

Optimizador de Carteras de Inversión

Implementación Interactiva del Modelo de Markowitz

Fernando Mamani Machaca

Métodos de Optimización

28 de febrero de 2025

Índice

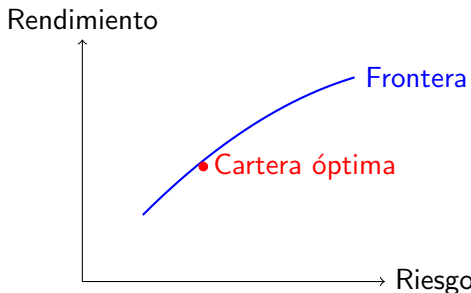
- 1 Introducción
- 2 Nuestra Aplicación
- 3 Matemáticas del Modelo
- 4 Estructura del Código
- 5 Características Principales
- 6 Limitaciones y Mejoras
- 7 Conclusiones

¿Qué es el Modelo de Markowitz?

- Desarrollado por Harry Markowitz en 1952
- Base de la Teoría Moderna de Carteras (MPT)
- Premio Nobel de Economía (1990)
- Busca construir carteras óptimas balanceando:
 - Maximizar rendimiento esperado
 - Minimizar riesgo (volatilidad)
- Introduce el concepto de **diversificación cuantitativa**

Principios clave:

- Rendimiento esperado
- Volatilidad (riesgo)
- Correlación entre activos
- Diversificación eficiente
- Frontera eficiente



¿Qué hace nuestra aplicación?

Características principales

- Aplicación web interactiva construida con Streamlit
- Implementa el modelo de Markowitz para optimizar carteras
- Permite al usuario:
 - Seleccionar acciones o activos financieros
 - Definir su nivel de aversión al riesgo
 - Establecer restricciones (ej. peso máximo por activo)
 - Visualizar la frontera eficiente
 - Simular rendimientos futuros (Monte Carlo)

1 Entrada de datos:

- Selección de símbolos de activos (ej. AAPL, MSFT)
- Periodo de tiempo para análisis
- Preferencias de riesgo-rendimiento

2 Procesamiento:

- Descarga de datos históricos
- Cálculo de rendimientos y volatilidades
- Optimización de carteras

3 Resultados:

- Visualización de la frontera eficiente
- Composición de la cartera óptima
- Métricas de rendimiento y riesgo
- Simulaciones de escenarios futuros

Problema de optimización

Sea w el vector de pesos de los activos en la cartera:

$$\text{Maximizar } w^T \mu - \lambda w^T \Sigma w$$

$$\text{Sujeto a } \sum_{i=1}^n w_i = 1$$

$$0 \leq w_i \leq w_{\max} \quad \forall i \in \{1, 2, \dots, n\}$$

donde:

- μ : Vector de rendimientos esperados
- Σ : Matriz de covarianzas
- λ : Parámetro de aversión al riesgo
- w_{\max} : Peso máximo por activo

Estructura General

```
1 # Estructura general de la aplicaci n
2 import streamlit as st
3 import pandas as pd
4 import numpy as np
5 import yfinance as yf
6 import plotly
7
8 # 1. Configuraci n de la interfaz
9 st.set_page_config(...)
10 st.title(...)
11
12 # 2. Entrada de datos
13 with st.sidebar:
14     # Par metros de la cartera
15     # Selecci n de activos
16     # Preferencias de riesgo
17
18 # 3. Funciones de optimizaci n
19 def calcular_rendimientos(precios):
20     # ...
21
22 def cartera_optima(rendimientos, aversi n a riesgo):
```


Optimización de Carteras

```
1 def cartera_optima(rendimientos, aversion_riesgo, max_peso
   =1.0):
2     """Encuentra la cartera ptima seg n la aversi n al
   riesgo."""
3     num_activos = len(rendimientos.columns)
4     args = (rendimientos, aversion_riesgo)
5     restricciones = ({
6         'type': 'eq',
7         'fun': lambda x: np.sum(x) - 1
8     })
9     limite = tuple((0, max_peso) for _ in range(num_activos)
10 )
11
12     def utilidad(pesos, rendimientos, aversion_riesgo):
13         r, v = calcular_estadisticas_cartera(pesos,
14         rendimientos)
15         return -(r - aversion_riesgo * v**2)
16
17     pesos_iniciales = np.array([1/num_activos] * num_activos
18 )
19     resultado = minimize(utilidad, pesos_iniciales,
```

Simulación Monte Carlo

```
1 # Simulación Monte Carlo para proyecciones futuras
2 np.random.seed(42)
3 for sim in range(num_simulaciones):
4     # Generar rendimientos aleatorios correlacionados
5     Z = np.random.multivariate_normal(media_diaria,
6                                       cov_diaria,
7                                       total_dias)
8     rendimientos_cartera = np.sum(Z * pesos_optimos, axis=1)
9
10    # Acumular capital
11    for t in range(1, total_dias):
12        resultados_simulacion[sim, t] =
resultados_simulacion[sim, t-1] * (1 +
rendimientos_cartera[t])
```

Funcionalidades básicas:

- Múltiples fuentes de datos:
 - Yahoo Finance (tiempo real)
 - Archivos CSV
 - Datos simulados
- Visualización interactiva
- Personalización de parámetros
- Restricciones personalizables

Análisis avanzados:

- Frontera eficiente
- Matriz de correlación
- Simulación Monte Carlo
- Análisis de escenarios
- Proyección de capital
- Estadísticas de probabilidad

Limitaciones del Modelo

Limitaciones

- Asume rendimientos normalmente distribuidos
- No considera eventos extremos (cisnes negros)
- Optimización sensible a errores de estimación
- No incluye costos de transacción ni impuestos
- Basado en datos históricos que pueden no reflejar el futuro

Advertencia

Esta herramienta es para fines educativos y de investigación, no constituye asesoramiento financiero. Consulte a un asesor financiero profesional antes de tomar decisiones de inversión.



Mejoras Futuras

- Rebalanceo automático
- Backtesting histórico
- Exportación de informes
- Integración con APIs de brokers
- Desarrollo de una app móvil

Conclusiones

- El modelo de Markowitz sigue siendo relevante después de 70 años
- Las herramientas modernas permiten su implementación interactiva
- Nuestra aplicación:
 - Democratiza el acceso a técnicas de optimización de carteras
 - Permite a inversores visualizar el trade-off riesgo-rendimiento
 - Proporciona una base sólida para la toma de decisiones
- Recordar siempre las limitaciones del modelo
- Complementar con análisis fundamental y consideraciones macroeconómicas

Referencias

-  Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, 7(1), 77-91.
-  Sharpe, W. F. (1964). Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk. *The Journal of Finance*, 19(3), 425-442.
-  Black, F., & Litterman, R. (1992). Global Portfolio Optimization. *Financial Analysts Journal*, 48(5), 28-43.
-  Streamlit Documentation. <https://docs.streamlit.io/>
-  Yahoo Finance API. <https://pypi.org/project/yfinance/>

¡Gracias por su atención!

¿Preguntas?

Código disponible en:

[<https://github.com/fernando-the-madness/metodos-de-optimizacion.git>]