# Spectral Backtests of Forecast Distributions

## with Application to Risk Management

Michael B. Gordy[1]  Alexander J. McNeil[2]

[1]Federal Reserve Board [2]York Management School

R in Finance 2018

The opinions expressed are our own, and do not reflect the views of the Board of Governors or its staff. Email: ⟨michael.gordy@frb.gov⟩.
Working paper available as FEDS 2018-021.
Software implemention in R available as the spectralBacktest package.

# Traditional setting for backtesting

- We take a new look at the old risk management problem of backtesting.
- Consider a bank with a one-day-ahead forecast distribution of its P&L, and a regulator who wants to assess the accuracy of the forecast.
- In the regulatory context, it is generally assumed that the bank
  - calculates value-at-risk (VaR) as a quantile of the forecast distribution, and
  - reports to the regulator whenever realized loss exceeds VaR.
- If the model is correct, then the VaR exceedances form a sequence of iid Bernoulli random variables.
- Regulatory and internal backtests based on VaR exceedances as test inputs.
- Increasingly, regulators can observe more than just the VaR exceedances.

# VaR as quantile of forecast distribution

- Notation:

  $\mathcal{F}_t$ : Information available at $t$ (filtration).
  $L_t$ : Loss realized at $t$ on portfolio formed at $t-1$.
  $F_t$ : $F_t(y) = \Pr(L_t \leq y | \mathcal{F}_{t-1})$; i.e., the df of the day-ahead forecast distribution.
  $\widehat{F}_t$ : The forecast distribution formed by the bank's risk-manager.

- VaR and VaR exceedances
  - $\widehat{\text{VaR}}_{\alpha,t} := \widehat{F}_t^{\leftarrow}(\alpha)$ is an estimate of $\alpha$-VaR constructed at time $t-1$.
  - Bank reports $\widehat{\text{VaR}}_{\alpha,t}$ and realized $L_t$.
  - VaR exceedance is simply $\mathcal{I}_t = \mathbb{1}_{\{L_t > \widehat{\text{VaR}}_{\alpha,t}\}}$.

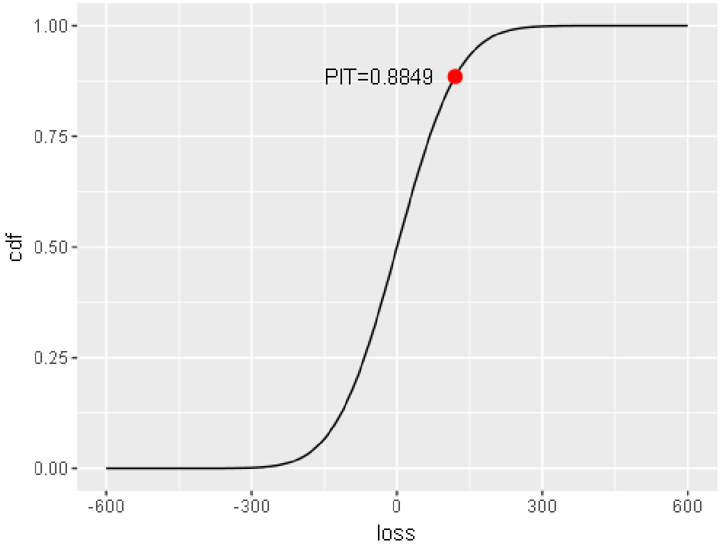
# Probability integral transform

- Define the PIT process given by $P_t = \widehat{F}_t(L_t)$.
- Reported PIT values contain information about VaR exceedances at every level $u$.

$$P_t \geq u \iff L_t \geq \widehat{\text{VaR}}_{u,t}$$

- If the $\{\widehat{F}_t\}$ coincide with the true $\{F_t\}$, then the process $\{P_t\}$ is iid $U[0,1]$.
- In the US, banks on the Internal Models Approach for the trading book have been required to report PIT values to regulators since 2013.
- Motivation: What is the best way to exploit this additional information?

# PIT value from the forecast distribution



# Some quantiles of greater interest than others

- Diebold, Gunther, and Tay (1998) develop forecast density tests based on PIT values. They show how to test the null hypothesis that $\{P_t\}$ is iid $U[0,1]$.
- In a risk-management context, some quantiles of the forecast distribution are more important than others.
  - Accuracy in "good tail" of high profits (low $P_t$) is generally much less important than accuracy in the "bad tail" of large losses (high $P_t$).
  - Models generally cannot expected to perform well in the extreme tail of once-per-generation shock.
- We study a class of backtests for forecast distributions in which the test statistic weights exceedance events by a function of the probability level $\alpha$.
- The choice of the kernel function makes explicit the priorities for model performance.

# Spectral transformations of PIT exceedances

- Our tests are based on transformations of indicator variables for PIT exceedances.
    - We mean "spectral" in the integral transform sense, not in the Durlauf (1991) sense of a transformed autocovariance sequence.
- The transformations take the form

$$W_t = \int_0^1 \mathbb{1}_{\{P_t > u\}} \mathrm{d}\nu(u) = \nu([0, P_t))$$

  where $\nu$ is a Lebesgue-Stieltjes measure defined on $[0, 1]$.
    - $\nu$ is chosen to apply weight to different levels in the unit interval, typically in the region of the standard VaR level $\alpha = 0.99$.
    - We refer to $\nu$ as the kernel measure for the transform.
    - Our framework allows for discrete, continuous, and mixed kernels.
- $W_t$ is (weakly) increasing in $P_t$.

# Spectral backtests

- Spectral backtests are backtests based on $W_1, \ldots, W_n$.
- Null hypothesis. Let $F_W^0$ denote df of $W_t = \nu([0, P_t))$ when $P_t$ is uniform.

$$H_0 : \quad W_1, \ldots, W_n \text{ are iid with df } F_W^0.$$

- A test for the correct distribution $F_W^0$ is a test of unconditional coverage.
- Test of conditional coverage is a joint test for $F_W^0$ and serial independence.
- Broadly, our tests fall into two categories, Z-tests and likelihood ratio tests.
  - Presentation focuses on Z-tests, which anyway tend to outperform the corresponding LRT.

# Spectral Z-tests of unconditional coverage

Z-tests are based on the asymptotic normality under $H_0$ of

$$\overline{W}_n = n^{-1} \sum_{t=1}^{n} W_t$$

- Theorem provides solution to $\mu_W = \mathbb{E}(W_t)$ and $\sigma_W^2 = \text{var}(W_t) = \mathbb{E}(W_t^2) - \mu_W^2$ in the null model $F_W^0$.
  - Note that moments depend only on choice of $\nu$, not the data.
- Trivially follows from CLT that, under $H_0$,

$$Z_n = \frac{\sqrt{n}(\overline{W}_n - \mu_W)}{\sigma_W} \xrightarrow[n \to \infty]{d} N(0,1).$$

# Introducing `spectralBacktest` package

- Our package is designed for extensibility.
  - We pre-define many kernel functions $\nu$, but want to facilitate user customization.
- A kernel may be an instance of a family of kernels that differ in $\text{supp}(\nu)$ and a vector of parameters, e.g., the beta kernel truncated to window $[\alpha_1, \alpha_2]$:

$$d\nu(u) = (u - \alpha_1)^{a-1}(\alpha_2 - u)^{b-1}, \qquad a, b > 0$$

  The package takes a building-block approach to kernel specification.
- Once the kernel is defined, Z-tests can be called with a uniform syntax.

# Tests of unconditional coverage in spectralBacktest

## Calling the Z-test

```
p_value <- spectral_Ztest(kernel, PIT, twosided=TRUE)
```

where

- kernel is a list structure that specifies the kernel function.
- PIT is a vector representing the backtest sample.
- p_value is a scalar *p*-value for the test.

## Apply() to a list of kernel transformations of the same PIT sample

```
p_val_list <- lapply(kernellist,
                     function(kern) spectral_Ztest(kern, PIT))
```

# Monokernel structure

## Examples of discrete and continuous monokernels

```r
linearUp <- list(
  name = 'LinearUp',    # optional description
  type = 'mono',        # other choices:  'bi' or 'multi'
  nu = nu_linear,       # a closure -- see next slide
  support = c(0.95, 0.995),   # rescale kernel to this window
  param = 1       # for nu_linear(), parameter controls slope
)

discreteWeighted4 <- list(
  name = 'Discrete 4-point kernel with non-uniform weights',
  type = 'mono',
  nu = nu_discrete,
  support = c(0.95, 0.98, 0.99, 0.995),
  param = c(1, 2, 4, 2)   # weights on support points
)
```

# Closure: a function written by a function

- In our setting, output function is $\nu$, which transforms vector of PIT values.
- The inputs to the closure are enclosed within the output function, so the latter can stand alone as a representation of the kernel.

## Example of nu closure in the continuous univariate case

```
nu_linear <- function(support=c(0,1), param=1, standardize=TRUE) {
  nu <- function(PIT) {
    # Truncate PIT to the support, then rescale to [0,1]
    P <- truncatePIT(PIT, support, rescale=TRUE)
    W <- P*(1-sign(param)*(1-P))
    if (standardize) {
      mu <- mu_linear(support,param))  # returns c(E[W], E[W^2])
      W <- (W-mu[1])/sqrt(mu[2]-mu[1]^2)
    }
    return(W)  # output of the function nu()
  }
  return(nu)    # the closure returns function nu as an object
}
```

# Bispectral Z-test

Consider $\mathbf{W}_t = (W_t^{(1)}, W_t^{(2)})'$ for two distinct kernel measures on same support.

- Our Z-test generalizes as a $\chi_2^2$ test.
- Straightforward to generalize to higher dimensions ($J$ kernels).
- Calling syntax unchanged but kernel structure takes generalized form.

## Example of continuous bikernel

```
ZAE <- list(
   name = 'Arcsin/Epanechnikov',
   type = 'bi',
   nu = list(nu_arcsin, nu_epanechnikov),
   correlation = rho_arcsin_epanechnikov,  # new element
   support = c(0.985, 0.995),   # common support
   param = list(NULL, NULL)   # element j passed to kernel j
)
```

New element correlation is function returning $\mathrm{Corr}(\nu_1(P), \nu_2(P))$.

# Tests of conditional coverage

- We test for the martingale difference (MD) property $\mathbb{E}(W_t \mid \mathcal{F}_{t-1}) = \mu_W$
- Test equivalent to a test of $\beta = \mathbf{0}$ in a regression of functions of lagged PIT-values on $W_t$.
- Generalizes the Dynamic Quantile test of Engle and Manganelli (2004).
- Unconditional spectral Z-test embedded as special case of $k = 0$ lags.

## Calling the MD-test

```
p_value <- spectral_MDtest(kernel, cvt, PIT)
```

where cvt is a list structure encoding the conditioning variable transformation.

# Conclusions

- The class of spectral backtests embeds many of the most widely used tests of unconditional coverage and tests of conditional coverage.
  - Viewing these tests in terms of the associated kernel functions facilitates construction of new tests.
  - Making explicit the choice of kernel function may help to discipline the backtesting process because the kernel function directly expresses the user's priorities for model performance.
- Empirical application to proprietary data on 10 bank portfolios.
  - Backtest $p$-values may be very sensitive to choice of kernel function and kernel window.
  - Backtest rejects when kernel weights heavily on neighborhoods where EDF of $(P_t)$ departs significantly from uniform.
  - We demonstrate the value to regulators of access to bank-reported PIT-values.
- Vignettes in `spectralBacktest` package demonstrate
  - constructing the `kernel` and `cvt` structures;
  - assessing test size and power with simulations.