Learning to Trade

A Proposal for the 2019 NBER Big Data and High-Performance Computing Workshop

Anton Lines
Columbia University

1 Abstract

"We all expect machines to play very solid and slow games but AlphaZero just does the opposite. It is surprising to see a machine playing so aggressively, and it also shows a lot of creativity."

-- Chess Grandmaster Garry Kasparov, on DeepMind's general-purpose game-playing AI system

A striking feature of recent advances in deep learning is the ability of the resulting algorithms to behave in a noticeably humanlike fashion. Professional Chess and Go players have consistently expressed surprise at the ability of these algorithms to think "intuitively" instead of merely employing "brute-force" calculation. This feature of modern deep learning offers great promise for modeling and understanding human behavior in financial markets.

Several recent papers highlight the outsized role of demand-side factors in determining asset prices. Coval and Stafford (2007) show that mutual fund trading in response to investor capital flows generates significant, long-lived price movements, while Edmans, Goldstein, and Jiang (2012) find that these price movements influence real firm decisions. Koijen and Yogo (forthcoming) decompose stock returns into supply and demand factors, showing that demand factors account for 88% of the cross-sectional variation. Yet despite the importance of investor demand, we know relatively little about investors' decision processes and information sets. For instance, in Koijen and Yogo's analysis, fully 81% of cross-sectional return variance is attributable to unexplained variation in demand.

Asset holdings are ultimately the sum of many individual trading decisions. In this paper I take a deep-learning-based approach to estimating traders' policy functions, using daily institutional trade data from Ancerno combined with a very large dataset of market, accounting, news, and macroeconomic variables. I train deep neural networks to approximate the behavior of real-world market participants, by learning general representational heuristics that are validated out of sample. I then compare the estimated policy functions to a reinforcement-learning-based model of optimal trading behavior calibrated on the same dataset.

What are the relevant informational and noninformational factors that cause investors to buy, hold, or sell? How important are macroeconomic data compared to firm-specific events or characteristics? How does news interact with fundamental data in determining behavior? Which information events lead to trades that are ex-post rational? Existing work on such questions (e.g. Grinblatt and Keloharju (2001)) have relied on traditional econometric estimation methods such as logistic regression, and consequently have needed to pre-select the relevant explanatory factors. While some of these models are already quite large (up to 60

variables), the actual universe of potentially important financial variables can run into the tens of thousands. Moreover, the relationships among variables may be arbitrarily nonlinear. By contrast, the techniques employed in this paper allow simultaneous estimation of the relevant information set along with flexible estimation of agents' policy functions.

2 Research Design

The paper attempts to develop a quantitative general model of human behavior in financial markets using two distinct but complimentary approaches. The first approach is "top-down": using actual human trades to train a neural network to discover the decision heuristics underlying those trades. The model takes as input all financial, news and economic variables in CRSP, Compustat, RavenPack, FRED, and FOMC communications, for a given time period, and outputs a ternary buy-hold-sell decision. Ideally the policy function should be general, and applicable to all stocks at all times, allowing one to run hypothetical policy experiments.[1] For example, how would investors react to a short-selling ban under different market conditions? This pseudo-experimental approach has the potential to overcome a common criticism of laboratory studies, namely that traders may behave differently in artificial situations.

This exercise is similar to the well-trodden ground of image classification using convolutional neural networks; thus, fortunately, many of the challenges facing this project have already been widely studied and workable solutions found. It is important to design the training environment carefully to avoid overfitting the data or simply memorizing the training set. A key challenge is to design the optimal network structure—e.g., network type (feed-forward, convolutional, long/short term memory), number of layers, types of layers—to give rise to a suitable level of generality of the latent features underlying the classification process. Another potential issue is whether there is enough data to learn meaningful features without overfitting. Standard out-of-sample estimation techniques such as nested cross-validation require very large samples to be effective. The benefit of this approach is that learned features are often interpretable. For instance, features discovered by image classification networks can be visualized by translating neuron activity back into the input space (see, for example, Google's "Deep Dream" project).

The second approach is "bottom-up": designing an agent-based reinforcement-learning algorithm to learn how to trade effectively using the available data. The agent accumulates wealth through trading and evaluates this wealth according to a standard utility function (CRRA or Epstein-Zin). The algorithm can be restricted in ways that mimic human limitations on information processing (e.g. expectations can be limited to the near future, allocating attention to different data inputs can be made costly, etc.). The advantage of this approach is that additional data can be simulated from the empirical distribution of the variables in the input dataset, and thus limited training data is less of a concern. Indeed, such "self-play" data generation is crucial to the success of DeepMind's game-playing AI.

Computationally, the project will be coded in Python and make extensive use of tensor processing hardware. Currently I am using a single Nvidia GV100 chipset, but satisfactory learning performance may require significantly more extensive resources, such as those available via XSEDE.

[1] While the true policy function may be time-varying, it may not be possible to estimate such changes with the available data (trades from around 2000 to 2011).

References

Coval, J., & Stafford, E. (2007). Asset fire sales (and purchases) in equity markets. *Journal of Financial Economics*, *86*(2), 479-512.

Edmans, A., Goldstein, I., & Jiang, W. (2012). The real effects of financial markets: The impact of prices on takeovers. *Journal of Finance*, *67*(3), 933-971.

Grinblatt, M., & Keloharju, M. (2001). What makes investors trade? *Journal of Finance*, *56*(2), 589-616.

Koijen, R. S., & Yogo, M. (forthcoming). A demand system approach to asset pricing. *Journal of Political Economy*