Numerical Methods: Lecture 3. Projectors. Least squares problem. QR factorization.

Konstantin Tikhonov

October 19, 2022

1 Suggested Reading

- Lectures 6-8, 10-11 of [1]
- Lecture 8 of [2]

2 Exercises

Deadline: 4 Nov

1. (3) Consider the matrices:

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

   - Derive orthogonal projectors on range(A) and range(B).
   - Derive (on a piece of paper) QR decomposition of matrices A and B.

2. (5) Consider a particle of unit mass, which is prepared at $t = 0$ at $x = 0$ at rest $v = 0$. The particle is exposed to piece–wise constant external force f_i at $i - 1 < t \leq i$, with $i = 1, 2, ..., 10$. Let $a = (x(t = 10), v(t = 10))$ be a vector composed of coordinate and velocity of a particle at $t = 10$. Derive the matrix A such that $a = Af$ (note that A is of a shape 2×10). Using (a numerical) SVD decomposition, evaluate f of minimal Euclidean norm such that $a = (1, 0)$.

3. (5) Consider the function $f(x) = 10\sin(x)$. Generate a dataset D that will consist of $n = 7$ points drawn as follows. For each point randomly draw x_i uniformly in $[0, 6]$ and define $y_i = f(x_i) + \epsilon_i$, where ϵ_i are iid standard gaussian random numbers. Generate a sample dataset from this distribution, plot it together with the true function $f(x)$. Fit a linear $l(x) = w_0 + w_1 x$ and a cubic $c(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$ models to D. Plot those models together with the dataset D.

4. (7) Download the file with matrices A and C (an image and a filter). Open it as follows:

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

It is convenient to order the matrix A into a column vector a:

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

with inverse transform, from vector a to matrix A given by

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

The image, stored in the matrix A is obtained from certain original image A_0 via convoluting it with the filter C and adding some noise. The filter C blurs an image, simultaneously increasing its size from 16×51 to 25×60. With the use of associated vectors a and a_0, one may write

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

where ϵ is a vector of iid Gaussian random numbers. Your task will be to recover an original image A_0, being supplied by the image A and the filter C.

- Plot the image A.
- Explore how the filter C acts on images.
- A naive way to recover A_0 from A would be to solve $a = Ca_0$ for a_0. Is this system under– or over–determined? Using SVD of the filter matrix C, evaluate a_0 and plot the corresponding A_0.
- In order to improve the result, consider keeping certain fraction of singular values of C. Choose a value delivering the best recovery quality.

5. (7) Consider the problem

$$\text{minimize} \quad \|Ax - b\|_2 \quad \text{subject to} \quad Cx = 0 \quad \text{with respect to} \quad x,$$

where A and C are matrices and x and b are vectors.

Using the method of Lagrange multipliers, and assuming $A^T A$ to be invertible, derive explicit expression for optimal x.

6. (20) Here we will consider the problem of localization of points in a 2D plane. Consider n points, for which we have the *approximate* locations $r_i = (x_i, y_i)$. We measure k angles between certain points: $\theta_{ijk} = \angle(r_k - r_i, r_j - r_i)$. Our goal is to use the results of the measurements to improve the estimation of the locations r_i.

   To be specific, consider $n = 3$ points, for which we have approximate locations $r_1 = (-1, 0)$, $r_2 = (0, 1)$, $r_3 = (1, 0)$ and $k = 1$ measurement $\theta_{123} = 9\pi/40$. Clearly, our estimates $r_{1,2,3}$ are not consistent with the measured angle and we have to adjust the estimate: $r_i \to \bar{r}_i = r_i + dr_i$ where dr_i should be found from the condition

$$(\bar{r}_3 - \bar{r}_1) \cdot (\bar{r}_3 - \bar{r}_1) = |\bar{r}_3 - \bar{r}_1||\bar{r}_3 - \bar{r}_1| \cos\theta_{123},$$

   which can be linearized in dr_i, assuming this correction will end up small. In this approach, one constructs a single (in general, k) equation for six (in general, $2n$) variables, so the system will typically by underdetermined. We can consider this system in the least squares sense, which amounts to determining the smallest correction to all r_i which makes the updated locations consistent with observations. In the particular numerical example above, one may find $dr_1 = (-h, 0)$, $dr_2 = (h, -h)$, $dr_3 = (0, h)$ where $h = \pi/80 \approx 0.04$.

   Your task is to write the code, which will accept the current estimate of the positions r_i ($n \times 2$, float) and measurement results θ_{ijk}, which are specified by i) indices of points ($k \times 3$, int) and ii) angles (k, float); and will output the derived correction to the point positions dr_i ($n \times 2$, float). You can use the numerical example in this exercise to test your code.

References

[1] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*. Vol. 50. Siam, 1997.

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