СПбГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

А. Ю. Филатов

Методы обработки данных

Фрагмент конспекта

СПбГЭТУ «ЛЭТИ», 2022 г.

1 Метод наименьших квадратов

1.1 Введение в метод наименьших квадратов?

Решаемая задача

Дан некоторый набор точек на плоскости. Хотим построить функцию, которая аппроксимирует заданный набор точек (см. рис. 2).

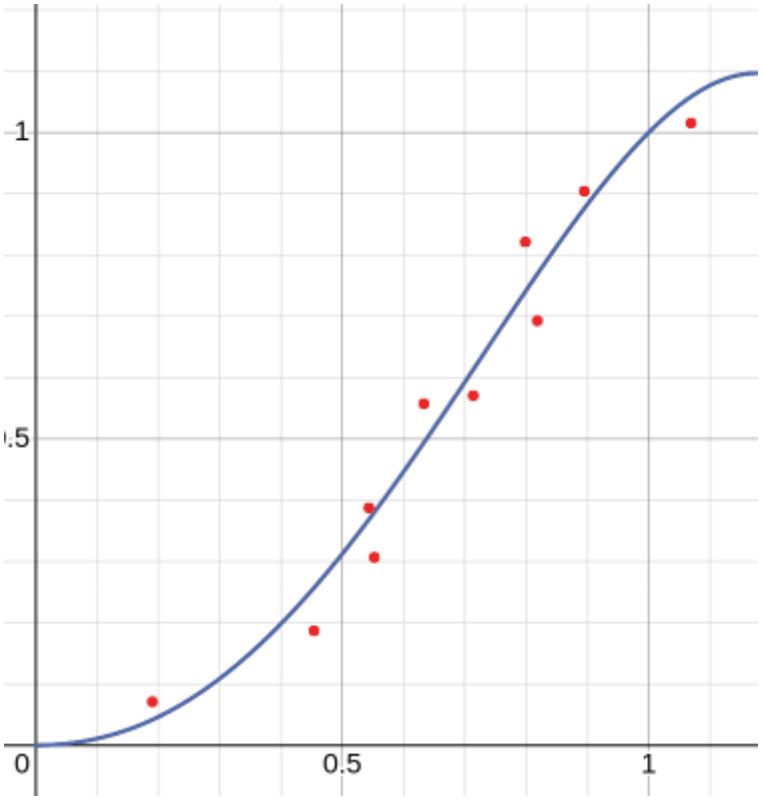

Рис. 2

Основная идея метода наименьших квадратов(МНК) заключается в поиске прямых, у которых сумма квадратов расстояний до заданных точек наименьшая.

МНК описание общего решения:

Пусть аппроксимирующая функция будет представлена в виде:

$$y = f(x, b_0, b_1, \ldots, b_n),$$

где f - известная функция(многочлен), а $b_0, b_1, \ldots, b_n$ - неизвестные параметры функции.

Для решения задачи необходимо найти такие значения параметров функции, чтобы значения исследуемой функции в точках x_i примерно совпадали со значениями аппроксимирующей функции.

Пусть $\varepsilon_i = f(x_i, b_0, b_1, \ldots, b_n) - y_i, i = 1, 2, \ldots, m$ - ошибка (отклонение) значений исходной и исследуемой функций в точке x_i.

Сумма квадратов ε_i будет величиной, показывающей меру отклонения f от известных значений y_i, обозначим её как S. Чтобы найти аппроксимирующую функцию, надо минимизировать это значение:

$$S = \sum_{i=1}^{m} \varepsilon_i^2 = \sum_{i=1}^{m} (f(x_i) - y_i)^2 \to min$$

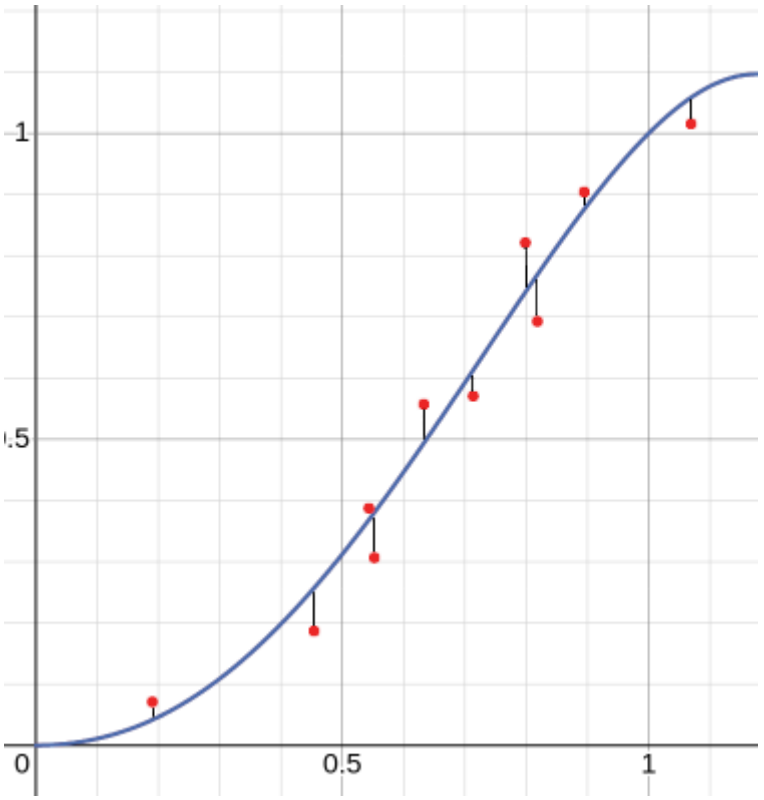

Рис. 3

Для этого потребуется найти его производную и приравнять к нулю. Величина S зависит от переменных функции $b_0, b_1, \ldots, b_n$, они независимы, поэтому для оценки производной мы можем оценить частные производные по этим переменным:

$$\frac{\partial S}{\partial b_0} = 0; \frac{\partial S}{\partial b_1} = 0; \ \ldots; \ \frac{\partial S}{\partial b_n} = 0$$

Из полученных соотношений можно составить систему уравнений для определения значений b_0, b_1, …, b_n .

МНК пример матричной реализации с линейным случаем

Пусть даны (x_i, y_i) - координаты точек из заданного набора $y = ax + b$ - прямая, которую мы хотим найти. Нам нужно минимизировать сумму квадратов расстояний, n - количество точек.

$$\sum_{i=1}^{m} (f(x_i) - y_i)^2 \to min$$

В идеальном случае должна быть верна система уравнений:

$$\begin{cases} ax_1 + b = y_1 \\ ax_2 + b = y_2 \\ ax_3 + b = y_3 \\ \ldots \\ ax_n + b = y_n \end{cases}$$

Запишем систему в матричном виде:

$$A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \ldots & \\ x_n & 1 \end{bmatrix}$$

$$x = \begin{bmatrix} a \\ b \end{bmatrix}$$

$$b = \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{bmatrix}$$

Тогда $Ax = b$, но на самом деле $Ax = b + e$, где e - ошибка(погрешность). Таким образом нам нужно минимизировать погрешности, а точнее их квадраты, так как e - вектор, то при скалярном произведении получается число, которое нужно минимизировать.

$$e = Ax - b$$

$$e^T e = (Ax - b)^T (Ax - b)$$

Нужно оценить производную:

$$\frac{d\varepsilon^T \varepsilon}{dx} = 0$$

Преобразуем

$$e^T e = (Ax - b)^T (Ax - b) = ((Ax)^T - b^T)(Ax - b) = (A^T x^T - b^T)(Ax - b) =$$
$$x^T A^T Ax - x^T A^T b - b^T Ax + b^T b = x^T A^T Ax - 2x^T A^T b + b^T b$$

Применяем матричное дифференцирование

$$\frac{d\varepsilon^T \varepsilon}{dx} = \left(A^T A + \left(A^T A\right)^T\right)x - 2A^T b = \left(A^T A + A^T A\right)x - 2A^T b = 2A^T Ax - 2A^T b$$

$$\frac{d\varepsilon^T \varepsilon}{dx} = 0$$

$$2A^T Ax - 2A^T b = 0$$

$$A^T Ax = A^T b$$

$$x = \left(A^T A\right)^{-1} A^T b$$

Это решение X и есть решение по МНК.

Список источников

Коломиец Л.В., Поникарова Н.Ю. Метод наименьших квадратов [Электронный ресурс]: метод. указания / сост.: Л.В. Коломиец, Н.Ю. Поникарова. – Самара: Изд-во Самарского университета, 2017. – 32 с. URL: http://repo.ssau.ru/bitstream/Metodicheskie-izdaniya/Metod-naimenshih-kvadratov-Elektronnyi-resurs-metod-ukazaniya-73397/1/%D0%9A%D0%BE%D0%BB%D0%BE%D0%BC%D0%B8%D0%B5%D1%86%20%D0%9B.%D0%92.%20%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%20%D0%BD%D0%B0%D0%B8%D0%BC%D0%B5%D0%BD%D1%8C%D1%88%D0%B8%D1%85.pdf (дата обращения: 02.05.2022)

Малышева Т.А. Численные методы и компьютерное моделирование [Электронный ресурс]. Лабораторный практикум по аппроксимации функций:

Учеб.-метод. пособие. СПб.: Университет ИТМО, 2016. 33 с. URL: https://books.ifmo.ru/file/pdf/1953.pdf (дата обращения: 02.05.2022)

Методы интерполяции и аппроксимации [Электронный ресурс]. URL: https://portal.tpu.ru/SHARED/m/MBB/uchebnaya_rabota/Model/Tab/Interp_app.pdf (дата обращения: 02.05.2022)

2 Фильтр Калмана

2.1 Определение

Фильтр Калмана — рекурсивный фильтр, оценивающий вектор состояния динамической системы, используя ряд неполных и зашумленных измерений. Основан на фильтре Байеса, который принимается на вход:

- Текущее измерение;
- Априорная вероятность оцениваемого параметра
- Функция оценки гипотезы

А на выходе дает гипотезу, оптимизирующую функцию оценки.

Основная задача фильтра Калмана заключается в том, чтобы из данных полученных, например, с сенсоров получить отфильтрованное текущее состояние от различных шумов. То есть на вход принимается:

- Текущее измерение Z_i
- Отфильтрованное предыдущее состояние $\widehat{X_{i-1}}$;
- Степень доверия к предыдущему состоянию $\widehat{P_{i-1}}$;
- Предположение, каким должно быть текущее состояние $\overline{X_i}$;
- Предположение, каким должно быть доверие к текущему состоянию $\overline{P_i}$;

По итогу своей работы фильтр Калмана возвращает:

- Отфильтрованное текущее состояние $\widehat{X_i}$;
- Степень доверия к предыдущему состоянию $\widehat{P_i}$;

Состоянием называется вектор, который описывает рассматриваемые характеристики объекта. То есть это набор параметров объектов, которые наиболее важны в рассматриваемой задаче. Например, описание положение объекта в пространстве:

$$X = (x\ y\ z)^T$$

Измерением называется вектор (не обязательно той же размерности), описывающий те же характеристики или некоторую функцию от них с

некоторым шумом. Измерением можно называть набор характеристик, которые мы можем наблюдать.

Графический пример работы можно посмотреть на рис. 1

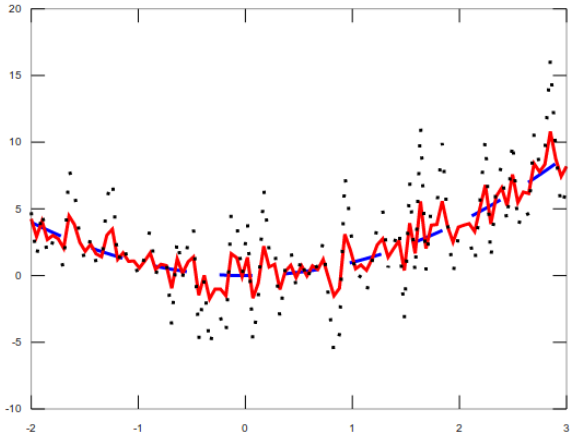

Рис. 1

2.2 Модель в фильтре Калмана

Общий вид модели фильтра Калмана выглядит:

$$X^{*} = f(X, U, W),$$

где:

1. X – это вектор из модели изменения состояния;
2. U – вектор внешнего воздействия;
3. W – вектор шума в модели.

Линейная модель фильтра Калмана:

$$X_{i+1} = FX_i + GU_i + \Gamma W_i,$$

где F, G, Г – подобранные матрицы.

Список источников

1. Цыплаков, А. (2011) Введение в моделирование в пространстве состояний. — Квантиль, № 9, стр. 1—24.

3 Фильтр частиц

3.1 Область применения

Фильтр частиц представляет собой набор алгоритмов, применяемых для решения задач фильтрации, требуемых при при обработке сигналов и байесовском статистическом выводе. Задача фильтрации состоит в оценке внутренних состояний в динамических системах при частичных наблюдениях и наличии случайных возмущений как в датчиках, так и в динамической системе. Целью фильтрации является вычисление апостериорного распределения состояний некоторого марковского процесса с учетом зашумленных и частичных наблюдений.

3.2 Основная идея

Идея фильтра частиц состоит в генерации множества гипотез о состоянии системы, их фильтрации на основе поступающих данных, а затем выборе наиболее вероятных из них. Каждая частица представляет собой одну гипотезу о состоянии системы.

3.3 Основные этапы алгоритма

Основной цикл фильтрации разделен на этапы:
Движение (Motion update)
Предсказание (Prediction)
Обновление (Measurement update)
Отсев (Resampling)

Движение (Motion update)

На этом этапе происходит движение робота с использованием его модели движения. Так как измерение его положения происходит с погрешностями, то робот теряет информацию о своем местоположении.

Предсказание (Prediction)

На этом этапе мы предсказываем новые состояния частиц в соответствии с моделью движения робота.

Обновление (Measurement update)

На этом этапе робот получает новую информацию о своем местоположении путем измерений. Происходит пересчет весов частиц, а затем нормализация, чтобы их сумма равнялась 1.

Отсев (Resampling)

Обновление прогноза фильтра частиц происходит статистическим образом. Образцы из раздачи представлены набором частиц, каждая из которых имеет свой вес правдоподобия, который представлен вероятностью, что эта частица будет выбрана из функции плотности вероятности. Соответственно, чем выше вес частицы, тем больше вероятность, что она переживет отсев. Чтобы избежать коллапса весов, вызванного большим различием их значений, частицы с малым весом заменяются новыми, приближенными к ним с большим весом.

Результирующее распределение, в котором каждая частица будет иметь свою вероятность того, что именно она является достоверной гипотезой о состоянии робота, будет иметь следующий вид:

$$p(x_t|d_{o..t}) = \eta\, p(z_t \mid x_t) \int p(x_t \mid u_{t-1}, x_{t-1})\, p(x_{t-1} \mid d_{o..t-1}) dx_{t-1}$$

, где

$p(x_t|d_{o..t})$ - функция плотности вероятности текущего состояния, апостериорная вероятность p(x). Вычисляется на этапе отсева частиц (Resampling)

$\eta\, p(z_t \mid x_t)$ - веса важности w(x). Вычисляются на этапе обновления весов с использованием константы нормализации η (Measurement update)

$\int p(x_t \mid u_{t-1}, x_{t-1})\, p(x_{t-1} \mid d_{o..t-1}) dx_{t-1}$ - априорная вероятность q(x). Вычисляется на этапе предсказания нового состояния частиц (Prediction) с использованием $p(x_{t-1} \mid d_{o..t-1})$ функции плотности вероятности предыдущего состояния и модели движения.

Таким образом, вероятностное распределение частиц на каждом шаге приобретает вид:

$$p(x) = w(x)\, q(x)$$

3.4 Псевдокод

```
particles = initialize particles(N_particles)
time_step = 0

while (execution() == true) do
 time_step += time_delta

        move = get robot movement(time_step)
        particles = get new particles from timesteps(particles, move + noise)
        weights = compute new particles weights(particles)
        particles = resampling particles (particles, weights)

        end while
```

3.5 Список источников

Particle Filters and Their Applications [Электронный ресурс] // Kaijen Hsiao, Henry de Plinval-Salgues, Jason Miller, Cognitive Robotics. [April 11, 2005]. URL: https://web.mit.edu/16.412j/www/html/Advanced%20lectures/Slides/Hsaio_plinval_miller_ParticleFiltersPrint.pdf (дата обращения: 02.05.2022)

A Tutorial on Particle Filtering and Smoothing: Fifteen years later [Электронный ресурс] // Arnaud Doucet, Adam M. Johansen. [December 2008]. URL: https://www.cs.ubc.ca/~arnaud/doucet_johansen_tutorialPF.pdf (дата обращения: 02.05.2022)

4 Байесовский фильтр спама

4.1 Байесовская фильтрация спама?

Байесовская фильтрация спама — метод для фильтрации спама, основанный на применении наивного байесовского классификатора, в основе которого лежит применение теоремы Байеса. Данный фильтр широко применяется в почтовых сервисах.

Формула для подсчета вероятности, что слово спам:

$$\Pr(S|W) = \frac{\Pr(W|S) \cdot \Pr(S)}{\Pr(W)} = \frac{\Pr(W|S) \cdot \Pr(S)}{\Pr(W|S) \cdot \Pr(S) + \Pr(W|H) \cdot \Pr(H)},$$

где

Pr(S|W) – условная вероятность того, что сообщение спам, при условии, что слово W находится в тексте S;

Pr(S) – полная вероятность того, что произвольное сообщение спам;

Pr(W|S) – условная вероятность того, что слово W появляется в сообщениях, если они являются спамом;

Pr(H) – полная вероятность того, что произвольное сообщение не спам;

Pr(W|H) – условная вероятность, что слово W появляется в сообщениях, если они не являются спамом.

Перед тем как использовать байесовский фильтр спама его надо обучить, то есть для того, чтобы уметь вычислять “спамность” слова, нам необходимо знать частоту попадания слова в предложение, которое является спамом, то есть:

$$Pr(W_i \mid S) = \frac{\text{count}(M : W_i \in M, M \in S)}{\sum_j \text{count}(M : W_j \in M, M \in S)}$$

Аналогично вычисляется и P(W|H) (условная вероятность, что слово W появляется в сообщениях, если они не являются спамом):

$$Pr(W_i \mid H) = \frac{\text{count}(M : W_i \in M, M \in H)}{\sum_j \text{count}(M : W_j \in M, M \in H)}$$

Недостатки

Недостатки фильтра спама:

- Работает только с текстом
- Основывается на вероятности слов быть спамом. Не реагирует на спам, если он состоит из слов, у которых маленькая вероятность быть спамом

Список источников

1. Metsis V., Androutsopoulos I., Paliouras G. Spam filtering with naive bayes-which naive bayes? //CEAS. – 2006. – Т. 17. – С. 28-69.