

Комментарии к моему решению 1-2:

В моей модели реализована неизвестность того, кто начнёт игру. В некоторых играх (не знаю как по поводу настольного тенниса), игрок у которого будет первая подача изначально не определен. В моём коде это определяет монетка. При желании эту особенность можно выключить: в реализации функции **monte_carlo** изменить значение переменной *serve* с `np.random.randint` на `1`, тоже самое в методе **descending** класса **Tree**, а в Цепи Маркова просто убрать создание 2-ого объекта в области кода **Main**.

Проработан момент с неточностью арифметических операций в python. Для этого использован модуль `decimal`.

```
>>> from decimal import Decimal
>>> 1-0.7
0.30000000000000004
>>> float(1-Decimal(f"0.7"))
0.3
```

Моя реализация дерева в задаче 1, относится к *плохой* реализации, о которой будет говориться в дальнейшем. Сделал я дерево таким образом только для экономии времени, так как суть задания было просто реализовать метод прохода по дереву который выдаёт распределение конечных результатов.

3) Опиши сравнительные плюсы и минусы методов А, Б, В для обоих пунктов.

Задание 1

А) Метод Монте-Карло.

Данный метод оценки предсказательной модели является самым простым из предоставленных на это задание. Его реализация занимает не много времени, при этом даёт достаточно эффективный результат.

В некоторых сложных задачах, является основным методом вычисления (многомерные интегралы, квантовая физика). Суть алгоритма в проходе по модели большого количества раз. По сути, он основывается на законе больших чисел, и для получения точных результатов, требуется провести большое количество вычислений, что в сложных моделях может занять много времени.

Б) Проход возможных исходов по дереву

По сути, метод Б является методом А. Мы создаем дерево, и после этого начинаем симулировать стохастический процесс и собирать результаты. То есть основная разница между этими методами — это создание дерева возможных исходов. Структура модели в форме дерева позволяет нам легко собирать статистические данные с его вершин(узлов). То есть, когда основная суть метода А, получить конечные результаты финальных счетов, то метод Б очень легко позволяет нам просмотреть количество проходов по интересующих нас вершинам (узлам) дерева. Важный момент – тоже самое можно реализовать и в обычном методе А (если немного видоизменить вывод в самом алгоритме), однако преимущество именно метода Б, это высокая читабельность кода, и простота просмотра статистики разных вершин.

В) Поиск стационарного распределения, используя матрицу переходов Марковского процесса

Данный метод является наилучшим для нашей предсказательной модели. Он позволяет получить точное вероятностное распределение счетов в матче. То есть данный метод полностью выполняет функции методов А и Б, и более того, делает это математически точно.

Используя матрицу переходов, и начальный вектор, мы можем найти стационарное распределение вероятностей, и более того, распределение вероятностей на любом из шагов

($c_{(i+t)} = c_i \times P^t$, где P – матрица переходных вероятностей). Данный функционал и “математическая точность”, даёт методу В уверенное первое место для данной предсказательной модели.

Задание 2

В общем и целом, все описанное в пункте 1, подходит и ко второму заданию. Единственное что, я опишу ключевые моменты в реализации данных методов во втором задании.

В данном задании основная “фишка” в том, что наша модель циклична. Мы выходим с этого цикла только после получения одним из игроков преимущества в 2 очка. Как раз по причине этой цикличности, в данном задании в методе Б у нас идет проход не по дереву, а по графу, так как:

Дерево — одна из наиболее широко распространённых [структур данных](#) в [информатике](#), эмулирующая [древовидную структуру](#) в виде набора связанных узлов. Является связным [графом](#), **не содержащим циклы**.

Реализация матрицы переходных вероятностей, по сути, такая-же как и реализация графа.

И напоследок, я опишу отдельно плюсы и минусы методов А, Б, и В.

Метод Монте-Карло:

Плюсы:

- Простота реализации для простых моделей

Минусы:

- Требуются не плохие вычислительные мощности

Дерево:

Плюсы:

- Высокая читабельность

- Вероятность на любом из шагов

Минусы:

- Построение дерева не быстрый процесс

- Требуются не плохие вычислительные мощности

Марковская цепь:

Плюсы:

- Точность

- Читабельность

- Вероятность на любом из шагов

Минусы:

- Не реализуем для сложных моделей

4) Допустим, мы делаем аналогичную оценку для игры Counter Strike.

Одна из главных особенностей – в случае взятия раунда, команда набирает преимущество в экономике, что может помочь во взятии следующего раунда. Опиши применимость вышеуказанных методов для такого случая.

В данной задаче оценка методом Монте-Карло только немного усложняется. По сути, в алгоритм добавится одна переменная, в которую нужно будет записывать результат прошлого раунда, и относительно значения этой переменной, будут отличаться вероятности победы/проигрыша команды.

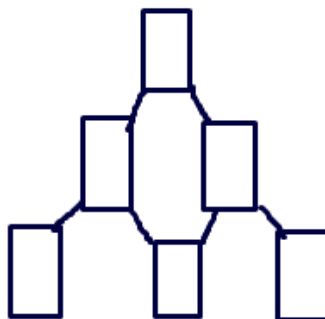
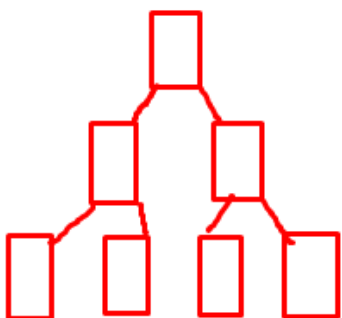
Что по поводу прохода по дереву, ситуация зависит исключительно от реализации данного метода. В плохом из вариантов, дерево просто увеличивается в размерах. Каждая ветка порождает собственное дерево возможных исходов, в котором вероятности зависят от результата предыдущего раунда, что очень плохо сказывается на времени работы данного метода. В другом варианте, при проходе по дереву у нас просто добавляется переменная, в которой храниться результат предыдущего раунда (как в методе Монте-Карло). В моей реализации данного метода как раз просто добавится одна переменная при спуске.

И последний метод, это матрица переходов Марковского процесса. Основное свойство простой Марковской цепи - при фиксированном настоящем будущее независимо от прошлого. То есть:

$$\mathbb{P}(X_{n+1} = i_{n+1} \mid X_n = i_n, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = \mathbb{P}(X_{n+1} = i_{n+1} \mid X_n = i_n).$$

Исходя из данного определения, использование данного метода не подходит для нашей задачи. Данный метод оценки является аналитическим, поэтому ему требуется чёткое соблюдение условий, **однако**, данный метод всё же можно реализовать для этой задачи.

Реализация данного метода в условиях этой задачи, по сути, является “плохим вариантом дерева”, о котором я говорил ранее (каждая ветка будет порождать своё дерево). Чтобы соблюсти основное свойство Марковской цепи нам придётся увеличить количество возможных состояний цепи. То есть:



Где чёрным пример графа (в данном случае дерева) к заданию 1, а красным к заданию 4.