Machine Learning Engineer Nanodegree

Capstone Project: Trading Strategy based on PCA

George Fedorneac

September 21, 2018

I. Definition

Welcome to your capstone project.

I'm interested in machine learning and finance. The project is based on the mean-variance theory promoted by Harry Markowitz. Harry Markowitz, in 1952, published a paper on “Modern Portfolio Theory” for which he also received the Nobel Price in Economics. The main concepts are:

1. The investor's goal is to maximize return for any level of risk
2. Risk can be reduced by creating a diversified portfolio of unrelated assets

The project creates a stock portfolio based on ideas seen in the research paper from MIT “Principal Components as a Measure of Systemic Risk” by Mark Kritzman and coauthors published in 2010 (http://web.mit.edu/finlunch/Fall10/PCASystemicRisk.pdf).

I will be using historical stock prices as inputs. The same approach would also work with price forecast data instead of historical data. This project if driven by my desire to build a trading system that is modular and can be scaled; combination of machine learning models, reporting, and analysis to simplify the complexity of the financial markets.

Problem Statement

My problem statement is simple and direct. To build a stock portfolio script as a simple trading strategy. An abundance of information is available in the form of historical stock prices and company performance data, suitable for machine learning algorithms to process. By building a mathematical model for portfolio construction based on principal component analysis (PCA), I will expect the results to be quantifiable, measurable and replicable.

From the execution standpoint, these are the project main objectives:

- construct eigen-portfolios
- implement a measure of market systemic risk
- develop simple trading strategy

Metrics

I propose the metrics absorption ratio delta, sharpe ratio, annualized return, and annualized volatility. Absorption ratio predicts the systemic risk in the markets or particular investment portfolios by measuring the concentration of risk. It shows when markets are “fragile” and vulnerable to loss and when markets are resilient.

Using the absorption ratio, we can anticipate shifts in the portfolio volatility and exposure to loss or opportunity for gain.

Absorption ration equals the fraction of the total variance of a set of asset returns explained or “absorbed” by a fixed number of eigenvectors.

Metrics breakdown by creation point in the project:

Input:

- Daily stock closing prices

Calculated:

- Log daily returns (percentage)
- Absorption ratio and absorption ratio delta
- Annualized returns (average returns over 1 yr)
- Annualized volatility (volatility -standard deviation- over 1 yr)
- Sharpe ratio (annualized returns/ annualized volatility)

I will use absorption ratio delta to create the portfolio and sharpe ratio, annualized returns, and annualized volatility to measure the performance of the portfolio.

II. Analysis

Data Exploration

This project will use historical daily stocks prices from S&P 500 Index stock data. It contains 3493 rows of daily closing price of 419 US stocks including the S&P 500 Index spanning 13 years from 2000 to 2013. In addition, I will use one additional dataset for benchmarking. It is composed of stock closing prices of two indices VTI (Vanguard Total Stock Market ETF) and AGG(iShares Core U.S. Aggregate Bond ETF) spanning 13 years from 2003 to 2017.

Please note, initially I used 10 years of data, however I wanted to capture the 2000 dot-com market crash. Furthermore, the moving window for computing principal components is minimum 2 years.

As stated, the only feature in the dataset is made of daily stock prices, a numerical (float) data type. The price feature will undergo a few transformations to produce logarithmic daily returns (%) that I will use for modeling.

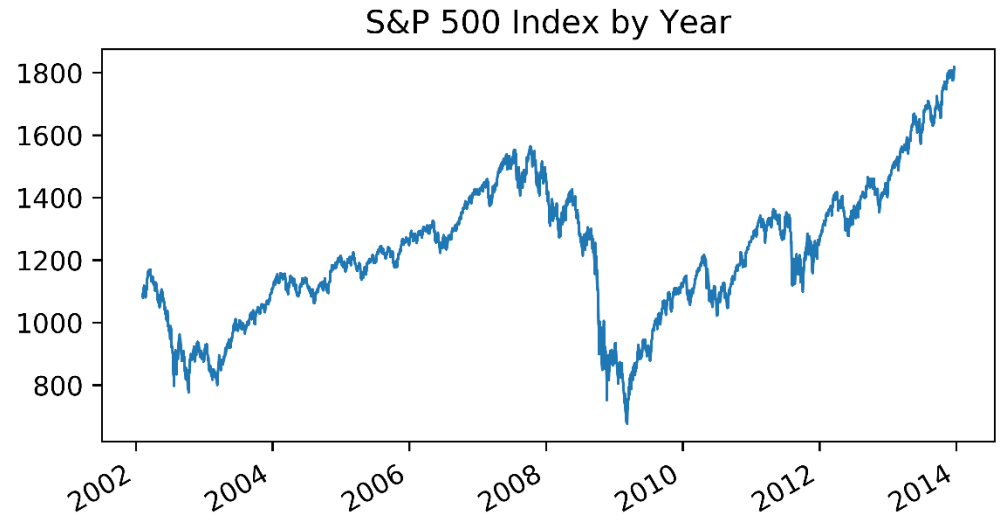
Exploratory Visualization

In this section, you will need to provide some form of visualization that summarizes or extracts a relevant characteristic or feature about the data.

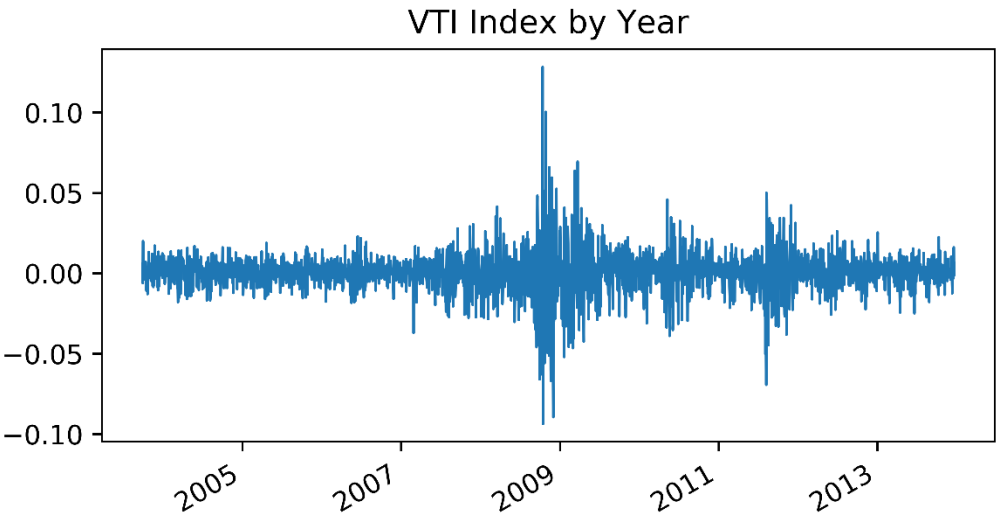
In the first dataset, we have the stocks prices of S&P 500 companies and the S&P 500 Index (SPX).

In the second dataset, we have two indices, VTI representing equity market and AGG index representing fixed income market.

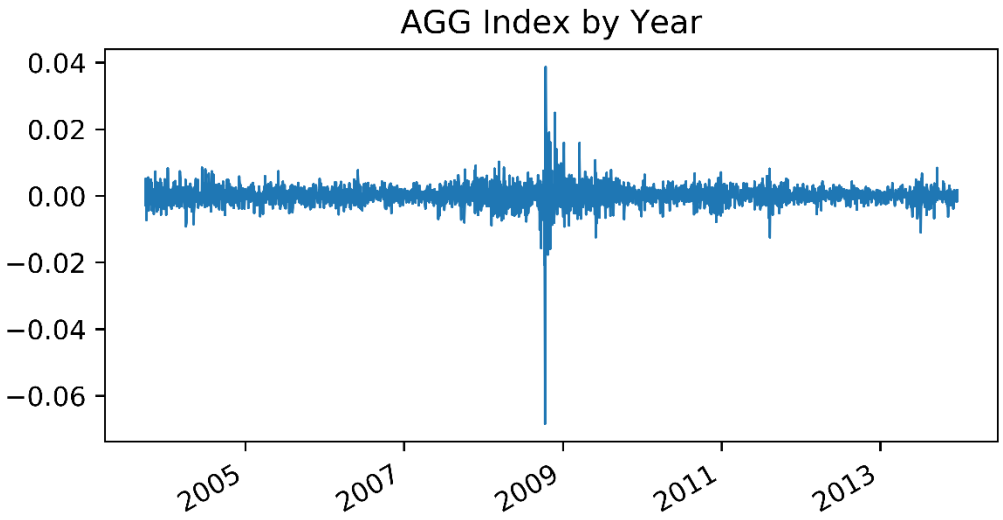
In the first chart, we see SPX index evolution over time. It's interesting to notice the big drop during the 2008 financial crisis.



In the second chart, I plot VTI index over time.



In the third chart, we have AGG index over time.



The second and third charts are a clear indication how both indices are a great measure of the market as a whole. The data is center around 0 with a strong deviation during the 2008 financial crisis.

Algorithms and Techniques

This project revolves around PCA as a dimensionality reduction technique to simplify the complexity of holding a portfolio of 420 stocks to holding just two index stocks and optimizing it as seldom as possible.

In a nutshell, the stock daily returns, the input feature, is calculated from daily closing prices and further log transformed to satisfy the Gaussian feature distribution required by the mean-variance portfolio theory. Following this transformation, I proceed to normalize the returns.

Now the data is ready to compute Absorption Ratio (AR). I do so by defining a moving look back window over which I get returns for computing PCA.

In order to calculate the absorption ratio, I use a window of 504 days (moving_window = 252 * 2) to estimate the covariance matrix and eigenvectors, and I fix the number of eigenvectors at approximately 1/5th the number of assets in the data sample.

The variances in the AR formula (numerator and denominator) are calculated with exponential weighting. This approach assumes that the market's memory of prior events fades away gradually as these events regress further into the past. The half time of the exponential weight decay is set to be half of the window; that is, 252 days (half_life = 252).

Next I compute AR Delta measure, generate the strategy weights and use them to build the portfolio trading strategy.

Formulas used in the project:

Formula for Absorption ratio:

$$AR = \frac{\sum_{i=1}^{n} \sigma_{Ei}^2}{\sum_{j=1}^{N} \sigma_{Aj}^2}$$

where,

- AR : Absorption ratio
- N : number of assets
- n : number of eigenvectors used to calculate absorption ratio
- σ_{Ei}^2 variance of the i-th eigenvector, sometimes called eigenportfolio
- σ_{Aj}^2 variance of the j-th asset

Formula for AR Delta:

$$AR\delta = \frac{AR_{15d} - AR_{1y}}{AR\sigma_{1y}}$$

where,

- AR_{15d} : moving average of the absorption ratio over 15 days
- AR_{1y} : moving average of the absorption ratio over 1 year
- $AR\sigma_{1y}$: standard deviation of the one-year absorption ratio
- $AR\delta$: AR Delta, also called `standardized shift in absorption ratio` in the research paper (Kritzman)

Formula for exponentially smoothed weights:

$$w_j = \frac{X_j}{\sum_{j=0}^{N} X_j}$$

where,

- X_j : a sequence where $j \subset [N, 0]$, an integer taking all values in the interval from 0 to N

$$X_j = e^{-\frac{log(2)}{H} \times j}$$

- w_j : a sequence of exponentially decaying weights
- H : is half-life which determines the speed of decay

Benchmark

The AR Delta trading strategy forms a portfolio of EQ and FI, following these simple rules:

- $-1\sigma < AR < +1\sigma$ 50 / 50 weights for EQ / FI
- $AR > +1\sigma$ 0 / 100 weights for EQ / FI
- $AR < -1\sigma$ 100 / 0 weights for EQ / FI

The strategy in words: if AR Delta is above 1 then invest in fixed income index, if AR Delta is below -1 invest in equity index only, otherwise invest 50/50.

I compute AR Delta strategy weights from the same data, apply it to the benchmark indices (VTI, AGG) compare it with the equal weights portfolio using the sharpe ratio, annualized return, annualized volatility measures. Based on the comparison I decide if I have to adjust the portfolio or use 50/50 or leave it as is.

I also compute the average yearly trades needed to optimize the portflio. As expected, the average number of trades per year is very low.

Definitions:

- Sharpe ratio is defined as annualized return divided by annualized volatility.
- Annualized return is the average amount of money earned by an investment each year over a given time period. It is calculated as an average to show what an investor would earn over a period of time if the annual return was compounded.
- Annualizing volatility is volatility (standard deviation) in annualized terms. We need to multiply our daily standard deviation by the square root of 252. This assumes there are 252 trading days in a given year because the markets are closed on weekends.

III. Methodology

Data Preprocessing

The data is stored in a csv file and contains daily stock closing prices for 419 stocks. It is simple and straight forward data, character for date and numeric for stock prices, however in order to properly use the price feature a number of transformations are needed.

1. Import data into pandas dataframe
2. Convert date column to pandas datetime and make it an index
3. Calculate daily log-returns
    - Log transform the prices
    - Calculate daily percentage price change of log returns
4. Normalize the returns

Now the data is ready to compute PCA and absorption ratio.

```
Stock prices shape (3493, 419)
Minimum Date 2000-01-27 00:00:00
Maximum Date 2013-12-20 00:00:00
```

	A	AA	AAPL	ABC	ABT
2000-01-27	46.1112	78.9443	3.9286	4.5485	13.7898
2000-01-28	45.8585	77.8245	3.6295	4.5485	14.2653
2000-01-31	44.5952	78.0345	3.7054	4.3968	14.5730
2000-02-01	47.8377	80.7640	3.5804	4.5333	14.7128
2000-02-02	51.5434	83.4934	3.5290	4.5788	14.7968

```
Stock prices shape (3508, 2)
Minimum Date 2003-09-29 00:00:00
Maximum Date 2017-08-31 00:00:00
```

Index	VTI	AGG
2003-09-29	0.005862	-0.002734
2003-09-30	-0.006036	0.005188
2003-10-01	0.020000	-0.000486
2003-10-02	0.004516	-0.001560
2003-10-03	0.010935	-0.007219

Implementation

The implementation process flow, step by step, in chronological order:

- Import data into pandas dataframe.
- Convert date column to pandas datetime and make it an index.
- Calculate daily log-returns
  - Log transform the prices
  - Calculate daily percentage price change of log returns
- Normalize the returns
  - Executed by function `normalize_returns` that centers and divides by standard deviation raw asset returns data
- Implement function `get_exponential_weighting` used to calculate exponentially smoothed weights. This function uses numpy methods to implement the exponential smoothing formula.
- Implement function `get_pca_absortion_ratio` used to compute PCA and absorption ratio. This function loops backward through the data in increments, creates covariance matrices and passes them to sklearn PCA to produce principal components and absorption ratio using standard numpy and pandas methods.
- Compute AR delta using simple moving averages and standard deviation of the one-year absorption ratio.
- Implement the AR Delta trading strategy function `get_weights` using these rules: if AR Delta is above 1 then invest in fixed income index, if AR Delta is below -1 invest in equity index only, otherwise invest 50/50.
- Compute the trading strategy portfolio weights and the average number of trades per year to optimize the portfolio. These weights will be used to build the portfolio strategy and compare them against a 50/50 split.
- Calculate the performance of the strategy using backtesting. Backtesting assesses the viability of a trading strategy by discovering how it would play out with historical data

    - Implement function run_backtest_strategy which given a DataFrame of strategy weights and a DataFrame of assets returns annualized return, volatility and Sharpe ratio of a strategy. This function uses simple numpy methods.
- Calculate portfolio returns and return portfolio strategy performance
    - I load the benchmark dataset comprised of two index stocks (VTI and AGG). These indices are designed to represent the whole equity market in the case of VTI and fixed income market in the case of AGG.
    - Run run_backtest_strategy once for the strategy weights calculated in the model and once more for 50/50 weights and compare their annualized return, volatility and Sharpe ratio. I select the weights that give the higher returns with the lowest volatility.

Refinement

In the refinement section of the project, I experimented with lookback window step size parameter. The research paper, from which I based this project, uses one single day.

Challenge: I order to test other values for step_size I had to modify the code that computes the moving averages and standard deviation in the AR Delta formula to use less than a year, as I don't get enough return values. This occurs because in order to estimate the covariance matrix and eigenvectors there is need for two years of data plus another year to estimate the standard deviation of the one-year moving average. Therefore, the more I would increase the step size window the more data is required.

Old code snippet:

```
# following Kritzman and computing AR_delta = (15d_AR -1yr_AR) / AR_sigma
ts_ar = ts_absorb_ratio
ar_mean_1yr = ts_ar.rolling(252).mean() # compute moving average of the absorption ratio over 1 year
ar_mean_15d = ts_ar.rolling(15).mean()  # compute moving average of the absorption ratio over 15 days
ar_std_1yr = ts_ar.rolling(252).std()   # standard deviation of the one-year absorption ratio
ar_delta = (ar_mean_15d - ar_mean_1yr) / ar_std_1yr    # standardized shift in absorption ratio

df_plot = pd.DataFrame({'AR_delta': ar_delta.values, 'AR_1yr': ar_mean_1yr.values, 'AR_15d': ar_mean_15d.values},
                       index=ts_ar.index)
df_plot = df_plot.dropna()
if df_plot.shape[0] > 0:
    df_plot.plot(figsize=(12, 6), title='Absorption Ratio Delta', linewidth=3)
    plt.savefig('AR_Delta.png', dpi=900)
```

New code snippet:

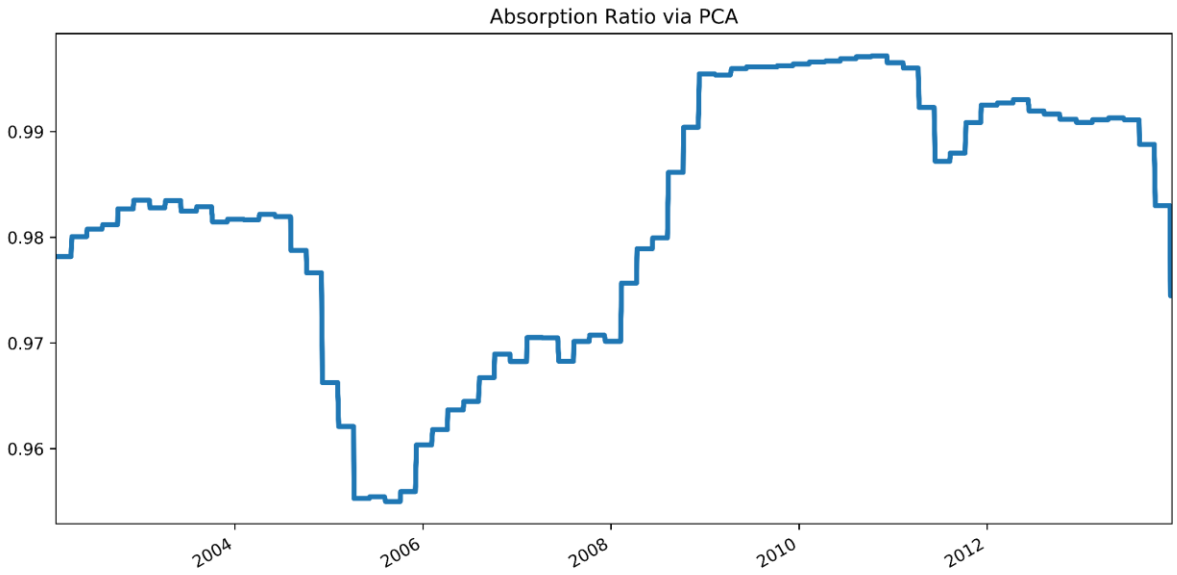
```
# following Kritzman and computing AR_delta = (15d_AR -1yr_AR) / AR_sigma
ts_ar = ts_absorb_ratio
# consider ts_ar size when computing rolling averages
max_i = ts_ar.count()

if max_i > 252:
    counter = 252
else:
    counter  = max_i

ar_mean_1yr = ts_ar.rolling(counter).mean() # compute moving average of the absorption ratio over 1 year
ar_mean_15d = ts_ar.rolling(15).mean()  # compute moving average of the absorption ratio over 15 days
ar_std_1yr = ts_ar.rolling(counter).std()   # standard deviation of the one-year absorption ratio
ar_delta = (ar_mean_15d - ar_mean_1yr) / ar_std_1yr    # standardized shift in absorption ratio

df_plot = pd.DataFrame({'AR_delta': ar_delta.values, 'AR_1yr': ar_mean_1yr.values, 'AR_15d': ar_mean_15d.values},
                       index=ts_ar.index)
df_plot = df_plot.dropna()
if df_plot.shape[0] > 0:
    df_plot.plot(figsize=(12, 6), title='Absorption Ratio Delta', linewidth=3)
    plt.savefig('AR_Delta_step_3.png', dpi=900)
```

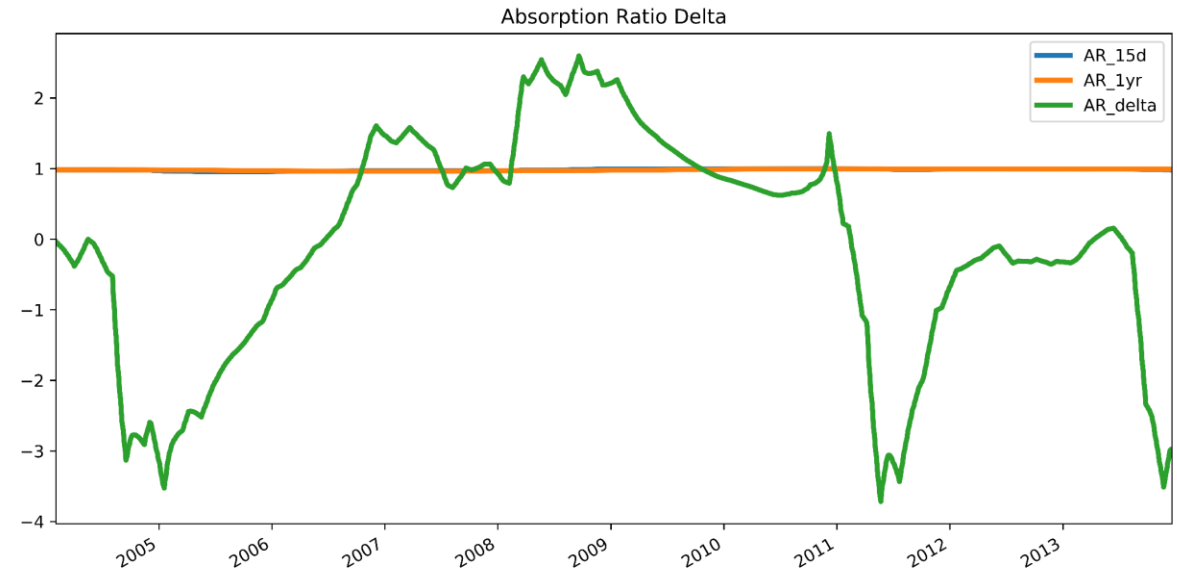
Adjusting step_size parameter:

Step_size = 2

Conclusion: When using two days as step_size the model recommends keeping computed model weights. As we notice in the charts, both the Absorption Ratio and Absorption Ratio Delta follow the market in similar manner wityh using one day step size.

- Absorption Ratio strategy: 0.052863750098499945 0.0986259608194 0.536002383747
- Equally weighted: 0.02431651866251021 0.102784078399 0.236578651492
- Average number of trades per year 1.50

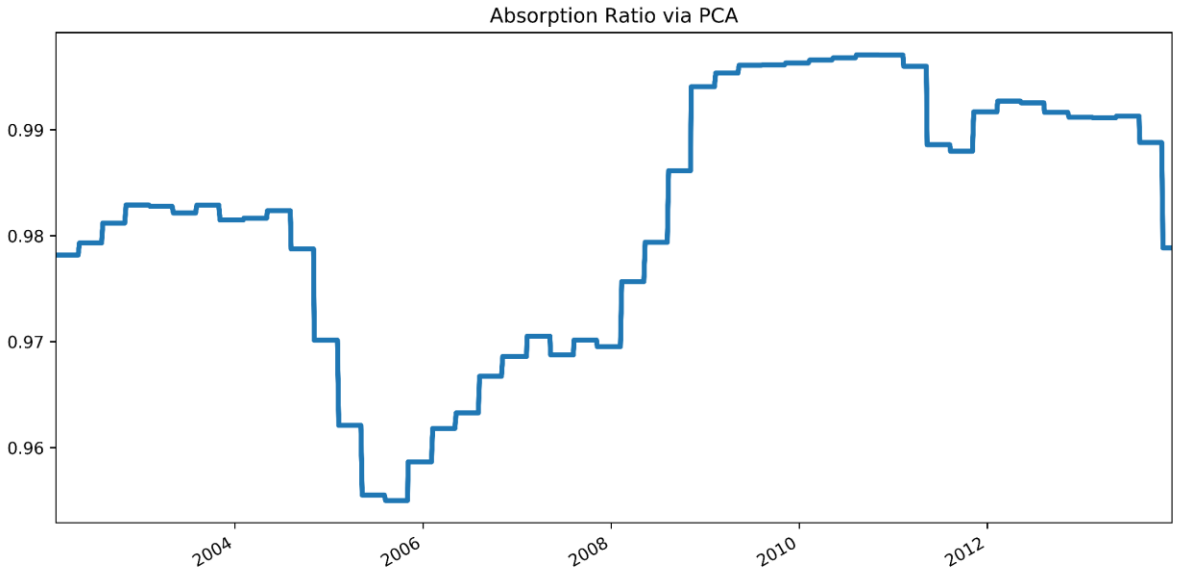


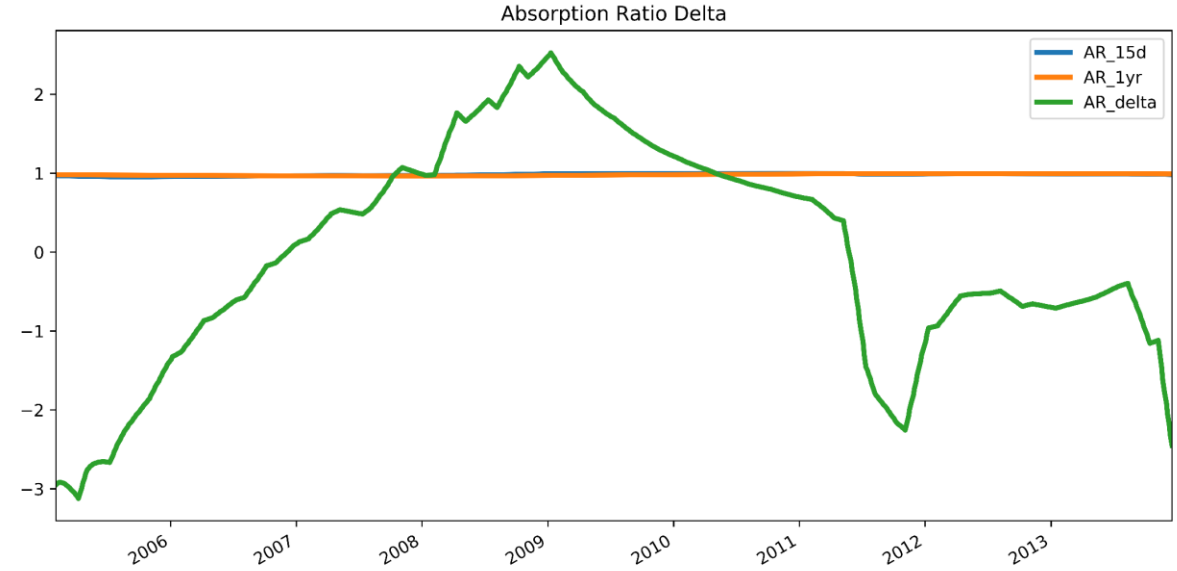


Step_size = 3

Conclusion: When using three days as step_size the model recommends keeping computed model weights.

- Absorption Ratio strategy: 0.08289358781369592 0.11120718432 0.745397775514
- Equally weighted: 0.018666151961176315 0.102621577463 0.181893052345
- Average number of trades per year 0.89

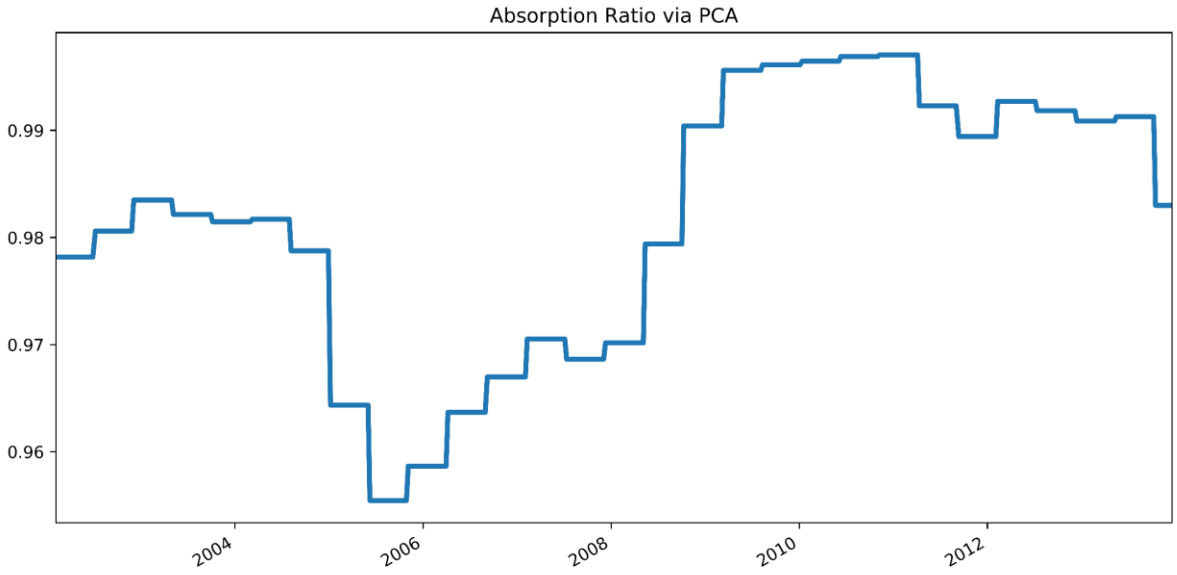


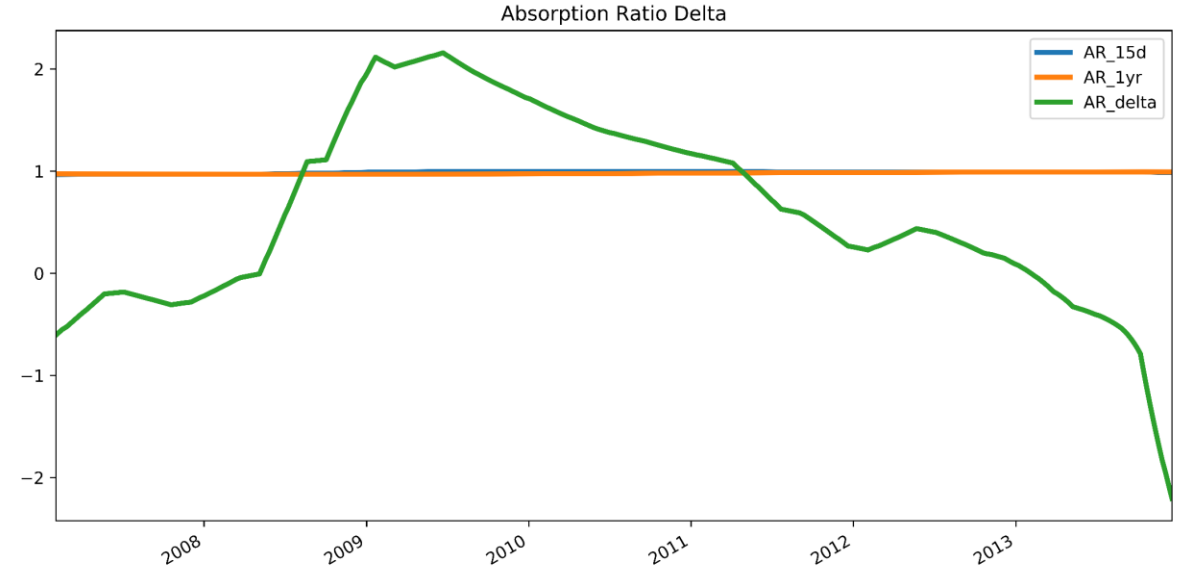


Step_size = 5

Conclusion: When using a week as step_size the model recommends keeping the 50/50 weights.

- Absorption Ratio strategy: -0.03849888738388968 0.0709232959552 -0.542824284537
- Equally weighted: 0.04012609626367268 0.11036081326 0.363590074033
- Average number of trades per year 0.43



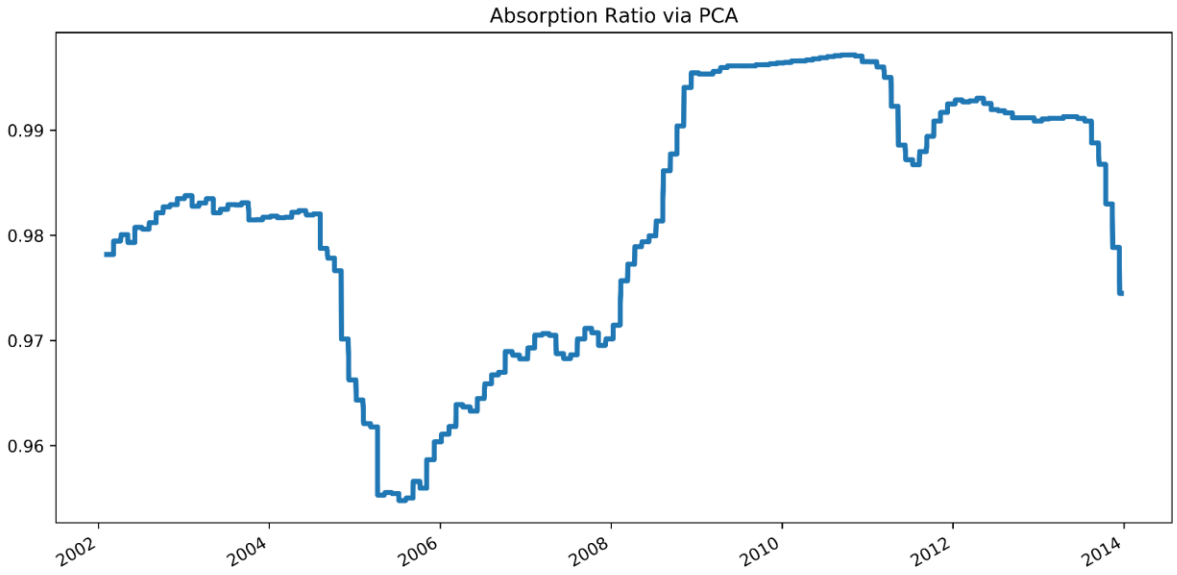


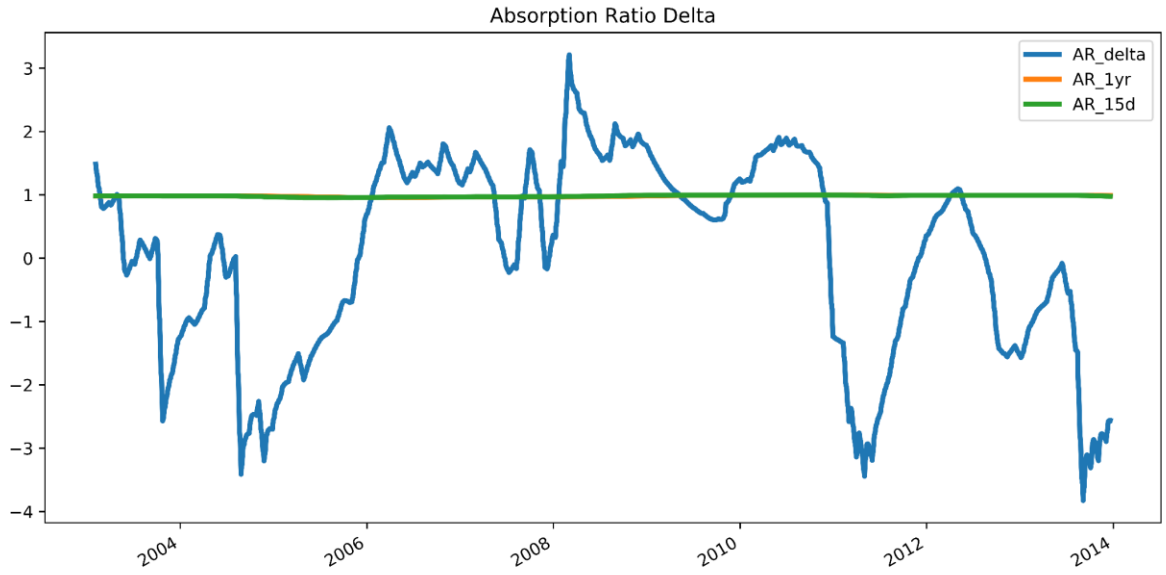
Best solution:

Step_size = 1

Conclusion: This is the best solution, although it requires a bit more maintenance with 2 trades per year. When using two days as step_size the model recommends keeping computed model weights.

- Absorption Ratio strategy: 0.0908649651277521 0.0993016731046 0.915039619041
- Equally weighted: 0.07471976752887036 0.102003147401 0.732524137078
- Average number of trades per year 2.18





	EQ	FI	VTI	AGG	EQ_returns	FI_returns
2003-09-29	0.5	0.5	0.005862	-0.002734	0.002931	-0.001367
2003-09-30	0.5	0.5	-0.006036	0.005188	-0.003018	0.002594
2003-10-01	0.5	0.5	0.020000	-0.000486	0.010000	-0.000243
2003-10-02	0.5	0.5	0.004516	-0.001560	0.002258	-0.000780
2003-10-03	0.5	0.5	0.010935	-0.007219	0.005468	-0.003610

IV. Results

Model Evaluation and Validation

To recap, the final model and its implementation is based on PCA and a measure of market risk called absorption ratio derived from the total variance of a set of PCA eigenvectors. I have opted for PCA as a dimensionality reduction algorithm because it is straightforward, easy to implement, and powerful at the same time.

In addition, principal components analysis is an unsupervised learning technique. In today's environment with exponential increase in data, unsupervised machine learning techniques are become more important.

As stated in the Refinement section of the project some model enhancements require more and more input data. I have used different financial data than the research paper yet arrived at similar conclusions. A high absorption ratio implies that financial markets are relatively compact. A low absorption ratio suggests that markets are less tightly coupled and therefore less vulnerable to shocks.

The research paper used financial datasets from the MSCI USA price index (index based on 51 U.S. industries), global financial stocks, and real estate prices. I have used S&P 500 index stocks and staple benchmark index (VTI, AGG) for model validation.

Justification

Although most investors might be reluctant to shift entirely in or out of stocks given a single market signal, this project reinforces the research paper and does offer sound evidence of the potential value of the absorption ratio as a market timing signal.

The end goal of portfolio optimization is to produce the best annualized returns for the risk/volatility selected. In other words, it is a measure of risk vs. return.

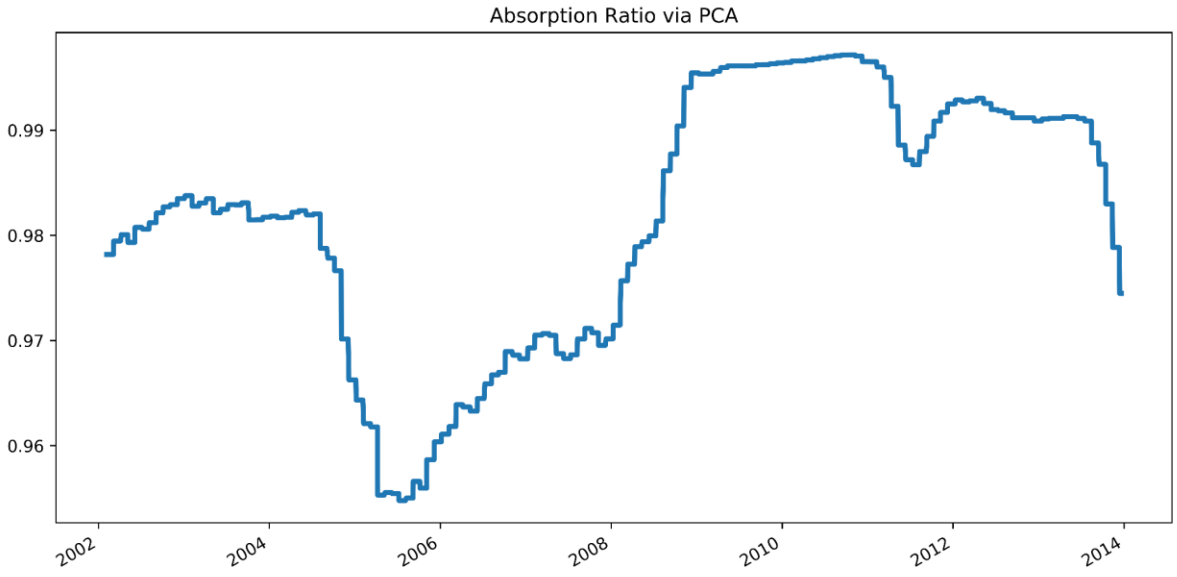
The model results comparison with the benchmark is as simple as comparing the annualized returns, volatility and sharpe ratio between the absorption ratio strategy with the 50/50 split strategy.

V. Conclusion

Free-Form Visualization

The most important aspect of the project is the power of absorption ratio to follow the market outlier events such as 2008 financial crisis. Significant increases in the absorption ratio are followed by significant stock market losses on average, while significant decreases in the absorption ration are followed by significant gains. This differential performance suggests that it could be profitable to reduce stock exposure after an increase in the absorption ratio and to raise exposure to stocks after the absorption ratio falls.

This is nicely illustrated below:



Reflection

In this project, I have explored a method of inferring systemic risk from asset prices by means of principal component analysis. Using stock prices as the only input feature I built covariance matrices of asset returns, fit them to PCA, use the results to compute the absorption ratio and built a simple trading strategy.

The most interesting part of the project was to apply this concept of market risk measure named absorption ratio to Principal Component Analysis (PCA) a statistical method of dimension reduction used to reduce the complexity of a data set while minimizing information loss.

The project and research paper offer indication that the absorption ratio effectively capture market fragility.

1. Most significant U.S. stock market drawdowns were preceded by spikes in the absorption ratio.
2. Stock prices, on average, depreciated significantly following spikes in the absorption ratio and, on average, appreciated significantly in the wake of sharp declines in the absorption ratio.
3. The absorption ratio systematically rose in advance of market turbulence.

Improvement

An aspect of this project implementation that I can see enhanced comes from the PCA is a linear machine learning algorithm. I have embarked on the AI and machine learning journey as a hobby.

In the future I plan on implementing a linear autoencoder instead of PCA by setting up a neural network using tensorflow fully connected layer with no activation function, a standard mean square error loss function, and a popular optimizer such as AdamOptimizer. After that, I may explore further with non-linearity by adding a sigmoid activation function and more layers.

From a financial modeling perspective there is potential for discovering unseen patterns if we could model the data at lower intervals such as hourly, by second, or even by tick. This requires humongous computational power. I have took a few lectures in quantum computing and PCA seems to be a machine learning technique transferable to quantum machine learning.