Machine learning Methods in asset pricing

* I thank Lior Metzker for his help in writing several paragraphs in this section
Machine learning Methods in asset pricing

- Machine learning typically prescribes a vast collection of high-dimensional models attempting to predict quantities of interest while imposing regularization methods.
- Below, I first provide general descriptions of machine learning methods.
- Then, I discuss how machine learning methods have been implemented in asset pricing.
- For perspective, it is well known that the ordinary least squares (OLS) estimator is the best linear unbiased estimator (BLUE) of the regression coefficients.
- “Best" means the lowest variance estimator among all other unbiased linear estimators.
- Notice that the regression errors do not have to be normal, nor do they have to be independent and identically distributed – but they have to be uncorrelated with mean zero and homoskedastic with finite variance.
- In the presence of either heteroskedasticity or autocorrelation, OLS is no longer BLUE.
Shortcomings of OLS

- With heteroskedasticity, we can still use the OLS estimators by finding heteroskedasticity-robust estimators of the variance, or we can devise an efficient estimator by re-weighting the data appropriately to incorporate heteroskedasticity.

- Similarly, with autocorrelation, we can find an autocorrelation-robust estimator of the variance. Alternatively, we can devise an efficient estimator by re-weighting the data appropriately to account for autocorrelation.

- Notice also that the requirement for an unbiased estimator is crucial since biased estimators do exist.

- This is where shrinkage methods come to play: the variance of the OLS estimator can be too high as the OLS coefficients are unregulated.

- If judged by Mean Squared Error (MSE), alternative biased estimators could be more attractive if they produce substantially smaller variance.

- In particular, let $\beta$ denotes the true coefficient and let $\hat{\beta} = (X'X)^{-1}X'Y$, where $X$ is a $T \times M$ matrix of explanatory variables and $Y$ is a $T \times 1$ vector of the dependent variable.
Shortcomings of OLS

Then

\[
\text{MSE}(\hat{\beta}) = E \left[ (\hat{\beta} - \beta)' (\hat{\beta} - \beta) \right] \\
= E \left\{ \text{tr} \left[ (\hat{\beta} - \beta)' (\hat{\beta} - \beta) \right] \right\} \\
= E \left\{ \text{tr} \left[ (\hat{\beta} - \beta)(\hat{\beta} - \beta)' \right] \right\} \\
= \text{tr} \left\{ E \left[ (\hat{\beta} - \beta)(\hat{\beta} - \beta)' \right] \right\} \\
= \sigma^2 \text{tr}[(X'X)^{-1}]
\]

- When predictors are highly correlated the expression $\text{tr}[(X'X)^{-1}]$ can explode.
- Moreover, in the presence of many predictors, OLS delivers nonzero estimates for all coefficients – thus it is difficult to implement variable selection when the true data generating process has a sparse representation.
- Moreover, the OLS solution is not unique if the design of $X$ is not full rank.
- Moreover, the OLS does not account for potential non-linearities and interactions between predictors.
- In sum, OLS is restrictive, often provide poor predictions, may be subject to over-fitting, does not penalize for model complexity, and could be difficult to interpret.
- From a Bayesian perspective, one can think of introducing priors on regression coefficients to shrink slope coefficients towards zero.
- From a classical (non-Bayesian) perspective, shrinkage methods are about penalizing complexity.
Economic restrictions on OLS

- While the next pages discuss shrinkage methods, here are two ways to potentially improve OLS estimates.
- Base case: the pooled OLS estimator corresponds to a panel (balanced) regression of future returns on firm attributes, where \( T \) and \( N \) represent again the time-series and cross-section dimensions.
- The objective is formulated as

\[
L(\theta) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} (r_{i,t+1} - f(c_{i,t}; \theta))^2
\]

where \( f(c_{i,t}; \theta) = c_{i,t}' \theta, r_{i,t+1} \) is stock return at time \( t+1 \) per firm \( i \), and \( c_{i,t} \) is an \( M \)-vector of firm \( i \) attributes realized at time \( t \).
- Predictive performance could be improved using an alternative optimization where stocks are weighted differently based on market size, volatility, credit quality, etc.:

\[
L(\theta) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} w_{i,t} (r_{i,t+1} - f(c_{i,t}; \theta))^2
\]

- An alternative economic based optimization takes account of the heavy tail displayed by stocks and the potential harmful effects of outliers. Then the objective is formulated such that squared (absolute) loss is applied to small (large) errors:

\[
L(\theta) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} H(r_{i,t+1} - f(c_{i,t}; \theta), \xi)
\]

where \( \xi \) is a tuning parameter and

\[
H(x, \xi) = \begin{cases} 
  x^2, & \text{if } |x| \leq \xi \\
  2\xi|x| - \xi^2, & \text{if } |x| > \xi
\end{cases}
\]
Ridge Regression

- There are various shrinkage methods.
- We start with Ridge.
- Hoerl and Kennard (1970a, 1970b) introduce the Ridge regression

\[
\min (Y - X\beta)'(Y - X\beta) \text{ s.t. } \sum_{j=1}^{M} \beta_j^2 \leq c
\]

- The minimization can be rewritten as

\[
\mathcal{L}(\beta) = (Y - X\beta)'(Y - X\beta) + \lambda(\beta'\beta)
\]

- We get

\[
\hat{\beta}_{\text{ridge}} = (X'X + \lambda I_M)^{-1}X'Y
\]

- Notice that including \( \lambda \) makes the problem non-singular even when \( X'X \) is non invertible.
- \( \lambda \) is an hyper-parameter that controls for the amount of regularization.
Ridge Regression

- As $\lambda \to 0$, the OLS estimator obtains.
- As $\lambda \to \infty$, we have $\hat{\beta}^{\text{ridge}} = 0$, or intercept-only model.
- Ridge regressions do not have a sparse representation, so using model selection criteria to pick $\lambda$ does not seem to be feasible.
- Instead, validation methods should be employed.
- The notion of validation is to split the sample into three pieces: training, validation, and test.
- The training sample considers various values for $\lambda$ each of which delivers a prediction. The validation sample is choosing that $\lambda$ that provides the best prediction. Hence, both training and validation samples are used to pick $\lambda$. Then the experiment is assessed through out-of-sample predictions given the choice of $\lambda$.
- As shown below, from a Bayesian perspective, the parameter $\lambda$ denotes the prior precision of beliefs that regression slope coefficients are all equal to zero.
- From a non Bayesian perspective, the ridge estimator is essentially biased:

$$E\left(\hat{\beta}^{\text{ridge}}\right) = [I_M + \lambda(X'X)^{-1}]\beta \neq \beta$$
Interpretations of the Ridge Regression

Interpretation #1: Data Augmentation

- The ridge-minimization problem can be formulated as
  \[
  \sum_{t=1}^{T} (y_t - x_t'\beta)^2 + \sum_{j=1}^{M} (0 - \sqrt{\lambda}_j)^2
  \]

- Thus, the ridge-estimator is the usual OLS estimator where the data is transformed such that
  \[
  X_\lambda = \left( \frac{X}{\sqrt{\lambda}I_M} \right), \quad Y_\lambda = \left( \begin{array}{c} Y \\ 0_M \end{array} \right)
  \]

- Then,
  \[
  \hat{\beta}_{\text{ridge}} = (X_\lambda'X_\lambda)^{-1}X_\lambda'Y_\lambda
  \]

Interpretation #2: Eigen-values and Eigen-vectors

- By the singular value decomposition, we can express \( X \) as
  \[
  X_{T \times M} = U_{T \times M} \Lambda_{M \times M}^{0.5} V_{M \times M}'
  \]

where \( U = [U_1, ..., U_M] \) is any \( T \times M \) orthogonal matrix, \( \Lambda^{0.5} = \begin{bmatrix} \lambda_1^{0.5} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_M^{0.5} \end{bmatrix} \) is an \( M \times M \) matrix so that
Interpretations of the Ridge Regression

- $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$, and $V = [V_1, V_2, \ldots, V_M]$ is an $M \times M$ orthogonal matrix.
- As $X'X = V\Lambda V'$, the matrix $V$ denotes the eigenvectors of $X'X$ and $(\lambda_1, \ldots, \lambda_M)$ are the corresponding eigenvalues.
- How is the predicted value of $Y$ related to eigen vectors?
- To answer, we would like first to find the eigenvectors and eigenvalues of the matrix $Z$
  
  $$Z = X'X + \lambda I_M$$
- We know that for every $j=1,2\ldots,M$, the following holds by definition
  
  $$(X'X)V_j = \lambda_j V_j$$
- Thus
  
  $$(X'X + \lambda I_M)V_j = (X'X)V_j + \lambda V_j$$
  
  $$= \lambda_j V_j + \lambda V_j = (\lambda_j + \lambda)V_j$$
- Telling you that $V$ still denotes the eigenvectors of $Z$ and $\lambda_j + \lambda$ is the $j$-th eigenvalue.
- Notice now that if $A = V\Lambda V'$ then $A^N = V\Lambda^N V'$.
- Same eigenvectors while eigenvalues are raised to the power of $N$. 
Interpretations of the Ridge Regression

- Then, the inverse of the matrix $Z$ is given by

$$Z^{-1} = V \left[ \text{diag} \left( \frac{1}{\lambda_1 + \lambda}, \frac{1}{\lambda_2 + \lambda}, \ldots, \frac{1}{\lambda_M + \lambda} \right) \right] V'$$

- And

$$\hat{\beta}_{\text{ridge}} = Z^{-1} X'Y = (X'X + \lambda I_M)^{-1} X'Y = V \left[ \text{diag} \left( \frac{\lambda_1^{0.5}}{\lambda_1 + \lambda'}, \frac{\lambda_2^{0.5}}{\lambda_2 + \lambda'}, \ldots, \frac{\lambda_M^{0.5}}{\lambda_M + \lambda'} \right) \right] U'Y$$

- And the fitted value is

$$\hat{Y}_{\text{ridge}} = X \hat{\beta}_{\text{ridge}} = \left[ \sum_{j=1}^{M} \left( U_j \frac{\lambda_j}{\lambda_j + \lambda} U'_j \right) \right] Y$$

- Hence, ridge regression projects $Y$ onto components with large $\lambda_j$.

- Or, ridge regression shrinks the coefficients of low variance components.
Informative Bayes Prior

Interpretation #3: Informative Bayes Prior

- Suppose that the prior on $\beta$ is of the form:

$$\beta \sim N\left(0, \frac{1}{\lambda} I_M\right)$$

- Then, the posterior mean of $\beta$ is:

$$(X'X + \lambda I_M)^{-1} X'Y$$

- While we discuss Bayesian approaches later in the notes, consider the family of priors for mean return $\mu$:

$$\mu \sim N\left(0, \frac{\sigma^2}{s^2} V^\eta\right)$$

where $s^2 = \text{trace}(V)$ which is also the sum of the eigenvalues of $V$, or $s^2 = \sum_{j=1}^{N} \lambda_j$.

- To see why, notice that $\text{trace}(V) = \text{trace}(Q \Lambda Q') = \text{trace}(\Lambda Q' Q) = \text{trace}(\Lambda) = \sum_{j=1}^{N} \lambda_j$.

- $\sigma^2$ is a constant controlling for the degree of confidence in the prior.

- The case $\eta = 1$ gives the asset pricing prior of Pastor (2000) and Pastor and Stambaugh (2000).

- The case $\eta = 2$ gives the prior motivated by Kozak, Nagel, and Santosh (2017b).

- The Pastor-Stambaugh prior seems more flexible since factors are pre-specified and are not ordered per their importance.
Informative Bayes Prior

- Let us again make the projection of the original space of returns into the space of principal components:

\[
M_t = 1 - \mu'V^{-1}(r_t - \mu) \\
= 1 - \mu'Q\Lambda^{-1}Q'(r_t - \mu) \\
= 1 - \mu'\Lambda^{-1}(Q_t - \mu_Q) \\
= 1 - b'_Q(Q_t - \mu_Q)
\]

- As \(\mu \sim N\left(0, \frac{s^2}{\xi^2}V\eta\right)\) it follows that \(\mu_Q = Q'\mu\) has the prior distribution

\[
\mu_Q = Q'\mu \sim N\left(0, \frac{s^2}{\xi^2}Q'V\eta Q\right) \\
\sim N\left(0, \frac{s^2}{\xi^2}Q'Q\Lambda\eta Q'Q\right) \\
\sim N\left(0, \frac{s^2}{\xi^2}\Lambda\eta\right)
\]
Informative Bayes Prior

- As $b_Q = \Lambda^{-1}\mu_Q$ its prior distribution is formulated as ($\Lambda$ is assumed known):

$$b_Q = \Lambda^{-1}\mu_Q \sim N\left(0, \frac{\sigma^2}{S^2} \Lambda^{-\eta} \right)$$

- Notice that for $\eta < 2$ the variance of the $b_Q$ coefficients associated with the smallest eigenvalues explodes.

- For $\eta = 2$, the pricing kernel coefficients $b = V^{-1}\mu$ have the prior distribution

$$b \sim N\left(0, \frac{\sigma^2}{S^2} I_N \right)$$

- Picking $\eta = 2$ makes the prior of $b$ independent of $V$.

- Further, the likelihood of $b$ is given by

$$b \sim N\left(V^{-1}\hat{\mu}, \frac{1}{T} V^{-1} \right)$$

where $\hat{\mu}$ is the sample mean return.

- Then, the posterior mean of $b$ is given by $E(b) = (V + \lambda I_N)^{-1}\hat{\mu}$ where $\lambda = \frac{s^2}{T\sigma^2}$.

- The posterior variance is $\text{var}(b) = \frac{1}{T} (V + \lambda I_N)^{-1}$.

- Similar to the ridge regression with a tuning parameter $\lambda$. 

13
The prior expected value of the squared SR is given by (V is assumed known):

\[ E(SR^2) = E(\mu'V^{-1}\mu) = E(\mu'Q\Lambda^{-1}Q'\mu) = E(\mu'_Q\Lambda^{-1}\mu_Q) = E\{\text{trace}(\mu'_Q\Lambda^{-1}\mu_Q)\} \]

\[ = \text{trace}[\Lambda^{-1}E(\mu_Q\mu'_Q)] = \frac{\sigma^2}{s^2} \text{trace}[\Lambda^{\eta-1}] \]

The Pastor-Stambaugh prior (\( \eta = 1 \)) tells you that

\[ E(SR^2) = \frac{\sigma^2}{s^2} \text{trace}(I_N) = N \frac{\sigma^2}{s^2} \]

That is, each principal component portfolio has the same expected contribution to the Sharpe ratio.

If \( \eta = 2 \), then

\[ E(SR^2) = \sum_{j=1}^{N} \frac{\sigma^2}{s^2} \lambda_j = \sigma^2 \]

Thus, the expected contribution of each PC is proportional to its eigenvalue.

This reinforces the notion that only the first few principal components could explain the cross section variation in expected returns.
We now consider various Lasso (least absolute shrinkage and selection operator) models. Tibshirani (1996) was the first to introduce Lasso. Lasso simultaneously performs variable selection and coefficient estimation via shrinkage. While the ridge regression implements an $l_2$-penalty, Lasso is an $l_1$-optimization:

$$
\min (Y - X\beta)'(Y - X\beta) \text{ s.t. } \sum_{j=1}^{M} |\beta_j| \leq c
$$

The $l_1$ penalization approach is called basis pursuit in signal processing. We have again a non-negative tuning parameter $\lambda$ that controls the amount of regularization:

$$
\mathcal{L}(\beta) = (Y - X\beta)'(Y - X\beta) + \lambda \sum_{j=1}^{M} |\beta_j|
$$

Both Ridge and Lasso have solutions even when $X'X$ may not be of full rank (e.g., when there are more explanatory variables than time-series observations) or ill conditioned.
Lasso

- Unlike Ridge, the Lasso coefficients cannot be expressed in closed form.
- However, Lasso can provide with a set of sparse solutions.
- This improves the interpretability of regression models.
- Large enough \( \lambda \), or small enough \( c \), will set some coefficients exactly to zero.
- To understand why, notice that LASSO can be casted as having a Laplace prior \( \beta \)

\[
P(\beta | \lambda) \propto \left( \frac{\lambda}{2\sigma} \right) \exp \left( - \frac{\lambda |\beta|}{\sigma} \right)
\]

- In particular, Lasso obtains by combining Laplace prior and normal likelihood.
- Like the normal distribution, Laplace is symmetric.
- Unlike the normal distribution, Laplace has a spike at zero (first derivative is discontinuous) and it concentrates its probability mass closer to zero than does the normal distribution.
- This could explain why Lasso (Laplace prior) sets some coefficients to zero, while Ridge (normal prior) does not.
Lasso

- Bayesian information criterion (BIC) is often used to pick $\lambda$.
- BIC (like other model selection criteria) is composed of two components: the sum of squared regression errors and a penalty factor that gets larger as the number of retained characteristics increases.

$$BIC = T \times \log \left( \frac{RSS}{T} \right) + k \times \log(T), \text{ where } k \text{ is the number of retained characteristics.}$$

- Notice that different values of lambda affect the optimization in a way that a different set of characteristics is selected.
- You choose $\lambda$ as follows: initiate a range of values, compute BIC for each value, and pick the one that minimizes BIC.
- The next page provides steps for assessing the maximum value of $\lambda$ in formulating the range.
Lasso

- For convenience, let us formulate again the objective function
  \[ L(\beta) = (Y - X\beta)'(Y - X\beta) + \lambda \sum_{j=1}^{M} |\beta_j| \]

- If a change of \( \beta \) does not decrease the objective function \( L(\beta) \), then it is a local minimum.

- An infinitesimal change in \( \beta_j, \partial \beta_j \) would change the objective function \( L(\beta) \) as follows
  - The penalty term changes by \( \lambda \text{sign}(\beta_j) \partial \beta_j \)
  - The squared error term changes by \( \partial \text{RSS} = (2Y'X^j + 2\beta_j) \partial \beta_j \), where \( X^j \) is the \( j \)'s row of the matrix \( X \).
  - If \( \beta = 0 \) then \( \partial \text{RSS} = (2Y'X^j) \partial \beta_j \)
  - For the objective function to decrease, the change in the RSS should be greater than the change in penalty:
    \[ \frac{|(2Y'X^j)\partial \beta_j|}{|\lambda \text{sign}(\beta_j) \partial \beta_j|} > 1 \], hence \( \lambda < |(2Y'X^j)| \), or
  \[ \lambda = \max_j |(2Y'X^j)| \]
Adaptive Lasso and Bridge regression

- LASSO forces coefficients to be equally penalized.
- One modification is to assign different weights to different coefficients:
  \[ L(\beta) = (Y - X\beta)'(Y - X\beta) + \lambda \sum_{j=1}^{M} w_j |\beta_j| \]
- It can be shown that if the weights are data driven and are chosen in the right way such weighted LASSO can have the so-called oracle properties even when the LASSO does not. This is the adaptive LASSO.
- For instance, \( w_j \) can be chosen such that it is equal to one divided by the absolute value of the corresponding OLS coefficient raised to the power of \( \gamma > 0 \). That is, \( w_j = \frac{1}{|\beta_j|^{\gamma}} \) for \( j = 1, ..., M \).
- The adaptive LASSO estimates are given by
  \[ L(\beta) = (Y - X\beta)'(Y - X\beta) + \lambda \sum_{j=1}^{M} w_j |\beta_j| \]
- The adaptive LASSO is a convex optimization problem and thus does not suffer from multiple local minima.
- Moving on, Frank and Friedman (1993) introduce the bridge regression (generalizes for \( \ell_q \) penalty)
  \[ L(\beta) = (Y - X\beta)'(Y - X\beta) + \lambda \sum_{j=1}^{M} |\beta_j|^q \]
- Notice that \( q = 0, 1, 2 \), correspond to OLS, LASSO, and ridge, respectively.
- Moreover, the optimization is convex for \( q \geq 1 \) and the solution is sparse for \( 0 \leq q \leq 1 \).
The Elastic Net

- The elastic net is yet another regularization and variable selection method.
- Zou and Hastie (2005) describe it as stretchable fishing net that retains “all big fish”.
- Using simulation, they show that it often outperforms Lasso in terms of prediction accuracy.
- The elastic net encourages a grouping effect, where strongly correlated predictors tend to be in or out of the model together.
- The elastic net is particularly useful when the number of predictors is much bigger than the number of observations.
- The naïve version of the elastic net is formulated through

  \[ L(\beta) = (Y - X\beta)'(Y - X\beta) + \lambda_1 \sum_{j=1}^{M} |\beta_j| + \lambda_2 \sum_{j=1}^{M} \beta_j^2 \]

- Thus, the elastic net combines $l_1$ and $l_2$ norm penalties.
- It still produces sparse representations.
Again, the data consist of $Y$, a $T$-vector of the dependent variable, and $X$, a $T \times M$ matrix of explanatory variables.

Suppose that the $M$ predictors are divided into $L$ groups, with $M_l$ denoting the number of predictors in group $l$.

$X_l$ represents the predictors corresponding to the $l$-th group, while $\beta_l$ is the corresponding coefficient vector.

Notice, $\beta=[\beta_1', \beta_2', ..., \beta_L']'$.

Assume that both $Y$ and $X$ have been centered.

The group Lasso solves the convex optimization problem.

$$\mathcal{L}(\beta) = \left( Y - \sum_{l=1}^{L} X_l \beta_l \right)' \left( Y - \sum_{l=1}^{L} X_l \beta_l \right) + \lambda \left( \sum_{l=1}^{L} \beta_l^T \beta_l \right)^{\frac{1}{2}}$$

The group Lasso does not yield sparsity within a group.

If a group of parameters is non-zero, they will all be non-zero.

The **sparse** group Lasso criterion does yield sparsity

$$\mathcal{L}(\beta) = \left( Y - \sum_{l=1}^{L} X_l \beta_l \right)' \left( Y - \sum_{l=1}^{L} X_l \beta_l \right) + \lambda_1 \left( \sum_{l=1}^{L} \beta_l^T \beta_l \right)^{\frac{1}{2}} + \lambda_2 \sum_{j=1}^{M} |\beta_j|$$
Non parametric shrinkage

- Lasso, Adaptive Lasso, Group Lasso, Ridge, Bridge, and Elastic net are all linear or parametric approaches for shrinkage.
- Group Lasso implements the same penalty to predictors belonging to some pre-specified group while different penalty applies to different groups.
- Some other parametric approaches (uncovered here) include the smoothed clip absolute deviation (SCAD) penalty of Fang and Li (2001) and Fang and Peng (2004) and the minimum concave penalty of Zhang (2010).
- In many application, however, there is little a priori justification for assuming that the effects of covariates take a linear form or belong to other known parametric family.
- Huang, Horowitz, and Wei (2010) thus propose to use a nonparametric approach: the adaptive group Lasso for variable selection.
- This approach is based on a spline approximation to the nonparametric components.
- To achieve model selection consistency, they apply the group Lasso in two steps.
- First, they use the group Lasso to obtain an initial estimator and reduce the dimension of the problem.
- Second, they use the adaptive group Lasso to select the final set of nonparametric components.
In finance, Cochrane (2011) notes that portfolio sorts are the same thing as nonparametric cross section regressions. Drawing on Huang, Horowitz, and Wei (2010), Freyberger, Neuhier, and Weber (2017) study this equivalence formally.

The cross section of stock returns is modelled as a non-linear function of firm characteristics:

\[ r_{it} = m_t \left( C_{1,it-1}, \ldots, C_{S,it-1} \right) + \epsilon_{it} \]

Notation:

- \( r_{it} \) is the return on firm \( i \) at time \( t \).
- \( m_t \) is a function of \( S \) firm characteristics \( C_1, C_2, \ldots, C_S \).
- Notice, \( m_t \) itself is not stock specific but firm characteristics are.

Consider an additive model of the following form

\[ m_t \left( C_1, \ldots, C_S \right) = \sum_{s=1}^{S} m_{t,s} \left( C_s \right) \]

As the additive model implies that \( \frac{\partial^2 m_t \left( c_1, \ldots, c_S \right)}{\partial c_s \partial c_{s'}} = 0 \) for \( s \neq s' \), apparently there should not be no cross dependencies between characteristics.

Such dependencies can still be accomplished through producing more predictors as interactions between characteristics.
For each characteristic \( s \), let \( F_{s,t}(\cdot) \) be a strictly monotone function and let \( F_{s,t}^{-1}(\cdot) \) denote its inverse.

Define \( \tilde{c}_{s,\text{it}-1} = F_{s,t}(C_{s,\text{it}-1}) \) such that \( \tilde{c}_{s,\text{it}-1} \in [0,1] \).

That is, characteristics are monotonically mapped into the \([0,1]\) interval.

An example for \( F_{s,t}(\cdot) \) is the rank function: \( F_{s,t}(C_{s,\text{it}-1}) = \frac{\text{rank}(C_{s,\text{it}-1})}{N_{t+1}} \), where \( N_t \) is the total number of firms at time \( t \).

The aim then is to find \( \hat{m}_t \) such that

\[
m_t(C_1, \ldots, C_S) = \hat{m}_t(\tilde{c}_{1,\text{it}-1}, \ldots, \tilde{c}_{s,\text{it}-1})
\]

In particular, to estimate the \( \hat{m}_t \) function, the normalized characteristic interval \([0,1]\) is divided into \( L \) subintervals (\( L+1 \) knots): \( 0 = x_0 < x_1 < \cdots < x_{L-1} < x_L = 1 \).

To illustrate, consider the equal spacing case.

Then, \( x_l = \frac{l}{L} \) for \( l=0,\ldots,L-1 \) and the intervals are:

\[
\bar{I}_1 = [x_0, x_1), \quad \bar{I}_l = [x_{l-1}, x_l) \text{ for } l=2,\ldots,L-1, \text{ and } \bar{I}_L = [x_{L-1}, x_L]
\]
Nonparametric Models

- Each firm characteristic is transformed into its corresponding interval.
- Estimating the unknown function \( \tilde{m}_{t,s} \) nonparametricaly is done by using quadratic splines.
  - The function \( \tilde{m}_{t,s} \) is approximated by a quadratic function on each interval \( \tilde{I}_l \).
  - Quadratic functions in each interval are chosen such that \( \tilde{m}_{t,s} \) is continuous and differentiable in the whole interval \([0,1]\).
  - \( \tilde{m}_{t,s}(\tilde{c}) = \sum_{k=1}^{L+2} \beta_{tsk} \times p_k(\tilde{c}) \), where \( p_k(\tilde{c}) \) are basis functions and \( \beta_{tsk} \) are estimated slopes.
  - In particular, \( p_1(y) = 1, p_2(y) = y, p_3(y) = y^2 \), and \( p_k(y) = \max\{y - x_{k-3}, 0\}^2 \) for \( k = 4, ..., L + 2 \).
  - In that way, you can get a continuous and differentiable function.
- To illustrate, consider the case of two characteristics, e.g., size and book to market (BM), and 3 intervals.
- Then, the \( \tilde{m}_t \) function is:

\[
\tilde{m}_t(\tilde{c}_{i,\text{size}}, \tilde{c}_{i,\text{BM}}) = \beta_{t,\text{size},1} \times 1 + \beta_{t,\text{size},2} \times \tilde{c}_{i,\text{size}} + \beta_{t,\text{size},3} \times \tilde{c}_{i,\text{size}}^2 + \beta_{t,\text{size},4} \times \max\{\tilde{c}_{i,\text{size}} - 1/3, 0\}^2 + \beta_{t,\text{size},5} \times \max\{\tilde{c}_{i,\text{size}} - 2/3, 0\}^2 \\
+ \beta_{t,\text{BM},1} \times 1 + \beta_{t,\text{BM},2} \times \tilde{c}_{i,\text{BM}} + \beta_{t,\text{BM},3} \times \tilde{c}_{i,\text{BM}}^2 + \beta_{t,\text{BM},4} \times \max\{\tilde{c}_{i,\text{BM}} - 1/3, 0\}^2 + \beta_{t,\text{BM},5} \times \max\{\tilde{c}_{i,\text{BM}} - 2/3, 0\}^2
\]
Adaptive group Lasso

- The estimation of $\tilde{m}_t$ is done in two steps:
- First step, estimate the slope coefficients $b_{sk}$ using the group Lasso routine:

$$\hat{\beta}_t = \arg\min_{b_{sk:s=1,..,S;k=1,..,L+2}} \sum_{i=1}^{N_t} \left( r_{it} - \sum_{s=1}^{S} \sum_{k=1}^{L+2} b_{sk} \times p_k(\tilde{C}_{s,it-1}) \right)^2 + \lambda_1 \sum_{s=1}^{S} \left( \sum_{k=1}^{L+2} b_{sk}^2 \right)^{1/2}$$

- Altogether, the number of $b_{sk}$ coefficients is $S \times (L + 2)$.
- The second is a penalty term applied to the spline expansion.
- $\lambda_1$ is chosen such that it minimizes the Bayesian Information Criterion (BIC).
- The essence of group Lasso is either to include or exclude all $L+2$ spline terms associated with a given characteristic.
- While this optimization yields a sparse solution there are still many characteristics retained.
- To include only characteristics with a strong predictive the adaptive Lasso is then employed.
Adaptive group Lasso

To implement adaptive group Lasso, define the following weights using estimates for $b_{sk}$ from the first step:

$$w_{ts} = \begin{cases} \left( \sum_{k=1}^{L+2} \tilde{b}_{sk}^2 \right)^{-1/2} & \text{if } \sum_{k=1}^{L+2} \tilde{b}_{sk}^2 \neq 0 \\ \infty & \text{if } \sum_{k=1}^{L+2} \tilde{b}_{sk}^2 = 0 \end{cases}$$

Then estimate again the coefficients $b_{sk}$ using the above-estimated weights $w_{ts}$

$$\hat{\beta}_t = \arg\min_{b_{sk:s=1,\ldots,S,k=1,\ldots,L+2}} \sum_{i=1}^{N_t} \left( r_{it} - \sum_{s=1}^{S} \sum_{k=1}^{L+2} b_{sk} \times p_k(\tilde{C}_{s,it-1}) \right)^2 + \lambda_2 \sum_{s=1}^{S} \left( w_{ts} \sum_{k=1}^{L+2} b_{sk}^2 \right)^{1/2}$$

$\lambda_2$ is chosen such that it minimizes BIC.

The weights $w_{ts}$ guarantee that we do not select any characteristic in the second step that was not selected in the first step.
Regression Trees

- Also a nonparametric approach for incorporating interactions among predictors.
  - At each step, a new branch sorts the data from the preceding step into two bins based on one of the predictive variables.
  - Let us denote the data set from the preceding step as $C$ and the two new bins as $C_{left}$ and $C_{right}$.
  - Let us denote the number of elements in $C$, $C_{left}$, $C_{right}$ by $N$, $N_{left}$, $N_{right}$, respectively.
  - The specific predictor variable and its threshold value is chosen to minimize the sum of squared forecast errors
    $$\mathcal{L}(C, C_{left}, C_{right}) = H(\theta_{left}, C_{left}) + H(\theta_{right}, C_{right})$$
    where
    $$H(\theta, \mathcal{X}) = \sum_{z_{it} \in \mathcal{X}} (r_{i,t+1} - \theta)^2.$$  
  - We should also sum through the time-series dimension. I skip it to ease notation.
  - We compute $\mathcal{L}(C, C_{left}, C_{right})$ for each predictor and choose the one with the minimum loss.
The predicted return is the average of returns of all stocks within the group

\[ \theta_{left} = \frac{1}{N_{left}} \sum_{z_{i,t} \in C_{left}} r_{i,t+1} ; \quad \theta_{right} = \frac{1}{N_{right}} \sum_{z_{i,t} \in C_{right}} r_{i,t+1} \]

The division into lower leaves terminates when the number of leaves or the depth of the tree reach a pre-specified threshold.

The prediction of tree with \( K \) leaves (terminal nodes), and depth \( L \), can be written as

\[ g(z_{i,t}; \theta, K, L) = \sum_{k=1}^{K} \theta_k 1\{z_{i,t} \in C_k(L)\} \]

where \( C_k(L) \) is one of the \( K \) partitions of the data. Each partition is a product of up to \( L \) indicator functions of the predictors.

The constant associated with partition \( k, \theta_k \) is defined as the sample average of outcomes within the partition.
Regression Trees - Example

\[ g(z_{i,t}; \theta, 3, 2) = \theta_1 1\{\text{size}_{i,t} < 0.5\} 1\{b/m_{i,t} < 0.3\} + \]

\[ \theta_2 1\{\text{size}_{i,t} < 0.5\} 1\{b/m_{i,t} \geq 0.3\} + \theta_3 1\{\text{size}_{i,t} \geq 0.5\} \]
Regression Trees

**Pros**
- Tree model is invariant to monotonic transformations of the predictors.
- It can approximate nonlinearities.
- A tree of depth $L$ can capture, at most, $L-1$ interactions.

**Cons**
- Prone to over fit and therefore they are not used without regularization.
Random Forest

- Random forest is an ensemble method that combines forecasts from many different shallow trees.
- For each tree a subset of predictors is drawn at each potential branch split.
- This lowers the correlation among different trees and improves the prediction.
- The depth, $L$, of the trees is a tuning (hyper) parameter and is optimized in the validation stage.
Neural Networks

- A nonlinear feed forward method.
- The network consists of an “input layer”, one or more “hidden layers” that interacts and an “output layer”.
- Analogous to axons in a biological brain, layers of the networks represent groups of “neurons” with each layer connected by “synapses” that transmit signals among neurons of different layer.
- Deep learning reflects the notion that the number of hidden layers is large (10-20 and more).
Neural Networks

- Each neuron applies a nonlinear “activation function” $f$ to its aggregated signal before sending its output to the next layer:

$$x^l_k = f(\theta_0 + \sum_j z_j \theta_j)$$

where $x^l_k$ is neuron $k \in 1, 2, \ldots, K^l$ in the hidden layer $l \in 1, 2, \ldots, L$.

- The function $f$ is usually one of the following functions:
  - Sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$
  - $\tanh(x) = 2\sigma(x) - 1$
  - $ReLU(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{otherwise} \end{cases}$

- The activation function is operationalized in each Neuron excluding the last one.

- Linear activation function boils down to OLS.

- It is often confusing – Neural Network only captures non linearities – it does not capture interactions between the characteristics.

- This could be shown by the formulation on the next page.
Neural Networks

- For the ReLU activation function, we can rewrite the neural network function as:

\[
Fitted\ Value = \max \left( \max \left( \max \left( X W_{h1}^1, 0 \right) W_{h1}^2, 0 \right) \ldots W_{h1}^n, 0 \right) W_{output}
\]

where \( X \) is the input, \( W_{h1}^i \) are the weight matrix of the neurons in hidden layer \( i \in 1, \ldots, n \), \( n \) is the number of hidden layers, and \( W_{output} \) are the weighs of the output layer.

- Then run an optimization that minimizes the sum of squared errors, just like OLS.

- Should also include a LASSO routine to zero out some coefficients (does not essentially translate into zeroing-out some characteristics).

- If the activation function is linear – simply ignore the MAX operator.

- Then the fitted value is \( XW \) – just like OLS.
A simple example

- Two inputs: size and BM (book to market)
- One hidden layer with three neurons: A, B, and C
- $W$s are the weights and $b$’s are the intercepts (biases).
- $input^A = size \times W^A_{size} + BM \times W^A_{BM} + b^A$
- $output^A = \max(input^A, 0)$
- $input^B = size \times W^B_{size} + BM \times W^B_{BM} + b^B$
- $output^B = \max(input^B, 0)$
- $input^C = size \times W^C_{size} + BM \times W^C_{BM} + b^C$
- $output^C = \max(input^C, 0)$
- Output layer (ol): $output = output^A \times W^A_{ol} + output^B \times W^B_{ol} + output^C \times W^C_{ol} + b^ol$
- Minimize the loss function, the sum squared errors (data versus output).
- Account for LASSO to zero out some of the weights
Neural network with classification

- The probability of the next month return being above or below the median return can be described as:

$$y_{i,t+1} = \begin{cases} 1, & \text{if } r_{i,t+1} > \text{median } (r_{t+1}) \\ 0, & \text{otherwise} \end{cases}$$

- The classification approach allows the modeling of the second order effects of inputs and on its predictions straightforwardly.
- Let us define $\hat{p}^a$ as the predicted probability of stock $i$’s return being above the median of the cross-sectional return distribution at time $t + 1$.
- A binary classifier models this probability as a function of the vector of time $t$’s features $X_{i,t}$ and estimated parameters $\hat{w}$:

$$\hat{p}^a_{i,t+1} = f(X_{i,t+1};\hat{w}) = \frac{1}{1 + e^{-h(X_{i,t+1};\hat{w})}}$$

where $h(X_{i,t+1};\hat{w})$ is the value of the output layer before the logistic transformation.
Neural network with classification

- Classification has several advantages:
  - The labels have the same magnitude and distribution over time, and thus, they simplify training by alleviating the problem in the returns.
  - Estimated probability is directly proportional to the expected return, under the assumption:
    \[ p_{i,t+1} = \hat{p}_{i,t+1} + \varepsilon_{i,t+1}, \]
    where \( \varepsilon_{i,t+1} \) is IID through time.
  - Binary classification targets the median portfolios sorted on model’s predictions.
  - The loss function is given by:
    \[ L(w) = -\frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \left[ y_{i,t+1} \log(\hat{p}_{i,t}^a) + (1 - y_{i,t}) \log(1 - (\hat{p}_{i,t}^a)) \right] \]
Generative Adversarial Network - GAN

- GAN is a setup in which two neural networks contest with each other in a (often zero-sum) game.
- For example, let \( w \) and \( g \) be two neural networks’ outputs.
- The loss function is defined over both outputs, \( L(w, g) \).
- The competition between the two neural networks is done via iterating both \( w \) and \( g \) sequentially:
  - \( w \) is updated by minimizing the loss while \( g \) is given
    \[ \hat{w} = \min_w L(w|g) \]
  - \( g \) is the adversarial and it is updated by maximizing the loss while \( w \) is given
    \[ \hat{g} = \max_g L(g|w) \]
Adversarial GMM

- Chen, Pelger, and Zhu (2019) employ an adversarial GMM (General Method of Moments) in order to estimate the Stochastic Discount Factor (SDF).

- Adversarial GMM is using a GAN for solving conditional moment conditions.

- The model is formulated as follows:
  - For any excess return the conditional expectation at time $t$ is
    \[ \mathbb{E}_t(M_{t+1}R_{t+1,i}^e) = 0 \]
    where $M_{t+1}$ is the SDF and $R_{t+1,i}^e$ is security’s $i$ excess return at time $t + 1$. This equality should hold for any $i = 1, \ldots, N$
  - The SDF is of the form
    \[ M_{t+1} = 1 - \sum_{i=1}^{N} w_{t,i}R_{t+1,i}^e \]
    where $w_{t,i}$ is security’s $i$ weight, which is a function of firm’s $i$ characteristics at time $t$, $l_{t,i}$.
  - In the adversarial GMM, the first neural network is generating the function $w_{t,i}$ for each $i$. 

Adversarial GMM

- To switch from the conditional expectation to the unconditional expectation we multiply moment conditions by a function measurable with respect to time $t$

$$E(M_{t+1} R_{t+1,i} g(I_{t,i})) = 0$$

- This equality should hold for any function $g$.
  - The function $g$ is the second or adversarial neural network.
  - Each output of $g$ is in fact a moment condition

- In the adversarial approach the moment conditions are those that lead to the largest mispricing:

$$\min_{w} \max_{g} \frac{1}{N} \sum_{j=1}^{N} \left\| \frac{1}{T} \sum_{t=1}^{T} \left( 1 - \sum_{i=1}^{N} w(I_{t,i}) R_{t+1,i}^e \right) R_{t+1,j}^e g(I_{t,j}) \right\|^2$$

- After convergence, we can construct time-series observations for SDF using $w$ and excess returns.
Sparse versus non sparse representations

- This research controversy crosses disciplines that deal with model uncertainty and variable selection.
- Fu (1998) advocates using general cross-validation to select the shrinkage parameter ($q$) and the tuning parameter ($\lambda$) in a bridge regression setup and shows that bridge regression performs well. The bridge regression could produce non sparse solutions.
- In asset pricing, Freyberger, Neuhierl, and Weber (2017) use the adaptive group LASSO to select characteristics and to estimate how they affect expected returns non-parametrically. Feng, Giglio, and Xiu (2017) implements a methodology that marries the double selection LASSO method of Belloni, Chernozhukov, and Hansen (2014b).
Sparse versus non sparse representations

- Both these studies adopt Lasso-style estimation with $l_1$-penalty and they suggest a relatively high degree of redundancy among equity predictors.
- In contrast, Kozak, Nagel, and Santosh (2017b) advocate against sparse solutions.
- They propose a Bayesian approach that shrinks the SDF coefficients towards zero.
- Their formulation is similar to a ridge regression with an important difference.
- As informative priors are about the pricing kernel parameters the degree of shrinkage is not equal for all assets. Rather there is more shrinkage to SDF coefficients associated with low-eigenvalue PCs.
- Notice that while the q-model or the Fama-French five factor model can imply sparse solution a caveat is in order. Take the present-value relation - it can indeed motivate why book-to-market and expected profitability could jointly explain expected returns. However, the notion that expected profitability is unobserved gives license to fish a large number of observable firm characteristics that predict future returns through their ability to predict future profitability.
Machine Learning methods in Asset Pricing

- Machine learning methods in asset pricing have been implicitly motivated by the presidential address of Cochrane.

- John Cochrane points out that in the beginning there was chaos.

- Practitioners thought that one only needed to be clever to earn high returns, but then came the CAPM.

- Every clever strategy that delivered high average return ended up delivering high market beta.

- Then anomalies (size, value, momentum, profitability, investment) erupted, and there was chaos again.

- To address the “zoo” of variables, Cochrane suggests, we should consider different methods, namely, beyond cross section regressions and portfolio sorts.
Zoo of Anomalies

Source: Harvey, Liu, and Zhu (2016)
Challenging the multi-dimensional challenge

- Harvey, Liu, and Zhu (2016): 296 anomalies, 27% to 53% are likely to be false discoveries
- McLean and Pontiff (2016): 97 anomalies, out-of-sample returns are 26% lower
- Hou, Xue, and Zhang (2018): 452 anomalies, 82% turn insignificant upon excluding microcaps + value-weighting
- Anomaly profits are mostly attributable to the short leg of the trade (Stambaugh, Yu, and Yuan, 2012)
- Anomalies characterize distressed stocks (Avramov, Chordia, Jostova, and Philipov, 2013)
- Anomalies attenuate, and often disappear, in recent years (Chordia, Subrahmanyam, and Tong, 2014)
Machine learning in asset pricing

- Counter to this ‘anomaly-challenging’ strand of literature, there has been a fast growing body of work that reports outstanding investment profitability based on signals from machine learning methods.

- Reduced form predictability
  - Freyberger, Neuhierl, and Weber (2018); Feng, He, and Polson (2019); Gu, Kelly and Xiu (2019)

- Predictability with asset pricing restrictions
  - Kelly, Pruitt, and Su (2018); Lettau and Pelger (2018); Chen, Pelger, and Zhu (2019); Feng, Giglio, and Xiu (2019); Kozak, Nagel, and Santosh (2019)
Machine learning – motivation

- So, papers on machine learning methods have been motivated by the Cochrane’s call for new methods.
- What needs to be done?
  - Address the high-dimension of noisy and correlated predictors
  - Utilize flexible, possibly non-linear, functional forms
  - Implement model selection
  - Mitigate overfitting biases through regularization
- Machine Learning: automated detection of complex patterns in data; combine multiple, possibly weak, sources of information into a meaningful composite signal
Do ML methods clear standard economic restrictions in asset pricing?

- **Cross section:** Exclude difficult-to-value and arbitrage stocks (microcaps or distressed stocks).

- **Time series:** Is ML based predictability manifested through market states associated with alleviated limits to arbitrage (e.g., high VIX, high illiquidity)?

- Assess turnover and potential impact of trading costs.

- Examine whether the tangency portfolio based on estimating SDF through machine learning is admissible.

- Assess the economic grounds for the seemingly opaque ML methods.

- We primarily focus on the two deep-learning methods noted earlier and also consider the shrinkage approach advocated by Kozak, Nagel, and Santosh (2018).
Deep Learning Method I: GKX

- Neural network with 3 hidden layers (NN3) (Gu, Kelly, and Xiu 2019)
  - 32, 16, and 8 neurons per layer
  - Reduced form, no economic restrictions
- 94 firm characteristics + 74 industry dummies + 8 macroeconomic predictors + interactions
  - \((8+1) \times 94 + 74 = 920\) predictors
- Training sample: 18 years, 1957 to 1974
- Validation sample: 12 years, 1975 to 1986
- Out-of-sample test: 31 years, 1987 to 2017
Deep Learning Method II: CPZ

- Multiple connected neural networks (Chen, Pelger, and Zhu 2019)
- Incorporate no-arbitrage conditions to estimate SDF and stock risk loadings

![Model architecture of GAN (Generative Adversarial Network) with RNN (Recurrent Neural Network) with LSTM cells.](image)
Deep Learning Method II: CPZ

- Multiple connected neural networks (Chen, Pelger, and Zhu 2019)
- 46 firm characteristics + 178 macroeconomic predictors + interactions
- Training sample: 20 years, 1967 to 1986
- Validation sample: 5 years, 1987 to 1991
Data
- CRSP: daily and monthly stock data
- COMPUSTAT: quarterly and annual financial statement data
- GKX: all NYSE/AMEX/Nasdaq stocks, set missing characteristics to cross-sectional median
  - 21,882 stocks, between 5,117 and 7,877 per month
- CPZ: all U.S. stocks from CRSP with available data on firm characteristics
  - 7,904 stocks, between 1,933 and 2,755 per month
Economic Restrictions

- Cross-sectional return predictability is concentrated in **microcaps** and **distressed** firms.
- **Microcaps**: market cap smaller than the 20th NYSE size percentile.
- **Rated firms**: firms with data on S&P long-term issuer credit rating.
- **Distressed firms**: lower rated firms that further undergo deteriorating credit conditions.
Sub-samples with Economic Restrictions: GKX
Sub-samples with Economic Restrictions: CPZ
GKX Portfolio Return Spread: EW

- Long (Short) stocks with the highest (lowest) NN3-predicted returns, monthly rebalanced decile portfolios
GKX Portfolio Return Spread: EW vs. VW

- VW performance is 47% lower than EW.
GKX Portfolio Return Spread: EW vs. VW

- Exclude microcaps: 48% lower than the full sample; Rated firms: 46% ↓; Exclude downgrades: 70% ↓
CPZ Portfolio Return Spread: EW vs. VW

- Long (Short) stocks with the highest (lowest) risk loadings on SDF, monthly rebalanced decile portfolios
- VW performance is 43% lower than EW.
- Exclude microcaps: 62% lower than the full sample; Rated firms: 72% ↓; Exclude downgrades: 64% ↓
Characteristics of VW ML Portfolios

<table>
<thead>
<tr>
<th>Panel A: Sorted by NN-Predicted Return</th>
<th>Sharpe Ratio</th>
<th>Skewness</th>
<th>Excess Kurtosis</th>
<th>Maximum Drawdown</th>
<th>Return in Crisis</th>
<th>Turnover</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Sample</td>
<td>0.944</td>
<td>0.631</td>
<td>5.222</td>
<td>0.350</td>
<td>4.100</td>
<td>1.127</td>
</tr>
<tr>
<td>Non-Microcaps</td>
<td>0.644</td>
<td>0.361</td>
<td>7.062</td>
<td>0.349</td>
<td>3.563</td>
<td>0.992</td>
</tr>
<tr>
<td>Credit Rating Sample</td>
<td>0.639</td>
<td>0.064</td>
<td>7.875</td>
<td>0.420</td>
<td>3.435</td>
<td>1.019</td>
</tr>
<tr>
<td>Non-Downgrades</td>
<td>0.449</td>
<td>0.146</td>
<td>8.550</td>
<td>0.333</td>
<td>2.931</td>
<td>1.068</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Panel B: Sorted by Risk Loading</th>
<th>Sharpe Ratio</th>
<th>Skewness</th>
<th>Excess Kurtosis</th>
<th>Maximum Drawdown</th>
<th>Return in Crisis</th>
<th>Turnover</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Sample</td>
<td>1.225</td>
<td>1.063</td>
<td>5.932</td>
<td>0.209</td>
<td>0.472</td>
<td>1.775</td>
</tr>
<tr>
<td>Non-Microcaps</td>
<td>0.839</td>
<td>0.326</td>
<td>1.582</td>
<td>0.246</td>
<td>0.677</td>
<td>1.721</td>
</tr>
<tr>
<td>Credit Rating Sample</td>
<td>0.566</td>
<td>0.267</td>
<td>1.440</td>
<td>0.407</td>
<td>-0.023</td>
<td>1.758</td>
</tr>
<tr>
<td>Non-Downgrades</td>
<td>0.602</td>
<td>0.344</td>
<td>1.675</td>
<td>0.447</td>
<td>0.903</td>
<td>1.795</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Panel C: CRSP Market Index</th>
<th>Sharpe Ratio</th>
<th>Skewness</th>
<th>Excess Kurtosis</th>
<th>Maximum Drawdown</th>
<th>Return in Crisis</th>
<th>Turnover</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market</td>
<td>0.516</td>
<td>-0.991</td>
<td>3.334</td>
<td>0.481</td>
<td>-7.051</td>
<td>0.093</td>
</tr>
</tbody>
</table>

- ML methods: positive skewness and excess kurtosis; smaller maximum drawdown than the market; positive return during the crisis period
Characteristics of VW ML Portfolios

<table>
<thead>
<tr>
<th>Panel A: Sorted by NN3-Predicted Return</th>
<th>Sharpe Ratio</th>
<th>Skewness</th>
<th>Excess Kurtosis</th>
<th>Maximum Drawdown</th>
<th>Return in Crisis</th>
<th>Turnover</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Sample</td>
<td>0.944</td>
<td>0.631</td>
<td>5.222</td>
<td>0.350</td>
<td>4.100</td>
<td>1.127</td>
</tr>
<tr>
<td>Non-Microcaps</td>
<td>0.644</td>
<td>0.361</td>
<td>7.062</td>
<td>0.349</td>
<td>3.563</td>
<td>0.992</td>
</tr>
<tr>
<td>Credit Rating Sample</td>
<td>0.639</td>
<td>0.064</td>
<td>7.875</td>
<td>0.420</td>
<td>3.435</td>
<td>1.019</td>
</tr>
<tr>
<td>Non-Downgrades</td>
<td>0.449</td>
<td>0.146</td>
<td>8.550</td>
<td>0.333</td>
<td>2.933</td>
<td>1.068</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Panel B: Sorted by Risk Loading</th>
<th>Sharpe Ratio</th>
<th>Skewness</th>
<th>Excess Kurtosis</th>
<th>Maximum Drawdown</th>
<th>Return in Crisis</th>
<th>Turnover</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Sample</td>
<td>1.225</td>
<td>1.063</td>
<td>5.932</td>
<td>0.209</td>
<td>0.472</td>
<td>1.775</td>
</tr>
<tr>
<td>Non-Microcaps</td>
<td>0.839</td>
<td>0.326</td>
<td>1.582</td>
<td>0.246</td>
<td>0.677</td>
<td>1.721</td>
</tr>
<tr>
<td>Credit Rating Sample</td>
<td>0.566</td>
<td>0.267</td>
<td>1.440</td>
<td>0.407</td>
<td>-0.023</td>
<td>1.758</td>
</tr>
<tr>
<td>Non-Downgrades</td>
<td>0.602</td>
<td>0.344</td>
<td>1.675</td>
<td>0.447</td>
<td>0.903</td>
<td>1.795</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Panel C: CRSP Market Index</th>
<th>Sharpe Ratio</th>
<th>Skewness</th>
<th>Excess Kurtosis</th>
<th>Maximum Drawdown</th>
<th>Return in Crisis</th>
<th>Turnover</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market</td>
<td>0.516</td>
<td>-0.991</td>
<td>3.334</td>
<td>0.481</td>
<td>-7.051</td>
<td>0.093</td>
</tr>
</tbody>
</table>

- ML methods: require high turnover in portfolio rebalancing → reduce the gross returns based on GKX (CPZ) method by at least 0.5% (0.85%)
- ML methods: require high turnover in portfolio rebalancing → reduce the gross returns based on GKX method by at least 0.5%
ML methods: require high turnover in portfolio rebalancing
→ reduce the gross returns based on CPZ method by at least 0.85%
An alternative ML Method

- CPZ: estimate SDF for individual stocks
- Kozak, Nagel, and Santosh (2018) (KNS): estimate SDF for equity portfolios, i.e., long-short portfolio return based on predictive characteristics
- Minimize the Hansen-Jagannathan (1991) distance
- Ridge regression with three-fold cross-validation
- Apply the 94 characteristics in GKW
- In-sample estimation: 1964 to 2004
- Out-of-sample test: 2005 to 2017
### Characteristics of SDF-Implied MVE Portfolios

<table>
<thead>
<tr>
<th></th>
<th>CAPM</th>
<th>FF6</th>
<th>Sharpe Ratio</th>
<th>SDF-Implied MVE Portfolio Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>10%</td>
<td>25%</td>
<td>Median</td>
</tr>
<tr>
<td>Full Sample</td>
<td>3.662*** (6.01)</td>
<td>3.338*** (5.90)</td>
<td>2.318</td>
<td>0.083</td>
</tr>
<tr>
<td>Non-Microcaps</td>
<td>1.543*** (3.88)</td>
<td>0.895*** (2.87)</td>
<td>0.977</td>
<td>0.084</td>
</tr>
<tr>
<td>Credit Rating Sample</td>
<td>1.418*** (2.97)</td>
<td>0.717* (1.93)</td>
<td>0.898</td>
<td>-0.006</td>
</tr>
<tr>
<td>Non-Downgrades</td>
<td>1.308*** (2.92)</td>
<td>0.545   (1.59)</td>
<td>0.828</td>
<td>-0.022</td>
</tr>
</tbody>
</table>

- Imposing economic restrictions reduces performance, as well as odds of extreme positions
- ML techniques face the usual challenge of cross-sectional predictability: profitability evolves from difficult-to-arbitrage stocks and there are sizable trading costs
- .. and we have not yet addressed the **time-series** channel
Time-Varying Return Predictability: GKX FF6 Alpha

- Binding limits to arbitrage → more profitable anomaly-based trading strategies
  - High sentiment, high volatility, and low liquidity
Time-Varying Return Predictability: GKX

\[ \text{HML}_t = \alpha_0 + \beta_1 \text{High SENT}_{t-1} + \beta_2 \text{High MKTVOL}_{t-1} + \beta_3 \text{High MKTILLIQ}_{t-1} + \beta_4 M_{t-1} + c'F_t + e_t \]

<table>
<thead>
<tr>
<th></th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
<th>Model 5</th>
<th>Model 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>1.275***</td>
<td>0.946***</td>
<td>0.016</td>
<td>-0.453</td>
<td>0.869</td>
<td>1.250</td>
</tr>
<tr>
<td></td>
<td>(3.88)</td>
<td>(2.99)</td>
<td>(0.03)</td>
<td>(-0.92)</td>
<td>(1.13)</td>
<td>(1.40)</td>
</tr>
<tr>
<td>1M HML_{t-1}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.181***</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3.03)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12M HML_{t-12:t-1}</td>
<td></td>
<td></td>
<td>0.398*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(1.94)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High SENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.534**</td>
<td>1.710**</td>
<td></td>
<td>0.252</td>
<td></td>
<td>0.026</td>
</tr>
<tr>
<td></td>
<td>(2.43)</td>
<td>(2.51)</td>
<td></td>
<td>(0.64)</td>
<td></td>
<td>(0.06)</td>
</tr>
<tr>
<td>High MKTVOL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.791</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(1.24)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High VIX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.851***</td>
<td>1.621***</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(2.85)</td>
<td>(3.19)</td>
<td></td>
</tr>
<tr>
<td>High MKTILLIQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.754</td>
<td>0.529</td>
<td>0.608</td>
<td>0.714</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(1.24)</td>
<td>(0.78)</td>
<td>(1.41)</td>
<td>(1.45)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Controls</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
Time-Varying Return Predictability: CPZ

\[ HML_t = \alpha_0 + \beta_1 High SENT_{t-1} + \beta_2 High MKTVOL_{t-1} + \beta_3 High MKTILLIQ_{t-1} + \beta_4 M_{t-1} + c'F_t + e_t \]

<table>
<thead>
<tr>
<th></th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
<th>Model 5</th>
<th>Model 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>2.006***</td>
<td>1.076**</td>
<td>0.103</td>
<td>-0.081</td>
<td>0.424</td>
<td>0.523</td>
</tr>
<tr>
<td></td>
<td>(5.79)</td>
<td>(2.52)</td>
<td>(0.16)</td>
<td>(-0.13)</td>
<td>(0.31)</td>
<td>(0.35)</td>
</tr>
<tr>
<td>1M HML(_{t-1})</td>
<td>0.076</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(1.05)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12M HML(_{t-12:t-1})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.503***</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3.51)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High SENT</td>
<td></td>
<td></td>
<td>1.412**</td>
<td>1.395**</td>
<td>1.170*</td>
<td>1.134*</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(2.32)</td>
<td>(2.23)</td>
<td>(1.93)</td>
<td>(1.77)</td>
</tr>
<tr>
<td>High MKTVOL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.787</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(1.29)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High VIX</td>
<td></td>
<td></td>
<td></td>
<td>1.255**</td>
<td></td>
<td>1.559**</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(2.09)</td>
<td></td>
<td>(2.27)</td>
</tr>
<tr>
<td>High MKTILLIQ</td>
<td></td>
<td></td>
<td></td>
<td>1.828***</td>
<td>1.918***</td>
<td>1.614**</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(2.86)</td>
<td>(2.89)</td>
<td>(2.21)</td>
</tr>
<tr>
<td>Controls</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
Unlike individual anomalies, GKX strategy remains viable in recent years.
Unlike individual anomalies, GGX strategy remains viable in recent years.
Stock Characteristics of ML Portfolios

- Both ML methods identify stocks in line with most anomaly-based trading strategies.
- Long positions: small, value, illiquid and old stocks with low price, low beta, low 1-month return, high 11-month return, low asset growth, low equity issuance, high ROA and ROE, low credit rating coverage, low analyst coverage, and high earnings surprise.
- Robust to sub-periods and market states
Intra-Industry vs. Inter-Industry Return Predictability

• Intra-industry strategy accounts for the majority of the unconditional profit → informs on stock selection not industry rotation
Intra-Industry vs. Inter-Industry Return Predictability

- Intra-industry strategy improves performance, especially on non-microcaps.
Potential of ML in Asset Management

- Mitigate the downside risk and hedge against crisis
- Remain profitable in recent years
- Profitable in long positions: e.g., GKY signal, exclude microcaps + VW
Conclusion

- ML-based investments extract profitability primarily from difficult-to-arbitrage stocks and during high volatility and illiquidity market states.
- Despite their opaque nature, ML methods identify mispriced stocks consistent with most anomalies.
- Beyond economic restrictions, ML signals are profitable in long positions, remain viable in recent years, and command low downside risk.
References


Borisenko, Dmitry., 2019, Dissecting Momentum: We Need to Go Deeper. Working paper.


Feng, Guanhao, Stefano Giglio, and Dacheng Xiu, 2017, Taming the factor zoo, Working Paper.

Feng, G., S. Giglio, and D. Xiu , 2019 , Taming the factor zoo. Forthcoming in *Journal of Finance*.


Lettau, M., and M. Pelger. 2018b. Factors that fit the time series and cross-section of stock returns. Working Paper


