Optimización de portafolios

Esthefany Ortiz Gutierrez 148571
Allan Eduardo Rosas García 160630
Antonio Manguart Páez 175701

Mayo 2019

1 Introducción

El objetivo de este trabajo es la implementación de la solución al modelo de selección de portafolios de Markowitz programación convexa y revisar la teoría que origina el problema, para la solución principal se ha usado la librería CVXOPT [1] de python que sirve para resolver problemas de optimización convexa pero también se presenta una solución por simulación.

2 Antecedentes

La teoría moderna de portafolio [2] (MPT) es una teoría de inversión que busca maximizar el retorno y minimizar el riesgo. Esta teoría fue desarrollada por el economista Henry Markowitz en 1952.

Con anterioridad al trabajo de Markowitz, por 1938, John Burr Williams [3] escribió un libro llamado "La teoría del valor de la inversión" donde describe: el modelo de descuento de dividendos donde el objetivo de hacer una buena inversión consistía en seleccionar activos individuales que presentaran buenas rentabilidades por dividendo y buenas perspectivas.

MPT se basa en construir carteras de activos para maximizar el retorno esperado en función del nivel de riesgo. Bajo este punto de vista una alta rentabilidad va necesariamente acompañada de un nivel más alto de riesgo.

El modelo de Markowitz se basa en varios supuestos sobre el comportamiento de los inversores, en [LL10] se enlistan los siguientes:

- Los inversores pueden estimar una distribución de probabilidad de posibles rendimientos durante un período entre la compra y venta.
- Los inversores tienen funciones de utilidad por períodos únicos en los que maximizan la utilidad en el marco minimizar el monto marginal invertido.
- Los inversores utilizan la variabilidad sobre los posibles valores de rendimiento para medir el riesgo.
- Los inversores sólo utilizan el rendimiento esperado y el riesgo para tomar decisiones de inversión.
- El rendimiento esperado y el riesgo son medidos por los dos primeros momentos de la distribución de probabilidad del rendimiento, el valor esperado y la variación.
- El retorno es deseable; El riesgo debe ser evitado.

[1] el sitio oficila puede ser cosultado en [Mar]
[2] Ver [LL10] para mas detalles
[3] Este documento [Wil38] describe el pensamiento de la epoca antes de Markowtiz

3 Teoría modernda de portafolios

Markowitz sugiere dos enfoques de optimización restringida para obtener los pesos óptimos de la cartera:

- Minimizar el riesgo o la variación de la cartera, sujeto a que la cartera alcance alguna tasa esperada de retorno, y también sujeto a que la suma de los pesos de la cartera sean igual a uno.
- Máximizar la tasa esperada del retorno del portafolio sujeto a un riesgo dado y con la suma de los pesos de la cartera igual a uno.

3.1 Activos y portafolios

Un protafolio es una inversión compuesta por N activos A_n, con retornos R_n para $n \in \{1, 2, 3, ..., N\}$ donde se ha invertido una cantidad W en todo el portafolio. Sea W_n [4] la cantidad invertida en el activo A_n podemos escribir

$$\sum_{n=1}^{N} W_n = W \tag{1}$$

Consideramos $w_n = \frac{W_n}{W}$ y expresamos esta suma en términos de sus valores realtivos

$$\sum_{n=1}^{N} w_n = 1 \tag{2}$$

y

$$\sum_{n=1}^{N} w_n R_n = R \tag{3}$$

El retorno esperado del portafolio se define como:

$$\rho = E(R) = E(\sum_{n=1}^{N} w_n R_n) = \sum_{n=1}^{N} w_i r_n \tag{4}$$

Donde $r_n = E(R_n)$ es el retorno esperado de cada activo A_n

Se define la covarianza entre cualesquiera dos activos i y j del portafolio como:

$$s_{ij} = E[(R_i - r_i)(R_j - r_j)] \tag{5}$$

De esta forma podemos expresar la matriz de varianzas y covarianzas

[4] Valores negativos de W_n son interpretados como short selling

$$S = \begin{bmatrix} s_{11} & s_{12} & ... & s_{1N} \\ s_{21} & s_{22} & ... & s_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N1} & s_{N2} & ... & s_{NN} \end{bmatrix} \tag{6}$$

Por último para terminar de caracterizar el portafolio se define el riesgo como la la medida que cuantifica que tanta dispersión existe alrededor del retorno esperado entonces tomamos a varianza del portafolio como esta medida y la expresamos como:

$$s^2 = E(|R - \rho|^2) = \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j s_{ij} = w^T S w \tag{7}$$

Donde $w = [w_1, w_2, ..., w_n]^T$ es el vector de pesos (inversión de cada activo)

3.2 Formulación del problema

En este documento nos centramos en resolver el problema con el enfoque de minimización establecido por Markowitz el cual indica que un portafolio es optimo si para un retorno esperado ρ el portafolio tiene mínima vairanza s^2. Encontrar tal portafolio requerie resolver el siguiente problema con restricciones:

$$w = arg \min_{w} \{w^T S w\} \tag{8}$$

Sujeto a

$$w^T u = \sum_{n=1}^{N} w_n = 1 \tag{9}$$

$$w^T r = \sum_{n=1}^{N} w_n r_n = \rho \tag{10}$$

Donde $u = [1, 1, ..., 1]^T$ y $r = [r_1, r_2, ..., r_N]^T$

Entonces el problema de optimización de portafolios es un problema de optimización cuadrática con restricciones [5]

[5] En [CT06] se pueden consultar la construcción de ejemplos simples

4 Datos usados para ajustar pesos

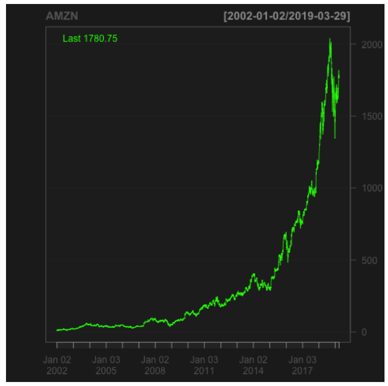
Los datos que usaremos para ajustar el modelo corresponden a acciones de las siguientes empresas:

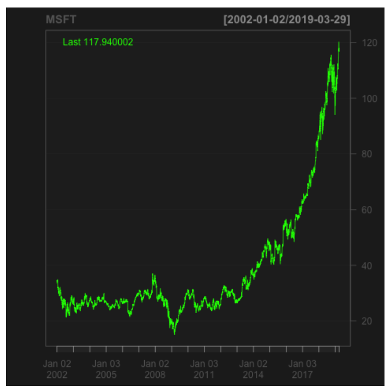
- apple
- amazon
- microsoft
- coca cola
- pepsi
- bank of america
- jp morgan
- nike
- oracle
- IBM
- procter and gamble
- walmart

Para obtener estos datos se creó un script [6] en R para automatizar la descarga

Todas bajadas desde: 2002-01-01 a 2019-03-31, se va a tomar el rendimiento mensual. Ejemplos (apple, amazon, microsoft):

[6] El script se encuentra en el apéndice A





Por ejemplo, si tomamos solo estas 3 acciones, un portafolio podría estar dado por:

$$P = 0.33 * apple + 0.33 * microsoft + 0.33 * amazon$$

4.1 Solución por simulación

La primera solución que trabajaremos está dada por simulación, es un método heurístico pero muy simple de implementar y el obetivo es tener un primer resultado sin entrar en mucha teoría

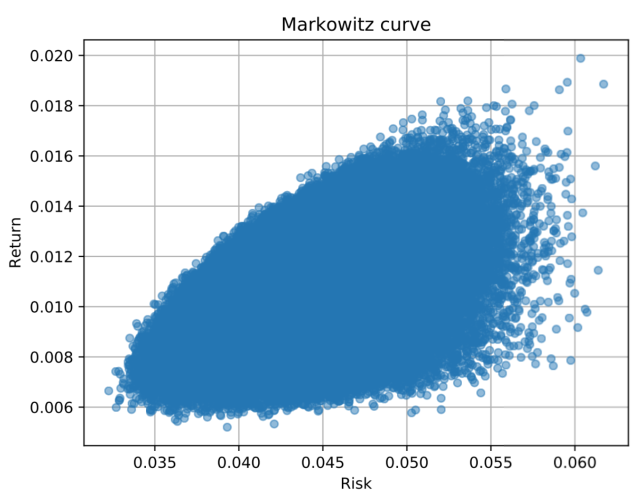
Supongamos que se tienen n activos, para simular un portafolio lo único que necesitamos es obtener la matriz de covarianza y la esperanza de los rendimientos de cada activo.

Result: Simulación de portafolio [a]
tasa de aprendizaje α, momentum μ;
$\mathrm{y}_0 = x_0$
for *t=0 to total simulaciones* **do**
  W = obten un vector que sume 1 aleatorio
  $\mathrm{E}(\mathrm{R}_p) = W^T R$
  $\sigma_p^2 = W^T S W$
end

[a]El código en python puede ser consultado en el apéndice B

return W

Al final tenemos un cantidad muy grande de portafolios aleatorios, y para cada uno vemos cual es su retorno y su rendimiento. Si extraemos esta información y calculamos la raíz cuadrada de la varianza por cada portafolio podemos graficarlo de la siguiente forma.



Hay muchos casos donde para un riesgo dado existen mayores rendimiento, y donde se obtiene el máximo por cada activo le llamamos frontera eficiente, que es la linea superior de los puntos azules.

5 Optimización

El problema de optimización de portafolios puede ser resuelto de la siguiente manera:

1. Incorporar las restricciones a la función objetivo por medio de penalizaciones (multiplicadores de lagrange)
2. Expresar el problema via la matriz de Karush–Kuhn–Tucker
3. Resolver por algún método de optimización numérica (descenso, newton, etc.)

Entonces considerando el problema original:

$$\min_{w} w^T S w \tag{11}$$

sujeto a:

$$\sum_{i=1}^{n} w_i = 1$$

$$\sum_{i=1}^{n} w_i r_i = \rho$$

Incorporamos las restricciones:

$$L(w, \lambda_1, \lambda_2) = w^T S w - \lambda_1 (w^T u - 1) - \lambda_2 (w^T r - \rho)$$

la condición de optimalidad indica que para resolver el problema general:

$$\min_{x} f(x) \tag{12}$$

sujeto a:

$$Ax = b$$

se debe cumplir:

- f es convexa y 2 veces continuamente diferenciable
- $A \epsilon R^{pxn}$ con rango $A = p$
- Asumimos que p^* es finita y acotada

y las condiciones de optimalidad: x^* es optimal si y sólo si existe v^* tal que

$$\nabla f(x^*) + A^T v^* = 0$$

$$Ax^* = b$$

Entonces para el problema general de minimización cuadrática [7] con restricciones de igualdad

$$\min_x \frac{1}{2}x^T Px + q^T x + r \tag{13}$$

sujeto a:

$$Ax = b$$

escribimos las condiciones de optimalidad de la siguiente forma:

$$\begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ v^* \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix}$$

Ejemplo: En el caso del problema de optimización de portafolios si consideramos los activos A y B las condiciones de optimalidad quedan expresadas de la siguiente forma:

$$\begin{bmatrix} 2\sigma_A^2 & 2\sigma_{AB} & 1 & r_A \\ 2\sigma_{AB} & 2\sigma_B^2 & 1 & r_B \\ 1 & 1 & 0 & 0 \\ r_A & r_B & 0 & 0 \end{bmatrix} \begin{bmatrix} w_A^* \\ w_B^* \\ \lambda_1^* \\ \lambda_2^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \rho \end{bmatrix}$$

Para encontrar los pesos tendriamos que resolver el sistema de ecuaciones.

5.1 Solución al problema de optimización de portafolios por el método de Newton

El problema consiste en resolver la aproximación de segundo orden, ver [NW06]

$$\min \hat{f}(x+v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v \tag{14}$$

sujeto a:

$$A(x+v) = b$$

De la linearización de las condiciones de optimalidad se sigue :

$$\nabla f(x+v) + A^T w \approx \nabla f(x) + \nabla^2 f(x) v + A^T w = 0 \tag{15}$$

$$A(x+v) = b$$

El paso de Newton Δx_{nt} de f en un x factible esta dado por la solución v de

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} -\nabla f(x) \\ 0 \end{bmatrix}$$

Ejemplo: En el caso del problema de optimización de portafolios si consideramos los activos A y B se debe resolver:

[7] Consultar el capítulo 4.4 de [BV04]

$$\begin{bmatrix} 2\sigma_A^2 & 2\sigma_{AB} & 1 & r_A \\ 2\sigma_{AB} & 2\sigma_B^2 & 1 & r_B \\ 1 & 1 & 0 & 0 \\ r_A & r_B & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta w_A^* \\ \Delta w_B^* \\ \lambda_1^* \\ \lambda_2^* \end{bmatrix} = \begin{bmatrix} -(2\sigma_A^2 w_A + 2\sigma_{AB} w_B) \\ -(2\sigma_{AB} w_A + 2\sigma_B^2 w_B) \\ 0 \\ 0 \end{bmatrix}$$

5.2 Algoritmo programado en CVXOPT para la solución al problema de optimización de portafolios

La libreria CVXOPT nos ayuda a resolver problemas de optimización cuadrática con restricciones donde sólo tenemos que especificar el problema en su forma general:

$$\min \frac{1}{2} x^T P x + q^T x \tag{16}$$

sujeto a:

$$Gx \leq h$$

$$Ax = b$$

Cargamos las librerias que usaremos

```
import numpy as np
import pandas as pd
from cvxopt import blas, solvers, matrix
```

Cargamos los datos con los que vamos a trabajar

```
return_data = pd.read_csv('monthly_return.csv')
return_data = return_data[[i for i in return_data.keys() if i not in ('date')]]
return_data = return_data.T.values
```

Obtenemos el vector de rendimientos promedio y la matriz de varianzas y covarianzas

```
mean_returns = np.mean(return_data, axis=1)
sigma_returns = np.cov(return_data)
```

Especificamos los parametros de nuestro problema y creamos una función para encontrar los pesos optimos

```
def iif(cond, iftrue=1.0, iffalse=0.0):
    return iftrue if cond else iffalse

def markowitz(mu, sigma, mu_p):
    d = len(sigma)
    P = matrix(sigma)
    q = matrix([0.0 for i in range(d)])

    G = matrix([
        [(-1.0) ** (1 + j % 2) * iif(i == j / 2) for i in range(d)]
```

```
        for j in range(2 * d)
    ]).trans()
    h = matrix([iif(j % 2) for j in range(2 * d)])

    A = matrix([
        [1.0 for i in range(d)],
        list(mu)
    ]).trans()
    b = matrix([1.0, float(mu_p)])

    sol = solvers.qp(P, q, G, h, A, b)

    w = list(sol['x'])
    return w
```

Probamos con nuestros datos y un rendimiento de .01

```
markowitz(mean_returns, sigma_returns, .01)
```

El cual conduce a los siguientes pesos

```
[0.0990662417729221,
 0.05936173379582145,
 0.0025740953038260296,
 0.10890647475406469,
 0.02990591823347158,
 1.6079465508885243e-05,
 2.430148116581949e-06,
 0.1349500435488952,
 0.03339997641183987,
 3.617924292545396e-05,
 0.32930641692842116,
 0.20247441039418715]
```

6 Apendice

A Código en R para bajar acciones

```
library(quantmod)
library(corrplot)
# Se establecen los parametros de fechas
start_date = as.Date("2002-01-01")
end_date= as.Date("2019-03-31")

# Se establecen los stocks
tickets <- c("AAPL", "AMZN", "MSFT", "KO", "PEP", "JPM",
             "BAC", "NKE", "ORCL", "IBM", "PG", "WMT")
portafolio_daily_prices <- NULL
portafolio_monthly_return <- NULL
for (Ticker in tickets){
   # Obtener precios
   prices <- getSymbols.yahoo(Ticker,
                              from=start_date, to = end_date,
                              periodicity = "daily", auto.assign=FALSE)[,4]
   portafolio_daily_prices <- cbind(portafolio_daily_prices, prices)
   # Generar grafico
   png(paste(Ticker, ".png", sep = ""))
   chartSeries(prices, name = Ticker)
   dev.off()
   # Rendimiento mensual
   portafolio_monthly_return <- cbind(portafolio_monthly_return,
                                      monthlyReturn(prices))
}

# Se limpian los datos para ser guardados
names_portafolio_prices <- names(portafolio_daily_prices)

replace_close <- function(str){
   return(gsub(pattern = ".Close", x = str, replacement = ""))
}

names_columns <- sapply(names_portafolio_prices, replace_close)

names(portafolio_monthly_return) <- names_columns
names(portafolio_daily_prices) <- names_columns

# Se escribe a CSV
portafolio_daily_prices <- data.frame(portafolio_daily_prices)
portafolio_monthly_return <- data.frame(portafolio_monthly_return)
```

```
portafolio_monthly_return$date <- as.Date(index(portafolio_monthly_return))
portafolio_daily_prices$date <- as.Date(index(portafolio_daily_prices))

write.csv(x = portafolio_monthly_return,
          file = "monthly_return.csv", row.names=FALSE)
write.csv(x = portafolio_daily_prices,
          file = "daily_price.csv", row.names=FALSE)
```

B Código en Python para la simulación

```
def generate_portfolio():
    """
    Para una matriz dada con retornos, se genera un portafolio aleatorio
    """
    weights = np.random.random(len(return_data))
    weights /= np.sum(weights)

    x = np.asmatrix(mean_returns)
    weights_matrix = np.asmatrix(weights)
    covariance_matrix = np.asmatrix(sigma_returns)

    expected_return = weights_matrix * x.T
    risk = np.sqrt(weights_matrix * covariance_matrix * weights_matrix.T)
    return expected_return, risk
```

Referencias

[Wil38] John Burr Williams. *The theory of investment value.* Tech. rep. 1938.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[CT06] Gerard Cornuejols and Reha Tütüncü. *Optimization methods in finance.* Vol. 5. Cambridge University Press, 2006.

[NW06] Jorge Nocedal and Stephen Wright. *Numerical optimization.* Springer Science & Business Media, 2006.

[LL10] Cheng-Few Lee and John Lee. *Handbook of quantitative finance and risk management.* Springer Science & Business Media, 2010.

[Mar] Joachim Dahl Martin Andersen. *Python Software for Convex Optimization.* URL: `https://cvxopt.org/`. (accessed: 01.05.2019).