

Machine Technical Analysis

TIME SERIES LEARNING WITH COMPUTER VISION

ALEX LEE

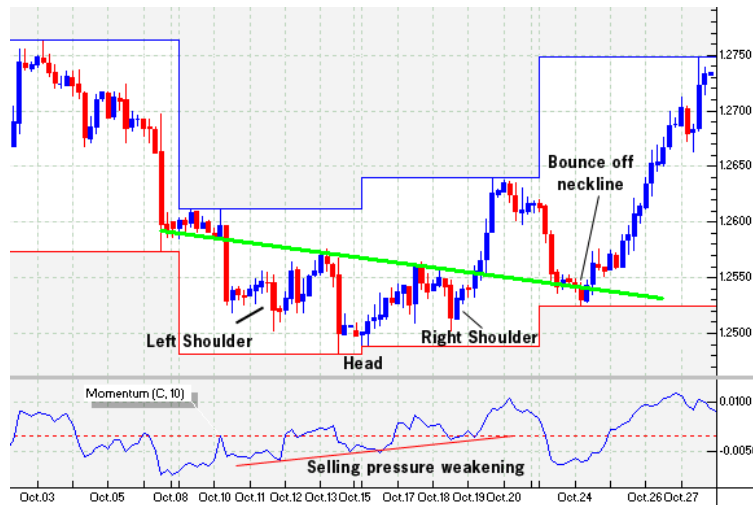


The Question

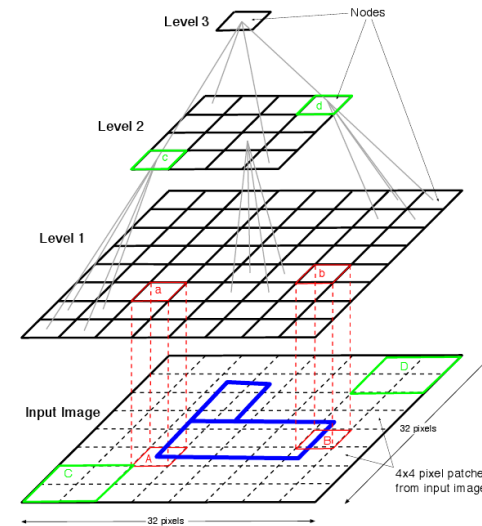
Can visual features be used for predictive modeling of security prices?

“Technical analysis” attempts to do so, with humans as the “algorithm”... but is basically bunk

Machine learning may be able to improve upon fallible human traits and perceptual biases



= baloney



= better (?)

The Data

Time series data were obtained from a Google Finance API that allows pulls of price data for a given ticker at the one-minute tick level (up to 20 days historical)

Data are provided for open, high, low, and close (OHLC) prices for each interval

Data are clean, but inconveniently timestamped (in Unix epoch format)...

...fortunately, Python and pandas allow for relatively easy cleanup and indexing of time data

Time indexing is important, to keep our models honest by barring them from peering into the future

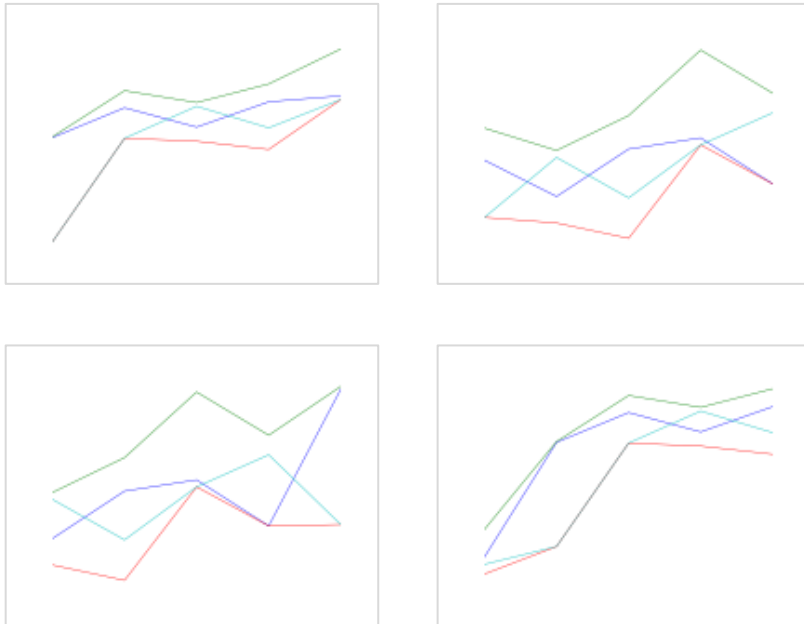
```
EXCHANGE%3DINDEXCBOE
MARKET_OPEN_MINUTE=510
MARKET_CLOSE_MINUTE=916
INTERVAL=60
COLUMNS=DATE,CLOSE,HIGH,LOW,OPEN,VOLUME
DATA=
TIMEZONE_OFFSET=-360
a1424701860,2107.67,2110.05,2107.61,2109.83,0
1,2105.86,2107.64,2105.84,2107.64,0
2,2104.32,2105.93,2104.28,2105.81,0
3,2105.11,2105.11,2104.32,2104.32,0
4,2105.08,2105.34,2105.01,2105.08,0
5,2105.79,2105.79,2104.77,2105,0
6,2106.11,2106.32,2105.8,2105.8,0
7,2105.9,2106.6,2105.9,2106.1,0
8,2105.81,2105.82,2105.53,2105.81,0
9,2105.73,2106.04,2105.73,2105.79,0
10,2105.89,2106.04,2105.49,2105.71,0
11,2105.95,2105.96,2105.72,2105.81,0
12,2105.79,2106.35,2105.79,2105.96,0
13,2105.13,2105.8,2105.13,2105.68,0
14,2105.02,2105.14,2104.7,2105.12,0
```

The Modeling Approach

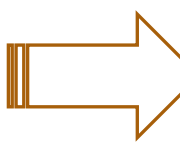
1. Obtain appropriate time series data (minute level)
2. Slice into windows
 - Window length is somewhat arbitrary, but impacts predictions, so can be tuned depending on specific use case (macro vs. HFT, e.g.)
3. Graph data for each window and save as images
4. Use computer vision (CV) to extract features:
 - Directly from images using linear feature extraction
 - Indirectly, by converting the graph to pixel intensity data, then unrolling the pixel data matrix into a vector 1000s of new features for each observation, “generated” from the underlying price data
5. Train models on extracted features

Modeling Approach, continued

FROM THIS:



TO THIS:



The figure shows four 8x8 binary matrices arranged in a 2x2 grid, representing the 'TO THIS' part of the modeling approach. Each matrix contains binary values (0 or 1) in a structured pattern.

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	1	1	0	0	0	1
1	1	1	1	0	1	1	1

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	1	1	0	0	0	1
1	1	1	1	0	1	1	1

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	1	1	0	0	0	1
1	1	1	1	0	1	1	1

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	1	1	0	0	0	1
1	1	1	1	0	1	1	1

Challenges

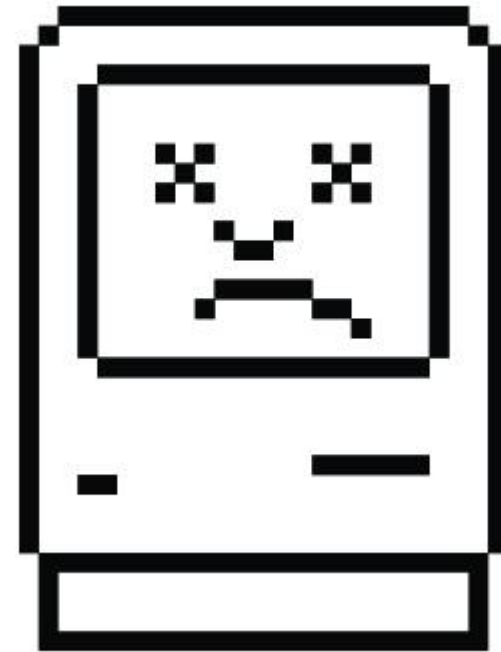
Wrangling the graphical data while not running out of RAM

Having patience while trying to train models on massive datasets ($\sim 6,000 \times 30,000$ matrix of raw pixel data)

Not being able to use regular TTS / Cross-Validation due to time series data

Not knowing much of anything about CV prior to embarking on this project

Time constraints on learning and applying new modeling techniques



Results

Modeling was carried out on basic OHLC price data + 5 created features as a baseline

Models varied in performance over these data; high end performance was impressive

CV-based extracted features also performed quite well (better than comparable base models for target of 5 minutes)

Raw pixel data did not perform well with standard modeling; ANN may be better

Model	Target	Base	Score	Delta
AdaBoost	1 min	.49	.60	.11
Random Forest	1 min	.49	.72	.23
Random Forest	5 min	.50	.64	.14
Support Vector Machine	1 min	.49	.81	.32
Support Vector Machine	5 min	.50	.71	.21
Logistic Regression	1 min	.49	.84	.35
Logistic Regression	5 min	.50	.72	.22
CV Features SVM	5 min	.50	.78	.28
CV Features LR	5 min	.50	.77	.27

Next Steps

Future avenues of exploration:

Different types of tuning at both the pre-processing step and model training steps

Alternative models to try on graphical data, particularly Artificial Neural Networks

Additional CV-based features, e.g. custom Haar cascades

Additional time-series features, e.g. weighted trailing feature stats baked into current observation

Further discretization of targets

Further “productionizing” of analytic processes as Python is very nice for this sort of thing:

- Web scraping a set of tickers
- Functions for more loops
- Grid searching optimal time slice and lookahead combinations

Questions?
