



Decision-focused Learning in Partial Index Tracking



Hyunglip Bae

Ph.D, KAIST

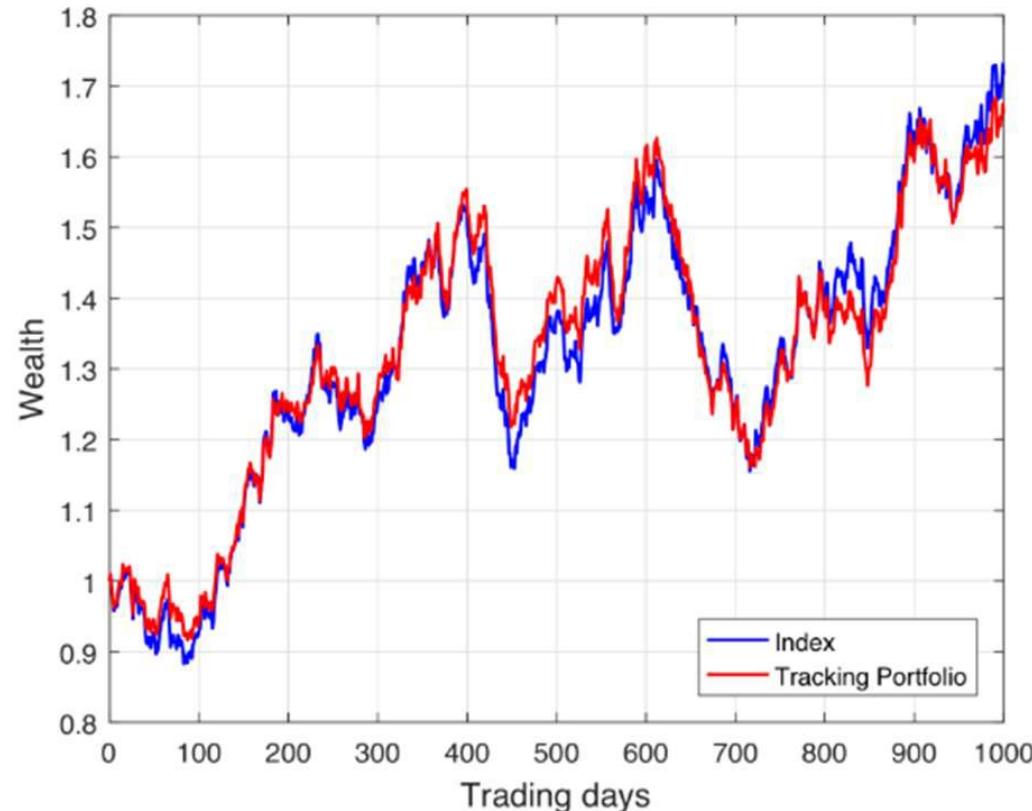
qogudflq@kaist.ac.kr

Financial Engineering Lab
Department of Industrial Engineering



Introduction

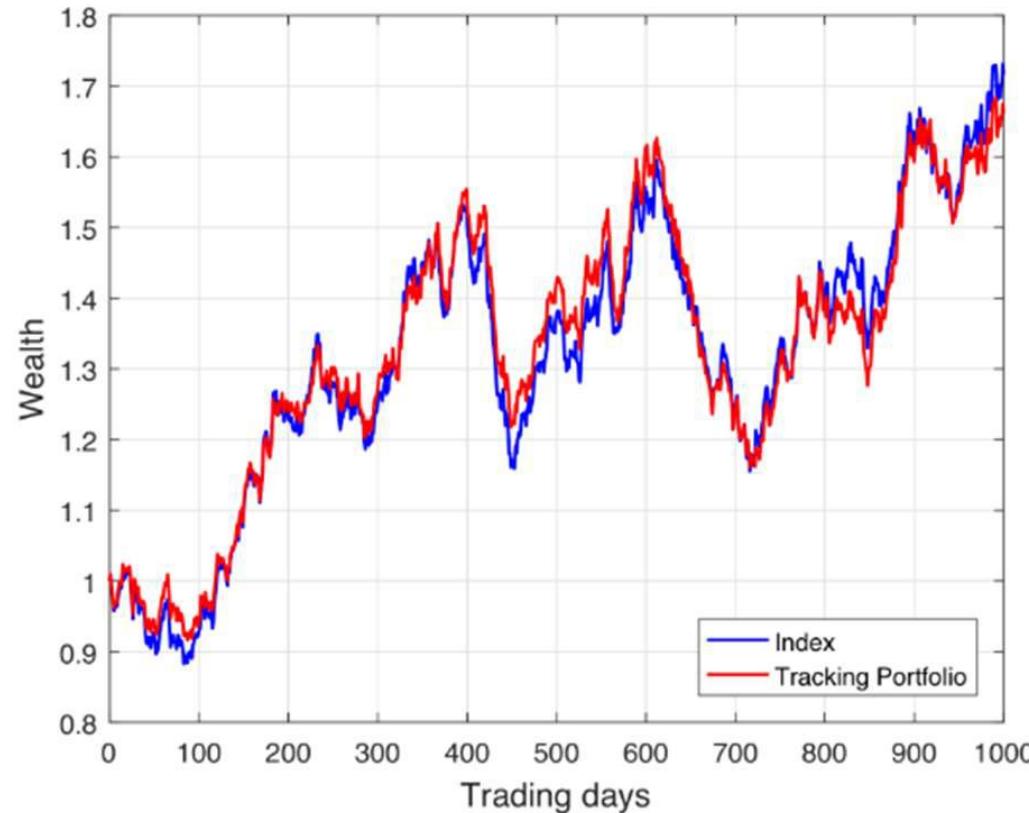
- Partial Index Tracking



- **Index:** measure the performance of finance market

Introduction

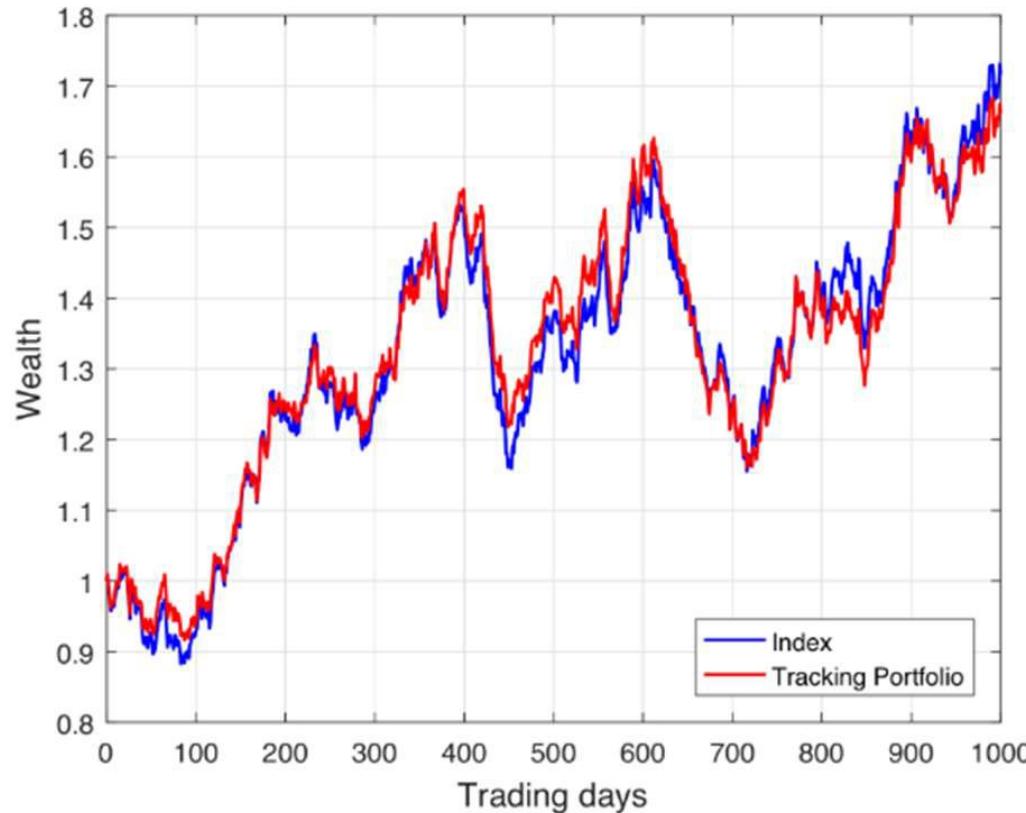
- Partial Index Tracking



- **Index:** measure the performance of finance market
- **Tracking:** aim to replicate performance of index

Introduction

- **Partial Index Tracking**



- **Partial:** use only subset of assets in index
- **Index:** measure the performance of finance market
- **Tracking:** aim to replicate performance of index

Basic Formulation

- **Variables & Functions**

Name	Definition
$n \in \mathbb{N}$	The number of assets
$\hat{r} \in \mathbb{R}^n$	The return of assets (Uncertain)
$w \in \mathbb{R}^n$	The weight of portfolio
$I \in \mathbb{R}$	Target Index
1	One vector

Name	Definition
Card (\cdot)	The number of positive component
$\ \cdot\ _2$	Euclidean 2-norm
$\ \cdot\ _1$	Euclidean 1-norm

Basic Formulation

- **Variables & Functions**

Name	Definition
$n \in \mathbb{N}$	The number of assets
$\hat{r} \in \mathbb{R}^n$	The return of assets (Uncertain)
$w \in \mathbb{R}^n$	The weight of portfolio
$I \in \mathbb{R}$	Target Index
1	One vector

Name	Definition
Card (\cdot)	The number of positive component
$\ \cdot\ _2$	Euclidean 2-norm
$\ \cdot\ _1$	Euclidean 1-norm

- **Formulation**

$$\text{Minimize}_w \quad \|\hat{r}^\top w - I\|^2 \quad \Rightarrow \text{Minimize tracking error}$$

Basic Formulation

- **Variables & Functions**

Name	Definition
$n \in \mathbb{N}$	The number of assets
$\hat{r} \in \mathbb{R}^n$	The return of assets (Uncertain)
$w \in \mathbb{R}^n$	The weight of portfolio
$I \in \mathbb{R}$	Target Index
1	One vector

Name	Definition
Card (\cdot)	The number of positive component
$\ \cdot\ _2$	Euclidean 2-norm
$\ \cdot\ _1$	Euclidean 1-norm

- **Formulation**

$$\begin{aligned} \text{Minimize}_w \quad & \|\hat{r}^\top w - I\|^2 && \Rightarrow \text{Minimize tracking error} \\ \text{s.t.} \quad & \mathbf{1}^\top w = 1 && \Rightarrow \text{Sum of portfolio weight} = 1 \end{aligned}$$

Basic Formulation

- **Variables & Functions**

Name	Definition
$n \in \mathbb{N}$	The number of assets
$\hat{r} \in \mathbb{R}^n$	The return of assets (Uncertain)
$w \in \mathbb{R}^n$	The weight of portfolio
$I \in \mathbb{R}$	Target Index
1	One vector

Name	Definition
Card (\cdot)	The number of positive component
$\ \cdot\ _2$	Euclidean 2-norm
$\ \cdot\ _1$	Euclidean 1-norm

- **Formulation**

$$\begin{array}{lll} \text{Minimize}_w & \|\hat{r}^\top w - I\|^2 & \Rightarrow \text{Minimize tracking error} \\ \text{s.t} & \mathbf{1}^\top w = 1 & \Rightarrow \text{Sum of portfolio weight} = 1 \\ & \text{Card}(w) \leq k & \Rightarrow \text{Cardinality constraint} \end{array}$$

DFL in partial index tracking

- Challenge
 - Cardinality constraint makes the problem discrete and not differentiable

DFL in partial index tracking

- **Challenge**

- Cardinality constraint makes the problem discrete and not differentiable
- In this presentation, we introduce two strategies for this challenge

1) Semi-definite Relaxation Approach

$$\begin{array}{ll}\min_{W \in S_n^+} & \text{Tr}(\hat{R}W) - 2(\mathbf{1}^\top W \hat{R} C) \\ \text{subject to} & \text{Tr}(I_N W) = 1 \\ & \mathbf{1}^\top |W| \mathbf{1} \leq k \text{Tr}(W) \\ & W \in S_n^+\end{array}$$

DFL in partial index tracking

- **Challenge**

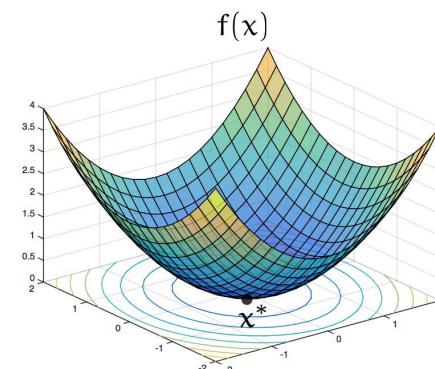
- Cardinality constraint makes the problem discrete and not differentiable
- In this presentation, we introduce two strategies for this challenge

1) Semi-definite Relaxation Approach

$$\begin{array}{ll}\min_{W \in S_n^+} & \text{Tr}(\hat{R}W) - 2(\mathbf{1}^\top W \hat{R}C) \\ \text{subject to} & \text{Tr}(I_N W) = 1 \\ & \mathbf{1}^\top |W| \mathbf{1} \leq k \text{Tr}(W) \\ & W \in S_n^+\end{array}$$

2) Surrogate Loss Approach

$$\mathcal{L}_{DFL} \approx$$

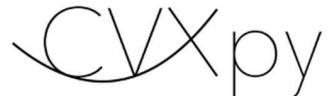


Semi-definite Relaxation Approach

- With semi-definite relaxation, we can transform the original problem to be convex
- Then, the CvxpyLayer can be used to obtain the gradients required for DFL

$$\begin{array}{ll} \min_{w \in \mathbb{R}^n} & \|\hat{r}^\top w - I\|^2 \\ \text{s.t.} & \mathbf{1}^\top w = 1 \\ & \text{Card}(w) \leq k \end{array}$$

Semi-definite
Relaxation


 $x^*(\theta) = \underset{x}{\operatorname{argmin}} f(x; \theta)$
subject to $g(x; \theta) \leq 0$
 $h(x; \theta) = 0$



 PyTorch
 TensorFlow

Semi-definite reformulation

- We first introduce semi-definite matrix variable $W = ww^\top \in \mathbb{R}^{n \times n}$ and $\hat{R} = \hat{r}\hat{r}^\top \in \mathbb{R}^{n \times n}$

Semi-definite reformulation

- We first introduce semi-definite matrix variable $W = ww^\top \in \mathbb{R}^{n \times n}$ and $\hat{R} = \hat{r}\hat{r}^\top \in \mathbb{R}^{n \times n}$
- The objective function can be reformulated as (**Tr**: Trace of matrix)

$$\|\hat{r}^\top w - I\|^2 = (\hat{r}^\top w)^2 - 2I(\hat{r}^\top w) + I^2 = \text{Tr}(\hat{R}W) - 2I(\mathbf{1}^\top W \hat{r}) + I^2$$

Semi-definite reformulation

- We first introduce semi-definite matrix variable $W = ww^\top \in \mathbb{R}^{n \times n}$ and $\hat{R} = \hat{r}\hat{r}^\top \in \mathbb{R}^{n \times n}$
- The objective function can be reformulated as (**Tr**: Trace of matrix)

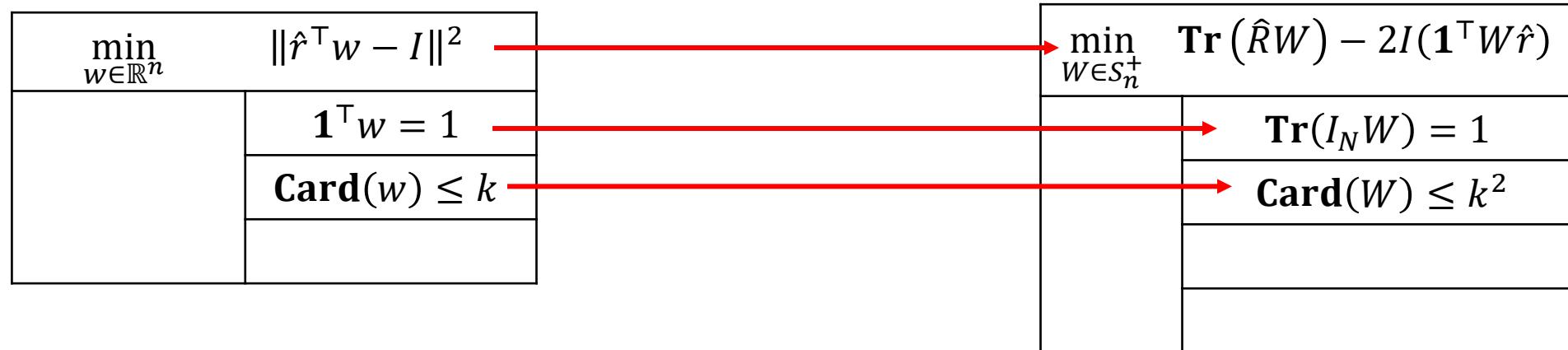
$$\|\hat{r}^\top w - I\|^2 = (\hat{r}^\top w)^2 - 2I(\hat{r}^\top w) + I^2 = \text{Tr}(\hat{R}W) - 2I(\mathbf{1}^\top W \hat{r}) + I^2$$

- Since I^2 is constant,



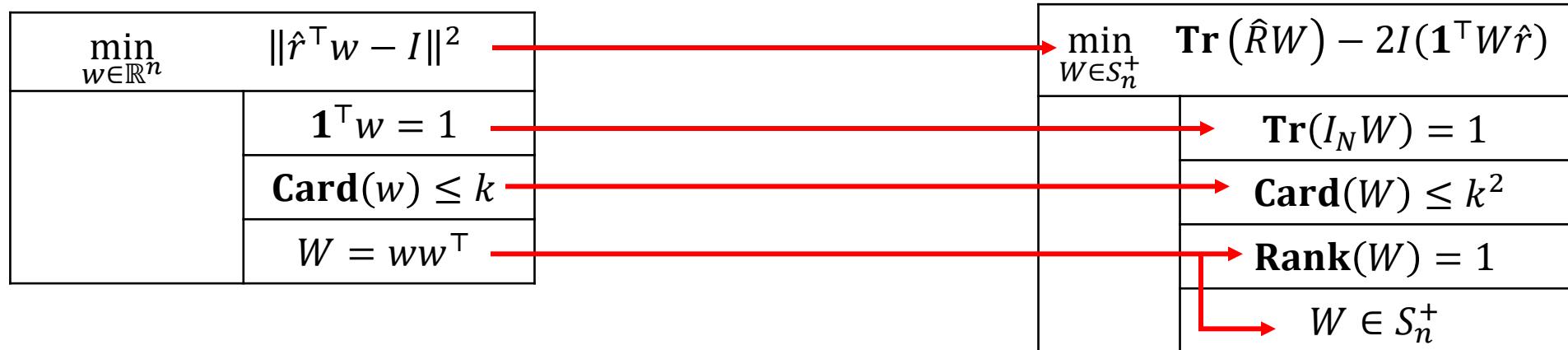
Semi-definite reformulation

- Furthermore, the weight sum constraint and cardinality constraint are reformulated as



Semi-definite reformulation

- Finally, $W = ww^\top$ is equivalent ($\text{Rank}(W) = 1$ & $W \in S_n^+$ (Positive semi-definite))



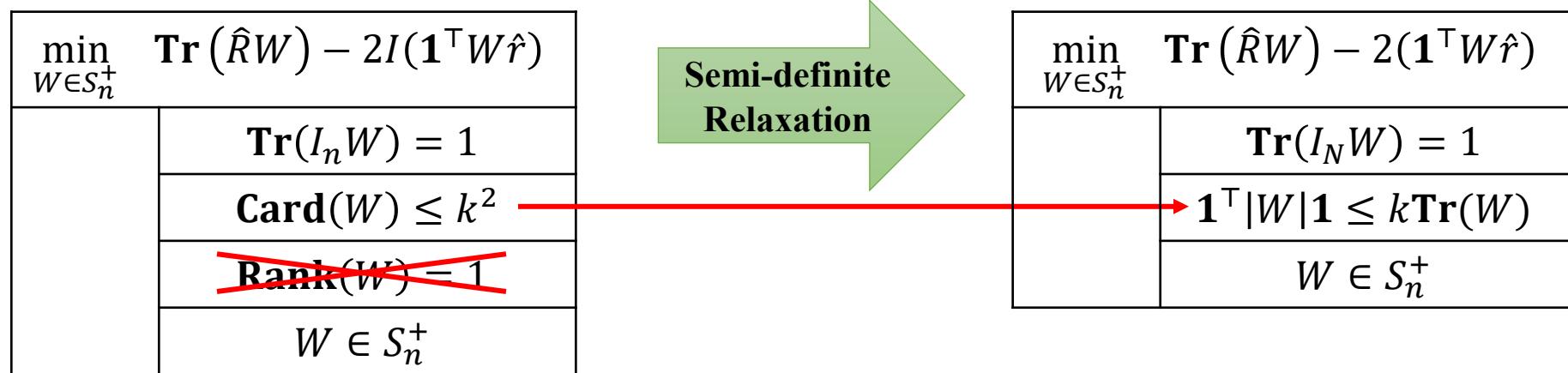
Semi-definite relaxation

- Next, Let's relax the reformulated problem. We first drop the Rank constraint.

$$\begin{array}{ll}\min_{W \in S_n^+} & \text{Tr}(\hat{R}W) - 2I(\mathbf{1}^\top W \hat{r}) \\ \hline & \text{Tr}(I_n W) = 1 \\ & \text{Card}(W) \leq k^2 \\ & \cancel{\text{Rank}(W) = 1} \\ \hline & W \in S_n^+\end{array}$$

Semi-definite relaxation (cont.)

- Next, by relaxing $\text{Card}(W) \leq k^2$ to $\|w\|_1^2 \leq k\|w\|_2^2$ and transforming to $\mathbf{1}^\top |W| \mathbf{1} \leq k\text{Tr}(W)$,



DPP-compliant

- When using CvxpyLayer, the formulation must satisfy DPP-compliant
 1. The optimization problem must be convex in variable $W \rightarrow \text{Already Satisfied}$

DPP-compliant

- When using CvxpyLayer, the formulation must satisfy DPP-compliant
 1. The optimization problem must be convex in variable $W \rightarrow \text{Already Satisfied}$
 2. The optimization problem must be linear in parameter $\hat{R} \rightarrow \text{Use } I = \hat{r}^\top C \text{ where } C \text{ is market cap weight}$

DPP-compliant

- When using CvxpyLayer, the formulation must satisfy DPP-compliant
 - The optimization problem must be convex in variable $W \rightarrow \text{Already Satisfied}$
 - The optimization problem must be linear in parameter $\hat{R} \rightarrow \text{Use } I = \hat{r}^\top C \text{ where } C \text{ is market cap weight}$

$$\begin{array}{ll}\min_{W \in S_n^+} & \mathbf{Tr}(\hat{R}W) - 2I(\mathbf{1}^\top W \hat{r}) \\ \hline & \mathbf{Tr}(I_N W) = 1 \\ & \mathbf{1}^\top |W| \mathbf{1} \leq k \mathbf{Tr}(W) \\ \hline & W \in S_n^+\end{array}$$

$$\begin{array}{ll}\min_{W \in S_n^+} & \mathbf{Tr}(\hat{R}W) - 2(\mathbf{1}^\top W \hat{R}C) \\ \hline & \mathbf{Tr}(I_N W) = 1 \\ & \mathbf{1}^\top |W| \mathbf{1} \leq k \mathbf{Tr}(W) \\ \hline & W \in S_n^+\end{array}$$

DFL loss for Partial Index Tracking

- As seen in previous session, the DFL loss is expressed as

$$\begin{aligned}\mathcal{L}(\widehat{\mathbf{R}}, \mathbf{R}) &:= \textit{Regret}(\mathbf{w}^*(\widehat{\mathbf{R}}), \mathbf{R}) \\ &= \mathbf{f}(\mathbf{w}^*(\widehat{\mathbf{R}}), \mathbf{R}) - \mathbf{f}(\mathbf{w}^*(\mathbf{R}), \mathbf{R})\end{aligned}$$

DFL loss for Partial Index Tracking

- As seen in previous session, the DFL loss is expressed as

$$\begin{aligned}\mathcal{L}(\widehat{\mathbf{R}}, \mathbf{R}) &:= \textit{Regret}(\mathbf{w}^*(\widehat{\mathbf{R}}), \mathbf{R}) \\ &= \underline{f(\mathbf{w}^*(\widehat{\mathbf{R}}), \mathbf{R})} - f(\mathbf{w}^*(\mathbf{R}), \mathbf{R})\end{aligned}$$

Realized decision quality with estimated $\widehat{\mathbf{R}}$

DFL loss for Partial Index Tracking

- As seen in previous session, the DFL loss is expressed as

$$\begin{aligned}\mathcal{L}(\widehat{\mathbf{R}}, \mathbf{R}) &:= \textit{Regret}(\mathbf{w}^*(\widehat{\mathbf{R}}), \mathbf{R}) \\ &= \underline{f(\mathbf{w}^*(\widehat{\mathbf{R}}), \mathbf{R})} - \overline{f(\mathbf{w}^*(\mathbf{R}), \mathbf{R})}\end{aligned}$$

Realized decision quality with estimated $\widehat{\mathbf{R}}$

Realized decision quality with ground truth \mathbf{R}

DFL loss for Partial Index Tracking

- As seen in previous session, the DFL loss is expressed as

$$\begin{aligned}\mathcal{L}(\hat{\mathbf{R}}, \mathbf{R}) &:= \textit{Regret}(\mathbf{w}^*(\hat{\mathbf{R}}), \mathbf{R}) \\ &= \underline{f(\mathbf{w}^*(\hat{\mathbf{R}}), \mathbf{R})} - \overline{f(\mathbf{w}^*(\mathbf{R}), \mathbf{R})}\end{aligned}$$

Realized decision quality with estimated $\hat{\mathbf{R}}$ Realized decision quality with ground truth \mathbf{R}

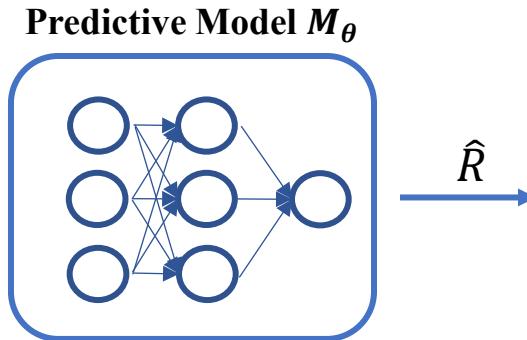


- Thus, in index tracking problem, the DFL loss is

$$\mathcal{L}(\hat{\mathbf{R}}, \mathbf{R}) = [\mathbf{Tr}(R\mathbf{W}^*(\hat{\mathbf{R}})) - 2(\mathbf{1}^\top \mathbf{W}^*(\hat{\mathbf{R}}) \mathbf{R} \mathbf{C})] - [\mathbf{Tr}(R\mathbf{W}^*(\mathbf{R})) - 2(\mathbf{1}^\top \mathbf{W}^*(\mathbf{R}) \mathbf{R} \mathbf{C})]$$

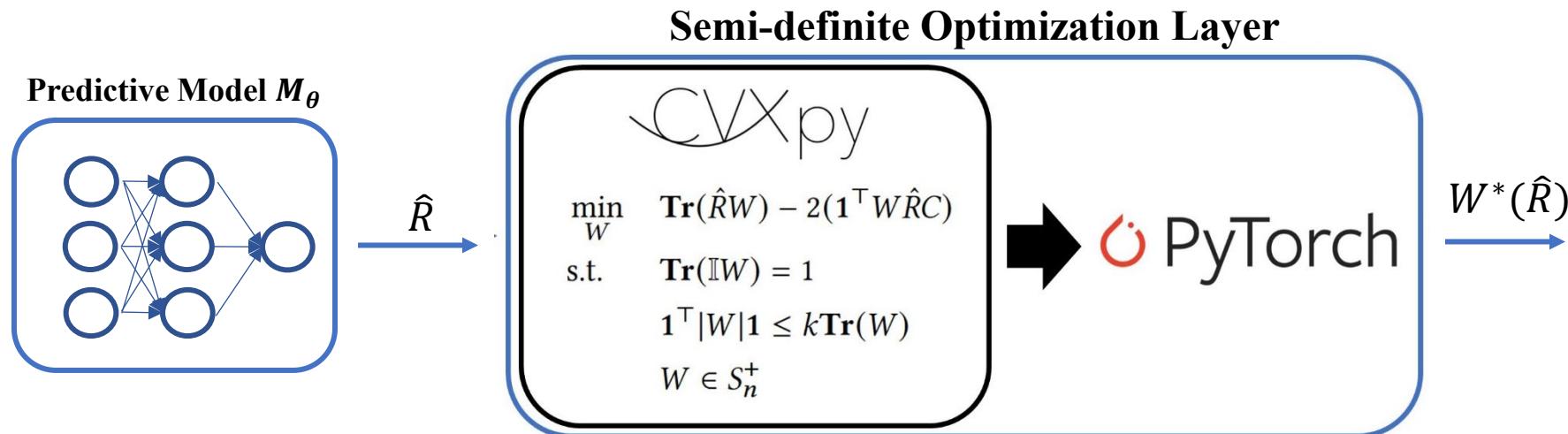
DFL with semi-definite relaxation

- The process of DFL in Partial Index Tracking



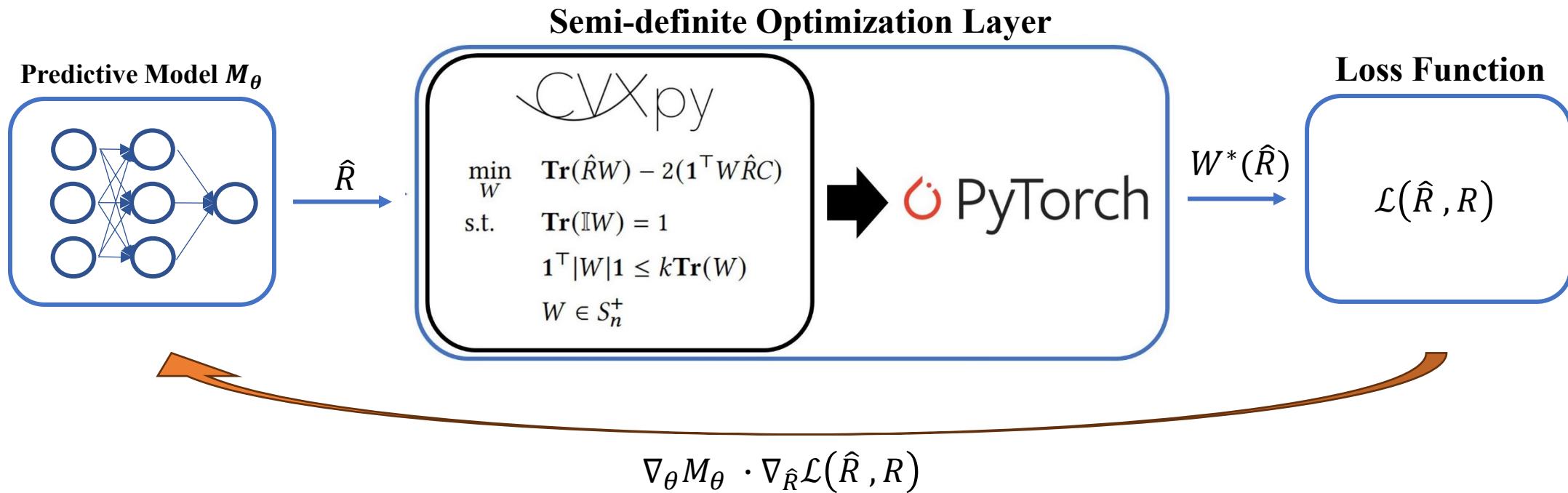
DFL with semi-definite relaxation

- The process of DFL in Partial Index Tracking



DFL with semi-definite relaxation

- The process of DFL in Partial Index Tracking



Surrogate Approach

- The DFL loss function of basic formulation is

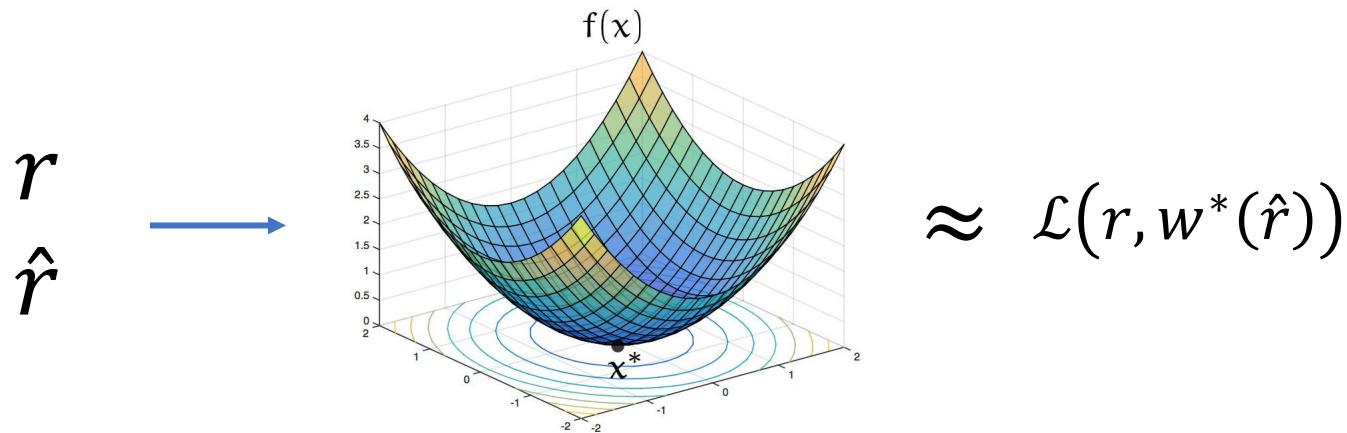
$$\mathcal{L}(\hat{r}, r) = \|r^\top w^*(\hat{r}) - I\|^2 - \|r^\top w^*(r) - I\|^2$$

Surrogate Approach

- The DFL loss function of basic formulation is

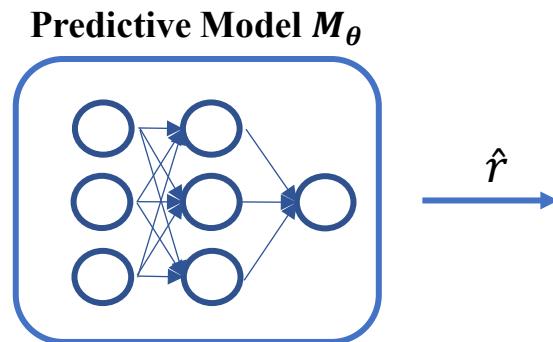
$$\mathcal{L}(\hat{r}, r) = \|r^\top w^*(\hat{r}) - I\|^2 - \|r^\top w^*(r) - I\|^2$$

- Surrogate approach aim to learn $\mathcal{L}(\hat{r}, r)$ with another predictive model



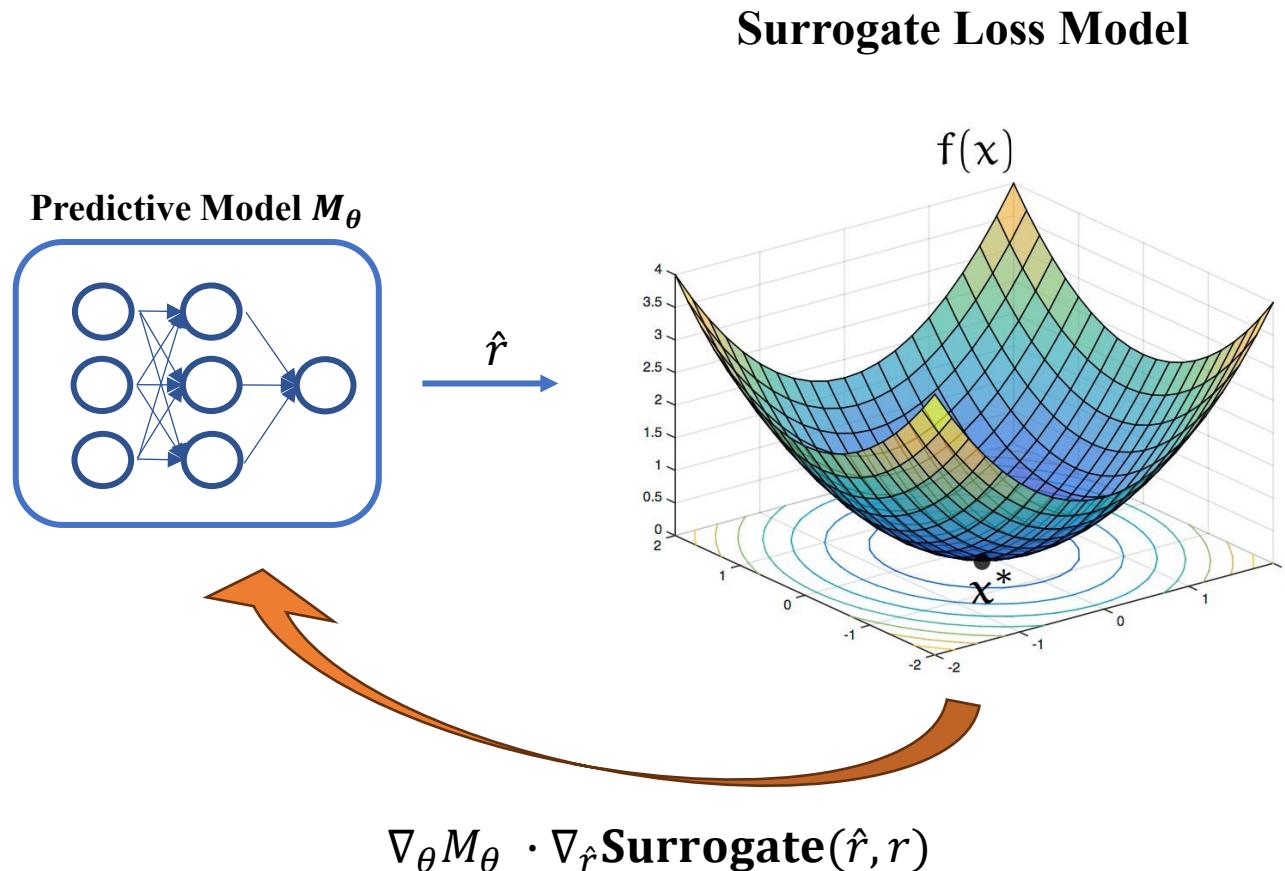
DFL with surrogate approach

- The process of DFL in Partial Index Tracking



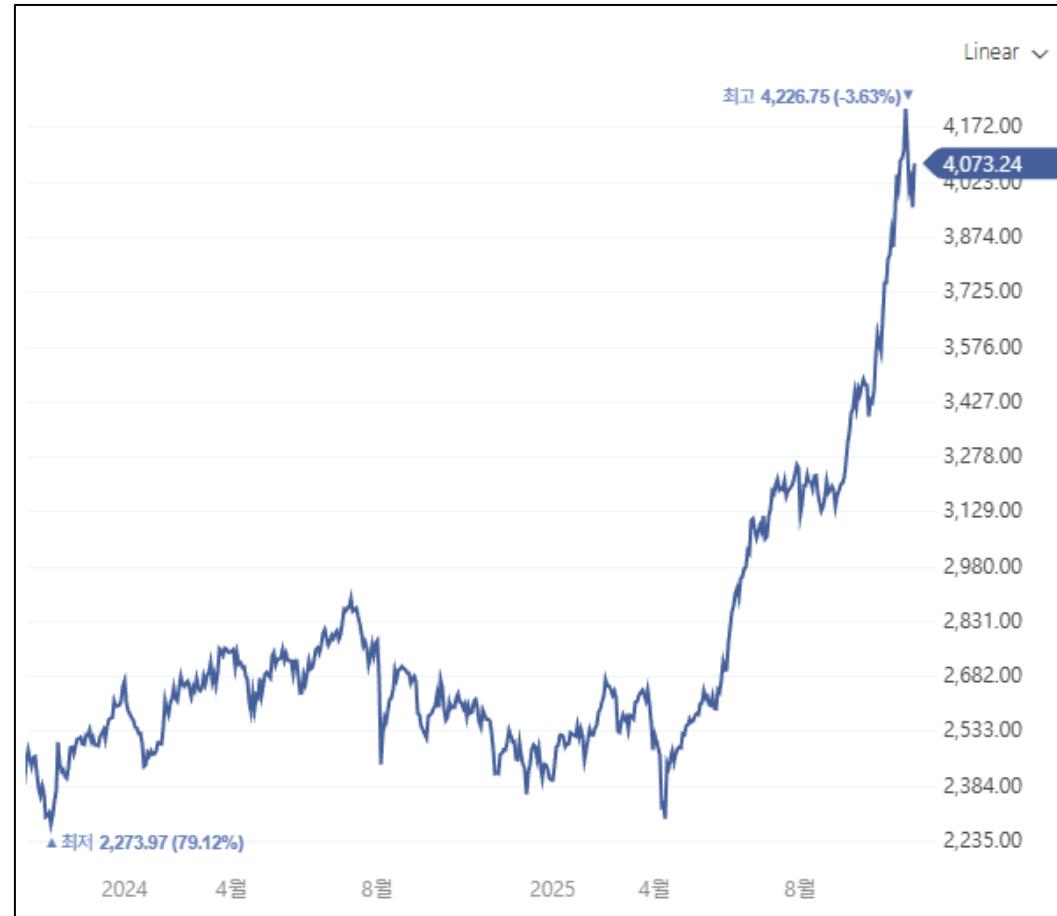
DFL with surrogate approach

- The process of DFL in Partial Index Tracking



Hands-on Exercise

- Partial Index Tracking problem with KOSPI index
 - For simplicity, with top 30 assets



Data

- Closed price and Market cap weight of KOSPI30 (2015/01/02 ~ 2024/12/30)

Data

- Closed price and Market cap weight of KOSPI30 (2015/01/02 ~ 2024/12/30)

Features

Return before rebalancing day

Cumulative return over the past period

Mean and variance of past returns

Mean and variance of past prices

Current return

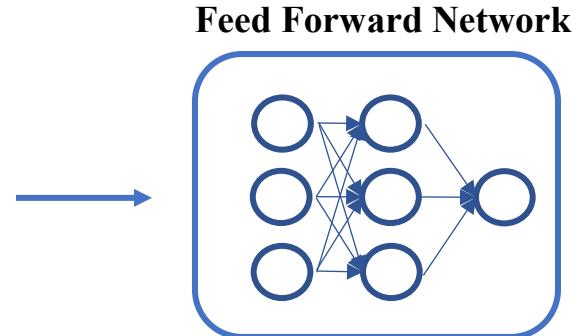
Current price

Data

- Closed price and Market cap weight of KOSPI30 (2015/01/02 ~ 2024/12/30)

Features

- Return before rebalancing day
- Cumulative return over the past period
- Mean and variance of past returns
- Mean and variance of past prices
- Current return
- Current price



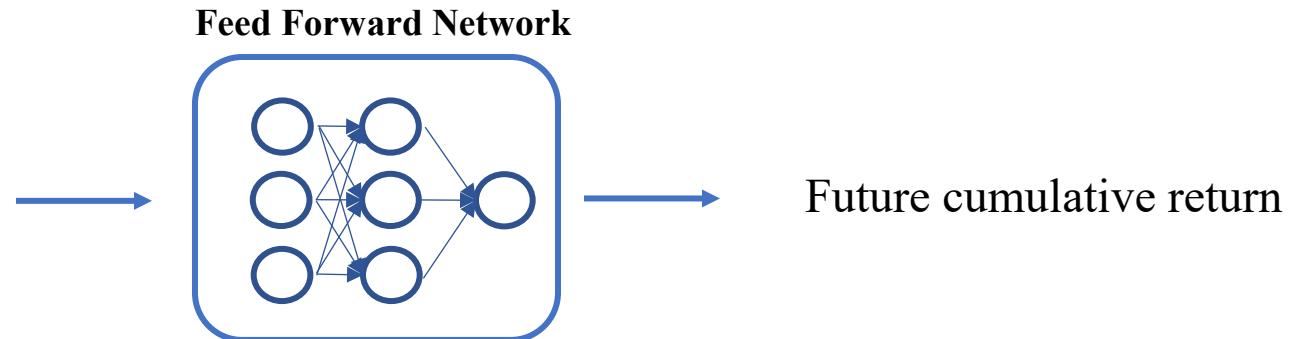
Data

- Closed price and Market cap weight of KOSPI30 (2015/01/02 ~ 2024/12/30)

Features

Return before rebalancing day
Cumulative return over the past period
Mean and variance of past returns
Mean and variance of past prices

Current return
Current price



Hands-on Exercise

- Let's implement partial index tracking directly with Colab!

Notebook & Data Link





Thank you for listening!