Superior Predictive Ability Test: Theory and Application

1 Introduction

In quantitative finance, one often compares multiple trading strategies to identify which yields the highest returns. However, mere outperformance in-sample or even out-of-sample can occur by chance, especially when many strategies are considered. Thus, statistical tests are needed to judge whether an observed performance difference is genuinely significant or a product of data-snooping. The Superior Predictive Ability (SPA) test is designed for this purpose: it evaluates whether any model or strategy in a set significantly outperforms a benchmark, while controlling the overall error rate.

The importance of such tests has been stressed in the literature: as the number of models grows, the probability of "discovering" a seemingly superior model by luck increases toward one [2]. Traditional pairwise tests like the Diebold-Mariano (DM) test compare only two forecasting models at a time and do not account for multiple comparisons. In contrast, the SPA test [1] accounts for data-snooping by using bootstrap methods to adjust for the selection of the best performer. This report explains the theory of the SPA test and its practical use, including how it is applied to compare a Bollinger Bands trading strategy against a benchmark in Python code.

2 Theoretical Background of the SPA Test

When evaluating forecasts or trading strategies, we define a loss function L_{it} for model i at time t, such that smaller losses indicate better performance [1]. For example, one can take $L_{it} = -R_{it}$ where R_{it} is the return of strategy i; then larger returns correspond to smaller losses. Suppose there are k alternative models or strategies and one benchmark model (indexed by 0). Let L_{0t} be the loss of the benchmark and L_{it} be the loss of model i at time t. We define the loss differential $d_{it} = L_{0t} - L_{it}$, so that $d_{it} > 0$ indicates model i beats the benchmark at time t. The average differential over T periods is $\bar{d}_i = \frac{1}{T}\sum_{t=1}^{T} d_{it}$.

The SPA test assesses the null hypothesis that no model has lower expected loss (higher predictive ability) than the benchmark. Formally, one tests

$$H_0 : \max_{1 \le i \le k} E[d_i] \le 0 \quad \text{versus} \quad H_1 : \max_i E[d_i] > 0.$$

Equivalently, in terms of losses,

$$H_0 : E[L_i] \geq E[L_0] \quad \text{for all } i, \quad H_1 : \exists i \text{ s.t. } E[L_i] < E[L_0].$$

This matches the statement that "no competing model can produce a more accurate prediction than the benchmark" under H_0 [2]. Equivalently, H_0 can be written

$$H_0 : \max_i E[L_i] \geq E[L_0],$$

as given in Hansen's formulation [1]. If the null is rejected, it implies at least one model has significantly smaller expected loss than the benchmark (better predictive performance).

Traditional pairwise tests, like Diebold–Mariano, compare two models at a time and test $H_0 : E[L_0] = E[L_i]$, assuming independence of tests. However, when many models are compared sequentially, the chance of spuriously rejecting H_0 grows: as Corradi and Swanson note, with more competitors "the probability of picking an alternative model just by 'luck'... eventually will reach one" [2]. This is the classic data-snooping problem. White's (2000) Reality Check addresses this by using a bootstrap to compute the distribution of the maximum d_i under the joint null [2], thereby controlling the overall error rate. Hansen (2005) improved on White by modifying the test statistic to be "studentized" and focusing on positive outperformance, which makes the test more powerful and less sensitive to poor alternatives [1].

In the SPA test, one typically computes for each model i the t-statistic

$$T_i = \frac{\sqrt{T}\,\bar{d}_i}{\hat{\sigma}_i},$$

where $\hat{\sigma}_i$ is an estimate of the standard deviation of $\sqrt{T}\bar{d}_i$ (for example via block or stationary bootstrap). The SPA test statistic is then $T_{\max} = \max_{1 \leq i \leq k} T_i$. Hansen's method also trims negative $\bar{d}_i$ values (setting them to zero) so that only models that beat the benchmark contribute to the max statistic [1]. A bootstrap procedure then simulates the distribution of $T_{\max}$ under H_0 by resampling the loss differences $\{d_{it}\}$ and recomputing $T_{\max}$ many times, yielding a p-value for the test. If p is below a significance level (e.g. 0.05), we reject H_0 and conclude that at least one trading rule has statistically superior predictive ability.

3 Key Concepts

- **Benchmark and Model Losses:** We define a loss series for the benchmark strategy L_{0t} and for each candidate strategy L_{it}. For example, if R_{it} is the strategy return on day t, one can define $L_{it} = -R_{it}$, so that higher returns mean lower losses.

- **Stationary Bootstrapping:** SPA uses bootstrap resampling that respects the time-series dependence of the data. The *stationary bootstrap* (Politis and Romano, 1994) is often employed: it constructs a bootstrap

sample by drawing blocks of random lengths (geometrically distributed) from the original series, thereby preserving the serial correlation structure.

- **Studentization:** A core feature of Hansen's SPA test is the use of a studentized test statistic. Rather than using raw $\sqrt{T}\bar{d}_i$, one divides by an estimate of its standard deviation.
- **Block Bootstrapping and Block Size:** Alternatively, one can use fixed-length block bootstraps (e.g. circular or moving block bootstrap) to generate resamples. In code, the block length b is a crucial parameter: too small b breaks the dependence structure, too large b yields few independent blocks.

4 Application in Code

In the provided Python code, the SPA test is applied to evaluate a Bollinger Bands trading strategy against a benchmark. The steps are as follows:

- **Construct Bollinger Bands Strategy:** Bollinger Bands are technical indicators consisting of a moving average μ_t of the asset price and upper/lower bands at $\mu_t \pm k\sigma_t$.
- **Compute Strategy Profits:** From the trading signals, the code calculates the profit and loss (PnL) or returns of the Bollinger strategy over time.
- **Form Loss Series:** These returns are converted to losses. If R_{it} is the strategy return on day t, define $L_{it} = -R_{it}$.
- **Call the SPA Function:** The code likely uses a library (such as `arch.bootstrap.SPA`) to conduct the test. One creates an SPA object with the losses.
- **Detailed Explanation of SPA Invocation:** The following snippet performs the SPA test using the `arch` package:

```
spa = SPA(
    benchmark=benchmark_losses,
    models=stratergy_losses,
    reps=1000,
    block_size=100,
    bootstrap="stationary",
    studentize=True,
    nested=False,
    seed=42
)
spa.compute()
```

Here's what each argument does:

- `benchmark=benchmark_losses`: This is the array of loss values for the benchmark strategy.
- `models=stratergy_losses`: A matrix (or list of arrays) where each row/column corresponds to the loss series of a candidate strategy.
- `reps=1000`: Sets the number of bootstrap replications to 1000. This controls the number of simulated SPA test statistics under the null hypothesis to compute the empirical p-value.
- `block_size=100`: Determines the average length of blocks in the stationary bootstrap. A larger block size preserves more of the time-series dependence structure.
- `bootstrap="stationary"`: Specifies that the stationary bootstrap (Politis and Romano, 1994) should be used, which allows blocks of random length to capture time dependence.
- `studentize=True`: Indicates that the test should use Hansen's studentized version of the test statistic, which typically improves finite-sample performance.
- `nested=False`: Disables the nested model structure assumption. With `False`, strategies are treated as distinct, not as improvements upon a base model.
- `seed=42`: Sets a random seed for reproducibility of the bootstrap procedure.

Calling `spa.compute()` runs the test and returns the test statistic and p-value. If the p-value is small (e.g., less than 0.05), the null hypothesis that none of the strategies outperforms the benchmark is rejected.

5 Conclusion

The SPA test provides a rigorous way to compare multiple trading strategies against a benchmark while controlling for data-snooping. In interpreting results, rejecting the SPA null ($p < 0.05$) means that there is statistically significant evidence that at least one strategy outperforms the benchmark. If we fail to reject, we cannot claim any strategy is reliably better. Nevertheless, the SPA test [1, 2] remains a valuable tool in evaluating whether trading strategy performance is robust or merely a product of chance.

References

[1] Peter Reinhard Hansen. "A Test for Superior Predictive Ability". In: *Journal of Business Economic Statistics* 23.4 (2005), pp. 365–380.

[2] Halbert White. "A Reality Check for Data Snooping". In: *Econometrica* 68.5 (2000), pp. 1097–1126.