

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

Отчет о программном проекте на тему:
Оптимизация портфеля с использованием количественных методов

Выполнил студент:

группы БПМИ2210, 2 курса Поляков Владислав

Принял руководитель проекта:

Соколов Евгений Андреевич

Соруководитель проекта:

Колесников Иван Андреевич

Содержание

1	Аннотация	3
2	Введение	4
3	Обзор литературы	4
4	Оптимизируемая функция	4
5	Алгоритм оптимизации	5
5.1	Basinhopping	5
5.2	Nevergrad	5
6	Архитектура программы	6
7	Nevergrad vs SciPy	9
8	Заключение	9
	Список литературы	10

1 Аннотация

В данной работе описывается разработка программы для оптимизации портфеля ценных бумаг. Цель проекта - создать инструмент для определения наилучшей возможной комбинации ценных бумаг инвестиционного портфеля с учетом пользовательских ограничений. Ключевые функции включают функцию создания ограничений, функцию ребалансировки портфеля, функцию расчёта метрики эффективности(коэффициент Шарпа).

Ключевые слова

Оптимизация инвестиционного портфеля, коэффициент Шарпа, риск, доходность, методы оптимизации, Nevergrad

2 Введение

Мне поступило предложение от компании реализовать программу на Python для решения оптимизационной задачи поиска весов для портфеля корпоративных и государственных облигаций на следующий период.

Проблема оптимизации портфеля инвестиций имеет большое значение в финансовой сфере, поскольку инвесторы постоянно стремятся достичь максимально возможной доходности своих инвестиций при минимальных рисках. Для решения данной задачи сперва нужно определить оптимизируемую функцию. В качестве таковой была выбрана функция(индекс) Шарпа. Затем - выбрать алгоритм оптимизации и имплементировать его. В данной работе для этого используются две библиотеки - SciPy и Nevergrad. После имплементации алгоритма он был протестирован на основе предварительно обработанных исторических данных. В конце работы проведено сравнение выбранных алгоритмов с бенчмарком.

3 Обзор литературы

Одной из самых популярных библиотек для портфельной оптимизации на Python является **PyPortfolioOpt** [2]. Библиотека обладает более широким функционалом, предоставляя больше возможностей для настройки оптимизации с финансовой точки зрения - в первую очередь, позволяя использовать другие подходы к построению портфеля помимо методики Марковица. Отличительным преимуществом данной работы по сравнению с библиотекой являются возможность настраивать все аргументы функции оптимизации, достигая необходимых пользователю точечных целей, и менять саму функцию оптимизации непосредственно.

4 Оптимизируемая функция

Основой современной портфельной теории являются работы американского экономиста Г. Марковица, согласно которым главным критерием формирования инвестиционного портфеля является соотношение его доходности и риска [1]. Исходя из этой логики, очевидным выбором для оптимизируемой функции является коэффициент Шарпа, который вычисляется как разница между доходностью инвестиций и безрисковой доходностью, делённая на стандартное отклонение доходности инвестиций.

$$S = \frac{E[R - R_f]}{\sigma} = \frac{E[R - R_f]}{\sqrt{Var[R - R_f]}}$$

где:

1 R — доходность портфеля

2 R_f — доходность от альтернативного вложения(бенчмарка)

5 Алгоритм оптимизации

5.1 Basinhopping

Для первой реализации было решено выбрать знакомую библиотеку SciPy [4] .

SciPy - это Python библиотека с открытым исходным кодом, которая опирается на возможности NumPy. Библиотека состоит из широкого спектра подпакетов для решения различных научных и инженерных задач. В частности, в подпакете `scipy.optimize` содержатся алгоритмы оптимизации. Один из базовых алгоритмов оптимизации библиотеки SciPy - метод `scipy.optimize.basinhopping()`.

Basin-hopping - это двухфазный метод, сочетающий глобальный пошаговый алгоритм с локальной минимизацией на каждом шаге. По своей сути он основан на широко известном методе отжига.

Среди достоинств данного алгоритма можно выделить тот факт, что он поддерживает добавление нелинейных ограничений и позволяет настраивать каким именно методом осуществлять локальную минимизацию.

5.2 Nevergrad

В качестве усовершенствования была выбрана оптимизационная платформа Nevergrad [3].

Nevergrad - это Python библиотека с открытым исходным кодом для решения задач непрерывной оптимизации. Nevergrad специализируется на алгоритмах оптимизации и является более гибким и эффективным инструментом по сравнению с SciPy для решения задач оптимизации. Nevergrad также предоставляет инструменты для параллельной и распределенной оптимизации, что делает его подходящим для высокопроизводительных вычислительных сред. Далее будет предоставлено сравнение работы `scipy` и Nevergrad

6 Архитектура программы

Функция вычисления коэффициента Шарпа содержится в классе *SharpRatio*, чьим полем является *bond_data* - обработанные данные для расчёта коэффициента в данный день. Для данного класса определена функция `__call__`, которая и содержит вычисление коэффициента Шарпа. Именно эта функция будет оптимизироваться.

Функция *create_constraints* принимает на вход целевой рейтинг и создаёт лист ограничений, который затем подаётся в качестве аргумента в функцию. При необходимости именно с помощью изменения этой функции можно добавлять специализированные ограничения, необходимые пользователю.

На данный момент по умолчанию стоят ограничения на веса (в сумме дают 1, каждый по отдельности не меньше нуля и меньше задаваемой верхней границы) и на средний рейтинг портфеля.

Функция *get_optimal_weights* возвращает новые веса с учетом ограничений. Именно в ней имплементированы функции из соответствующих библиотек.

Отдельно был написан код для бектеста. Для обработки и хранения данных создан класс *BondDataProvider* с полями *price*, *duration*, *static*, *rating*, *volume*, *yeild*, *rgbtr*, *rucbtr*, в каждом из которых хранятся соответствующие pandas дата-фреймы. В свою очередь, эти дата-фреймы получаются из парсинга csv таблиц - сырых данных с биржи. В качестве индекса в каждом из этих датафреймов установлен мультииндекс из **isin** (Международный идентификационный код ценной бумаги) и даты торгов.

Поле *price* отвечает за историю цен облигаций и состоит из колонок:

```
'isin','date','spread','ask','bid','ai','last','mid'
```

Поле *duration* отвечает за предсчитанную дюрацию ценных бумаг и состоит из колонок

```
'id','date','isin','emission_id','emission_emitent_id','trading_ground_id','avar_price'  
, 'convexity','convexity_offer','dur','dur_to','dur_mod','dur_mod_to'
```

Поле *rating* отвечает за рейтинг ценных бумаг и состоит из колонок:

```
'isin','update_date','fintoolid','rating','rating_cleaned','num_rating'
```

Поле *yeild* отвечает за ожидаемую доходность бумаг и состоит из колонок:

```
'isin','date','yield'
```

Поле *rgbtr* отвечает за индекс RGBTR, с которым будет сравниваться наш алгоритм на бектесте.

Оставшиеся поля содержат дополнительную информацию о бумагах, в том числе название эмитента, объем торгов и т.д., что требуется при наличии у пользователя специфических ограничений (например, игнорирование бумаг определенного эмитента).

Класс *BondPortfolio* хранит историю портфеля - историю весов по датам и веса на данный момент. Помимо этого, он также рассчитывает и хранит суммарную стоимость активов с данными весами.

С помощью него производится сравнение алгоритма с индексом государственных облигаций России(rgbtr). Результаты сравнения по метрике profit and loss(разница между всеми расходами и доходами за выбранный период) можно увидеть на Рисунке 6.1 для оптимизации с помощью SciPy и на Рисунке 6.2 для оптимизации с помощью Nevergrad.

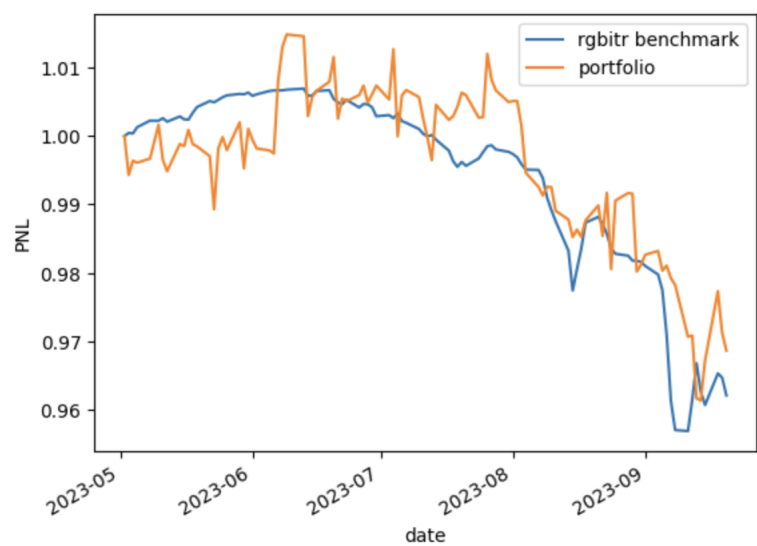


Рис. 6.1: График, изображающий PnL статистику портфеля, созданного с помощью SciPy, по сравнению с бенчмарком rgbtr

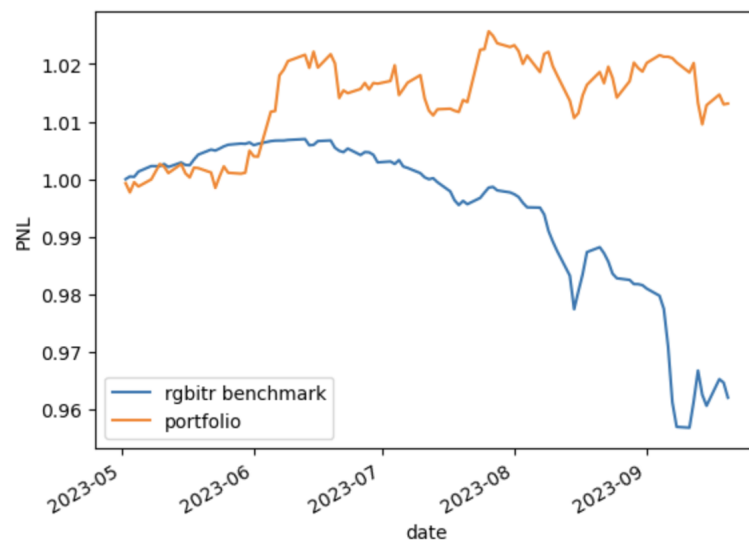


Рис. 6.2: График, изображающий PnL статистику портфеля, созданного с помощью Nevergrad, по сравнению с бенчмарком rgbtr

7 Nevergrad vs SciPy

Как можно заметить из рисунков выше, Nevergrad показывает лучший результат по сравнению с SciPy. Оптимизация с SciPy хоть и показывает результаты несколько лучше бенчмарка, в целом имеет те же тенденции, что и индекс rgibtr. В частности, на данном промежутке индекс показывает падение рынка, и портфель следует за ним. В то же время Nevergrad на коротких промежутках хоть и идёт ниже рынка, в целом показывает позитивную динамику. В частности, портфель демонстрирует рост в то время, как рынок падает.

8 Заключение

По итогам данной курсовой работы была разработана программа для оптимизации портфеля ценных бумаг. Программа определяет наилучшую возможную комбинацию ценных бумаг, позволяя накладывать на поиск весов пользовательские ограничения, в том числе нелинейные. Стратегия на рынке облигаций, созданная при помощи данной программы, показывает высокую доходность при сравнении с бенчмарком - индексом гособлигаций rgibtr. В дальнейшем, данный проект можно развивать в нескольких направлениях. Во-первых, исследовать и имплементировать другие алгоритмы оптимизации, добавить пользовательский интерфейс, добавить возможность выбора функции оптимизации - по сути, возможность выбора принципа построения стратегии.

Список литературы

- [1] G. Peter Todd Harry M. Markowitz. *Mean-Variance Analysis in Portfolio Choice and Capital Markets*. John Wiley Sons, 2000.
- [2] Robert Andrew Martin. *PyPortfolioOpt documentation*. URL: <https://pyportfolioopt.readthedocs.io/en/latest/index.html>.
- [3] Facebook AI Research. *Nevergrad documentation*. URL: <https://facebookresearch.github.io/nevergrad/index.html>.
- [4] *SciPy official website*. URL: <https://scipy.org/>.