Week 10 Newton's Method

Zhe Zhao

Stevens Institute of Technology

zzhao6@stevens.edu

November 9, 2015

Agenda

1. Project Ideas

2. Newton's Method

Project Ideas

R Packages

Packages Covered:

- lubridate
- fBasics
- quantmod
- TRTH
- Rmarkdown

Suggested Packages

- plyr: Tools for splitting, applying and combining data
- car: Applied Regression.
- ggplot2: An implementation of the grammar of graphics in R.
- ...

Connect to Bloomberg

- Package: Rbbg
  - Have to use on a Bloomberg Terminal
  - Dependencies: Java, rJava, Bloomberg Java API V3

Example

```
# sample 1
> conn <- blpConnect()
> bdp(conn, "AMZN US Equity", "NAME")

# sample 2
> securities <- c("AMZN US Equity", "OCN US Equity")
> fields <- c("NAME", "PX_LAST", "TIME", "SETTLE_DT",
"HAS_CONVERTIBLES")
> bdp(conn, securities, fields)
```

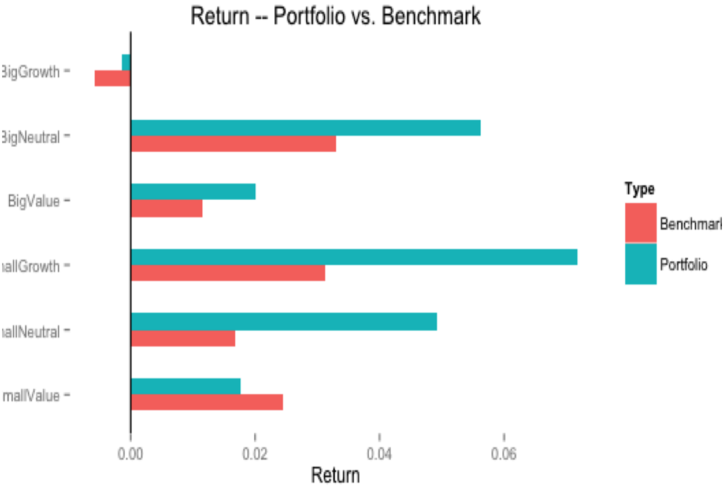
Connect to Database

- Package: RPostgreSQL
    - A database interface and PostgreSQL driver for R
    - It could let you connect to database and execute database command through R
    - Structured Query Language, PostgreSQL, MySQL, SQL Server

Portfolio Perfomance Analysis

- Package: pa
- Objective:
  - Create a portforlio based on the Fama & Frech style factors.
  - Apply Brinson model which is build in the package "pa" to analyze the Brinson.

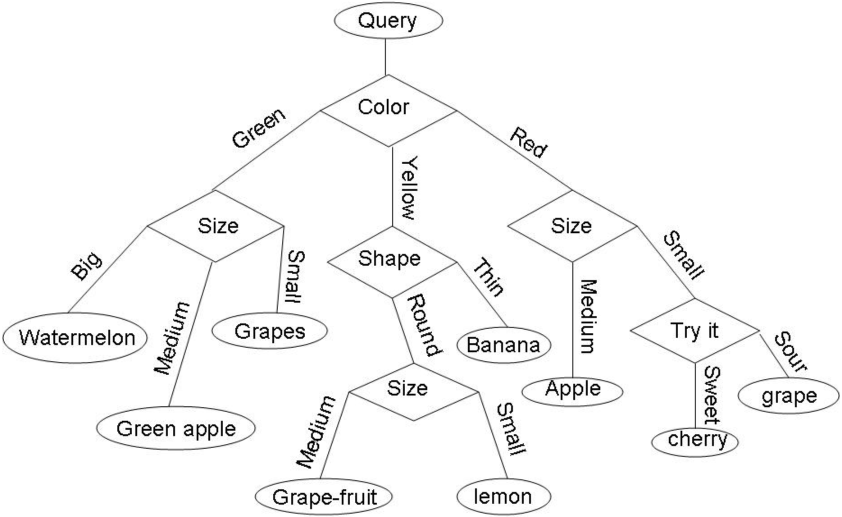
Portfolio Perfomance Analysis



Decision Trees

- Package: Party
  - Decision Trees are used as a classification tool in machine learning.
  - Data sets consist of large number of data points, where each data point is defined based on the set of the attributes and a label.
  - Decision Trees are a predictive method based on a branching series of Boolean tests

Decision Trees



Trading Indicators

- Package: TTR
  - TTR package serves as an excellent platform for developing and back-testing technical indicators.
  - Functions and data to construct technical trading rules with R
  - R topics documented: adjRatios, ADX, aroon, ATR, Bbands, CCI, chaikinAD, chaikinVolatility, CLV, CMF, CMO, DonchianChannel, DPO, DVI, EMV, GMMA, KST, lags, MACD, MFI, OBV, PBands, ROC, rollSFM, RSI, runPrecentRank, runSum, SAR, SMA, stoch, stockSymbols, TDI, TRIX, TTR, ttrc, VHF, volatility, williamsAD, WPR, ZigZag

Newton's Method

- The Newton's Method, is a powerful technique for solving equations numerically.
- It is based on the Taylor Series.
- Usually converge on a root with devastating efficiency.

Newton's Method

Assume r is one root of function $f(x)$. Let x_0 be a "good guess" of r, $r = x_0 + h$. Then we have the following:

Newton's Method

$$f(r) = f(x_0 + h) = f(x_0) + h * f'(x_0) \tag{1}$$

$$h = -\frac{f(x_0)}{f'(x_0)} \tag{2}$$

$$r = x_0 + h \approx x_0 - \frac{f(x_0)}{f'(x_0)} \tag{3}$$

Newton's Method

Actually, by Taylor's expansion:

$$
\begin{aligned}
f(x) &= \sum_{i=0}^{\infty} \frac{f^n(a)}{n!}(x-a)^n \\
&= f(a) + f'(a)(x-a) + \ldots
\end{aligned}
$$

Substitute $x = r, a = x_0$, we have:

$$f(r) \approx f(x_0) + h * f'(x_0) \tag{4}$$

Newton's Method

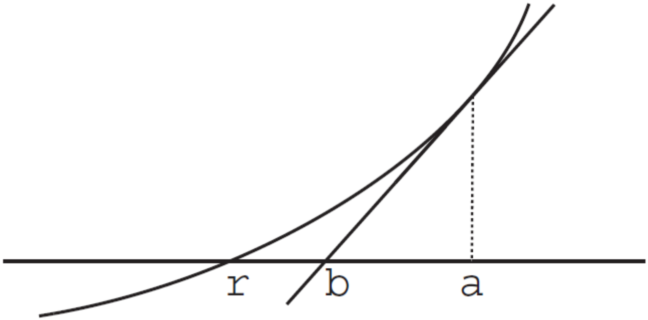
Our new improved estimate x_1 is therefore given by

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Continue in this way, if x_n is the current estimate, then the next estimate x_{n+1} is given by:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Geometrical Interpretation



Newton's Method

Here is a simple example: $f(x) = x^2 - 2$

$$x_{n=1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^2 - 2}{2x_n}$$

- $x_0 = 1,\ \Delta x = \frac{1-2}{2} = -0.5,\ x_1 = 1.5$
- $x_0 = 1.5,\ \Delta x = \frac{1.5^2-2}{3} = 0.08,\ x_1 = 1.42$
- $x_0 = 1,\ \Delta x = \frac{1.42^2-2}{2.84} = 0.006,\ x_1 = 1.414$

Theoretically,

$$x^2 - 2 = 0, x = \pm\sqrt{2} \approx \pm 1.414$$

Newton's Method

Algorithm

```
loop {
```

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

```
    # until converge

}
```

Newton's Method

Compare with gradient, we have almost the same implementation in R:

Example (pseudo code)

```
# set initial
# set convergence condition
# loop:
while(1)
{
    # algorithm (probably the only difference)
    # ...
    # if (converge)
    # {
    #    break
    # }
}
```

Newton's Method in R

Use Newton's Method to solve the root of a function: $f(x) = x^2 - 2$

Example

```
# f(x) = x^2 - 2
fx <- function(x)
{
    y = x^2 - 2
    return (y)
}

# f'(x) = 2*x
dfx <- function(x)
{
    y = 2*x
    return(y)
}
```

Newton's Method in R

Example

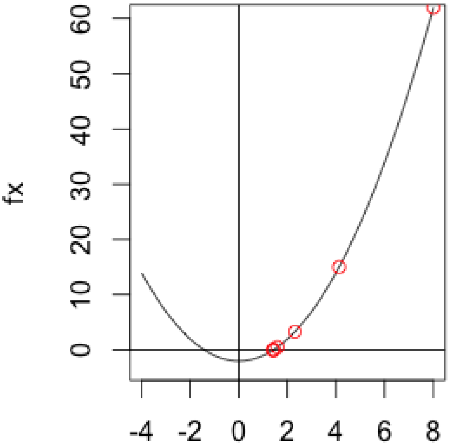
```
# initial plot
plot(fx, ylim = c(-3, 60), xlim = c(-4, 8))
abline(h=0, v=0)

x0 = 8            # initial guess
epsilon = 0.01    # convergent condition
points(x0, fx(x0), col = "red")

step = 1          # step count
```

Newton's Method in R

Example

```
while(1)
{
    tmp = x0
    deltaX = fx(x0) / dfx(x0)
    x0 = x0 - deltaX
    points(x0, fx(x0), col = "red")
    step = step + 1
    # convergence
    if (abs(x0-tmp) < epsilon)
    {
        points(x0, fx(x0), col = "red")
        print(paste("x0 = ", x0, ", step = ", step, sep=""))
        break
    }
}
```

Newton's Method in R



Example: Calculate the Bond Yield

We have a bond, paying coupon of \$3 every 6 months. The maturity is 2 years and the face value is \$100. If the bond price is \$98.39, calculate the yield of the bond using Newton's Method.

$$3e^{-0.5y} + 3e^{-1y} + 3e^{-1.5y} + 103e^{-2y} = 98.39$$

- $f(y) = 3e^{-0.5y} + 3e^{-1y} + 3e^{-1.5y} + 103e^{-2y} - 98.39$
- $\frac{df}{dy} = -1.5e^{-0.5y} - 3e^{-1y} - 4.5e^{-1.5y} - 206e^{-2y}$

Example: Calculate the Bond Yield

Example

```
bond <- function(y) {
value <- 3 * exp(-0.5 * y) + 3 * exp(-1 * y) +
         3 * exp(-1.5 * y) + 103 * exp(-2 * y) - 98.39
return(value)
}

dbond <- function(y) {
value <- -1.5 * exp(-0.5 * y) + -3 * exp(-1 * y) +
 -4.5 * exp(-1.5 * y) + -206 * exp(-2 * y)
return(value)
}
```

Example: Calculate the Bond Yield

Example

```
y0 = 0.02
while(1)
{
    y1 <- y0 - bond(y0) / dbond(y0)
    if(abs(y0-y1) < 1e-5)
    {
        print("converged!")
        cat("y = ", y1, sep = "")
        break
    }
    y0 <- y1
}
```