Machine Learning for Factor Investing

An overview

Guillaume Coqueret

Stanlib **Friday** seminar 2020-12-27

Introduction

Overview of AI use-cases in finance

- **task automation** (ex: reading/summarizing, or writing reports)
- **customer relationship managment** (ex: chatbots, churn minimization)
- **fraud detection** (ex: credit card hacking)
- **credit scoring** (for firms & individuals)
- **asset allocation** (return prediction, risk management... option pricing?)

Machine Learning (ML) for trading: why?

1. Because **we can** (data availability, software democratization (open source!), and academic maturity).
2. Because it's **fancy** (it makes great marketing pitches because people like seeming sophistication).
3. Because it **works** (well… it depends).

What is AI / ML?

- AI refers to tools (algos, robots) which are aimed at **mimicking human cognitive functions** (and, subsequently, motor functions).
- One obvious application of AI is **task automation**.
- Nowadays, building intelligent systems requires **huge amounts of data** (which sometimes can be simulated).
- ML (a subfield of AI) is intended to "make sense" (i.e., **detect patterns**) of these big datasets (akin to data mining)
- If these patterns are *genuine*, they can be incredibly valuable (more on that soon).

→ in practice, ML is very efficient for **marketers**.

ML for trading: a tale of frequencies

The type of ML algo & data used depends on the rebalancing rythm:

- **high-frequency**: intraday trades (~mins). The volume of data comes from **chronology** and the aim is to unveil recent patterns in the **microstructure** process (order book). Usually, the focus is on a limited number of assets (ex: futures, indices, forex) and a limited number of indicators (prices).
- **low frequency**: monthly/quarterly/annual rebalancing. The volume comes from **cross-sections**: width of assets

  (hundreds/thousands of stocks, bond issuers), and width of attributes (hundreds)

Low frequency trading with factor investing

In **financial economics**, one central topic is to explain the profitability of firms, that is, why do different firms experience different returns.

The key word is *different*: in what ways are companies different?
→ we need to **characterize** them. To do so, we resort to attributes:

- **size** (market cap),
- **value** (accounting ratios),
- **momentum** (past perf),
- **risk** (volatility),
- etc.

Characteristics-based models: a primer

The equation (there is just one, but it's important to understand it)

Here it goes*:

$$r_i = f(\mathbf{x}_i) + e_i,$$

- r_i is the (future) return of firm i,
- f is some possibly complex function,
- $\mathbf{x}_i$ are the firms characteristics (market cap, earning/debt ratios, past returns, etc.),
- e_i is the error made by the model ($f(\mathbf{x}_i)$)

Note: it a **panel** model: `(f)` is the same for all stocks (the model learns from the cross-section.)

*It's the simple version: it can (should?) be made time-dependent.

A simple example

Assume

$$r_i = a + b * \mathrm{Size}_i + e_i,$$

where Size is a **proxy** of the size of the company (e.g. market capitalization - rescaled/normalized/standardized).
If $b > 0$: large firms earn higher returns (according to the model).
$\rightarrow$ Usually, it is considered that $b < 0$: small firms have more potential for **growth**, and thus experience enhanced performance (more on that soon).

This is related to the so-called **size premium** (or anomaly).

There are many anomalies: value, momentum, low risk (?), profitability, etc.

Generalizations

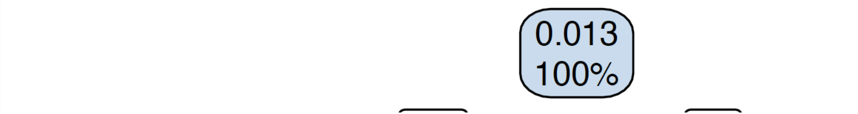
Extensions include:

- adding more characteristics (accounting ratios, risk, sentiment, ESG, etc.);
- going beyond linear forms (that where the ML kicks in);
- reinforcing conditionality (ex: via macro indicators).

BUT! You should always be wary about the error terms e_i!
Gaussian? Independent (in time, in the cross-section)?
Maybe not...

Illustrating nonlinearity with many features

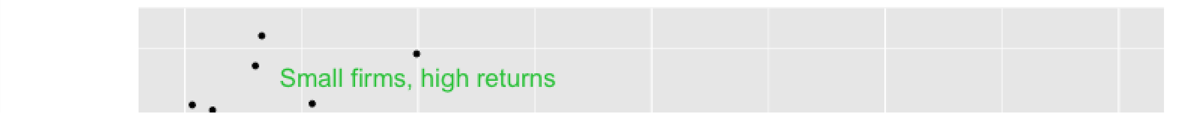
A simple decision tree.



Factor models: limitations

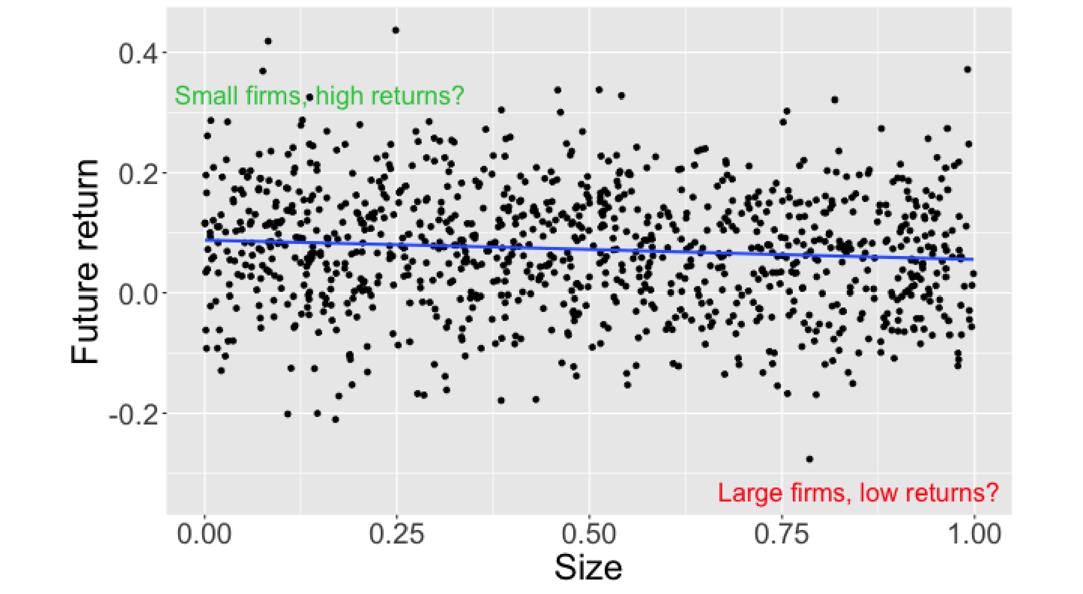
First issue: noise! (1/2)

Optimal case: low noise (stylized graph).



First issue: noise! (2/2)

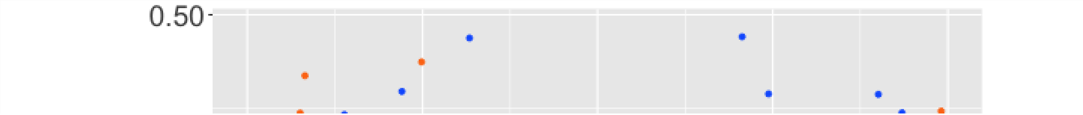
Second configuration: overwhelming noise (more realistic).



In practice, things are much more complicated (many predictors, nonlinearities, etc.)

Second issue: everything is time-varying (1/2)

Average returns, volatility, factor loadings, they all **change**! (No arbitrage..)



Second issue: everything is time-varying (2/3)

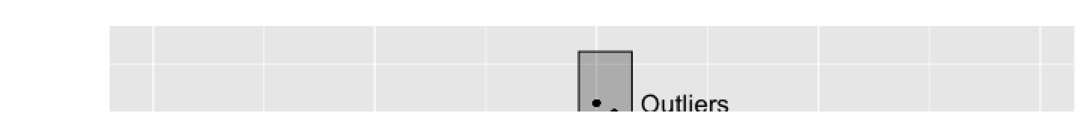
Second issue: everything is time-varying (3/3)

Any solutions?

- first, make sure your models evolve & react to new data! One natural inclination is to **fix** the model once & for all... that's a bad idea. Updating is key (though the details are far from obvious).
- second, think broadly. Does the **macroeconomy** help explain some variations? Stocks do not move out of nowhere... The credit spread may matter for the Size factor. There are many ways to integrate macro indicators in predictive models.

Third isse: algorithmic overfitting (1/2)

Big models $\neq$ better. (exaggerated version below)



Third isse: algorithmic overfitting (2/2)

Some **solutions** to overfitting:

- often, they are technical and algorithm-dependent. Penalization for regressions, trees and neural networks takes various forms. This requires a bit of practice.
- one heuristic tip is: avoid complex models!
- if you like sophistication, invest time in robustness checks (HP, time windows)

Fourth issue: backtest overfitting (1/3)

Imagine a quant fund manager with 5 modelling engines (Random Forests, Boosted Trees, Neural Nets, Recurrent NN, Ensemble).

Say, she wishes to test 10 HP values for each family of models (that's a pretty small number to evaluate sensitivity).
This makes 50 combinations. Then, assume there are 5 ways to translate predictive signals into portfolio weights.

→ In the end, this makes 250 options to build a strategy.

And I haven't even mentioned data pre-processing. 😉

Fourth issue backtest overfitting (2/3)

So the fund manager is going to **backtest** these strategies, that is, test them on data. Sadly, she only has past data at her disposal… And in the end, she is going to pick the one that works best (after robustness checks, sensitivity analyses, etc.)

Is this choice truly the best? Will it work well in live trading (out-of-sample)? Probably not, because the strategy is optimized on only one dataset! In factor investing, artificial data is hard to generate.

A rule of thumb: take the Sharpe ratio of the best strat and divide it by two to get a better estimate of what will happen

Fourth issue: backtest overfitting (3/3)

This relates to a crisis of reproducibility:

- Academic research is plagued by the "*publish or perish*" paradigm (with a strong bias towards positive results).
- Likewise, money managers are pressured to craft **winning strategies**, but can only backtest on past data.

→ we are pushed towards **false positives**. We so badly want to find recipes that succeed, that we end up forgetting the framework in which we work. Often, the best strategies perform well by chance! (one lucky random trajectory of the world).

Final words

Going further: the book!

It is located at http://www.mlfactor.com.

The material can be accessed at

https://github.com/shokru/mlfactor.github.io/tree/master/material

It includes a reasonably sized dataset & all the R codes (Python soon to come). Nothing will replace practice!

If you want a review of recent advances in ML & econometrics in financial economics, have a look at my html presentation

Wrap up: key takeaways

Factor investing aims to explain or predict financial returns with firms characteristics.

People tend to focus on sophistication, which sometimes does add value, but requires a huge amount of expertise. Often it is preferable to spend more time on **simple approaches** & have an integrated understanding of the models.

One major topic I left out is **causality**. It's incredibly important, but personally, I think it's still out of reach in factor investing.

Going forward & recommendations

XXXX

About this nexus

Many people think of machine learning as a **magic wand** (it's sophisticated, those who master it are smart & make money). **NO!**

The only way to make ML truly work, is by understanding the **environment** in which it is applied. For instance, this requires knowledge of **corporate finance** (factor investing), **market microstructure** (HFT), **lending industry** (credit scoring). → domain-specific expertise.

This is a prerequisite to making **enlightened modelling** choices (e.g., to choose the HPs).

For factor investing, **asset pricing models** (and financial economics more generally) are key.

THANK YOU!

What are your questions?