Modeling Corporate Bond Returns Using The IPCA Model

By: Marcus Walton, Neil Nieh

Agenda

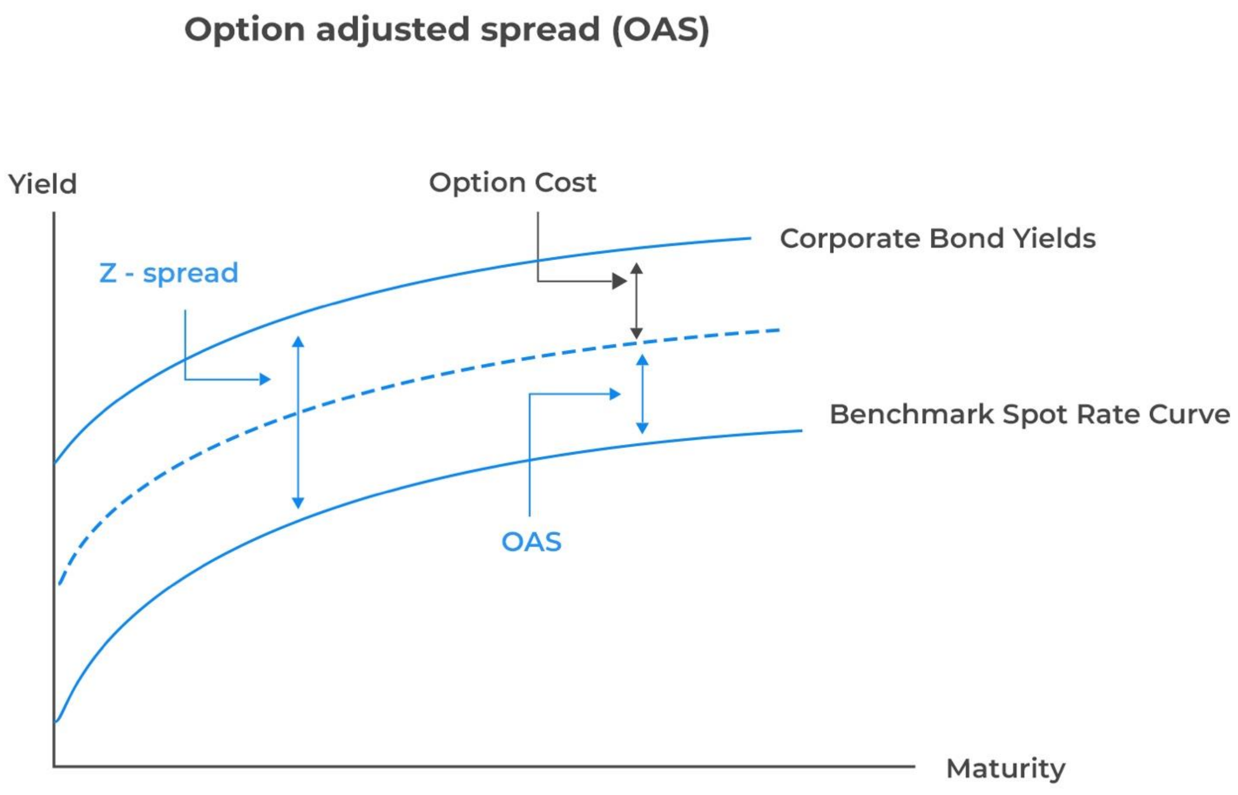
- Recap
- In-Sample Analysis
- Out-Of-Sample Analysis
- Model Performance
- Next Steps
- Machine Learning Models
- Questions

Recap: Option-Adjusted-Spread (OAS)

OAS is the spread over the Treasury curve after accounting for embedded options

Why it matters:

- Removes distortion from callable/puttable options
- Allows comparisons across bonds on same scale
- Captures credit and liquidity risk



Distance-to-Default

- **One of the features in the IPCA Model**
- D2D appears **indirectly** in the model as part of **“spread-to-D2D”**, one of the five key bond characteristics used to instrument time-varying factor betas.
- The authors identify **“spread divided by D2D”** as a bond “value” characteristic: it compares market compensation (spread) to a structural measure of risk (D2D).

IPCA in Detailed

- Core Formula:
- $r_{i,t+1} = \beta_{i,t} f_{t+1} + \varepsilon_{i,t+1}$
- **But what makes IPCA different?**
  - Most models assume β is fixed
  - **IPCA lets the bond's risk exposures change over time**, depending on observable characteristics (e.g., bond spread, duration, rating, etc.).
  - It says:

$$\beta_{i,t} = z_{i,t}\Gamma$$

Kelly’s Approach – Duration Match Treasury Hedge

Total Bond Return
(Coupon + Price)

Subtract Duration-Matched Treasury Return ⟶ Isolate Credit Return

Credit Excess Return
Input into IPCA

📌 **Objective**: Strip out interest rate risk, focus purely on credit return.

“The credit excess return is the total return minus a **hedging portfolio of Treasuries**. The hedge portfolio is built such that if there is a small move on the risk-free curve, **the hedge portfolio offsets the response in the risky bond’s price**.”

Calculating excess return by this approach, like key rate duration matching, allows us to isolate credit risks.

Kelly's Approach – Duration Match Treasury Hedge

Kelly (duration-matched hedge)

Uses Treasury ladder whose dollar-duration equals the bond's

Easier to compute, less noisy

Residual rate exposure is learnt and priced

ICE KRD bucket hedge

Builds seven buckets (3 m, 2 y, 5 y, … 30 y). Hedge weights are chosen so every bucket's key-rate duration is neutral → better for **non-parallel** twists.

Useful when we care about specific curve-twist scenarios.

KRD-neutral excess returns allows feeding an even "cleaner" credit series into IPCA; **empirically the incremental difference is modest unless you look at very long bonds.**

What is an In-Sample Analysis?

- Fit the model on a given historical data set and then evaluate its performance **on that same data**.
- **Purpose:**
  - **Diagnostics**: Check that model "learns" the main patterns (e.g. factor loadings, intercepts) without obvious misspecification.
  - **Goodness of Fit Metrics**: Compute R^2, residual diagnostics—judged on the very data used to estimate the model.
- **Overfitting Risk**: A model with many parameters can "memorize" noise and look great in-sample, but perform poorly on new data.

What is an Out-Of-Sample Analysis?

- Out-of-sample analysis evaluates a model’s predictive performance on data not used to estimate its parameters. In a time-series setting, we group the data into:
- Training (Estimation) Window: the initial subset used to fit the model.
- Test Period: all subsequent observations, where at each date t we “freeze” the model estimated up through *t* and generate a one-step-ahead forecast for *t+1*.

Training Window and Lags

- Length & Type
  - Rolling window: a fixed number of most recent observations is used to estimate parameters at each date. As *t* advances, the window "rolls" forward, dropping the oldest data and adding the newest.
- Lags
  - To forecast returns at *t+1*, predictor variables must be lagged so that only information available at or before time *t* enters the model. This ensures no future information "leaks" into the forecast.

Evaluate Performance – R-squared

- $Total\ R^2 = 1 - \frac{\sum(r_{i,t+1} - \widehat{r_{i,t+1}})}{\sum r_{i,t+1}^2}$
- $Time\ Series\ R^2 = avg\ across\ bonds\ of\ \left(1 - \frac{\sum_t (r_{i,t+1} - \widehat{r_{i,t+1}})^2}{\sum_t r_{i,t+1}^2}\right)$
- $Cross - Sectional\ R^2 = avg\ across\ time\ of\ \left(1 - \frac{\sum_t (r_{i,t+1} - \widehat{r_{i,t+1}})^2}{\sum_t r_{i,t+1}^2}\right)$
- Time Series measure how well the model explain to a specific bond over time.
- Cross-Sectional looks at all bonds at one time. Measures how well do factor exposures explain why some bonds did better than others

In-Sample Analysis

Num of Factors	Intercept	Total R^2	Time Series R^2	Cross Sectional R^2
1	True	0.443	0.042	0.329
	False	0.434	0.013	0.305
2	True	0.488	0.306	0.368
	False	0.482	0.290	0.355
3	True	0.505	0.314	0.392
	False	0.502	0.292	0.393
4	True	0.520	0.292	0.412
	False	0.517	0.285	0.407
5	True	0.528	0.275	0.420
	False	0.527	0.262	0.416

Factor-Count Effects:

- **Rising Total R^2**: As **K** goes from 1 to 5 factors, **Total R^2** climbs from ~44% to ~53%.
- **Sharply Up at K=2**: The biggest jump is between 1→2 factors, indicating that a two-factor model explains more of the **joint** variation in bond returns than a single-factor model.

- **Diminishing Returns**: Gains taper off beyond 3 factors, suggesting that the marginal benefit of each extra latent factor is smaller.
- Kelly's report almost the same pattern in-sample: their Total R^2 goes from ~42% at K=1 to ~50% at K=5 .

Model Performance

Time-Series vs. Cross-Sectional R^2

- **Time-Series R^2** jumps from near zero at K=1 to ~30% at K=2–3, peaking around 31–32% at K=3, then dips slightly.
- **Cross-Sectional R^2** : starts around ~30% at K=1, rises steadily to ~42% at K=5.
- Kelly's Table III (Panel A), for individual bonds they find (in-sample):
  - Time-Series $R^2 \approx$ 33%, 37%, 39%, 42%, 43% at K=1–5
  - Cross-Section $R^2 \approx$ 23%, 26%, 28%, 32%, 33%
- Results are similar— jump at K=2–3 and then plateau—though cross-section figures are a bit higher, and time-series figures a bit lower.
- **I'll assume this comes from small differences in index - Kelly use ICE's H0A0 (High Yield) + C0A0 (Inv Grade) universes**

Model Performance

Role of the Intercept (α)

- Comparing **True** (intercept included) vs **False** (zero intercept) rows shows **very little change** in any R^2.
- Meaning **most** of explanatory power comes from the **factors themselves** rather than an alpha term.
- Kelly shows (in their bootstrap tests) that once you have enough factors ($K \geq 4$ or 5), adding intercepts (α’s) no longer significantly improves fits.

Portfolio Analysis

	Training Window	Lag	Factors	Intercept	Sharpe Ratio	Net Cost Sharpe	T-test
Kelly	Expanding	Unspecified	5	No	3.5	1.4	6.4
Our Model	120	25	5	Yes	3.48	1.46	7.8
				No	1.6	-0.68	1.3
	60	25	5	Yes	4.50	2.22	7.0
	120	20	5	Yes	3.81	1.61	8.5
	120	15	5	Yes	3.09	0.98	7.7

OOS Results

	Training Window	Lag	Factors	Intercept	Total R_2
Kelly	36	Unspecified	5	No	50.7
Our Model	120	25	5	Yes	48.1
				No	0.38
	60	25	5	Yes	4.50
	120	20	5	Yes	-5
	120	15	5	Yes	-10

Interpretation

Alignments :

- Our (120, lag=25, intercept=True) setting is very similar to Kelly's baseline. Net Sharpe = 1.46 vs. Kelly's 1.4, and t-test = 7.8 vs. Kelly's 6.4.
- Confirmed IPCA forecasts generate statistically significant long-short portfolios.

Misalignment :

- Without the intercept term, model performs poorly (Sharpe = 1.6 → Net = –0.68). This shows that the intercept is crucial in implementation.
- Kelly excludes the intercept to avoid absorbing pricing effects that should be captured by latent factors.

Machine Learning Implementation

Project Goal: Use the IPCA model to better understand what drives corporate bond returns and make return estimates more reliable

Next Step: Add machine learning to the model to help improve how accurately we can price bonds and spot patterns we might miss otherwise

Random Forest Regression

- **What it is:** A machine learning approach that combines many decision trees and averages their outputs to make more reliable and accurate predictions.

- **Why it fits with IPCA:** While IPCA captures key latent factors linearly, Random Forest can model complex, non-linear interactions between those factors and other market variables, providing a richer understanding of bond pricing.

Gradient Boosting

- **What it is:** A step-by-step learning method that builds a series of models, each one improving on the mistakes of the previous, to boost overall accuracy.

- **Why it fits with IPCA:** It complements IPCA by fine-tuning predictions, which is especially helpful in scenarios where even small pricing differences—like in illiquid or high-risk bonds—can have a big impact.

LSTM (Long Short-Term Memory Networks)

- **What it is:** A specialized neural network built to recognize patterns over time and remember important trends in sequential data.

- **Why it fits with IPCA:** While IPCA gives a snapshot of bond pricing at a specific moment, LSTM adds value by tracking how those prices change over time—making it especially useful for forecasting future bond returns using past trends.

Side by Side Analysis

Model	Strengths	Limitations
Random Forest	Easy to use, captures non-linearities, robust	Can overfit, less precise than boosting in financial use cases
Gradient Boosting	Highly accurate, great with structured data, fine-grained control	Requires tuning, slightly longer training times
LSTM	Excellent for time series forecasting, captures long-term patterns	Requires large datasets, more complex and harder to interpret

Which Machine Learning Model Best Complements IPCA?

Top Choice: Gradient Boosting

- Delivers high accuracy on structured financial data
- Handles non-linear relationships and subtle patterns in pricing
- Performs well even with noisy or imbalanced datasets (like illiquid or high-yield bonds)
- Efficient and interpretable — feature importance can guide financial insights

Questions?