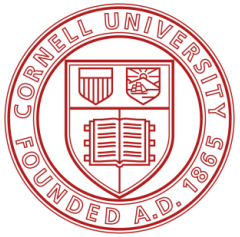


Three Machine Learning Solutions to the Bias-Variance Dilemma

Prof. Marcos López de Prado
Advances in Financial Machine Learning
ORIE 5256

Background

- The error of a supervised learning algorithm has three components:
  - **Bias**: error caused by erroneous assumptions
  - **Variance**: error caused by overfitting
  - **Noise**: irreducible error, due to the random nature of the variable
- A central problem in classical statistics (e.g., Econometrics) is to find the MVUE
  - Fit the estimator with **m**inimum **v**ariance among all **u**nbiased **e**stimators
- A central problem in Machine Learning (ML) is to minimize overall errors
  - Capture the regularities in the data (low bias), while
  - generalizing well to unseen data (low variance)
- This is a key difference between Econometrics and ML
  - Econometric models are willing to sacrifice performance in exchange for zero bias
  - ML models find a balance between bias and variance, in order to maximize performance
- In this seminar, we will study how ML finds this balance

The Problem

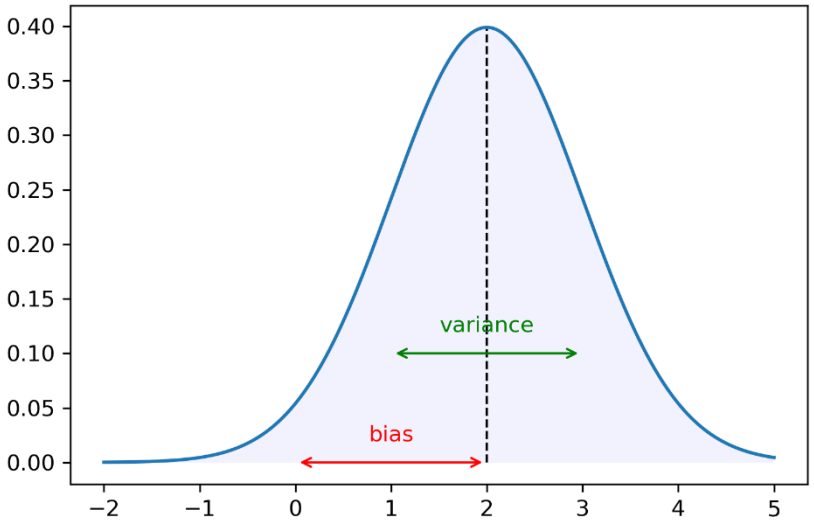
Error Decomposition

- Consider N observations, $\{(x_n, y_n)\}_{n=1,\ldots,N}$, and a function $f[x_n]$ that predicts y_n, such that errors $\varepsilon_n = y_n - f[x_n]$ are unpredictable, with $\varepsilon_n \sim N[0, \sigma_\varepsilon^2]$
- This "true model" $f[x_n]$ is unknown to us, so instead we propose a statistical model $\hat{f}[x_n]$ that approximates $f[x_n]$, with error $e_n = y_n - \hat{f}[x_n]$
- In fitting the statistical model, we would like to minimize the mean squared *true* error $\mathrm{E}\left[\left(f[x_n] - \hat{f}[x_n]\right)^2\right]$. But because $f[x_n]$ is unknown to us, the best we can do is to minimize the mean squared *empirical* error (MSE) associated with $\hat{f}[x_n]$,

$$MSE = \mathrm{E}[e_n^2] = \underbrace{\left(\mathrm{E}\left[f[x_n] - \hat{f}[x_n]\right]\right)^2}_{bias^2} + \underbrace{\mathrm{V}\left[\hat{f}[x_n]\right]}_{variance} + \underbrace{\sigma_\varepsilon^2}_{noise}$$

Visualizing Bias and Variance

- We generate 10,000 trial datasets, $\{(x_n, y_n)\}_{n=1,\ldots,N}$, where $N = 1{,}000$
- We generate one out-of-sample (OOS) pair,$\{(\bar{x}, \bar{y})\}$
- For each trial dataset, we
  - fit the statistical model, $\hat{f}[x_n]$
  - compute the true error, $\epsilon = f[\bar{x}] - \hat{f}[\bar{x}]$
    - Note: The true error does not incorporate noise, because we use $f[\bar{x}]$ instead of $\bar{y}$
- Then,
  - the estimated bias is $\mathrm{E}[\epsilon]$ across all trials
  - the estimated variance is $\mathrm{V}[\epsilon]$ across all trials



Monte Carlo experiments help us visualize the bias and variance associated with a particular statistical model, $\hat{f}[x_n]$.

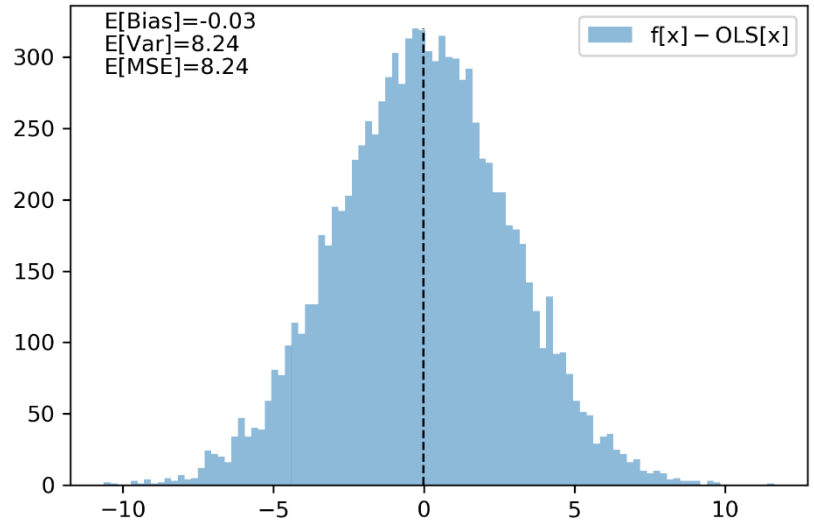
Gauss-Markov Theorem (GMT)

- GMT assumes that:
  - $f[x_n]$ is linear
  - $\hat{f}[x_n]$ incorporates all k variables
  - $\{e_n\}$ is white noise, and uncorrelated to $\{x_n\}$
- Then, the ordinary least squares (OLS) estimator has the lowest $\mathrm{V}\left[\hat{f}[x_n]\right]$ among all *linear unbiased* estimators

$$\text{bias:} = f[x_n] - \mathrm{E}\left[\hat{f}[x_n]\right] = 0$$

$$\text{variance:} = \mathrm{V}\left[\hat{f}[x_n]\right] \propto \frac{\sigma_\varepsilon^2}{N} k$$

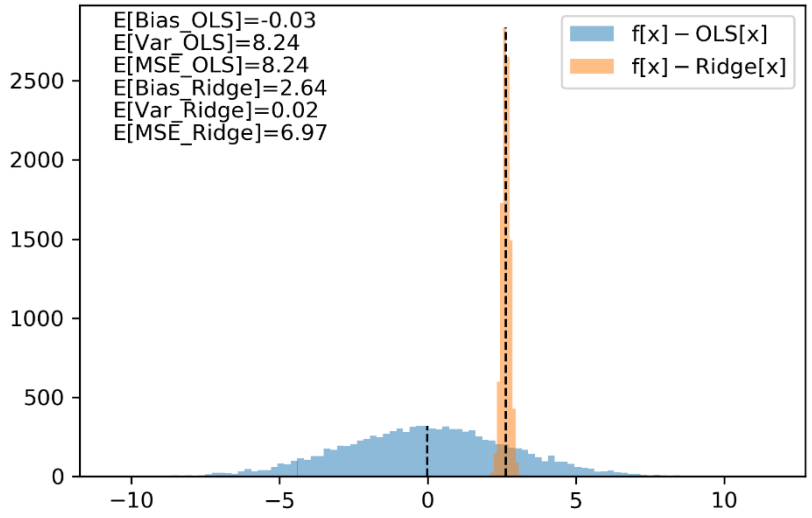
- GMT is the fundamental reason why OLS is the statistical workhorse in Econometrics



When the true model is linear, there are no omitted variables, and errors are white noise, then OLS gives us the MVUE (minimum variance estimate among all linear unbiased).

When GMT Assumptions Are True

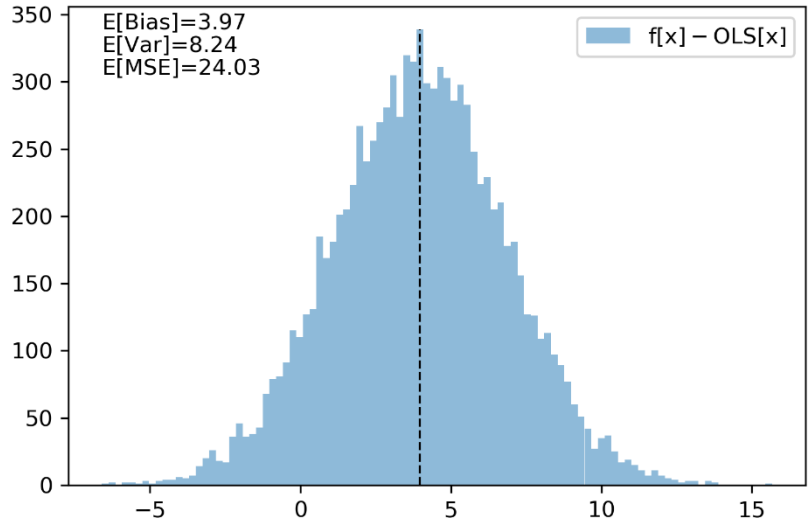
- GMT tells us that OLS has minimum variance among *linear unbiased* estimators, however
  - there are non-linear biased estimators with lower variance (e.g., James-Stein estimator)
  - there are linear biased estimators with lower MSE (e.g., Ridge, LASSO)
- **Even if all GMT assumptions are correct, OLS does not necessarily yield the minimum MSE**
- Econometrics' reliance on unbiased estimators (including, but not limited to OLS) is not justified by model performance



In the above experiment, all GMT assumptions are true, and yet OLS underperforms a biased ML estimator (Ridge), because one of the regressors is spurious. As a result of selection bias, many financial studies include spurious regressors, thus setting OLS for delivering poor predictions.

When GMT Assumptions Are False

- GMT does not apply under three common assumption violations in financial datasets:
  - **Specification errors**: The true model is not linear in its parameters, and the statistical model omits variables or incorporates spurious variables
  - **Stochastic regressors**: The explanatory variables are measured with error, are endogenous, are mutually dependent, etc.
  - **Non-spherical errors**: The errors are heteroskedastic or autocorrelated
- As a result, Econometric models often are inefficient (high bias and/or variance)
- One motivation for using ML in finance is that it relies on more realistic assumptions



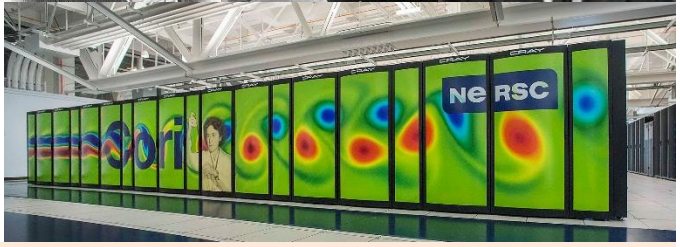
Because financial systems are extremely complex, Econometric models are often misspecified and miss variables (particularly interaction effects). In the above experiment, the OLS specification is correctly linear, however it is missing one interaction, leading to high bias and variance.

ML Solutions to the Bias-Variance Dilemma

- In finance, there is no strong theoretical reason to discard biased models
  - Financial systems are too complex for expecting fully well-specified (unbiased) models
- Accordingly, minimizing variance subject to zero bias (MVUE) is the wrong objective
- ML methods can achieve lower MSE than Econometric models through
  - Cross-validation
  - Complexity reduction
    - Regularization
    - Feature selection, dimensionality reduction
  - Ensemble methods
    - Bootstrap aggregation (Bagging)
    - Boosting
    - Stacking

The History of Empirical Research

The Future of Quantitative Finance



What underlies these three solutions is computational statistics. When Gauss, Pearson, Fisher, and others created the tools used by econometricians, they did not have access to computers. They had no choice but to rely on closed-form solutions based on strong (unrealistic) assumptions. Today, we have more practical tools to solve the bias-variance dilemma.

Solution 1:
Cross-Validation

A Different MSE Decomposition

- Recall from the Problem statement that we wish to find the statistical model ($\hat{f}$) that minimizes the mean squared *true* error, $\mathrm{E}\left[\left(f[x_n] - \hat{f}[x_n]\right)^2\right]$
- The problem is, f is unknown, and so is the true error
- Instead, we minimize the mean squared *empirical* error,

$$\mathrm{E}\left[\left(y_n - \hat{f}[x_n]\right)^2\right] = \underbrace{\mathrm{E}\left[\left(f[x_n] - \hat{f}[x_n]\right)^2\right]}_{\text{true error}} + \underbrace{\sigma_\varepsilon^2}_{\text{noise}} + \underbrace{2\mathrm{E}\left[\left(f[x_n] - \hat{f}[x_n]\right)\varepsilon_n\right]}_{\text{cross-term}}$$

- A couple of notes:
  - The noise (σ_ε^2) does not displace the minimum, because it is a constant
  - However, the cross-term ($2\mathrm{E}\left[\left(f[x_n] - \hat{f}[x_n]\right)\varepsilon_n\right]$) is not necessarily a constant, and may lead to a function $\hat{f}$ that does not minimize the true error
- We need to study the cross-term

Cross-Term Under Test Set Errors

- Consider a datapoint (x_0, y_0) that is not part of the **train set** used to fit $\hat{f}$, $(x_0, y_0) \notin \{(x_n, y_n)\}_{n=1,\dots,N}$
- In this case, $\mathrm{E}\left[\left(f[x_0] - \hat{f}[x_0]\right)\varepsilon_0\right] = 0$, because
  - f is not fitted, so it does not depend on any observations (train set or test set)
  - $\hat{f}$ depends on $\{(x_n, y_n)\}_{n=1,\dots,N}$, but it does not depend on (x_0, y_0), because (x_0, y_0) is not part of the train set
  - The value of ε_0 does depend on y_0, because $\varepsilon_0 = y_0 - f[x_0]$
- Thus,

$$\mathrm{E}\left[\left(y_0 - \hat{f}[x_0]\right)^2\right] = \mathrm{E}\left[\left(f[x_0] - \hat{f}[x_0]\right)^2\right] + \sigma_\varepsilon^2$$

- In conclusion, minimizing MSE on observations *outside the train set* (i.e., a test set) is equivalent to minimizing the mean squared true error
  - This important result is the foundation for **cross-validation**

Cross-Validation (CV)

- CV is a validation technique that assesses how the forecasts of a model generalize on unseen data (test set). The observations are rows of (X,y). A score is computed on the forecasts across multiple test sets

Type	Name	Description
Exhaustive	**Leave-one-out**	Use one observation as the test set, and the remaining observations as the train set. Out of N observations, there are N train-test splits.
	Leave-p-out	Use p observations as the test set, and the remaining observations as the train set. Out of N observations, there are $\binom{N}{p}$ train-test splits.
Non-exhaustive	**Holdout**	Randomly assign observations to the train and test sets. Caveat: There is a single run, and the result may not be representative.
	K-Fold	Split the N observations into K regular folds, and apply a leave-one-out CV on the folds. The data may or may not be shuffled prior to forming the folds.
	Combinatorial K-Fold	Split the N observations into K regular folds, and apply a leave-p-out CV on the folds. The data may or may not be shuffled prior to forming the folds.
	Monte Carlo	Train sets are generated from random subsampling of the rows of (X,y) (random sampling without replacement)

Solution 2:
Complexity Reduction

Cross-Term Under Train Set Errors

- Let us compute $\mathrm{E}\left[\left(f[x_0] - \hat{f}[x_0]\right)\varepsilon_0\right]$ when (x_0, y_0) is part of the **train set** used to fit $\hat{f}$, $(x_0, y_0) \in \{(x_n, y_n)\}_{n=1,\ldots,N}$
- Applying Stein's lemma we obtain that

$$\mathrm{E}\left[\left(f[x_0] - \hat{f}[x_0]\right)\varepsilon_0\right] = -\sigma_\varepsilon^2 \mathrm{E}\left[\frac{\partial \hat{f}[x_0]}{\partial y_0}\right]$$

- We can think of $D_0 = \mathrm{E}\left[\frac{\partial \hat{f}[x_0]}{\partial y_0}\right]$ as a measure of model complexity relative to the train set
  - As N grows, $\hat{f}$ is less sensitive to changes in y_0
  - The more complex $\hat{f}$ is, the more sensitive it becomes to changes in y_0
- Thus,

$$\mathrm{E}\left[\left(y_0 - \hat{f}[x_0]\right)^2\right] = \mathrm{E}\left[\left(f[x_0] - \hat{f}[x_0]\right)^2\right] + \sigma_\varepsilon^2(1 - 2D_0)$$

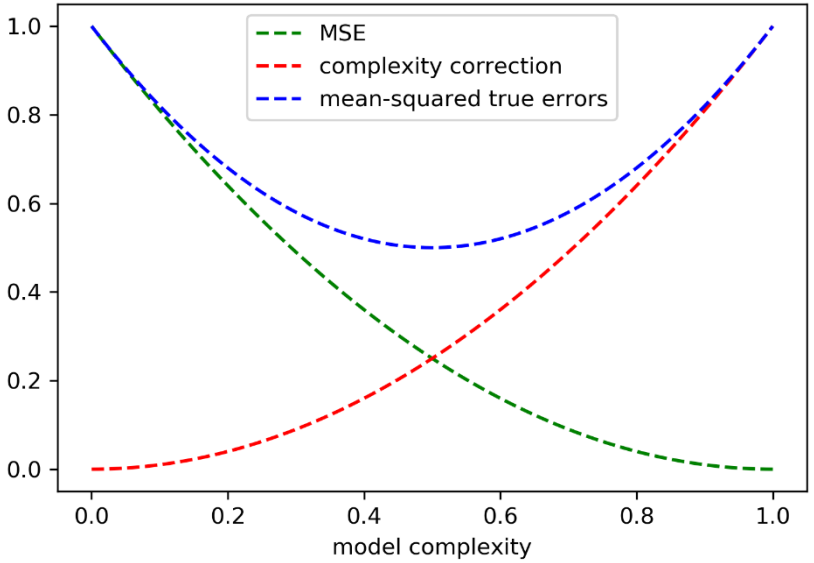
Stein's Unbiased Risk Estimator (SURE)

- In the context of train set errors,

$$\frac{1}{N}\sum_{n=1}^{N}\left(f[x_n]-\hat{f}[x_n]\right)^2 =$$

$$\underbrace{\frac{1}{N}\sum_{n=1}^{N}\left(y_n-\hat{f}[x_n]\right)^2}_{\text{MSE}} + \underbrace{\sigma_\varepsilon^2\left(\frac{2}{N}\sum_{n=1}^{N}D_n-1\right)}_{\text{complexity correction}}$$

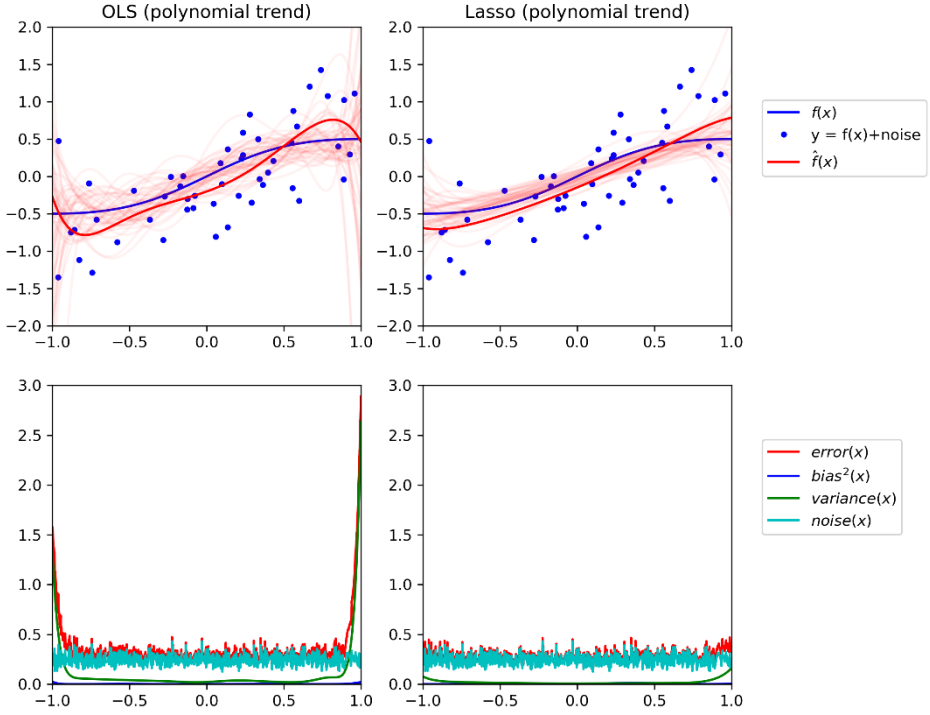
- When D_n is unknown, we can improve our estimate of $\mathrm{E}\left[\left(f[x_n]-\hat{f}[x_n]\right)^2\right]$ by adding a penalty for complexity to MSE's estimate
  - This important result is the foundation for **regularization** *and* **feature selection**



In the train set, MSE monotonically decreases with complexity, until $MSE = 0$ for a perfectly overfit model. There is an optimal level of complexity that minimizes the mean-squared true errors.

Regularization

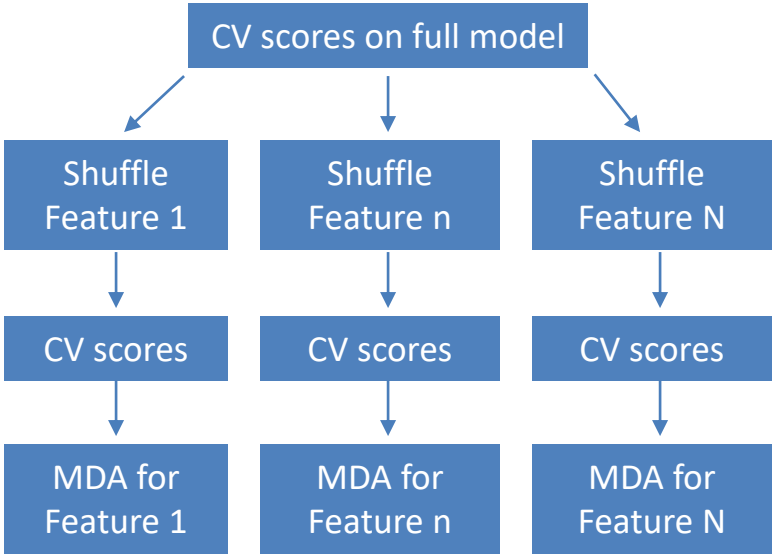
- Common forms of ML regularization include
  - **Early stopping**: The training step is stopped before convergence, in order to prevent an overly complex (overfit) model
    - E.g., place a limit to the growth of trees in a random forest
  - **Tikhonov regularization**: Also known as Ridge regression, it adds as a penalty to the objective function the L_2-norm of the vector of estimated parameters
    - E.g., penalize solutions with larger norms
  - **Sparsity regularizers**: Similarly to Tikhonov regularization, it adds as a penalty to the objective function the L_1-norm of the vector of estimated parameters
    - E.g., penalize solutions with many non-zero parameters



In the above example, LASSO (a sparsity regularizer) zeroes the estimated coefficients for unnecessary regressors, thus achieving a lower MSE than OLS.

Feature Selection

- Another way to reduce the complexity of a model is by eliminating unnecessary variables
- Feature selection is a difficult task for classical statistical models
  - An important variable will be discarded under the wrong specification
  - An unimportant variable will be selected under the wrong specification
- **ML decouples the specification search from the variable search**, hence enabling the reduction of complexity via feature selection

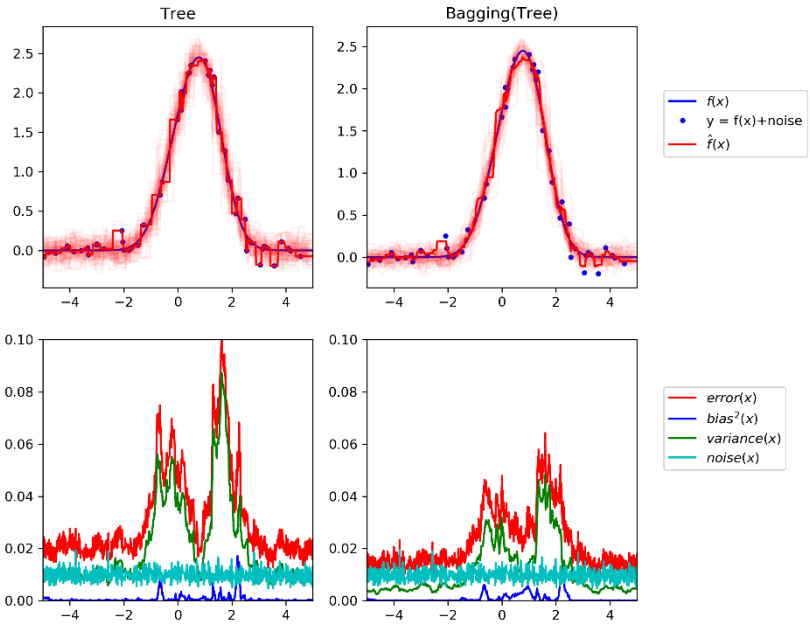


Permutation tests (e.g., MDA) allow us to determine the predictive power associated with each feature. This helps us identify which variables are important, regardless of the true model's specification.

Solution 3:
Ensemble Estimators

Overcoming the Bias-Variance Tradeoff

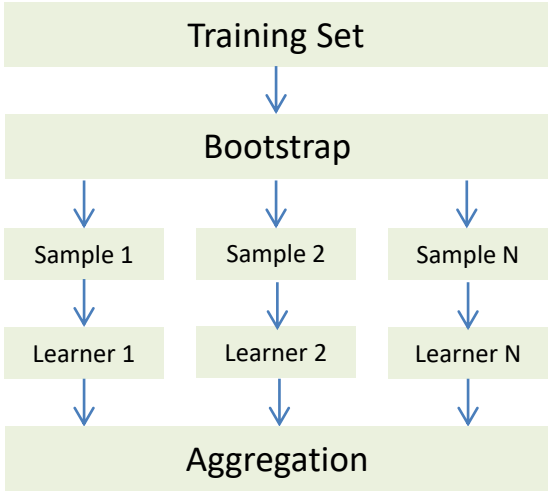
- In general, models with lower variance exhibit higher bias, and models with lower bias exhibit higher variance
- One solution to break this bias-variance tradeoff is to
  1. Overfit a large number of uncorrelated predictors (each of them will have low bias)
  2. Generate an ensemble forecast, based on forecasts from (1)
- Ensemble forecasts exhibit lower variance, and under some circumstances also lower bias, than the individual forecasts



On a non-linear dataset, we fit a decision tree and an ensemble of decision trees. The ensemble achieves lower variance and somewhat lower bias, particularly around the turning points of f.

Bagging

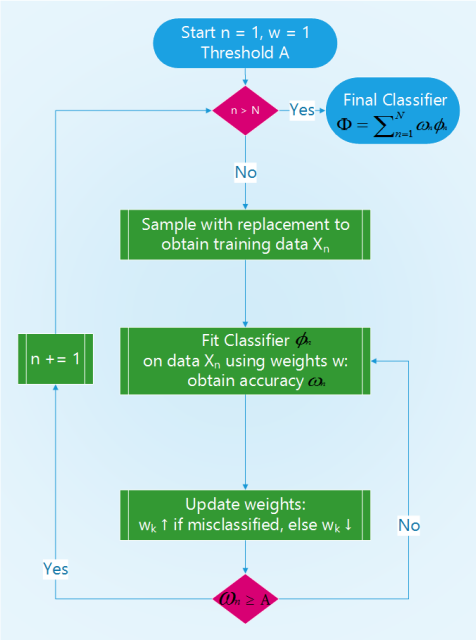
- A bagging algorithm forms a strong learner by
  - training several weak learners independently (in parallel), and
  - aggregating their forecasts using a deterministic procedure
- For a sufficiently large number of uncorrelated learners, bagging algorithms effectively reduce the **variance**
  - For the details, see section 6.3.1 of AFML
- In addition, if the individual learners have a minimum accuracy, bagging algorithms may also reduce the **bias**
  - For the details, see section 6.3.2 of AFML
- In classification problems, two popular forms of aggregation are
  - Hard voting: Majority voting across all the forecasts
  - Soft voting: Average the probabilities of the individual forecasts, and select the class with highest average probability



Bagging almost always achieves variance reduction, and in some cases, also bias reduction. Because of its parallelization, bagging is a good default method to prevent overfitting.

Boosting

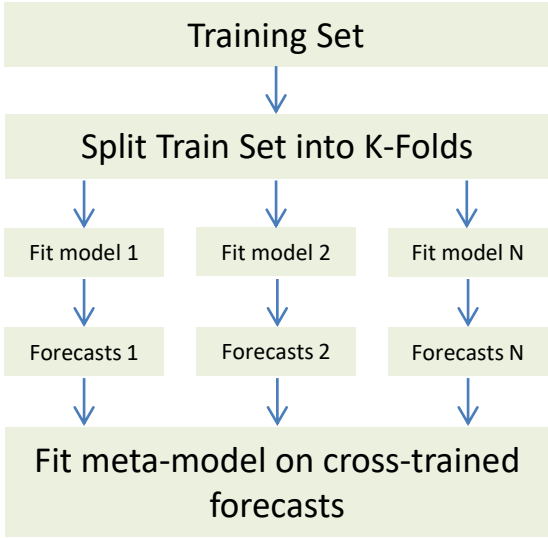
- A boosting algorithm forms a strong learner by
  - training several weak learners iteratively, where subsequent models learn from prior errors, and
  - aggregating their forecasts using a deterministic procedure
- Because of its adaptive nature, boosting is generally more effective than bagging at reducing **bias**
- Boosting can also be effective at reducing **variance**
- Unlike bagging, boosted learners cannot be fit in parallel
  - Consequently, boosting often takes longer to fit
- Two main types of boosting algorithms
  - Adaboost: It adapts by updating the sample weights
  - Gradient boosting: It adapts by updating the observations (residuals)



Adaboost's decision flow.

Stacking

- A stacking algorithm forms a strong learner by
  - training several weak learners in parallel, and
  - training a meta-model on the individual predictions
- Differences with bagging and boosting:
  - Stacking often considers heterogeneous weak learners
  - Stacking uses a meta-model to combine the weak learners (rather than a deterministic method)
- Stacking often involves K-fold cross-training
  1. Split the train set into K-folds
  2. Train N models on K-1 folds, and make predictions on held-out
  3. Train meta-model, with the held-out predictions as features
- Training can be parallelized, however it is computationally expensive



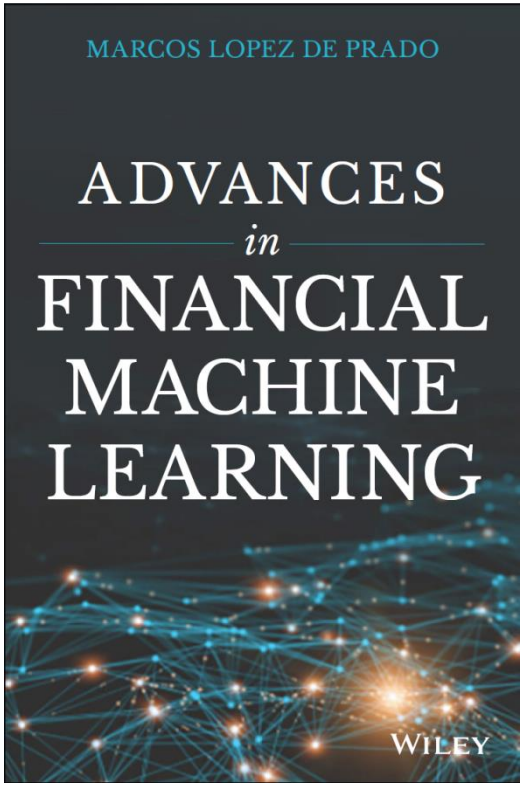
Stacking allows the training of heterogeneous weak learners, which are (non-deterministically) combined by a meta-model.

Conclusions

ML has an Important Role to Play in Finance

- Classical statistics (e.g., Econometrics) relies on assumptions that are often unrealistic in finance
  - In particular, the assumptions that: (i) we have perfect knowledge about the model's specification, and (ii) that we know all the variables involved in a phenomenon (including all interaction effects)
- Classical statistical models applied to financial series often result in poor performance (high bias and high variance)
  - Low *p*-values are meaningless when researchers wrongly assume a linear relationship between variables, or omit interaction effects
    - E.g., there is no theoretical reason to expect a linear relationship between financial returns and risk factors, hence the low *p*-values found in the factor literature can be misleading
- **In the context of financial datasets, ML is a more appropriate quantitative tool**
  - It relies on fewer assumptions, and does not impose a specification, thanks to its numerical approach
  - It maximizes performance, because it does not restrict itself to unbiased estimators
  - It decouples the specification search from the variable search, which is conducive to better theories

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.

Disclaimer

- The views expressed in this document are the author's, 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.
-

www.QuantResearch.org