версия от 2 ноября 2022 г. 16:00

# Численные методы, осень 2022

## Задание 3 [Проекторы. Задача наименьших квадратов. QR-разложение.]

## Всего баллов: 47 Срок сдачи (после переноса): 11 ноября

### РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА

- Лекции 6–8, 10–11 из [1]
- Лекция 8 из [2]

### УПРАЖНЕНИЯ

1. (3) Даны матрицы:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

   - Получите ортогональные проекторы на пространства $\mathrm{colsp}(A)$ и $\mathrm{colsp}(B)$.
   - Вычислите (вручную) QR-разложения матриц $A$ and $B$.

2. (5) Частица единичной массы в начальный момент времени покоится в точке $x = 0$. Затем она подвергается воздействию кусочно-постоянной внешней силы $f_i$ $(i = 1 \dots 10)$. Пусть $a = (x,\, v)\big|_{t=10}$ — вектор, состоящий из координаты и скорости частицы в конечный момент времени. Найдите матрицу $A$ такую, что $a = Af$. Заметьте, что она будет иметь размер $2 \times 10$. С помощью SVD-разложения численно найдите вектор $f$ наименьшей евклидовой нормы, при котором $a = (1,\, 0)$.

3. (5) Создайте датасет следующим образом: выберите $n = 7$ точек $x_i$ из равномерного распределения на отрезке $[0,\, 6]$. Затем вычислите $y_i = f(x_i) + \epsilon_i$, где $f(x) = 10\sin(x)$, а $\epsilon_i$ — независимые стандартные нормальные случайные величины. Изобразите на одном графике точки датасета и функцию $f$.

   Аппроксимируйте датасет с помощью линейной $l(x) = w_0 + w_1 x$ и кубической $c(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$ функций. Изобразите на одном графике получившиеся функции и исходный датасет.

4. (7) Загрузите файл с матрицами $A$ и $C$ (изображение и фильтр). Откройте его:

```
with np.load('data.npz') as data:
    A, C = data['A'], data['C']
```

Матрицу $A$ полезно преобразовать в вектор $a$:

```
def mat2vec(A):
    A = np.flipud(A)
    a = np.reshape(A, np.prod(A.shape))
    return a
```

Обратное преобразование из вектора $a$ в матрицу $A$:

```
def vec2mat(a, shape):
    A = np.reshape(a, shape)
    A = np.flipud(A)
    return A
```

Изображение, хранящееся в матрице $A$, было получено из исходного изображения $A_0$ с помощью свёртки с фильтром $C$ и добавления шума. Записывая матрицы как векторы описанным выше способом, мы можем написать

$$a_0 \to a = Ca_0 + \epsilon,$$

где $\epsilon$ — вектор из независимых одинаково распределённых гауссовских случайных величин. Фильтр $C$ размывает изображение, увеличивая при этом его размер с $16 \times 51$ до $25 \times 60$. Вашей задачей будет восстановить исходное изображение $A_0$ по размытому изображению $A$ и фильтру $C$.

- Постройте изображение $A$.
- Исследуйте, как фильтр $C$ меняет изображения.
- Наивный способ восстановить $A_0$ — это найти $a_0$ из системы $a = Ca_0$. Является ли эта система недо- или переопределённой? Вычислите $a_0$ с помощью сингулярного разложения матрицы $C$ и постройте получившееся изображение.
- Чтобы улучшить результат, попробуйте использовать только часть сингулярных чисел матрицы $C$. Выберите наиболее удачное количество.

5. (7) Рассмотрим задачу оптимизации

$$\min_x \left\|Ax - b\right\|_2 \quad \text{при условии} \quad Cx = \vec{0}$$

Предполагая, что матрица $A^T A$ обратима, воспользуйтесь методом множителей Лагранжа и получите выражение для точки минимума $x$.

6. (20) Эта задача посвящена локализации точек на плоскости. Рассмотрим $n$ точек с *неточными* координатами: $A_i = (x_i,\, y_i)$. Измерим $m$ углов между некоторыми из них: $\theta_{ijk} = \angle\big(\overrightarrow{A_iA_j},\ \overrightarrow{A_iA_k}\big)$. Наша цель — используя результаты измерений, уточнить координаты точек.

   В качестве примера рассмотрим $n = 3$ точки с приближенными координатами: $A_1 = (-1,\, 0),\ A_2 = (0,\, 1),\ A_3 = (1,\, 0)$ — и один измеренный угол $\theta_{123} = 9\pi/40$. Так как наши оценки координат не согласуются с измеренным углом, мы должны их скорректировать:

$$x_i \to x_i + dx_i, \quad y_i \to y_i + dy_i,$$

   где $dx_i$ и $dy_i$ находятся из условия

$$\overrightarrow{A_1A_2} \cdot \overrightarrow{A_1A_3} = |A_1A_2| \cdot |A_1A_3| \cdot \cos(\theta_{123})$$

   Это условие может быть линеаризовано в предположении, что поправки окажутся малыми. При таком подходе мы конструируем $m = 1$ уравнений для $2n = 6$ переменных. Система обычно оказывается недоопределённой. Но мы можем рассматривать её в смысле метода наименьших квадратов, т.е. пытаться определить наименьшие поправки к координатам, которые делали бы их согласованными с измерениями. В рассмотренном примере можно найти решение

$$dr_1 = (dx_1,\, dy_1) = (-h,\, 0), \quad dr_2 = (h,\, -h), \quad dr_3 = (0,\, h), \qquad \text{где } h \approx \pi/80 \approx 0.04$$

   Ваша задача — написать код, который принимает текущие оценки координат точек $(n \times 2)$ и измеренные углы $\theta_{ijk}$ ($m$ значений и $m \times 3$ индексов). Возвращать он должен найденные поправки $(n \times 2)$.

   Чтобы протестировать свой код, вы можете использовать пример из этого упражнения, а также другой пример с бóльшим числом точек. Загрузить данные можно следующим образом:

```
with np.load('./data_1.npz') as data:
    r = data['r']          # исходные точки
```

```
p = data['p'] - 1      # индексы измеренных углов
theta = data['theta'] # углы в радианах
dr = data['dr']        # предполагаемый ответ
```


[1] L. N. Trefethen and D. Bau III, *Numerical linear algebra*, Vol. 50 (Siam, 1997).

[2] E. E. Tyrtyshnikov, *A brief introduction to numerical analysis* (Springer Science & Business Media, 2012).