Statistical Finance for Investors Unfamiliar with Quantitative Methods Using stockPortfolio in R

Nicolas Christou
David Diez
UCLA Department of Statistics

Overview

The stockPortfolio package is very easy to navigate. There are three simple steps to select an optimal portfolio using three functions:

1. Download data (automated) using a vector of stock tickers and a range of dates.

```
> stockData <- getReturns(vectorOfTickers,
+   start="2004-09-01", end="2009-09-01")
```

2. Model the stocks in one of the four offered models.

```
> model1 <- stockModel(stockData)
> model2 <- stockModel(stockData, model="CCM")
...
```

3. Identify the optimal portfolio.

```
> optPort <- optimalPort(model1)
```

Obtaining data

- Pick your stocks and get their tickers in a vector (`ticker`).
- Raw stock data are the stock *prices*.
- The function `getReturns` retrieves the adjusted close prices from http://finance.yahoo.com and computes the *returns*.
- A return is just the percent gain/loss in decimal form: 10.3% gain means a return of 0.103.

```
> ticker <- c("C", "IBM",
+   "JPM", "WFC")
> gR <- getReturns(ticker,
+   start="2005-01-01",
+   end="2009-10-01")
```

Function declaration: getReturns

```
getReturns(ticker,
     freq = c("month", "week", "day"),
     get = c("overlapOnly", "all"),
     start = "1970-01-01",
     end = NULL)
```

The default, `overlapOnly`, will return the stock returns for which all stocks had data and drop any dates with NA.
Warning: setting `get="all"` often results in problems with missing values.

Example

We will use the tickers from the stock data in stock94Info.

```
> data(stock94Info)
> ticker <- stock94Info$ticker
 [1] "C"     "KEY"   "WFC"   "JPM"   "SO"    "DUK"
 [7] "D"     "HE"    "EIX"   "LUV"   "CAL"   "AMR"
[13] "AMGN"  "GILD"  "CELG"  "GENZ"  "BIIB"  "CAT"
[19] "DE"    "HIT"   "IMO"   "MRO"   "HES"   "YPF"
[25] "^GSPC"
> ind <- stock94Info$industry # for later

> theData <- getReturns(ticker,
+   start="2004-09-01", end="2009-09-01")
```

The print, summary, and plot methods can be applied to theData.

Types of investments

Other investments also exist, and the `stockPortfolio` takes into account: risk-free investments and short selling.

- Argument name in `stockPortfolio`: `Rf`. The value of `Rf` is standardized for the period, e.g. 3% annual return equates to setting `Rf=0.0025` for monthly data.
- Short selling will be referred to via `shortSelling` in function arguments, and it takes values `"y"` and `"n"`.

Modeling stocks

There are four models offered:

- **Variance-covariance (default).** Use $\bar{R}$, Σ, and R_f to select a portfolio that minimizes risk but maximizes return.
- **Constant correlation model (CCM).** Smooth Σ and then do variance covariance method.
- **Multigroup model (MGM).** Compromise strategy: do some smoothing on Σ (less than CCM) and then optimize.
- **Single index model (SIM).** Use a linear model to analyze stock behavior, where we regress stock returns against some stock index.

Implementation

The 25^{th} ticker – the S&P500 – is dropped for the first three models.

```
> model1 <- stockModel(theData, drop=25)
> model2 <- stockModel(theData, drop=25, model="CCM")
> model3 <- stockModel(theData, drop=25, model="MGM",
+   industry=ind)
> model4 <- stockModel(theData, model="SIM", index=25)
```

By default, `Rf=0` and `shortSelling="y"`. Short selling is always permitted when the model is the variance-covariance or multigroup model.

Single index model

The *Single Index Model* is the most well-known of the four models. If R_M describes the returns of the stock index (S&P500) and R_i describes the returns of stock i, then the linear model that relates the two:

$$R_i = \alpha_i + \beta_i R_M + \epsilon_i$$

where α_i and β_i are constants and ϵ_i is a vector of the model errors for stock i. Example where no short selling is permitted:

```
> sim <- stockModel(theData, model="SIM", index=25,
+   industry=ind, shortSelling="n")
```

Finding the optimal portfolio

For any model, the goal is to minimize risk while maximizing return. There is a single function to identify the optimal portfolio of a model: `optimalPort`.

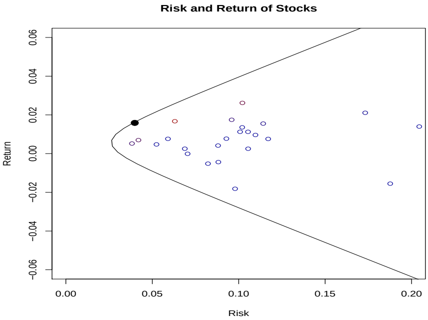
The first argument is an output of `stockModel`. The next two arguments permit adjustments to the model (`Rf` and `shortSelling`).

```
> simOP <- optimalPort(sim)

> summary(simOP)
Model:  single index model
Expected return:  0.0159
Risk estimate:    0.0399

> simOP
...  same output as above ...

Portfolio allocation:
...
```

Visualization of optimal portfolio

The optimal stock portfolio is shown by the black dot on the efficient frontier when no short selling is permitted. Allocation shown by heat coloring.

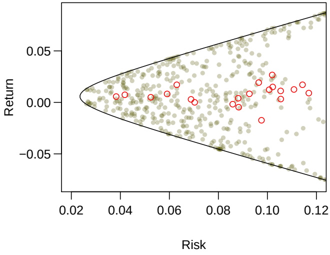
```
> plot(simOP, xlim=c(0,0.2),
+   ylim=0.06*c(-1,1))
> portPossCurve(sim, 10,
+   add=TRUE)
```



Other topics

Finding the optimal allocation using one of the models would be a relatively simple task using getReturns, stockModel, and optimalPort. What was not covered:

- Finer details of the models.
- Comparison of models (testPort is useful in this respect).
- Creation of portfolio clouds (portCloud) and portfolio possibilities curves (portPossCurve).



Example of testPort

Before farewells, a brief examination of the utility of these models (using testPort).