RelValAnalysis
An R package for portfolio analysis

Ting-Kam Leonard Wong

Department of Mathematics, University of Washington, Seattle

useR!2014  July 1, 2014
library(RelValAnalysis) # "Relative Value Analysis"
data(applestarbucks) # Load data
market <- toymkt(applestarbucks,
                 initial.weight = c(0.5, 0.5))
plot(market)
Apple-Starbucks example

- Benchmark: buy-and-hold starting with weights (0.5, 0.5).
- Portfolio: equal-weighted, rebalanced monthly.

```r
performance <- Invest(market, c(0.5, 0.5), plot = TRUE)
```

Growth of $1

Log relative value
Relative Value Analysis

- Relative value:

\[ V(t) = \frac{\text{growth of $1 of the portfolio}}{\text{growth of $1 of the benchmark}}. \]

- The R package `RelValAnalysis` implements tools for analyzing \( V(t) \) and constructing portfolios:
  - Pal and Wong, The geometry of relative arbitrage (2014).
The equal-weighted portfolio beats the market every time a buy-low is matched with a sell-high.

Diversity: Number of unmatched moves is bounded.

Cross-sectional volatility: A lot of matches.
Fernholz’s decomposition of relative value

portfolio <- ConstantPortfolio(c(0.5, 0.5))
decomp <- FernholzDecomp(market, portfolio, plot = TRUE)
Functionally generated portfolios

Benchmark is capitalization-weighted.

- \((\mu_1(t), \mu_2(t), \ldots, \mu_n(t))\): benchmark weights.
- \((\pi_1(t), \pi_2(t), \ldots, \pi_n(t))\): portfolio weights.

<table>
<thead>
<tr>
<th>Name</th>
<th>Portfolio weights</th>
<th>Generating function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market</td>
<td>(\pi_i = \mu_i)</td>
<td>(\Phi(\mu) = 1)</td>
</tr>
<tr>
<td>Diversity-weighted</td>
<td>(\pi_i = \frac{\mu_i^p}{\sum_{j=1}^{n} \mu_j^p})</td>
<td>(\Phi(\mu) = \left(\sum_{j=1}^{n} \mu_j^p\right)^{\frac{1}{p}})</td>
</tr>
<tr>
<td>Equal-weighted</td>
<td>(\pi_i = \frac{1}{n})</td>
<td>(\Phi(\mu) = \left(\mu_1 \mu_2 \cdots \mu_n\right)^{\frac{1}{n}})</td>
</tr>
<tr>
<td>Entropy-weighted</td>
<td>(\pi_i = \frac{-\mu_i \log \mu_i}{\sum_{j=1}^{n} -\mu_j \log \mu_j})</td>
<td>(\Phi(\mu) = \sum_{j=1}^{n} -\mu_j \log \mu_j)</td>
</tr>
</tbody>
</table>

Table: Examples of functionally generated portfolios
Diversity-weighted portfolio

portfolio <- fgp("Diversity-Weighted Portfolio, p = 0.5", function(x) (sum(sqrt(x)))^2, function(x) sqrt(x)/sum(sqrt(x)))

# This has the same effect as
# portfolio <- DiversityPortfolio(p = 0.5)
decomp <- FernholzDecomp(market, portfolio, plot = TRUE)
Energy-entropy framework

- Portfolio performance is determined by the joint movement of benchmark weights and portfolio weights.

\[ \mu(t) \]: benchmark weights
\[ \pi(t) \]: portfolio weights
Energy-entropy decomposition (Pal and Wong (2013)):

$$\log \frac{V(t+1)}{V(t)} = \Delta_{\text{energy}} + \Delta_{\text{control}} - \Delta_{\text{relative entropy}}.$$ 

- $\Delta_{\text{energy}}$: free energy (cross-sectional volatility).
- $\Delta_{\text{control}}$: rebalance towards the market or away from it?
- $\Delta_{\text{relative entropy}}$: market movement relative to portfolio.
- Want: energy + control accumulates.
Example: Spend constant proportion of energy

- The $\lambda$-strategy in Pal and Wong (2013):

\[
\pi(t) \approx \mu(t+1) \quad \text{drifted rebalance} \quad \pi(t+1)
\]

\[
\tau = \frac{1}{15}
\]
Example: Spend constant proportion of energy

- The $\lambda$-strategy in Pal and Wong (2013):

$$\pi(t) \mu(t+1) \tilde{\pi}(t+1)$$

drifted

consume free energy

$\frac{11}{15}$
Example: Spend constant proportion of energy

- The $\lambda$-strategy in Pal and Wong (2013):

\[
\pi(t) \rightarrow \mu(t+1) \rightarrow \tilde{\pi}(t+1)
\]

\[
\mu(t+1) \quad \text{drifted} \quad \tilde{\pi}(t+1)
\]
Example: Spend constant proportion of energy

- The $\lambda$-strategy in Pal and Wong (2013):

\[
\pi(t) \rightarrow \mu(t+1) \rightarrow \tilde{\pi}(t+1) \rightarrow \pi(t) \\
\text{drifted}
\]
Example: Spend constant proportion of energy

- The $\lambda$-strategy in Pal and Wong (2013):

\[ \pi(t) \rightarrow \tilde{\pi}(t+1) \rightarrow \mu(t+1) \rightarrow \pi(t+1) \]

- Drifted rebalance
Example: Spend constant proportion of energy

- The $\lambda$-strategy in Pal and Wong (2013):

\[
\pi(t) \rightarrow \tilde{\pi}(t+1) \rightarrow \mu(t+1) \rightarrow \pi(t+1)
\]

- Drifted rebalance
- Consume free energy
Energy-entropy decomposition

```r
weight <- GetLambdaWeight(market, lambda = 0.3,
                          initial.weight = c(0.5, 0.5))
decomp <- EnergyEntropyDecomp(market, weight, plot = TRUE)
```
Hierarchical portfolios

- Portfolio of portfolios.
- Countries → sectors → stocks (grouping).
- Attribute performance at each level.

- Pal and Wong (2013):

\[
\Delta \text{energy} = \Delta \text{energy across sectors} + \sum \lambda_i \Delta(\text{energy within sector } i)
\]

\[
\Delta \text{RE} = \Delta \text{RE across sectors} + \sum \lambda_i \Delta(\text{RE within sector } i)
\]
Hierarchical energy-entropy decomposition

```r
# See example(EnergyEntropyDecomp)
model <- AtlasModel(n = 6, g = 0.1, sigma = 0.2)
market <- SimMarketModel(model)  # default settings
grouping <- c(1, 1, 2, 2, 2, 2)
weight <- GetWeight(market,
                     EntropyPortfolio$weight.function)
decomp <- EnergyEntropyDecomp(market, weight, grouping)
```

![Energy-entropy decomposition](attachment:image.png)
Thank you!