Portfolio Allocation with Machine Learning

Final Presentation

Lvcheng Dong
Guangqi Li
Changting Song

Team G1, PAFBMA-402002

March 5, 2023

Section 1

Introduction

Contents

- What is *portfolio optimization*?

- What is *portfolio optimization*?
- How people do it?
  - traditional: mean-variance optimization
  - recent: machine learning, risk parity

- What is *portfolio optimization*?
- How people do it?
    - traditional: mean-variance optimization
    - recent: machine learning, risk parity
- Key about portfolio: **diversify!**
    - the GameStop short squeeze in early 2021
    - pandemic-induced market volatility in 2020

- What is *portfolio optimization*?
- How people do it?
  - traditional: mean-variance optimization
  - recent: machine learning, risk parity
- Key about portfolio: **diversify!**
  - the GameStop short squeeze in early 2021
  - pandemic-induced market volatility in 2020
- How well do we do?
  - benchmark assets (*S&P 500*)

- What is *portfolio optimization*?
- How people do it?
  - traditional: mean-variance optimization
  - recent: machine learning, risk parity
- Key about portfolio: **diversify!**
  - the GameStop short squeeze in early 2021
  - pandemic-induced market volatility in 2020
- How well do we do?
  - benchmark assets (*S&P 500*)
- Importance of data
  - to make accurate and robust estimations

"In God we trust, all others must bring ***data****."* (W. Edwards Deming)

Contents

10 Assets from Different Sectors

- **Source**: `yfinance`
- **Time**: Jan. 2nd, 2019 to May. 26th, 2019 (100 days)
- Use `Adj Close` to compute returns
- We choose 10 stocks from 10 market sectors to diversify

10 Assets from Different Sectors

Datasets

- **Source**: `yfinance`
- **Time**: Jan. 2nd, 2019 to May. 26th, 2019 (100 days)
- Use `Adj Close` to compute returns
- We choose 10 stocks from 10 market sectors to diversify

Energy	Material	Industrial	C. Cyclical	Healthcare
SHEL	CBT	UPS	TSLA	PFE
Technology	**Real Estate**	**Communication**	**C. Defensive**	**Financial**
AAPL	EQIX	NFLX	WMT	GS

Table 1: Stock tickers for 10 Assets

Our Benchmark

S&P 500

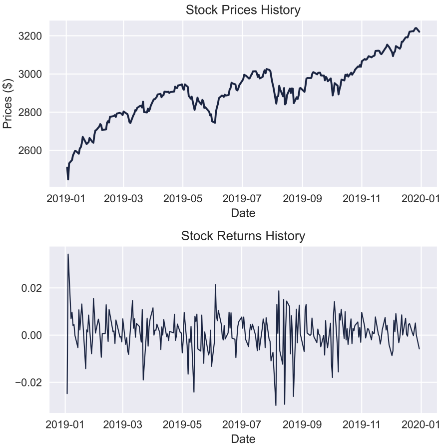


Figure 1: S&P 500

Tear sheet for S&P 500 in 2019:

Returns	28.5%
Volatility	12.5%
Sharpe Ratio	2.07
Max Drawdown	−6.8%

Table 2: Simple Tear Sheet

Our Benchmark

S&P 500

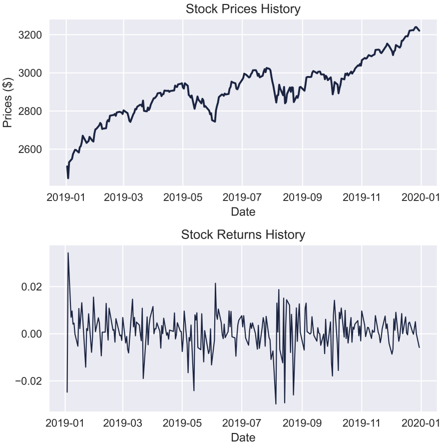


Figure 1: S&P 500

Tear sheet for S&P 500 in 2019:

Returns	28.5%
Volatility	12.5%
Sharpe Ratio	2.07
Max Drawdown	-6.8%

Table 2: Simple Tear Sheet

Can we do better?

Why these stocks?

Why these stocks?

This dataset is ideal for portfolio optimization as it includes stocks from a variety of sectors, allowing for diversification and risk reduction.

Why these stocks?

This dataset is ideal for portfolio optimization as it includes stocks from a variety of sectors, allowing for diversification and risk reduction.

Some Reasons

Why these stocks?

This dataset is ideal for portfolio optimization as it includes stocks from a variety of sectors, allowing for diversification and risk reduction.

Some Reasons

- Apple & Tesla: high market capitalization and popularity

Why these stocks?

This dataset is ideal for portfolio optimization as it includes stocks from a variety of sectors, allowing for diversification and risk reduction.

Some Reasons

- Apple & Tesla: high market capitalization and popularity
- Pfizer: one of the largest pharmaceutical companies in the world

Why these stocks?

This dataset is ideal for portfolio optimization as it includes stocks from a variety of sectors, allowing for diversification and risk reduction.

Some Reasons

- Apple & Tesla: high market capitalization and popularity
- Pfizer: one of the largest pharmaceutical companies in the world
- Netflix: a popular streaming service provider

Why these stocks?

This dataset is ideal for portfolio optimization as it includes stocks from a variety of sectors, allowing for diversification and risk reduction.

Some Reasons

- Apple & Tesla: high market capitalization and popularity
- Pfizer: one of the largest pharmaceutical companies in the world
- Netflix: a popular streaming service provider
- Walmart: one of the largest retail companies in the world

Section 2

Methodology

Contents

Our Framework

Our Framework

- **Estimation Side**
  - Several prediction methods
  - Based on *history information* ($\{\boldsymbol{x_t}\}_{-\infty}^{n}$) to predict $\hat{\boldsymbol{x}}_{\boldsymbol{t+1}}$
  - Provide $\hat{\boldsymbol{x}}_{\boldsymbol{t+1}}$ and covariance matrix for *optimization* to process

Our Framework

- **Estimation Side**
  - Several prediction methods
  - Based on *history information* ($\{\boldsymbol{x_t}\}_{-\infty}^{n}$) to predict $\hat{\boldsymbol{x}}_{\boldsymbol{t+1}}$
  - Provide $\hat{\boldsymbol{x}}_{\boldsymbol{t+1}}$ and covariance matrix for *optimization* to process
- **Optimization Side**
  - Based on predictions ($\hat{\boldsymbol{x}}_{\boldsymbol{t+1}}$ and Σ), return portfolio weights ($\boldsymbol{w_{t+1}}$)

Our Framework

- **Estimation Side**
  - Several prediction methods
  - Based on *history information* ($\{\boldsymbol{x_t}\}_{-\infty}^{n}$) to predict $\hat{\boldsymbol{x}}_{\boldsymbol{t+1}}$
  - Provide $\hat{\boldsymbol{x}}_{\boldsymbol{t+1}}$ and covariance matrix for *optimization* to process
- **Optimization Side**
  - Based on predictions ($\hat{\boldsymbol{x}}_{\boldsymbol{t+1}}$ and Σ), return portfolio weights ($\boldsymbol{w_{t+1}}$)
- **Validation & Evaluation**
  - Since we do actually know every $\boldsymbol{x_{t+1}}$, use it to *backtest* our model
  - Compute daily returns, use `pyfolio` to generate reports

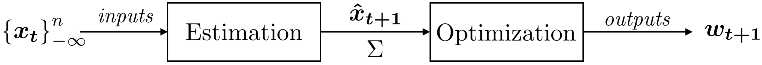


Figure 2: Flowchart of our Project

Contents

Support Vector Regression

- Fitting method
  - finding a hyperplane in a N dimension space
  - some spots named support vector near the hyperplane can influence the hyperplane
  - our purpose is to figure out a best hyperplane
- Steps
  - ascertain kernel function
  - try some possible parameters

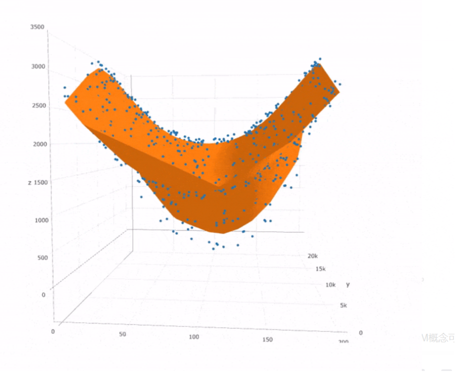


Figure 3: SVR model in the 3D space

Long Short-Term Memory

- Merits
  - memory cells and gates
  - deal with gradient loss
  - extract long term dependency
- Our setup
  - tensorflow, `Sequential()`
  - 2 LSTM layers + 2 FC layers

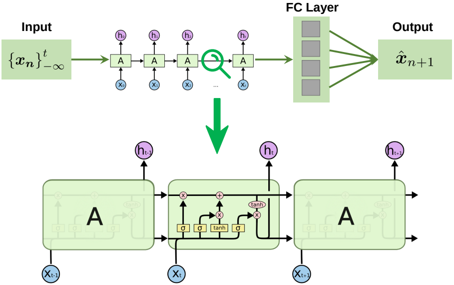


Figure 4: Structure of LSTM

Mean Variance Optimization

- Risk and Return based optimization
  - minimising the risk given a specified return
  - maximising the return given a specified risk

- Returns

$$\sigma_p^2 = \sum_i \sum_j w_i w_j \sigma_{ij} \mathrm{E}\left(R_p\right)$$
$$= \sum_i w_i \mathrm{E}\left(R_i\right)$$

- Covariance

$$\sigma_p^2 = \sum_i \sum_j w_i w_j \sigma_{ij}$$

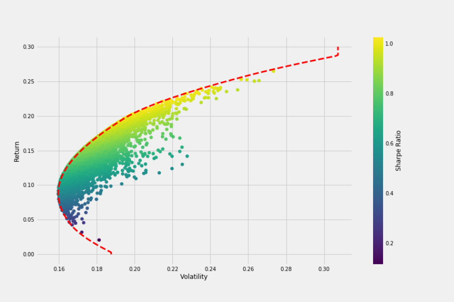


Figure 5: Efficient Frontier

Covariance Shrinkage

- Simply `df.cov()`?

Covariance Shrinkage

- Simply `df.cov()`?
  - limited sample size, noisy data, and non-stationarity

Covariance Shrinkage

- Simply `df.cov()`?
  - limited sample size, noisy data, and non-stationarity
- Two improvement
  - Ledoit-Wolf shrinkage: basically regularization
  - Oracle Approximating shrinkage: more of a data-driven method

Covariance Shrinkage

- Simply `df.cov()`?
  - limited sample size, noisy data, and non-stationarity
- Two improvement
  - Ledoit-Wolf shrinkage: basically regularization
  - Oracle Approximating shrinkage: more of a data-driven method

For the benefit of low computational cost, we opt for Ledoit-Wolf method.

Covariance Shrinkage

- Simply `df.cov()`?
  - limited sample size, noisy data, and non-stationarity
- Two improvement
  - Ledoit-Wolf shrinkage: basically regularization
  - Oracle Approximating shrinkage: more of a data-driven method

For the benefit of low computational cost, we opt for Ledoit-Wolf method.

Ledoit-Wolf Shrinkage

Let $\hat{\Sigma}_o$ denote the target matrix and let $\hat{\Sigma}_S$ be the sample covariance matrix, and δ is the shrinkage intensity parameter that balances the **trade-off** between estimation **bias and variance**. Construct new covariance matrix as

$$\hat{\Sigma}_{LW} = \delta\hat{\Sigma}_o + (1-\delta)\hat{\Sigma}_S$$

The optimal value of δ can be obtained through **cross-validation**.

Covariance Shrinkage

- Simply `df.cov()`?
    - limited sample size, noisy data, and non-stationarity
- Two improvement
    - Ledoit-Wolf shrinkage: basically regularization
    - Oracle Approximating shrinkage: more of a data-driven method

For the benefit of low computational cost, we opt for Ledoit-Wolf method.

Ledoit-Wolf Shrinkage

Let $\hat{\Sigma}_o$ denote the target matrix and let $\hat{\Sigma}_S$ be the sample covariance matrix, and δ is the shrinkage intensity parameter that balances the **trade-off** between estimation **bias and variance**. Construct new covariance matrix as

$$\hat{\Sigma}_{LW} = \delta\hat{\Sigma}_o + (1-\delta)\hat{\Sigma}_S$$

The optimal value of δ can be obtained through **cross-validation**.

Provided in the `PyPortfolioOpt` or `sklearn`.

Section 3

Implementation

Contents

Main idea

We want to improve the performance of MVO.

- We need to combine ML/DL with MVO
- We need to figure out the performance of each method and make comparison between them.

Main idea

We want to improve the performance of MVO.

- We need to combine ML/DL with MVO
- We need to figure out the performance of each method and make comparison between them.
- Expected Return
  - Historical average
  - ML (SVR)
  - DL (LSTM)

Main idea

We want to improve the performance of MVO.

- We need to combine ML/DL with MVO
- We need to figure out the performance of each method and make comparison between them.
- Expected Return
  - Historical average
  - ML (SVR)
  - DL (LSTM)
- Covariance Matrix
  - Sample covariance matrix
  - Covariance shrinkage

Contents

Basic Steps
SVR

- Data prepossessing
  - first 80% training and last 20% test
  - normalization for stock price

Basic Steps
SVR

- Data prepossessing
  - first 80% training and last 20% test
  - normalization for stock price
- Correlation among different variables

Basic Steps
SVR

- Data prepossessing
  - first 80% training and last 20% test
  - normalization for stock price
- Correlation among different variables
- How we train the data
  - input:open, high, low price and volume
  - output: close price

Basic Steps
SVR

- Data prepossessing
  - first 80% training and last 20% test
  - normalization for stock price
- Correlation among different variables
- How we train the data
  - input:open, high, low price and volume
  - output: close price
- Model buildup
  - Kernel function selecting
  - Parameters choosing

Basic Steps
SVR

- Data prepossessing
  - first 80% training and last 20% test
  - normalization for stock price
- Correlation among different variables
- How we train the data
  - input:open, high, low price and volume
  - output: close price
- Model buildup
  - Kernel function selecting
  - Parameters choosing
- Overfitting
  - grid and randomized search

Basic Steps
SVR

- Data prepossessing
  - first 80% training and last 20% test
  - normalization for stock price
- Correlation among different variables
- How we train the data
  - input:open, high, low price and volume
  - output: close price
- Model buildup
  - Kernel function selecting
  - Parameters choosing
- Overfitting
  - grid and randomized search
- Evaluation
  - MSE RMSE and R^2
  - forecasting plot

Correlation

Based on SVR

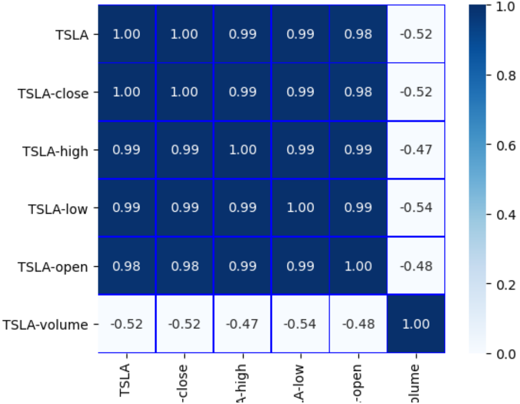


Figure 6: Correlation among Different Variables

Some problems during fitting

Based on SVR

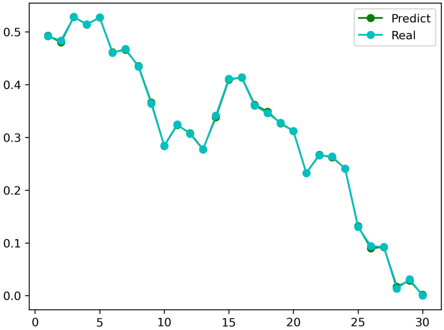


Figure 7: Overfitting

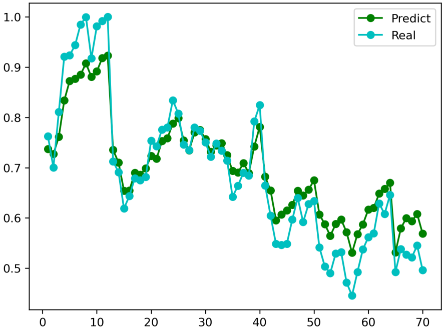


Figure 8: Underfitting

Parameters selecting
Based on SVR

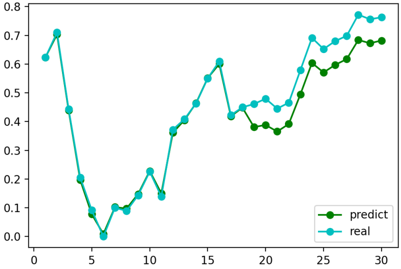


Figure 9: Stock price of PFE in 30 days

1. The simple things are often the best one
2. Persistence provides great feedback

Basic Steps
LSTM

- train-test split
  - first 80% training, last 20% test

Basic Steps
LSTM

- train-test split
  - first 80% training, last 20% test
- how the model is trained?
  - inputs: $\{[x_{t-6} : x_t] \in \mathbb{R}^7 \mid t \in \text{training set}\}$
  - outputs: $\{x_{t+1} \mid t \in \text{training set}\}$

Basic Steps
LSTM

- train-test split
  - first 80% training, last 20% test
- how the model is trained?
  - inputs: $\left\{[x_{t-6} : x_t] \in \mathbb{R}^7 \mid t \in \text{training set}\right\}$
  - outputs: $\{x_{t+1} \mid t \in \text{training set}\}$
- model buildup
  - tensorflow, `Sequential()`
  - 2 LSTM layers + 2 FC layers
  - 30 epochs

- train-test split
  - first 80% training, last 20% test
- how the model is trained?
  - inputs: $\left\{[x_{t-6} : x_t] \in \mathbb{R}^7 \mid t \in \text{training set}\right\}$
  - outputs: $\{x_{t+1} \mid t \in \text{training set}\}$
- model buildup
  - tensorflow, `Sequential()`
  - 2 LSTM layers + 2 FC layers
  - 30 epochs
- evaluation
  - RMSE, MAE, R^2
  - make plots

- Take one stock `AAPL` as an example

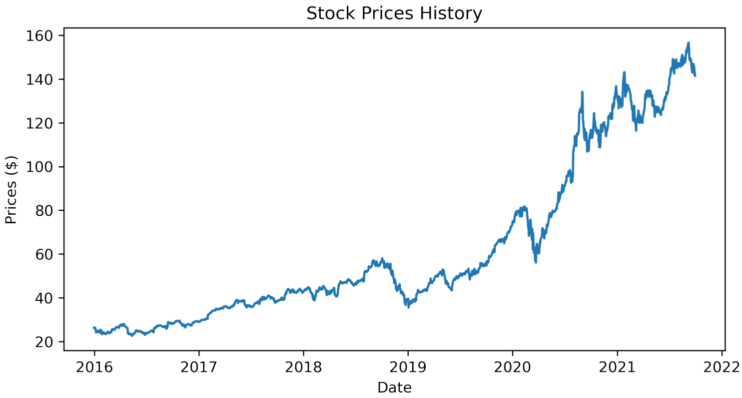


Figure 10: Adj Close of Apple

Stock Returns Estimation

Based on LSTM

- Plot the # *of epochs* against the *loss*
- We choose 10 epochs for further predictions

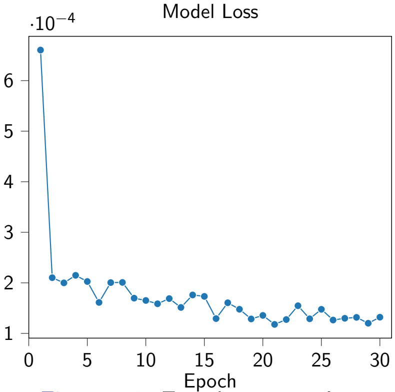


Figure 11: Epochs versus Loss

Prediction Results Demo

Based on LSTM

- LSTM proves to be quite accurate and robust.

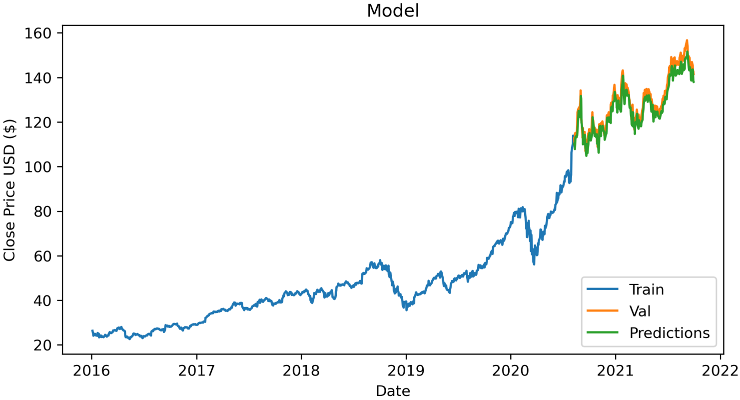


Figure 12: Demo Prediction

Covariance Estimation

We estimate covariance matrix $\hat{\Sigma}$ as below, using Ledoit-Wolf shrinkage.

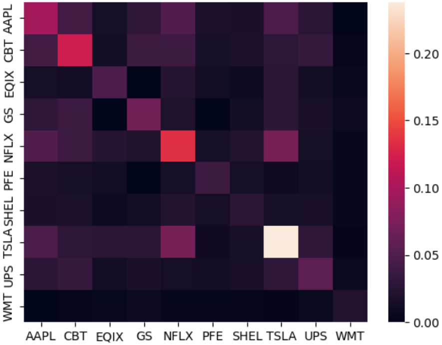


Figure 13: Covariance Heatmap

Section 4

Evaluation of Effectiveness

Efficient Frontier

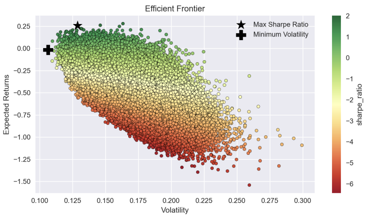


Figure 14: Efficient Frontier(SVR)

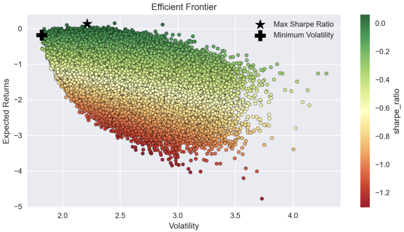


Figure 15: Efficient Frontier(LSTM)

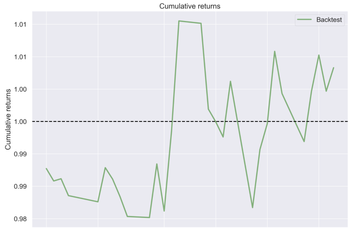
Results

	S&P 500	1/n	average	SVR	LSTM
Returns	−21.9%	−42.8%	7.4%	16.8%	19.1%
Volatility	12.1%	16%	12.6%	12.5%	13.1%
Sharpe Ratio	−1.98	−3.42	0.63	1.30	1.40
Max Drawdown	−4.5%	−7.2%	−2.8%	−2.7%	−3.4%

Table 3: Annualized Tear Sheet for *Apr 14 to May 24, 2019*

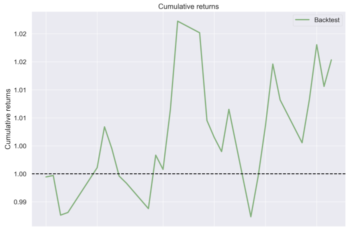
Accumulated Return

(a) 1/n



(b) Average Return

(c) Support Vector Regression



(d) Long Short Term Memory

Section 5

Conclusion

Discussion of Results

Discussion of Results

Highlight of Our Project

- Out-performance of market benchmark
- Positive growth during a bearish period
- Various econometrics and machine learning techniques

Discussion of Results

Highlight of Our Project

- Out-performance of market benchmark
- Positive growth during a bearish period
- Various econometrics and machine learning techniques

Further Developments

- Incorporate macroeconomic factors
- Explore constraints (transaction cost, short-position, leverage, etc.)

Discussion of Results

Highlight of Our Project

- Out-performance of market benchmark
- Positive growth during a bearish period
- Various econometrics and machine learning techniques

Further Developments

- Incorporate macroeconomic factors
- Explore constraints (transaction cost, short-position, leverage, etc.)

Acknowledgement

- Reinforce what we learn
- Practicality and applicability in real-world scenarios
- Shed light on the potential of quant-techniques in portfolio optimization

Member Contributions

Lvcheng Dong	Guangqi Li	Changting Song
Introduction	Optimization	ARIMA
LSTM Model	Tear Sheet Reports	SVR Model
Conclusion	Evaluation	Conclusion
Slides, LaTeX Support	Slides	Slides

Table 4: Group Member Contribution Sheet

Thanks for your attention!