ELEC70037
Topics in Large Dimensional Data Processing
Exercise

Dr Wei Dai

October 21, 2024

EEE Department, Imperial College London

Contents

Part I.

Teaching Organisaion

Teaching Organisation 1.

1.1. Overview

- Lectures
  - Mondays 07/10 - 25/11/2024, 14:00-15:50, Room **305**
  - Mondays 02/12 - 09/12/2024, 14:00-15:50, Room **508A&B**
- Office hours
  - Mondays 07/10 - 02/12/2024, 12:00-12:55, Room 503.
- Communications: Should you have any questions or inquiries:
  - Attend the designated office hours.
  - Post them on the Q&A channel.
  - Refrain from using my college email.
- Assessment
  4 coursework assignments: each counts 25%

1.1.1. Invited Lectures

To enrich students' learning experience with diverse perspectives, particularly in the areas of optimization and machine learning, the lectures on Monday, 9th December (the final Monday of this term) will feature invited seminar lectures presented by guest colleagues. These sessions will offer insights into advanced topics:

- Advanced Sparsity and Learning Approach for Imaging
  Dr Chen Qin,
  14:00-14:50, 09/12/2024, Room 508A&B
- Introduction to Distributed Optimisation and Learning
  Dr Stefan Vlaski,
  15:00-15:50, 09/12/2024, Room 508A&B

1.2. Coursework Assignments

1.2.1. Schedule

Coursework Overview:
Individual Data + Team Work + Individual Assessment

The procedure for the coursework assignments is outlined below.

- Each student will receive a unique data file provided by the GTAs.
- Students collaborate in groups while working on the coursework. The group information is input in an excel file.
- By the due time, each student must submit their own individual coursework, clearly specifying their contributions to the group effort. Note that submissions cannot be altered after the deadline.

Table 1.1.: Coursework Schedule

	Tasks	Due Date/Time
Coursework 1	Assignment	(Week 3) 14/10
	Due	(Week 4) 25/10 11:59
	Feedback	(Week 6) 04/11
Coursework 2	Assignment	(Week 5) 28/10
	Due	(Week 6) 08/11 11:59
	Feedback	(Week 8) 18/11
Coursework 3	Assignment	(Week 7) 11/11
	Due	(Week 8) 22/11 11:59
	Feedback	(Week 10) 02/12
Coursework 4	Assignment	(Week 9) 25/11
	Due	(Week 10) 06/12
	Feedback (Marks only)	(Week 11) 13/12

- GTAs will assess individual submissions. Let $M_{i,0}$ denote the raw mark received by the student i. An adjusted mark, calculated using Equation (1.1), will then be provided to the student.
- Some coursework questions will be selected for in-depth discussion. Students chosen to lead these discussions will receive bonus marks to encourage active participation.

1.2.2. Marking

The marking formula for the first three coursework is given by

$$M_i = \min\left(\left(\sum_{k\in\mathcal{G}_l} M_{k,0}\right) * C_i * W_l, M_{i,0}\right), \tag{1.1}$$

where

- M_i is the adjusted mark received by the student i,
- $M_{k,0}$ is the raw mark obtained by the student k,
- $\sum_{k\in\mathcal{G}_l} M_{k,0}$ is the total raw mark obtained by the group $\mathcal{G}_l$,
- C_i is the contribution of the student i in the group,[1]
- W_l is the weighting coefficient for the group $\mathcal{G}_l$, depending on the size of the group,
- and we apply the min function to ensure the adjusted mark does not exceeds the raw mark. [2]

1: Note that

$$\sum_{k\in\mathcal{G}_l} C_k = 1.00 = 100\%.$$

2: It is typical that $M_i = M_{i,0} \times W_l$, i.e., the adjusted mark is simply the raw mark multiplied by the weighting coefficient.

The weighting coefficients for groups are detailed in Table 1.2. There are several considerations behind the design.

- We encourage both individual work and team work within the group.
- The default size of the group is 2 or 3.
- Students from the same group may share the same codes.
- **Cheating offences and plagiarism** are taken very seriously and are dealt with according to the College's Academic Misconduct Policy and Procedure.

Size of the group	1	2	3	4	5	≥ 6
Weighting coefficient W	1.00	0.97	0.94	0.86	0.70	0.50

Table 1.2.: The weighting coefficient W for the group is decided by the size of the group.

Remark 1.2.1 (Bonus marks) GTAs will select students to present their solutions to specific questions in-depth. Students can earn bonus marks by sharing and explaining their coursework solutions during the feedback/discussion sessions.
The bonus marks awarded will be determined by the GTAs and the module leader, based on the quality of the solutions and the clarity of the explanation. The awarded marks will be communicated to the participating student.

1.3. Software for Coursework

The coursework requires Julia programming. The coursework submission files include a Jupyter notebook file and a data file in JLD2 format. We recommend VSCode as the default editor.

The software that you need to install/have

- Jupyter Notebook
  Download and install the package management software Anaconda. By default, it will install Python and Jupyter Notebook to your system.
- Julia programming language
  - Download and install the current stable version of Julia.
  - Check whether Julia has been installed properly: Run Julia interactive session (a.k.a. REPL) by following the instructions.
- VSCode
  - Download and install VSCode.
  - We need to install some extensions for VSCode. See here for how to install VSCode extensions.
    - Anaconda should have automatically installed VSCode Extensions `Python` and `Pylance` for you. If not, install them.
    - Install extension `Julia` for Julia programming in VSCode. See here and here for more details.

1.3.1. Julia for Programming

- A collection of tutorials.
- Introduction to Scientific Programming and Machine Learning with Julia.
- A tutorial in pdf format.

Your Feedback Matters | 2.

2.1. You Said, We Did

Over the past few years, we've listened carefully to your feedback and implemented substantial changes to enhance your learning experience.

2.1.1. Lecture Notes

- **Transition to Book Format**: We've transformed our lecture notes from slides into a comprehensive book format. This allows for a more cohesive and self-contained learning experience.
- **LaTeX Template**: We've adopted the LaTeX template `kaobook`, which gives you more space for note-taking.
- **Increased Examples**: We've incorporated more examples and illustrations to reinforce key concepts and make the material more engaging.

2.1.2. Assessments

- **Shift to Coursework**: We've transitioned from traditional paper-based exams to coursework assignments. This approach allows us to focus more on the practical application of theoretical knowledge.
- **Skill Development**: Coursework assignments are designed to help you hone your analytical and programming skills.
- **Peer Marking Removed**: We've removed peer marking to ensure that your performance is accurately reflected in your final grade.
- **Individual Recognition**: Our coursework marking process is designed to fairly assess and appreciate your unique contributions and efforts.

2.2. Recent Changes

Based on your feedback from the last academic year, we've implemented the following changes:

- **Reduced Coursework Load**: We've reduced the overall workload of coursework assignments by removing a learn-and-tell presentation coursework and spreading the questions across four assignments instead of three.
- **Coursework submission deadlines**. Coursework deadlines are now on Fridays, providing you with a more relaxed weekend and allowing the department to efficiently handle extension requests.
- **Enhanced Feedback**: We've increased the amount of information provided with your coursework marks. In addition to your individual scores, we'll share statistical information about the class marks.

- **In-Lecture Problem Discussion**: To foster deeper understanding and active engagement, we've incorporated in-lecture problem discussions and solving sessions.

2.3. Your Feedback

Please take the time to complete the **MEQ** questionnaires at the end of the term. Your continued feedback is invaluable in helping us improve the learning experience for this module.

2.4. Our Commitment to You

- **Accessibility and Support**: We're dedicated to providing you with the support you need to succeed. We offer regular **office hours** (Mondays, 12:00-12:55, Room 503) where you can discuss any questions or difficulties you may encounter. Additionally, we maintain a **Q&A channel** on Teams for more immediate assistance.
- **Timely Feedback**: We understand the importance of timely feedback. We aim to complete coursework marking within 10 working days of the submission deadline. You'll receive your marked assignments along with statistical information about the class performance.
- **Engaging Learning Experiences**: We're committed to creating an engaging and supportive learning environment. **In-lecture problem discussions and solving sessions** provide opportunities for you to actively participate, ask questions, and deepen your understanding of the material.

Important Notes | 3.

3.1. Plagiarism

PLAGIARISM/COLLUSION DECLARATION

Coursework submitted for assessment must be the original work of you and your group. Assignments are subjected to regular checks for plagiarism and/or collusion. Plagiarism is the presentation of another person's thoughts or words (those outside your group) as if they were your own. Collusion involves obtaining help from someone outside your group to complete your work. In preparing your coursework, you should not seek help, or copy from any other person or source, including the Internet, without proper and explicit acknowledgement.

There is a procedure in place for you to declare individual contributions within your group for courswork. You must declare the contributions fairly and accurately.

You must not disclose your solutions or insights related to coursework with anyone else, including future students or the Internet.

By acknowledging the the statements above, you are declaring that both this and all subsequent pieces of coursework are, and will remain, the original work of you and your group.

- Submissions will not be accepted without the aforementioned declaration.
- Members of a group are deemed to have collective responsibility for the integrity for work submitted and are liable for any penalty imposed, proportionate to their contributions.

3.2. Notes for Coursework Submission

- Each registered student will get a data file. The data in the data file can be unique.
- Each registered student needs to submit the solutions related to their own data, no matter whether they are in groups or not.

3.2.1. Handling Solutions

In our coursework, we use the following convention.

- If the solution to a question is a unique integer, you need to assign an integer value to your solution variable.
- If the solution to a question does not exist, you need to create a 1-D array of length zero.
- If the solution to a question is not unique, you need to create a 1-D array, of which the length is the number of distinct solutions, and then specify the values in the **ascending** order.

Examples:

```
    x = 3;
    y = Array{Int64,1}(undef,0);
    z = Array{Int64,1}(undef,3);
    z[1] = 3; z[2] = 4; z[3] = 5;
```

Part II.

Coursework 1

Linear Algebra: Basics | 4.

4.1. Trace and Inner Product of Matrices

Give a sqaure matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, its trace is defined as the sum of the diagonal elements:

$$\operatorname{trace}(\boldsymbol{A}) := \sum_{i=1}^{n} A_{i,i}.$$

Use the parameters in your data file to answer the following questions.

1. Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m \times n}$.
   a) Identify the dimension of $\boldsymbol{A}\boldsymbol{B}^{\mathsf{T}}$. [2]
   b) Identify the dimension of $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{B}$. [2]
   c) Is the following statement true or false?

   $$\operatorname{trace}\left(\boldsymbol{A}\boldsymbol{B}^{\mathsf{T}}\right) = \operatorname{trace}\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{B}\right).$$

   [1]
2. For any two matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m \times n}$, define the **inner product** between them:

   $$\langle \boldsymbol{A}, \boldsymbol{B} \rangle := \operatorname{trace}\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{B}\right).$$

   Is the following statement true or false?

   $$\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \langle \operatorname{vect}(\boldsymbol{A}), \operatorname{vect}(\boldsymbol{B}) \rangle.$$

   [1]
3. Let $\boldsymbol{A} \in \mathbb{R}^{k \times l}$ and $\boldsymbol{B} \in \mathbb{R}^{m \times n}$.
   a) Specify the dimensions of $\boldsymbol{U}$ and $\boldsymbol{V}$ such that

   $$\left\langle \boldsymbol{U}\boldsymbol{A}\boldsymbol{V}^{\mathsf{T}}, \boldsymbol{B} \right\rangle$$

   is well-defined. [4]
   b) Is the following statement true or false?

   $$\left\langle \boldsymbol{U}\boldsymbol{A}\boldsymbol{V}^{\mathsf{T}}, \boldsymbol{B} \right\rangle = \left\langle \boldsymbol{A}, \boldsymbol{U}^{\mathsf{T}}\boldsymbol{B}\boldsymbol{V} \right\rangle.$$

   [1]

4.2. Adjoint Operator

Let $\mathscr{A}$ be a linear operator mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$. It always can be represented by a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, that is,

$$\begin{aligned} \mathscr{A} :\ & \mathbb{R}^n \to \mathbb{R}^m \\ & \boldsymbol{x} \mapsto \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}. \end{aligned}$$

Definition 4.2.1 *The adjoint of the linear operator $\mathscr{A}$, denoted by $\mathscr{A}^*$, is defined via*

$$\langle \mathscr{A}(\boldsymbol{x}), \boldsymbol{y} \rangle = \langle \boldsymbol{x}, \mathscr{A}^*(\boldsymbol{y}) \rangle$$

for all $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$.

It can be verified that

$$\mathscr{A}^*(\boldsymbol{y}) = \boldsymbol{A}^{\mathsf{T}} \boldsymbol{y}.$$

If $\mathscr{A} : \mathbb{C}^n \to \mathbb{C}^m$, then $\mathscr{A}$ is generally represented by a matrix $\boldsymbol{A} \in \mathbb{C}^{m \times n}$. That is,

$$\mathscr{A}^*(\boldsymbol{y}) = \boldsymbol{A}^{\mathsf{H}} \boldsymbol{y},$$

where the superscript H denotes 'conjugate transpose'.

The following exercise questions involve identification of linear operators and their adjoint operators. One way to check the correctness of your adjoint operator is to see whether the equality

$$\langle \mathscr{A}(\boldsymbol{x}), \mathscr{A}(\boldsymbol{y}) \rangle = \langle \boldsymbol{x}, \mathscr{A}^* \left(\mathscr{A}(\boldsymbol{y}) \right) \rangle$$

holds.

1. Consider the subsampling operator $\mathscr{P}_\Omega$ where $\Omega = \{i_1, i_2, \cdots, i_m\}$ is a subset of $[1:n] := \{1, 2, \cdots, n\}$ ($m \leq n$), defined as

$$\begin{aligned} \mathscr{P}_\Omega : \mathbb{C}^n &\to \mathbb{C}^m \\ \boldsymbol{x} &\mapsto \boldsymbol{y} \text{ with } y_i = x_{i_l}. \end{aligned}$$

   a) Write a function to find the corresponding matrix representations of $\mathscr{P}_\Omega$ and $\mathscr{P}_\Omega^*$.
   Use the data in the data file for a test. [4]
   b) Write a function to evaluate $\boldsymbol{y} = \mathscr{P}_\Omega(\boldsymbol{x})$ without using the matrix representation of the linear map.
   Use the data in the data file for a test. [2]
   c) Write a function to evaluate $\boldsymbol{z} = \mathscr{P}_\Omega^*(\boldsymbol{y})$ without using the matrix representation of the linear map.
   Use the data in the data file for a test. [2]

2. Consider the Hankel operator given by

$$\begin{aligned} \mathscr{H}_{p,q} : \mathbb{C}^{p+q-1} &\to \mathbb{C}^{p \times q} \\ \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{p+q-1} \end{bmatrix} &\mapsto \boldsymbol{H} = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_q \\ x_2 & x_3 & x_4 & \cdots & x_{q+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_p & x_{p+1} & x_{p+2} & \cdots & x_{p+q-1} \end{bmatrix}. \end{aligned}$$

   Or equivalently,

$$H_{m,n} = x_{m+n-1}.$$

   a) Write a function to find the corresponding matrix representations of $\mathscr{H}_{p,q}$ and $\mathscr{H}_{p,q}^*$.

Use the data in the data file for a test. [4]

b) Write a function to evaluate $\boldsymbol{H} = \mathscr{H}_{p,q}(\boldsymbol{x})$ without using the matrix representation of the linear map.
Use the data in the data file for a test. [2]

c) Write a function to evaluate $\boldsymbol{z} = \mathscr{H}^{*}_{p,q}(\boldsymbol{H})$ without using the matrix representation of the linear map.
Use the data in the data file for a test. [2]

d) Evaluate $\boldsymbol{z}./\boldsymbol{x}$ where the symbol ./ denotes element division. [1]

Least Squares | 5.

5.1. The Netflix Problem

1. The task is to recover an unknown matrix $\boldsymbol{X}$ from its partial observations given by

$$\boldsymbol{y} = \mathcal{P}_\Omega(\boldsymbol{X}) + \boldsymbol{w},$$

where $\boldsymbol{w}$ denotes noise.
The alternating minimization method to solve the Netflix problem is stated below. Suppose the prior information that the rank of $\boldsymbol{X}$ is at most r. Then one would find the matrices $\boldsymbol{U} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times r}$ such that $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^\mathsf{T}$. In the k-th iteration of the alternating minimization, one solves the following two least squares sequentially:

$$\begin{aligned} \boldsymbol{V}^{k+1} &= \arg\min_{\boldsymbol{V}} \left\| \boldsymbol{y} - \mathcal{P}_\Omega\left(\boldsymbol{U}^k\boldsymbol{V}^\mathsf{T}\right)\right\|_2^2, \\ \boldsymbol{U}^{k+1} &= \arg\min_{\boldsymbol{U}} \left\| \boldsymbol{y} - \mathcal{P}_\Omega\left(\boldsymbol{U}\boldsymbol{V}^{k+1,\mathsf{T}}\right)\right\|_2^2. \end{aligned}$$

The partial observations $\boldsymbol{y}$ and the index sets of the observed entries Ω are given in the data file. Suppose that $\text{rank}(\boldsymbol{X}) = 10$.

a) Write a function to update $\boldsymbol{V}$ for a given $\boldsymbol{U}$. Use the $\boldsymbol{U}_0$ in the data file for a test. [2]
b) Write a function to update $\boldsymbol{U}$ for a given $\boldsymbol{V}$. Use the $\boldsymbol{V}$ from the previous sub-question for a test. [2]
c) Write a function for the alternating miminimization process. Use the $\boldsymbol{U}_0$ in the data file as the initial point. Run a test. [2]

MRI01 - Least Squares 6.

6.1. Linear Operator

This question requires background in

- 2D Discrete Fourier Transform. See Wikipedia page https://en.wikipedia.org/wiki/Discrete_Fourier_transform, Sections 'Definition', 'Properties > The unitary DFT', and 'Multidimensional DFT'.
- `fftshift`: See https://www.matecdev.com/posts/julia-fft.html#fftshift-and-fftfreq.

Let $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ be an MRI image. The MRI data compressed sensing can be mathematically expressed as

$$\boldsymbol{y} = \underbrace{\mathcal{P}_{\Omega} \circ \mathcal{S} \circ \mathcal{F}_{2\mathrm{D}}}_{\mathcal{A}}(\boldsymbol{X})$$

where

- $\mathcal{F}_{2\mathrm{D}}$ denotes the 2D Discrete Fourier Transform,
- $\mathcal{S}$ denotes the 2D `fftshift` operator (which is the straightforward extension of 1D `fftshift`), and
- $\mathcal{P}_{\Omega}$ is the subsampling operator.

In the following, you are allowed to use `FFTW.fft`, `FFTW.ifft`, and `FFTW.fftshift`.

Make sure that your function works for both odd and even m and n.

1. Write a function to evaluate
$$\boldsymbol{y} = \mathcal{P}_{\Omega} \circ \mathcal{S} \circ \mathcal{F}_{2\mathrm{D}}(\boldsymbol{X}) = \mathcal{A}(\boldsymbol{X})$$
without using matrix representation of the linear map $\mathcal{A}$.
The outputs of your function must include $\boldsymbol{O}_1 = \mathcal{F}_{2\mathrm{D}}(\boldsymbol{X})$, $\boldsymbol{O}_2 = \mathcal{S} \circ \mathcal{F}_{2\mathrm{D}}(\boldsymbol{X})$, and $\boldsymbol{O}_3 = \mathcal{P}_{\Omega} \circ \mathcal{S} \circ \mathcal{F}_{2\mathrm{D}}(\boldsymbol{X})$.
Use the data in the data file for a test. [6]
2. Write a function to evaluate
$$\boldsymbol{Z} = \left(\mathcal{P}_{\Omega} \circ \mathcal{S} \circ \mathcal{F}_{2\mathrm{D}}\right)^{*}(\boldsymbol{y}) = \mathcal{A}^{*}(\boldsymbol{y}).$$
without using matrix representation of the linear map.
The outputs of your function must include $\boldsymbol{O}_1 = \mathcal{P}_{\Omega}^{*}(\boldsymbol{y})$, $\boldsymbol{O}_2 = \mathcal{S}^{*} \circ \mathcal{P}_{\Omega}^{*}(\boldsymbol{y})$, and $\boldsymbol{O}_3 = \mathcal{F}_{2\mathrm{D}}^{*} \circ \mathcal{S}^{*} \circ \mathcal{P}_{\Omega}^{*}(\boldsymbol{y})$.
Use the data in the data file for a test. [6]

6.2. Least Squares

1. The goal of this question is to find the pseudo-inverse of the linear operator

$$\mathcal{A} = \mathcal{P}_\Omega \circ \mathcal{S} \circ \mathcal{F}_{2D}$$

without using matrix representation of the linear map.

a) Let $\boldsymbol{Y}_1 = \mathcal{F}_{2D}(\boldsymbol{X})$. Compute $\boldsymbol{Z}_1 = \mathcal{F}_{2D}^\dagger(\boldsymbol{Y}_1)$. [1]
b) Let $\boldsymbol{Y}_2 = \mathcal{S}(\boldsymbol{Y}_1)$. Compute $\boldsymbol{Z}_2 = \mathcal{S}^\dagger(\boldsymbol{Y}_2)$. [1]
c) Let $\boldsymbol{y}_3 = \mathcal{P}_\Omega(\boldsymbol{Y}_2)$. Compute $\boldsymbol{Z}_3 = \mathcal{P}_\Omega^\dagger(\boldsymbol{y}_3)$. [1]
d) Write a function to compute $\boldsymbol{Z} = \mathcal{A}^\dagger(\boldsymbol{y})$, where $\boldsymbol{y} = \mathcal{A}(\boldsymbol{X})$. [1]

Remark 6.2.1 Further insights into the pseudo-inverse can be gained by relating your solution to the singular value decomposition (SVD) of the matrix representations of the linear operators involved.

2. Consider the following recovery of undersampled MRI data:

$$\begin{aligned}\hat{\boldsymbol{X}} &= \arg\min_{\boldsymbol{X}} \frac{1}{2}\|\boldsymbol{y} - \mathcal{P}_\Omega \circ \mathcal{S} \circ \mathcal{F}_{2D}(\boldsymbol{X})\|_2^2 + \frac{\alpha}{2}\|\boldsymbol{X}\|_F^2 \\ &= \arg\min_{\boldsymbol{X}} \frac{1}{2}\|\boldsymbol{y} - \mathcal{A}(\boldsymbol{X})\|_2^2 + \frac{\alpha}{2}\|\boldsymbol{X}\|_F^2\end{aligned}$$

The above formulation can be written as

$$\left\|\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{0}\end{bmatrix} - \mathcal{B}(\boldsymbol{X})\right\|_2^2,$$

where $\boldsymbol{0}$ denotes a zero vector of proper dimension.

a) Find the dimension of the above zero vector. [1]
b) Write a function to compute

$$\boldsymbol{b} = \mathcal{B}(\boldsymbol{X}).$$

without using matrix representations of linear operators. [2]
c) Write a function to compute

$$\boldsymbol{Z} = \mathcal{B}^*\left(\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{0}\end{bmatrix}\right)$$

without using matrix representations of linear operators. [2]
d) It is clear that

$$\hat{\boldsymbol{X}} = (\mathcal{B}^*\mathcal{B})^{-1}\mathcal{B}^*\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{0}\end{bmatrix} = \mathcal{B}^\dagger\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{0}\end{bmatrix}.$$

Write a function to compute $\hat{\boldsymbol{X}}$. [5]
(*Hint*: Understanding of the SVD of the matrix representation of $\mathcal{B}^*\mathcal{B}$ is helpful in solving this problem. Let $\boldsymbol{B}$ be the matrix representation of $\mathcal{B}$. The matrix $\boldsymbol{B}^{\mathsf{H}}\boldsymbol{B}$ corresponds to the matrix representation of $\mathcal{B}^*\mathcal{B}$, and its SVD exhibits a convenient structure. Although deriving the explicit SVD of $\mathcal{B}^*\mathcal{B}$ is beyond the scope of this exercise, this insight will help

you to efficiently compute $(\mathscr{B}^*\mathscr{B})^{-1}$.)

Remark 6.2.2 The questions in this chapter adopt a matrix-free approach. As every finite-dimensional linear operator can be represented as a matrix, the standard method for solving least squares problems involves converting the problem into a matrix/vector form and then solving it using matrix (pseudo-)inverse. However, this approach is often computationally expensive and doesn't scale well. Instead, the questions are designed to explore the structure of the linear operators without explicit matrix inverse, allowing for much more efficient computations.

Part III.

Coursework 2

Greedy Algorithms | 7.

7.1. Estimation Accuracy

1. Alice, Bob, and Charlie's jobs are to estimate unknown $\boldsymbol{x}$ from their respective measurement vector $\boldsymbol{y}$ given by

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{w},$$

by solving the minimization problem

$$\hat{\boldsymbol{x}} = \arg \min_{\boldsymbol{z}} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}\|^2 .$$

Daniel knows the ground truth $\boldsymbol{x}_0$ and has the power to choose the noise vector $\boldsymbol{w}$ with $\|\boldsymbol{w}\|_2 = 1$ in generating $\boldsymbol{y}$.

a) Daniel is a friend of Alice. He would like to choose a noise vector $\boldsymbol{w}$ so that the estimation error

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}_0\|^2$$

is minimised.
Find the best $\boldsymbol{w}$ and compute the corresponding estimation error for Alice. [2]

b) Daniel is a foe of Bob. He would like to choose a noise vector $\boldsymbol{w}$ so that the estimation error

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}_0\|^2$$

is maximised.
Find the best $\boldsymbol{w}$ and compute the corresponding estimation error for Bob. [2]

c) For Charlie, Daniel randomly generates a noise vector $\boldsymbol{w}$ with i.i.d. Gaussian entries and then normalises it so that $\|\boldsymbol{w}\|_2 = 1$ (i.e., $\boldsymbol{w}$ has unit l_2 norm and statistically invariant under rotations). Find the expectation of the estimation error, i.e.,

$$\mathrm{E}\left[\|\hat{\boldsymbol{x}} - \boldsymbol{x}_0\|^2\right]$$

that Charlie may expect. [2]

7.2. Greedy Algorithms

1. Implement the OMP algorithm. Test its performance using the data given in the data file. [4]
2. Implement the SP algorithm. Test its performance using the data given in the data file. [4]
3. Implement the HIT algorithm. Test its performance using the data given in the data file. [4]

Dictionary Learning 8.

The dictionary learning problem refers to the task of learning a set of atoms that can be used to efficiently represent data. The mathematical formulation of the dictionary learning problem is as follows. Given a set of data consisting of N data points

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_N \end{bmatrix} \in \mathbb{R}^{M \times N},$$

dictionary learning seeks to solve the optimisation problem:

$$\begin{aligned} \min_{\boldsymbol{D},\boldsymbol{C}} \quad & \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2 \\ \text{subject to} \quad & \|\boldsymbol{d}_k\|_2 = 1,\ 1 \le k \le K \\ & \|\boldsymbol{c}_n\|_0 \le S,\ 1 \le n \le N, \end{aligned}$$

where $\boldsymbol{D}$ is the dictionary matrix, with each column $\boldsymbol{d}_k$ being a dictionary atom, the constraint $\|\boldsymbol{d}_k\|_2 = 1$ is to normalise the ℓ_2-norm of each atom, $\boldsymbol{C}$ is the coefficient matrix, with each column $\boldsymbol{c}_n$ representing the sparse coefficients for $\boldsymbol{y}_n$, and the constraint $\|\boldsymbol{c}_n\|_0 \le S$ means that each data point is represented using only up to S many dictionary atoms.

8.1. The MOD Method

Engan, Kjersti, Sven Ole Aase, and J. Hakon Husoy. "Method of optimal directions for frame design." In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 5, pp. 2443-2446. IEEE, 1999.

The MOD (Method of Optimal Directions) is a simple algorithm for solving the dictionary learning problem through alternating minimisation. Each iteration of the MOD algorithm consists of two main steps: sparse coding and dictionary update. In the sparse coding step, the current dictionary $\boldsymbol{D}$ is used to update the sparse coefficients $\boldsymbol{C}$ by solving the following problem:

$$\begin{aligned} \min_{\boldsymbol{C}} \quad & \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2 \\ \text{subject to} \quad & \|\boldsymbol{c}_n\|_0 \le S,\ 1 \le n \le N. \end{aligned}$$

In the dictionary update step, the entire dictionary is updated by solving a least squares problem:

$$\min_{\boldsymbol{D}} \quad \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2$$

followed by normalising each column of the dictionary to have unit ℓ_2-norm.

The data given in the data file includes the dataset $\boldsymbol{Y}$, and the initial values of the dictionary and sparse coefficients, denoted by $\boldsymbol{D}_0$ and $\boldsymbol{C}_0$ respectively. In the j-th iteration of the MOD algorithm, the process starts with $\boldsymbol{C}_{j-1}$ and $\boldsymbol{D}_{j-1}$. After the sparse coding step, $\boldsymbol{C}_{j-1}$ is updated into $\boldsymbol{C}_j$, and following the dictionary update step, $\boldsymbol{D}_{j-1}$ is updated to $\boldsymbol{D}_j$.

1. Implement the sparse coding step as a function `SC_OMP` using the OMP algorithm. Test it by using $\boldsymbol{Y}$ and $\boldsymbol{D}_0$ as inputs, and generate the output $\boldsymbol{C}_{OMP}$. [3]

2. Implement the sparse coding step as a function `SC_SP` using the Subspace Pursuit (SP) algorithm. Unlike the standard SP algorithm, here $\boldsymbol{C}_{j-1}$ is used as as the initialisation of the SP process in the j-th iteration of the MOD algorithm. Test it by using $\boldsymbol{Y}$, $\boldsymbol{D}_0$, and $\boldsymbol{C}_0$ as inputs, and generating the output $\boldsymbol{C}_{SP}$. [3]
3. Implement the dictionary update step as a function `DU_MOD`. Test it by using $\boldsymbol{Y}$ and $\boldsymbol{C}_0$ as inputs, and generating the output $\boldsymbol{D}_{MOD}$. [3]
4. Now implement the full MOD algorithm for dictionary learning by combining `SC_SP` and `DU_MOD`. Run 50 iterations. Save $\boldsymbol{C}_{50}$ and $\boldsymbol{D}_{50}$ into the data file. [3]
5. Let

$$f(\boldsymbol{C}, \boldsymbol{D}) = \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{C}\|_F^2 .$$

Compute and save $f(\boldsymbol{C}_j, \boldsymbol{D}_j)$, and the norm of the gradient of f with respect to $\boldsymbol{C}$, i.e., $\left\|\nabla_{\boldsymbol{C}} f(\boldsymbol{C}_j, \boldsymbol{D}_j)\right\|$, $j = 1, 2, \cdots, 50$. [3]

8.2. The K-SVD Method

Aharon, Michal, Michael Elad, and Alfred Bruckstein. “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation.” IEEE Transactions on signal processing, no. 11, vol. 54, (2006): 4311-4322.

K-SVD (K-means Singular Value Decomposition) is a widely-used algorithm designed to solve the dictionary learning problem. Like the MOD method, each iteration of K-SVD consists of a sparse coding step and a dictionary update step. The sparse coding step in K-SVD is the same as in the MOD algorithm. However, the dictionary update step in K-SVD is significantly different. Specifically, K-SVD updates dictionary atoms $\boldsymbol{d}_k$ one by one. For each column $k = 1, 2, \cdots, K$ in $\boldsymbol{D}_{j-1}$, the update follows the following steps:

(a) Identify the group of data samples that use atom $\boldsymbol{d}_k$

$$\omega_k = \{n : \boldsymbol{C}_{k,n} \neq 0, \ 1 \leq n \leq N\}.$$

(b) Compute the overall representation error matrix, $\boldsymbol{E}_k$, by

$$\boldsymbol{E}_k = \boldsymbol{Y} - \sum_{p \neq k} \boldsymbol{d}_p \boldsymbol{C}_{p,:}.$$

(c) Restrict $\boldsymbol{E}_k$ to the columns corresponding to ω_k, yielding $\boldsymbol{E}_{k,\omega}$.
(d) Perform SVD on $\boldsymbol{E}_{k,\omega}$: $\boldsymbol{E}_{k,\omega} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$. Update the k-th column of the dictionary $\boldsymbol{d}_k$ to be the first column of $\boldsymbol{U}$, and the coefficient vector $\boldsymbol{C}_{k,\omega_k}$ to be the first column of $\boldsymbol{V}$ multiplied by $\boldsymbol{\Sigma}_{1,1}$.

It is important to note that during the dictionary update step, both the dictionary and the sparse coefficients are updated, while the sparsity pattern remains unchanged.

1. Implement the dictionary update step of K-SVD as a function `DU_KSVD`. Test it by using $\boldsymbol{Y}$, $\boldsymbol{C}_0$, and $\boldsymbol{D}_0$ as inputs, and generating the output $\boldsymbol{C}_{KSVD}$ and $\boldsymbol{D}_{KSVD}$. [3]
2. Implement the full K-SVD algorithm by combining `SC_SP` and `DU_KSVD`. Run 50 iterations. Save $\boldsymbol{C}_{50}$ and $\boldsymbol{D}_{50}$ into the data file. [3]
3. Compute and save $f(\boldsymbol{C}_j, \boldsymbol{D}_j)$, and the norm of the gradient of f with respect to $\boldsymbol{C}$, i.e., $\left\|\nabla_{\boldsymbol{C}} f(\boldsymbol{C}_j, \boldsymbol{D}_j)\right\|$, $j = 1, 2, \cdots, 50$. [3]

Part IV.

Coursework 3

Part V.

Coursework 4