Midlincoln Research

February - 5 2025

Analyst: Ovanes Oganisian

Quants

Ranking Before Prediction

Notes on Selection, Order, and Why Many Models “Work” for the Same Reason

Most quantitative finance discussions start with prediction. Mine started with selection.

Years ago, working with a PHP-based stock selection system long before today’s ML ecosystem, I found myself repeatedly returning to the same uncomfortable realization:

I didn’t actually need accurate return forecasts — I needed a way to order stocks under uncertainty.

That distinction sounds minor. It isn’t.

This post is the first in a series of exploratory notes meant to reinterpret an old z-ranking–based stock selection framework using modern mathematical ideas: order statistics, ranking geometry, stability, and noise.

There is no code here. No backtests. No claims of optimality.

This is about conceptual foundations — and about understanding *why* certain simple systems keep reappearing across finance, language, and decision-making, even when theory lags behind practice.

The object is not returns — it is order

Let's begin with a simple observation.

In most equity selection systems, the decision is not:

> "What return will this stock generate?"

The decision is:

> "Which stocks should be at the top of the list?"

This is subtle but fundamental.

If you ultimately:

- rank stocks,
- take the top-k,
- and ignore the rest,

then the *ordering* is the object of interest, not the predicted value.

Two models that produce very different numerical forecasts can induce **the same ordering**. Conversely, two models with similar average accuracy can produce **radically different top-k selections**.

This means that much of classical statistics — which is optimized for average error — is misaligned with the actual decision being made.

Normalization as a statement of invariance

My original system relied heavily on **z-scores**.

Given raw indicators like:

- PE
- EV/EBITDA
- Debt/EBITDA
- EPS growth

I would normalize them, typically **within industry**, before combining them into a total score.

This wasn't theory-driven at the time. It was necessity-driven.

But in hindsight, normalization is not just preprocessing.
It is a **theoretical statement**:

> "Comparisons should be invariant to industry level effects."

Industry normalization says:

- banks should be compared to banks,

- software companies to software companies,
- absolute levels are less informative than *relative position among peers*.

Mathematically, this removes dominant between-group variance and focuses attention on **within-group dispersion**, which is where ranking pressure actually lives.

This step already places the system closer to **order statistics** than to regression.

Linear scores are projections, not models of reality

After normalization, indicators are combined linearly:

$Score_i=a_1Z_{i,1}+a_2Z_{i,2}+\cdots+a_dZ_{i,d}$

This looks like regression. It isn't.

The weights a are not parameters of a data-generating process.
They are **coordinates of a projection**.

Geometrically:

- each stock is a point in d-dimensional factor space,
- the weight vector defines a direction,
- projecting onto that direction induces an ordering.

Crucially:

- scaling the weights does nothing to the ordering,
- many different directions produce the same ranking,
- small rotations often leave the top-k unchanged.

This already tells us something important:

> **Coefficient estimation is ill-posed when the objective is ranking.**

Why regression feels unsatisfying (and why that feeling is correct)

It is tempting to estimate weights using linear regression.

After all, regression gives us coefficients.
But regression solves a different problem.

OLS minimizes **mean squared error**:

- it cares about fitting everyone,
- it treats large middle errors as important,
- it has no notion of rank inversion.

Two weight vectors with nearly identical MSE can:

- disagree dramatically on the top 5 stocks.

So when regression "fails" in stock selection, it is not underperforming.
It is simply answering the wrong question.

Ranking is governed by tails, not averages

This brings us to a key conceptual shift.

Classical statistics is about **typical behavior**:

- means,
- variances,
- convergence.

Ranking systems live in a different world:

- rare winners,
- heavy tails,
- sparse outcomes.

When you select the top-k, you are working with **order statistics** — the extremes of a distribution — not its center.

Order statistics are:

- highly sensitive to noise,
- dominated by tail behavior,
- unstable under small perturbations.

This explains why:

- stability matters more than precision,
- simple heuristics outperform “optimal” models,
- over-optimization consistently disappoints.

Why clustering feels tempting — and why it usually misleads

A natural question arises:

> If industries and countries are crude labels, shouldn’t we let the data discover its own structure via clustering?

Clustering answers:

> “Which stocks are similar?”

Ranking asks:

> “Which stocks are extreme?”

These are different questions.

On absolute financial metrics, clustering almost always rediscoveres **industry structure** — because industry level differences dominate geometry.

After industry normalization, clustering often:

- becomes unstable,
- discretizes what is actually a smooth gradient,

- introduces hard boundaries where none exist.

Apparent “winner clusters” often turn out to be nothing more than:

> a continuous ranking sliced into bins.

This is not discovery. It is quantization.

Clustering can be a useful **diagnostic lens** — but it is rarely a good backbone for tail-driven selection systems.

Country is not a statistical partition — it is a decision constraint

Industry normalization is about **comparability**.

Country segmentation is different.

Country grouping reflects:

- market structure,
- capital flows,
- implementation constraints,
- regime exposure.

Seen this way:

- industry is epistemic (what can be compared),
- ranking is statistical (who is extreme),
- country is operational (how decisions are expressed).

This framing explains why country steps feel “bold”: they are not statistical necessities, but **allocation grammar**.

Learning weights by Monte Carlo, not optimization

Eventually, I tried something simple.

Instead of *estimating* weights, I **sampled** them.

Procedure:

1. Draw random weight vectors.
2. Normalize them to unit length (remove scale).
3. Rank stocks by the induced score.
4. Compare the ranking to realized outcomes.
5. Keep weight directions that are “good enough”.

What emerged was surprising — and clarifying.

There was never a single best weight vector.
There was a **region of directions** that all produced similar rankings.

This is not a failure.
It is the correct mathematical picture.

Ranking invariance and the geometry of solutions

Because ordering is invariant to scaling, what matters is not the vector a, but its **direction**.

Normalizing weights to unit norm collapses all equivalent scalings into a single point on the unit sphere.

Each correct pairwise ordering imposes a half-space constraint:

$a^\top(Z_i - Z_j) > 0$

The set of acceptable weights becomes:

- an intersection of half-spaces,
- a cone of directions,
- a feasible region, not a point.

Monte Carlo sampling doesn't hide this — it **reveals it**.

Multiple solutions are not ambiguity; they are **robustness**.

Why "many solutions work" is the real result

When multiple weight directions produce the same ordering, the correct conclusion is not:

> "The model is underdetermined."

It is:

> "The ordering is stable under a family of projections."

This reframes the problem:

- weights are not truths,
- they are coordinates,
- their ranges matter more than their point estimates,
- sign stability matters more than magnitude.

This also explains why expert constraints emerge naturally:

- some weights must be positive,
- some must be negative,
- some can trade off.

This is not bias.
It is **learning the shape of the solution space**.

Why this perspective generalizes beyond finance

Once seen this way, the structure appears elsewhere:

- Language generation: selecting words that jointly produce meaning.
- Scientific research: choosing hypotheses under sparse validation.
- Venture capital: ranking opportunities under extreme uncertainty.

- Recommendation systems: surfacing top items, not predicting ratings.

All share the same pattern:

- outcomes are rare,
- rewards are asymmetric,
- evaluation is delayed,
- ordering matters more than prediction.

Finance simply makes these constraints unavoidable.

What this series will (and will not) do

This series will not:

- propose a new alpha model,
- optimize hyperparameters,
- claim superiority.

It will:

- explore ranking as a mathematical object,
- connect practice to theory,
- explain why simple systems persist,
- and show where modern tools help — and where they don't.

Later posts will cover:

- order statistics and tails,
- stability vs accuracy,
- why backtests lie even when honest,
- and how Monte Carlo exploration clarifies uncertainty.

Closing thought

I didn't start with theory.
I started with selection.

Only later did I realize that what I had built — imperfectly and heuristically — was not a predictive model at all, but a **framework for navigating order under noise**.

The mathematics came later, not to replace the system, but to explain why it behaved the way it did.

This notes series is an attempt to make that explanation explicit — for myself first, and for others who have felt the same friction between elegant models and stubborn reality.