Backtesting I

Marcos López de Prado, Ph.D.

Advances in Financial Machine Learning

ORIE 5256

What are we going to learn today?

- Bet Sizing
  - Using Predicted Probabilities
  - Averaging Bets and Size Discretization
- The Dangers of Backtesting
  - Mission Impossible: The Flawless Backtest
  - Even Flawless Backtests are Usually Wrong
  - Some General Recommendations
- Backtesting Methods
  - Walk Forward
  - Cross-Validation
  - Combinatorial Purged Cross-Validation
  - Backtesting on Synthetic Data

Bet Sizing

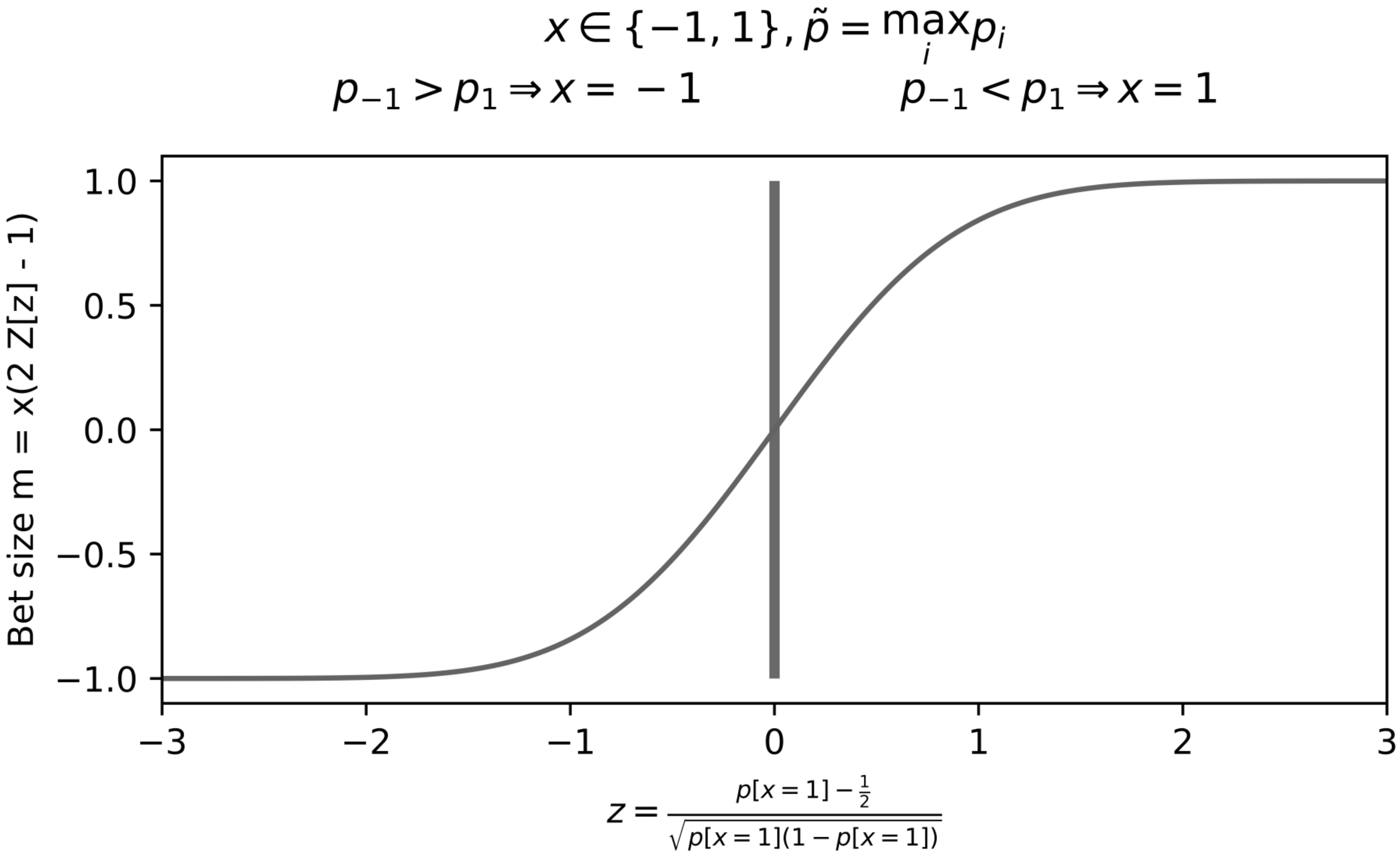
Using Predicted Probabilities (1/3)

- Let us denote $p[x]$ the probability that label x takes place. We would like to use this predicted probability to derive the bet size.
- For two possible outcomes, $x \in \{-1,1\}$, we test the null hypothesis $H_0 \colon p[x = 1] = \frac{1}{2}$.
- We compute the test statistic

$$z = \frac{p[x = 1] - \frac{1}{2}}{\sqrt{p[x = 1](1 - p[x = 1])}} = \frac{2p[x = 1] - 1}{2\sqrt{p[x = 1](1 - p[x = 1])}} \sim Z$$

  with $z \in (-\infty, +\infty)$ and where Z represents the standard Normal distribution.
- We derive the bet size as $m = 2Z[z] - 1$, where $m \in [-1,1]$ and $Z[.]$ is the CDF of Z.

Using Predicted Probabilities (2/3)



Using Predicted Probabilities (3/3)

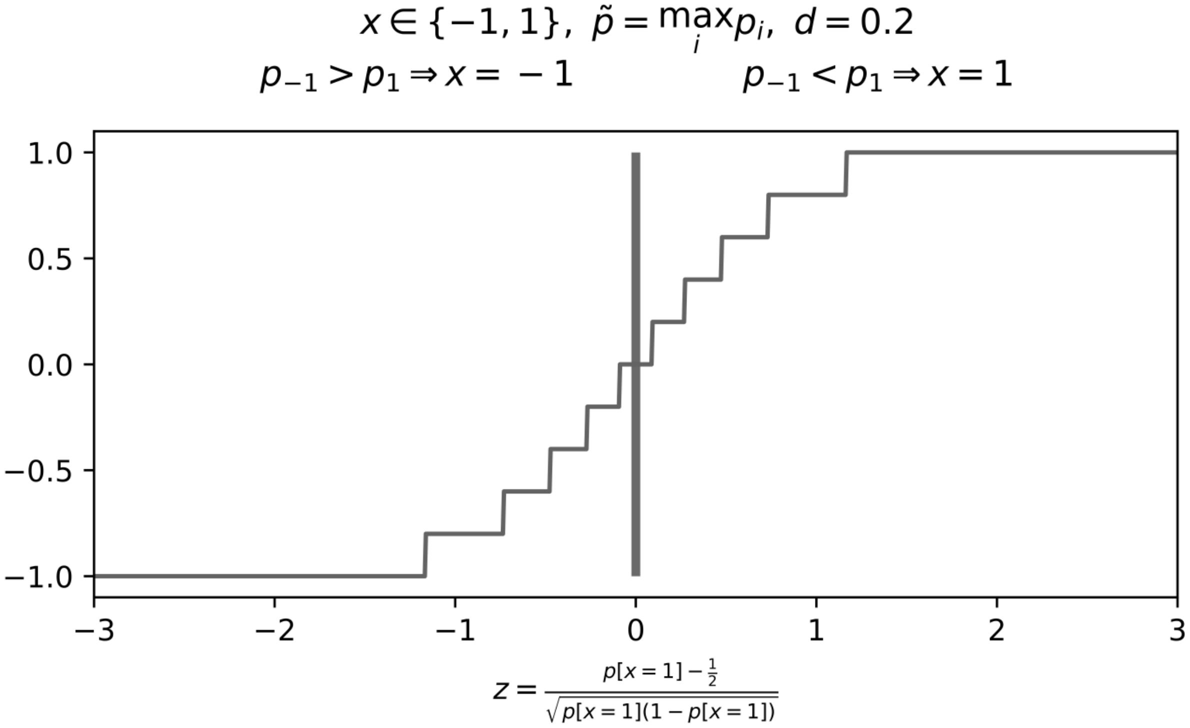
- For more than two possible outcomes, we follow a **one-versus-all method**.
- Let $X = \{-1, \ldots, 0, \ldots, 1\}$ be various labels associated with bet sizes, and $x \in X$ the predicted label. In other words, the label is identified by the bet size associated with it.
- For each label $i = 1, \ldots, \|X\|$, we estimate a probability p_i, with $\sum_{i=1}^{\|X\|} p_i = 1$.
- We define $\tilde{p} = \max_i\{p_i\}$ as the probability of *x*, and we would like to test for $H_0 : \tilde{p} = \frac{1}{\|X\|}$.
- We compute the test statistic $z = \frac{\tilde{p} - \frac{1}{\|X\|}}{\sqrt{\tilde{p}(1-\tilde{p})}} \sim Z$, with $z \in [0, +\infty)$. We derive the bet size as $m = x \underbrace{(2Z[z] - 1)}_{\in[0,1]}$, where $m \in [-1,1]$ and $Z[z]$ regulates the size for a prediction x (where the side is implied by *x*). Uncertainty is absolute when all outcomes are equally likely.

Averaging Bets

- Every bet is associated with a holding period, spanning from the time it originated to the time the first barrier is touched, `t1` (see Chapter 3).
- One possible approach is to override an old bet as a new bet arrives; however, that is likely to lead to excessive turnover.
- A more sensible approach is to average all sizes across all bets still active at a given point in time.

Size Discretization

- Averaging reduces some of the excess turnover, but still it is likely that small trades will be triggered with every prediction.



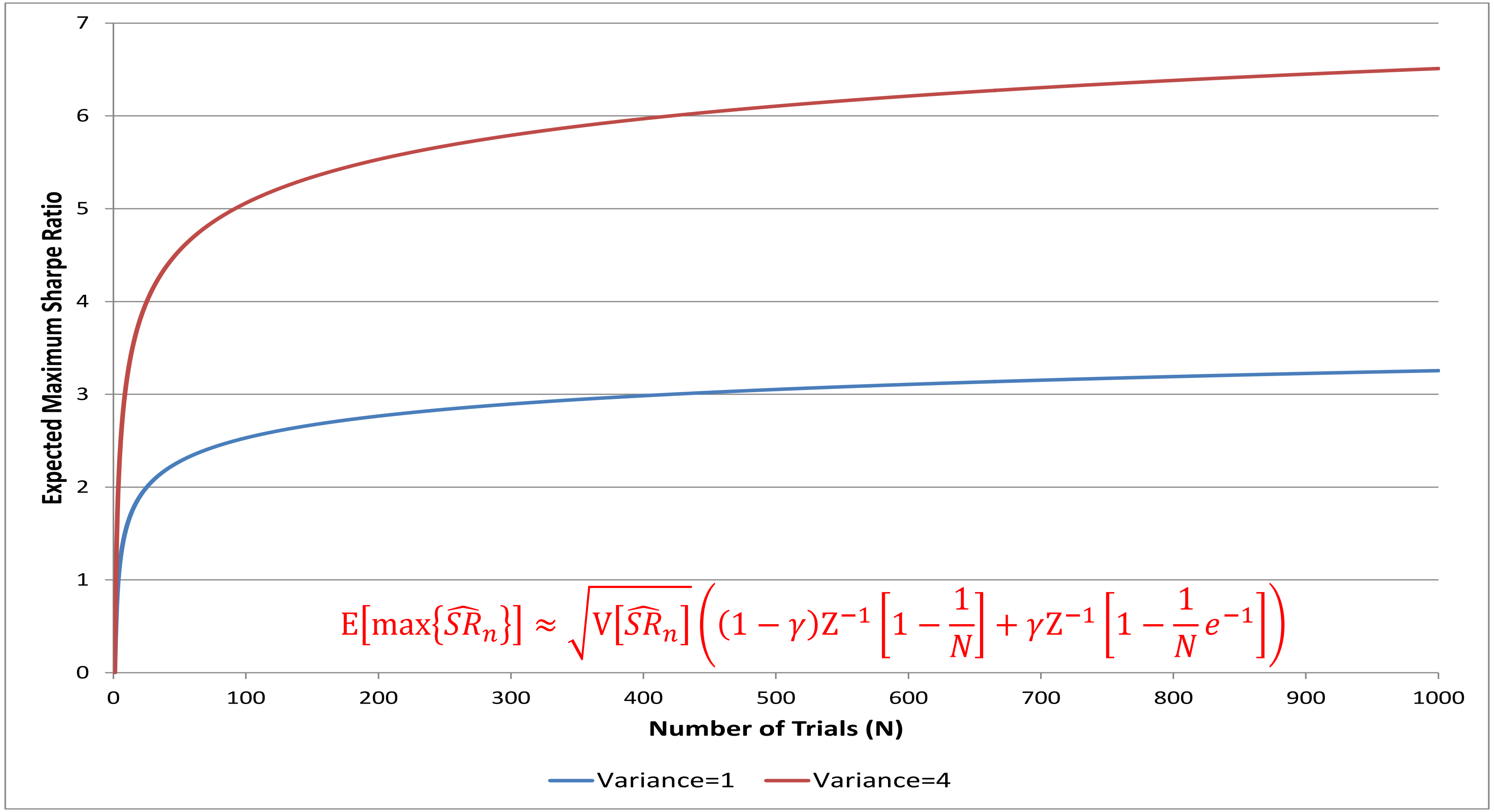
As this jitter would cause unnecessary overtrading, we can discretize the bet size as $m^* = \text{round}\left[\frac{m}{d}\right] d$, where $d \in (0, 1]$ determines the degree of discretization.

The Dangers of Backtesting

Mission Impossible: The Flawless Backtest

- Common backtesting errors include:
  - **Survivorship bias:** Using as investment universe the current one, hence ignoring that some companies went bankrupt and securities were delisted along the way.
  - **Look-ahead bias:** Using information that was not public at the moment the simulated decision would have been made. Be certain about the timestamp for each data point. Take into account release dates, distribution delays, and backfill corrections.
  - **Storytelling:** Making up a story *ex-post* to justify some random pattern.
  - **Data mining and data snooping:** Training the model on the testing set.
  - **Transaction costs:** Simulating transaction costs is hard because the only way to be certain about that cost would have been to interact with the trading book (i.e., to do the actual trade).
  - **Outliers:** Basing a strategy on a few extreme outcomes that may never happen again as observed in the past.
  - **Shorting:** Taking a short position on cash products requires finding a lender. The cost of lending and the amount available is generally unknown, and depends on relations, inventory, relative demand, etc.

Even Flawless Backtests are Usually Wrong



$$\mathrm{E}\left[\max\left\{\widehat{SR}_n\right\}\right] \approx \sqrt{\mathrm{V}\left[\widehat{SR}_n\right]}\left((1-\gamma)\mathrm{Z}^{-1}\left[1-\frac{1}{N}\right]+\gamma\mathrm{Z}^{-1}\left[1-\frac{1}{N}e^{-1}\right]\right)$$

Expected Maximum Sharpe Ratio as the number of independent trials *N* grows, for $\mathrm{E}\left[\widehat{SR}_n\right] = 0$ and $\mathrm{V}\left[\widehat{SR}_n\right] \in \{1,4\}$.

Data Dredging: Searching for empirical findings regardless of their theoretical basis is likely to magnify the problem, as $\mathrm{V}\left[\widehat{SR}_n\right]$ will increase when unrestrained by theory.

We must always control for *N* and $\mathbf{V}\left[\widehat{\boldsymbol{SR}}_{\boldsymbol{n}}\right]$.

This is a consequence of pure random behavior. We will find high SR strategies *even if there is no investment skill associated with this strategy class* ($\mathrm{E}\left[\widehat{SR}_n\right] = 0$).

Some General Recommendations

- While there is no easy way to prevent overfitting, a number of steps can help reduce its presence.

1. **Develop models for entire asset classes** or investment universes, rather than for specific securities (Chapter 8). Investors diversify, hence they do not make mistake *X* only on security *Y*. If you find mistake *X* only on security *Y*, no matter how apparently profitable, it is likely a false discovery.
2. **Apply bagging** (Chapter 6) as a means to both prevent overfitting and reduce the variance of the forecasting error. If bagging deteriorates the performance of a strategy, it was likely overfit to a small number of observations or outliers.
3. **Do not backtest until all your research is complete** (Chapters 1–10).
4. **Record every backtest conducted on a dataset** so that the probability of backtest overfitting may be estimated on the final selected result (see Bailey, Borwein, López de Prado and Zhu [2017a] and Chapter 14), and the Sharpe ratio may be properly deflated by the number of trials carried out (Bailey and López de Prado [2014b]).
5. **Simulate scenarios rather than history** (Chapter 12). A standard backtest is a historical simulation, which can be easily overfit. History is just the random path that was realized, and it could have been entirely different. Your strategy should be profitable under a wide range of scenarios, not just the anecdotal historical path. It is harder to overfit the outcome of thousands of "what if" scenarios.
6. **Do not research under the influence of a backtest**. If the backtest fails to identify a profitable strategy, start from scratch. Resist the temptation of reusing those results.

Backtesting Methods

Walk Forward (1/2)

- WF is a historical simulation of how the strategy would have performed in past.
  - Each strategy decision is based on observations that predate that decision.
- WF enjoys two key advantages:
  - WF has a clear historical interpretation. Its performance can be reconciled with paper trading.
  - History is a filtration; hence, using trailing data guarantees that the testing set is out-of-sample (no leakage), as long as purging has been properly implemented (see Chapter 7, section 7.4.1).
- A common mistake found in WF backtests is **leakage caused by improper purging**, where `t1.index` falls within the training set, but `t1.values` fall within the testing set (see Chapter 3).
- **Embargoing** is not needed in WF backtests, because the training set always predates the testing set.

Walk Forward (2/2)

- WF suffers from three major disadvantages:
  - A single scenario is tested (the historical path), which can be easily overfit (Bailey et al. [2014]).
  - WF is not necessarily representative of future performance, as results can be biased by the particular sequence of datapoints. The truth is, it is as easy to overfit a walk-forward backtest as to overfit a walk-backward backtest, and the fact that changing the sequence of observations yields inconsistent outcomes is evidence of that overfitting.
  - The initial decisions are made on a smaller portion of the total sample. Even if a warm-up period is set, most of the information is used by only a small portion of the decisions.

Cross-Validation (1/2)

- The goal of backtesting through cross-validation (CV) is not to derive historically accurate performance, but to infer future performance from a number of out-of-sample scenarios. For each period of the backtest, we simulate the performance of a classifier that knew everything except for that period.
- **Advantages**
  1. The test is not the result of a particular sequence. Some outcomes are the result of walking forward, other outcomes result from walking backwards, and other outcomes mix training data from before and after the testing set. Reversing the sequence should not lead to an entirely different CV result.
  2. Every decision is made on sets of equal size. This makes outcomes comparable across periods, in terms of the amount of information used to make those decisions.
  3. Every observation is part of one and only one testing set. There is no warm-up subset, thereby achieving the longest possible out-of-sample simulation.

Cross-Validation (2/2)

- **Disadvantages**
  1. Like WF, a single backtest path is simulated (although not the historical one). There is one and only one forecast generated per observation.
  2. CV has no clear historical interpretation. The output does not simulate how the strategy would have performed in the past, but how it may perform *in the future* under various stress scenarios (a useful result in its own right).
  3. Because the training set does not trail the testing set, leakage is possible. Extreme care must be taken to avoid leaking testing information into the training set. See Chapter 7 for a discussion on how purging and embargoing can help prevent informational leakage in the context of CV.

Combinatorial Purged Cross-Validation (1/2)

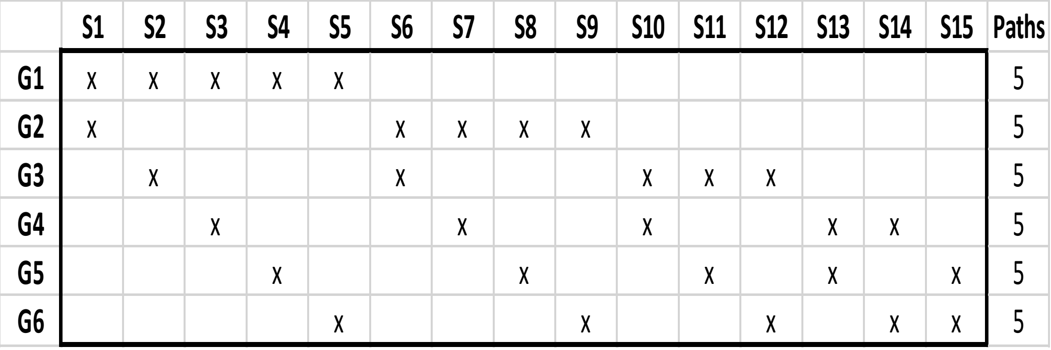
- CPCV addresses the main drawback of the WF and CV methods, namely that those schemes test a single path.
- Consider *T* observations partitioned into *N* groups without shuffling, where groups $n = 1, \dots, N-1$ are of size $\lfloor T/N \rfloor$, the *N*th group is of size $T - \lfloor T/N \rfloor (N-1)$, and $\lfloor . \rfloor$ is the floor or integer function. For a testing set of size *k* groups, the number of possible training/testing splits is

$$\binom{N}{N-k} = \frac{\prod_{i=0}^{k-1}(N-i)}{k!}$$

- Since each combination involves *k* tested groups, the total number of tested groups is $k\binom{N}{N-k}$. And since we have computed all possible combinations, these tested groups are uniformly distributed across all *N* (each group belongs to the same number of training and testing sets). The implication is that from *k*-sized testing sets on *N* groups we can backtest a total number of paths $\varphi[N,k]$,

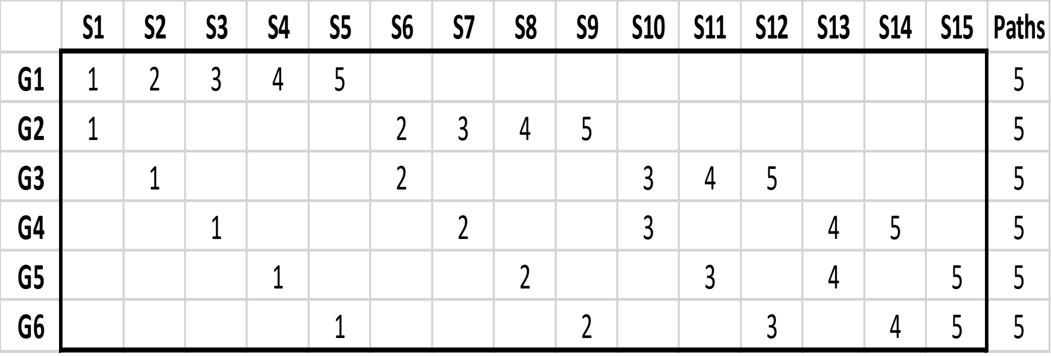
$$\varphi[N,k] = \frac{k}{N}\binom{N}{N-k} = \frac{\prod_{i=1}^{k-1}(N-i)}{(k-1)!}$$

Combinatorial Purged Cross-Validation (2/2)



	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	Paths
G1	x	x	x	x	x											5
G2	x					x	x	x	x							5
G3		x				x				x	x	x				5
G4			x				x			x			x	x		5
G5				x				x			x		x		x	5
G6					x				x			x		x	x	5

Composition of train/test splits for $N = 6$ and $k = 2$. There are $\binom{6}{4} = 15$ splits, indexed as *S1,…,S15*. For each split, the figure marks with a cross (*x*) the groups included in the testing set, and leaves unmarked the groups that form the training set. Each group forms part of $\varphi[6,2] = 5$ testing sets, therefore this train/test split scheme allows us to compute 5 backtest paths.



	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	Paths
G1	1	2	3	4	5											5
G2	1					2	3	4	5							5
G3		1				2				3	4	5				5
G4			1				2			3			4	5		5
G5				1				2			3		4		5	5
G6					1				2			3		4	5	5

Assignment of each tested group to one backtest path. For example, path 1 is the result of combining the forecasts from $(G1, S1)$, $(G2, S1)$, $(G3, S2)$, $(G4, S3)$, $(G5, S4)$ and $(G6, S5)$. Path 2 is the result of combining forecasts from $(G1, S2)$, $(G2, S6)$, $(G3, S6)$, $(G4, S7)$, $(G5, S8)$ and $(G6, S9)$, and so on.

- In the example above we have generated only 5 paths, however CPCV allows us to generate thousands of paths on a sufficiently long series. The number of paths $\varphi[N, k]$ increases with $N \rightarrow T$ and with $k \rightarrow N/2$.
- A key advantage of CPCV is that it allows us to derive a distribution of Sharpe ratios, as opposed to a single (likely overfit) Sharpe ratio estimate.

Backtesting on Synthetic Data (1/2)

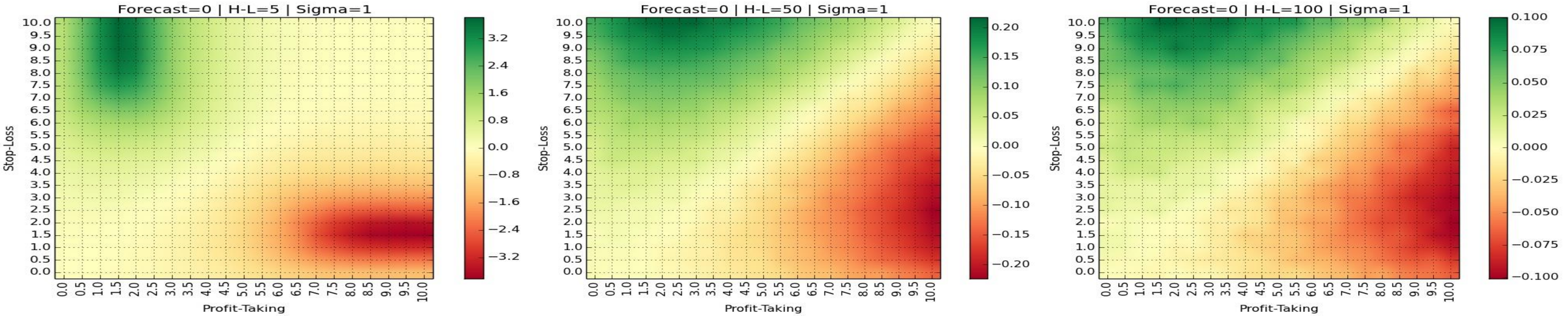
- A key problem in finance is that datasets are limited. All we have is the past.
- Alternatively, we could generate synthetic datasets, representative of a particular phenomenon that we wish to study. For example:
  1. Estimate the input parameters $\{\sigma, \varphi\}$ from
  $$P_{i,t} = E_0\left[P_{i,T_i}\right] + \varphi\left(P_{i,t-1} - E_0\left[P_{i,T_i}\right]\right) + \xi_t$$
  2. Construct a mesh of stop-loss and profit-taking pairs, $\left(\underline{\pi_i}, \overline{\pi_i}\right)$.
  3. Generate a large number of paths (e.g., 100,000) for $\pi_{i,t}$ applying our estimates $\{\hat{\sigma}, \hat{\varphi}\}$.
  4. Apply the 100,000 paths generated in Step 3 on each node of the mesh $\left(\underline{\pi_i}, \overline{\pi_i}\right)$ generated in Step 2.
     - For each node, we apply the stop-loss and profit-taking logic, giving us 100,000 values of π_{i,T_i}.
     - For each node we compute the Sharpe ratio associated with that trading rule.

Backtesting on Synthetic Data (2/2)

5. Compute the solution:

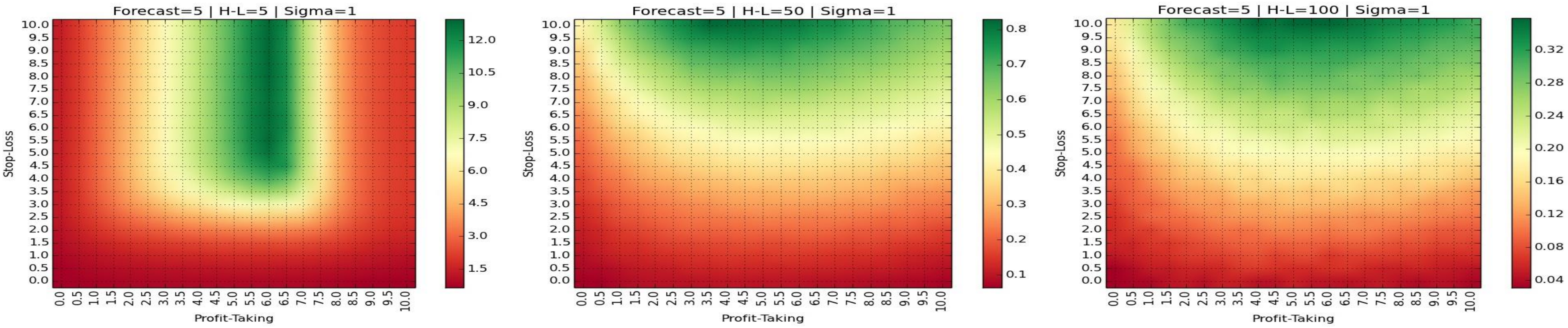
   - We determine the pair $\left(\underline{\pi_i}, \overline{\pi_i}\right)$ within the mesh of trading rules that is optimal, given the input parameters $\{\hat{\sigma}, \hat{\varphi}\}$ and the observed initial conditions $\left\{P_{i,0}, E_0\left[P_{i,T_i}\right]\right\}$.
   - If strategy *S* provides a profit target $\overline{\pi_i}$ for a particular opportunity *i*, we can use that information in conjunction with the results in Step 4 to determine the optimal stop-loss, $\underline{\pi_i}$.
   - If the trader has a maximum stop-loss $\underline{\pi_i}$ imposed by the fund's management, we can use that information in conjunction with the results in Step 4 to determine the optimal profit taking $\overline{\pi_i}$ within the range of stop-losses $\left[0, \underline{\pi_i}\right]$.

Cases with Zero Long-Run Equilibrium



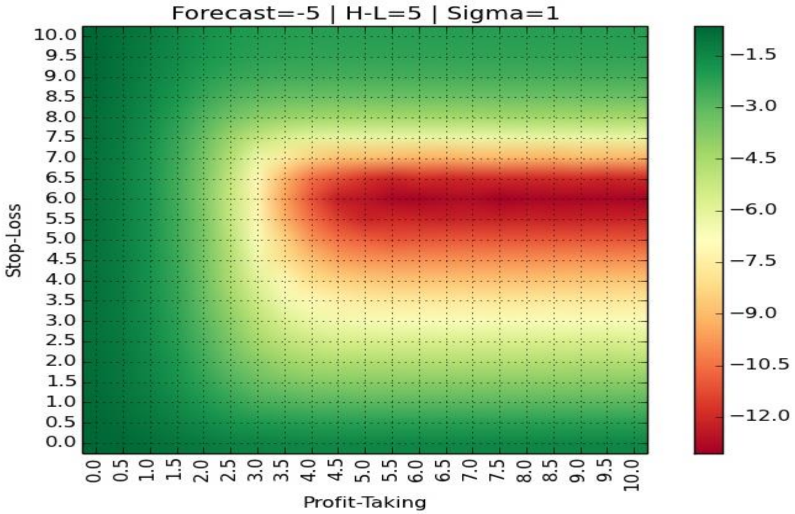
- From left to right, these figures show the Sharpe ratios for parameter combinations $\{\mu, \tau, \sigma\} = \{0,5,1\}$, $\{\mu, \tau, \sigma\} = \{0,50,1\}$, $\{\mu, \tau, \sigma\} = \{0,100,1\}$.
- For small half-life, performance is maximized in a narrow range of combinations of small profit-taking with large stop-losses: The optimal trading rule is to hold an inventory long enough until a small profit arises, even at the expense of experiencing 5 or 7-fold losses.
- **This is in fact what many market-makers do in practice, and is consistent with the “asymmetric payoff dilemma”** described in Easley et al. [2011].
- The worst possible trading rule in this setting would be to combine a short stop-loss with large profit-taking threshold, a situation that market-makers avoid in practice.

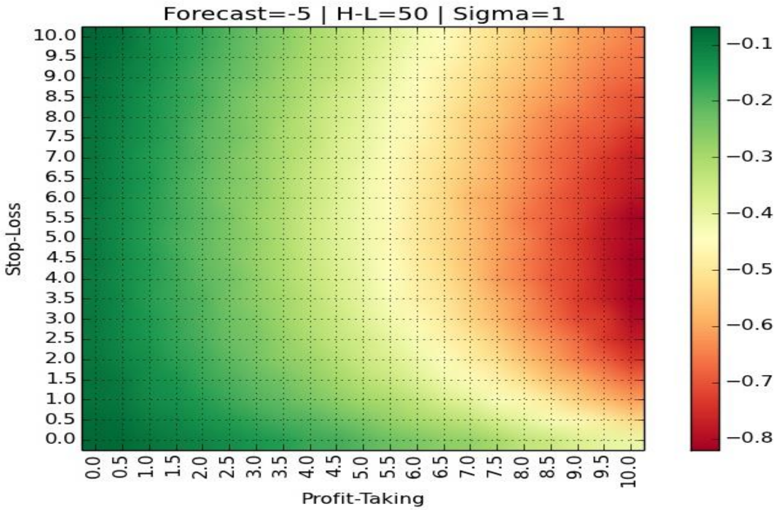
Cases with Positive Long-Run Equilibrium

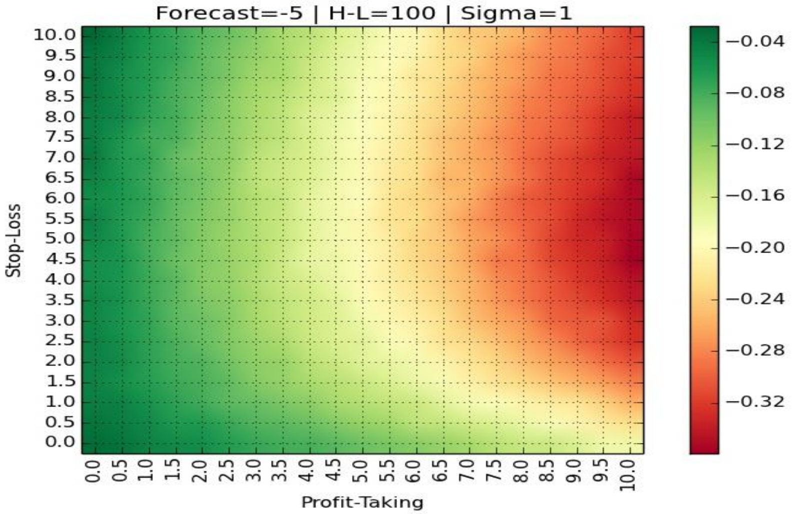


- From left to right, these figures show the Sharpe ratios for parameter combinations $\{\mu, \tau, \sigma\} = \{5,5,1\}$, $\{\mu, \tau, \sigma\} = \{5,50,1\}$, $\{\mu, \tau, \sigma\} = \{5,100,1\}$.
- Because positions tend to make money, the optimal profit-taking is higher than in the previous cases.
- As we increase the half-life, the range of optimal profit-taking widens, while the range of optimal stop-losses narrows, shaping the initial rectangular area closer to a square and then a semi-circle.
- Again, a larger half-life brings the process closer to a random walk.

Cases with Negative Long-Run Equilibrium

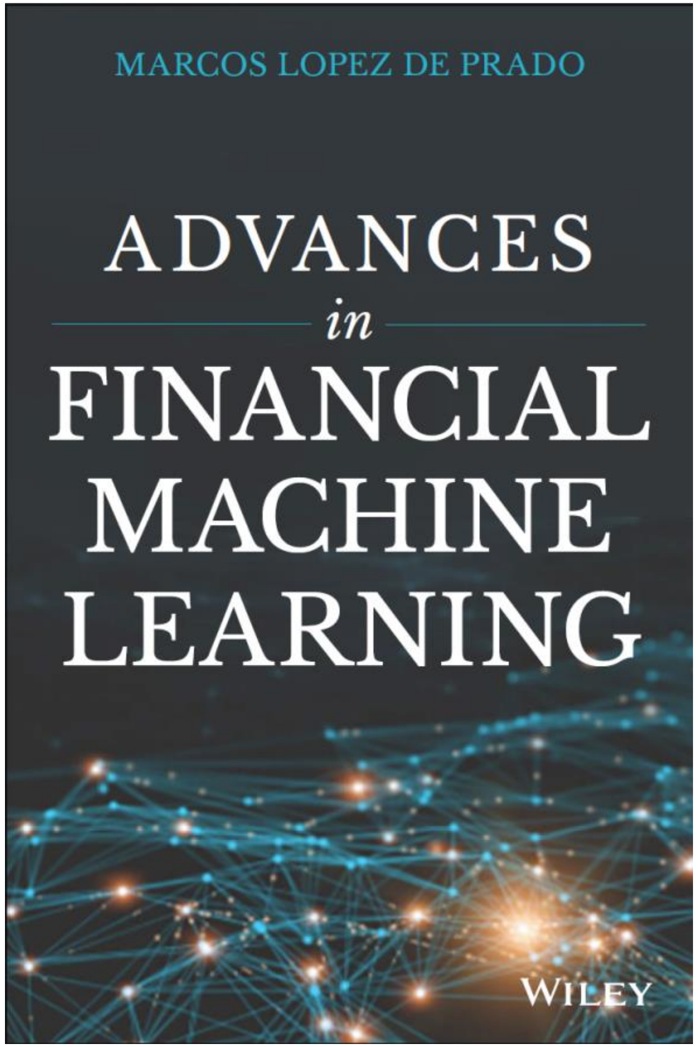






- From left to right, these figures show the Sharpe ratios for parameter combinations $\{\mu, \tau, \sigma\} = \{-5,5,1\}$, $\{\mu, \tau, \sigma\} = \{-5,50,1\}$, $\{\mu, \tau, \sigma\} = \{-5,100,1\}$.
- Results appear to be rotated complementaries of what we obtained in the Positive case (like rotated photographic negatives).
- The reason is, that the profit in the previous figures translates into a loss in these figures, and vice versa: One case is an image of the other, just as a gambler’s loss is the house’s gain.

For Additional Details



The first wave of quantitative innovation in finance was led by Markowitz optimization. Machine Learning is the second wave and it will touch every aspect of finance. López de Prado's Advances in Financial Machine Learning is essential for readers who want to be ahead of the technology rather than being replaced by it.
— Prof. **Campbell Harvey**, Duke University. Former President of the American Finance Association.

Financial problems require very distinct machine learning solutions. Dr. López de Prado's book is the first one to characterize what makes standard machine learning tools fail when applied to the field of finance, and the first one to provide practical solutions to unique challenges faced by asset managers. Everyone who wants to understand the future of finance should read this book.
— Prof. **Frank Fabozzi**, EDHEC Business School. Editor of The Journal of Portfolio Management.

THANKS FOR YOUR ATTENTION!

Disclaimer

- The views expressed in this document are the authors' and do not necessarily reflect those of the organizations he is affiliated with.
- No investment decision or particular course of action is recommended by this presentation.
-