Chris, Hodapp
chodapp3@gatech.edu

2016-08-25, CS7646 (Machine Learning for Trading)

To read:
Python for Finance, Ch. 4 & 6

Intro

- Tucker Balch & Devpriya Dave (not that Tucker is co-author on 'What Hedge Funds Really Do')
- Three Mini-Courses
  - Manipulating Financial Data in Python
  - Computational Investing
  - Machine Learning Algorithms for Trading

01-01 (Reading & Plotting Stock Data)

- We use Python in this course (with numpy, pandas, & matplotlib) though R & MATLAB are also good choices
- Our data sources are mostly CSV (and from Yahoo)
  e.g. HCP.csv has columns:
  - Date
  - Open (what was price when markets opened in morning?)
  - High (high over whole day)
  - Low (low " " ")
  - Close (what was final price at close of markets at 4 pm)
  - Volume (how many shares traded altogether that day)
  - Adj. Close (adjusted close - more on that later)
- Adjusted close: Data provider computes this to account for things like splits & dividend payments (usually, equals close)
- Pandas, the Python library, was made at a hedge fund.
  Its main datatype is the dataframe:
  - Typical for us: One column for each stock symbol, one row for each date
  - If we see NaN, that's the Pythonic way of showing that data is missing.
  - Pandas gives another dimension too, in additions to rows/columns!
    suppose we wanted Close, Volume, Adj. Close
- To read CSV in pandas: read_csv
  Show just top few lines of dataframe df: df.head()
  Likewise, df.tail() for last few values.
  Indexing works like normal Python lists.

plt = matplotlib.pyplot.

- Plot from dataFrame with: [df(...) .plot()
  plt.show
- 2 columns: [df[['Close', 'Adj Close']] for instance

01-02
Working with
Multiple Stocks

- Pandas can join, like in a database
  - e.g. df1 has 2010-01-22, 2010-01-23, 2010-01-24, 2010-01-25, 2010-01-26 while dfSPY has 1993-01-12 .... 2012-12-31 but only trading days (which excludes weekends - which 2010-01-23 & 24 were).
    - We want just the rows of df1 also in dfSPY based on date.
    - Need a DateTimeIndex for this:
      [dates = pd.date_range('2010-01-22', '2010-01-26')
        df1 = pd.DataFrame(index=dates)
    - df1 has no columns, just an index.

SPY
= S&P 500
index
(roughly)
= really ETF

  - Now... [dfSPY = pd.read_csv("data/SPY.csv")
             df1 = df1.join(dfSPY)
    This fails; dfSPY needs the right index. So:
    [dfSPY = pd.read_csv("data/SPY.csv", index_col = "Date",
                          parse_dates = True)
    If we wanted only certain columns, see 'usecols' keyword of read_csv
    Also add: na_values = ['nan'] & df1 = df1.dropna()
  - Or: df1.join(dfSPY, how = 'inner') as this is just an inner join.
    It does other types too.
- If we want to add other stocks we must rename columns so they don't conflict. See 'rename' method.

Is this
wrong?

- To just slice rows (for instance): df[sd:ed]   (sd → start date, ed → end date)
  Both rows & columns: df[sd:ed, ['GOOG', 'GLD']] for GOOG&GLD cols
  - More on slicing when we cover NumPy
- Can also do: df.ix['2010-01-01':'2010-01-31'] e.g. for 2010 Jan.
  - see DataFrame.ix

Chris Hodapp
chodapp3@gatech.edu

2016-08-25b, CS7646 (ML for Trading)

01-02 (cont.)

- For column slicing: df['GOOG'] for one
  df[['IBM', 'GLD']] for multiple
- ix can also do row & column slicing:
  df.ix['2010-03-10': '2010-03-15', ['SPY', 'IBM']] for instance
- You may still plot from these too.
- If we wanted to normalize so all prices start at 1:
  [df = df / df.ix[0] (or df.ix[0, :] to be explicit)
  Pandas can vectorize this just fine, though you can do it with a for-loop.

01-03 (Power of Numpy)

- Numpy focuses on matrices (ndarrays) and using underlying C & FORTRAN routines
  - I'm going to skip most of this...
  - Creating NumPy array...
    - From Pandas dataframe: pandas.DataFrame.values
    - From Python seq: numpy.array
    - from constant vals: ones, zeros; from random: random
  - Array attribs: shape, ndim, size, dtype
  - Stats: sum, min, max, mean; arithmetic: add, subtract, multiply, divide
  - Exec. time: time.time, profile
  - Manipulation via indices/slices, integer arrays, boolean arrays
- Pandas dataframe is just a wrapper around an ndarray
  - But a lot of numpy methods work directly on the dataframe without 'values' explicitly being called.

01-04 (Statistical Analysis of Time Series)

- See mean, median, std, sum, rolling_mean, rolling_std in Pandas
  - df.mean does mean of each <u>column</u>; likewise median, std, & others.
- <u>Rolling statistics</u>: Take a snapshot over windows, rather than a global measure
  - e.g. 20-day rolling mean: Compute mean over every 20-day window. This has the effect of smoothing (and lagging a little)

"simple moving average"

01-04 (cont.)

- One thing technical analysts look for: When the rolling mean intersects the price, particularly if price is on its way down ?
  - Hypothesis: Rolling mean is a better estimate of "real" price of stock, and deviation far enough is a mis-pricing.
    (But how far of a deviation is significant?)
    - We'd use rolling standard deviation to check that.

Is it a "trading signal"?

Tucker doesn't necessarily believe this.

It's trademarked.

- Bollinger bands®- created in '80s.
  If stock is very volatile, perhaps we ignore its movements a bit; if less volatile, they're more significant.
  His idea was to plot 2 standard deviations above & below the moving average, and consider the price being over 2 stdevs above average as a 'sell' signal, and below 2 stdevs below average as a 'buy' signal.
  → The professor doesn't advocate this (it doesn't always work), but gives it as an example.

Or this.

- See rolling_mean in Pandas.

Apparently now deprecated in Pandas. See...
pd.rolling(...).mean()
" " .std()

- Daily returns: How much did price go up or down each day? (It's just $\frac{price[t]}{price[t-1]} - 1$)
  - A single numpy statement handles it (plus pandas glue):

```
daily_returns = df.copy()
daily_returns[1:] = (df[1:] / df[:-1].values) - 1
daily_returns.ix[0,:] = 0
```

    - Other parts keep overall length the same

- Another important statistic is cumulative returns:
  cumret[t] = (price[t] / price[0]) - 1
  - For return through time t

typ. both are percents

01-05 (Incomplete Data)

- One reason machine learning is often used with financial data is that that data is very meticulous and well-documented.
- However, it can still be faulty. It can be wrong & it can have gaps. We may have multiple sources of data that aren't precisely in sync (especially minute-by-minute price data from multiple exchanges). Not all stocks trade every day too.

Chris Hodapp
chodapp3@gatech.edu

2016-08-25c, CS7646 (ML for Trading)

01-05 (cont.)

- Usually we use SPY under the assumption that if the market's open, SPY is being traded.
- Example of gaps in data: Sun Microsystems traded under name 'JAVA', then Oracle bought Sun in 2010 & its symbol disappeared.
  Prior to Sun being public, 'JAVA' was Mr. Coffee, also...
- It's fairly common to have stock data that only starts at some date, and ends at another date.
  Smaller, less-traded stocks might also have multiple gaps.
- Why not interpolate between gaps? That's not really correct either (it didn't trade in those gaps)
  The 'proper' way (e.g. for rolling means) is to fill the past data forward in time.
  If we use 'future' data at all, this is sort of cheating, and interpolation would do this. Thus, all we can do is use past data for that point.
  - But what about missing data at the beginning? We fill backwards here, for lack of a better method.

~~So fill~~ ... then fill forward.

- See 'fillna' in Pandas, e.g. fillna(method = 'backfill')
- Fill forward, then fill backward - as a general rule.

01-07 (Sharpe ratio & other portfolio stats)

To read: Python for Finance, Ch. 5 & 11

- Our focus now is on statistics over a portfolio of stocks, not single stocks. (Portfolio = allocation of funds to a set of stocks, and we assume a buy-and-hold strategy)
  - e.g. maybe 40% SPY, 40% XOM, 10% GOOG, 10% GLD (or 0.4, 0.4, 0.1, 0.1) - these are allocations at start of portfolio.
- To compute from dataframe:

```
normed = prices/prices[0]            (now prices start at 1.0)
alloced = normed * allocs            (alloced now starts: 0.4,0.4,0.1,0.1)
pos_vals = alloced * start_val       (= position values, $ each day per asset)
port_val = pos_vals.sum(axis=1)      (total portfolio value)
```

01-07 (cont.)

- Other statistics on portfolios that are important:
  - Daily returns (but always ~~[illegible]~~ ignore 1st day)!
  - Cumulative returns: cum_ret = (port_val[-1] / port_val[0]) - 1
    - It's just a measure of how much a portfolio has increased from the start
  - Average daily returns: avg_daily_ret = daily_rets.mean()
  - Standard dev. of daily returns: std_daily_ret = daily_rets.std()
  - Sharpe ratio is a little more involved.

Though I'm sure Taleb has things to say on this.

- The idea of Sharpe ratio: Consider rewards relative to risk, which they (try to) measure with volatility \ standard dev.
  - It is "risk-adjusted return".
  - All else equal: lower risk is better, higher return is better.
  - It uses 'risk free rate of return' too, like a bank account or short-term treasury bond. As of now this is basically 0% because interest rate is so low.
- Rough formula, given — R_p - portfolio return
  R_f - risk-free rate of return
  σ_p - stdev. of portfolio return

$$= \frac{R_p - R_f}{\sigma_p}$$

  - Higher value is better. It's only negative if R_f has a higher return than the (probably riskier) R_p.
  - More explicitly: $\underset{=SR}{S} = \frac{E[R_p - R_f]}{Std[R_p - R_f]}$ — This is "ex ante", looking back at old values
    but R_f is constant, so $Std[R_p - R_f] = Std[R_p]$.
    (well, R_f isn't always constant...)
  - Note that R_f is daily risk-free rate here. If we have only annual interest rate, we can convert this, e.g. 10% interest/year, thus 1.0 at start gains 0.1, so for 252 trading days, $R_f = \sqrt[252]{1+0.1} - 1$
- But... Sharpe ratio can vary widely based on when you sample prices, so it requires an adjustment.

"Annualized" Sharpe ratio

$SR_{annualized} = K * SR$ where $K = \sqrt{\#\ samples/year}$, e.g. $\sqrt{252}$ for daily on trading days only

Chris Hodapp
chodapp3@gatech.edu

2016-08-26, CS7646 (ML for Trading)

01-07 (cont.)

- Example: 60 days of data on portfolio
  Average daily return = 10 bps = 0.001 (1 bps = 1 "bip" = 1 basis point = $\frac{1}{100}$ %)
  Daily risk free = 2 bps = 0.0002
  Stdev. daily ret = 10 bps = 0.001
  - So: $SR_{annualized} = \sqrt{252}\ \frac{10-2}{10} \approx 15.9 \cdot 0.8 \approx 12.7$

  (a really good ratio, but numbers are made-up)
  Note that we use 252, not 60. 252 is still the frequency.
  Units cancel, so we need not convert basis points to percents/ratios.

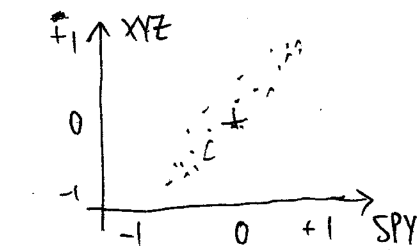
01-06 (Histograms & Scatter plots)

- Info on single stock is less useful. Returns of stock against each other can tell much more.
- Histograms of daily returns can also tell a lot (and supposedly they'll typically resemble a normal distribution) | Taleb would love this.
  - We've already used mean & stdev of this, but kurtosis is also helpful. It tells about fat tails and 'skewness' - positive value indicates more values out in 'tails' than expected (fat tails), and negative value indicates fewer values out in tails (thin/skinny tails)
- To plot this: use .hist() on daily returns.
  - The usual .mean() & .sd() also provide mean/stdev; we may also use:

    'plt.' not 'plot.'

    ```
    plt.axvline(mean, color='w', linestyle='dashed', linewidth=2)
    plt.axvline(std,  "  "  "  "  "
    plt.axvline(-std, "  "  "  "  "      etc.
    ```

    to add these metrics to the plot
  - Likewise, .kurtosis() for that.
- Plotting histograms of multiple stocks can be useful to compare; higher volatility is apparent. (And .hist(...) will do this with multiple calls; just add label= keyword)



Like...

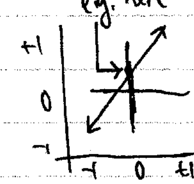
01-06 (cont.)

- Scatter plots of daily returns are also handy...
For most plots you see something like above, indicating that they sort of move with each other. If it's far off of that diagonal, particularly to the opposite corners, it can indicate stocks moving opposite to each other.

I guess this assumes that a market index is on X axis

- Trying to fit a line is common. The <u>slope</u> of that line is beta (β), and β tells "how reactive is this stock to the market?". $\beta = 2$ means that if market goes up 1%, we'd expect stock to go up 2%
- That line's intersection with vertical (i.e. vertical intercept) is alpha (α), and $\alpha > 0$ tells us that a stock tended to outperform market. ($\alpha < 0$ tells that overall it underperformed market).

e.g. here



- These measures don't have to be against the market; they may compare any stocks.
- N.B. Do <u>not</u> conflate β and correlation! β is just slope. Correlation is how tightly the points fit the line. (0 = no correlation, 1 = tight correlation). They're independent.

Assume df = daily returns

- In Python:

columns for X & Y

```
df.plot(kind='scatter', x='SPY', y='XOM')
```
for instance

- Numpy will fit lines for us:

```
beta_XOM, alpha_XOM = np.polyfit(daily_returns['SPY'], daily_returns['XOM'],
                                  1)
```
(1 for degree-1 polynomial, ie. line)

then $y = \beta x + \alpha$, so to plot: (x = SPY returns)

```
plt.plot(daily_returns['SPY'], beta_XOM * daily_returns['SPY'] + alpha_XOM,
         '-', color='r')
```

- Use df.corr(method='pearson') to find correlation one way among all stocks in 'df' (it's an NxN matrix for N stocks)
- It's dangerous to assume returns are normally-distributed and ignore kurtosis, and it's also dangerous to assume that assets are independent. In early 2000s, banks assumed bonds based on mortgages were normally-distributed (and from this, computed low risk) and ~~[illegible]~~ that the mortgages were independent, but these were wrong & ~~[illegible]~~ led to 2008 crash eventually.

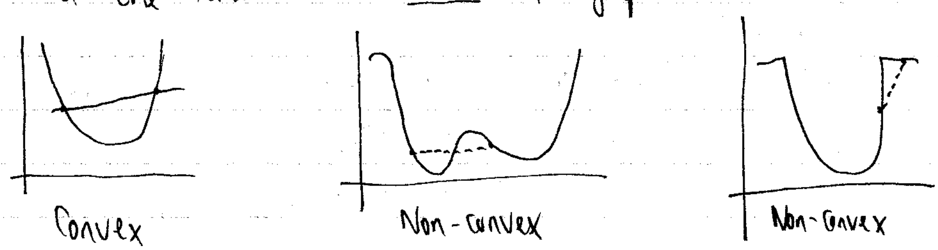
Chris Hodapp
chodapp3@gatech.edu

2016-08-25d, CS 7646 (ML for Trading)

- Notes on 'Python for Finance', Ch. 6
  - Look at 'npv' in SciPy module (net present value)
  - Also: 'pmt', 'pv'
  - CAPM, Capital Assets Pricing Model?

01-08, Optimizers: Building a Parametrized Model

- Optimizers
  - Optimizer: an algorithm for finding min/max values of functions. Such as... a parametrized model based on data, where our parameters might be the allocations in a portfolio.
  - See: minimize in scipy.optimize (he uses method='SLSQP')
  - Convex problems are easiest to solve
    - convex: "line segment between any two points in the graph of the function lies above the graph"



    - Convexity implies that a global minima exists, and it cannot have 'flat' parts. Optimizers work best with convex functions, but (with help of randomness) can work elsewhere. This can also work in more dimensions.
  - Suppose we have 2D points and want the line, $f(x) = C_0 x + C_1$, which minimizes error; how do we define this?
    - Maybe sum-of-squared-error, or of absolute error, and finding coefs C_0 & C_1 to minimize that error
  - We can fit polynomials with the same technique
    - See numpy.polyval, numpy.polyld

01-09
(How to Optimize a Portfolio)

- Portfolio optimization
  - Given: assets, and a time period
    Find: allocation of funds to assets
    Optimizing: performance (cumulative return, volatility, risk-adjusted return, i.e. Sharpe ratio, various other things)
- This is easy to do given past data, but telling us what we ~~should~~ should have done isn't really helpful, so how is this useful?
  - One answer: Periodically rebalancing a portfolio to optimize for Sharpe ratio (based on data so far) tends to do better than even allocation.
- Optimizing for cumulative return is easy but dubiously useful, especially if risk matters at all. How do we express Sharpe ratio (which is more difficult) in optimization?
  - N.B. We are maximizing Sharpe ratio. Or, minimize its negative.
  - We also need to limit the ranges of allocations, e.g. between 0 and 1 (for 0% to 100%), and must constrain so that sum of allocations = 100%.
  - See MC1 Project 2 for how to actually express this in Python.

Python for Finance, Ch. 11

- Monte Carlo simulation & options
  - If returns are normal, then prices are lognormal with (μ, σ)
  - See: Easley, Kiefer, O'Hara, Paperman (1996) - PIN, Probability of informed trading, based on # of buyer-derived & seller-derived trades. It assumes a Poisson distribution in their arrivals
  - This talks a bit about options but I may wait before trying to comprehend it.

Chris Hodapp
chodapp3@gatech.edu

2016-09-01, CS 7646 (ML for Trading)

Textbook notes

- This time, from the correct book ("Python for Finance: Analyze Big ~~Data~~ Financial Data" by Yves Hilpisch, published by O'Reilly), not the one from Packt ~~P~~ Publishing.
- Ch. 4:
  - Mostly Python review
  - p101: NumPy lets us have arrays with different types per-column, and with those columns accessed by name. They're more SQL-like. — "structured Arrays"
- Ch. 6 (Financial Time Series) - A lot of review here from lectures too
  - Main Python library we use here: pandas, from Wes McKinney (while an analyst at AQR Capital Mgmt.)
  - Its DataFrame & Series classes were modeled after R (and built on top of NumPy)
  - DataFrame: Data (various shapes & types)
    Labels (data is in columns with custom names)
    Index (index can have various formats - numbers, strings, time)
  - See p141 for some ~~n~~ nuances in joins & indices
  - If you make a DataFrame from a numpy.ndarray object, this keeps basic structure and only adds meta-information
  - We may set indices directly on 'df' with: [df.index = ...
  - See date_range function (pandas purposely has good support for dates as indices)
  - Pandas is very error-tolerant in that it will just insert NaN values & work around them, but be careful of this.
  - Its plotting is a wrapper around matplotlib.
- Chapter 5, Data Visualization
  - Like mentioned elsewhere, Pandas uses matplotlib for this.
  - This is a good summary to refer back to
- Pandas note: If row/column slicing, use .iloc, not .ix. The latter will go from the index and this may not properly use row/column numbers.

- Likewise, when adding in Pandas, note that adding goes along index not position.

Chapter 11, Statistics

- 4 Focal points of chapter
  - Normality tests — Portfolio Theory; Capital asset pricing model (CAPM); Efficient markets hypothesis; Option pricing theory
  - Modern portfolio theory
  - Principal component analysis
  - Bayesian regression
- Normality Tests
  - Much of financial theory assumes normal distribution, and so testing how 'normal' something is is important
  - Portfolio theory: IF returns are normal then only mean & variance of return (volatility), and covariance between stocks, are needed for finding optimal portfolio
  - CAPM: if returns are normal, prices can be expressed relative to some broad market index with β
  - Efficient Markets Hypothesis - An efficient market has all available information already reflected in prices. If market are efficient, stock prices are random & returns are normal.
  - Option pricing theory: Brownian motion & Black-scholes
- For geometric Brownian motion S:
  - Normal log returns: For two times $0 < s < t$, log returns ($\log S_t/S_s = \log S_t - \log S_s$) are normally-distributed.
  - Log-normal values: any time $t > 0$, values S_t are log-normally distributed.
  - See: scipy.stats.describe
    - " " .skew
    - " " .skewtest → Test if skew is close enough to 0
    - " " .kurtosis (it is for normal data)
    - " " .kurtosistest → Test likewise for kurtosis
    - " " .normaltest → Combine both skew & kurtosis

provide a p-value?

  - We can check if data is log-normal simply by taking log of it, then using the above normality tests.

Chris Hodapp
chodapp3@gatech.edu

2016-09-02, CS7646 (ML For Trading)

Python for Finance (Ch. 11), cont.

- See: pandas.io.data.DataReader - it can acquire stock data through sources like Yahoo.
- p319: S&P 500 data has "fat tails", one way of deviating from normality (see qqplot), or log-normality in this case.
  - So, geometric Brownian motion model isn't justified here, and perhaps we need richer model - jump diffusion or stochastic volatility, for instance.
- Portfolio Optimization
  - MPT (Modern or mean-variance portfolio theory) from Markowitz is a cornerstone of financial theory.
  - It considers only long positions and assumes that mean & variance completely describe log-returns. Consider weights of assets, w_i, with $\sum w_i = 1$, r_i are state-dependent future returns, μ_i = expected return. Then μ_p (return of whole portfolio):
    $\mu_p = E(\sum w_i r_i) = \sum w_i E(r_i) = \sum_i w_i \mu_i = w^T \mu$
  - Expected portfolio variance also matters:
    $\sigma_{ij} = E(r_i - \mu_i)(r_j - \mu_j)$ - that's covariance between assets i & j.
    Variance is just $\sigma_i^2 = E((r_i - \mu_i)^2)$.

.cov() in numpy

$$\text{Portfolio covariance matrix} = \Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots \\ \sigma_{21} & \sigma_2^2 & \cdots \\ \vdots & & \ddots \\ \vdots & & \end{bmatrix}$$

Σ matrix, not summation

then portfolio variance $\sigma_p^2 = E((r-\mu)^2) = \sum\sum w_i w_j \sigma_{ij} = w^T \Sigma w$

- Efficient frontier: set of portfolios either maximizing return given some risk, or minimizing risk given some return
- We can use scipy.optimize.minimize to maximize Sharpe ratio.
- They give a decent example (p328-335) but I don't dwell further on it here.

Chris Hodapp
(chodapp3@gatech.edu)

2016-09-05, CS7646 (Machine Learning for Trading)

MC3-01

- How machine learning is used in hedge funds
  - ML focuses on building a <u>model</u>

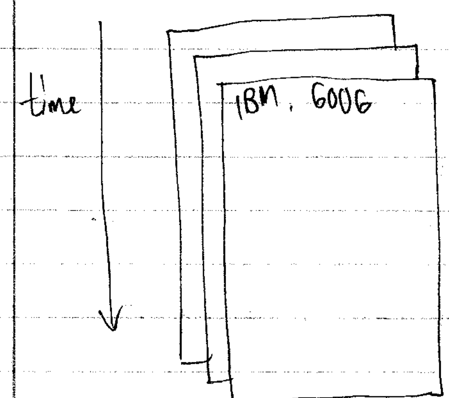
$$X \rightarrow \boxed{\text{Model}} \rightarrow Y$$

X = observations
Y = predictions

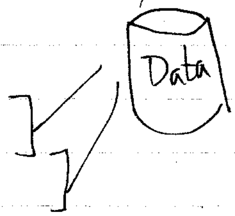
where here, we want: X = stock price history, market conditions

- Other models try to do this (e.g. Black-Scholes) but the difference with ML is that it is very data-centric in how we build a model versus deriving a formula from principles/rules.
- We're focusing first on: supervised regression learning.
  (Recall CS7641, which explains this in more detail.)
  - Methods: (those we use at least)
    - Linear regression (parametric learning)
    - kNN (instance-based)
    - Decision trees
    - Decision forests (averaging/bagging over trees)
- Suppose we have a multidimensional Pandas frame:

X Historical data ; Y = historical price, and we want to predict future price



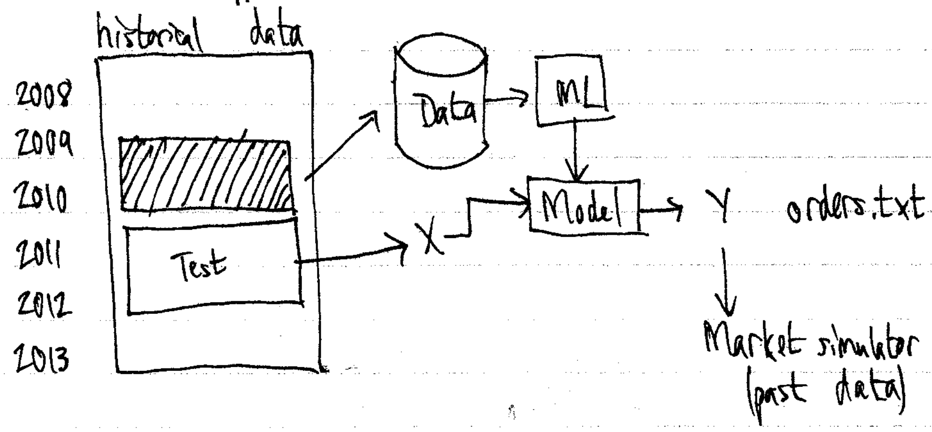
  - Perhaps we're pairing X with Y five days in the future, and build up a database of this
- Example at fintech company:
  - Select X1, X2, X3... predictive factors (Bollinger bands?)
  - Select Y



- Choose time period (how far back do we go?) & stock universe (what stocks do we consider?)
- Train, predict...
- Look up cloud application called <u>Quantdesk</u> (Balch co-founded the company)
  - It used a GA to look up the factors that seemed relevant in its "Default" model, and I guess it just uses KNN past that.
  - Back Test Score = how well past predictions matched up

→ <u>Backtesting</u>: Rolling back time window & testing system, e.g. by making mock portfolios and running numbers on 'future' data (simply historical data we didn't ~~the~~ train over)

- Problems with regression
  - Noisy & uncertain
  - Challenging to estimate confidence
  - Uncertain how long to hold a position from a forecast, or how much to allocate to it
- <u>Policy learning reinforcement learning</u> can work better; here, rather than trying to estimate price, it tries to derive the best <u>policy</u>
- How we'll approach:



Chris Hodapp
chodapp3@gatech.edu

2016-09-05b, CS7646 (Machine Learning for Trading)

MC3-02 (Regression)

- Parametric regression
  - That is, we represent a model with different parameters that we derive from data. Our linear model, $y = mx + b$, is just this with parameters (m, b), or we can use a polynomial of degree n with $(n+1)$ parameters.
- KNN is not parametric, but rather instance-based.
  - We must choose the value of k (how many neighbors do we consider?), and how we combine their values (simple mean? weighted mean?). We keep data for querying - we don't derive parameters.
  - Other things to look at: kernel regression
  - Non-parametric approaches better fit cases where behavior is more ill-defined (and for which we might not have any underlying model we're trying to fit). Or: parametric models are <u>biased</u> in a certain way, and this can be appropriate if we have information\guesses.
  - Updating parametric models is slower (as is training in general), but testing is fast. KNN is the reverse.
- "Out-of-sample testing" - testing on different data than you trained on.
  In our case, we also usually try to train on data that is earlier in time, and test on later data. Doing the opposite can introduce a sort of lookahead bias.
- See last two videos for Python skeleton\API.

MC3-03

- Assessing a learning algorithm
  - KNN (if we just use unweighted mean) tends to produce jagged results and can't really extrapolate, though it doesn't really overfit (if k is high enough to still average some)
    But, too large k can lose too much detail
  - Polynomials are similar (somewhat) but will overfit with higher degree - and they can extrapolate

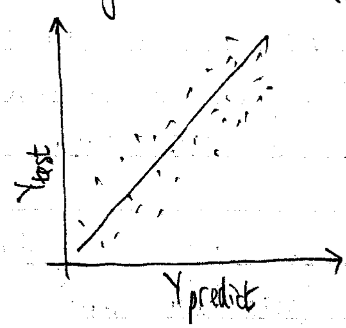
$$RMSE = \sqrt{\frac{\sum (y_{test} - y_{predict})^2}{N}}$$

- note that it emphasizes larger errors

- Metric 1: RMSE

ML3-03 (cont.)

- We may use RMS error to assess how well it matches
  - This can be computed both in-sample & out-of-sample (i.e. over both training & testing), and we expect out-of-sample error - that is, testing error - to be higher.
- Cross-validation is a good practice here (they use 5-fold as example)... except that in financial data it makes less sense because of issue of having training set put <u>after</u> test set, which in some sense is letting our model see into the future (relative to our test) and this may produce overly optimistic results.
  - → Roll-forward cross validation - always put test set ahead of training data. Note that this might mean only using some time range once (unlike in normal CV where it is reused many times.
- Metric 2: Correlation
  - Take X_{test} for timeframe, Y_{test} for same, and then $Y_{predict}$
    - running model on X_{test}. Plot $Y_{predict}$, Y_{test}:

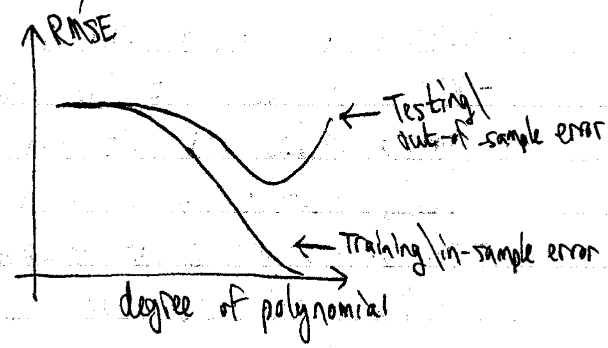


Ideally, something like this, where points fit fairly closely to line. <u>numpy.corrcoef</u> computes it.

+1 = perfect correlation
0 = no correlation
-1 = anticorrelation?

- Increasing RMSE generally decreases correlation, but opposite can happen too (e.g. with large bias)
- Overfitting may look like...

(for fitting polynomial to data)



- On kNN, training error may get higher rather than lower. (as k increases)

Chris Hodapp
chodapp3@gatech.edu

2016-09-15, CS7646 (Machine Learning for Trading)

MC3-04

- Ensemble Learners, Bagging, & Boosting
  - These aren't learning algorithms, but rather, a way of working with existing learners.
  - With ensemble learners, we may have various learners, all trained on the same data, and need a way to combine the outputs of all of them when querying.
    - With classification: Maybe they vote.
    - With regression: Maybe we take the average.
  - Ensembles often have lower error & less issue with overfitting.
    - The averaging tends to cancel the biases out from individual learners (often different ones with different biases)
  - Bagging (bootstrap aggregation)
    - Invented in late '80s; uses an ensemble of the same learner.
    - Create many subsets of the data, randomly (bags), with replacement

      n = number of (training) instances
      n' = number in a bag
      m = number of bags

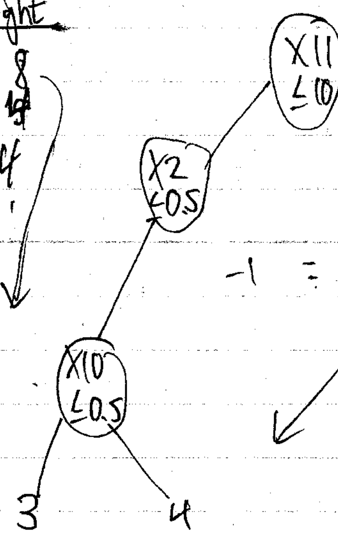
      Typically, $n' \approx 60\%$ of n
    - Train a different model on each bag (thus, m models); For prediction, just supply same input and average outputs
    - Example: Train several k-NN learners with $k=1$, and observe how it helps to smooth results
  - Boosting (e.g. AdaBoost)
    - Variant on bagging which focuses learners on parts of the input that the ensemble underperforms on
    - For each new learner added, it more highly weights the instances with higher error

Decision trees part 1 (from Youtube live stream)

- Decision trees
  - Factors: Might be binary, might be discrete, might be numbers
  - Root nodes, decision nodes, leaves

- Factors: $X_1, X_2, X_3 \ldots$
- Labels, Y: can be classes, or numbers (for a regression tree)
- He does demo with UC Irvine red wine dataset
- First thing he tries: Correlation of each factor with quality
- What if we took half the samples & built a tree (an exact tree perhaps - every single sample is represented in a leaf)?
  - Our in-sample error (training error) would be 0
  - Our out-of-sample error could be computed from predictions done on the rest of the data
- See around 54:00 in video for some Python implementation ideas?

Factor	split val	left	right
1	10.0	1	8
2	0.5	2	~~1~~
10	0.5	3	4
⋮	⋮	⋮	⋮
-1	3		
-1	4		



-1 = leaf node

- left/right values are absolute indices into array (can also be made relative)
- each row = one node, row 0 = root node, row with factor -1 = leaf node (split val = final value for that node)
- Need $\log_2 n$ questions (max) to predict
- One benefit of trees: Data requires no normalization & missing data is easily handled
  - KNN for instance requires normalization (or "closer" in one dimension depends on scale more than actual closeness)

Chris Hodapp
chodapp3@gatech.edu

2016-09-19, CS7646 (Machine Learning for Trading)

Decision trees, part 2

- For this assignment (MC3-P1), we are to use a NumPy array in the same tabular form as in these slides
- Binary trees are more efficient than higher numbers of branches (also, they're easy to build), because <CPU handwaving>
- See JR Quinlan's paper (slides have his algorithm)
- Common way to split on the "best" feature: just pick its median
- To find that "best" feature:
  - For classification, we generally try to find most similar groups, & divide & conquer
  - Common approaches: Entropy (de facto standard)
    Correlation
    Gini index
  - We may measure entropy even for regression (likewise correlation)
- Choosing the best feature is one of the more expensive parts; median is also a bit slow
- Here is where <u>random trees</u> come in
- In this algorithm, it's much the same but one chooses a random feature i & splits randomly (rather than median, just pick 2 & take their mean).
- Obviously, this constructs pretty bad trees overall. However... an ensemble of them (a random forest) can perform quite well. It can be different random seeds, or different subsets.
  In general with ensembles you want <u>uncorrelated</u> learners.

2016-09-20

- Notes on "What Hedge Funds Really Do" (Ch. 2)
  - Investors have a variety of financial instruments. Simplest is specific ones - individual stocks & bonds.
    - Problem: Small investors don't have enough capital to diversify these over a range

- Mutual Funds pool capital of many investors, & buy diverse portfolio consistent with ~~[illegible]~~ charter of fund
  - "Unit investment trusts" is their name in the UK
  - SEC heavily regulates them in US.
  - Prospectus outlines what investments
- First 'hedged fund' involved going long on one big company, and going short on its competitor, expecting they'd move opposite. The latter part hedged against the entire market going down.
- Hedge funds exploded in early '90s.
- Compared to mutual funds, hedge funds are very lightly regulated. They're also available only to "accredited" investors. They're prohibited from advertising. (at least, right now).
  - For one thing, since IP is nearly impossible to protect here, they preserve secrecy as much as they can. Holdings & strategies aren't public in most cases.
- Hedge fund strategies:
  - Equity: Focus on stock selection (often the long\short model)
  - Arbitrage: Look for prices between assets being outside of normal variation, & bet on them returning
  - Momentum\direction: Guess on probable movement, large-scale, of market
  - Event-driven: Trade based on triggers like war, supply disruption, merger
- Leverage is common, e.g. a 130/30 equity strategy: 130% of available capital is long (100% + borrowed 30%), 30% of portfolio is shorted as a hedge.
- Even finding out about hedge funds can be hard. Funds-of-funds exist, letting a manager figure out which hedge funds to invest in (and usually add fees on top of fees the hedge fund already has)

Chris Hodapp
chodapp3@gatech.edu

2016-09-21, CS 7646 (Machine Learning for Trading)

What Hedge Funds Really Do, Ch. 2

- Mutual funds typically have "expense ratio" on funds managed, median of 1% per year, <u>not</u> dependent on performance.
- Hedge funds are typically "2 and 20": (like in private equity firms - which buy a privately held company & improve it for something like an IPO later):
  - 2% expense ratio
  - 20% of fund's returns
- Renaissance Capital's Medallion Fund was 5% & 44% at its peak
  - Post-2008 crash, 1%/10% is seen often
  (Lesson: Hedge fund managers can make a lot from fees.)
- Evaluating hedge fund:
  - Return - already covered
  - CAGR, compound annual growth rate:
    - A portfolio growing at ~7.2%/year will double in 10 years. That 7.2% is CAGR: We must handle compounding; it's not simple growth.
    - <u>Rule of 72</u>: The number of periods needed for a sum to double is about 72 divided by its CAGR (as a percent). 8% growth → 9 years to double; 6% growth → 12 years.
      - If a portfolio took 15 years to double, its CAGR was around $\frac{72}{15} \approx 4.8\%$.
      - This extends easily to other powers of 2.
  - Return doesn't tell the whole story; we need to compare the return of one hedge fund manager to what other returns would be possible; e.g. an S&P 500 index may be the <u>benchmark</u> we compare with.
  - Notably, when one accounts for fees, actively-managed funds rarely outperform passive ones. Those fees are there even with negative returns.
- Between 1998 & 2010, <u>managers</u> of hedge funds absorbed 84-98% of profits.

- Hedge funds are supposed to also control risk, which we may measure with volatility, which we might calculate with standard deviation.
  - See 'drawdown' & Sortino ratio for focusing particularly on downside as a better measure of risk.
  - We already covered Sharpe ratio.
    - Note that Sharpe ratio ≥ 1.0 is very uncommon. 1.0 can be seen as "fairly compensated" for risk - each unit of risk produces a unit of reward.

Ch. 4, Market-Making Mechanics

- Market: Location, virtual or physical, where sellers & customers convene to exchange goods, including financial instruments.
  - Extreme example: New York Stock Exchange.
    - though, typically, people interact with these through brokerages.
    - Brokers trade with market makers. NYSE market makers are "specialists" and are the only ones transacting anything on the floor.
  - NASDAQ has no "floor" - it's all electronic, thus, no specialists.
- Bid price: Price at which buyer may buy
- Ask price: " " " seller will sell
  - Market spread - difference between these, and the incentive for brokers & specialist to connect buyers & sellers
- High trading volume attracts more market makers; highly liquid markets (large volume & many market makers) tend to have smallest spread.
  - Widening spread can even be a sign of liquidity slowing

Exchange only does these, typically

- Basic order types: at the market \ Market order = whatever price market has
  - limit order = order only under certain conditions, e.g. "limit order for 100 IBM at $200" = buy 100 shares of IBM only if ≤ $200\share or likewise to sell only if ≥ $200\share

A broker might do these

- Intermediate order types:
  - selling short = betting on price falling: Investor borrows shares from holder, sells them, & earns proceeds. Later, they buy the same number & return to lender.

Chris Hodapp
chodapp3@gatech.edu

2016-09-26, CS7646 (Machine Learning for Trading)

What Hedge Funds Really Do, Ch.4 (cont.)

- If shares fell in price, investor pockets difference. (If they rose, they lose it; regardless, they pay interest - "rebate" - on shares borrowed, and any dividends to which owner is entitled.)
- Contrast w/ "going long", betting that a stock's price will rise.
- Thus, two transactions: "sell to open" / "buy to close"] Reverse order of a normal "long" transaction
  - Shorting is riskier. When going long, loss is at most 100%. When shorting, loss is unbounded; also, it's betting against markets that overall rise over time.
  - Also, companies really don't like being shorted...
- Stop orders are often used to manage risk.
  - Stop loss is common - e.g. broker is to sell stock if price goes below 25% of purchase price.
  - Trailing stops are same idea, but wait for most recent high on an asset whose price has risen.
  - Free ride (less common) - broker sells half of a position when it's doubled in price; original capital is recovered and only profit is at risk.
- Buy & sell orders need to be paired up later in order books; sell orders are filled with highest buy orders first, then others
  - The spread in prices changes the equilibrium price; if more sell orders than buy orders, price tends to fall, and if the reverse, they tend to rise.
- Brokerages might clear their trades internally (say, if two clients have buy & sell orders on same stock) rather than go through exchange, and earn the market maker's spread themselves & avoid fees
- Order books can have a lot of very useful information for short-term price changes; key is speed
- Front running - broker issues trade in advance of clients, knowing the way the prices will move when order is actually executed for client

ETFs are commonly tied to an index (hence lower expense ratios)

MC2-1

- ETFs: Buy & sell like stocks, but may represent various underlying assets (basket of stocks, bonds); transparent. ETFs trade throughout the day.] Very liquid (as stocks are)
- Mutual funds: Only trade at the end of the day, and only must disclose their contents quarterly. Less transparent.] Less liquid
- Hedge Funds: Least transparent. You might not be allowed to buy/sell freely, and even investors might not know what's in them.] Buy/sell by agreement

Usually large cap

- Large cap = large capitalization as reckoned by #shares x price
  - Likewise small cap. Price of a stock is rather irrelevant on its own.
- Examples: VTINX = mutual fund
  DSUM = ETF
  FAGIX = mutual fund
  Bridgewater Pure Alpha = hedge fund
  SPLV = ETF
- Mutual funds have 5 letters. ETFs have 3 or 4. Hedge funds don't really trade - so, no symbol.
- AUM = assets under management
- Example of 2 & 20: Suppose \$100M grew to \$115M in a year.
  2% of AUM = \$100M x 0.02 = \$2M
  20% of profit = \$15M x 0.2 = \$3M
  → \$5M compensation

Supposedly, 2&20 is less common now (it was much more in late '90s, '00s)

- Structure like 2 & 20 encourages chasing profits & risk-taking
- How funds attract investors
  - Who?
    - Individuals (typically very wealthy; hedge funds typ. have ~100 investors)
    - Institutions (big retirement funds, university trusts)
    - Funds of funds (many individuals or institutions)
      - Often in many funds at once
  - Why?
    - Track record (they often want 5+ years history)
    - Simulation & story (backtest your strategy on historical data)
    - Good portfolio fit (is it the same sort of funds they're in need of?)

Chris Hodapp
chodapp@gatech.edu

2016-09-21c, CS 7646 (Machine Learning for Trading)

MC2-1 (cont.)

- Goals & Metrics of Hedge Funds
  - Beat a benchmark (e.g. S&P500)
  - Absolute return (positive returns no matter what)
  - Metrics - Cumulative return, volatility, risk/reward
    - We already covered much of this in Python

MC2-2 - Market Mechanics

- What is in an order?
  - Buy/Sell
  - Symbol
  - # shares
  - Limit or Market order
  - Price (if a limit order: don't buy above this price, don't sell below it)
- e.g. BUY, IBM, 100, LIMIT, 99.95 - Buy 100 IBM shares at $99.95/share or less
  SELL, GOOG, 150, MARKET - Sell 150 GOOG shares at market price
- Exchange keeps order book for every stock.
  The above IBM order comes in as: BID 99.95 100
  - Exchange makes this public. Exchange knows who made bid, but keeps it private
  - Maybe other people issue same order & it rises to: BID 99.95 1000
  - Suppose SELL, IBM, 1000, LIMIT, 100 comes in...
  - Then ASK, 100.00, 100 is added to order book.
  - Maybe eventually we have:

BID/ASK	Price	Size
ASK	100.10	100
ASK	100.05	500
ASK	100.00	1000
BID	99.95	100
BID	99.90	50
BID	99.85	50

  - Then BUY, IBM, 100, Market comes in. It will be satisfied here and that 1000 reduced to 900.
  - Stock would likely do down based on this order book. (There is much more pressure to sell.)

Note that exchange gives best price to client

- Consider a market order to <u>sell</u> 500 shares, and to <u>buy</u> 500 shares.

More explanation later.

  - The sell would go through (partially) immediately. The buy also would, but price wouldn't change.
- BUY, 100, Market, ~~$100~~ would go through (ASK, 100, 1000 would go to 900)
  BUY, 100, Limit, $100.02 would too for $100 (" " 900 → 800)
  SELL, 175, Market would too: BID 99.95 100 → 0 (sold these first)
  BID 99.90 50 → 0 (then these)
  BID 99.85 50 → 25 (Finally, $\frac{1}{2}$ of these)
  - See that the 'executed' prices have been decreasing
    → More sell pressure
- How orders get to the exchange
  - Typically you deal with a broker, who in turn deals with all the exchanges to satisfy the order at the best price.
    - This pressure from many brokers is what keeps order books at most exchanges pretty much the same.
  - If broker has several clients and they have opposing buy/sell orders, broker might satisfy it himself and bypass exchange (but by law, must do it for at least as good of a price as the exchange). It still must be registered at the exchange (typ. where stock resides).
  - <u>Dark pool</u> also exists - a pool that brokers may look at and satisfy beneficial trades between each other and again bypass exchange.

80-90% of "retail" trade are internal or dark pool trades. That is, only 10-20% go through exchange.

- Order book exploit
  - Consider: You're in Seattle. You issue a 'buy' order, which takes 12 msec to reach exchange. A hedge fund is 100 m away and needs only 0.3 μs to see order book. It suspects price will go up, and buys stock ahead of you. Meanwhile, price goes up. HF sells you stock at slightly higher price, and pockets difference.

Chris Hodapp
chodapp3@gatech.edu

2016-09-21d, CS7646 (Machine Learning For Trading)

MC2-2 (cont.)

- HF profits from a few extra milliseconds of seeing order book & holding stock
- Geographic arbitrage
  - Suppose HF has colocated computers at NYSE & London, and an ultra-high-speed link between them:
    Order books at NYSE show a lower price than at London - so HF very quickly buys shares at NYSE, and sells shares at London (maybe not even same shares - doesn't matter).
  - However, since lots of this already occurs, it tends to equalize the prices too quickly for this to be anything but a fleeting price difference.
- Broker handles limit orders, stop loss, stop gain, trailing stop, selling short... not exchange.
  - That's how selling short - selling stock you don't even own - is possible.
  - e.g. You want to sell IBM short.
    Joe has 100 shares of IBM.
    Lisa wants to buy IBM.
    You borrow Joe's shares, immediately sell to Lisa, get $10,000.
    You have $10K in your account but owe Joe 100 shares.
    You must buy shares to pay Joe back (and if they're cheaper than $100\share, you profit), then everyone's settled.

    Broker orchestrates all this. You need not find Joe & Lisa yourself.

    Every dollar the stock drops in price gives you $100.

2016-09-26,

MC2-3, What is a company Worth?

- What is a company worth if it produces $1\year, every year, for all the future? His answer: $10 to $25 depending on interest.
- Why we care: Because in buying stocks we want to buy when a company's "true" value exceeds market value, and sell when reversed

- "True" value estimates:
  - Intrinsic value - estimate with future dividends expected
  - Book value - assets of company
  - Market cap - based on stock price & shares outstanding
- What would you pay to receive a dollar in a year?
  - It's worth less than a promise for the government to do the same, versus someone else.
  - But both are worth less than a dollar right now.
- But... why?
  - Risk that it won't be delivered
  - Also interest rate

$$P.V. = \frac{F.V.}{(1+IR)^i}$$

FV = future value
PV = present value
IR = interest rate
i = time

- For 1% interest rate, $PV = \frac{\$1}{1+0.01} = \$0.99$
  - " 5% " " " " $\$0.95$
  - Thus, higher interest rates may be more attractive to lender by making present value worth <u>less</u>
- The analogy with companies: Dividends when owning a stock
- IR = discount rate; it is <u>higher</u> when the company is riskier
- N.B. We normally use interest rate when using present value to find future value and discount rate for using future value to find present value

Intrinsic value
- Sum of all future values = FV/DR (e.g. \$2/year dividends, 4% → \$50)

- Book value: "Total assets minus intangible assets & liabilities"
  - Intangible assets count things like patents. Ignore them, don't subtract them

You can buy the whole company for this
- Market cap: #shares × price
- Why information affects stock price
  - If investors expect future dividends to drop due to some information (can be company-specific, sector-wide, market-wide), they'll tend to sell

This is why stocks rarely go below book value
- Note that if market cap < book value, you could buy company for market cap & immediately sell assets for book value, and profit from difference (in theory at least)

Chris Hodapp
chodapp3@gatech.edu

2016-09-26(b), CS7646 (Machine Learning for Trading)

- Notes on Cramer video:
  - Cyclical companies benefit the most when economy is booming (think of construction companies for instance)
  - Secular companies thrive regardless - consider consumer staples, and stocks like P&G. They're stable, but don't tend to outperform cyclical stocks during booms.
  - 50% of stock's movement comes from sector, remainder comes from cyclical\secular.

    Does this actually work historically?
  - Rotation: Money flows from cyclical → secular growth, vice versa
  - Blah blah, buy & hold is nonsense, according to actual empirical performance of market (so Cramer claims)
  - What he advocates: diversifying, but holding onto the right kind of stocks (cyclical vs. secular)

WHFRD. Ch.5

- Introduction to Company Valuation
  - "A successful investment indicates that the buyer had a more accurate view of the asset's true value than the seller had."
  - See: "The Intelligent Investor" (Benjamin Graham) - Value investing
  - Fundamental analysis - based on fundamentals of business & finances (vs. technical analysis, which relies on past stock prices)
  - Discounting reflects time value of money

    [Stream of future cash flows] ⟷ [Lump sum]

    - Opportunity cost on money not available now
  - Even book value can overestimate if conditions prevent full value in a sell-off (e.g. bankrupcy, recession)
  - Asset-based valuation ignores future earnings.
  - Cash flow-based valuation focuses exclusively on them.

2016-10-06

MC2-04 (CAPM)

- The Capital Asset Pricing Model
  - Sharpe, Markowitz, & Miller received 1990 Nobel prize for this
  - Led to development of index funds & belief you can't beat the market.
- Portfolio:
  w_i: portions of funds in asset i. $\sum w_i = 1$, except for in leveraged portfolios, but that's another matter.
  You might also combine long & short positions, and shorting has $w_i < 0$.
  Then, $\sum_i |w_i| = 1.0$
  - Returns: $r_p(t) = \sum_i w_i r_i(t)$

Example:
Stock A: +1%
" B: -2%
w_A = 75%
w_B = -25%
$\rightarrow r_p = 1.25\%$

- "The Market Portfolio"
  - in US: S&P 500 (500 largest stocks traded on exchanges)
  - In UK: FTA
  - in Japan: TOPIX
  - These are normally cap-weighted: $w_i = \frac{\text{market cap}(i)}{\sum_j \text{market cap}(j)}$
- These indexes are like "oceans", and they may have "islands" (sectors) in them: energy, technology, manufacturing, finance
- Apple & Exxon are around 5% each of S&P because of their size

Memorize this

- The CAPM equation: $\underbrace{r_i(t)}_{\text{movement of stock } i \text{ on a given day}} = \underbrace{\beta_i \overbrace{r_m(t)}^{\text{market index}}}_{\text{due to market}} + \overbrace{\alpha_i(t)}^{\text{residual}}$

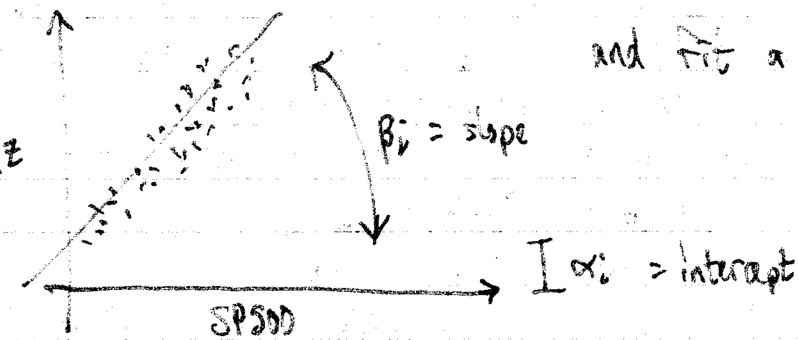
If $\beta_i = 1$ the stock tracks market exactly
α_i is a random variable with expected value of 0 according to CAPM.
β and α come from daily returns

Chris Hodapp
chodapp3@gatech.edu

2016-10-05(?), CS7646 (Machine Learning for Trading)

MC2-04 (cont.)

- Consider... scatter plot XYZ of daily returns



and fit a line...

- Note that these are historical - they're always over past results.
- CAPM vs. active management
  - Passive: Buy index & hold
  - Active: Pick stocks (some will overweight, some will underweight, relative to index market)
- Both agree with the CAPM equation. Where they disagree is that passive investors believe α_i is a random variable with expected value 0 & they can't predict it, while active investors believe that on average they can predict it better than a coin toss.
- CAPM for portfolio: $r_p(t) = \sum_i w_i (\beta_i r_m(t) + \alpha_i(t))$

  but then $\beta_p = \sum_i w_i \beta_i$ and $r_p(t) = \beta_p r_m(t) + \alpha_p(t)$

  that is, β_p = entire portfolio's β

  Active manager will break $\alpha_p(t)$ into $\sum_i w_i \alpha_i(t)$, rather than just assume it's random with expectation 0.
- You want higher β in "upward" markets, lower β in "downward"
  - If CAPM says that we can't predict α, then the only thing we can choose is β... but, it also says that due to efficient markets hypothesis you can't really tell if it's an upward market or downward one, thus...
  you can't beat the market.
- Having said that, the professor believes that this is incorrect!

MC2-04 - Arbitrage Pricing Theory

- Stephen Ross, 1976. He looked at $r_i = \beta_i r_m + \alpha_i$ and asked: why just one β_i? Why not compare a stock's exposure to more specific sectors of the market? You'd get something more like...

$$r_i = \underbrace{\beta_{iF} r_F}_{\text{Finance}} + \underbrace{\beta_{iT} r_T}_{\text{Tech}} + \underbrace{\beta_{im} r_m}_{\text{Mfg}} + \cdots + \alpha_i$$

- We don't use that here but he brings it to our attention anyhow.

MC2-05 - How Hedge Funds use the CAPM

- The informational edge they seek is usually market-relative: They want to find stocks that rise more when market does, but fall less when market does.
- Suppose a fund predicts that:
  Stock A will be +1% over market, and $\beta_A = 1.0$
  Stock B will be -1% below market, and $\beta_B = 2.0$
  They'd want to take long position on A, short on B.
- Scenario 1: Market stays flat, and we have \$50 in A, -50% in B. $r_A = \beta_A r_m + \alpha_A$, but $r_m = 0$ and assume our prediction is right - thus, $r_A = 1\%$. Likewise, $r_B = -1\%$ but we had negative bet, so our total gain for A & B was: $(1\%)\$50 + (-1\%)(-\$50) = \$1$, or 1%.
- Scenario 2: Market goes up 10%. Then $r_A = 11\%$, $r_B = -19\%$, so total of -4% or -\$4 (+\$5.50 from A, -\$9.50 from B).
- Scenario 3: Market goes down 10%. Then $r_A = -9\%$, $r_B = 21\%$; A gives -\$4.50 but B gives +\$10.50 for a total of +6% and \$6.
- CAPM can help us optimize this. Note that even with perfect knowledge of β and α, we still lost under certain markets!

Chris Hodapp
chodapp3@gatech.edu

2016-10-03c, CS7646 (Machine Learning for Trading)

MC2-05 (cont.)

- $r_p = \sum_i w_i (\beta_i r_M + \alpha_i) = (w_A \beta_A + w_B \beta_B) r_M + w_A \alpha_A + w_B \alpha_B$ for 2 stocks.
  - For our previous allocations, this equals: $-0.5 r_M + 1\%$.
- How can we remove r_M from this? (Assuming we can't predict it) or, at least minimize it?
  Note that if $\beta_p = 0$ then r_M is irrelevant.
  So... $\beta_p = w_A \cdot 1 + w_B \cdot 2$ and leave $w_A \geq 0$, $w_B \leq 0$ (long A, short B).
  $\beta_p = 0 \rightarrow w_A = -2w_B$
  And $|w_A| + |w_B| = 1$, thus (knowing this): $w_A - w_B = 1$
  $-2w_B - w_B = 1$ → $w_B = -1/3$, $w_A = -2(\frac{1}{3}) = \frac{2}{3}$
- Run the actual numbers for $r_m = 10\%$: $r_p = 1\%$
  and it's the same for all other r_m!
- Huge caveats: β values aren't guaranteed to carry into future! Neither are α values!
- This is just an illustrative example of long/short investing to "reduce" exposure to the market.
- Summary:
  - If we know β and α, we can use CAPM to reduce market exposure in a portfolio.

MC2-06 Technical Analysis

- Technical Analysis
  - Technical analysis & Fundamental analysis are two broad ways of looking at stocks to buy/sell.
  - Fundamental analysis looks at aspects of the company; investors look for things like the price of a company being below its value.
  - "Technicians" don't care about the company itself. They look for patterns & trends in a stock's price.
  - Tech. analysis looks only at price & volume, while fundamental analysis looks at book value, assets, cash flow, earnings, dividends...

Many folks criticize technical analysis for its blindness to actual company value. Think of it as a trading approach rather than an investing approach.

- It computes statistics called indicators, which are heuristics to hint at buy/sell.
- There is still information in price, and heuristics can work.
- When is it effective? (according to professor)
  - Individual indicators are weak (he believes they were effective in '80s but then lost effectiveness through people trading on them).
  - Combinations of indicators are stronger - e.g. 3-5 in ML context
  - Contrasts (between other stocks, between market)
  - Shorter time periods (milliseconds to days)
- Fundamental factors are useful longer-term (days to decades)
- His view: Computers handle the faster decisions of tech. analysis best; humans handle the slower, more complex/intuitive decisions of fundamental analysis.
- Hundreds of indicators exist. We'll focus on 3:
  - Momentum
  - SMA (simple moving average)
  - Bollinger® Bands
- Momentum: What is direction & strength of price change?
  In Python: price[t]/price[t-n]-1, n = 5, 10 day or so on
  -0.5 = 50% drop, +0.5 = 50% increase
- SMA: Just compute average over n-day window. It tends to lag the price. One common thing is to look for strong momentum and SMA crossing the price.
  - It's also used as a proxy for underlying value, and people hope to find arbitrage opportunities.
  - It will often be visualized as how far real price is above or below SMA

movement inside → outside is never a signal.

- Bollinger Bands: Put a band, $\pm 2\sigma$, around SMA. Movement from outside of this band is a signal to buy/sell; inside of it is just supposed to be normal volatility.
  - Or, look for movement outside of band, back inside. If from above, sell; if from below, buy.

Chris Hodapp
chodapp3@gatech.edu

2016-10-3rd, CS7646 (Machine Learning for Trading)

MC2-D6 cont.,

- In Python: $BB[t] = \frac{price[t] - SMA[t]}{2 \cdot std[t]}$ & look for $BB = \pm 1$ - that is the boundary of those bands.
- We may need to normalize these values when using them for machine learning. We usually just do: $\frac{v - mean(v)}{std(v)}$ → mean 0 variance 1
- Usual caveat: Don't start trading yet. The professor has given just his own approach, and a lot is still to come.

WHFRD, Ch. 7

- A market's efficiency is the speed at which information on events reaches buyers & sellers. (or: it reflects this speed)
- Sellers might unload a stock upon negative news (in hopes that buyers don't know it yet, and still value the stock higher)
- Buyers might buy upon positive news (in hopes that sellers don't know it yet, and haven't upgraded price yet)
- See: "Fooling Some of the People All of the Time" David Einhorn)

Actually Ch. 6

Ch. 7

- Text says Miller, Modigliani, not Markowitz, & Sharpe for CAPM, not 1990.
- Hedge funds exist on the assumption that CAPM & EMH are wrong. However, CAPM still provides a useful framework.
- ABCs of CAPM: Alphas, Betas, Correlations
  - Correlation is very important but discussed later
  - Particularly, uncorrelated & anticorrelated assets can be vital to reducing risk.
- If we have a scatter-plot of two assets' daily returns, and fit a line:
  α = return over the other (esp. over market), β = volatility relative to market, correlation coef = 'tightness' of scatter around line

To read: Chs. 8, 9

- "buying beta" = returns from a generally upward market
- "seeking alpha" = returns from investment skill.
  (N.B. mutual fund industry finds little evidence of persistent alpha in fund managers)
- Note that $\beta = 1$ is easy to purchase: just buy index funds.
- "buying beta" means investing in stocks riskier than the index, or more volatile at least.
- "beta-balanced" is like the approach in last lecture: getting $\beta_p = 0$ so that risk from market is small.
  However, this produces excess returns that are small, so hedge funds often make up for this in volume, especially with leverage (20-30X their invested capital, even).
  Leverage cuts both ways - hence long/short strategies.

(Sorry, gave up on homemade walnut ink.)

2016-10-10

MC2-07

- A successful buy/sell is denoted with a "tick" in a timeline. Exchanges maintain it and can let people subscribe; various exchanges aren't guaranteed to have the same price
  - A stock might have hundreds of thousands of these
- Normally we see aggregated data, e.g. once per hour, once per minute:
  - open: first transaction price
  - high: highest price
  - low: lowest price
  - close: final transaction price
  - vol: total number of shares

In this course we deal with daily data though. But, it generalizes to smaller periods.

- What happens when we see huge drops in a day (75%, 50%)?
  - Probably stock splits. Total value is preserved, but unit price changed.
  - Why do they split? Sometimes, because price is too high. Options usually are in units of 100, and this makes them more feasible to deal with.
    It helps with liquidity too, and with getting specific amounts in a portfolio.

Chris Hodapp
chodapp3@gatech.edu

2016-10-10(b), CS7646 (Machine Learning For Trading)

MC2-07 (cont.)

- Use adjusted close to avoid having to account for splits.
  - Adjust prices before the split, starting at newest splits & working back
- Dividends also affect the value of a stock
  - Particularly, stock price rises before dividend, and drops after dividends paid
  - How might we adjust historical prices for this? The exact same way as a split. Adjusted close takes this into account.
- On the final day in a series, adjusted & actual are always equal.
- Survivor bias:
  - If you look at stocks existing today (say, in S&P 500) to form a portfolio, you are only considering stocks that survived until today, and ignoring stocks that went away.

MC2-08

- Efficient Markets Hypothesis
  - With both technical & fundamental analysis we assume that information exists which we can exploit.
  - Efficient Market Hypothesis says this is false, under the assumption that a market has a large number of investors, new info. arrives randomly, prices adjust quickly, & reflect all available information.
- Information source examples (most public to least):
  - Price/volume
  - Fundamental (made public, quarterly)
  - Exogenous (information about world that affects company, e.g. cost of oil)
  - Company insiders (information available only to company insiders)
- 3 Forms of EMH:
  - Weak: Future prices can't be predicted by analyzing historical prices — Ignores fundamentals, insider info
  - Semi-strong: Prices adjust rapidly to new public information
  - Strong: Prices reflect all information, public & private (not even insider information helps)

- If EMH is correct, then this course is nonsense. However, hedge funds sort of contradict it. The professor is very skeptical of the strong form (because people <u>do</u> profit from insider information)
  - He also showed data suggesting that semi-strong EMH is also false.

all review...

WHFRD
~~MDR~~ Ch. 8
MC2-09

- The Fundamental Law of Active Portfolio Management
  - Grinold's Fundamental Law
    - Performance
    - Skill
    - Breadth

    $\Big\}$ performance $=$ skill $\cdot \sqrt{\text{breadth}}$

    breadth = number of opportunities available? (how many stocks, for instance)
  - Performance relates to IR, information ratio (loosely, excess returns over market)
  - Skill relates to IC, information coefficient
- Thought experiment: "coin flipping casino"
  - Flip coins rather than stocks. Coin is biased (like α), 0.51 probability for heads; uncertainty is like β
  - Bet N coins; if win, now have $2 \cdot N$, if loss, ~~[illegible]~~ then 0
  - Casino: 1000 tables, you have 1000 tokens, game runs in parallel, and you may distribute tokens however you want
- Betting 1 token each on 1000 tables is better than all 1000 tokens on one table. The former has a very high risk, but no higher return. (In fact, their expected rewards are both \$20.)
- One way to consider risk: What is chance of losing everything?
  - First bet: That probability $= 0.49$.
  - Second bet: $= (0.49)^{1000} \approx 10^{-310}$
- Another way: Standard deviations (31.62 vs 1.0, respectively)
- We can do risk-adjusted reward, like with Sharpe ratio
  exp. reward / stdev = 0.63 & 20, respectively.

Chris Hodapp
chodapp3@gatech.edu

2016-10-10c, CS7646 (Machine Learning for Trading)

MC2-08 (cont.)

- Interestingly, $20 = 0.63\sqrt{1000}$ – and this holds for other examples (note similarity with $\text{performance} = \text{skill} \cdot \sqrt{\text{breadth}}$)
- Real world, consider two funds: (with similar performance)
  - RenTec trades 100K/day
  - Warren Buffett holds (emphasis on <u>holds</u>) 120 stocks
- IR, IC, breadth:
  - IR, Information Ratio: $r_p(t) = \overbrace{\beta_p r_m(t)}^{\text{Market}} + \overbrace{\alpha_p(t)}^{\text{skill}}$
    - Sharpe ratio of 'skill' component is IR: $IR = \frac{\text{mean}(\alpha_p(t))}{\text{stdev}(\alpha_p(t))}$ – historically computed
      - It's basically Sharpe ratio on excess returns
  - IC, Information Coefficient, = correlation of forecasts to returns

    +1.0 for perfect predictor. -1.0 for perfectly wrong.
  - BR = number of trading opportunities per year
- Fundamental law: $\underbrace{IR}_{\text{perf}} = \underbrace{IC}_{\text{skill}} \cdot \underbrace{\sqrt{BR}}_{\text{breadth}}$

  Grinold & Kahn
  - If you want to improve ~~breadth~~ performance, improving <u>breadth</u> (number of trading opportunities) – look to additional stocks, additional markets

WHFRD, Ch. 9

- "Wide diversification is only required when investors do not understand what they are doing." – Warren Buffett
  - What are BRK-A\BRK-B (Berkshire Hathaway)?
  - Buffett is known for investing strongly in a few companies
- See: Grinold & Kahn, <u>Active Portfolio Management</u>
- One way to consider: Buffett's skill & attention can't scale while RenTec's computer models can (enough to benefit at $\sqrt{\text{breadth}}$)
- Two approaches to increasing breadth:
  - Hold more equities at once
  - Turn over assets more frequently

  But: The more breadth in portfolio, the less skill one can apply to each part of it
- MPT distinguishes
  - systematic risk (exposure to entire asset class)
  - specific risk (risk of specific asset)

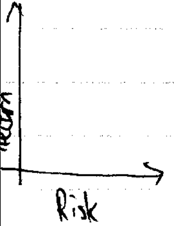
$$\text{Leverage} = \frac{\sum |\text{investments}|}{\sum \text{investments} + \text{cash}}$$

investments = value of portfolio if liquidated.

Broker typically limits leverage ≤ 2.0.

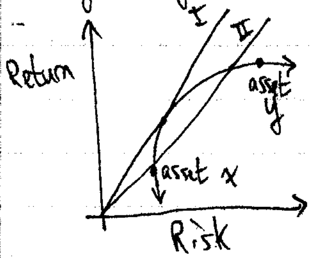
WHFRD, Ch.10

- Modern Portfolio Theory: Efficient Frontier & Portfolio Optimization
  - Harry Markowitz, 1950's
  - Assets can be classified on risk & return axis; this has an upward bias (low-risk, high-return assets would just be priced higher, thus diminishing their returns).
    - Efficient frontier connects assets lying generally above and to the left; the right assets can reduce risk without costing return



- Low correlation is prized; it can reduce a portfolio's volatility to less than the components, particularly for negatively correlated assets

Here, up and to the left = highest Sharpe ratio



Note that the slopes of lines I & II are the respective Sharpe ratios of portfolios with certain weights.

- Negatively-correlated assets with similar returns can be combined for the same return, yet lower risk than individual assets. (Negative correlation is rare, so we often settle for just low correlation)
- See: QSTK, http://wiki.quantsoftware.org/index.php?title=QSTK_Tutorial_8
- For each level of target return, some weights provide the lowest risk. The curve this forms (given some collection of assets) is the efficient frontier.
- Returns and correlations are not stable values.
  - High returns → asset is popular to buy → price goes up → returns go down
  - Events can cause uncorrelated assets to suddenly be highly correlated.
- Thus: Optimize regularly.
- Caveat: Standard deviation (on which Sharpe is based) isn't the best measure of risk. See Sortino ratio or downside capture.
  - Other caveat: Beware tail risk.

Chris Hodapp
chodapp3@gatech.edu

2016-10-19, CS7646 (Machine Learning for Trading)

WHFRD, Ch. 11

- Event Studies
  - EMH rejects idea that event knowledge can give an advantage.
  - However, MacKinlay's 1997 study shows expected effect on stock at time of news - but also an effect before news
  - Also, markets tend to overshoot & revert to the mean, which can be an arbitrage opportunity.

2016-10-25

WHFRD, Ch. 12

- Actual vs. Adjusted stock prices:
  - Splits & dividends make them differ
- Some boards avoid splits (they believe higher stock prices attract long-term investors and deter casual traders).
  - BRKA is this way - > $100K\share (but BRKB is 1/1500 this)
- Reverse splits are also a thing

2016-10-27

0-25 video lecture pt 1

- Covers material for MC3-P2
  - 1st component: Rule-based strategy from technical indicator
  - 2nd: Instead of rules to trigger trades, use ML (in this case, use decision trees for classification, not regression)
    - Classifying buy\sell\whatever
- This is first example where we consider time.
- N.B. If we use market close data, we can't really trade on it till the next day.
- See: roll-forward cross-validation, not multi-fold.
  Important: only test on data from future (relative to training).
- Backtesting still can lie.
  1. In-sample backtesting (backtesting over training data).
  2. Survivor bias (selective use of data biased by what's 'alive' at end)

- Use historic index membership
- Pair with SBF-free data
- Use these indices as universe for testing
  - eg. you must account for assets of companies that are no longer around (perhaps they died in 2008)

3. ?

Also, if someone shows you a Sharpe ratio of 7, it's bullshit.

4. Ignoring market impact
   - The act of trading affects the market. Even if you can accurately predict the market, you might not be able to trade on this information without invalidating it (or eg. market moving against your trade)
   - This is particularly true when dealing with smaller stocks & larger volumes of trade. You might also find that inefficiencies exist in smaller assets here because HFs can't do much (too small for them) so they ignore it.

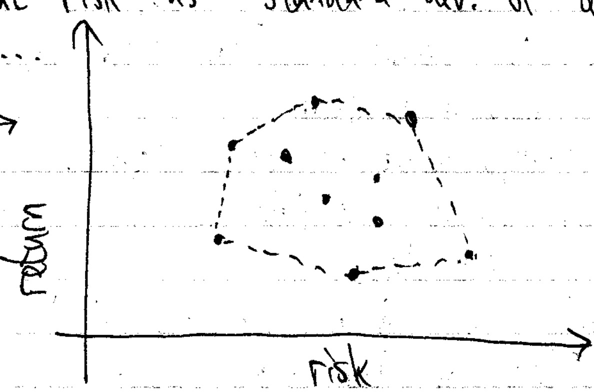
- Switch over to David Byrd... "Vectorize Me!"
  - "Basket" indicator: Look for divergence between stock & index

2016-11-07

MC2-P10

- Portfolio Optimization & Efficient Frontier
  - If you have some "good" investments, how do you decide how much of each to put in your portfolio?
  - One approach: <u>mean-variance portfolio optimization</u>
    - Given: target equities & desired return, what is allocation that hits expected return but minimizes risk?
- We treat risk as standard dev. of daily returns.
- Consider...

high return is clearly here



low risk is clearly here

Each dot is an asset. We may produce any point inside that dotted perimeter by how we weight their allocations. (Actually, outside too!)

Chris Hodapp
chodapp3@gatech.edu

2016-11-07(b), CS 7646 (Machine Learning For Trading)

MC2-P10 (cont.)

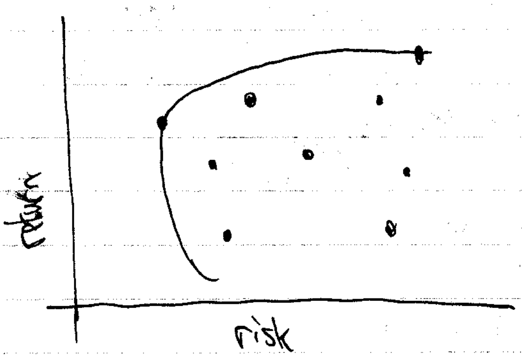
- However... We can actually produce a risk that is <u>lower</u> than any of the components, according to Markowitz.
  - Markowitz showed that - contrary to the wisdom of the time - stocks & bonds in a mixture actually were <u>lower</u> risk than bonds alone.
- <u>Covariance</u> matters. Particularly: We want to blend assets with negatively-correlated (or at least low-correlation) assets.
- Generally we want short-term uncorrelation/anticorrelation, but long-term correlation (they should still grow with each other).
- MVO - mean-variance optimization

  Expected return for each asset
  Volatility " " "
  Covariance matrix between assets
  Target return
  (can be anywhere from min. of expected return, to max.)

  → Asset weights for portfolio that minimize risk

- Efficient frontier
  - For any target return is an "optimal" (lowest-risk) portfolio.
  - Example:

Also: Tangent lines of efficient frontier represent max Sharpe ratio!



Note on this line that it's possible to <u>decrease</u> return, yet <u>increase</u> risk! We want the top-left parts.

Note, however, that efficient frontier is more a <u>theoretical</u> tool than anything.

That line above is the <u>efficient frontier</u>. Anything 'inside' it is ~~[illegible]~~ inefficient (it's possible to either decrease risk with same reward, or increase reward without risk). 'outside' of it is impossible.

MC3-05 (Reinforcement Learning)

- Supervised learning might predict price, but don't really tell us when to trade or how long to hold a position (or whether to). Also, it may not cope well with uncertainty.
- Reinforcement learning on the other hand can learn a specific policy to follow.
- Sense, think, act cycle...

→ MDP

  s = state, π = policy, a = action. $\pi(s)$ → new state...
  $T(s,a)$ transition function, r = reward
- But how do we find π?
- Our 'environment' is the market. Our actions are trades (or not). States are information on the market.
- Recall discounted rewards? This is a natural fit for how we treated 'money in the future' prior, and interest rates.

MC3-06 (Q-Learning)

- Lot of review... We can use Q-learning when lacking an explicit T & R. $Q[s,a]$ = immediate reward + discounted reward (we can see it as a Q table, not a function, if we like)... use $\langle s, a, s', r \rangle$ experience tuple
  - Learning rate α, $\alpha \approx 0.2$ perhaps
- How do we handle rewards?
  - Daily returns? (Converges faster.)
  - Or, 0 until we exit the position.
- How do we handle state?
  - One good one: $\frac{\text{adjusted close}}{\text{SMA}}$
  - Bollinger band value
  - P/E
  - Are we holding a stock?
  - What is return since we bought stock?

We typically must discretize each ~~state~~ value, and combine

Why not adj. close: Because stocks vary widely in this value's range, which complicates learning

- Typically we must discretize inside of some limited range
-

Chris Hodapp
chodapp3@gatech.edu

2016-11-07c, CS7646 (Machine Learning for Trading)

MC3-07 (Dyna)

- One problem with Q-Learning: It's not very efficient with regard to <u>experience</u>, i.e. it requires a lot of extra interactions with the real world.
  - Hence, Sutton created <u>Dyna</u>, which estimates a model (Q-learning is model-free)
- Dyna-Q blends model-based & model-free

Q-learning:
- init Q table
- observe s
- execute a, observe s', r
- update Q with ⟨s, a, s', r⟩

In Dyna-Q we add:
- Learn model T, R
- "Hallucinate" experience
- Update Q

(much cheaper)

& maybe iterate this 100-200 times, then return to Q learning

- To "hallucinate" experience: Pick random s & a, and update Q based on estimated T & R.
- Learning T:
  - T[s, a, s'] is probability that s & a lead to state s'
  - Make a table T_c (for T count)
    - Init all T_c = 0.00001 or other small value
    - While executing, observe s, a, s' & increment $T_c[s, a, s']$
  - Then: estimated $T[s,a,s'] = \frac{T_c[s,a,s']}{\sum_i T_c[s,a,i]}$

Think about its meaning & probability!

- Learning R: $R'[s,a] = (1-\alpha) R[s,a] + \alpha r$ for immediate reward r
- Recap:

Q-Learn (Q):
- Init Q table
- observe s
- execute a, observe s', r
- update Q with ⟨s, a, s', r⟩

Dyna-Q:
- T'[s, a, s'] update
- R'[s, a] update

(Update model)

- Infer s' from T[], random s & a
- r = R[s, a]

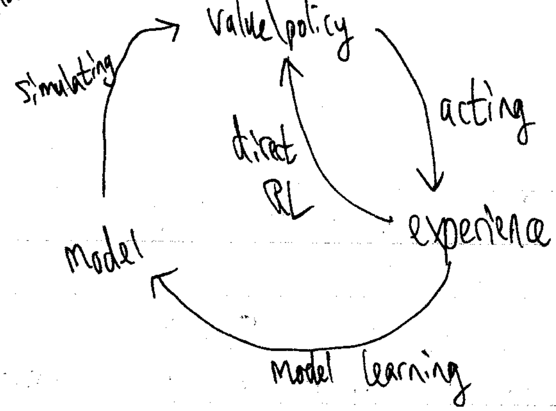
("Hallucinate")

- Update Q with ⟨s, a, s', r⟩

(Q update)

(Iterate a few hundred times)

Dyna:



2016-11-21

Event Studies & Deep Dive into a Forex Strategy

- Idea: How much do positive & negative news affect prices?
  - See: "Event Studies in Economics & Finance" (MacKinlay, 1997)
  - There can be opportunities in the market overreacting to good & bad news (e.g. selling stocks overpriced temporarily from good news).
    - Some funds monitor intraday news too, or even use voice recognition to trade stocks that news talks about
    - Some services look at Twitter too but evidence shows (TM) that it has useful info - but you must act quickly on it
  - Other 'events': Earnings report; spike in activity
    - e.g. Lucena handles events like this
  - We might: Collect all events, put them at an axis where $t=0$ is the time where the event happened, & plot mean of returns + stdev relative to this for various historical stocks
- How do we apply this to Forex?
  - Forex = foreign exchange = buying currency of other countries
  - It is a very liquid market, traded nearly 24/7, and nearly 5 times the daily volume of stocks
  - Allows very high leverage, 250:1
  - Also, because of the size of currencies, even very large HFs don't make much impact overall
  - These assets are all literally just cash, but it may earn some interest; exchange rates change very abruptly as well

Chris Hodapp
chodapp3@gatech.edu

2016-11-21(b), CS7641 (Machine Learning for Trading)

Event Studies etc. ~~[illegible]~~ (cont.)

- Forex works on these things - interest & exchange rates
- Every trade is a <u>currency pair</u>, and always is long one trade & short another. For instance, trading USD & JPY (USDJPY) means being long on USD, short on JPY
- G10 = top 10 currencies (thus 90 possible pairs to trade)
- Common strategy: Go long on currency with high interest rate, short on currency with low interest rate
- Note that stocks have a long-term upward trend, but this isn't the case for currencies. (You can't really just buy-and-hold cash.)
  - Profiting revolves around guessing what's on its way up or its way down. It's basically zero-sum.
- Kadane Method
- Quantdesk can do a lot of this analysis
- "Nice" returns in Forex might mean 1-2% per year, but very solid & low-volatility

However, leverage multiplies that.

- ETN = exchange-traded <u>notes</u> (like ETFs)
- Why Chinese currency's not used: It's manipulated like crazy (so is Venezuela's).
- 'carry trade'

MC3-P3 notes

- Indicator: Sortino ratio? or something based on it
→ Look at time-series notes again
  - ~~Ulcer Index?~~ ~~(Ulcer performance index, Martin ratio)~~
- Contrarian versions of moving-average crossover, channel breakout, Bollinger trading rule?
- Random Walk Index
→ Get plots

Different from:
SMA
Bollinger Bands
RSI, Rel. Strength Index

- VWAP: Volume Weighted Moving Average
  - but supposedly this is good only for intraday

Candlestick Pattern

ADX
-Average Directional Index

RSI Divergence?

MACD

Trix

Part 2

- Long or short. 10 day hold, that is you must remain in a position (once you take it) at least 10 days.
- Add EMA?
  Add 'signal line'?

-Can short, but only -500.
Positions are:
-500
+500
0

Am I ever interested in indicators on a portfolio rather than a single asset?
For instance, would I want to use UI on a short position? It's rather reversed there since a short position benefits from downward volatility

---

Outperform IBM in in-sample period - that's part 2 goal.
- Generate orders file, run it through market sim
- Make chart

- Signal ideas:
  - TRIX zero-crossing +/-, sell? (or short) -/+, buy? (or sell short position)

TRIX-14 does a little better but still seems to be caught in little downturns.
It seems to profit nicely during the 2008 crash, and not do as well elsewhere.

- Signal line might help...
- It keeps shorting when the market is back on its way up.
- However, a few "correct" shorts are responsible for almost all of the gain over benchmark. (But what else could they be? Benchmark is like buy orders only.)