Stanford | ENGINEERING
Management Science & Engineering

MS&E 349 – Final Presentation 2025

Stock Index Return Predictions

Bocheng Dai, Yifan Geng, James Liu, Tiankai Yan

Jagannathan, Ravi, Yuan Liao, and Andreas Neuhierl. "Robust Stock Index Return Predictions Using Deep Learning." Available at SSRN 4890466 (2023).

Bryzgalova, Svetlana, et al. "Missing financial data." The Review of Financial Studies 38.3 (2025): 803-882.

Introduction

- This paper proposes a robust **conditional machine learning** approach for predicting stock index returns.
- Unlike traditional models that rely on stable relationships over time, this model addresses forecast instability by leveraging the **rich cross-sectional data on asset returns and observable firm characteristics**.
- It uses **period-by-period machine learning** framework to estimate firm-level expected returns that are free of idiosyncratic noise while preserves the factor structure of realized returns.
- A key innovation is the use of **stock betas as instrumental variables** to construct more stable book-to-market factors, improving predictive accuracy over short horizons.

Model

- The market index return as weighted average of individual stock returns: $y_{t+1} = \sum_{i=1}^{N} w_{i,t}\, x_{i,t+1}$
- The paper is based on the following assumptions:
  - Firm-level returns and book-to-market ratios are driven by conditional factor models :

$$x_{i,t+1} = \beta_{i,t}^{\top} f_{t+1} + u_{i,t+1} \qquad v_{i,t} = \lambda_{i,t-1}^{\top} g_t + \eta_{i,t}$$

  - Stock factors are driven by lagged book-to-market factors:

$$f_{t+1} = \Phi_0 + \Phi_g\, g_t + e_{t+1}$$

  - Factor loadings are modeled as functions of observable firm characteristics:

$$\beta_{i,t-1} = h_{\beta,t}(z_{i,t-1})$$

- Given the assumptions, we get the final model:

$$y_{t+1} = \rho_{0,t} + \rho_{g,t}^{\top}\, g_t + \epsilon_{t+1}$$

We further assume that the coefficients are constants or slowly moving over time.

Algorithm Overview

Input:

- Firm characteristics $z_{i,t-1}$
- Realized returns $x_{i,t}$
- Valuation signal (e.g. book-to-market) $v_{i,t}$

Algorithm Overview

Step 1: Estimate conditional expected returns via cross-sectional DNN

Denoise $x_{i,t}$ by regressing on $z_{i,t-1}$ at each time t

$$\hat{m}_t = \arg\min_{m \in \mathrm{DNN}} \sum_{i=1}^{N} \left(x_{i,t} - m(z_{i,t-1})\right)^2$$

$$\hat{x}_{i,t} = m_t(z_{i,t-1}) \approx \mathbb{E}_t(x_{i,t} \mid z_{i,t-1})$$

Algorithm Overview

Step 2: Estimate time-varying betas using local PCA on $\hat{x}_{i,t}$

Extract principal components from the weighted covariance matrix of denoised returns

$$S_t = \sum_{s=1}^{T} K_{s,t}\, \hat{x}_s \hat{x}_s^\top \quad \Rightarrow \quad \beta_{i,t} = \text{Top eigenvectors of } S_t$$

With kernel weights:

$$K_{s,t} = \frac{1}{h} K\left(\frac{s-t}{Th}\right) A_t^{-1} \quad \text{(e.g., quartic kernel)}$$

Algorithm Overview

Step 3: Construct BM-factor g_t via instrumental variables

Model assumption:

$$v_{i,t} = \lambda_{i,t-1}^{\top} g_t + \eta_{i,t}$$

We use projected betas as IVs to estimate g_t from $v_{i,t}$:

$$\tilde{\lambda}_{i,t-1}^{IV} = P_{z_{i,t-1}} \beta_{i,t-1} \quad \Rightarrow \quad \hat{g}_t = \left(\sum_i \tilde{\lambda}_{i,t-1}^{IV} \tilde{\lambda}_{i,t-1}^{IV\top} \right)^{-1} \sum_i \tilde{\lambda}_{i,t-1}^{IV} v_{i,t}$$

Algorithm Overview

Step 4: Forecast market return y_{t+1} using the estimated BM-factor $\hat{g}_t$

We use $\hat{g}_t$ as a predictive feature in a time-series regression to forecast the market excess return:

$$\hat{y}_{T+1|T} = \hat{\rho}_0 + \hat{\rho}_g^\top \hat{g}_T$$

where:

- $\hat{g}_T$ is the BM-factor estimated in Step 3
- $\hat{\rho}_0$, $\hat{\rho}_g$ are estimated via the time-series model:

$$y_t = \rho_0 + \rho_g^\top \hat{g}_{t-1} + u_t$$

This allows us to link valuation-based signals to future market return forecasts.

Algorithm Overview

Step 5: Construct confidence interval for $\hat{y}_{T+1|T}$

Given the forecast $\hat{y}_{T+1|T}$, we can construct a $100(1-\tau)\%$ confidence interval using the estimated standard error:

$$\left[\hat{y}_{T+1|T} - z_\tau \cdot \mathrm{SE}(\hat{y}_{T+1|T}), \quad \hat{y}_{T+1|T} + z_\tau \cdot \mathrm{SE}(\hat{y}_{T+1|T})\right]$$

where:

- z_τ is the critical value from the standard normal distribution (e.g., 1.96 for 95%)
- $\mathrm{SE}(\hat{y}_{T+1|T})$ is the estimated standard error from the regression

Contribution

Comparison with Commonly Used ML Approaches

- **Naive CML**: a cross-sectional DNN is trained on the latest period and then used to forecast returns based on updated firm characteristics:

$$\hat{m}_T(z) = \arg \min_{m \in \text{DNN}} \sum_{i=1}^{N} \left(x_{i,T} - m(z_{i,T-1})\right)^2 \quad \Rightarrow \quad \hat{y}_{T+1}^{\text{Naive}} = \sum_{i=1}^{N} w_i \hat{m}_T(z_{i,T})$$

Recall that

$$x_{i,t} = h_{\beta,t}(z_{i,t-1})^\top f_t + u_{i,t} \quad \text{and} \quad \hat{m}_T(z) \xrightarrow{p} h_{\beta,T}(z)^\top f_T$$

Under this setup, we have

$$\hat{y}_{T+1}^{\text{Naive}} \xrightarrow{p} \sum_{i=1}^{N} w_i\, h_{\beta,T}(z_{i,T})^\top f_T \quad \text{vs.} \quad y_{T+1} \approx \sum_{i=1}^{N} w_i\, \beta_{i,T}^\top f_{T+1}$$

Contribution

Comparison with Commonly Used ML Approaches

- **Pooled ML**: this approach trains a single model using data pooled over all time periods and cross-sections:

$$\hat{m}(z) = \arg\min_{m \in \mathrm{ML}} \sum_{t=1}^{T} \sum_{i=1}^{N} \left(x_{i,t} - m(z_{i,t-1})\right)^2 \quad \Rightarrow \quad \hat{y}_{T+1}^{\text{Pooled}} = \sum_{i=1}^{N} w_i \hat{m}(z_{i,T})$$

It's been shown that:

$$\hat{y}_{T+1}^{\text{Pooled}} \xrightarrow{p} \mathbb{E}\left[y_{T+1} \mid \mathcal{F}_{z,T}\right]$$

Thus, we have

$$y_{T+1} = \hat{y}_{T+1}^{\text{Pooled}} + \rho_g^{\top}\left(g_T - \mathbb{E}\left[g_T \mid \mathcal{F}_{z,T}\right]\right) + \epsilon_{T+1} \quad \text{vs.} \quad y_{T+1} = \hat{y}_{T+1|T} + \epsilon_{T+1}$$

Characteristics

Characteristic Section	Examples	Relevance
Past Returns	r2_1, r12_2, LT_Rev	Use historical return patterns to forecast.
Investment	Investment, NOA, DPI2A, NI	Measure capital allocation and financing activities.
Profitability	PROF, ATO, PM, ROA, SGA2S	Measure operating and earnings generation to indicate the corporate performance.
Intangibles	AC, OA, OL, PCM	Capture accounting-based, non-cash signals of earnings quality.
Value	BEME, A2ME, CF2P, D2P, Q	Measure the corporate value thought fundamentals, including valuation ratios and balance-sheet metrics.
Trading Frictions	Spread, IdioVol, LTurnover, Resid_Var, SUV	Quantify liquidity constraints and short-term return anomalies from market microstructure.

Table 1: Overview of Characteristic Sections

- Dataset of Characteristics and Returns
- Monthly
- Rank Transformed
- Imputation against Missing Ratio

Algorithm Implementation

- Use a rolling estimation window of 60 months
- Fit model on months t−59 through t, then predict market excess return for month t+1
- Slide window forward one month at a time

Conditional Machine Learning (CML)

Single-factor linear model using only the book-to-market (BM) factor

$$\hat{y}_{b,t+1|t} = \rho_0^b + \rho_g^b \, g_{b,t}$$

Algorithm Implementation

Traditional Tuning Approaches

Fixed “tuning period”: Select hyperparameters once on a historical window and keep them fixed thereafter

- Drawback: assumes stable predictive relationships, which may break down

Time-series cross-validation: Slide internal train/validation splits through time

- Drawback: still sensitive to a few extreme outliers, can overfit those periods

Positive-Forecast Frequency Proposed by the paper

- For each candidate model **M**, count how many times $\hat{y}_{b,t+1|t}(M) > 0$ during a tuning window
- Choose the model **M** that maximizes the number of positive forecasts in that window (or consecutive # of positive forecasts)

Every 12 months, re-run tuning using the most recent 60 months of data

In each “post-tuning” year, use the selected model that maximizes #positive forecasts to generate monthly out-of-sample predictions

Key Benefits: Avoids overly complex in-sample fitting that fails out-of-sample

Imposes an economically motivated sign constraint

Algorithm Implementation

Feed-Forward Neural Networks for Expected Returns

- we used a standard feed-forward MLP network
- Train by stochastic gradient descent (SGD) with the Adam optimizer

Network Hyperparameters

Activation	ReLU
LR	0.001
# epochs	2000

One hidden layer: 32 nodes

Two hidden layers: 32 → 16 nodes

Three hidden layers: 32 → 16 → 8 nodes

Economic Explainability

Forecast uncertainty

We quantify that confidence or uncertainty by the model's **forecast standard error, (written as w_t):**

When **w_t** is large, the model is saying "I'm not sure, my prediction could be off by a lot."
When **w_t** is small, the model is saying "I'm pretty sure, I think my prediction will be quite accurate."

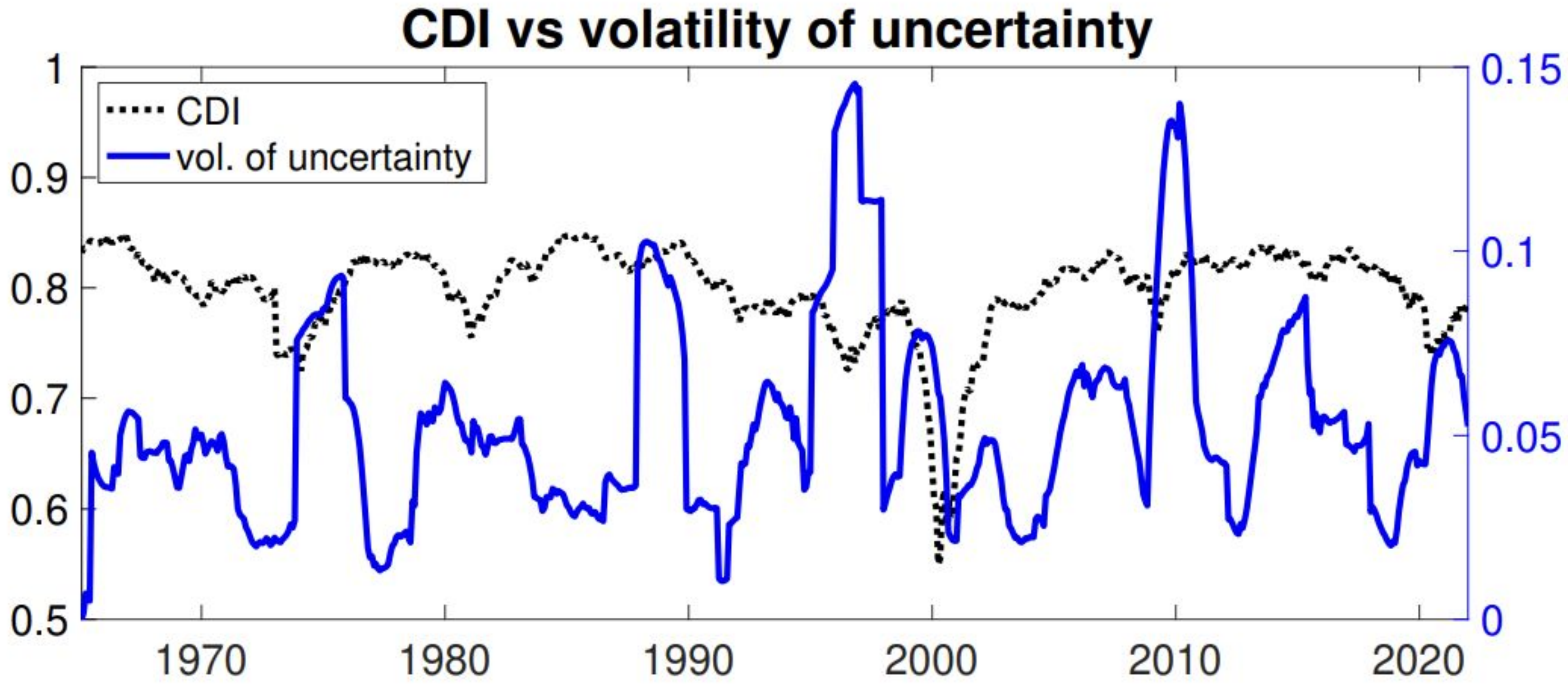
We use std(**w_t**) to measure "How unstable has the model's own uncertainty been over the past two years?"

CDI: Creative Destruction Index

The correlation between the two ranking lists (sales-rank vs. market-cap-rank)

- **When CDI is low** (market ranks by sales and by market cap disagree), it signals that the market is undergoing heavy "creative destruction,"old leaders are being overtaken. In that environment, stock prices and company values can change very unpredictably.
- **Conversely, if std(w_t) suddenly rises**, it means our model's uncertainty has been bouncing around a lot. That volatility of uncertainty is often triggered by a phase of market restructuring—so we should check CDI, which is likely already falling or at a low point. In other words, a spike in sdt(w_t) signals impending creative destruction.

Experiment Result & Evaluation



Experiment Result & Evaluation

		Forecast periods			
		1964-1999	2000-2007	2008-2020	full period
$[t : \text{end}]$-R_t^2	CML	0.186	0.235	0.570	0.839
	PCA	3.581	0.231	2.367	3.755
	PCA-ker	0.845	0.753	4.238	4.247
	GW-linear	10.41	2.747	5.991	12.56
	GW-Fourier	0.061	0.166	0.266	0.193
	Pooled-ML	0.463	0.585	1.115	0.911
$[1 : t]$-R_t^2	CML	0.919	0.121	0.106	0.891
	PCA	4.831	0.460	0.447	4.725
	PCA-ker	0.916	0.197	0.406	0.904
	GW-linear	19.66	3.059	2.531	15.91
	GW-Fourier	0.644	0.062	0.059	0.512
	Pooled-ML	1.746	0.291	0.275	1.368