

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

Konstantin Tikhonov

October 11, 2021

1 Suggested Reading

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

2 Exercises

Deadline: 22 Oct

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 $\text{range}(A)$ and $\text{range}(B)$.
 - 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 $i = 1, 2, \dots, 10$. 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 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 $l(x) = w_0 + w_1x$ and a cubic $c(x) = w_0 + w_1x + w_2x^2 + w_3x^3$ 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):
    h, w = A.shape
    a = np.zeros(h*w, dtype=A.dtype)
    A = np.flipud(A)
    for i, row in enumerate(A):
        a[i*w:i*w+w] = row
    return a
```

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

```
def vec2mat(a, shape):
    h, w = shape
    A = np.zeros(shape, dtype=a.dtype)
    for i in range(h):
        A[i, :] = a[i*w:i*w+w]
    return np.flipud(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

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

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

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

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

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.