Indian Institute of Technology Kanpur
CS771 Introduction to Machine Learning

Instructor: Purushottam Kar
Date: August 17, 2025
Total: 50 marks

MINOR ASSIGNMENT 1

1 What should I submit, where should I submit and by when?

Your submission for this minor assignment will be one PDF (.pdf) file only. Instructions on how to prepare and submit this file are given below.

Assignment Package:
`https://www.cse.iitk.ac.in/users/purushot/courses/ml/2025-26-a/material/assignments/minor1.zip`
Deadline for all submissions: August 24, 2025, 9:59PM IST
Code Submission: ***None needed – do not submit any code (ipynb, py, etc) files for this minor assignment.***
Report Submission: on Gradescope
There is no provision for "late submission" for this minor assignment.

1.1 How to submit the PDF report file

1. The PDF file must be submitted using Gradescope in the *group submission mode*.
2. Note that unregistered auditors cannot make submissions to this minor assignment.
3. Make only one submission per assignment group on Gradescope, not one submission per student. Gradescope allows you to submit in groups - please use this feature to make a group submission.
4. Link all group members in your group submission. If you miss out on a group member while submitting, Gradescope will think that person never submitted anything.
5. You may overwrite your group's submission as many times as you want before the deadline (submitting again on Gradescope simply overwrites the old submission).
6. Do not submit Microsoft Word or text files. Prepare your report in PDF using the style file we have provided (instructions on formatting given later).

Problem 1.1 (Mess Made by Melbo)**.** Melbo was very excited to code up the algorithm discussed in the lecture "My First Solver" and wrote code to solve a C-SVM problem. Melbo's code is given to you in a Jupyter notebook in the assignment package. Unfortunately, in the rush to finish things, Melbo messed up at a few places so the resulting code does not really do its job properly.

1. Melbo's code gives horrible results (only around 30% test accuracy) on the arbiter PUF problem whereas we know that the C-SVM is known to give more than 95% test accuracy.

2. There is a interesting result in optimization theory called the *weak-duality* theorem that assures that the value of the dual objective function is always smaller than that of the primal objective function for all feasible solutions – at the optimal solution, the dual and primal objective values meet in the special case called *strong duality*. Melbo plotted the primal and dual objectives and they do not seem to be correct.

3. It is known that all mistakes made by Melbo are confined to code in cell number 5 of the Jupyter notebook. Other cells in the notebook are not expected to contain any coding errors – they mostly contain hyperparameter values such as value of C, initialization etc.

Your Data. We have provided you with data from an arbiter-PUF with 32-bit challenges. The file is the exact same one as used in the lecture demonstration. There are 100,000 challenges and corresponding responses in the file. Of these, a few (say 10,000) will be chosen randomly to be used for training and the rest will be used for testing. (50 marks)

Your Task. The following enumerates 7 parts to the question. All parts need to be answered in the PDF file containing your report. **Do not submit any code files for this minor assignment**. Thus, submit only a single PDF file as your submission.

1. Discover all coding errors made by Melbo in implementing the SDCM algorithm. Recall that all coding errors are confined to cell number 5 of the Jupyter notebook. Describe the errors in your report along with a description of why you think it is an error. (10 marks)

2. Once you have corrected Melbo's mistakes, what test accuracy do you get? Mention this and show what primal and dual objective curve do you get. Do not change any hyperparameter values or settings (e.g. value of C, number of training points, initialization, horizon etc) in any cell other than cell number 5 while answering this question. (10 marks)

3. Change the value of C between very small values e.g. 0.00001 and very large values e.g. 10000 without changing any of the other hyperparameters/settings. What changes do you observe in the test accuracy? What changes do you observe in the primal-dual objective curve? Please do these experiments after you have fixed Melbo's errors. (10 marks)

4. Experiment with at least two of the hyperparameters mentioned below and comment on their effect on test accuracy and primal-dual convergence curve. You must report the test accuracies and the convergence curves in your report after changing these hyperparameters. Please do these experiments after you have fixed Melbo's errors. (20 marks)

   (a) Effect of changing the coordinate generator. Currently it is set to randperm. What happens if we use cyclic? random?

   (b) Effect of changing the initialization initDual. Currently it is set to "all-C" values. What happens if we do an all zero initialization? Random initialization? You may use `numpy.random` submodule to generate random numbers.

   (c) Effect of horizon (the number of iterations) – note that this is passed as an argument to the function doSDCM. Currently it is set to 20,000. Does increasing it by a lot get a much better solution? Does decreasing it a lot make the solution much worse?

   (d) Number of training points (currently set to 1000). Does increasing or decreasing the number of training points by a lot affect test accuracy and convergence? If so, how?

   (e) Random chance – even if you keep all hyperparameters and settings fixed (say to their default values), does running the entire Jupyter notebook from first-to-last-cell again and again change the test accuracy? What could be the reason behind this?

Using Internet Resources. You are allowed to refer to textbooks, internet sources, research papers to find out more about this problem and for specific derivations e.g. the arbiter-PUF problem. However, if you do use any such resource, cite it in your PDF file. There is no penalty for using external resources with attribution but claiming someone else's work (e.g. a book or a research paper) as one's own work without crediting the original author will attract penalties.

Restrictions on Code Usage. You should not use any library other than what is already imported in the Jupyter notebook in the assignment package (you do not need to anyway). Since this minor assignment does not require any code submission, we will trust you on this one. However, you may use fancier libraries like seaborn or plotly to plot graphs etc.

2 How to Prepare the PDF File

Use the following style file to prepare your report.
`https://media.neurips.cc/Conferences/NeurIPS2023/Styles/neurips_2023.sty`

For an example file and instructions, please refer to the following files
`https://media.neurips.cc/Conferences/NeurIPS2023/Styles/neurips_2023.tex`
`https://media.neurips.cc/Conferences/NeurIPS2023/Styles/neurips_2023.pdf`

You must use the following command in the preamble

```
\usepackage[preprint]{neurips_2023}
```

instead of `\usepackage{neurips_2023}` as the example file currently uses. Use proper LaTeX commands to neatly typeset your responses to the various parts of the problem. Use neat math expressions to typeset your derivations. Remember that all parts of the question need to be answered in the PDF file. All plots must be generated electronically - hand-drawn plots are unacceptable. All plots must have axes titles and a legend indicating what are the plotted quantities. Insert the plot into the PDF file using LaTeX `\includegraphics` commands.