Forecast Combination and Multiple Testing

The Econometrics of Predictability

This version: May 12, 2014

May 13, 2014

UNIVERSITY OF OXFORD

Overview

- Model Combination
- Multiple Hypothesis Testing (2 weeks)

The Standard Forecasting Model

- Standard forecasts are also popular for predicting economic variables
- Generically expressed

$$y_{t+1} = \beta_0 + \mathbf{x}_t \boldsymbol{\beta} + \epsilon_{t+1}$$

- $\mathbf{x}_t$ is a 1 by k vector of predictors ($k = 1$ is common)
- Includes both exogenous regressors such as the term or default premium and also autoregressive models
- Forecasts are $\hat{y}_{t+1|t}$

The forecast combination problem

- Two level of aggregation in the combination problem

1. Summarize individual forecasters' private information in point forecasts $\hat{y}_{t+h,i|t}$
    - Highlights that "inputs" are not the usual explanatory variables, but forecasts
2. Aggregate individual forecasts into consensus measure $C\left(\mathbf{y}_{t+h|t}, \mathbf{w}_{t+h|t}\right)$

- Obvious competitor is the "super-model" or "kitchen-sink" – a model built using all information in each forecasters information set
- Aggregation should increase the bias in the forecast relative to SM but may reduce the variance
- Similar to other model selection procedures in this regard

Why not use the “Super Model”

- Could consider pooling information sets

$$\mathcal{F}_t^c = \cup_{i=1}^n \mathcal{F}_{t,i}$$

- Would contain all information available to all forecasters
- Could construct consensus directly $C\left(\mathcal{F}_t^c; \boldsymbol{\theta}_{t+h|t}\right)$
- Some reasons why this may not work
  - Some information in individuals information sets may be qualitative, and so expensive to quantitatively share
  - Combined information sets may have a very high dimension, so that finding the best super model may be hard
    - Potential for lots of estimation error
- Classic bias-variance trade-off is main reason to consider forecasts combinations over a super model
  - Higher bias, lower variance

Linear Combination under MSE Loss

- Models can be combined in many ways for virtually any loss function
- Most standard problem is for MSE loss using only linear combinations
- I will suppress time subscripts when it is clear that it is $t+h|t$
- Linear combination problem is

$$\min_{\mathbf{w}} \mathrm{E}\left[e^2\right] = \mathrm{E}\left[\left(y_{t+h} - \mathbf{w}'\hat{\mathbf{y}}\right)^2\right]$$

- Requires information about first 2 moments of he joint distribution of the realization y_{t+h} and the time-t forecasts $\hat{\mathbf{y}}$

$$\left[\begin{array}{c} y_{t+h|t} \\ \hat{\mathbf{y}} \end{array}\right] \sim F\left(\left[\begin{array}{c} \mu_y \\ \boldsymbol{\mu}_{\hat{\mathbf{y}}} \end{array}\right], \left[\begin{array}{cc} \sigma_{yy} & \boldsymbol{\Sigma}'_{y\hat{\mathbf{y}}} \\ \boldsymbol{\Sigma}_{y\hat{\mathbf{y}}} & \boldsymbol{\Sigma}_{\hat{\mathbf{y}}\hat{\mathbf{y}}} \end{array}\right]\right)$$

Linear Combination under MSE Loss

UNIVERSITY OF OXFORD

- The first order condition for this problem is

$$\frac{\partial \mathrm{E}\left[e^2\right]}{\partial \mathbf{w}} = -\mu_y \boldsymbol{\mu}_{\hat{\mathbf{y}}} + \boldsymbol{\mu}_{\hat{\mathbf{y}}} \boldsymbol{\mu}_{\hat{\mathbf{y}}}' \mathbf{w} + \boldsymbol{\Sigma}_{\hat{\mathbf{y}}\hat{\mathbf{y}}} \mathbf{w} - \boldsymbol{\Sigma}_{y\hat{\mathbf{y}}} = \mathbf{0}$$

- The solution to this problem is

$$\mathbf{w}^{\star} = \left(\boldsymbol{\mu}_{\hat{\mathbf{y}}} \boldsymbol{\mu}_{\hat{\mathbf{y}}}' + \boldsymbol{\Sigma}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}\right)^{-1} \left(\boldsymbol{\Sigma}_{y\hat{\mathbf{y}}} + \mu_y \boldsymbol{\mu}_{\hat{\mathbf{y}}}\right)$$

- Similar to the solution to the OLS problem, only with extra terms since the forecasts may not have the same conditional mean

Linear Combination under MSE Loss

- Can remove the conditional mean if the combination is allowed to include a constant, w_c

$$\begin{aligned} w_c &= \mu_y - \mathbf{w}^{\star}\boldsymbol{\mu}_{\hat{\mathbf{y}}} \\ \mathbf{w}^{\star} &= \boldsymbol{\Sigma}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1}\boldsymbol{\Sigma}_{y\hat{\mathbf{y}}} \end{aligned}$$

- These are identical to the OLS where w_c is the intercept and $\mathbf{w}^{*}$ are the slope coefficients
- The role of w_c is the correct for any biases so that the squared bias term in the MSE is 0

$$\mathrm{MSE}\left[e\right] = \mathrm{B}\left[e\right]^2 + \mathrm{V}\left[e\right]$$

Understanding the Diversification Gains

- Simple setup

$$e_1 \sim F_1\left(0, \sigma_1^2\right),\ e_2 \sim F_2\left(0, \sigma_2^2\right),\ \operatorname{Corr}[e_1, e_2] = \rho,\ \operatorname{Cov}[e_1 e_2] = \sigma_{12}$$

- Assume $\sigma_2^2 \leq \sigma_1^2$
- Assume weights sum to 1 so that $w_1 = 1 - w_2$ (Will suppress the subscript and simply write w)
- Forecast error is then

$$y - w\hat{y}_1 - (1 - w)\,\hat{y}_2$$

- Error is given by

$$e^c = we_1 + (1 - w)\,e_2$$

- Forecast has mean 0 and variance

$$w^2\sigma_1^2 + (1 - w)^2\,\sigma_2^2 + 2w\,(1 - w)\,\sigma_{12}$$

Understanding the Diversification Gains

- The optimal w can be solved by minimizing this expression, and is

$$w^\star = \frac{\sigma_2^2 - \sigma_{12}}{\sigma_1^2 + \sigma_2^2 - 2\sigma_{12}}, \;\; 1 - w^\star = \frac{\sigma_1^2 - \sigma_{12}}{\sigma_1^2 + \sigma_2^2 - 2\sigma_{12}}$$

- Intuition is that the weight on a model is higher the:
  - Larger the variance of the other model
  - Lower the correlation between the models
- 1 weight will be larger than 1 if $\rho \geq \frac{\sigma_2}{\sigma_1}$
- Weights will be equal if $\sigma_1 = \sigma_2$ for any value of correlation
  - Intuitively this must be the case since model 1 and 2 are indistinguishable from a MSE point-of-view
  - When will "optimal" combinations out-perform equally weighted combinations? Any time $\sigma_1 \neq \sigma_2$
- If $\rho = 1$ then only select model with lowest variance (mathematical formulation is not well posed in this case)

Constrained weights

- The previous optimal weight derivation did not impose any restrictions on the weights
- In general some of the weights will be negative, and some will exceed 1
- Many combinations are implemented in a relative, constrained scheme

$$\min_{\mathbf{w}} \mathrm{E}\left[e^2\right] = \mathrm{E}\left[\left(y_{t+h} - \mathbf{w}'\hat{\mathbf{y}}\right)^2\right] \text{ subject to } \mathbf{w}'\boldsymbol{\iota} = 1$$

- The intercept is omitted (although this isn't strictly necessary)
- If the biases are all 0, then the solution is dual to the usual portfolio minimization problem, and is given by

$$\mathbf{w}^{\star} = \frac{\boldsymbol{\Sigma}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1}\boldsymbol{\iota}}{\boldsymbol{\iota}'\boldsymbol{\Sigma}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}^{-1}\boldsymbol{\iota}}$$

- This solution is the same as the Global Minimum Variance Portfolio

Combinations as Hedge against Structural Breaks

- One often cited advantage of combinations is (partial) robustness to structural breaks
- Best case is if two positively correlated variables have shifts in opposite directions
- Combinations have been found to be more stable than individual forecasts
    - This is mostly true for static combinations
    - Dynamic combinations can be unstable since some models may produce large errors from time-to-time

Weight Estimation

- All discussion has focused on "optimal" weights, which requires information on the mean and covariance of both y_{t+h} and $\hat{\mathbf{y}}_{t+h|t}$
  - This is clearly highly unrealistic
- In practice weights must be estimated, which introduces extra estimation error
- Theoretically, there should be no need to combine models when all forecasting models are generated by the econometrician (e.g. when using $\mathcal{F}^c$)
- In practice, this does not appear to be the case
  - High dimensional search space for "true" model
  - Structural instability
  - Parameter estimation error
  - Correlation among predictors

Clemen (1989): "Using a combination of forecasts amounts to an admission that the forecaster is unable to build a properly specified model"

Weight Estimation

- Whether a combination is needed is closely related to forecast encompassing tests
- Model averaging can be thought of a method to avoid the risk of model selection
  - Usually important to consider models with a wide range of features and many different model selection methods
- Has been consistently documented that *prescreening* models to remove the worst performing is important before combining
- One method is to use the SIC to remove the worst models
  - Rank models by SIC, and then keep the x% best
- Estimated weights are usually computed in a 3rd step in the usual procedure
  - R: Regression
  - P: Prediction
  - S: Combination estimation
  - $T = P + R + S$
- Many schemes have been examined

Weight Estimation

- Standard least squares with an intercept

$$y_{t+h} = w_0 + \mathbf{w}'\hat{\mathbf{y}}_{t+h|t} + \epsilon_{t+h}$$

- Least squares without an intercept

$$y_{t+h} = \mathbf{w}'\hat{\mathbf{y}}_{t+h|t} + \epsilon_{t+h}$$

- Linearly constrained least squares

$$y_{t+h} - \hat{y}_{t+h,n|t} = \sum_{i=1}^{n-1} w_i \left(\hat{y}_{t+h,i|t} - \hat{y}_{t+h,n|t}\right) + \epsilon_{t+h}$$

  - This is just a constrained regression where $\sum w_i = 1$ has been implemented where $w_n = 1 - \sum_{i=1}^{n-1} w_i$
  - Imposing this constraint is thought to help when the forecast is persistent

$$e^c_{t+h|t} = -w_0 + \left(1 - \mathbf{w}'\boldsymbol{\iota}\right) y_{t+h} + \mathbf{w}'\mathbf{e}_{t+h|t}$$

  - $\mathbf{e}_{t+h|t}$ are the forecasting errors from the n models
  - Only matters if the forecasts may be biased

Weight Estimation

- Constrained least squares

$$y_{t+h} = \mathbf{w}'\hat{\mathbf{y}}_{t+h|t} + \epsilon_{t+h} \text{ subject to } \mathbf{w}'\boldsymbol{\iota}=1,\ w_i \geq 0$$

  - This is not a standard regression, but can be easily solved using quadratic programming (MATLAB `quadprog`)
- Forecast combination where the covariance of the forecast errors is assumed to be diagonal
  - Produces weights which are all between 0 and 1
  - Weight on forecast i is

$$w_i = \frac{\frac{1}{\sigma_i^2}}{\sum_{j=1}^{n} \frac{1}{\sigma_j^2}}$$

  - May be far from optimal if ρ is large
  - Protects against estimator error in the covariance

Weight Estimation

- Median
  - Can use the median rather than the mean to aggregate
  - Robust to outliers
  - Still suffers from not having any reduction in parameter variance in the actual forecast
- Rank based schemes
  - Weights are inversely proportional to model's rank

$$w_i = \frac{\mathcal{R}_{t+h,i|t}^{-1}}{\sum_{j=1}^{n} \mathcal{R}_{t+h,j|t}^{-1}}$$

  - Highest weight to best model, ratio of weights depends only on relative ranks
  - Places relatively high weight on top model
- Probability of being the best model-based weights
  - Count the proportion that model *i* outperforms the other models

$$\begin{aligned} p_{t+h,i|t} &= T^{-1} \sum_{t=1}^{T} \cap_{j=1,j\neq i}^{n} I\left[L\left(e_{t+h,i|t}\right) < L\left(e_{t+h,j|t}\right)\right] \\ y_{t+h|t}^{c} &= \sum_{i=1}^{n} p_{t+h,i|t} \hat{y}_{t+h,i|t} \end{aligned}$$

Broad Recommendations

- Simple combinations are difficult to beat
  - $1/n$ often outperforms estimated weights
  - Constant usually beat dynamic
  - Constrained outperform unconstrained (when using estimated weights)
- Not combining and using the best fitting performs worse than combinations – often substantially
- Trimming bad models prior to combining improves results
- Clustering similar models (those with the highest correlation of their errors) *prior* to combining leads to better performance, especially when estimating weights
  - Intuition: Equally weighted portfolio of models with high correlation, weight estimation using a much smaller set with lower correlations
- Shrinkage improves weights when estimated
- If using dynamic weights, shrink towards static weights

Equal Weighting

- Equal weighting is hard to beat when the variance of the forecast errors are similar
- If the variance are highly heterogeneous, varying the weights is important
    - If for nothing else than to down-weight the forecasts with large error variances
- Equally weighted combinations are thought to work well when models are unstable
    - Instability makes finding “optimal” weights very challenging
- Trimmed equally-weighted combinations appear to perform better than equally weighted, at least if there are some very poor models
    - May be important to trim both “good” and “bad” models (in-sample performance)
        - Good models are over-fit
        - Bad models are badly mis-specified

Shrinkage Methods

- Linear combination

  $$\hat{y}^c_{t+h|t} = \mathbf{w}'\hat{\mathbf{y}}_{t+h|t}$$

  Standard least squares estimates of combination weights are very noisy
- Often found that "shrinking" the weights toward a *prior* improves performance
- Standard prior is that $w_i = \frac{1}{n}$
- However, do not want to be *dogmatic* and so use a distribution for the weights
- Generally for an arbitrary *prior weight* $\mathbf{w}_0$,

  $$\mathbf{w}|\tau^2 \sim N\left(\mathbf{w}_0, \boldsymbol{\Omega}\right)$$

- $\boldsymbol{\Omega}$ is a correlation matrix and τ^2 is a parameter which controls the amount of shrinkage

Shrinkage Methods

- Leads to a weighted average of the prior and data

$$\bar{\mathbf{w}} = \left(\mathbf{\Omega} + \hat{\mathbf{y}}'\hat{\mathbf{y}}\right)^{-1}\left(\mathbf{\Omega}\mathbf{w}_0 + \hat{\mathbf{y}}'\hat{\mathbf{y}}\hat{\mathbf{w}}\right)$$

- $\hat{\mathbf{w}}$ is the usual least squares estimator of the optimal combination weight
- If $\mathbf{\Omega}$ is very large compared to $\mathbf{y}'\mathbf{y} = \sum_{t=1}^{T}\mathbf{y}_{t+h|t}\mathbf{y}'_{t+h|t}$ then $\bar{\mathbf{w}} \approx \mathbf{w}_0$
- On the other hand, if $\mathbf{y}'\mathbf{y}$ dominates, then $\bar{\mathbf{w}} \approx \hat{\mathbf{w}}$
- Other implementation use a g-prior, which is scalar

$$\bar{\mathbf{w}} = \left(g\hat{\mathbf{y}}'\hat{\mathbf{y}} + \hat{\mathbf{y}}'\hat{\mathbf{y}}\right)^{-1}\left(g\hat{\mathbf{y}}'\hat{\mathbf{y}}\mathbf{w}_0 + \hat{\mathbf{y}}'\hat{\mathbf{y}}\hat{\mathbf{w}}\right)$$

- Large values of $g \geq 0$ least to large amounts of shrinkage
- 0 corresponds to OLS

$$\bar{\mathbf{w}} = \mathbf{w}_0 + \frac{\hat{\mathbf{w}} - \mathbf{w}_0}{1+g}$$

Inference for Many Forecasts

- Six papers:
  - White, H. “A reality check for data snooping”. *Econometrica*
  - Hansen, P. “A Test for Superior Predictive Ability”. *JBES*
  - Sullivan, Timmermann & White. “Data-Snooping, Technical Trading Rule Performance, and the Bootstrap”. *Journal of Finance*
  - Romano & Wolf. “Stepwise Multiple Testing as Formalized Data Snooping”. *Econometrica*
  - Hansen, Lunde & Nason. “The Model Confidence Set”. *Econometrica*
  - Bajgrowicz & Scaillet. “Technical trading revisited: false discoveries, persistence tests and transaction costs”. *Journal of Financial Economics*

Diebold-Mariano-West

- The Diebold-Mariano-West test examines whether two forecasts have equal predictive ability
- DMW tests are all based on the difference of two loss functions

$$\delta_t = L\left(y_{t+h}, \hat{y}^A_{t+h|t}\right) - L\left(y_{t+h}, \hat{y}^B_{t+h|t}\right)$$

- The test statistic is based on the asymptotic normality of $\bar{\delta} = P^{-1}\sum_{t=R+1}^{T}\delta_t$
- If $P/R \to 0$ then

$$\sqrt{P}\left(\bar{\delta} - \mathrm{E}\left[\delta\right]\right) \xrightarrow{d} N\left(0, \sigma^2\right)$$

- σ^2 is the long-run variance, that is

$$\sigma^2 = \lim_{P\to\infty} \mathrm{V}\left[P^{-\frac{1}{2}}\sum_{t=R+1}^{T}\delta_t\right]$$

- Must account for autocovariances, so a HAC estimator is used (Newey-West)

DMW with the Bootstrap

- Alternatively could estimate the variance using the bootstrap
- For example, the stationary bootstrap could be used as long as the window length grows with the size of the evaluation sample
- To implement the stationary bootstrap, the loss differentials would be directly re-sampled to construct $\bar{\delta}_b^*$ for $b = 1, \ldots, B$
- The variance would then be computed as

$$\hat{\sigma}^2_{BS} = \frac{P}{B} \sum_{b=1}^{b} \left(\bar{\delta}_b^\star - \bar{\delta} \right)^2$$

- The test statistic is then

$$DMW = \frac{\bar{\delta}}{\sqrt{\hat{\sigma}^2_{BS}}}$$

  - Note: the $\sqrt{P}$ term is implicit in the denominator since σ^2_{BS} will decline as the sample size grows ($\hat{\sigma}^2_{BS} \approx \hat{\sigma}^2 / P$)

DMW using percentile method

- Alternatively, inference could be made using the percentile method
- To implement the percentile method, it is necessary to enforce the null $H_0 : \mathrm{E}[\delta_t] = 0$
- This can be done by re-centering the loss differentials around the average in the data: $\tilde{\delta}_t = \delta_t - \bar{\delta}$
- The centered loss differentials $\tilde{\delta}_t$ could then be re-sampled to compute an estimate of the average loss-differential $\bar{\tilde{\delta}}_b^\star$
- Inference using the percentile method would be based on the empirical frequency where $\bar{\delta} < \bar{\tilde{\delta}}_b^\star$ or $\bar{\delta} > \bar{\tilde{\delta}}_b^\star$

DMW using percentile method

- Since the test is 2-sided$||$

$$2 \times B^{-1} \sum_{b=1}^{B} I\left[\left|\bar{\bar{\delta}}_b^{\star}\right| < \left|\bar{\delta}\right|\right]$$

  - If many of the re-sampled centered means are less then $\bar{\delta}$, then the loss differential does not appear large
  - If few of the re-sampled centered means are less than $\bar{\delta}$, then the loss differential appears large
- Since the distribution is asymptotically normal, there is no need to use the percentile method since the bootstrap t-stat is simple to construct

Reality Check

- The *Reality Check* extends DMW to testing for *Superior Predictive Ability* (SPA)
- Tests of SPA examine whether a set of forecasting models can outperform a benchmark
- Suppose forecasts were available for m forecasts, $j = 1, \ldots, m$
- The vector of loss differentials *relative to a benchmark* could be constructed as

$$\boldsymbol{\delta}_t = \left[\begin{array}{c} L\left(y_{t+h}, \hat{y}_{t+h,BM|t}\right) - L\left(y_{t+h}, \hat{y}_{t+h,1|t}\right) \\ L\left(y_{t+h}, \hat{y}_{t+h,BM|t}\right) - L\left(y_{t+h}, \hat{y}_{t+h,2|t}\right) \\ \vdots \\ L\left(y_{t+h}, \hat{y}_{t+h,BM|t}\right) - L\left(y_{t+h}, \hat{y}_{t+h,m|t}\right) \end{array} \right]$$

- $\hat{y}_{t+h,BM|t}$ is the loss from the *benchmark forecast*

Asymptotic distribution in the RC

- Under similar arguments as in Diebold & Mariano and West,

$$\sqrt{P}\left(\bar{\boldsymbol{\delta}} - \mathrm{E}\left[\bar{\boldsymbol{\delta}}\right]\right) \xrightarrow{d} N\left(\mathbf{0}, \boldsymbol{\Sigma}\right)$$

- $\boldsymbol{\Sigma}$ is the asymptotic covariance matrix of the average loss differentials

$$\boldsymbol{\Sigma} = \lim_{P \to \infty} \mathrm{V}\left[P^{-\frac{1}{2}} \sum_{t=R+1}^{T} \boldsymbol{\delta}_t\right]$$

- This looks virtually identical to the case of the univariate DMW test

Hypotheses of SPA

- If the benchmark model is as good as the other models, then the mean of each element of $\boldsymbol{\delta}_t$ should be 0 *or* negative
  - These are *losses*, so if the BM is better, then its loss is smaller then the loss from the other model
- A total of m models
- The null in a test of SPA is

$$H_0 : \max_{j=1,\ldots,m} \left(\mathrm{E}\left[\delta_{j,t}\right]\right) \leq 0$$

- The alternative is the natural one,

$$H_1 : \max_{j=1,\ldots,m} \left(\mathrm{E}\left[\delta_{j,t}\right]\right) > 0$$

- ***Note****: If no models are statistically better than the benchmark, then there is no point in implementing the RC*

Examples of SPA: MSE

- The standard example is for comparing models using MSE (or MAE, or similar)

$$L\left(y_{t+h}, \hat{y}_{t+h,j|t}\right) = \left(y_{t+h} - \hat{y}_{t+h,j|t}\right)^2$$

- The vector of loss differentials is then

$$\boldsymbol{\delta}_t = \left[\begin{array}{c} \left(y_{t+h} - \hat{y}_{t+h,BM|t}\right)^2 - \left(y_{t+h} - \hat{y}_{t+h,1|t}\right)^2 \\ \left(y_{t+h} - \hat{y}_{t+h,BM|t}\right)^2 - \left(y_{t+h} - \hat{y}_{t+h,2|t}\right)^2 \\ \vdots \\ \left(y_{t+h} - \hat{y}_{t+h,BM|t}\right)^2 - \left(y_{t+h} - \hat{y}_{t+h,m|t}\right)^2 \end{array}\right]$$

- This is the simplest form of an SPA test

Examples of SPA: Return Predictability

- SPA can also be used to test whether the returns of a set of trading models are equal
- In this case the "loss" function is the *negative* of the return from the strategy

$$L\left(y_{t+h}, \hat{y}_{t+h,j|t}\right) = -\ln\left(1 + y_{t+h} S\left(\hat{y}_{t+h,j|t}\right)\right)$$

- $S\left(\hat{y}_{t+h,j|t}\right)$ is a signal which indicates the size of the portfolio
  - y_{t+h} is the holding period return of the asset
  - Could be -1, 0, 1 for short, out, long strategies
  - $\hat{y}_{t+h,j|t}$ is the input for the signal function, e.g. a Moving Average Oscillator
- The vector of loss differentials is then

$$\boldsymbol{\delta}_t = \left[\begin{array}{c} \ln\left(1 + y_{t+h} S\left(\hat{y}_{t+h,1|t}\right)\right) - \ln\left(1 + y_{t+h} S\left(\hat{y}_{t+h,BM|t}\right)\right) \\ \vdots \\ \ln\left(1 + y_{t+h} S\left(\hat{y}_{t+h,m|t}\right)\right) - \ln\left(1 + y_{t+h} S\left(\hat{y}_{t+h,BM|t}\right)\right) \end{array}\right]$$

- The benchmark could be a simple strategy, e.g. buy-and-hold ($S(\cdot) = 1$)
- Ultimately the "loss differential" is the difference between the returns of a set of strategies and the benchmark strategy

Example: Predictive Likelihood

- SPA can be used to test distribution fit
- The loss function is just the *negative* of the likelihood

$$L\left(y_{t+h},\hat{y}_{t+h,j|t}\right)=-l_j\left(y_{t+h}|\hat{y}_{t+h,j|t}\right)$$

  - $\hat{y}_{t+h,j|t}$ contains any time-t information needed to compute the log-likelihood

- The vector of loss differentials is then

$$\boldsymbol{\delta}_t=\left[\begin{array}{c} l_1\left(y_{t+h}|\hat{y}_{t+h,1|t}\right)-l_{BM}\left(y_{t+h}|\hat{y}_{t+h,BM|t}\right)\\ l_2\left(y_{t+h}|\hat{y}_{t+h,2|t}\right)-l_{BM}\left(y_{t+h}|\hat{y}_{t+h,BM|t}\right)\\ \vdots\\ l_m\left(y_{t+h}|\hat{y}_{t+h,m|t}\right)-l_{BM}\left(y_{t+h}|\hat{y}_{t+h,BM|t}\right)\end{array}\right]$$

- The benchmark could be a simple strategy, e.g. buy-and-hold ($S\left(\cdot\right)=1$)
- Ultimately the differential is just the difference between the returns of a set of strategies and the benchmark strategy

Example: α from a multifactor model

- Suppose you were interested in testing for excess performance
- Usual APT type regression

$$r_{j,t}^{e} = \alpha_j + \mathbf{f}_t' \boldsymbol{\beta}_j + \epsilon_{j,t}$$

- The "benchmark α" is 0 – the test is implemented directly on the estimated αs
- Loss function is just $-\hat{\alpha}$ (*negative* excess performance)
- The vector of loss differentials is then

$$\boldsymbol{\delta}_t = \begin{bmatrix} r_{1,t}^{e} - \mathbf{f}_t' \hat{\boldsymbol{\beta}}_1 \\ \vdots \\ r_{m,t}^{e} - \mathbf{f}_t' \hat{\boldsymbol{\beta}}_m \end{bmatrix} = \begin{bmatrix} \hat{\alpha}_1 + \hat{\epsilon}_{1,t} \\ \vdots \\ \hat{\alpha}_m + \hat{\epsilon}_{m,t} \end{bmatrix}$$

- Used to test fund manager skill

Implementing the Reality Check

- The Reality Check is implemented using the P by m matrix of loss differentials
  - P out-of-sample periods
  - m models
- The original article describes two methods
  - Monte Carlo Reality Check
  - Bootstrap Reality Check
- In practice, only the Bootstrap Reality Check is used
- The distribution of the *maximum* of normals is not normal, and so only the percentile method is applicable

Implementing the Reality Check

Algorithm (Bootstrap Reality Check)

1. *Compute* $T^{RC} = \max\left(\bar{\boldsymbol{\delta}}\right)$
2. *For* $b = 1, \ldots, B$ *re-sample the vector of loss differentials* $\boldsymbol{\delta}_t$ *to construct a bootstrap sample* $\left\{\boldsymbol{\delta}^{\star}_{b,t}\right\}$ *using the stationary bootstrap*
3. *Using the bootstrap sample, compute*

$$T_b^{\star RC} = \max\left(P^{-1}\sum_{t=R+1}^{T}\boldsymbol{\delta}^{\star}_{b,t} - \bar{\boldsymbol{\delta}}\right)$$

4. *Compute the Reality Check p-value as the percentage of the bootstrapped maxima which are larger than the sample maximum*

$$p - value = B^{-1}\sum_{b=1}^{b} I\left[T_b^{\star RC} > T^{RC}\right]$$

Intuition

- The bootstrap means are like draws (simulation) from the asymptotic distribution $N(\mathbf{0}, \mathbf{\Sigma})$
- Taking the maximum of these draws simulates the distribution of a set of correlated normals
- Each bootstrap mean is centered at the sample mean
  - This is known as using the *Least Favorable Configuration* (LFC) point
  - Simulation is done assuming any model could as good as the benchmark
- Since the asymptotic distribution can be simulated, asymptotic critical values and p-values can be constructed directly
- The Monte Carlo Reality Check works by first estimating $\mathbf{\Sigma}$ using a HAC estimator, and then simulating random normals directly
  - MCRC is equivalent to BRC, only requires estimating:
    - A potentially large covariance is m is big
    - The Choleski decomposition of this covariance
    - B drawn from this Choleski
  - In practice, m may be so large that the covariance matrix won't fit in a normal computer's memory

Revisiting: α from a multifactor model

- The original formulation had

$$\boldsymbol{\delta}_t = \begin{bmatrix} r^e_{1,t} - \mathbf{f}_t' \hat{\boldsymbol{\beta}}_1 \\ \vdots \\ r^e_{m,t} - \mathbf{f}_t' \hat{\boldsymbol{\beta}}_m \end{bmatrix} = \begin{bmatrix} \hat{\alpha}_1 + \hat{\epsilon}_{1,t} \\ \vdots \\ \hat{\alpha}_m + \hat{\epsilon}_{m,t} \end{bmatrix}$$

- Alternatively distribution could be built up by directly re-sampling the returns and factors jointly
- This would allow $T_b^{\star RC} = \max_{j=1,\ldots,m}\left(\alpha^*_{j,b} - \hat{\alpha}_j\right)$ to be computed form a cross-sectional regression in each bootstrap
- Reality check allow for parameter estimation error as long as $(P/R)\ln\ln R \rightarrow 0$ which is similar to $P/R \rightarrow 0$
- Also works if $P/R \rightarrow \infty$, in which case it is essential to re-sample returns and factors and re-estimate $\hat{\boldsymbol{\beta}}^{\star}_{j,b}$ in each bootstrap

Application in Original Paper

- The original paper is applied to the BLL-type trading rules
- Used S&P 500 rather than DJIA
- Constructed 4 types of trading rule primitives:
  - Momentum measures: $\left(p_t - p_{t-j}\right)/p_{t-j}$ for $j \in \{1, \ldots, 11\}$ (11 rules)
  - Trend: $p_{t-i} = \alpha + \beta\left(m - i\right) + \epsilon_j$ for $m \in \{5, 10, 15, 20\}$ day periods (4 rules)
  - Relative strength: $\tau^{-1}\sum_{i=-\tau+1}^{0} I\left[\left(p_{t-i} - p_{t-i-1}\right) > 0\right]$ for $\tau \in \{5, 10, 15, 20\}$(4 rules)
  - Moving average oscillator for fast speeds of $\{1, 5, 10, 15\}$ and slow speeds of $\{5, 10, 15, 20\}$ (10 rules)
    - Note: Slow has to be strictly longer than fast, so a total of $4 + 3 + 2 + 1 = 10$ rules
- All combinations of 3 of these 29 variables were fed into a linear regression to produce forecasts

$$r_{t+1} = \beta_1 + \beta_2 x_{i,t} + \beta_3 x_{j,t} + \beta_4 x_{k,t} + \epsilon_{t+1}$$

- For $i, j, k \in \{1, \ldots, 29\}$ without repetition, so ${}_{29}C_3 = 3654$ rules

Application in Original Paper

- Benchmark is a model which includes only a constant

$$r_{t+1} = \beta_1 + \epsilon_{t+1}$$

- Models compared in terms of MSE

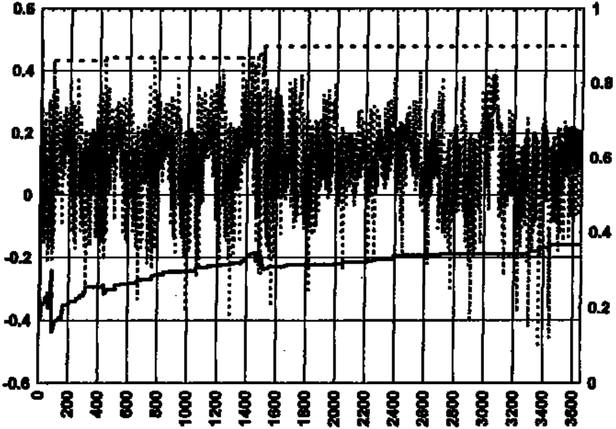
$$L\left(y_{t+1}, \hat{y}_{t+1|t}\right) = \left(y_{t+1} - \hat{\beta}_0 - \hat{\beta}_1 x_{i,t} - \hat{\beta}_2 x_{j,t} - \hat{\beta}_3 x_{k,t}\right)^2$$

- Models also compared in terms of directional accuracy

$$L\left(y_{t+1}, \hat{y}_{t+1|t}\right) = -I\left[y_{t+1}\left(\hat{\beta}_0 + \hat{\beta}_1 x_{i,t} + \hat{\beta}_2 x_{j,t} + \hat{\beta}_3 x_{k,t}\right) > 0\right]$$

  - The negative is used to turn a “good” (same sign) into a “bad”
  - Modification allows application of RC without modification since null is $H_0 : \max\left(\mathrm{E}\left[\delta_{j,t}\right]\right) \leq 0$

MSE Differential



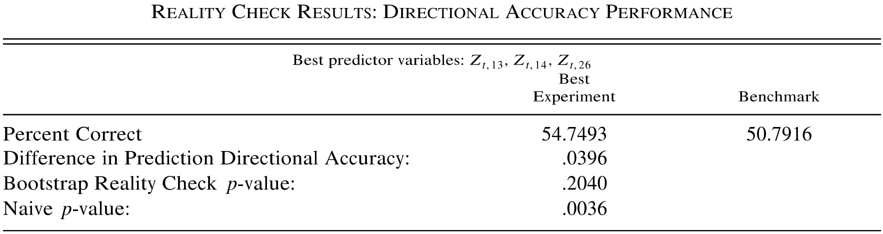
$\max \bar{\delta}_i$

RC P-val

Experiment Number

- Negative MSE differential plotted (higher is better)

Sign Prediction

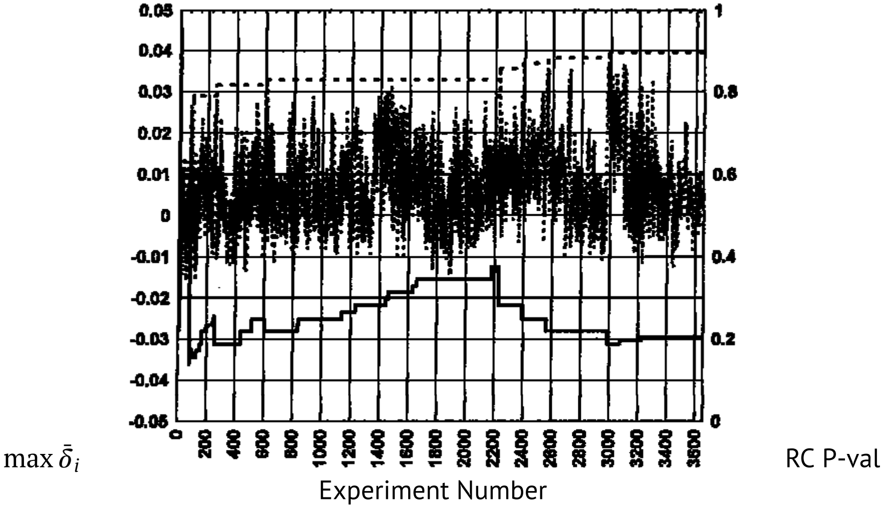


Reality Check Results: Directional Accuracy Performance

Best predictor variables: $Z_{t,13}$, $Z_{t,14}$, $Z_{t,26}$

	Best Experiment	Benchmark
Percent Correct	54.7493	50.7916
Difference in Prediction Directional Accuracy:	.0396	
Bootstrap Reality Check *p*-value:	.2040	
Naive *p*-value:	.0036	

Sign Prediction



Hansen’s Test of SPA

- Hansen (2005, *JBES*) provided two refinements of the RC
  1. Studentized loss differentials
  2. Omission of very bad models from the distribution of the test statistic
- From a practical point-of-view, the first is a very important consideration
- From a theoretical point-of-view, the second is the important issue
  - The second can be ignored if no models are are very poor
  - This may be difficult if using automated model generation schemes

Studentization of Loss Differentials

- The RC uses the loss differentials directly
- This can lead to a loss of power if there is a large amount of cross-sectional heteroskedasticity
- Bad, high variance model can mask a good, low variance model
- The solution is to use the Studentized loss differential
- The test statistic is is based on

$$T^{SPA} = \max_{j=1,\ldots,m}\left(\frac{\bar{\delta}_j}{\sqrt{\hat{\omega}_j^2/P}}\right)$$

- $\hat{\omega}_j^2$ is an estimator of the asymptotic (long-run) variance of $\bar{\delta}_j$

$$\hat{\omega}_j^2 = \hat{\gamma}_{j,0} + 2\sum_{i=1}^{P-1} k_i \hat{\gamma}_{j,i}$$

  - $\gamma_{j,i}$ is the i$^{\text{th}}$ sample autocovariance of the sequence $\{\delta_{j,t}\}$
  - $k_i = \frac{P-i}{P}\left(1-\frac{1}{w}\right)^i + \frac{i}{P}\left(1-\frac{1}{w}\right)^{P-i}$ where w is the window length in Stationary Bootstrap)
- Alternatively use bootstrap variance $\hat{\omega}_j^2 = \frac{P}{B}\sum_{b=1}^{B}\left(\bar{\delta}^{\star}_{b,j} - \bar{\delta}_j\right)^2$

Studentized SPA

Algorithm (Studentized Bootstrap Reality Check)

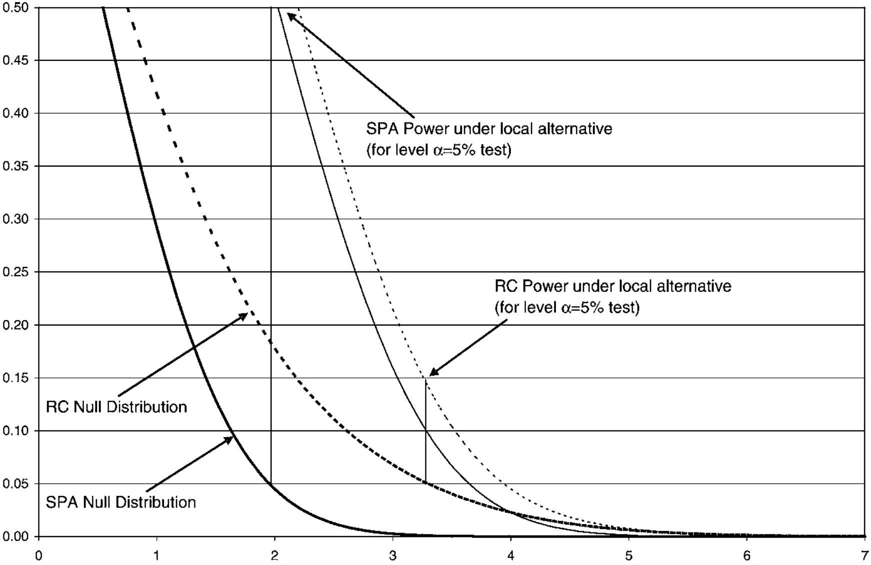
1. *Estimate* $\hat{\omega}_j^2$ *and compute* $T_u^{SPA} = \max\left(\bar{\delta}/\sqrt{\hat{\omega}_j^2/P}\right)$
2. *For* $b = 1, \ldots, B$ *re-sample the vector of loss differentials* $\boldsymbol{\delta}_t$ *to construct a bootstrap sample* $\left\{\boldsymbol{\delta}_{b,t}^\star\right\}$ *using the stationary bootstrap*
3. *Using the bootstrap sample, compute*

$$T_{u,b}^{\star SPA} = \max\left(\frac{P^{-1}\sum_{t=R+1}^{T}\delta_{j,b,t}^\star - \bar{\delta}_j}{\sqrt{\hat{\omega}_j^2/P}}\right)$$

4. *Compute the Studentized Reality Check p-value as the percentage of the bootstrapped maxima which are larger than the sample maximum*

$$p - value = B^{-1}\sum_{b=1}^{b} I\left[T_{u,b}^{\star SPA} > T_u^{SPA}\right]$$

Gains from Studentization



The u in $T_u^{SPA}$is for *upper*

- The U is included to indicate that the p-value derived using the LFC may not be the best p-value
- Suppose the some of the models have a very low mean and a high standard deviation
- In the RC and SPA-U, all models are assumed to be as good as the benchmark
- This is implemented by always re-centering the bootstrap samples around $\bar{\delta}_j$
- If a model is rejectably bad, then it may be possible to improve the power of the RC/SPA-U by excluding this model
- This is implemented using a "pre-test" of the form

$$I_j^u = 1, \;\; I_j^c = \frac{\bar{\delta}_j}{\sqrt{\hat{\omega}_j^2/P}} > -\sqrt{2 \ln \ln P}, \;\; I_j^l = \bar{\delta}_j > 0$$

  - The first (c for *consistent*) tests whether the standardized mean loss differential is greater than a HQ-like lower bound
  - The second (l for *lower*) only re-centers if the loss-differential is positive (e.g. the benchmark is out-performed)

General SPA

Algorithm (Test of SPA)

1. *Estimate* $\hat{\omega}_j^2$ *and compute* $T^{SPA} = \max\left(\bar{\delta}/\sqrt{\hat{\omega}_j^2/P}\right)$
2. *For* $b = 1, \ldots, B$ *re-sample the vector of loss differentials* $\boldsymbol{\delta}_t$ *to construct a bootstrap sample* $\left\{\boldsymbol{\delta}^{\star}_{b,t}\right\}$ *using the stationary bootstrap*
3. *Using the bootstrap sample, compute*

$$T^{\star SPA}_{s,b} = \max\left(\frac{P^{-1}\sum_{t=R+1}^{T}\delta^{\star}_{j,b,t} - I^s_j\bar{\delta}_j}{\sqrt{\hat{\omega}_j^2/P}}\right), \quad s = l, c, u$$

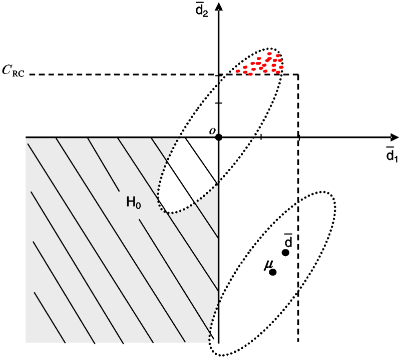
4. *Compute the Studentized Reality Check p-value as the percentage of the bootstrapped maxima which are larger than the sample maximum*

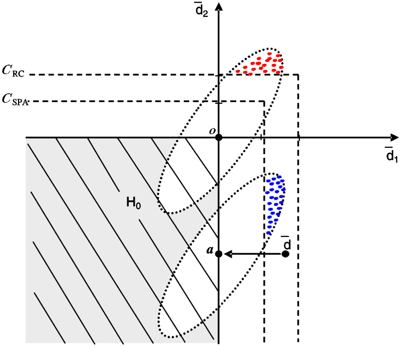
$$p - value = B^{-1}\sum_{b=1}^{b} I\left[T^{\star SPA}_{s,b} > T^{SPA}\right], \quad s = l, u, c$$

Comments on SPA

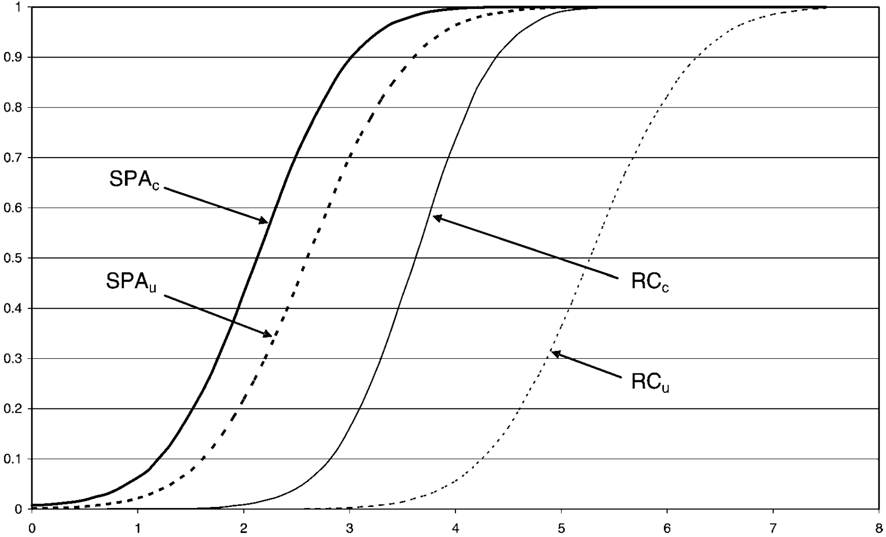
- The three versions only differ on whether a model is re-centered
- If a model is *not* re-centered, then it is unlikely to be the maximum in the re-sample distribution
  - This is how “bad” models are discarded in the SPA
- Can compute 6 different p-values statistics
  - Studentized or unmodified
  - Indicator function in l, c, u
    - Test statistic does not depend on l, c, u, only p-value does
- Reality Check uses unmodified loss differentials and u
- In practice Studentization beings important gains
- Using c is important if using SPA on large universe of automated rules if some may be very poor

Power Gains in SPA from Re-centering





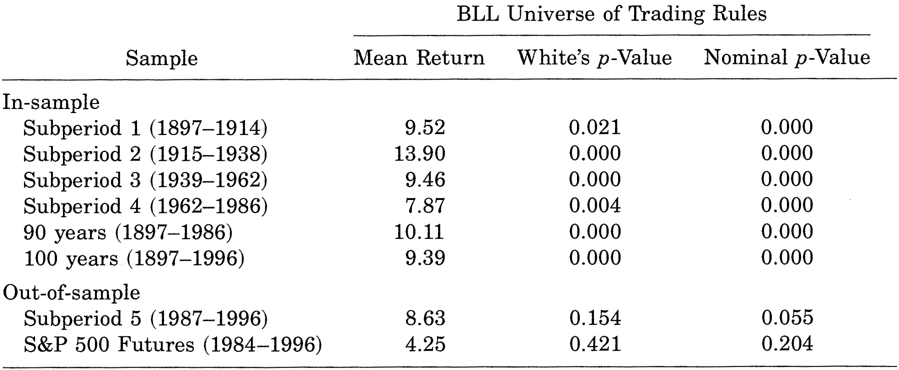
Combined Power Gains



Application of RC to Technical Trading Rules

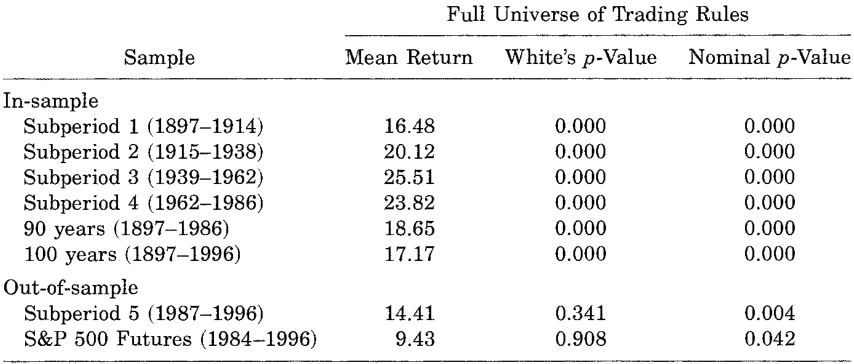
- Sullivan, Timmermann and White (1999) apply the RC to a large universe of technical trading rules
- Rules include:
  - Filter Rules
  - Moving Average Oscillators
  - Support and Resistance
  - Channel Breakout
  - On-balance Volume Averages
    - Tracks volume times return sign
    - Similar to Moving Average rules for prices
- Total of 7,846 trading rules
- Only use 1 at a time
- Use DJIA as in BLL, updated to 1996
- Consider mean return criteria and Sharpe Ratio

Mean Return Performance BLL Universe



	BLL Universe of Trading Rules		
Sample	Mean Return	White's p-Value	Nominal p-Value
In-sample			
Subperiod 1 (1897–1914)	9.52	0.021	0.000
Subperiod 2 (1915–1938)	13.90	0.000	0.000
Subperiod 3 (1939–1962)	9.46	0.000	0.000
Subperiod 4 (1962–1986)	7.87	0.004	0.000
90 years (1897–1986)	10.11	0.000	0.000
100 years (1897–1996)	9.39	0.000	0.000
Out-of-sample			
Subperiod 5 (1987–1996)	8.63	0.154	0.055
S&P 500 Futures (1984–1996)	4.25	0.421	0.204

Mean Return Performance Expanded



	Full Universe of Trading Rules		
Sample	Mean Return	White's *p*-Value	Nominal *p*-Value
In-sample			
Subperiod 1 (1897–1914)	16.48	0.000	0.000
Subperiod 2 (1915–1938)	20.12	0.000	0.000
Subperiod 3 (1939–1962)	25.51	0.000	0.000
Subperiod 4 (1962–1986)	23.82	0.000	0.000
90 years (1897–1986)	18.65	0.000	0.000
100 years (1897–1996)	17.17	0.000	0.000
Out-of-sample			
Subperiod 5 (1987–1996)	14.41	0.341	0.004
S&P 500 Futures (1984–1996)	9.43	0.908	0.042

RC based on Sharpe Ratio

- From any strategy it is simple to compute the Sharpe Ratio

$$SR = \frac{P^{-1}\sum_{t=R+1}^{T}\tilde{r}_{t+1} - r_{f,t+1}}{\sqrt{P^{-1}\sum_{t=R+1}^{T}\left(\tilde{r}_{t+1} - \bar{\tilde{r}}\right)^{2}}}$$

- The strategy return is $\tilde{r}_{t+1} = r_{t+1} S\left(\hat{y}_{j,t+1|t}\right)$
- $\bar{\tilde{r}}$ is the mean of the strategy return
- $r_{f,t+1}$ is the risk-free rate

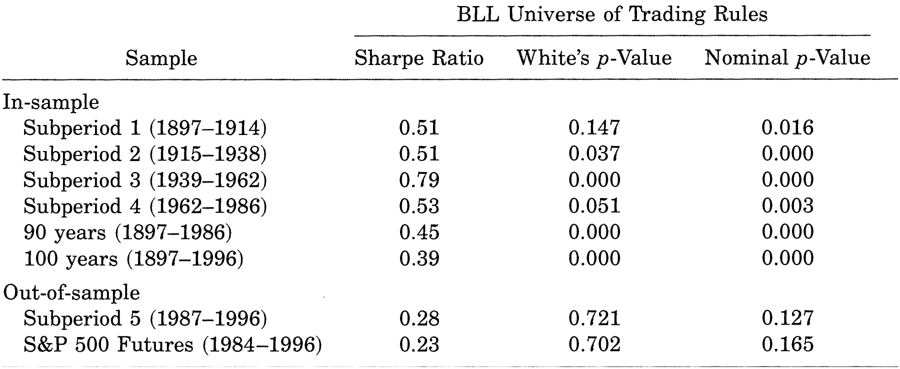
RC based on Sharpe Ratio

- The bootstrap can be used to compute a bootstrap version of the same rule by jointly re-sampling $\left\{\tilde{r}_{t+1}, r_{f,t+1}\right\}$
- The bootstrap Sharpe Ratio is then

$$
\begin{aligned}
SR_b^\star &= \frac{a}{\sqrt{b - c^2}} \\
a &= P^{-1} \sum_{t=R+1}^{T} \tilde{r}_{b,t+1} - r_{f,b,t+1} \\
b &= P^{-1} \sum_{t=R+1}^{T} \tilde{r}_{b,t+1}^2 \\
c &= P^{-1} \sum_{t=R+1}^{T} \tilde{r}_{b,t+1}
\end{aligned}
$$

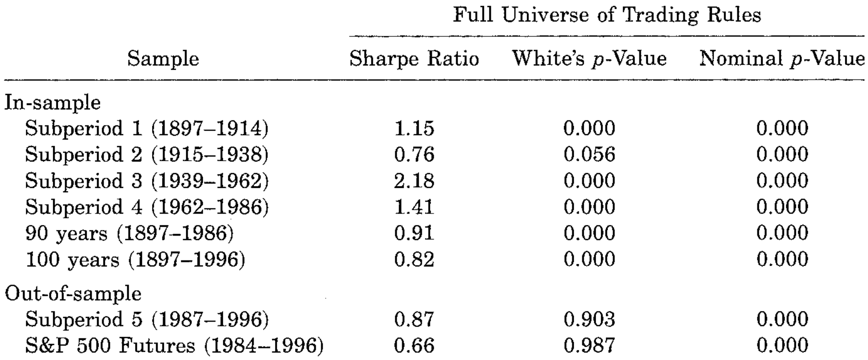
- The SR can be computed for all models
- The RC can then be applied to the (negative) SR, rather than the (negative) return

Sharpe Ratio Performance: BLL Universe



Sample	BLL Universe of Trading Rules		
	Sharpe Ratio	White's p-Value	Nominal p-Value
In-sample			
Subperiod 1 (1897–1914)	0.51	0.147	0.016
Subperiod 2 (1915–1938)	0.51	0.037	0.000
Subperiod 3 (1939–1962)	0.79	0.000	0.000
Subperiod 4 (1962–1986)	0.53	0.051	0.003
90 years (1897–1986)	0.45	0.000	0.000
100 years (1897–1996)	0.39	0.000	0.000
Out-of-sample			
Subperiod 5 (1987–1996)	0.28	0.721	0.127
S&P 500 Futures (1984–1996)	0.23	0.702	0.165

Sharpe Ratio Performance: Expanded



Sample	Full Universe of Trading Rules		
	Sharpe Ratio	White's p-Value	Nominal p-Value
In-sample			
Subperiod 1 (1897–1914)	1.15	0.000	0.000
Subperiod 2 (1915–1938)	0.76	0.056	0.000
Subperiod 3 (1939–1962)	2.18	0.000	0.000
Subperiod 4 (1962–1986)	1.41	0.000	0.000
90 years (1897–1986)	0.91	0.000	0.000
100 years (1897–1996)	0.82	0.000	0.000
Out-of-sample			
Subperiod 5 (1987–1996)	0.87	0.903	0.000
S&P 500 Futures (1984–1996)	0.66	0.987	0.000

Stepwise Multiple Testing

- The main issue with the Reality Check and the Test for SPA is the null
- These tests ultimately test one question:
  - Is the largest out-performance consistent with a random draw from the distribution when there are not superior models to the benchmark?
- If the null is rejected, only the best performing model can be determined to be better than the benchmark
- What about the 2nd best model? Or the kth best model?
- The *StepM* extends that reality check by allowing individual models to be tested
- It is implemented by repeatedly applying a RC-like algorithm which controls the *Familywise Error Rate (FWE)*

Basic Setup

- The basic setup is identical to that of the RC/SPA
- The test is based on $\delta_{j,t} = L\left(y_{t+h}, \hat{y}_{t+h,BM|t}\right) - L\left(y_{t+h}, \hat{y}_{t+h,j|t}\right)$
- Can be used in the same types of tests as RC/SPA
  - Absolute return
  - Sharpe Ratio
  - Risk-adjusted α comparisons
  - MSE/MAE
  - Predictive Likelihood
- Can be implemented on both raw and Studentized loss differentials

Null and Alternative Hypotheses

- The null and alternatives in StepM are not a single statement as they were in the RC/SPA
- The null**s** are

$$H_{0,j} : \mathrm{E}\left[\delta_t\right] \leq 0, \quad j = 1, \ldots, m$$

- The alternative**s** are

$$H_{1,j} : \mathrm{E}\left[\delta_t\right] > 0, \quad j = 1, \ldots, m$$

- StepM will ultimately result in a set of rejections (if any are rejected)
- Goal of StepM is to identify as many false nulls as possible while controlling the Familywise Error Rate

Familywise Error Rate

Definition (Familywise Error Rate)

For a set of null and alternative hypotheses $H_{0,i}$ and $H_{1,i}$ for $i = 1, \ldots, m$, let $\mathcal{I}_0$ contain the indices of the correct null hypotheses. The Familywise Error Rate is defined as

$$\Pr\left(\text{Rejecting at least one } H_{0,i} \text{ for } i \in \mathcal{I}_0\right) = 1 - \Pr\left(\text{Reject no } H_{0,i} \text{ for } i \in \mathcal{I}_0\right)$$

- The FWE is concerned only with the probability of making at least one Type I error
- Making 1, 2 or m Type I errors is the same to FWE
    - This is a criticism of FWE
    - Other criteria exist such as *False Discovery Rate* which controls the percentage of rejections which are false (# False Rejection/# Rejections)

Bonferoni Bounds

- Bonferoni bounds are the first procedure to control FWE

Definition (Bonferoni Bound)

Let $T_1, T_{2,} \ldots, T_m$ be a set of m test statistics, then

$$\underbrace{\Pr\left(T_1 \cup \ldots \cup T_m | H_{1,0}, \ldots H_{m,0}\right)}_{\text{Joint Probability}} \leq \sum_{j=1}^{m} \underbrace{\Pr\left(T_j | H_{0,j}\right)}_{\text{Individual Probability}}$$

where $\Pr\left(T_j | H_{0,j}\right)$ is the probability of observing $T_j$given the null $H_{0,j}$ is true.

- Bonferoni bounds are a simple method to test m hypotheses using only univariate test statistics
- Let $\{pv_j\}$ be a set of m p-values from a set of tests
- The Bonferoni bound will reject the set of nulls is $pv_j \leq \alpha/m$ for all j
    - α is the size of the test (e.g. 5%)
- When m is moderately large, this is a very conservative test
- Conservative since assumes worst case dependence among statistics

Holm's procedure

Definition (Holm's Procedure)

Let $T_1, T_{2,} \ldots, T_m$ be a set of m test statistics with associated p-values pv_j, $j = 1, \ldots, m$ where it is assumed $pv_i < pv_j$ if $i < j$. If

$$pv_j \leq \alpha / \left(m - j + 1\right)$$

then $H_{0,j}$ can be rejected in factor of $H_{1,j}$ while controlling the famliywise error rate at α.

- Example: p-values of $.001$, $.01$, $.03$, $.05$, $m = 4$, $\alpha = .05$
- Improves Bonferoni by ordering the p-values and using a stepwise procedure
- Allows subsets of hypotheses to be tested – Bonferoni is joint
- Less strict, except when $j = 1$ (same as Bonferoni)
- **Note**: Holm's procedure ends as soon as a null cannot be rejected

Relationships between testing procedures

- The RC/SPA, Bonferoni and Holm are all related

	Worst-case Dependence	Accounts for Dependence in Data
Single-step	Bonferoni	RC, SPA
Stepwise	Holm	StepM

StepM Algorithm

Algorithm (StepM)

1. *Begin with the active set* $\mathcal{A} = \{1, 2, \ldots, m\}$*, superior set* $\mathcal{S} = \{\}$
2. *Construct* B *bootstraps sample* $\left\{\boldsymbol{\delta}^{\star}_{b,t}\right\}$, $b = 1, \ldots, B$
3. *For each bootstrap sample, compute* $T^{\star StepM}_{k,b} = \max_{j \in \mathcal{A}} \left\{\bar{\delta}^{\star}_{b,j} - \bar{\delta}_j\right\}$
4. *Compute* $q_{k,\alpha}$ *as the* $1 - \alpha$ *quantile of* $\left\{T^{\star StepM}_{k,b}\right\}$
5. *If* $\max_{j \in \mathcal{A}} \left(\bar{\delta}_j\right) < q_{k,\alpha}$ *stop*
6. *Otherwise for each j* $\in \mathcal{A}$
    a. *If* $\bar{\delta}_j \geq q_{k,\alpha}$ *add j to* $\mathcal{S}$ *and delete from* $\mathcal{A}$
    b. *Return to 2*

Comments

- StepM would be virtually identical to RC if only the largest $\bar{\delta}_j$ was tested
- Improves on the RC since (weakly more) individual out-performing models can be identified
- If no model outperforms, will stop with none and RC p-value will be larger than α
- Steps 2–4 are identical to the RC using the models in $\mathcal{A}$
- The stepwise testing can improve power by removing models
  - The improvement comes if a model with substantial out-performance also has large variance
  - Removing this model allows the critical value to be reduced
- StepM only guarantees that FWE$\leq \alpha$, and in general will be $< \alpha$
  - Will only $= \alpha$ if $\mathrm{E}\left[\delta_{j,t}\right] = 0$ for all j
  - Example: $N\left(\mu, \sigma^2\right)$ when $\mu < 0$, $H_0 : \mu = 0$

Studentization

- Like the SPA to the RC, the StepM can be implemented using Studentized loss differentials
- Romano & Wolf argue that the Studentization should be done *inside* each bootstrap sample, not globally as in the SPA
- Theoretically both are justified and neither makes a difference asymptotically
- Computing the variance inside each bootstrap will more closely match the re-sampled data than when using a global estimate

Studentized StepM Algorithm

Algorithm (Studentized StepM)

1. *Begin with the active set* $\mathcal{A} = \{1, 2, \ldots, m\}$, *superior set* $\mathcal{S} = \{\}$
2. *Compute* $\bar{z}_j = \bar{\delta}_j / \sqrt{\hat{\omega}_j^2 / P}$ *where* $\hat{\omega}_j^2$ *was previously defined*
3. *Construct B bootstraps sample* $\left\{ \boldsymbol{\delta}^\star_{b,t} \right\}$, $b = 1, \ldots, B$
4. *For each bootstrap sample, compute*

$$T^{\star StepM}_{k,b} = \max_{j \in \mathcal{A}} \left\{ \frac{\bar{\delta}^\star_{b,j} - \bar{\delta}_j}{\hat{\omega}^\star_j} \right\}$$

   where $\hat{\omega}_j^{2\star}$ *is an estimate of the long-run variance of the bootstrapped data*

5. *Compute* $q^z_{k,\alpha}$ *as the* $1 - \alpha$ *quantile of* $\left\{ T^{\star StepM}_{k,b} \right\}$
6. *If* $\max_{j \in \mathcal{A}} \left(\bar{z}_j \right) < q^z_{k,\alpha}$ *stop*
7. *Otherwise for each* $j \in \mathcal{A}$
   a. *If* $\bar{z}_j \geq q^z_{k,\alpha}$ *add j to* $\mathcal{S}$ *and delete from* $\mathcal{A}$
   b. *Return to 2*

Why Studentization Help

- StepM is built around confidence intervals of the form

$$\left[\bar{\delta}_1 - q_{1,\alpha}, \infty\right] \times \ldots \times \left[\bar{\delta}_m - q_{1,\alpha}, \infty\right]$$

- Null hypotheses are rejected for models where 0 is *not* in its confidence interval
- In the raw form, the confidence interval is a square – the same for every loss differential
- When Studentization is used, the confidence intervals take the form

$$\left[\bar{\delta}_1 - \sqrt{\omega_1^2/P} q^z_{1,\alpha}, \infty\right] \times \ldots \times \left[\bar{\delta}_m - \sqrt{\omega_m^2/P} q^z_{1,\alpha}, \infty\right]$$

- This “customization” allows for more rejections if the loss differentials have cross-sectional heteroskedasticity

Block-size Selection

- Paper proposes a procedure to make data driven block size
- Basic idea is to use a (V)AR on $\left\{\delta_{j,t}\right\}$ to approximate the dependence
  - Similar to Den Hann-Levine HAC
- Fit AR & estimate residual covariance (or use short block bootstrap on errors)
- Simulate from model
- For $w = 1, \ldots, \overline{W}$ compute the bootstrap confidence region with size $1-\alpha$using percentile method
- For each block size, compute the empirical coverage – percentage of simulated $\bar{\delta}$ in their confidence region
- Choose optimal w which most closely matches $1-\alpha$
  - Alternative: Use Politis & White

Empirical Application

- Applied StepM to a set of 105 Hedge Fund Returns with long histories
- Returns net of management fees
- Benchmark model was *risk-free rate*
- $m = 105, P = 147$ (all out-of-sample)
- Results:
  - Raw data: No out-performers
    - Max ratio of standard deviation $\hat{\omega}_i / \hat{\omega}_j = 22$
  - Studentized: 7 funds identified
- **Note**: Will *always* identify funds with the largest $\bar{\delta}$ (or $\bar{z}$) first

Empirical Application

$\bar{x}_{T,s} - \bar{x}_{T,S+1}$	Fund	$(\bar{x}_{T,s} - \bar{x}_{T,S+1})/\hat{\sigma}_{T,s}$	Fund
1.70	Libra Fund	10.63	Market Neutral*
1.41	Private Investment Fund	9.26	Market Neutral Arbitrage*
1.36	Aggressive Appreciation	8.43	Univest (B)*
1.27	Gamut Investments	6.33	TQA Arbitrage Fund*
1.26	Turnberry Capital	5.48	Event-Driven Risk Arbitrage*
1.14	FBR Weston	5.29	Gabelli Associates*
1.11	Berkshire Partnership	5.24	Elliott Associates**
1.09	Eagle Capital	5.11	Event Driven Median
1.07	York Capital	4.97	Halcyon Fund
1.07	Gabelli Intl.	4.65	Mesirow Arbitrage Trust

Improving StepM using SPA

- The main step in the StepM algorithm is identical to the RC
- The important difference is that the test is implemented for each null, rather than globally
- StepM will suffer if very poor models are included with a large variance
  - Especially true for raw version, but also relevant for Studentized version
  - Example

$$\left[\begin{array}{c}\bar{\delta}_1\\ \bar{\delta}_2\end{array}\right]\sim N\left(\left[\begin{array}{c}0\\ -5\end{array}\right],\left[\begin{array}{cc}1&0\\ 0&1\end{array}\right]\right)$$

  - Reality Check critical value will be 1.95, while “best” critical value would be 1.645 (since only 1 relevant for asymptotic distribution)
- The RC portions of StepM can be replaced by SPA versions which addresses this problem
- Simple as adding in the indicator function I_j^c when subtracting the mean in step 3 (step 4 in Studentized version)
- Using SPA modification will always find more out-performing models

Model Confidence Set (MCS)

- RC, SPA and StepM were all testing superior predictive ability
- This type hypothesis is common when there is a natural benchmark
- In some scenarios there may not be a single benchmark, or there may more than one models which could be considered benchmarks
- When this occurs, it is not clear
  - How to implement RC/SPA/StepM
  - How to make sound conclusions about superior predictive ability
- The model confidence set addresses this problem by *bypassing the benchmark*
- The MCS aims to find the *best model* and all models which are *indistinguishable from the best*
  - The model with the lowest loss will always be the best – identifying the others is more challenging
- Also returns p-values for models with respect to the MCS

Notation Preliminaries

- The outcome of the MCS is a *set of models*
  - All model sets will be denoted using $\mathcal{M}$
- The initial model set is $\mathcal{M}_0$
- The goal is to find $\mathcal{M}^\star$ which is the set of all models which are indistinguishable from the best
- The output of the MCS algorithm is $\widehat{\mathcal{M}}_{1-\alpha}$ where α is the size of the test
  - The size is interpreted as a Familywise Error Rate – same as StepM
  - In general $\widehat{\mathcal{M}}_{1-\alpha}$ will contain more than 1 model
- In between $\mathcal{M}_0$ and $\widehat{\mathcal{M}}_{1-\alpha}$ are other sets of models

$$\mathcal{M}_0 \supset \mathcal{M}_1 \supset \ldots \supset \widehat{\mathcal{M}}_{1-\alpha}$$

Notation Preliminaries

- To construct the model confidence set, two tools are needed
  - An equivalence test $d_{\mathcal{M}}$: Determines whether the model in $\mathcal{M}$ are equal in terms of loss
  - An elimination rule $e_{\mathcal{M}}$: Determines which model to eliminate if $d_{\mathcal{M}}$ finds that the models are not equivalent
- The generic form of the algorithm, starting at $i = 0$:
  1. Apply $d_{\mathcal{M}}$ to $\mathcal{M}_i$
  2. If $d_{\mathcal{M}}$ rejects equivalence, use $e_{\mathcal{M}}$to eliminate 1 model to produce $\mathcal{M}_{i+1}$
     a. If not, stop
  3. Increment i, return to 1
- Has a similar flavor to StepM
  - Also gains from eliminating models with high variance

The Model Confidence Set

- When the algorithm ends, the final set $\widehat{\mathcal{M}}_{1-\alpha}$ has the property

$$\lim_{P\to\infty} \Pr\left(\mathcal{M}^\star \subset \widehat{\mathcal{M}}_{1-\alpha}\right) \geq 1-\alpha$$

- The result follows directly since the FWE is $\leq \alpha$
- If there is only 1 "best" model, then the result can be strengthened

$$\lim_{P\to\infty} \Pr\left(\mathcal{M}^\star \subset \widehat{\mathcal{M}}_{1-\alpha}\right) = 1$$

  - The MCS will find the "best" model asymptotically
  - The intuition behind this is that the "best" model will have:
    - Lower loss than all other models
    - The variance of the average loss differential will decline as $P \to \infty$
- When 2 or more models are equally good, there is always a α chance that at least 1 will be rejected
- In large samples, models which are not in $\mathcal{M}^\star$ will be eliminated with probability 1 since the individual test statistics are consistent

Model Confidence Set

- The MCS takes loss functions as inputs, but ultimately works on loss differentials
- Since there is no benchmark model, all loss differentials are considered

$$\delta_{ij,t} = L\left(y_{t+h}, \hat{y}_{t+h,i|t}\right) - L\left(y_{t+h}, \hat{y}_{t+h,j|t}\right)$$

- There are many pairs, and so the actual test examines whether the average loss for model j is different from that of all models

$$\bar{\delta}_i = \frac{1}{m-1} \sum_{i=1, i \neq j}^{m} \bar{\delta}_{ij}$$

- If $\bar{\delta}_i$ is sufficiently positive, then model i is worse then the other models in the set

Null and Alternative

- The MCS can be based on two test statistics
- Both satisfy some technical conditions on $d_{\mathcal{M}}$ and $e_{\mathcal{M}}$
- The first is based on $T = \max_{i \in \mathcal{M}} (\bar{z}_i)$ where $\bar{z}_i = \bar{\delta}_i / \hat{\sigma}_i$ and $\hat{\sigma}_i^2$ is an estimate of the (log-run) variance of $\bar{\delta}_i$
    - The elimination rule is $e_{\mathcal{M}} = \operatorname{argmax}_{i \in \mathcal{M}} z_i$
- The second is based on $T_R = \max_{i,j \in \mathcal{M}} \left| \bar{z}_{ij} \right|$ where $\bar{z}_{ij} = \bar{\delta}_{ij} / \hat{\sigma}_{ij}$ and $\hat{\sigma}_{ij}$ is an estimate of the (log-run) variance of $\bar{\delta}_{ij}$
    - The elimination rule is $e_{R,\mathcal{M}} = \operatorname{argmax}_{i \in \mathcal{M}} \sup_{j \in \mathcal{M}} \bar{z}_{ij}$
    - Eliminate the model which has the largest loss differential to some other model, relative to its standard deviation
- At each step the null is $H_0 : \mathcal{M} = \mathcal{M}^\star$ and the alternative is $H_1 : \mathcal{M} \supsetneq \mathcal{M}^\star$

Model Confidence Set Setup

Algorithm (Model Confidence Set Components)

1. *Construct a set of bootstrap indices which will be reused throughout the MCS construction using a bootstrap appropriate for the data*
2. *Construct the average loss for each model*

$$\bar{L}_j = P^{-1} \sum_{t=R+1}^{T} L_{j,t}$$

   where $L_{j,t} = L\left(y_{t+h}, \hat{y}_{t+h,j|t}\right)$

3. *For each bootstrap replication, compute centered the bootstrap average loss*

$$\eta^{\star}_{b,j} = P^{-1} \sum_{t=R+1}^{T} L^{*}_{b,j,t} - \bar{L}_j$$

Model Confidence Set

Algorithm (Model Confidence Set)

1. *Being with $\mathcal{M} = \mathcal{M}_0$ containing all models where m is the number of models in $\mathcal{M}$*
2. *Calculate $\bar{L} = m^{-1} \sum_{j=1}^{m} \bar{L}_j$, $\eta_b^\star = m^{-1} \sum_{j=1}^{m} \eta_{b,j}^\star$, and*
   $\hat{\sigma}_j^2 = B^{-1} \sum_{b=1}^{B} \left(\eta_{b,j}^\star - \bar{\eta}_j^\star\right)^2$ *where $\bar{\eta}_j^\star$ is the average of $\eta_{b,j}^*$ for model j*
3. *Define $T = \max_{j \in \mathcal{M}} \left(\bar{z}_j\right)$ where $\bar{z}_j = \bar{L}_j / \hat{\sigma}_j$*
4. *For each bootstrap sample, compute*
   $T_b^\star = \max_{j \in \mathcal{M}} \left(\left(\bar{L}_{b,j}^\star - \bar{L}_b^\star\right) / \hat{\sigma}_j\right) = \max_{j \in \mathcal{M}} \left(\left(\eta_{b,j}^\star - \eta_b^\star\right) / \hat{\sigma}_j\right)$
5. *Compute the p-value of $\mathcal{M}$ as $\hat{p} = B^{-1} \sum_{b=1}^{B} I\left[T_b^\star > T\right]$*
6. *If $\hat{p} > \alpha$ stop*
7. *If $\hat{p} < \alpha$, set $e_{\mathcal{M}} = \operatorname{argmax}_{j \in \mathcal{M}} \left(\bar{z}_j\right)$ and eliminate the model with the largest test statistic from $\mathcal{M}$*
8. *Return to step 2, using the reduced model set*

Comments

- It is important that the variance estimates are re-computed in each step of algorithm
- This allows the standard errors to decline if poor models are excluded since the cross-sectional variance of $\bar{L}_j$ should be smaller when a bad model is dropped
- In practice the MCS should be implemented by computing in order
  1. A set of bootstrap indices
  2. The P by m set of bootstrapped losses $L^*_{b,j,t}$
  3. The 1 by m vector containing $\eta^\star_{b,j}$
- By iterating over these B times only the B by m matrix containing $\eta^\star_{b,j}$ has to be retained
  - Plus the 1 by m vector containing $\bar{L}_j$

Model Confidence P-value

- The MCS can also provide p-values for each model
- If model i is eliminated, then the p-value of model i is the maximum of the $\hat{p}$ found when model i is eliminated and *all previous p-values*
- Suppose $\alpha = .05$, and the first three rounds eliminated models with $\hat{p}$ of .01,.04,.02, respectively
- The three p-values would then be:
    - $.01$(nothing to compare against)
    - $.04 = \max(.01, .04)$
    - $.04 = \max(.02, .04)$
- The output of the MCS algorithm is $\widehat{\mathcal{M}}_{1-\alpha}$ which contains the true set of best models with probability weakly larger than $1 - \alpha$
- This is similar to a standard frequentist confidence interval which contains the true parameter with probability of at least $1 - \alpha$
- The MCS p-value is not a statement about the probability that a model is the best
    - For example, the model with the lowest loss always has p-value = 1

Model Confidence P-value

Table 1: Computation of MCS p-values

Elimination Rule	p-value for $H_{0,\mathcal{M}_k}$	MCS p-value
$e_{\mathcal{M}_1}$	$P_{H_{0,\mathcal{M}_1}} = 0.01$	$\hat{p}_{e_{\mathcal{M}_1}} = 0.01$
$e_{\mathcal{M}_2}$	$P_{H_{0,\mathcal{M}_2}} = 0.04$	$\hat{p}_{e_{\mathcal{M}_2}} = 0.04$
$e_{\mathcal{M}_3}$	$P_{H_{0,\mathcal{M}_3}} = 0.02$	$\hat{p}_{e_{\mathcal{M}_3}} = 0.04$
$e_{\mathcal{M}_4}$	$P_{H_{0,\mathcal{M}_4}} = 0.03$	$\hat{p}_{e_{\mathcal{M}_4}} = 0.04$
$e_{\mathcal{M}_5}$	$P_{H_{0,\mathcal{M}_5}} = 0.07$	$\hat{p}_{e_{\mathcal{M}_5}} = 0.07$
$e_{\mathcal{M}_6}$	$P_{H_{0,\mathcal{M}_6}} = 0.04$	$\hat{p}_{e_{\mathcal{M}_6}} = 0.07$
$e_{\mathcal{M}_7}$	$P_{H_{0,\mathcal{M}_7}} = 0.11$	$\hat{p}_{e_{\mathcal{M}_7}} = 0.11$
$e_{\mathcal{M}_8}$	$P_{H_{0,\mathcal{M}_8}} = 0.25$	$\hat{p}_{e_{\mathcal{M}_8}} = 0.25$
$\vdots$	$\vdots$	$\vdots$
$e_{\mathcal{M}_{(m_0)}}$	$P_{H_{0,\mathcal{M}_{m_0}}} \equiv 1.00$	$\hat{p}_{e_{\mathcal{M}_{m_0}}} = 1.00$

Model Confidence Set using T_R

Algorithm (Model Confidence Set Components)

1. *Construct a set of bootstrap indices which will be reused throughout the MCS construction using a bootstrap appropriate for the data*
2. *Construct the average loss for each model* $\bar{L}_j = P^{-1} \sum_{t=R+1}^{T} L_{j,t}$ *where* $L_{j,t} = L\left(y_{t+h}, \hat{y}_{t+h,j|t}\right)$
3. *For each bootstrap replication, compute centered the bootstrap average loss*

$$\bar{L}^{\star}_{b,j} = P^{-1} \sum_{t=R+1}^{T} L^{*}_{b,j,t} - \bar{L}_j$$

4. *Calculate*

$$\hat{\sigma}^2_{ij} = B^{-1} \sum_{b=1}^{B} \left(\left(\bar{L}^{\star}_{b,i} - \bar{L}^{\star}_{i} \right) - \left(\bar{L}^{\star}_{b,j} - \bar{L}^{\star}_{j} \right) \right)^2$$

where $\bar{L}^{\star}_{j}$ *is the average of* $\bar{L}^{\star}_{b,j}$ *for the model* j *across all bootstraps*

Model Confidence Set

Algorithm (Model Confidence Set)

1. *Being with $\mathcal{M} = \mathcal{M}_0$ containing all models where m is the number of models in $\mathcal{M}$*
2. *Define $T_R = \max_{i,j\in\mathcal{M}} \left(\bar{z}_{ij}\right)$ where $\bar{z}_{ij} = \left|\bar{L}_i - \bar{L}_j\right| / \hat{\sigma}_{ij}$*
3. *For each bootstrap sample, compute $T^\star_{R,b} = \max_{i,j\in\mathcal{M}} \left(\left|\bar{L}^\star_i - \bar{L}^\star_j\right| / \hat{\sigma}_{ij}\right)$*
4. *Compute the p-value of $\mathcal{M}$ as*

$$\hat{p} = B^{-1} \sum_{b=1}^{B} I\left[T^\star_{R,b} > T_R\right]$$

5. *If $\hat{p} > \alpha$ stop*
6. *If $\hat{p} < \alpha$, set $e_{\mathcal{M}} = \operatorname{argmax}_{i\in\mathcal{M}} \sup_{j\in\mathcal{M}} \left(\bar{z}_{ij}\right)$ and eliminate the model with the largest test statistic from $\mathcal{M}$*
7. *Return to step 2, using the reduced model set*

Comments

- The main difference is that the variance is *not* re-estimated in each iteration
- This happens since T_R is based on the maximum DMW test statistic in each iteration
    - DMW only depends on the properties of the pair
- However, the bootstrapped distribution does depend on which models are included and so this will vary across the iterations
- This version of the algorithm requires storing the B by m matrix of $\bar{L}_j^\star$

Confidence sets for ICs

- The MCS can be used to construct confidence sets for ICs
- This type of comparison does not directly use forecasts, and so is in-sample
- This differs from traditional model selection where only the model with the best IC is chosen
- The MCS for an IC could be used as a pre-filtering mechanism prior to combining
- Implementing the MCS on an IC is slightly more complicated than the default MCS since it is necessary to jointly bootstrap the vector $\left\{y_t, \mathbf{x}_{j,t}\right\}$ where $\mathbf{x}_{j,t}$ are the regressors in model j
- Paper recommends using T_R statistic to compare models using IC
- The object of interest is

$$IC_j = T \ln \hat{\sigma}_j^2 + c_j$$

- c_j is the penalty term
  - AIC: $2k_j$, BIC: $k_j \ln T$
  - AIC*: $2k_j^\star$, BIC*: $k_j^\star \ln T$
- $k_j^\star$ is known as *effective degrees of freedom* (in mis-specified model $k^\star \neq k$)
- MCS paper discusses how to estimate $k^\star$

Confidence sets for ICs

- Using T_R MCS construction algorithm, the test statistic is based on

$$T_R = \max_{i,j \in \mathcal{M}} \left| \left[T \ln \hat{\sigma}_i^2 + c_i \right] - \left[T \ln \hat{\sigma}_j^2 + c_j \right] \right|$$

- The bootstrap critical values are computed from

$$T_{R,b}^\star = \max_{i,j \in \mathcal{M}} \left(\left[T \ln \hat{\sigma}_i^{2\star} + c_i - T \ln \hat{\sigma}_i^2 \right] - \left[T \ln \hat{\sigma}_j^{2\star} + c_j - T \ln \hat{\sigma}_j^2 \right] \right)$$

- $\hat{\sigma}_i^{2\star}$ is the variance computed using

$$\epsilon_{b,t}^\star = y_{b,t}^\star - \mathbf{x}_{b,j,t}^{\star\prime} \hat{\boldsymbol{\beta}}_{b,j}^\star$$

- $\hat{\boldsymbol{\beta}}_{b,j}^\star$ is re-estimated using the bootstrapped data $\left\{ y_{b,t}^\star, \mathbf{x}_{b,j,t}^\star \right\}$
- Errors are computed using the bootstrapped data and parameter estimates
- Aside from these changes, the remainder of the algorithm is unmodified

False Discovery Rate and FWER

- Controlling False Discover Rate (FDR) is an alternative to controlling Family Wise Error Rate (FWER)

Definition (k-Familywise Error Rate)

For a set of null and alternative hypotheses $H_{0,i}$ and $H_{1,i}$ for $i = 1, \ldots, m$, let $\mathcal{I}_0$ contain the indices of the correct null hypotheses. The k-Familywise Error Rate is defined as

$$\Pr\left(\text{Rejecting at least } k\ H_{0,i} \text{ for } i \in \mathcal{I}_0\right) = 1 - \Pr\left(\text{Reject no } H_{0,i} \text{ for } i \in \mathcal{I}_0\right)$$

- k is typically 1, so the testing procedures control the probability of any number of false rejections
  - Type I errors
- The makes FWER tests possibly conservative
  - Depends on what the actual intent of the study is

False Discovery Rate

Definition

The False Discovery Rate is the percentage of false null hypothesis relative to the total number of rejections, and is defined

$$FDR = {}^{F}\!/_{R}$$

where F is the number of false rejections and R is the total number of rejections.

- Unlike FWER, methods that control FDR explicitly assume that some rejections are false.
- Ultimately this leads to a (potentially) procedure that might discover more actual rejections
- For standard DMW-type tests, both FWER and FDR control fundamentally reduce to choosing a critical value different from the usual ± 1.96
  - Most of the time larger in magnitude
  - Can be smaller in the case of FDR when there are many false nulls

False Discovery Rate

- FDR is naturally *adaptive*
- When the number of false nulls is small (~0), then FDR should choose a critical value similar to the FWER-based procedures
  - $R \approx F$, $F/R \approx 1$ so any F is too large
  - On the other hand, when the percentage of false nulls is near 100%, can reject all nulls
    - $F \approx 0$, $F/R \approx 0$ and all nulls can be rejected
    - Critical value can be arbitrarily small since virtually no tests have small values
    - Hypothetically, could have a critical value of 0 if all nulls were actually false
- FDR controls the false rejection rate, and it is common to use rates in the range of 5-10%
  - Ultimately should depend on risk associated with trading a bad strategy against the cost of missing a good strategy
  - Adding a small percentage of near 0 excess return strategies to a large set of useful strategies shouldn't deteriorate performance substantially

Operationalizing FDR

- Operationalizing FDR requires some estimates
- In standard trading strategy setup, $H_0 : \mu = 0$, $H_A : \mu \neq 0$ where μ is the expected return in excess of some benchmark
  - Benchmark might be risk-free rate, or could be buy-and-hold strategy
- π is the proportion of false nulls
  - Estimated using information about the distribution of p-values "near" 1 since these should all be generated from true nulls
  - Entire procedure relies on only p-values
    - Similar to Bonferoni or Bonferoni-Holm
  - For standard 2-sided alternative

$$p_i = 2\left(1 - \Phi\left(|t_i|\right)\right)$$

where t_i is (normalized) test statistic for strategy i.

Computing FDR

- Key idea is to find γ, which is some number in $[0, 1]$ such that

$$\alpha = \widehat{FDR} \equiv \frac{\hat{\pi} l \gamma}{\sum_{i=1}^{l} I\left[p_i < \gamma\right]}$$

- where
    - α is the target FDR rate
    - $\hat{\pi}$ and an estimate of the percentage of nulls that are true (no abnormal performance)
    - l is the number of rules
    - γ is the parameter that is used to find the p-value cutoff
    - $\sum_{i=1}^{l} I\left[p_i < \gamma\right]$ is the number of rejections using γ
- The numerator is simply an estimate of the number of false rejections, which is
  Probability of Null True $\times$ Number of Hypotheses = Number of True Hypotheses
  Number of False Hypotheses $\times$ Cutoff = Number of False that are Rejected using γ
- Exploits the fact that under the null p-values have a uniform distribution, so that if there are M false nulls, then, using a threshold of γ will reject γM

Positive and Negative FDR

- Can further decompose FDR into upper (better) and lower (worse) measures

$$\widehat{FDR}^{+} \equiv \frac{{}^{1}\!/_{2}\hat{\pi} l \gamma_U}{\sum_{i=1}^{l} I\left[p_i < \gamma_U, t_i > 0\right]}, \quad \widehat{FDR}^{+} \equiv \frac{{}^{1}\!/_{2}\hat{\pi} l \gamma_L}{\sum_{i=1}^{l} I\left[p_i < \gamma_L, t_i < 0\right]}$$

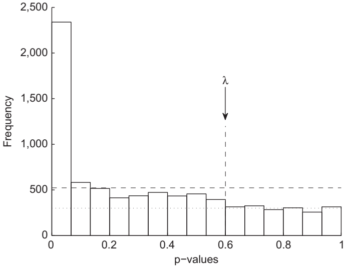
- This version assumes a symmetric 2-sided test statistic, so that on average 50% of the false rejections are in each tail
- Allows for tail-specific choice of γ which would naturally vary if the number of correct rejections was different
  - Suppose for example that many rules were bad, then γ_L would be relatively large

Estimation of π

- π is estimated as

$$\hat{\pi} = \frac{\sum_{i=1}^{l} I\left[p_k > \lambda\right]}{l\,(1-\lambda)}$$

- $\lambda$is a tuning parameter
  - Simple to choose using visual inspection
  - Recall that true nulls lead to a flat p-value histogram
  - Find point where histogram looks non-flat, use cutoff for λ
- Histogram from BS



Estimating π

- $\hat{\pi}$ allows percentage of correct rejections to be computed as $\hat{\pi}^A = 1 - \hat{\pi}$
- In the decomposed FDR the number of good (bad) rules can be computed as

$$\alpha \times \sum_{i=1}^{l} I\left[p_i < \gamma_U, t_i > 0\right]$$

  - Note that γ_U is fixed here

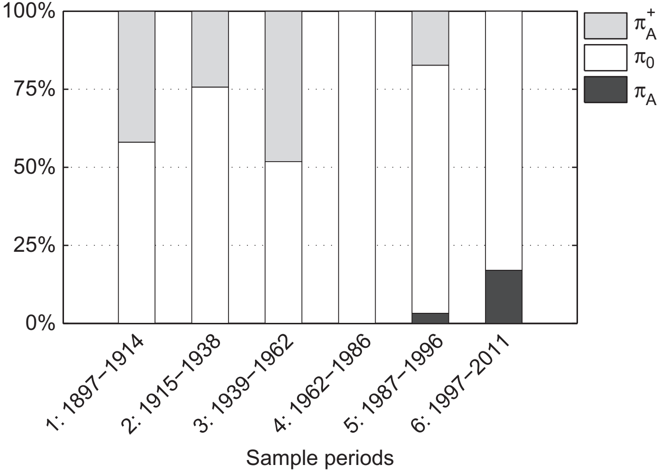
Bajgrowicz & Scaillet (*JFE*, 2012)

- Apply FDR to technical trading rules of STW
- Use DJIA
  - 1897-2011
- Find similar results, although importantly consider transaction costs for break even
  - Strategies that trade more can have higher means while not violating EMH

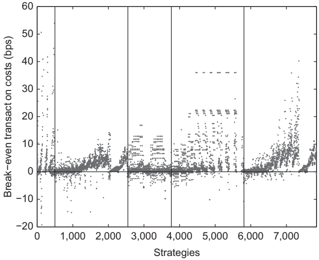
Background on Competitor Methods

Sample period	RW portfolio		Best rule		DJIA
	Sharpe ratio	Portfolio size	Sharpe ratio	BRC *p*-value	Sharpe ratio
1: 1897–1914	1.24	45	1.18	0.00	−0.12
2: 1915–1938	–	0	0.73	0.11	0.06
3: 1939–1962	1.49	62	2.34	0.00	0.41
4: 1962–1986	1.52	15	1.45	0.00	−0.16
5: 1987–1996	–	0	0.84	0.93	0.66
6: 1997–2011	–	0	0.48	1.00	0.12
1897–1996	0.70	88	0.82	0.00	0.12

Good and Bad Rules



Transaction Costs Required for 0-profit (-1962)

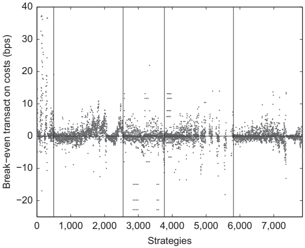


- Transaction costs are important when assessing rules
- Rather than apply arbitrary TC, look for break even
- Transaction costs are a function of mean and number of transactions

$$0 = \mu_i - TC \times \# \{trades\}$$

- μ_i is the full-sample mean, not the annualized

Transaction Costs Required for 0-profit (1962-)



- Transaction for break even are lower
- Actual transaction costs are lower
- Unclear whether this is driven by more trading signals or worse mean

Persistence of Rules

Sample period	FDR portfolio			RW portfolio			50 best rules		Best rule	
	IS	OOS	Median size	IS	OOS	Median size	IS	OOS	IS	OOS
1: 1897–1914	3.41	0.47	14	1.31	0.51	0	5.79	0.50	6.34	0.03
2: 1915–1938	4.62	0.01	13	0.90	0.17	0	5.39	−0.03	5.98	0.09
3: 1939–1962	4.77	0.55	15	1.85	0.09	0	5.78	0.43	6.70	0.12
4: 1962–1986	5.34	−0.31	13	1.36	0.14	0	6.17	−0.18	6.95	−0.59
5: 1987–1996	4.52	−0.34	12	–	–	–	5.44	−0.37	6.07	0.08
6: 1997–2011	4.55	−0.74	12	0.78	0.07	0	5.22	−0.51	5.97	−0.27

- Sharpe-Ratios
- Persistence is low
- Conservative Romano-Wolf appears to have more persistence
- Combination appears to be not help