

АВТОМАТИЗИРОВАННЫЙ АЛГОРИТМ ОПТИМИЗАЦИИ ИНВЕСТИЦИОННОГО ПОРТФЕЛЯ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON 3

© 2020 Петров Даниил Никитич

Высшая инженерно-экономическая школа

Санкт-Петербургский политехнический университет Петра Великого, Россия, Санкт-Петербург

E-mail: daniil.klev99@gmail.com

© 2020 Конников Евгений Александрович

Кандидат экономических наук, доцент, Высшая инженерно-экономическая школа

Санкт-Петербургский политехнический университет Петра Великого, Россия, Санкт-Петербург

E-mail: konnikov.evgeniy@gmail.com

В рамках существующего на данный момент высокого уровня волатильности экономики, все более актуальной становится задача формирования пассивного дохода и сохранение сбережений. Данная задача в первую очередь решается посредством инструментов инвестирования, одним из которых является портфельное инвестирование. Однако, на данный момент значимая часть населения является не в достаточной мере финансово-образованной, что приводит к значительному повышению инвестиционных рисков. В рамках данной статьи представлен алгоритм оптимизации инвестиционного портфеля, полностью автоматизированный на языке программирования Python 3. Данный алгоритм позволяет значительно снизить риски неквалифицированных инвесторов.

Ключевые слова: портфельное инвестирование, оптимизация портфеля, риск, доходность, теория Марковица.

Портфельные инвестиции — инвестирование в различные ценные бумаги с целью получения прибыли и сохранения инвестируемых средств. Инвестиционный портфель имеет отличия от инвестирования в отдельные финансовые инструменты характеристики, которые обусловлены совокупностью ценных бумаг входящих в портфель. Также стоит отметить, что в инвестиционный портфель могут входить не только акции, но и облигации, векселя и прочие финансовые инструменты. Отличительной чертой является то, что инвестор не участвует в управлении предприятием, как при прямых инвестициях. Основной задачей портфельного инвестирования является формирование портфеля с оптимальным уровнем риска и доходности. Данный процесс основан на теории портфельного инвестирования.

Теория портфельного инвестирования берет свое начало в начале XX века, когда профессор Йельского университета И. Фишер издал книгу «Теория процента». В основу данной книги лег принцип сравнения дисконтируемой разницы между выгодами и затратами каждого инвестиционного проекта. После нескольких работ Дж. М. Кейнса и И. Фишера, в 1952 году, про-

фессор Чикагского университета Г. Марковиц предлагает свою портфельную теорию. Данная теория и положила начало современному портфельному инвестированию и является базой для составления портфеля даже в наши дни. В 1964 году ученик Г. Марковица — В. Шарп разработал новую модель, которая послужила развитию следующего этапа инвестиционной теории — модель САРМ (capital asset pricing model). На сегодняшний день применяются обе модели для формирования инвестиционного портфеля. Теория Г. Марковица применяется на первом этапе составления портфеля, когда инвестор решает в какие типы финансовых инструментов вкладываться, а модель САРМ применяется на втором этапе, когда инвестор решает в какие конкретные ценные бумаги того или иного сегмента рынка инвестировать. Однако, это не значит, что обязательно необходимо использовать обе модели. Модель Марковица, также работает и при втором этапе составления портфеля инвестиций.

На ряду с теорией портфельного инвестирования развивались технологии алгоритмического трейдинга. В начале 2000-х годов у трейдеров появилась задача: «как выставить большие по объему заявки от институциональных инвесто-

ров на рынок без риска потерь». Данный вопрос и послужил основным началом «алгоритмического трейдинга», который в наши дни применяется почти каждым банком, хедж и паевыми фондами, а также и частными инвесторами. С середины 2000-х годов эту проблему удалось решить при помощи создания «алгоритмических движков», которые исполняли все те же действия, что и трейдер, автономно. Задача трейдера состояла только в том, чтобы перенаправить заявку в данный движок и выбрать алгоритм исполнения заявки. Данная технология позволила трейдером не заниматься рутинной работой и концентрироваться на исполнении более сложных заявок, а также ускорила исполнения заявок до 1–2 секунд, что значительно снизило риск «реквота». В США, в 2007 году первые торговые роботы, на основе арбитражных стратегий, пошли в «массы». В России развитие алгоритмического трейдинга происходила с опозданием от американского рынка. Оно началось только в 2006 году и началось с той же схемы, которая была описана в начале — алгоритмического «движка». Начиная с середины 2000-х годов началась активная разработка автоматизированных торговых систем — торговых роботов/советников. А с появлением языка программирования «Python» разработка роботов стала еще проще. На сегодняшний день существует огромное количество советников, которое можно разделить по определенным признакам.

В первую очередь стоит отметить «автоматизм» торгового алгоритма. В данной классификации есть всего два типа торговых алгоритмов:

- Автоматические торговые системы
- Полуавтоматические торговые системы

Исходя из названия видов, можно сказать, что первая группа алгоритмов запрограммирована на полную автоматизацию торговли, которая требует только первоначальных настроек от трейдера, таких как: объем лота, ограничения рисков, управление капиталом и т.д. Такие торговые системы не только сами находят сигналы для входа в рынок, но также и сами открывают и закрывают сделки. Ко второй группе можно отнести советников, где окончательное решение принимает сам трейдер. Такие советники только проводят технический анализ и выдают сигнал или точку для входа в рынок, но тип исполнения, объем сделки и уровни закрытия сделки выбирает сам трейдер. Такие советники можно назвать торговыми сигналами или же, если это совокуп-

ность торговых сигналов и индикаторов — торговыми стратегиями.

Вторая классификация — по принципу работы автоматической торговой системы:

- Трендовые.
- Контртрендовые.
- Мультивалютные.
- Скальпирующие.
- Комбинированные.
- Мартингейловые.

Более подробно расскажем о каждом типе торгового советника. Трендовые — это системы, в основе которых лежит поиск тренда. После нахождения тренда советник открывает по нему сделку. Тренд может быть медвежий или же бычий. Контртрендовые — принцип работы данных советников построен на осцилляторах. Данные советники работают на рынке в условиях «флэта». Мультивалютные — данные советники могут оперировать несколькими валютами сразу, однако их также можно настроить и на работу только с одной валютой. Данный советник применяется только на рынке «Форекс». Скальпирующие — исходя из названия можно понять, что данные советники работают по системе «скальпинг». Однако стоит заметить, что такие роботы являются наиболее рисковыми. Комбинированные — обычно работают сразу с несколькими стратегиями. В основе таких роботов может входить несколько советников, которые время от времени, сменяют друг друга, естественно в автоматическом режиме. Мартингейловые — в основе таких систем лежит всем известная стратегия Мартингейла, которая подразумевает удвоение объемов каждой последующей сделки. С точки зрения риск-менеджмента, данные роботы могут выглядеть не очень привлекательными, однако если трейдер оперирует большим капиталом, то он может позволить себе использовать данную систему.

Существует еще огромное множество классификаций и типов автоматизированных торговых систем, такие как: предиктивные алгоритмы, алгоритмы поведенческих факторов, алгоритмы исполнения приказов, алгоритмы оптимизации портфеля и т.д. Нашем же объектом изучения являются алгоритмы для автоматизированной оптимизации инвестиционного портфеля.

Основная задача, которую нам предстоит решить — это написание теории Г. Марковица на языке программирования Python. Python можно смело назвать довольно старым языком — он

появился в 1991 году, то есть практически 30 лет назад. За это время он постепенно собрал вокруг себя большое сообщество. У него понятный синтаксис, похожий на обычный, «человеческий» язык. Python существует так долго, что разработчики смогли специализированные специальные библиотеки, частично автоматизирующие поставленную цель. Например:

- Для многомерных массивов и высокоуровневых матриц — NumPy.
- Для расчетов в инженерном деле — SciPy.
- Для исследования, анализа и манипулирования данными — Pandas.
- Для работы с искусственным интеллектом — Scikit-Learn.

Одно из достоинств Python — его логичность и относительная простота. Он интерпретируемый, то есть исходники не нужно компилировать. Разработка на Python идёт быстрее, чем на многих других языках.

Задача, которую нам предстоит решить является, относительно, новой для Российского рынка, так как большинство современных инвестиционных приложений нацелены, на самостоятельное формирование портфеля инвесторами. Также стоит отметить, что большинство населения страны, которые используют инвестиционные приложения (например: Тинькофф инвестиции, Альфа — капитал и т.д.) не являются квалифицированными участниками рынка ценных бумаг и наше решение сможет сократить сумму убытков и количество ошибок, совершаемое неквалифицированными инвесторами.

Перед тем как приступить к написанию кода для нашего алгоритма нам нужно погрузиться в теорию оптимизации портфеля. Согласно определению московской биржи: «инвестиционный портфель — это набор биржевых инструментов, которые помогают в достижении определенных финансовых целей».

Выделяется 3 основных цели формирования инвестиционного портфеля:

- Портфель для защиты
- Портфель для дохода
- Портфель для роста

В нашем случае мы будем рассматривать портфель для дохода, так как наша основная цель — это создание систематического дохода на длительное время.

После того как мы разобрались с понятием инвестиционного портфеля, можно переходить

к теории оптимизации портфеля инвестиций. Согласно Г.Марковицу данная теория звучит следующим образом: «Вложения заданного объема инвестиционного капитала в один объект инвестиций является более рискованным, чем инвестирование той же суммы в различные объекты». В данной теории речь идет о диверсификации, которая позволяет уменьшить совокупный риск инвестиций. Получается, что портфель инвестиций будет считаться оптимальным, если, с одной стороны, за одинаковым уровнем доходности не будет существовать каких-либо иных инвестиционных возможностей с меньшим уровнем риска и с другой стороны — не будет иных, более прибыльных инвестиционных портфелей, которые характеризуются таким же уровнем риска (несистематические риски). Говоря более простым языком — это наилучший баланс между прибылью и риском.

Оптимизация портфеля по методу Марковица включает в себя следующий порядок действий:

- Вычисление ожидаемой доходности и риска каждой выбранной нами акции
- Вычисление коэффициентов корреляции для каждой пары рассматриваемых акций
- Генерация возможных вариантов портфеля для рассматриваемых акций и вычисление его характеристик
- Выбор оптимального варианта инвестиционного портфеля, основываясь на поставленные условия

Давайте рассмотрим каждый этап более подробно. Расчет доходности и риска акции производится по следующим формулам:

$$r = \frac{D + (P_1 - P_0)}{P_0} * 100\% \quad (1)$$

Где: «D» — размер дивиденда за год; «P1» — цена продажи акции; «P0» — цена покупки акции.

$$\beta = \frac{Cov(r_i, r_m)}{\sigma_m^2} \quad (2)$$

Где: «β» — коэффициент меры систематического риска; «r_i» — доходность i-й акции; «r_m» — рыночная доходность; «σ_m²» — дисперсия рыночной доходности.

Теперь, когда мы знаем основную теорию оптимизации, мы можем перейти к основам написания кода. Нашу работу можно разделить на

несколько частей.

Первая — загрузка библиотеки «*ufinance*» для сбора котировок. На этом этапе мы применяем описанную выше библиотеку, которая позволяет нам выгрузить цены закрытия из сервиса «*Yahoo Finance*». Важно отметить, что мы будем выгружать цены закрытия активов за предыдущий период.

Вторая — формирование дата-фрейма на основе выгруженных данных. На этом этапе мы формируем дата-фрейм, который будет иметь вид таблицы с интересующими нас данными. Также на основе этого дата-фрейма нам надо будет сформировать динамику изменения цен для будущего анализа. Для этого нам надо загрузить еще одну библиотеку, предусмотренную Python — «*matplotlib.pyplot*». Данная библиотека поможет нам в формировании графиков динамики изменения цен каждого актива.

Третья — прогнозирование цен на будущий период (период 1 неделя). В этой части нам надо использовать еще одну библиотеку «*Statsmodels*». Данная библиотека очень полезна, когда мы работаем с множеством статистических моделей, в том числе и с моделями временных рядов. Данные модели нам понадобятся, так как мы изменяем привычный алгоритм оптимизации инвестиционного портфеля по Марковицу. Вместо расчета доходности за предыдущие периоды, мы будем использовать модели временных рядов (АРИМА, САРИМА, ARMA, Аддитивная) для расчетов доходности на будущий период, что улучшит точность оптимизации портфеля и снизит риски. На этом этапе мы будем применять все модели, перечисленные выше к каждому активу, чтобы понять какая модель временных рядов наиболее точная в прогнозировании будущих цен. Здесь надо учитывать, что для разных активов будет своя наилучшая модель.

Четвертая — формирование нового дата-фрейма после прогнозирования цен активов. На этом этапе мы формируем новый дата-фрейм, который состоит из активов с рассчитанной будущей доходностью.

Пятая — расчет рисков. Как нам уже известно оптимизация портфеля — это наилучшее соотношение доходности и риска именно для этого нам надо сформировать ковариационную матрицу для расчетов риска портфеля. И также после расчетов формируется дата-фрейм в виде самой матрицы.

Шестая — формирование облака портфелей. После того, как мы сформировали все дата-фреймы мы можем сгенерировать значительное число инвестиционных портфелей — облако портфелей. Из полученного облака мы можем видеть наиболее подходящие портфели, для определенных нужд инвестора.

Седьмая — автоматическая переоптимизация портфеля. В третьей части мы указали, что период, на который мы прогнозируем цены равен 1 недели. Из этого следует, что мы формируем портфель на одну неделю. В начале торговой недели у нас есть уже сформированный портфель, далее наш алгоритм в течение всей недели высчитывает наилучшие возможности соотношения риска и доходности. В конце торговой недели наш алгоритм, автоматически, переоптимизирует существующий портфель из расчета новых цен на будущую неделю. Данным действием мы понижаем риски, связанные с инвестированием, и повышаем доходность.

Схематично алгоритм представлен на рисунке 1.

В заключение можно сказать, что с развитием современных технологий, а также экономики, депозиты в банке и инвестиции в облигации постепенно уходят на второй план. Люди хотят инвестировать свои накопления более эффективно, однако большая часть населения России не являются квалифицированными участниками рынка ценных бумаг и понимают данную сферу очень поверхностно. Описанное решение, может послужить эффективным инструментом инвестирования для этого процента населения и в добавок может положить основу новой модели хедж-фондов, основанных на технологии робо-эдвайзинга.

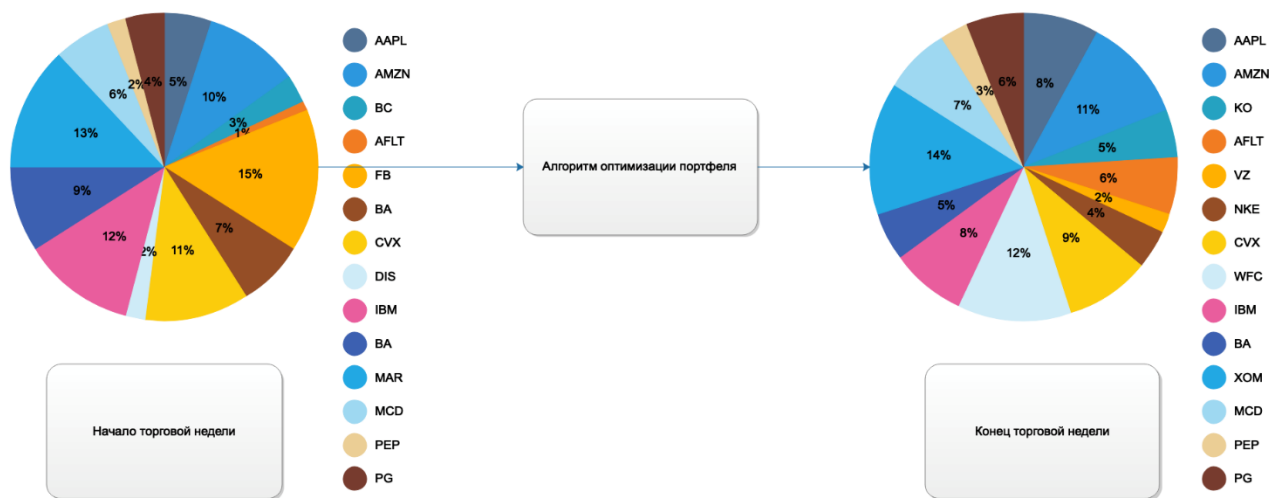
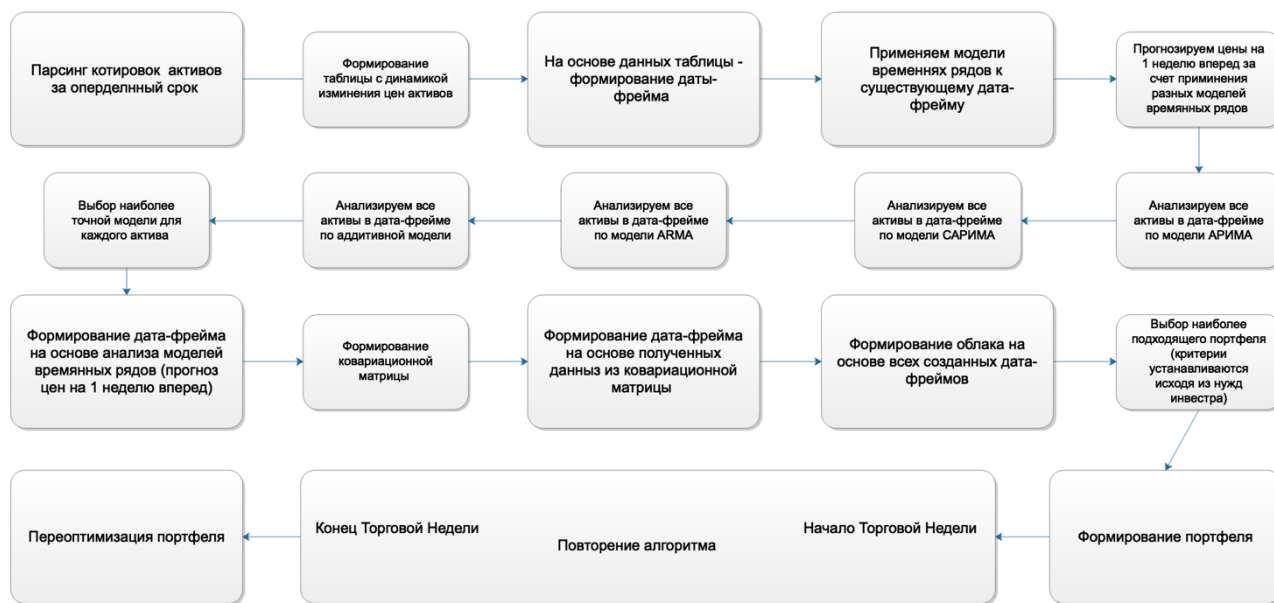


Рисунок 1. Автоматизированный алгоритм оптимизации инвестиционного портфеля

Библиографический список

1. Буренин А. Н. Рынки производных финансовых документов. -М.: ИНФРА-М, 2006. —325 с.
2. Галанов В. А. Производные инструменты срочного рынка: фьючерсы, опционы, свопы. Учебник. -М.: Финансы и статистика, 2005. —464 с.
3. Галиц Л. Финансовая инженерия: инструменты и способы управления финансовым риском. -М.: ТВП, 2006. —104 с.
4. Селищев А. С., Маховикова Г. А. Рынок ценных бумаг -М., Юрайт, 2012—128 с.
5. Ширяев В. И. Финансовые рынки. Стохастические модели, опционы, форварды, фьючерсы. -М.: Либроком, 2009. —224 с.