FINTECH FINAL PROJECT
A.Y. 2022/2023

Investment Replica

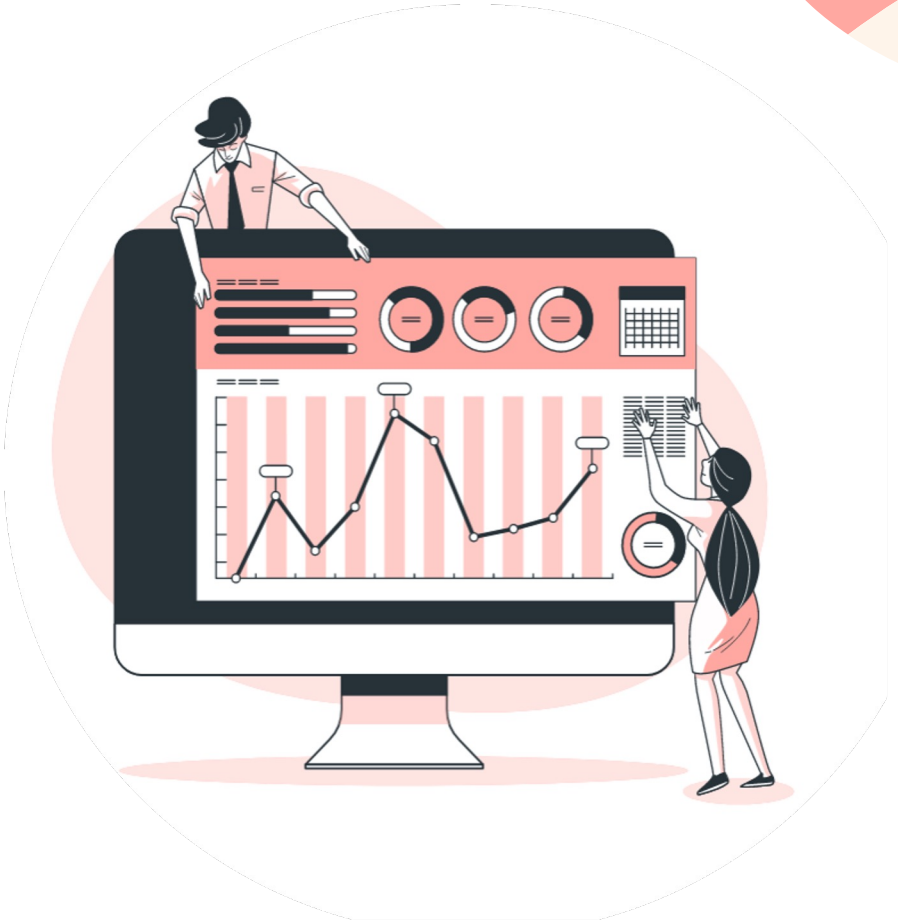
Ghesini Matteo

Mazzè Alice

Tegon Anna

Volterra Camilla

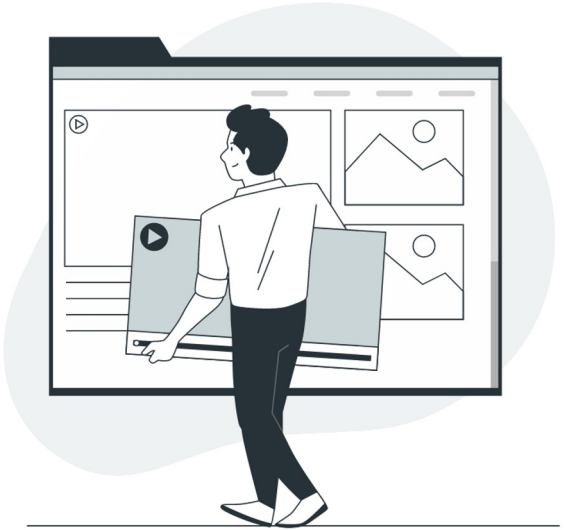
Wu Qiao



OUR GOAL

Replicate a Monster index with Futures

Our steps



01

Construction of the Monster index and selection of replicating portfolio

02

Construction of a Regression Benchmark

03

Application of the Kalman Algorithm

04

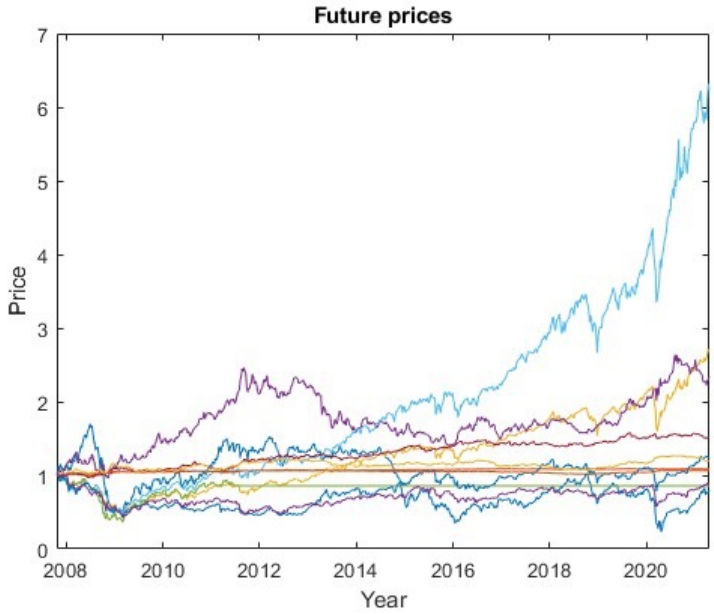
Cost considerations of replication strategies

05

Conclusions

MONSTER INDEX	WEIGHTS
FRXGL	$\frac{1}{3}$
MXWO	$\frac{1}{3}$
LEGATRUU	$\frac{1}{3}$

REPLICATING PORTFOLIO FUTURES



- Stationarity of Monster index returns
- Stationarity of Futures returns

LASSO REGRESSION

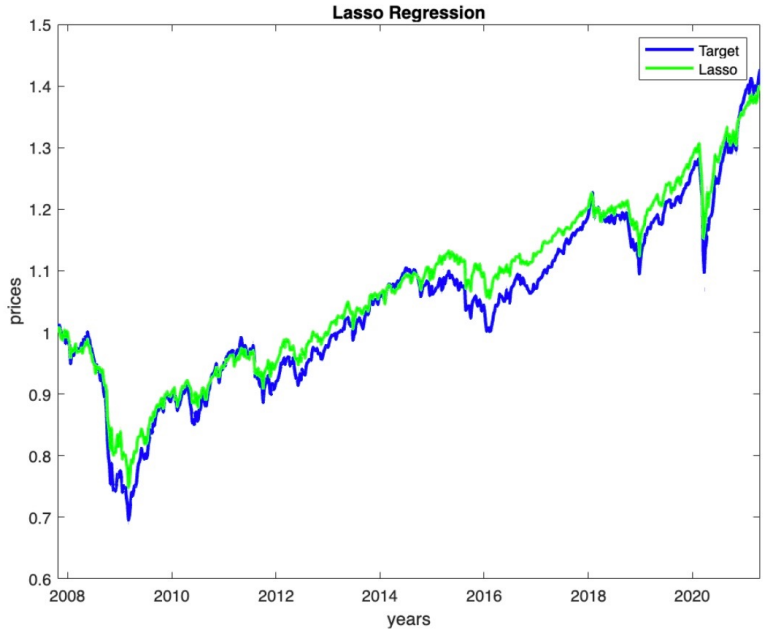
TEV_lasso = 0.0285

vs

RIDGE REGRESSION

TEV_ridge = 0.0716

BENCHMARK



ASSET REPLICATION ALGORITHM PROCESS FLOW
KALMAN FILTERING

TIME UPDATE («PREDICT»)

1. Project the state ahead

$$\hat{x}^{-} = \hat{x}_{t-1}$$

2. Project the error covariance ahead

$$P_t^{-} = P_{t-1} + V1$$

Initial condition
for $\hat{x}_{t-1}$ and P_{t-1}

MEASUREMENT UPDATE («CORRECT»)

1. Compute the Kalman gain

$$K_t = P_t^{-} H_t^T (H_t P_t^{-} H_t^T + V2)^{-1}$$

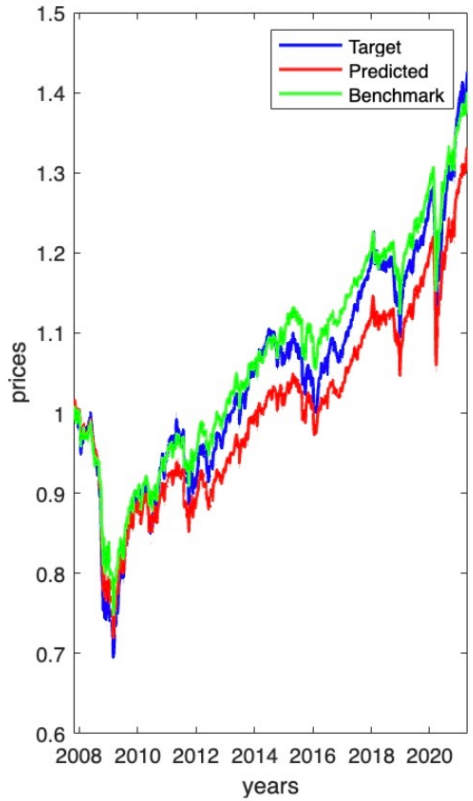
2. Update estimate with measurement:

$$\hat{x}_t = \hat{x}_t^{-} + K_t(r_t^{observed} - H_t \hat{x}_t^{-})$$

3. Update the error covariance

$$P_t = (I - K_t H_t) P_t^{-}$$

TEV Kalman=0.020639
TEV benchmark=0.0288



Kalman Filter Algorithm is very sensitive to initial conditions

EMPIRICAL INITIAL WEIGHTS

RANDOM COMBINATION OF INITIAL WEIGHTS

UNIFORM INITIAL WEIGHTS

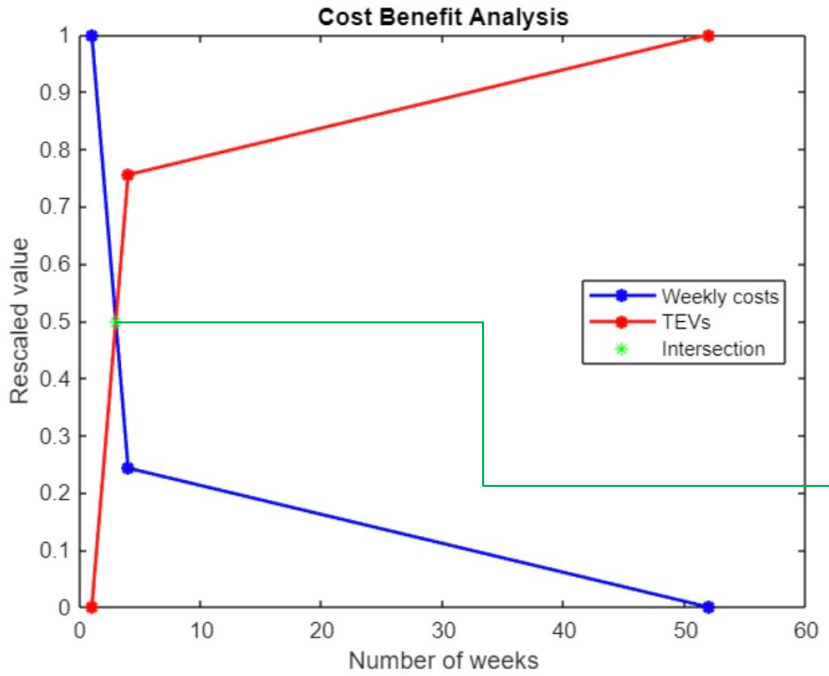
COST BENEFIT ANALYSIS

WEIGHTS	WEEKLY COSTS ($)	TEV
WEEKLY WEIGHTS	43.5156	0.0206
MONTHLY WEIGHTS	11.4347	0.0296
YEARLY WEIGHTS	1.1364	0.0325

Actual cost of transaction

It is assumed to be 5$ for each transaction, indipendently of quantity selled or buyed, but just of the open transaction.

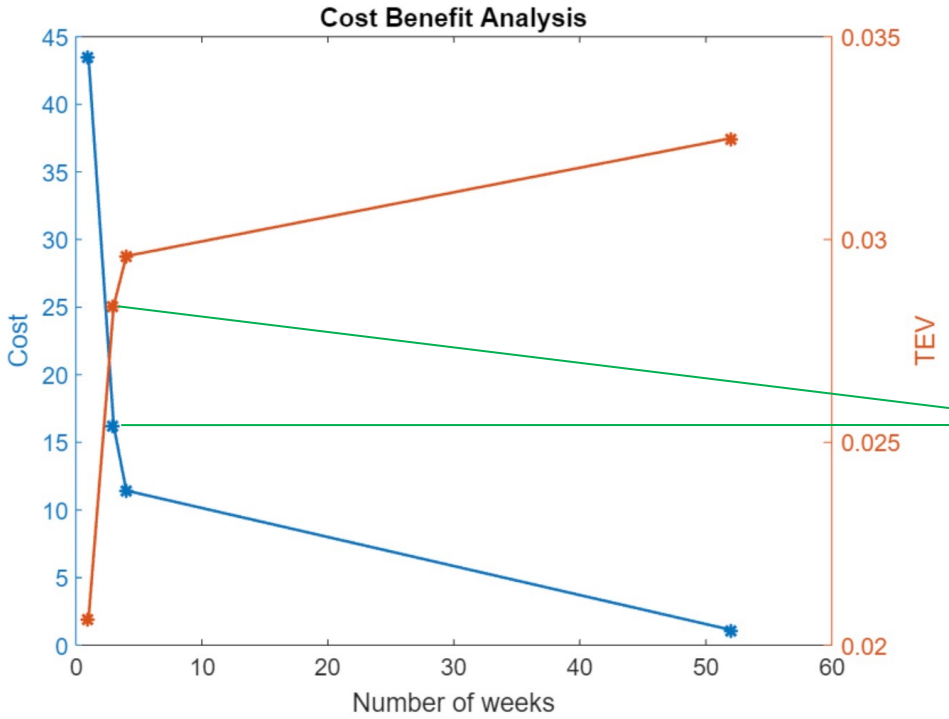
COST BENEFIT ANALYSIS



* Costs: week, month, year
* TEVs: week, month, year

THEORICAL OPTIMAL POINT
2.9840 weeks

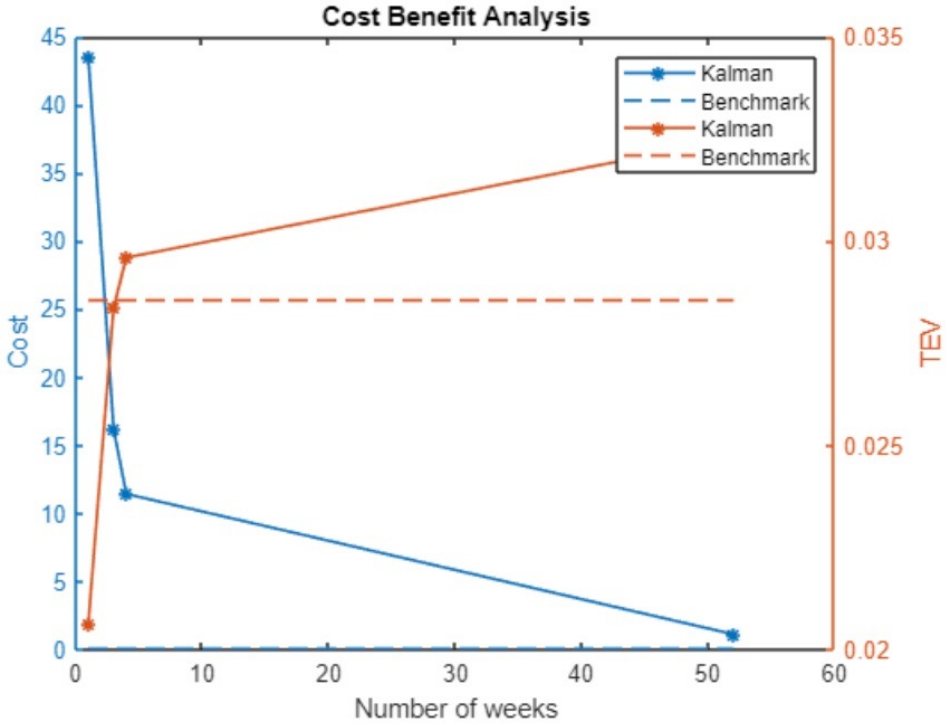
COST BENEFIT ANALYSIS



* Costs: week, 3 week, month, year
* TEVs: week, 3 week, month, year

EMPIRICAL OPTIMAL POINT
3 weeks

LASSO vs KALMAN



TEV_lasso = 0.0285 > TEV_3week = 0.0284

!

Cost_lasso= 0.0781 > Cost_3week = 16.2145

DYNAMICAL WEIGHTS
Kalman Filter Algorithm

- Sensible to calibration
- Limited capability of the model to be reused in different context
- High transaction costs
- Interesting approach
- Capable to catch fluctuations

CONSTANT WEIGHTS
Lasso Regression

- Reduction of transaction costs
- Robust
- Less precision

Thanks!

REFERENCES

- r-project.org
- it.mathworks.com
- Stevens institute of technology, *Asset replication via Kalman Filtering FE 800 special problems in FE*
- Youngjoo Kim and Hyochoong Bang, *Introduction to Kalman Filter and its application*, Felix Govaeres, 2018.