Machine Learning in Trading
Things to Consider

Brian G. Peterson

updated 13 October 2017

Introductions

Who is this Guy ?

Brian Peterson:

- quant, author, open source advocate
- DV Trading Partner | Head Trader | Automated Trading Strategies
- Senior Lecturer (Part-Time), University of Washington Computational Finance and Risk Management
- author or co-author of over 10 packages for using R in Finance
- organization admin for **R**'s participation in Google Summer of Code

Proprietary Trading:

- proprietary or principal traders are a specific "member" structure with the exchanges
- high barriers to entry, large up-front capital requirements
- many strategies pursued in this structure have capacity constraints
- benefits on the other side are low fees, potentially high leverage
- money management needs to be very focused on drawdowns, leverage, volatility

So what is Machine Learning?

- after Artificial Intelligence models failed to produce human-like intelligence
- Machine Learning was adopted as the term to include a huge range of (mostly) non-parametric computational inference models
- the goal is to provide inference from data
- lives at the intersection of computer science and statistics

see:

- Domingos (2015) *The Master Algorithm*
- Efron & Hastie (2016) *Computer Age Statistical Inference*

for book-length treatments.

Data and Features

So it's all about the data?

- market data capture going from ~100GB/day to ~1-10+TB/day as more information sent by exchanges
- many vendors, including huge players like Reuters and IBM, are providing additional parsed data sources
- top users of ML in trading are collecting, curating, and cultivating rare or private data sources
- now common to have tens to hundreds of additional descriptive time series in addition to market data

THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

https://xkcd.com/1838/[1]

What about the Features?

- how do we go from descriptive to predictive?
- the raw data is usually not the main feature in time series prediction
- feature selection (Guyon and Elisseeff 2003) ranks and filters multiple possible features
- it is often better to first engage in pro-active feature engineering:
    - transformation of data to regularize scale, variation
    - coding of data into categorical or integer states
    - adding features that are indicative of the objective
    - using extra models for things like seasonality or colinearity
- many of the most successful users of ML spend significant time and money crafting features from data

About the Models

Model Basics I

Rules Engines

- rules created by experts have largely fallen out of favor (*or have they?*)
- knowledge engineering and expert systems of the 1980's
- perhaps knowledge encoded as features is the most important legacy of the expert systems

Gradient Descent

- gradient descent is the preference to move towards a local minima
- used widely in solving optimization problems
- the famous Perceptron (Rosenblatt 1958) is a gradient descent algorithm (and also the

Optimization

- almost all machine learning has its roots in optimization
- most ML models contain optimization solvers inside them
- global stochastic models can handle non-smooth objectives, … and *learn* or adapt as they gather more data

Policy Gradients

Model Basics II

Back Propagation

- adjusts previously established views based on new data
- the most famous models employing back propagation are neural networks
- 'deep learning' is a neural network with more layers, requiring bigger computers and more data
- one of the oldest true 'learning' approaches as the model adapts itself (Chauvin and Rumelhart 1995, Horikawa, Furuhashi, and Uchikawa (1992))

Evolution (Crossover & Mutation)

- features critical to 'learning' optimizers are crossover and mutation
- crossover is the mixing of features from more successful trials (W. M. Spears and others 1992, W. Spears et al. (1993))
- mutation randomizes features (even if successful) to avoid local minima

Similarity

- similarity between training set and sample used to draw inferences

Probabalistic Inference

A big computer, a complex algorithm and a long time does not equal science. - Robert Gentleman

Using Machine Learning in Trading

Regression

- regression results may be the most common uses of machine learning
- you want the algorithm to estimate the value (mean/median), slope, etc. of some variable
- you may do this directly (e.g. logistic regression, Bayesian regression via Stan, Random Forest, Boost)
- or indirectly (MCMC, gradient descent, state space model, optimization model with the correct objective)
- in many cases, simpler regression models (e.g. dlm, ridge regression, LASSO) may do just as well

Optimization

- not just a model feature, optimization is often the end goal
- consider whether simpler linear/quadratic/gradient optimizers will work – less model risk
- optimizer parameters for global optimizers can have huge impacts on model outputs and computation
- many problems may be recast as optimization problems:
    - capital allocation, see PortfolioAnalytics
    - risk management, see PortfolioAnalytics and PerformanceAnalytics
    - adaptive parameters
    - (optimal) process control, see gym and MDPtoolbox

Classification / Clustering

- separation of observations into groups or states is classification or clustering (Japkowicz and Shah 2011)
- many ML models are used as classifiers
- proceeding from simple models is usually advisable
- key goal is typically to *label* the observed data or state
- so that some other model can take action based on that label/state
- common classification uses:
    - volatility regimes, e.g. Markov Switching GARCH
    - country/sector/volatility clustering
    - bull/bear market classification
    - trending/consolidating market

Composite Models

- lots of little models
- ensembles, boosting, weak learners
- in trading, this means mostly feature engineering
- or combining weak alpha trading models
- Worldquant (Tulchinsky 2015), Kepos (Carhart/Litterman), Jump, Renaissance, Bridgewater, Citadel, Two Sigma, others all doing this to one degree or another

Doing this well takes scale!

Model Discovery

- the 'Holy Grail' of ML in trading

Like the Grail, mostly a myth

- feature engineering is what makes models work
- many good features and data sets make model 'discovery' possible
- focus on curating your data, building features
- with enough data, models, and features, you have broad choice of algorithms

Compute and Data Implications

Compute for Research

- choose a technology stack to build on
  - R, python
  - libraries such as TensorFlow, AzureML
  - cluster architecture (OpenStack, AWS, Azure, Google Compute)
  - use specialized hardware where appropriate (e.g. GPU, TPU, ASIC)
- normalize all data access and metadata
  - specialized time series databases (e.g. InfluxDB, Timebase, OneTick, kdb, Streambase)
- catalog all the data, version it
- build a process for continuously evaluating models (Breck et al. 2016)

Cloud vs. Premise for Research

- cloud compute is generally cheaper than premise
- cloud libraries like AzureML or Tensorflow may speed results

but you need to get your data into the cloud

- cloud data can be much more expensive than on-premise
- cloud data is charged for storage and access, so access patterns need to be optimized
- huge opportunity for vendor cloud delivery of data sources (e.g. Tick Data, OneTick, Deltix, QuantGo, NASDAQ, Thesys)

consider an on-premise stack that is cloud-scalable

Compute for Production

- "targeted" machine learning is all around us (e.g. voice recognition, face recognition)
- some of the most spectacular advances for large scale ML have been games (Jeopardy, Chess, Go, DOTA2)
- those wins had huge custom built computers behind them, doing online ML

- most uses of ML in Trading are *offline*,
    - models trained on large data, clusters, GPU's
    - model parameters fixed for 24 hours or more
- or use a simplified online model that does not refit model parameters

Case Study

Hierarchical Hidden Markov Model

Bayesian Hierarchical Hidden Markov Models applied to financial time series (Damiano, Peterson, and Weylandt 2017)

based on **Regime Switching and Technical Trading with Dynamic Bayesian Networks in High-Frequency Stock Markets** (Tayal 2009)

Problem: predicting price trends systematically in a profitable way

Stylized Facts:

- Market behavior is complex and partially unknown
- Non-linear interactions between price and volume
- Multi-resolution: short-term trends within long-term trends
- High-frequency: noisy and large data sets need fast online computations

A Proposal: ensemble of statistical and machine learning techniques

1. Create intermediate indicator variables
2. Combine them into discrete features using technical analysis rules
3. Build a hierarchy to link all the features in a logical way
4. Apply clustering with Markovian memory

How to make features useful?

[…] some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used. (Domingos 2012)

A plain **HMM** is a probabilistic learner of the unobserved dynamics behind a sequence of discrete features. The Tayal **HHMM** maps the features and the corresponding learned states into bull and bear markets by establishing a hierarchy.

The strengths of the features:

- **Underlying theory**: representative of our beliefs about how markets work (interactions between price and volume)
- **Empirical support**: when applied on real data, results are consistent with empirical evidence
- **Statistical properties**: captures non-linearities in a simple, parsimonious, and tractable way
- **Noise reduction**: by discretization
- **Computational complexity**: improved by reducing data set size

We test the model out of sample. After one feature (a "zig-zag") ends, we predict the next state. We buy one unit in bull states and sell one unit in bear states.

Feature Detail

(1) Identify local extrema, where e_n is the price at the extreme.

(2) Create intermediate variables ν, $\tilde{\nu}$ and features f_n^0 direction, f_n^1 price trend, f_n^2 volume trend.

$$f_n^0 = \begin{cases} +1 & \text{if } e_n \text{ is a local maximum (positive zig-zag)} \\ -1 & \text{if } e_n \text{ is a local minimum (negative zig-zag),} \end{cases}$$

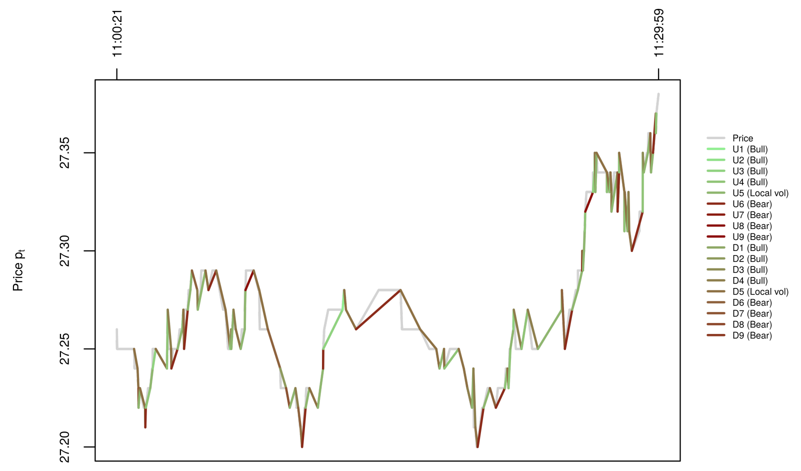
$$f_n^1 = \begin{cases} +1 & \text{if } e_{n-4} < e_{n-2} < e_n \wedge e_{n-3} < e_{n-1} \text{ (up-trend)} \\ -1 & \text{if } e_{n-4} > e_{n-2} > e_n \wedge e_{n-3} > e_{n-1} \text{ (down-trend)} \\ 0 & \text{otherwise (no trend).} \end{cases}$$

$$\nu_n^1 = \frac{\phi_n}{\phi_{n-1}}, \quad \nu_n^2 = \frac{\phi_n}{\phi_{n-2}}, \quad \nu_n^3 = \frac{\phi_{n-1}}{\phi_{n-2}}, \quad \tilde{\nu}_n^j = \begin{cases} +1 & \text{if } \nu_n^j - 1 > \alpha \\ -1 & \text{if } 1 - \nu_n^j > \alpha \\ 0 & \text{if } |\nu_n^j - 1| \le \alpha, \end{cases}$$

$$f_n^2 = \begin{cases} +1 & \text{if } \tilde{\nu}_n^1 = 1, \tilde{\nu}_n^2 > -1, \tilde{\nu}_n^3 < 1 \text{ (volume strengthens)} \\ -1 & \text{if } \tilde{\nu}_n^1 = -1, \tilde{\nu}_n^2 < -1, \tilde{\nu}_n^3 > -1 \text{ (volume weakens)} \\ 0 & \text{otherwise (volume is indeterminant).} \end{cases}$$

(3) Combine into 18 meaningful features using the following table. They will be linked hierarchically by the model.

Feature	Zig-zag	Price trend	Volume trend	Market State	Feature	Zig-zag	Price trend	Volume trend	Market State
U_1	Up +1	Up +1	Strong +1	Bull	D_1	Dn -1	Up +1	Weak -1	Bull
U_2	Up +1	Dn -1	Strong +1	Bull	D_2	Dn -1	Dn -1	Weak -1	Bull
U_3	Up +1	Up +1	Indet 0	Bull	D_3	Dn -1	Up +1	Indet 0	Bull
U_4	Up +1	No 0	Strong +1	Bull	D_4	Dn -1	No 0	Weak -1	Bull
U_5	Up +1	No 0	Indet 0	Local	D_5	Dn -1	No 0	Indet 0	Local
U_6	Up +1	No 0	Weak -1	Bear	D_6	Dn -1	No 0	Strong +1	Bear
U_7	Up +1	Dn -1	Indet 0	Bear	D_7	Dn -1	Dn -1	Indet 0	Bear
U_8	Up +1	Up +1	Weak -1	Bear	D_8	Dn -1	Up +1	Strong +1	Bear
U_9	Up +1	Dn -1	Weak -1	Bear	D_9	Dn -1	Dn -1	Strong +1	Bear

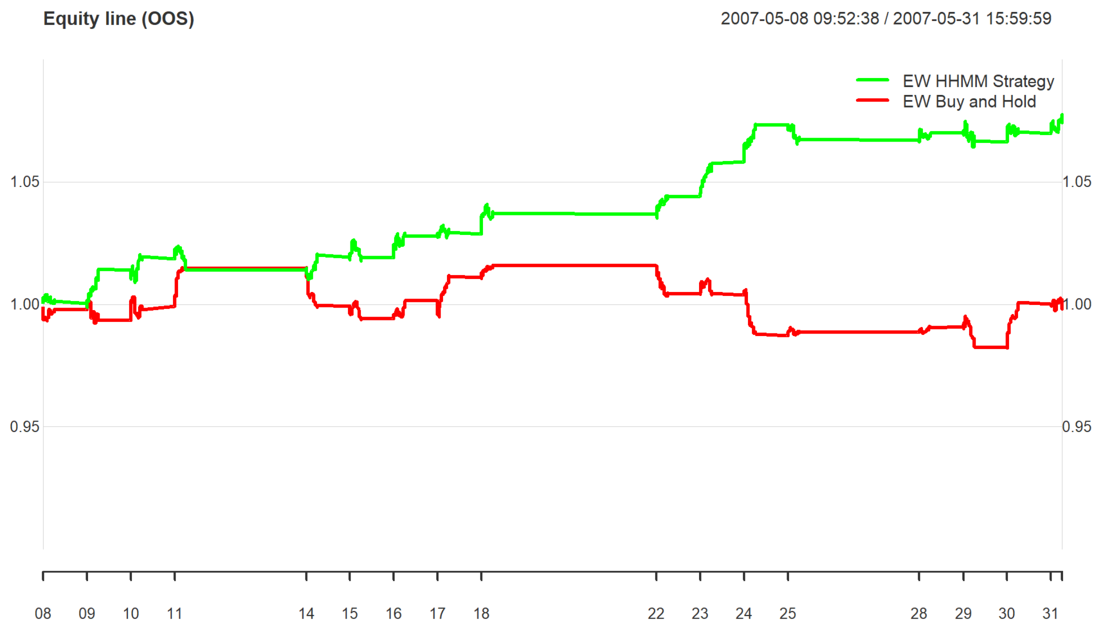


Features extracted from TSE:G 2007-05-11 11:00:00/2007-05-11 11:30:00.

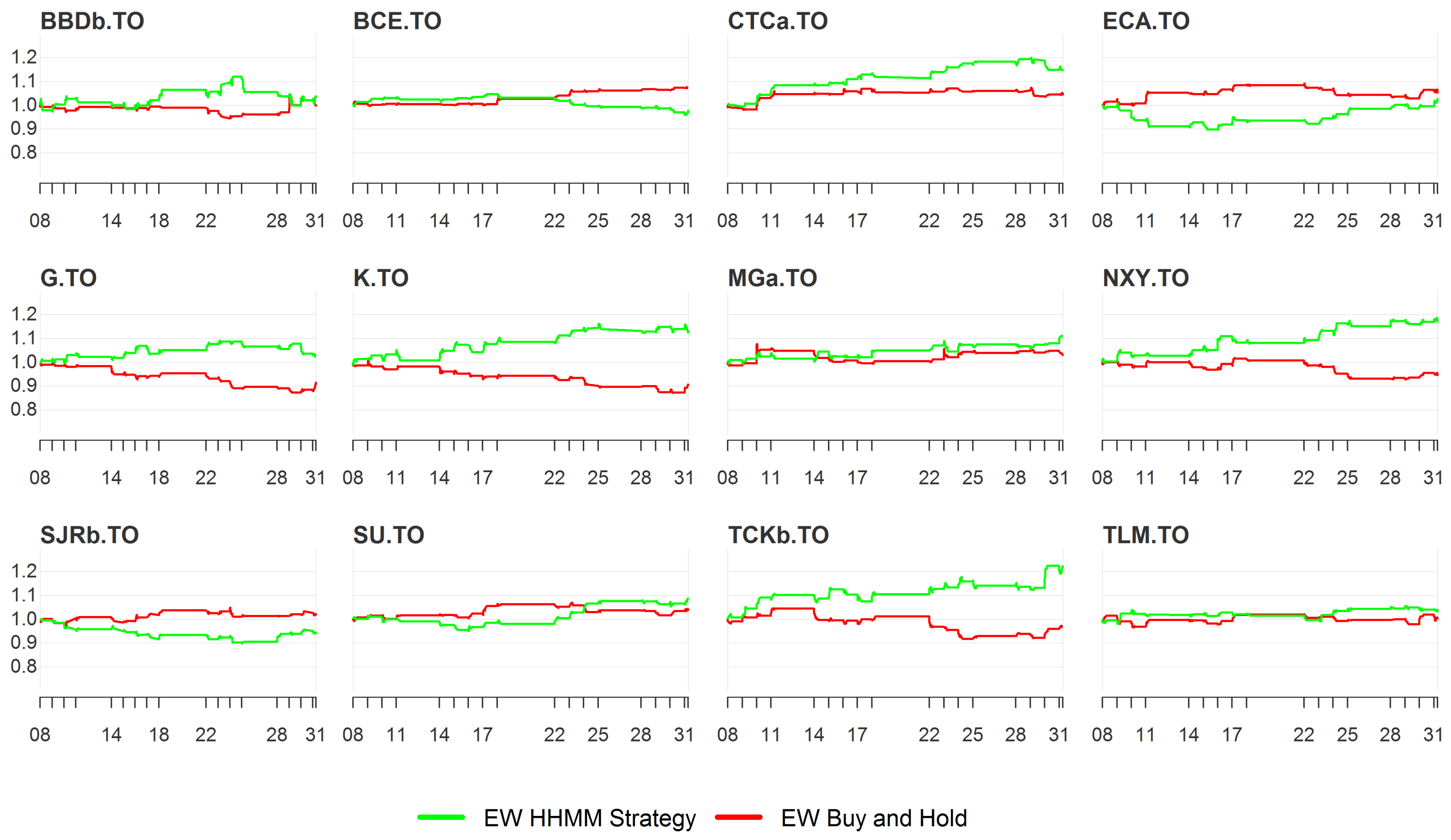
Results

Back tested on 12 stocks, 17 days, 7 configurations: $12 \times 17 \times 7 = 1,428$ out of sample daily returns.

- For most stocks, HHMM outperforms buy & hold (B&H).
- Returns virtually uncorrelated with B&H.
- Sometimes HHMM offers less variance than B&H (further research needed).



Per-Instrument Results



Thanks

Thank You for Your Attention

Thanks to Luis Damiano for his work in GSoC2017, my team, and my family, who make it possible.

Code to apply the techniques discussed here may be found in the *mlr*, *TensorFlow*, *rstan*, *blotter*, *quantstrat*, *PerformanceAnalytics*, *PortfolioAnalytics*, and other **R** packages.

Resources

Breck, Eric, Shanqing Cai, Eric Nielsen, Michael Salib, and D Sculley. 2016. "What's Your Ml Test Score? A Rubric for Ml Production Systems." In *Reliable Machine Learning in the Wild-Nips 2016 Workshop*. https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45742.pdf.

Chauvin, Yves, and David E Rumelhart. 1995. *Backpropagation: Theory, Architectures, and Applications*. Psychology Press.

Chen, Yihua, Eric K Garcia, Maya R Gupta, Ali Rahimi, and Luca Cazzanti. 2009. "Similarity-Based Classification: Concepts and Algorithms." *Journal of Machine Learning Research* 10 (Mar): 747–76.

Cooper, Gregory F. 1990. "The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks." *Artificial Intelligence* 42 (2-3). Elsevier: 393–405.

Damiano, Luis, Brian G Peterson, and Michael Weylandt. 2017. "Bayesian Hierarchical Hidden Markov Models Applied to Financial Time Series." *GitHub Repository*. https://github.com/luisdamiano/gsoc17-hhmm; Google Summer of Code 2017.

Domingos, Pedro. 2012. "A Few Useful Things to Know About Machine Learning." *Communications of the ACM* 55 (10). ACM: 78–87.

———. 2015. *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Penguin Books.

Efron, Bradley, and Trevor Hastie. 2016. *Computer Age Statistical Inference*. Vol. 5. Cambridge University Press.

Guyon, Isabelle, and André Elisseeff. 2003. "An Introduction to Variable and Feature Selection." *Journal of Machine Learning Research* 3 (Mar): 1157–82. http://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf.