lead to a positive demand shock and positive price pressure, shortly before the upcoming EA. If these extrapolative beliefs are biased, we would also expect a reversal after earnings information is released and fundamental traders respond (Engelberg et al. 2018). We posit that extrapolative beliefs help explain return patterns around EAs.

The idea that investors extrapolate EA returns is grounded in significant theoretical and experimental work. In their seminal work, Tversky and Kahneman (1971, 1974) document evidence of representativeness. Representativeness captures a belief in the "law of small numbers." That is, people think the true distribution should be reflected in small samples. Consequently, when the underlying distribution is unknown, people infer it from a small sample of data. We hypothesize that people will infer the distribution of EA returns from recent EA returns and use these inferences to predict EA returns—in other words, they will extrapolate future EA returns based on past ones. We speculate that investors will focus on past EA returns, rather than past fundamentals, in keeping with the observed tendency of investors to focus on returns, rather than earnings (Blankespoor et al. 2019).

First test

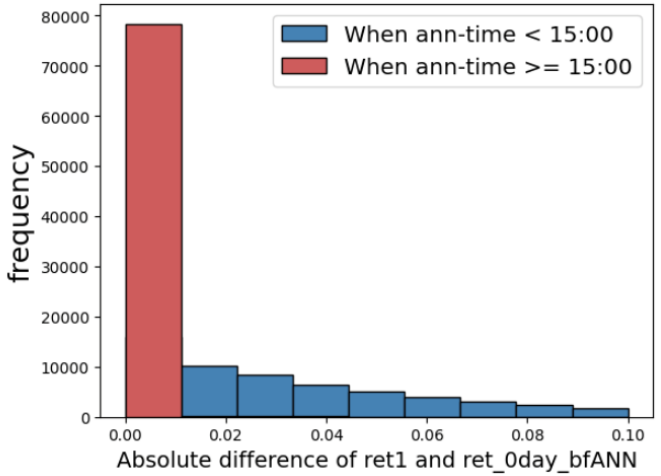
- **ret_0day_before_earn_ann_date:** return on the day of the earning announcement date. Keep the value until next earning announcement day
- Coverage:3443
- empty_value: None
- Units: None

	accuracy	ir	ret	tvr	dd	liqn
train	51.413	0.085	0.138	0.369	13.115	41.074
valid	51.436	0.085	0.131	0.332	11.119	41.074



- Grouping the data into 2 groups: ann_time < 1500 and ann_time ≥ 1500:
  - The difference between ret1 and ret_0day_before_ann is significantly smaller for ann_time ≥ 1500, **and the correlation tends to 1.00**
  - If the ann_time ≥ 1500, **the ret_0day_before_ann is already updated for the current EA we are focusing on**. So it overlaps with the ret1 on the target day.

Where is the high corr coming from?



<15:00	ret_0_bfEA	ret1
ret_0_bfEA	1.000000	0.144816
ret1	0.144816	1.000000

≥15:00	ret_0_bfEA	ret1
ret_0_bfEA	1.000000	0.994948
ret1	0.994948	1.000000

Purify the feature

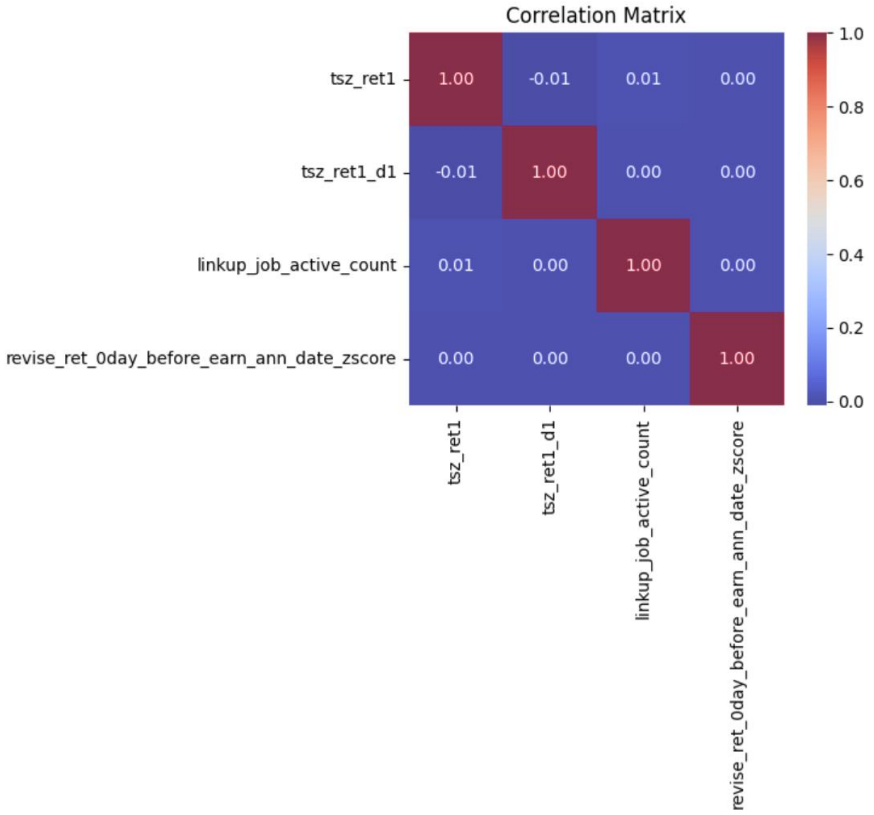
```
revise_ret_0day_before_earn_ann_date = np.where(op.ts_delay(np.array(fs_next_ann_time, dtype=np.float32), 75)<1500.0, \
                                                op.ts_delay(ret_0day_before_earn_ann_date, 1),
                                                op.ts_delay(ret_1day_after_earn_ann_date, 1))
```

revise_ret_0day_before_earn_ann_date

	accuracy	ir	ret	tvr	dd	liqn
train	51.372	0.080	0.134	0.366	15.841	38.014
valid	51.436	0.126	0.215	0.331	5.549	38.014

ret_0day_before_earn_ann_date_zscore'

	accuracy	ir	ret	tvr	dd	liqn
train	51.413	0.085	0.138	0.369	13.115	41.074
valid	51.436	0.085	0.131	0.332	11.119	41.074



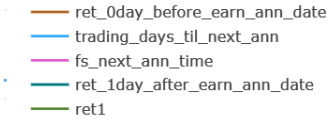
- **ret_1day_after_earn_ann_date:** return for 1 day after the earning announcement date
- Coverage:3466
- empty_value: None

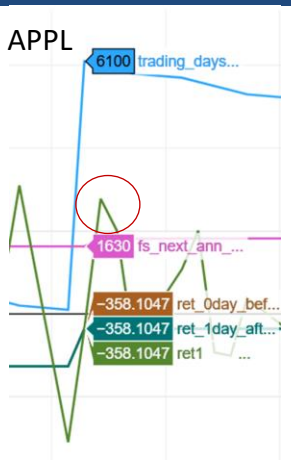
ret_1day_after_earn_ann_date

	accuracy	ir	ret	tvr	dd	liqn
train	51.343	0.078	0.134	0.366	17.044	38.620
valid	51.391	0.117	0.212	0.334	7.951	38.620

Revise version 2

- There must still have some problems.






```
# find the earning annoucement jump on next day when flatten_mask==1
earning_jump = np.where(flatten_mask[:,:-1] == 1, ret1[:,1:], np.nan)
# Cannot get the last day date,so use np.full to create the same shape
earning_jump = np.concatenate((earning_jump, np.full((ret1.shape[0], 1), np.nan, dtype=np.float32)), axis=-1)
# using ts_fill to make the last EA return fill the nan position between 2 EA days
# this way will let us to know what is the last EA return during no-EA daysa
earning_jump_fill = op.ts_fill(earning_jump)
# Cannot use the newest EA return, we need to use the last one,so we need to find the values from the days between 2 EA days(last EA return)
revise2_ret_0day_before_earn_ann_date = op.ts_delay(earning_jump_fill, 5)
```

ret_0day_before_earn_ann_date_zscore'

	accuracy	ir	ret	tvr	dd	liqn
train	51.413	0.085	0.138	0.369	13.115	41.074
valid	51.436	0.085	0.131	0.332	11.119	41.074

revise_ret_0day_before_earn_ann_date

	accuracy	ir	ret	tvr	dd	liqn
train	51.372	0.080	0.134	0.366	15.841	38.014
valid	51.436	0.126	0.215	0.331	5.549	38.014

revise2_ret_0day_before_earn_ann_date_zscore

	accuracy	ir	ret	tvr	dd	liqn
train	51.356	0.094	0.163	0.363	16.066	39.276
valid	51.432	0.099	0.161	0.328	12.102	39.276

ret_3day_around_ann_date

	accuracy	ir	ret	tvr	dd	liqn
train	51.533	0.091	0.147	0.367	12.944	42.212
valid	51.474	0.091	0.139	0.329	7.500	42.212

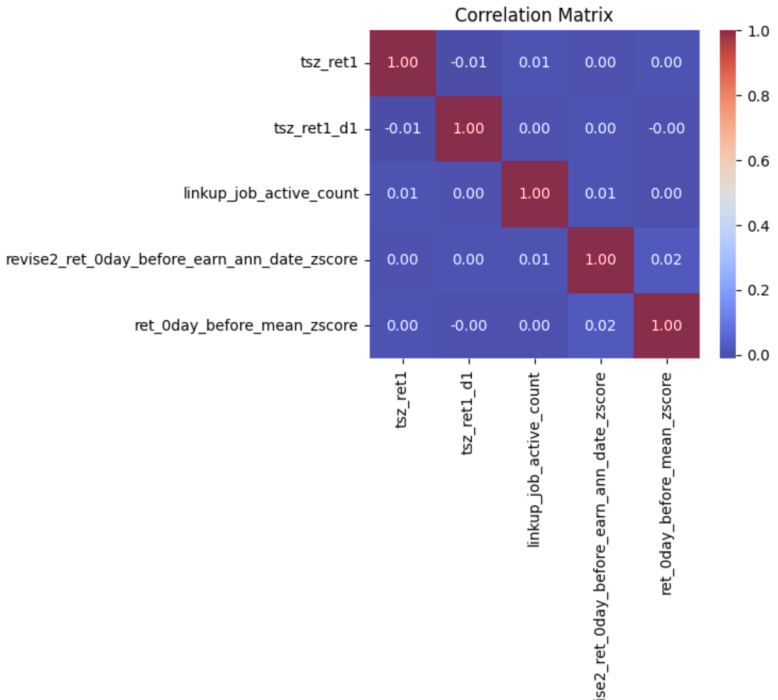
Results

- Only one past earning return can not express the whole picture
- Calculate returns on earnings announcement dates for the past four quarters
- se their EMW decay values as predictive features

'ret_0day_before_mean_zscore

	accuracy	ir	ret	tvr	dd	liqn
train	51.363	0.091	0.146	0.366	16.125	27.873
valid	51.548	0.097	0.155	0.338	8.463	27.873

	accuracy	ir	ret	tvr	dd	liqn
train	51.362	0.100	0.174	0.364	19.635	37.727
valid	51.394	0.110	0.177	0.336	11.396	37.727



ret_3day_after_earn_ann_date

	accuracy	ir	ret	tvr	dd	liqn
train	51.372	0.078	0.134	0.367	19.431	42.625
valid	51.391	0.075	0.120	0.331	12.819	42.625

ret_5day_after_earn_ann_date

	accuracy	ir	ret	tvr	dd	liqn
train	51.337	0.071	0.118	0.367	18.050	37.415
valid	51.244	0.073	0.122	0.331	10.595	37.415

Thanks

Trexquant week 4 Notebook Stratagy

June 1st, 2024

Jack(Wensheng) Li

Appendix A

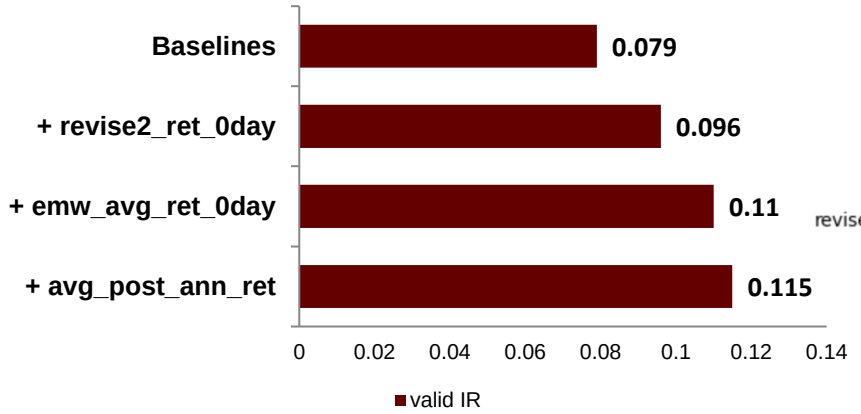
- Extrapolation includes 2 types, historical Announcement jump & historical post-Announcement
- Another Extrapolation lies in **historical 5days post-Announcement** in the past year.

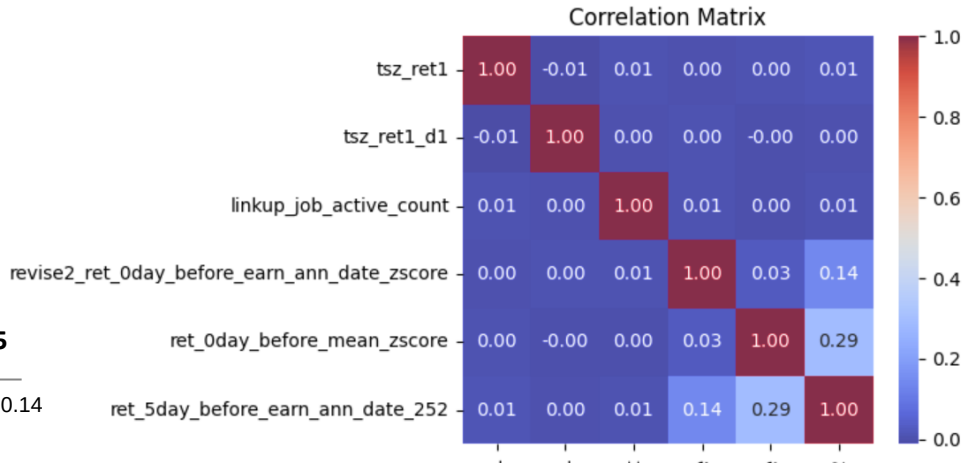
ret_5day_before_earn_ann_date_252

	accuracy	ir	ret	tvr	dd	liqn
train	51.518	0.093	0.143	0.364	13.157	44.115
valid	51.470	0.087	0.124	0.325	6.476	44.115

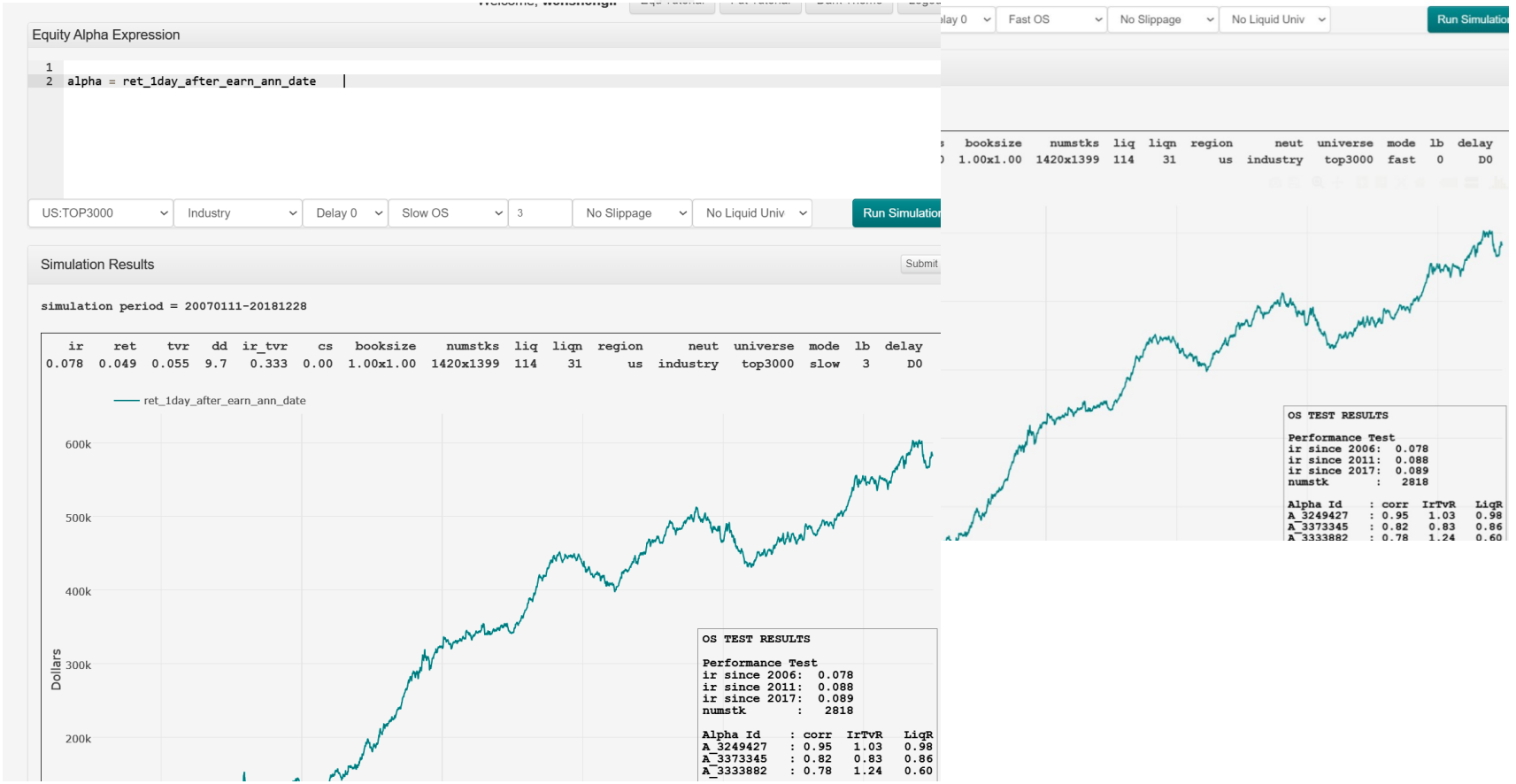
overall_result

	accuracy	ir	ret	tvr	dd	liqn
train	51.369	0.105	0.175	0.363	20.146	37.221
valid	51.340	0.115	0.173	0.331	9.607	37.221





Appendix


Equity Alpha Expression
1
2
alpha = ret_1day_after_earn_ann_date
US:TOP3000
Industry
Delay 0
Slow OS
3
No Slippage
No Liquid Univ
Run Simulation
Simulation Results
Submit
simulation period = 20070111-20181228
ir ret tvr dd ir_tvr cs booksize numstks liq liqn region neut universe mode lb delay
0.078 0.049 0.055 9.7 0.333 0.00 1.00x1.00 1420x1399 114 31 us industry top3000 slow 3 D0
ret_1day_after_earn_ann_date
Dollars
600k
500k
400k
300k
200k
OS TEST RESULTS
Performance Test
ir since 2006: 0.078
ir since 2011: 0.088
ir since 2017: 0.089
numstk : 2818
Alpha Id : corr IrTvR LiqR
A_3249427 : 0.95 1.03 0.98
A_3373345 : 0.82 0.83 0.86
A_3333882 : 0.78 1.24 0.60