Decision-focused Learning in Partial Index Tracking

Hyunglip Bae
Ph.D, KAIST
qogudflq@kaist.ac.kr

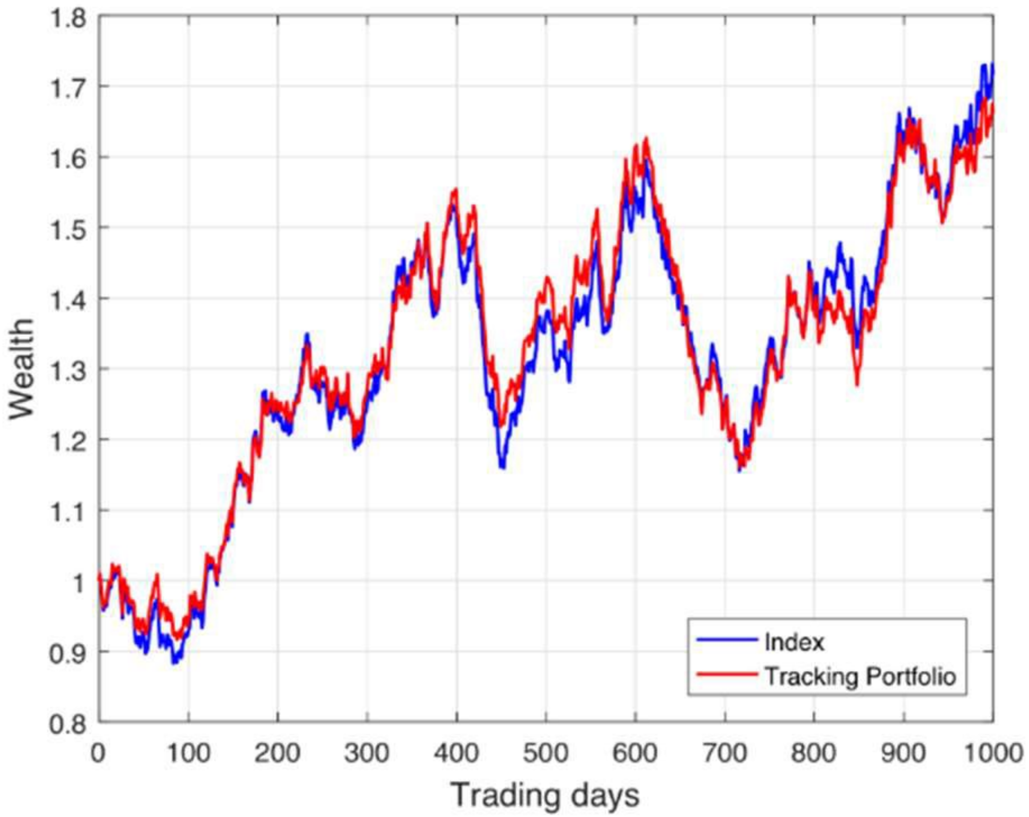
Financial Engineering Lab
Department of Industrial Engineering

ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY
Korea Advanced Institute of Science and Technology

ACM ICAIF 2025

Introduction

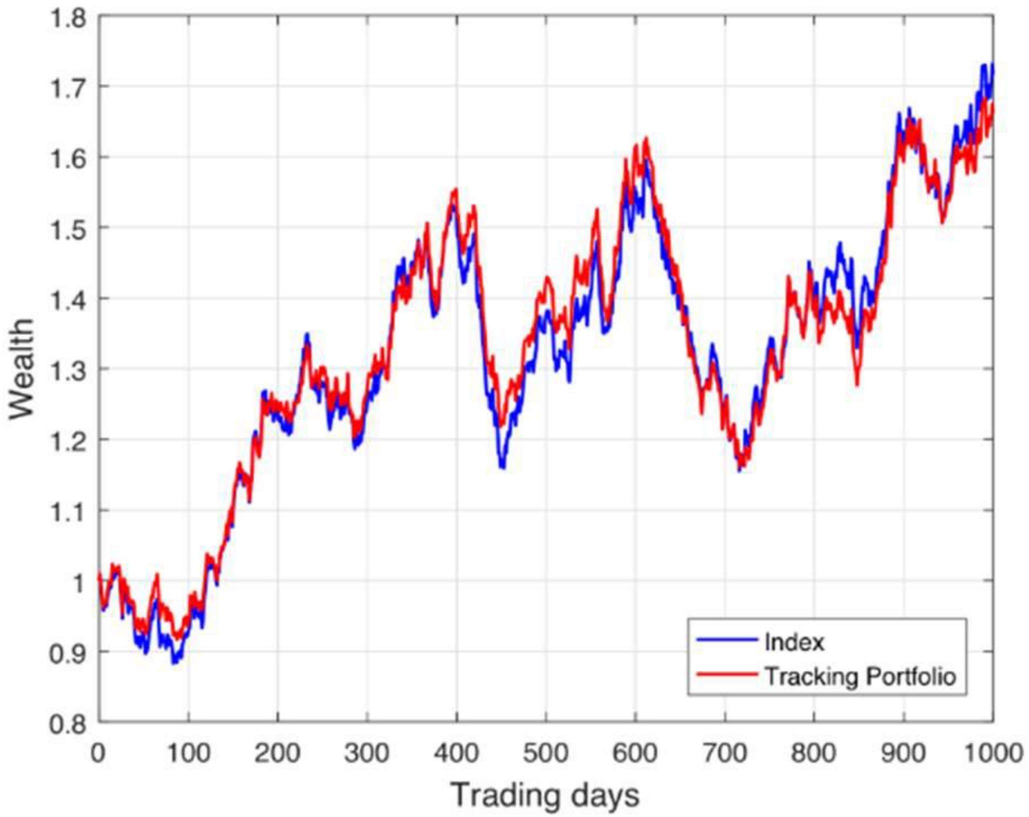
- **Partial Index Tracking**



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

Introduction

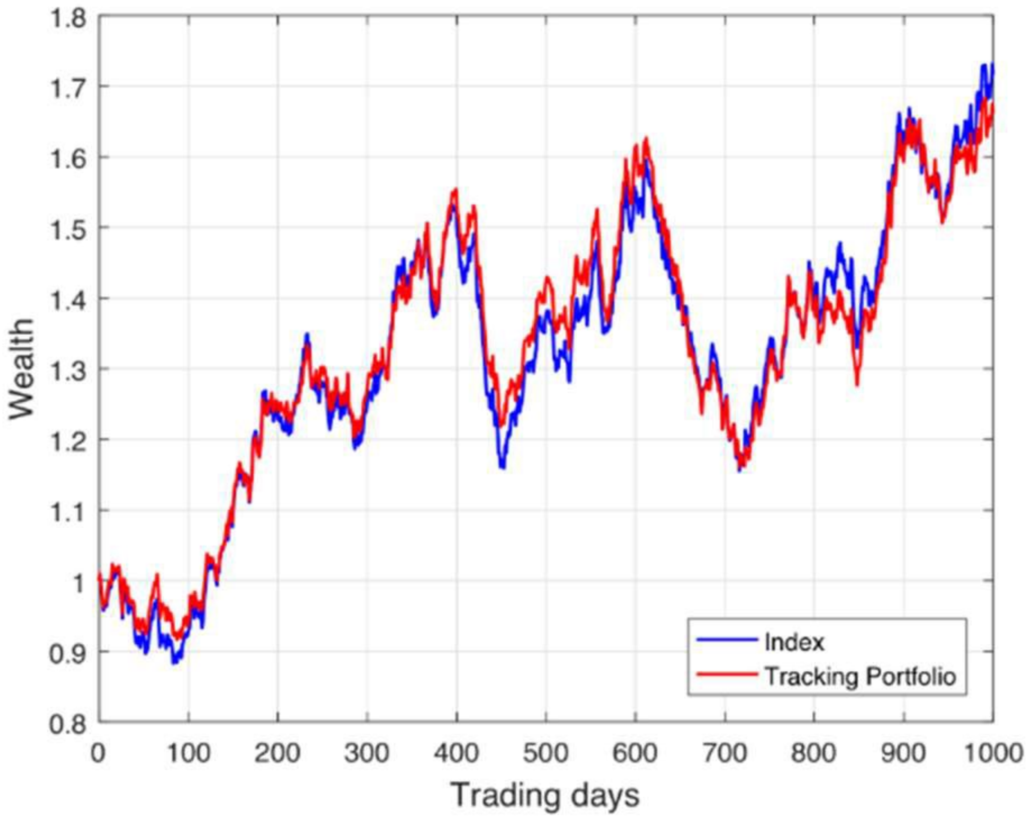
- **Partial Index Tracking**



- **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
$\mathbf{1}$	One vector

Name	Definition
$\mathbf{Card}(\cdot)$	The number of positive component
$\lVert\cdot\rVert_2$	Euclidean 2-norm
$\lVert\cdot\rVert_1$	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
$\mathbf{1}$	One vector

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

- **Formulation**

$$Minimize_w \qquad \|\hat{r}^\top w - I\|^2 \qquad \Longrightarrow \text{Minimize tracking error}$$

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
$\mathbf{1}$	One vector

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

- **Formulation**

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

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
$\mathbf{1}$	One vector

Name	Definition
$\mathbf{Card}(\cdot)$	The number of positive component
$\lVert\cdot\rVert_2$	Euclidean 2-norm
$\lVert\cdot\rVert_1$	Euclidean 1-norm

- **Formulation**

$$Minimize_w \quad \lVert \hat{r}^\top w - I \rVert^2 \quad \Longrightarrow \text{Minimize tracking error}$$

$$s.t \quad \mathbf{1}^\top w = 1 \quad \Longrightarrow \text{Sum of portfolio weight} = 1$$

$$\mathbf{Card}(w) \leq k \quad \Longrightarrow \text{Cardinality constraint}$$

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

$\min_{W \in S_n^+}$	$\mathbf{Tr}\left(\hat{R}W\right) - 2\left(\mathbf{1}^\top W \hat{R} C\right)$
	$\mathbf{Tr}(I_N W) = 1$
	$\mathbf{1}^\top \lvert W \rvert \mathbf{1} \leq k\mathbf{Tr}(W)$
	$W \in S_n^+$

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

$$\min_{W \in S_n^+} \quad \mathbf{Tr}\left(\hat{R}W\right) - 2\left(\mathbf{1}^\top W \hat{R} C\right)$$

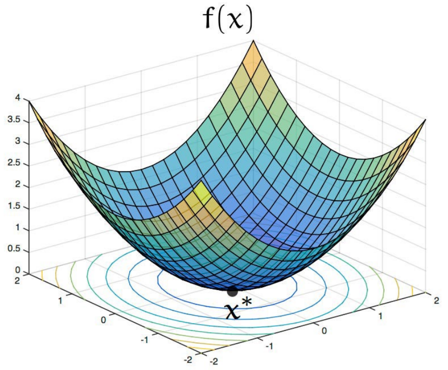
$$\mathbf{Tr}(I_N W) = 1$$

$$\mathbf{1}^\top |W| \mathbf{1} \leq k\mathbf{Tr}(W)$$

$$W \in S_n^+$$

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

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

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

Semi-definite Relaxation

CVXpy

$$
\begin{aligned}
x^\star(\theta) = \underset{x}{\operatorname{argmin}} \quad & f(x;\theta) \\
\text{subject to} \quad & g(x;\theta) \leq 0 \\
& h(x;\theta) = 0
\end{aligned}
$$

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)

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

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)

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

- Since I^2 is constant,

$\min_{w\in\mathbb{R}^n}$	$\|\hat{r}^\top w - I\|^2$

$\rightarrow$

$\min_{W\in S_n^+}$	$\mathbf{Tr}\left(\hat{R}W\right) - 2I(\mathbf{1}^\top W\hat{r})$

Semi-definite reformulation

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

$$
\begin{aligned}
\min_{w \in \mathbb{R}^n} \quad & \|\hat{r}^\top w - I\|^2 \\
& \mathbf{1}^\top w = 1 \\
& \mathbf{Card}(w) \le k
\end{aligned}
\qquad \longrightarrow \qquad
\begin{aligned}
\min_{W \in S_n^+} \quad & \mathbf{Tr}\left(\hat{R}W\right) - 2I(\mathbf{1}^\top W \hat{r}) \\
& \mathbf{Tr}(I_N W) = 1 \\
& \mathbf{Card}(W) \le k^2
\end{aligned}
$$

Semi-definite reformulation

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

$\min_{w \in \mathbb{R}^n}$	$\lVert \hat{r}^{\top} w - I \rVert^2$
	$\mathbf{1}^{\top} w = 1$
	$\mathbf{Card}(w) \leq k$
	$W = ww^{\top}$

$\min_{W \in S_n^+}$	$\mathbf{Tr}\left(\hat{R}W\right) - 2I(\mathbf{1}^{\top} W \hat{r})$
	$\mathbf{Tr}(I_N W) = 1$
	$\mathbf{Card}(W) \leq k^2$
	$\mathbf{Rank}(W) = 1$
	$W \in S_n^+$

Semi-definite relaxation

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

$\min_{W \in S_n^+}$	$\mathbf{Tr}\left(\hat{R}W\right) - 2I(\mathbf{1}^\top W\hat{r})$
	$\mathbf{Tr}(I_n W) = 1$
	$\mathbf{Card}(W) \leq k^2$
	$\cancel{\mathbf{Rank}(W) = 1}$
	$W \in S_n^+$

Semi-definite relaxation

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

$$
\begin{array}{|l|}
\hline
\min_{W \in S_n^+} \quad \mathbf{Tr}\left(\hat{R}W\right) - 2I(\mathbf{1}^\top W \hat{r}) \\
\hline
\mathbf{Tr}(I_n W) = 1 \\
\hline
\mathbf{Card}(W) \leq k^2 \\
\hline
\cancel{\mathbf{Rank}(W) = 1} \\
\hline
W \in S_n^+ \\
\hline
\end{array}
\xrightarrow{\text{Semi-definite Relaxation}}
\begin{array}{|l|}
\hline
\min_{W \in S_n^+} \quad \mathbf{Tr}\left(\hat{R}W\right) - 2(\mathbf{1}^\top W \hat{r}) \\
\hline
\mathbf{Tr}(I_N W) = 1 \\
\hline
\mathbf{1}^\top|W|\mathbf{1} \leq k\mathbf{Tr}(W) \\
\hline
W \in S_n^+ \\
\hline
\end{array}
$$

DPP-compliant

- When using CvxpyLayer, the formulation must satisfy DPP-compliant
    1. The optimization problem must be convex in variable W → **Already Satisfied**

DPP-compliant

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

DPP-compliant

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

$\min_{W \in S_n^+}$	$\mathbf{Tr}\left(\hat{R}W\right) - 2I(\mathbf{1}^{\top}W\hat{r})$
	$\mathbf{Tr}(I_N W) = 1$
	$\mathbf{1}^{\top}\lvert W\rvert\mathbf{1} \leq k\mathbf{Tr}(W)$
	$W \in S_n^+$

$\longrightarrow$

$\min_{W \in S_n^+}$	$\mathbf{Tr}\left(\hat{R}W\right) - 2\left(\mathbf{1}^{\top}W\hat{R}C\right)$
	$\mathbf{Tr}(I_N W) = 1$
	$\mathbf{1}^{\top}\lvert W\rvert\mathbf{1} \leq k\mathbf{Tr}(W)$
	$W \in S_n^+$

DFL loss for Partial Index Tracking

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

$$\mathcal{L}(\hat{\boldsymbol{R}}, \boldsymbol{R}) := \boldsymbol{Regret}(\boldsymbol{w}^*(\hat{\boldsymbol{R}}), \boldsymbol{R})$$

$$= \boldsymbol{f}(\boldsymbol{w}^*(\hat{\boldsymbol{R}}), \boldsymbol{R}) - \boldsymbol{f}(\boldsymbol{w}^*(\boldsymbol{R}), \boldsymbol{R})$$

DFL loss for Partial Index Tracking

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

$$\mathcal{L}(\widehat{\boldsymbol{R}}, \boldsymbol{R}) := \boldsymbol{Regret}(\boldsymbol{w}^*(\widehat{\boldsymbol{R}}), \boldsymbol{R})$$

$$= \underline{\boldsymbol{f}(\boldsymbol{w}^*(\widehat{\boldsymbol{R}}), \boldsymbol{R})} - \boldsymbol{f}(\boldsymbol{w}^*(\boldsymbol{R}), \boldsymbol{R})$$

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

DFL loss for Partial Index Tracking

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

$$\mathcal{L}(\widehat{R}, R) := Regret(w^*(\widehat{R}), R)$$

$$= \underbrace{f(w^*(\widehat{R}), R)}_{\text{Realized decision quality with estimated } \widehat{R}} - \underbrace{f(w^*(R), R)}_{\text{Realized decision quality with ground truth } R}$$

DFL loss for Partial Index Tracking

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

$$\mathcal{L}(\widehat{\boldsymbol{R}}, \boldsymbol{R}) := \boldsymbol{Regret}(\boldsymbol{w}^*(\widehat{\boldsymbol{R}}), \boldsymbol{R})$$

$$= \underbrace{\boldsymbol{f}(\boldsymbol{w}^*(\widehat{\boldsymbol{R}}), \boldsymbol{R})} - \underbrace{\boldsymbol{f}(\boldsymbol{w}^*(\boldsymbol{R}), \boldsymbol{R})}$$

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

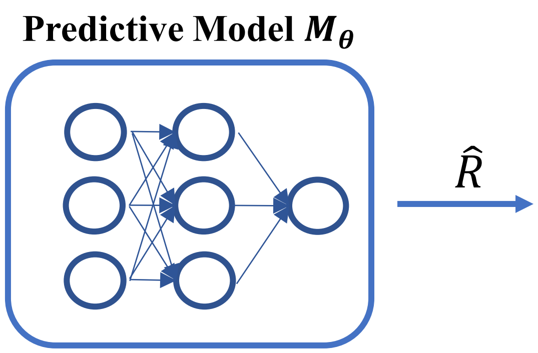
Realized decision quality with ground truth R

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

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

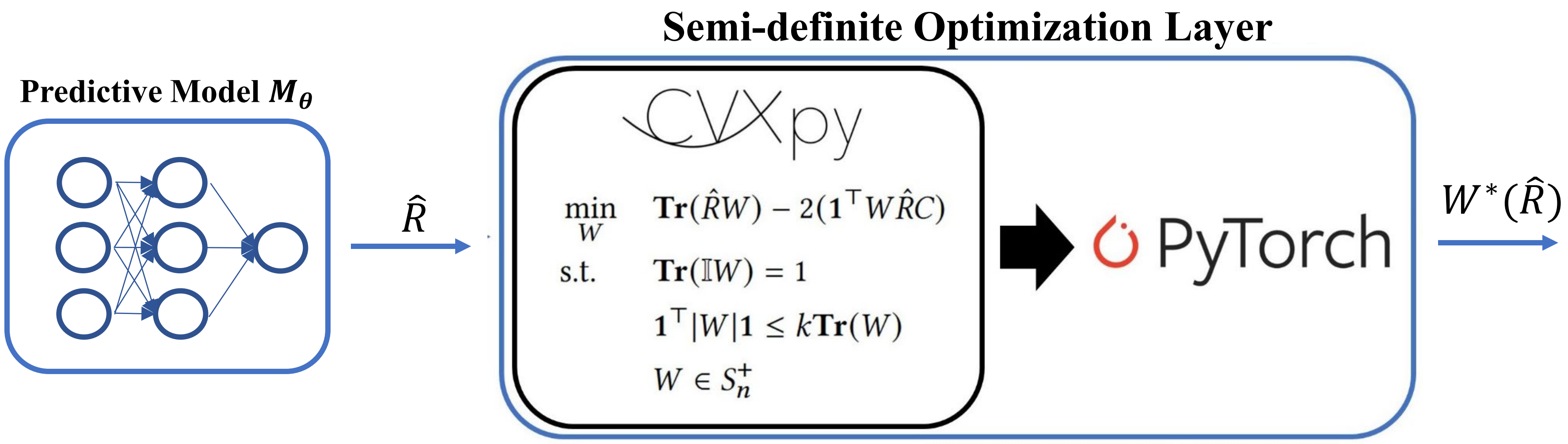
DFL with semi-definite relaxation

- The process of DFL in Partial Index Tracking



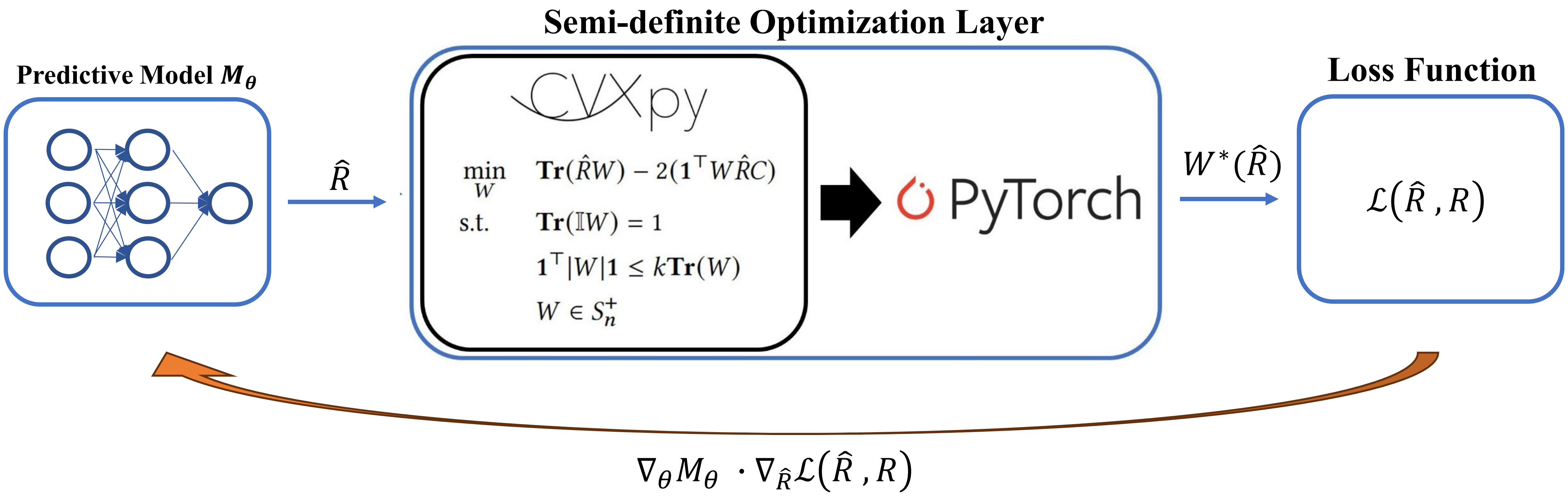
DFL with semi-definite relaxation

- 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

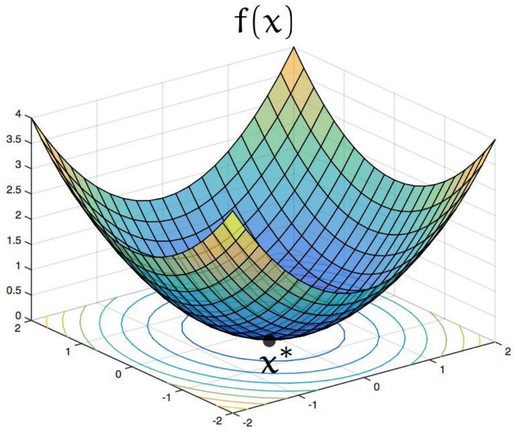
$$\mathcal{L}(\hat{r}, r) = \|r^\top w^*(\hat{r}) - I\|^2 - \|r^\top w^*(r) - I\|^2$$

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

r

$\hat{r}$

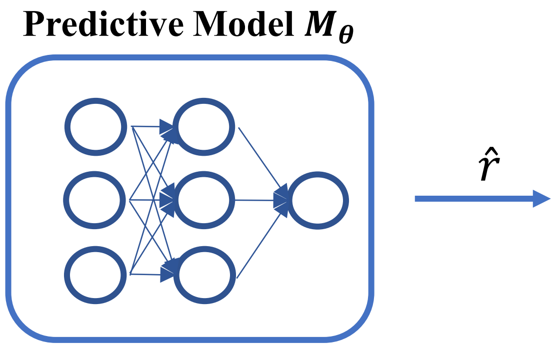
$\longrightarrow$



$\approx \; \mathcal{L}(r, w^*(\hat{r}))$

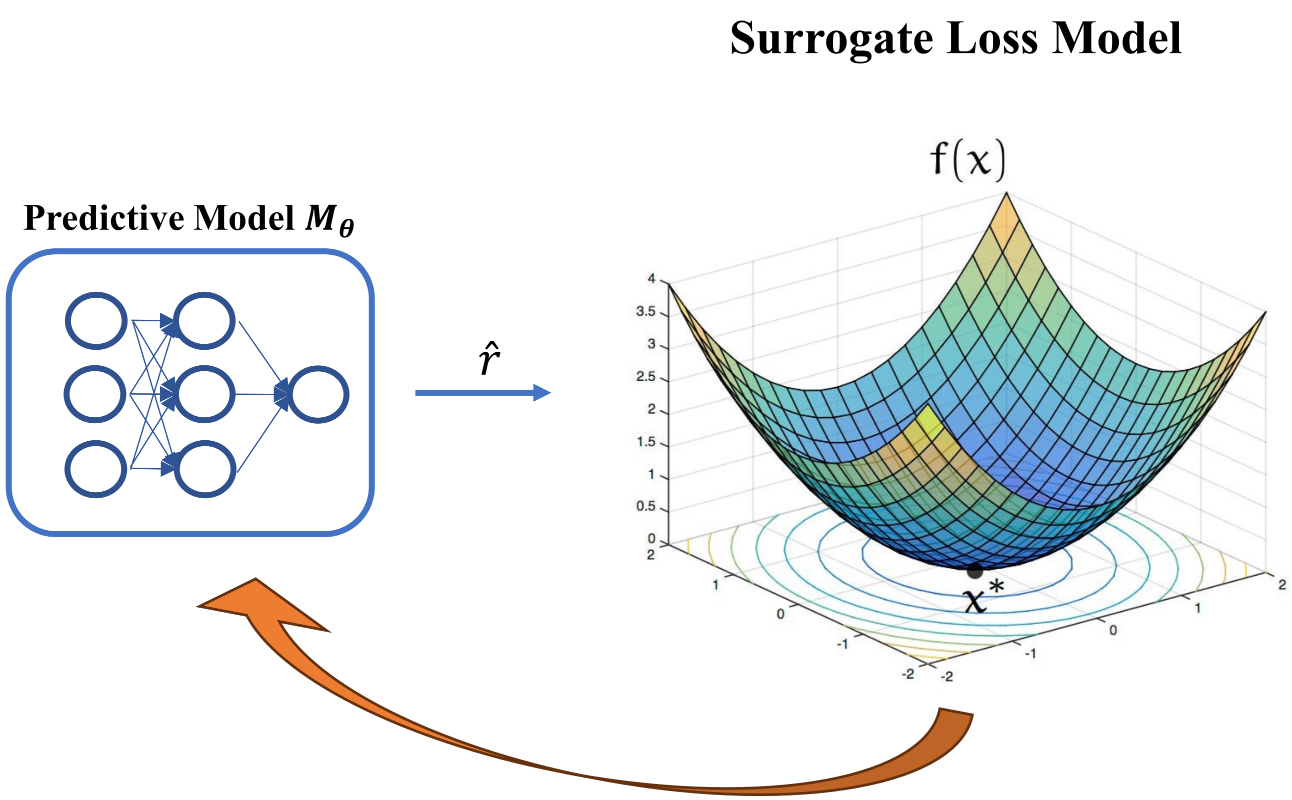
DFL with surrogate approach

- The process of DFL in Partial Index Tracking



DFL with surrogate approach

- The process of DFL in Partial Index Tracking



$$\nabla_\theta M_\theta \cdot \nabla_{\hat{r}} \mathbf{Surrogate}(\hat{r}, r)$$

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

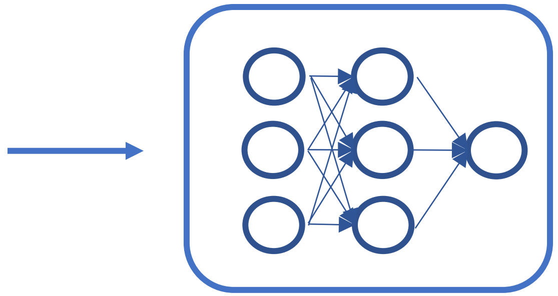
Mean and variance of past returns

Mean and variance of past prices

Current return

Current price

Feed Forward Network



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

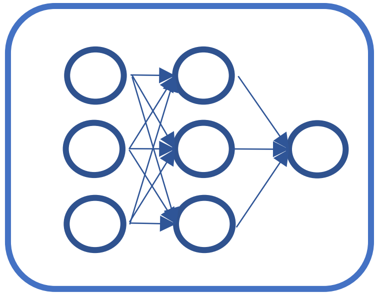
Mean and variance of past returns

Mean and variance of past prices

Current return

Current price

Feed Forward Network



Future cumulative return

Hands-on Exercise

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

Notebook & Data Link

Thank you for listening!