Анализ данных.
Лекции. Неделя 3.
Симуляция и бутстраппинг.

Филиппов Максим

14 октября 2022 г.

Содержание

1 Симуляция Монте-Карло

1.1 Основные шаги в симуляции Монте-Карло

- **Определите алгоритм генерации данных**
  Обратите внимание, что данные генерируются с использованием случайных чисел, что влияет на итоговую выбоку.
- **Оцените неизвестную переменную**
  Используя выборку, сгенерированную на первом этапе, оцените целевую переменную.
- **Повторите процесс N раз**
  Итоговая оценка переменной - среднее N измерений.

1.2 Генерация случайных чисел

Конгруэнтный линейный генератор это базовый алгоритм генерации случайных чисел. Он использует параметры a, b, m и сид y_0, и определен формулой $y_{i+1} = (ay_i + b) \mod m$. a, b, m выбираются таким образом, чтобы итоговая последовательность удовлетворяла свойствам случайной, а первые k элемкентов послеовательности обычно откидываются. На практике, испольщуются более сложыне генераторы, но они более "тяжелые"вычислительно.

2 Ошибка сэмплирования Монте-Карло и способы ее понизить

2.1 Ошибка сэмплирования Монте-Карло

Точность симуляции может быть оценена стандартной ошибкой истинного значения $s/\sqrt{N}$, где s - стандартное отклонение, а N - количество попыток. Чтобы увеличить точность оценки в 10 раз, требуется взять в 10^2 раз больше измерений, что очень тяжело вычислительно. Лучше пытаться уменьшить s.

2.2 Антитетические переменные

Пусть u_t равномерно распределена на [0,1]. Пусть g - плотность распределения. Тогда $g(u_t)$ это множество первоначально выбранных значения, а $g(1 - u_t)$ - множество антитетичесих значений. Эту идею можно использовать чтобы увеличить количество сгенерированных случайных переменных вдвое при фиксированном N.

$Corr(u_t, 1 - u_t) = -1 \Rightarrow Corr(g(u_t), g(1 - u_t)) < 0 \Rightarrow Error = \frac{s\sqrt{1+\rho}}{\sqrt{N}}$, $\rho = Corr(g(u_t), g(1 - u_t))$

2.3 Контрольная переменная

Пусть x - неизвестная целевая переменная. Пусть y - переменная, похожая на x, но с известными свойствами. Пусть $\hat{x}$ и $\hat{y}$ - оценки x и y соответственно. Тогда x можно оценить точнее как $x^* = \hat{x}+(y-\hat{y})$, но улучшение по сравнению с обычным методом Монте-Карло будет достигнуто только при условии $Var(\hat{y}) < 2Cov(\hat{x}, \hat{y})$. Это неравенство можно вывести из $Var(x^*) = Var(\hat{x}+(y-\hat{y})) = Var(\hat{x})+Var(\hat{y})-2Cov(\hat{x}, \hat{y})$

2.4 Переиспользование наборов случайных чисел

Привлекательным кажется переиспользование уже сгенерированных случайных чисел для генерации объектов. Однако, это не позволяет уменьшить ошибку оценок по сравнению с моделированием с меньшим N, но без переиспользования тех же случайных чисел. Этот метод может быть использовать для повышения воспроизводимости результатов. Кроме того, это используется в бутстраппинге.

3 Метод бутстраппинга

3.1 Основные шаги метода бутстраппинга

Основная идея метода бутстраппинга это выбор подвыборки N раз и подсчет целевой переменной как среднее N оценок на подвыборках, причем все подвыборки одного размера и объекты выбираются с повторением. Ключевое отличие от метода Монте-Карло это тот факт, что используемые данные исторические, а не сгенерированные.

3.2 Проблемы и ограничения

Метод бутстраппинга строится на предположение о том что все подвыборки имеют одинаковое распределение, что может быть неверно. Выбросы в выборке могут привсти к неоходиомсти выбирать очень большое N для устранения смещения оценок. Если распределение меняется значительно на разных частях выборки, то подвыборки могут быть смещены, а информация о краткосрочных изменениях будет теряться в процессе бутстраппинга. В случае автокорреляции, подвыборки стоит выбирать последовательно плавающим окном.

4 Преимущества и недостатки симуляционных подходов

4.1 Недостатки

- Высокая вычислительная стоимость
- Неточность оценок
- Низкая воспроизводимость результатов
- Результаты сильно зависят от предпосылок

4.2 Преимущества

- Возможность моделирования ситуаций возможных лишь в теории, но для описания которых не существует исторических данных.