# Data-Driven Portfolio Optimization Utilizing Machine Learning

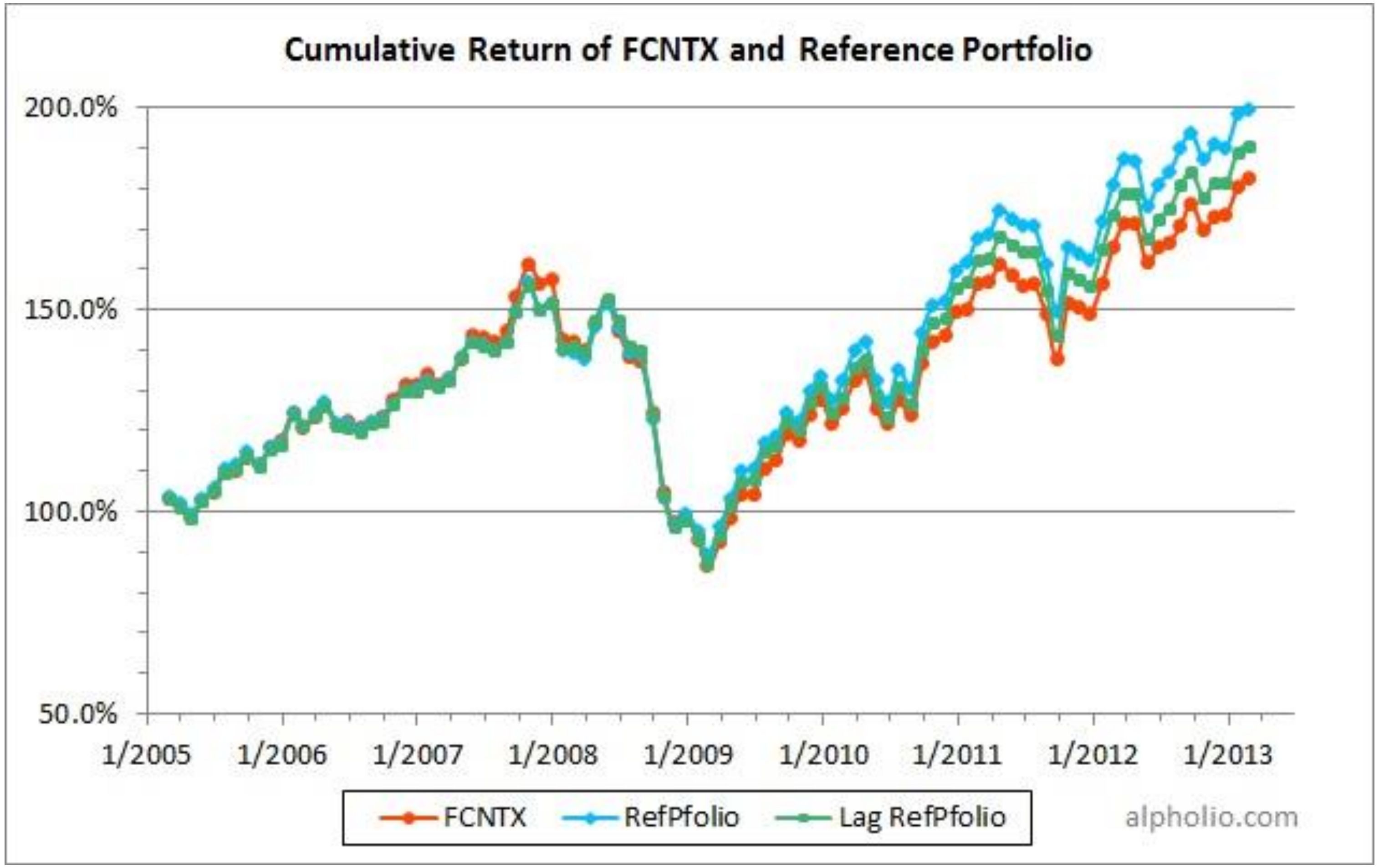
MELINDA HSIEH

RIDER UNIVERSITY

# Topics

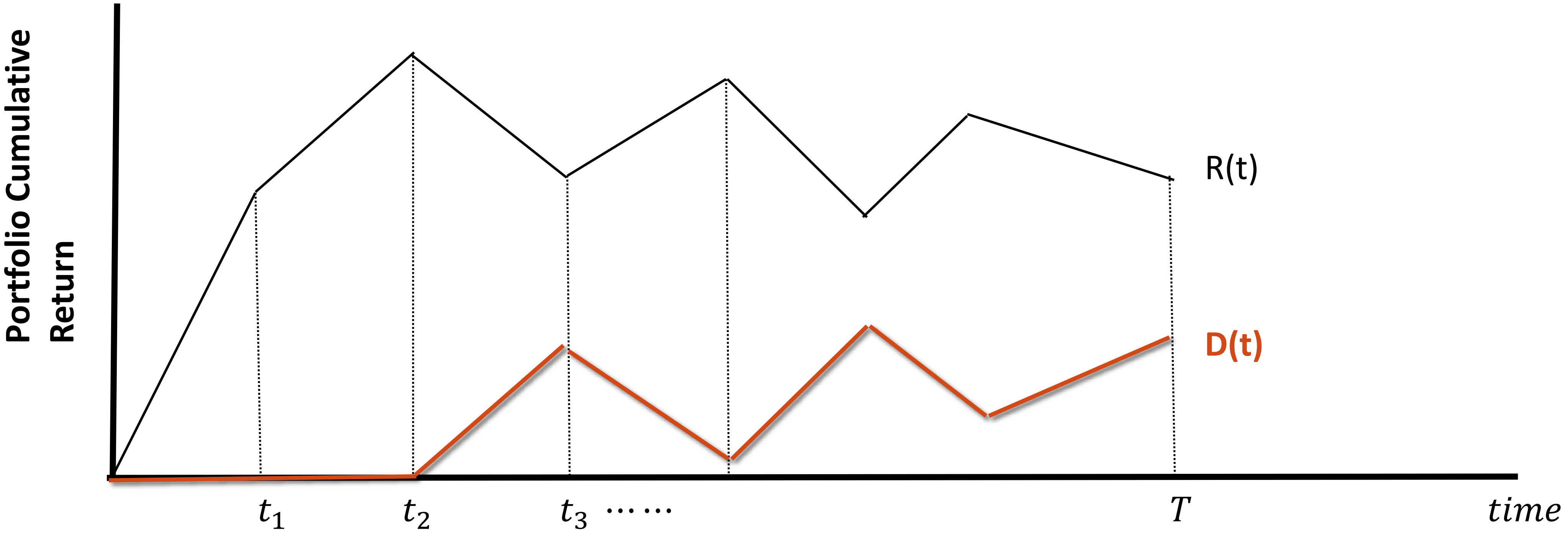
- Data-Driven Portfolio Optimization with Drawdown Constraints
- Prescribe Optimal Portfolio utilizing Machine Learning Methods
- Portfolio Performance Assessment

# Portfolio Performance



# Drawdown – Peak to Trough decline

$$D(t) = \max_{0 \le \tau \le t} R(\tau) - R(t)$$



# Portfolio Optimization – with constraint in drawdown

$$\max_{\boldsymbol{\omega}} \quad E[\boldsymbol{\omega}^T \tilde{R}_N]$$

s.t. $\quad \max \; D(\boldsymbol{\omega}, t) \leq C$

$$C_1 \leq \boldsymbol{\omega}^T 1_N \leq C_2$$

$\tilde{\boldsymbol{R}}_{\boldsymbol{N}}$: the annualized cumulative returns of N assets over the time period [0,T]

$\boldsymbol{\omega}$: investment weights of the N assets

# Data-Driven Portfolio Optimization – with constraint in drawdown

$$\max_{\boldsymbol{\omega}} \quad \boldsymbol{\omega}^T \hat{R}_{\boldsymbol{N}}$$

$$\text{s.t.} \qquad \max\ D(\boldsymbol{\omega}, t) \le C$$

$$C_1 \le \boldsymbol{\omega}^T 1_N \le C_2$$

$\hat{\boldsymbol{R}}_{\boldsymbol{N}}$: the estimated average annualized cumulative returns of N assets over the time period [0,T]

$\boldsymbol{\omega}$: investment weights of the N assets

# Big Data

- An explosion in the availability and accessibility in data
- For example, people frequently browse the internet: shopping, streaming, and searching for key words
- Online activities generate footprints - source of data
- A variety of data sources may be related to stock returns
- Optimal portfolio weights should take into account of these auxiliary variables

# Portfolio Optimization with Auxiliary Variables

$$\max_{\boldsymbol{\omega}(\boldsymbol{x})} \quad \boldsymbol{\omega}^T(x)E[\tilde{R}_N \,|\boldsymbol{x}]$$

$$\text{s.t.} \qquad \max \; D(\boldsymbol{\omega}(x), t) \leq C$$

$$C_1 \leq \boldsymbol{\omega}(\boldsymbol{x})^T 1_N \leq C_2$$

$\boldsymbol{x} \in \boldsymbol{R^d}$ are auxiliary variables with d-dimensions

# Data-Driven Portfolio Optimization with Auxiliary Variables

$$\max_{\boldsymbol{\omega}(\boldsymbol{x})} \quad \boldsymbol{\omega}^T(x)\hat{R}_N(\boldsymbol{x})$$

$$\text{s.t.} \qquad \max\ D(\boldsymbol{\omega}(x), t) \le C$$

$$C_1 \le \boldsymbol{\omega}(\boldsymbol{x})^T 1_N \le C_2$$

$\hat{R}_N(x)$: the estimated conditional mean of annualized cumulative returns of N assets

# Searching Optimal Portfolio Weights – Linear Programming Problem

- The portfolio optimization can be represented as a linear programming problem
- Compute average accumulative returns to obtain $\widehat{R}_N$
- The derived optimal portfolio weight is data-driven, depending on the estimated $\widehat{R}_N$

# Machine Learning

Apply machine learning methods to estimate the conditional means of returns $\widehat{\boldsymbol{R}}_N(\boldsymbol{x})$ and derive the optimal portfolio weight $\boldsymbol{\omega}(\boldsymbol{x})$:

1. Use machine learning methods to classify asset returns into several groups based on the values of auxiliary variables
2. For each group, compute $\widehat{\boldsymbol{R}}_N(\boldsymbol{x})$
3. Solve the linear programing problem to find the optimal portfolio weight $\boldsymbol{\omega}(\boldsymbol{x})$

## Prescribe Investment Weights

## Train data

$$\left(\hat{R}_N^{(1)};\, x_1^{(1)}, x_2^{(1)} \ldots, x_d^{(1)}\right)$$
$$\left(\hat{R}_N^{(2)};\, x_1^{(2)}, x_2^{(2)} \ldots, x_d^{(2)}\right)$$
$$\left(\hat{R}_N^{(3)};\, x_1^{(3)}, x_2^{(3)} \ldots, x_d^{(3)}\right)$$
$$\vdots$$
$$\left(\hat{R}_N^{(T)};\, x_1^{(T)}, x_2^{(T)} \ldots, x_d^{(T)}\right)$$

**Machine Learning**

$$\left(\hat{R}_N^{(1)};\, x_1^{(1)}, x_2^{(1)} \ldots, x_d^{(1)}\right)$$
$$\left(\hat{R}_N^{(3)};\, x_1^{(3)}, x_2^{(3)} \ldots, x_d^{(3)}\right)$$
$$\left(\hat{R}_N^{(4)};\, x_1^{(4)}, x_2^{(4)} \ldots, x_d^{(4)}\right)$$

$$\Rightarrow \quad \boldsymbol{\omega}\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(3)}, \boldsymbol{x}^{(4)}\right)$$

$$\left(\hat{R}_N^{(5)};\, x_1^{(5)}, x_2^{(5)} \ldots, x_d^{(5)}\right)$$
$$\left(\hat{R}_N^{(10)};\, x_1^{(10)}, x_2^{(10)} \ldots, x_d^{(10)}\right)$$

$$\Rightarrow \quad \boldsymbol{\omega}\left(\boldsymbol{x}^{(5)}, \boldsymbol{x}^{(10)}\right)$$

$$\left(\hat{R}_N^{(2)};\, x_1^{(2)}, x_2^{(2)} \ldots, x_d^{(2)}\right)$$
$$\left(\hat{R}_N^{(6)};\, x_1^{(6)}, x_2^{(6)} \ldots, x_d^{(6)}\right)$$
$$\left(\hat{R}_N^{(20)};\, x_1^{(20)}, x_2^{(20)} \ldots, x_d^{(20)}\right)$$
$$\left(\hat{R}_N^{(T)};\, x_1^{(T)}, x_2^{(T)} \ldots, x_d^{(T)}\right)$$

$$\Rightarrow \quad \boldsymbol{\omega}\left(\boldsymbol{x}^{(2)}, \boldsymbol{x}^{(6)}, \boldsymbol{x}^{(20)}, \boldsymbol{x}^{(T)}\right)$$

$\hat{R}_N^{(i)}$: cumulative returns observed at time *i*

# KNN - K Nearest Neighborhood

$$\hat{\boldsymbol{\omega}}_{KNN}(\boldsymbol{x_0}) \in \max_{\boldsymbol{x}^{(i)} \in \aleph_{\boldsymbol{k}}(\boldsymbol{x_0})} \boldsymbol{\omega}^T \widehat{\boldsymbol{R}}_N \left(\boldsymbol{x}^{(i)}\right)$$

s.t. $\quad \max\ D(\boldsymbol{\omega}, t) \le C$

$$C_1 \le \boldsymbol{\omega}^T 1_N \le C_2$$

where $\aleph_K(\boldsymbol{x_0}) = \left\{i = 1, \dots, T : \sum_{j=1}^{T} I\left[\left\|\boldsymbol{x_0} - \boldsymbol{x}^{(i)}\right\| \ge \left\|\boldsymbol{x_0} - \boldsymbol{x}^{(j)}\right\|\right] \le k\right\}$ is the neighborhood of k data points that are closest to the point $\boldsymbol{x_0}$

# Ctree - Conditional Inference Tree (Hothorn et al (2006)

$$\widehat{\omega}_{Ctree}(\boldsymbol{x}) \in \max_{\boldsymbol{i}:\boldsymbol{R}(\boldsymbol{x}^{(\boldsymbol{i})}) \in \boldsymbol{R}(\boldsymbol{x})} \boldsymbol{\omega}^T \widehat{\boldsymbol{R}}_N\,(\boldsymbol{x}^{(\boldsymbol{i})})$$

$$\text{s.t.} \qquad \max\ D(\boldsymbol{\omega}, t) \leq C$$

$$C_1 \leq \boldsymbol{\omega}^T 1_N \leq C_2$$

where R(x) is the splitting rule implied by a regression tree trained based on the training data

- Ctree applies the inference test to determine if a possible split is significant

# Random Forest

$$\hat{\boldsymbol{\omega}}_{RF}^{k}(\boldsymbol{x}) \in \max_{\boldsymbol{i}:\boldsymbol{R}^{k}(\boldsymbol{x}^{(i)}) \in \boldsymbol{R}^{k}(\boldsymbol{x})} \boldsymbol{\omega}^{T} \hat{\boldsymbol{R}}_{N} (\boldsymbol{x}^{(i)})$$

s.t. $\max \; D(\boldsymbol{\omega}, t) \leq C$

$$C_1 \leq \boldsymbol{\omega}^{T} 1_{N} \leq C_2$$

$$\boxed{\hat{\boldsymbol{\omega}}_{RF}(\boldsymbol{x}) = \frac{\mathbf{1}}{\boldsymbol{m}} \sum_{\boldsymbol{k}=\mathbf{1}}^{\boldsymbol{m}} \hat{\boldsymbol{\omega}}_{RF}^{k}(\boldsymbol{x})}$$

# Simulation

- Let $X_1, X_2, X_3$ be the market factors that are related to the returns of the underlying assets in the portfolio.
- Assume $X(t) = \{X_1(t), X_2(t), X_3(t)\}$ follows a multivariate ARMA(2,2) process
- Generate 12 time series of returns based on the three market factors

$$R_N(t) = A^T\left(X(t) + \frac{\delta}{4}\right) + (B^T(X(t))\eta$$

where $\delta$ and $\eta$ are independent Gaussian noises

# Portfolio Performance – Validation

For each realization of the simulated returns

- split the series into in-sample (training) and out-of-sample (validation) data.
- Trained the in-sample data based on the machine learning algorithm and prescribe optimal portfolio weights
- Apply the prescribed weights to the out-of-sample (validation) returns and compute the portfolio returns

# Performance Metrics

1. Average Cumulative Return

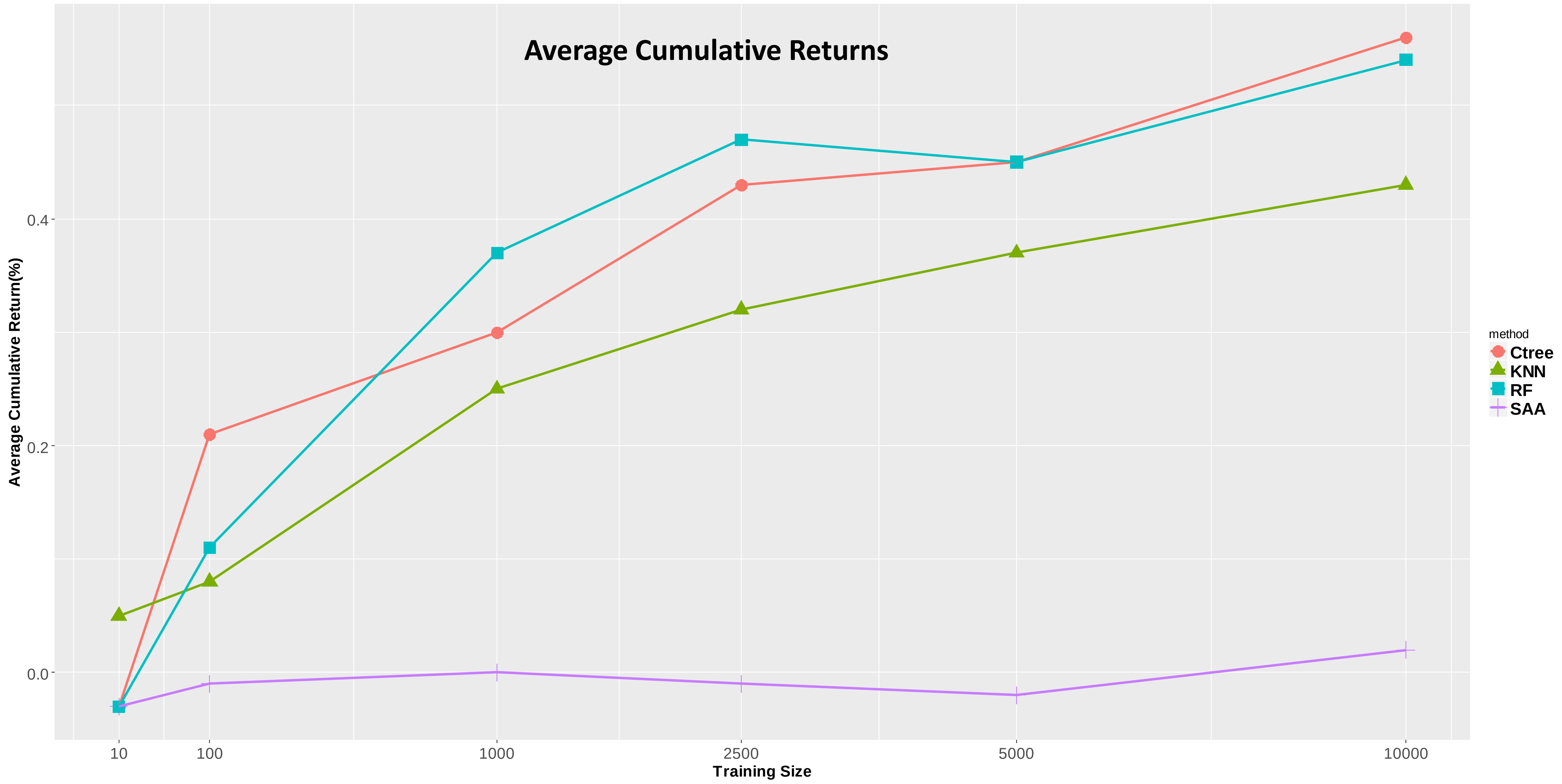
$$\frac{1}{n}\sum_{t=1}^{n} R_t^v$$

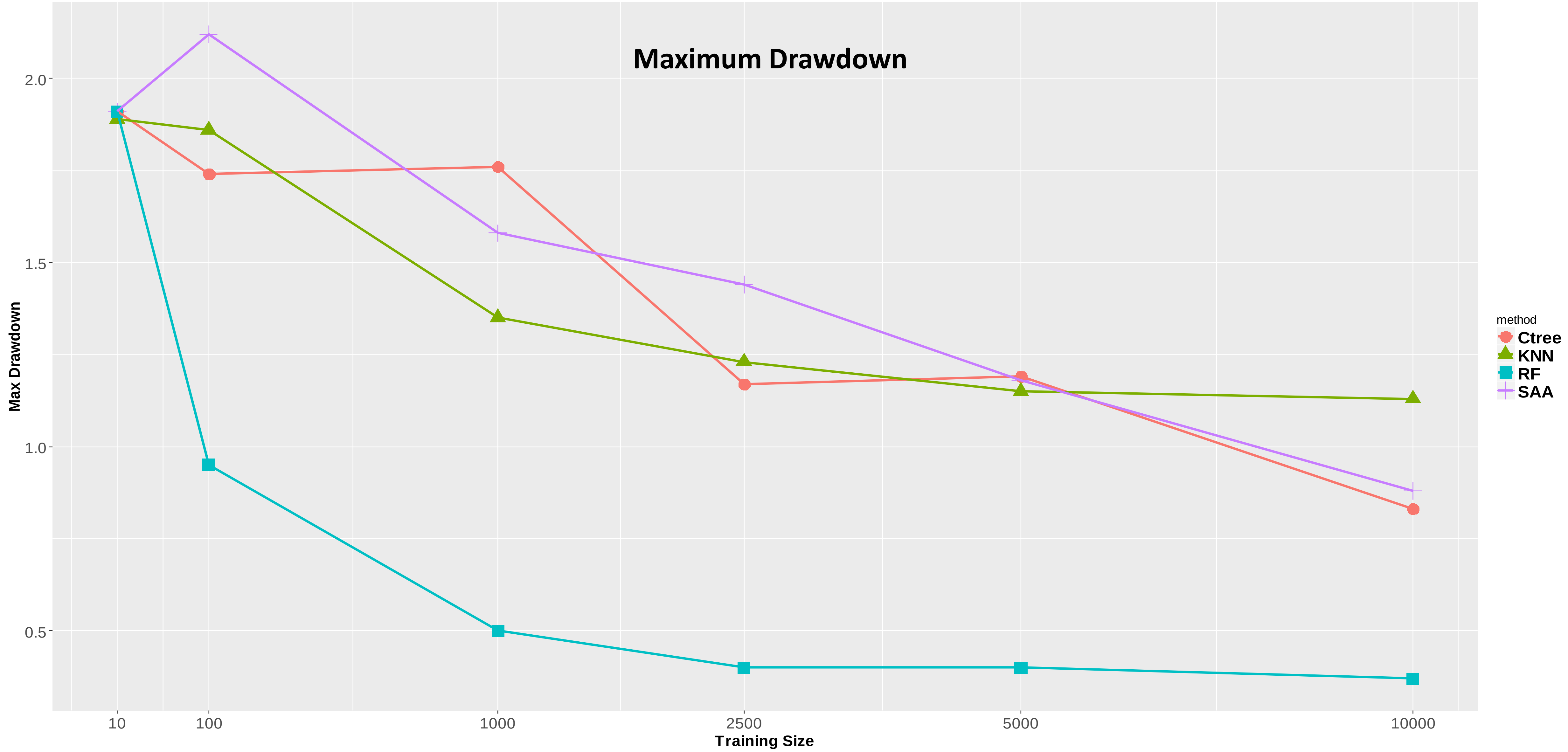
2. Maximum Drawdown

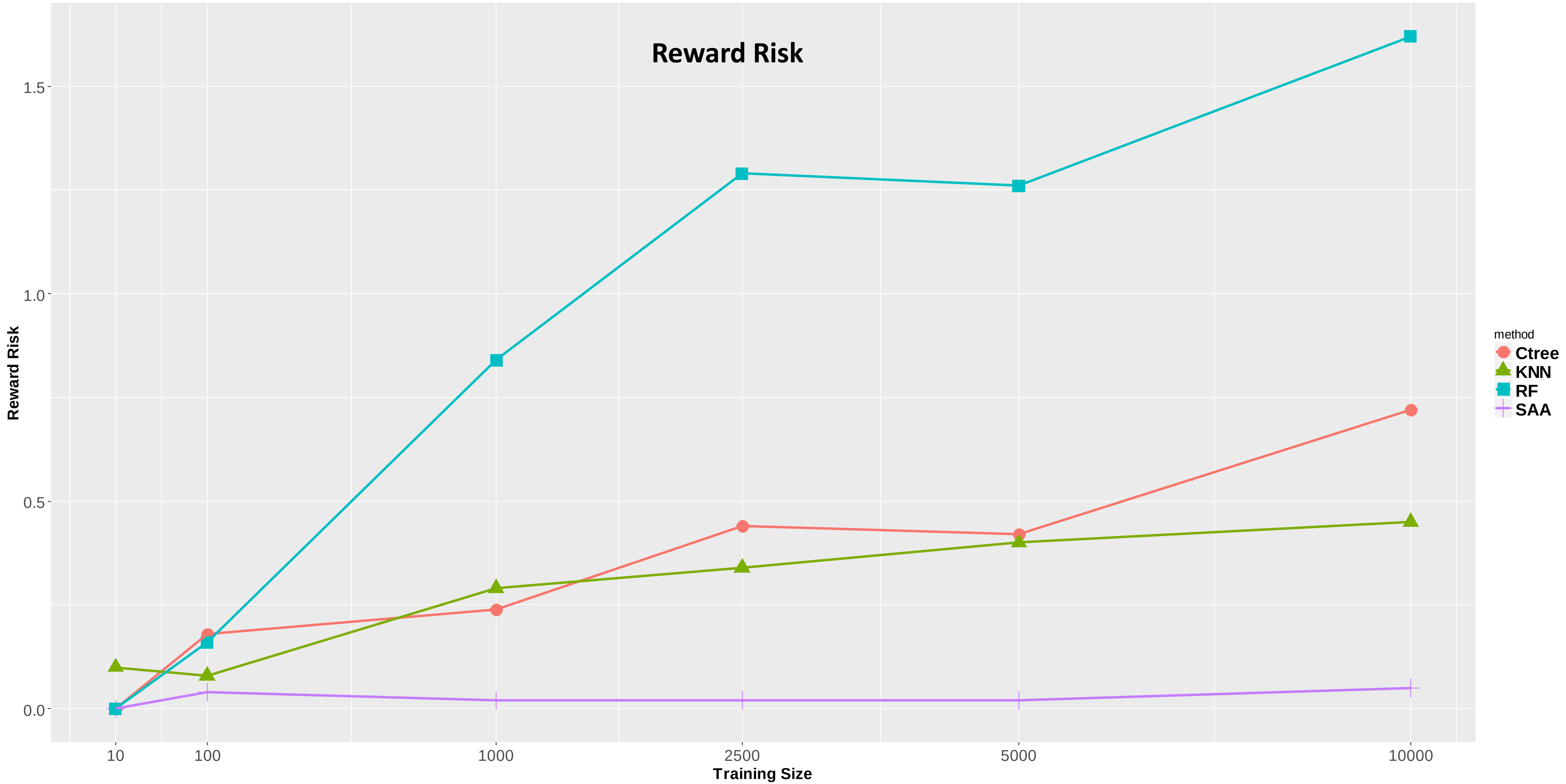
$$\max_{0\le t\le n}\left\{\max_{0\le\tau\le t} R(\tau) - R(t)\right\}$$

3. Reward Risk

$$\frac{\frac{1}{n}\sum_{t=1}^{n} R_t^v}{\max_{0\le t\le n}\left\{\max_{0\le\tau\le t} R(\tau) - R(t)\right\}}$$


Average Cumulative Returns
Average Cumulative Return(%)
0.4
0.2
0.0
10
100
1000
2500
5000
10000
Training Size
method
Ctree
KNN
RF
SAA


Maximum Drawdown
Max Drawdown
2.0
1.5
1.0
0.5
10
100
1000
2500
5000
10000
Training Size
method
Ctree
KNN
RF
SAA


Reward Risk
Reward Risk
1.5
1.0
0.5
0.0
10
100
1000
2500
5000
10000
Training Size
method
Ctree
KNN
RF
SAA

# Summary

- Machine Learning method improves the out-of sample performance of the data-driven optimal portfolio
- The performance improves as the size of training data increases
- Tree-based approaches such as random forest and Ctree outperform the SAA method which does not incorporate the inputs from auxiliary variables.

# Future Work

- how the tuning parameters affect optimal portfolio weights and portfolio performance?

- how the correlations of the auxiliary variables affect optimal portfolio weights and portfolio performance?

# Appendix

# Searching Optimal Portfolio Weights – Linear Programming Problem

$$\max_{\boldsymbol{\omega}} \quad \boldsymbol{\omega}^T \widehat{\boldsymbol{R}}_{\boldsymbol{T\times N}}$$

$$\text{s.t.} \quad u_k - \ \boldsymbol{\omega}^T \widehat{R}_{k\times N} \le C \,, 1 \le k \le N$$

$$u_k \ge \ \boldsymbol{\omega}^T \widehat{R}_{k\times N} \ , 1 \le k \le N$$

$$u_k \ge u_{k-1}, 1 \le k \le N$$

$$u_0 = 0$$

$$C_1 \le \boldsymbol{\omega}^T 1_N \le C_2$$

where $u_k$ are auxiliary variables, $1 \le k \le N$