Численные методы, Весна 2021 ВШЭ. Задание 2.[a]

Задачи типа А. (автопроверка)

1. **(15) [LU]** Реализуйте LU разложение квадратной матрицы с выбором главного элемента. Следуйте инструкциям в classroom.

Задачи типа Б.

1. **(7)** Докажите следующие неравенства и приведите примеры вектора x и матрицы A, при которых эти неравенства насыщаются:

   - $\|x\|_2 \leq \sqrt{m}\|x\|_\infty$
   - $\|A\|_\infty \leq \sqrt{n}\|A\|_2$

   где x – вектор длины m и A – матрица размера $m \times n$.

2. **(10)** Постройте руками SVD разложение следующих матриц:

$$(a) \quad \begin{bmatrix} 3 & 0 \\ 0 & -2 \end{bmatrix}, \quad (b) \quad \begin{bmatrix} 0 & 2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad (c) \quad \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

3. **(10)** Напишите код на Python, который для данной действительной матрицы A графически изображает правые сингулярные вектора $v_1,\ v_2$ (вписанные в окружность) и левые сингулярные вектора $u_1,\ u_2$ (вписанные в эллипс) – аналогично Fig. 4.1 Trefethen, Bau. Используйте этот код для матриц из Задачи 2.

4. **(15)** Рассмотрите матрицы X размером $n \times m$, Ω размером $m \times m$ и Δ размером $n \times n$. Пусть

$$f(A) = A^{-1}X(X^TA^{-1}X)^{-1}.$$

   Докажите, что

$$f(X\Omega X^T + \Delta) = f(\Delta)$$

   предполагая, что все матрицы, которые обращаются в этом уравнении, действительно являются обратимыми.

5. **(15)** Ознакомьтесь с Woodbury matrix identity, справедливом для матриц подходящих размеров:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U\left(C^{-1} + VA^{-1}U\right)^{-1}VA^{-1}. \tag{1}$$

   Рассмотрите частный случай диагональной ($p \times p$) матрицы A и единичной ($k \times k$) матрицы C и напишите функцию woodbury(A, U, V) вычисляющую $(A + UV)^{-1}$ по формуле (1). Проверьте, что Ваша имплементация верна, сравнивая результат с полученным прямолинейным вычислением. Сравните быстродействие этих двух способов: какой оказывается быстрее и почему? (рассмотрите случайные матрицы с $p = 5000, k = 100$).

6. **ODR (15)** Скачайте файл, содержащий координаты облака точек в трехмерном пространстве, и распакуйте его с помощью numpy:

```
with np.load('data_distance_svd.npz') as data:
    xp, yp, zp = data['xp'], data['yp'], data['zp']
```

[a] Дополнительно указаны: (количество баллов за задачу)[имя задачи на nbgrader]

Массивы содержат координаты точек: первая точка имеет координаты `(xp[0], yp[0], zp[0])`, вторая точка — координаты `(xp[1], yp[1], zp[1])` и т.д. Нарисуйте точки на трехмерном графике. Можно заметить, что точки группируются вблизи некоторой плоскости. Задача состоит в том, чтобы построить данную плоскость. В качестве критерия выберем следующий: назовем наилучшей такую плоскость, что сумма квадратов *ортогональных* расстояний от точек до этой плоскости минимальна. Изобразите найденную плоскость на графике с облаком точек. Прокомментируйте полученные результаты.

Указание. Пусть плоскость с вектором нормали $\vec{n}$ проходит через точку с координатами $\vec{c}$. Тогда расстояние от точки $\vec{p}$ до плоскости дается скалярным произведением $\langle(\vec{p}-\vec{c})\cdot\vec{n}\rangle$. Таким образом, задача поиска наилучшей плоскости сводится к задаче наименьших квадратов относительно единичного вектора нормали $\vec{n}$.

7. **Прокрустово преобразование (20)**

См. напр. гл. 14[1]. Цель задания – максимально хорошо “подогнать” левую фигуру под правую трансляцией в плоскости и поворотом, см. рис 1.

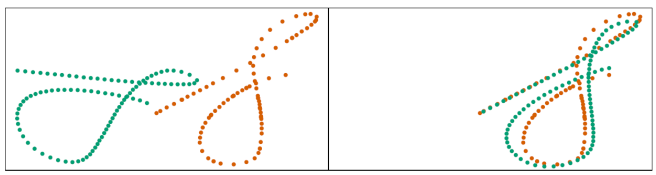

Рис. 1.

Пусть $\mathbf{X}_1$ и $\mathbf{X}_2$ – $n\times 2$ матрицы, содержащие n x,y-координат точек левой и правой фигур соответственно. Задача сводится к минимизации “прокрустова расстояния” между фигурами $\min_{\vec{\mu},\mathbf{R}}||\mathbf{X}_2-(\mathbf{X}_1\mathbf{R}-\mathbf{1}\vec{\mu}^T)||_F$, где $||\mathbf{X}||_F=\mathrm{tr}(\mathbf{X}^T\mathbf{X})$ – норма Фробениуса, $\mathbf{R}$ – 2×2 ортогональная матрица поворота, $\vec{\mu}$ – двумерный вектор трансляции в плоскости, $\mathbf{1}$ – столбец из n единиц.

Пусть $\bar{x}_1$, $\bar{x}_2$ – двумерные вектора, содержащие среднее по каждому из двух столбцов матриц $\mathbf{X}_1$ и $\mathbf{X}_2$. Центрируем $\mathbf{X}_1$ и $\mathbf{X}_2$, т.е. вычтем из них среднее по столбцам, $\tilde{\mathbf{X}}_1\equiv\mathbf{X}_1-\mathbf{1}\bar{x}_1^T$, $\tilde{\mathbf{X}}_2\equiv\mathbf{X}_2-\mathbf{1}\bar{x}_2^T$. Сделаем SVD разложение матрицы $\tilde{\mathbf{X}}_1^T\tilde{\mathbf{X}}_2=\mathbf{U}\mathbf{D}\mathbf{V}^T$, тогда решение поставленной задачи дается формулами:

$$\vec{\mu}=\bar{x}_2-\mathbf{R}^T\bar{x}_1,\quad \mathbf{R}=\mathbf{U}\mathbf{V}^T. \tag{2}$$

Задание:

(a) (5 баллов) Воспользовавшись формулами (2) и файлом с данными[2], получить правую картинку на рис. 1.

(b) (15 баллов) Получить формулы (2). Для этого:

i. Расписать целевую функцию, пользуясь инвариантностью следа при циклических перестановках матриц и ортогональностью матрицы поворота $\mathbf{R}^T\mathbf{R}=\mathbf{R}\mathbf{R}^T=I$.

ii. Формально продифференцировать получившееся выражение по $\vec{\mu}^T$, при этом считая нетранспонированный вектор $\vec{\mu}$ константой (т.е. независимой переменной). Приравнивая получившееся выражение нулю, получить первое из равенств (2).

iii. Единственное слагаемое, содержащее неизвестную $\mathbf{R}$ в целевой функции, должно иметь вид $\mathrm{tr}(\mathbf{S}^T\mathbf{R})$, $\mathbf{S}\equiv\tilde{\mathbf{X}}_1^T\tilde{\mathbf{X}}_2$. Представить матрицу вращений в виде $\mathbf{R}=\begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}$, а $\mathbf{S}=\begin{pmatrix}s_{11} & s_{12}\\ s_{21} & s_{22}\end{pmatrix}$, продифференцировать получившееся выражение по θ и приравнять его нулю.

iv. Убедиться проверкой, что последнее равенство эквивалентно матричному уравнению $\mathbf{S}^T\mathbf{R}=\mathbf{R}^T\mathbf{S}$. Пользуясь этим равенством и условием ортогональности $\mathbf{R}$, получить второе равенство в (2).

---

[1] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition, Springer Series in Statistics, Springer New York, 2009.
URL `https://books.google.com/books?id=tVIjmNS3Ob8C`

[2] первые две колонки в этом файле – координаты x, y первой фигуры, 3-я и 4-я колонки – координаты второй фигуры; прочитать файл можно, например, командой `np.loadtxt('signatureData2.csv', delimiter=',')`.