Machine Learning in Finance

Ernie Chan, Ph.D.
QTS Capital Management, LLC.

About Me

- Principal, QTS Capital Management, LLC.
  - Commodity pool operator and trading advisor
- Author
  - Machine Trading (Wiley 2017)
  - Algorithmic Trading (Wiley 2013)
  - Quantitative Trading (Wiley 2009)
- Adjunct Faculty
  - Northwestern U. MS in Data Science program.

My Journey

- ML -> Finance -> ML.
- Started at IBM T. J. Watson Research Center
  - Natural language processing researcher (NLP)
- Continued at Morgan Stanley's Data Mining and AI group, Credit Suisse equity prop trading, ...
- Gave up ML because it hadn't worked in trading.
- Focused on "simple" (few predictors) strategies to trade for myself (and my investors).

My Journey

- “Simple” strategies very profitable from 2006-2015!
  - “Simple”=Traditional quant models, as advertised in finance journals.
- Many (but not all) such alpha strategies stopped working circa 2016.
  - Smart beta strategies continued to work, of course.
- Created many new “simple” alpha strategies 2016-7, but few are profitable in live trading.
- Back to ML with new understanding and new tricks!

Pros of ML

- Single-predictor strategies have been losing alpha.
  - "… quantitative trading became more challenging with every passing year." -D.E. Shaw, as told to Prof. Andrew Lo in *Adaptive Markets, 2017*.
  - Barclay FX index (10 largest investable currency trading programs): -5.44% in 2016, -10.77% in 2017, and -3.70% YTD.
  - Many obvious market inefficiencies have been discovered by discretionary traders, traditional quants, and academic researchers.

Pros of ML

- Combining multiple predictors is non-trivial.
  - Naïve multiple linear regression likely result in overfitting.
  - Features selection is necessary.
    - E.g. LASSO (l_1 regularization)
    - Drop features with small "regression coefficients"
  - Too many ways to combine predictors nonlinearly.
  - Efficient search for nonlinear models is part of ML!

Question

- What are l_1 and l_2 regularizations?
- Why does only l_1 gives sparse solutions in linear regression?

Answer

- l_1 norm is sum of **absolute value** of regression coefficients.
- Regularization means penalizing this norm in fitting: many coefficients will have to zero, achieving "sparseness".
- l_2 norm is sum of **square** of regression coefficients: many coefficients can be simultaneously small but non-zero: sparseness not achieved.
- E.g. $||(1,0)||_1$=1 < $||(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}})||_1$=$\sqrt{2}$, but $||(1,0)||_2$=1 == $||(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}})||_2$.
- See Kevin Murphy, "Machine Learning: A Probabilistic Perspective", section 13.3.1.

Pros of ML

- ML does not have to be "black box".
  - Features are created by humans.
    - Features incorporate market knowledge.
  - Some ML models have intuitive rules.
    - E.g. classification tree may say "If 20-day MA > 40-day MA, and VIX < 15, and Close > Open; then Buy!"
  - ML can be used as risk management layer: only accepts/rejects trades from another model.
    - "Quantamental" approach.
  - ML can be used as capital allocator.

Question

- Suppose we have a classification tree to decide whether we should buy or short SPY, given a whole bunch of technical indicators as features.
  - What predictive output can be used to determine capital allocation to a trade?

Answer

- Classification tree generates probability of each class in addition to the most probable class.
- This probability can be used as capital weight.

Pros of ML

- Many choices of learning algorithms.
  - “Smart” linear models such as LASSO or stepwise regression
  - CART/RF
  - SVM
  - Neural networks
    - Feedforward
    - Time-delay
    - LSTM (“Deep learning”)
    - CNN (“Deep learning”)
  - Reinforcement learning
  - Genetic programming
  - Bayesian network

Pros of ML

- Many tricks to improve learning efficiency and generalizability.
  - Reducing both “bias” (underfit) and “variance” (overfit).
  - Reducing both “Type I” (false positives) and “Type II” (false negatives) errors.
- Bagging, cross validation, boosting, dropout, early stopping, “deep” learning, regularization, …

Cons of ML

- Overfitting is very easy.
  - Many more parameters than traditional quant models.
- Need *a lot* of data to work.
- Data and features engineering requires strong programming skills.
- Surprise: extremely labor-intensive to develop a ML model!
- When it fails, we usually cannot explain why.

The Upshot

- It is hard and time-consuming to do, and it may not work.
- But we have **no choice** but to start using ML.
- Just as truck driving as a profession is doomed due to self-driving trucks and drones, quant trading is doomed without ML.
  - Discretionary trading may still work, but it can still be improved by apply ML (“Quantamental” models).

Where to Start? [Data]

- Data needs to voluminous, multi-dimensional, and clean.
  - E.g. unlikely you can learn real patterns from daily prices of a single stock.
  - Model need to learn from data accumulated from multiple stocks.
  - Features can be created from various “technical” indicators (functions of prices), “fundamental” (functions of non-price) indicators, and “alternative” data (e.g. news).

Data

- General rule: You get what you pay for.
  - Cheap data is seldom good, good data is seldom cheap.
  - Without good data, models are severely compromised (see petewarden.com)
  - Especially important and expensive for equity models
    - Survivorship bias
    - Not using bid-ask or auction prices at open/close.

Data

- First step in any ML models: check data integrity.
  - Noisy/wrong data?
  - Missing data?
  - Time stamps integrity?
    - E.g. Often earnings announcement dates are not Point-In-Time!
    - (Companies revise expected announcement dates up till day of expected announcement.)
  - Look ahead bias in features?
  - Are features stationary?
    - E.g. Cannot use price as features.
    - Need “fractional differentiation”.
  - Are features synchronous?
    - E.g. Cannot simply combine daily closing prices of stocks with futures and options.

Questions

1. What is the problem with using, say, I/B/E/S earnings announcement dates for backtesting an event-driven model?
2. Is return a stationary variable? Is it suitable as feature?
3. Can you use closing value VIX index on day t as feature to trade SPY at stock market close of same day.

Answers

1. I/B/E/S historical earnings data only tells you the actual announcement date, not the expected one.
   – We often have to enter trade based on expected date.
2. Returns are stationary – suitable as feature.
3. VIX index closing value is obtained at 16:15 ET. SPY closes at 16:00 ET.

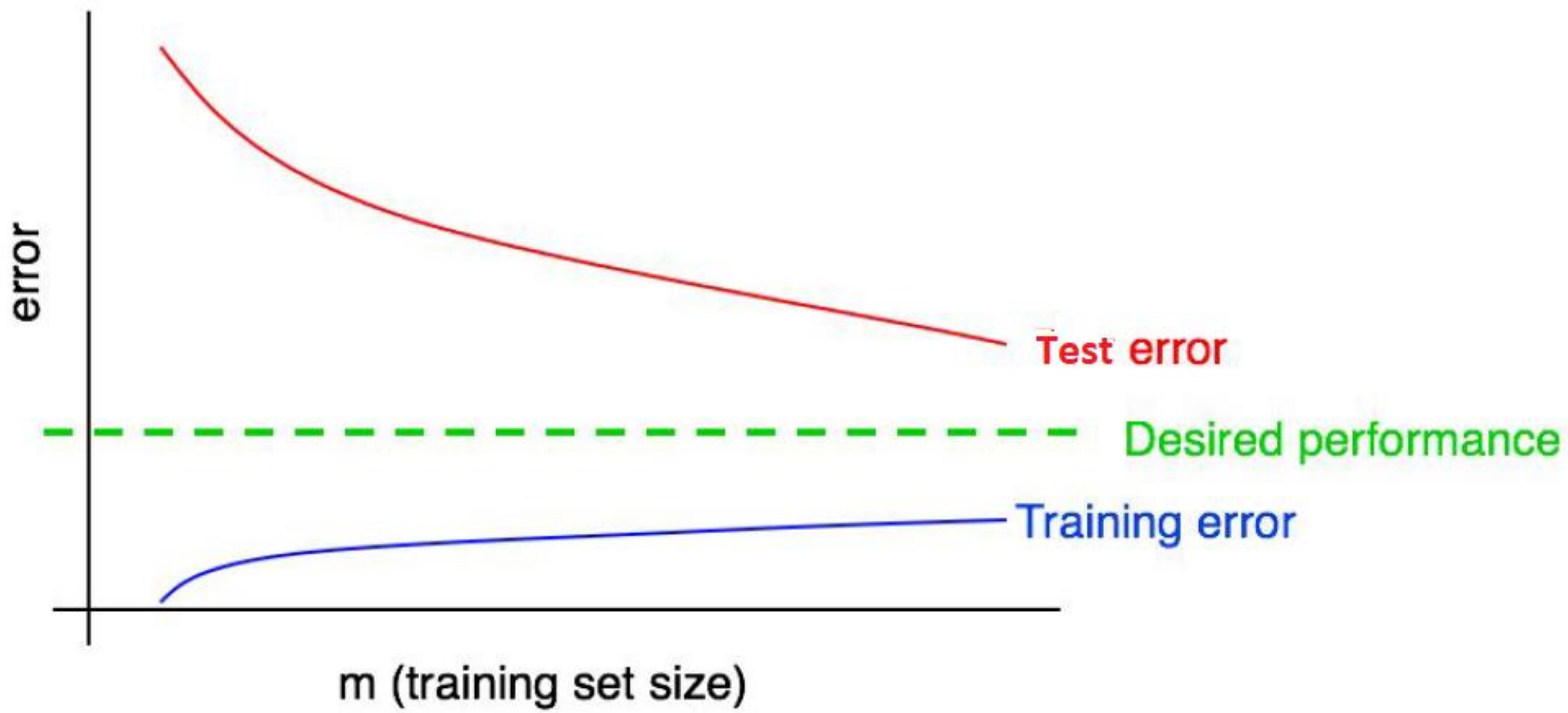
Data

- Non-price data: especially tricky to ascertain quality.
  - Best to extract features from raw data (e.g. news from Thomson Reuters) instead of relying on 3rd party sentiment models.
- More data reduces “variance”
  - But how much is enough?
  - Plot “learning curve”

Questions

1. Does more data also reduce “bias”?
2. What is a “learning curve”?

Learning curve



Source: Andrew Ng, “Machine Learning Yearning”

Features Engineering

Raw data $\xrightarrow{\quad function \quad}$ Features

Question: What possible $function$ can you think of?

Examples of *function*

1. Lagged values, lagged differences.
2. Technical indicators.
3. Signal processing (e.g. Fourier transform).
4. Time series models coefficients.
5. Parametric distributions parameters.
    a) E.g. higher moments
6. Products and $function$ of $function$.

Features selection

- Too many features reduce bias but increase variance.
  – Lead to overfitting.
- Many methods of features selection.
- Question: which features selection method we have already discussed?

Answer

- We already discussed l_1 regularization.
  - That is mainly applicable to linear models.
- We also discussed l_2 regularization.
  - That is used in deep learning.

Features selection

- Stepwise regression
  - Add features one at a time to maximize BIC.
  - Drop features one at a time to maximize BIC.
- MDA: mean decrease accuracy
  - Randomizes (permutes) the values of one feature at a time, and note how much this decreases out-of-sample predictive accuracy.
  - Larger the decrease in accuracy -> higher feature importance.

Where to Start? [Algo]

- CART/RF is the first tool I used for ML in finance.
  - Classification And Regression Trees.
  - Random Forest (through bagging and random subspace).
- CART/RF is still the most useful and easy-to-learn tool.
- Pick your language: Python (Scikit-learn), Matlab (Statistics Toolbox), R all have packages that implement CART/RF.
- Easy to pick up, but hard to use correctly.
  - Numerous subtleties in training, validating, hyperparameter optimization, features selection, etc.

Where to Start? [Books]

- Learn CART and other ML algos from chapter in my book *Machine Trading*.
- Read machine learning theory from Kevin Murphy's *Machine Learning: A Probabilistic Perspective*.
- Then read their multitudinous subtleties in Marcos Lopez de Prado's book *Advances in Financial Machine Learning*.
- Then read *Deep Learning with Python* by François Chollet and *Machine Learning Yearning* by Andrew Ng.
- Then follow Twitter feeds (*@chanep, @ml_review, @mxlearn*), and the voluminous and ever-increasing ML literature cited.

Where to Start? [People]

- Focus on your comparative advantage: features creation.
  - Hire programmers, data scientists, quants… as consultants, interns, etc.
  - Hire college students: they are more familiar with ML than us!
- A good team forces you to
    - Confront your short-comings, blindspots.
    - Extend your strategies.
    - Improve your strategies.
    - Improve your research rigor.
- Best quant funds use a team (“production chain”, “assembly line”) approach.
  - E.g. RenTec, Two Sigma, PDT Partners.
  - “Advances in Financial Machine Learning” by Marcos Lopez de Prado.

Final Word

- Use ML to predict things that are not subject to "reflexivity".
  - Hint 1: returns is reflexive.
  - Hint 2: realized volatility is not.
  - Hint 3: Weather in the Midwest is not.
  - Hint 4: Gasoline consumption is not.

Final Word

- To succeed in ML, you need to throw EVERYTHING including the kitchen sink at a problem.
  - Very different from math, physics, or even finance/economics/trading research where an elegant insight/intuition can carry the day.
  - Accumulation of small, incremental improvements is the norm in ML research.

Thank you for joining us!

www.epchan.com
Blog: epchan.blogspot.com
Twitter: @chanep