CSCI 2952-Q: Robust Algorithms for Machine Learning (Fall 2023)
Final Project Guidelines

Objectives:

1. **Topics.** Your project must be related to machine learning or theoretical computer science (preferably related to robustness but not required).

2. **In-Class Presentation.** Give a 15-minute presentation on your project (12-minute talk and 3-minute Q&A).

3. **Project Report.** Write a research paper/project report of at least 8 pages.

Teams: Each project team should have 1-2 students. All students in the same team will receive the same grade for the final project.

Submission Instructions:

- Submit your project report as a single .pdf file via Gradescope. You can include links to other materials (e.g., website, code) in your report.

- Project reports must be written in LaTeX. It is recommended (but not required) that you use style files from conferences such as ICLR, ICML, or NeurIPS.

- The main body of the project report must be **exactly** 8 pages, and there is no limit on the number of pages for references and appendices. Material beyond the first 8 pages will be read at the discretion of the reviewers.

- If any part of your project was done before taking the course or was used (or will be used) as a project for another course, you must clearly state this in your report. All project reports will be shared with the class and possibly future students.

Timeline:

- **Oct 9:** Decide on teams and project titles.
- **Nov 20 – Dec 4:** Present your project in class.
- **Dec 16:** Submit your project report.
- **Dec 21:** Review and submit ratings for all projects (optional).

All deadlines are 11:59pm on the due date.
Grading:  Projects will be graded out of 60 points (with up to 15 bonus points):

• **Meeting the deadlines** (20 points):
  – (10 points) Submit teams and project titles by Oct 9.
  – (10 points) All team members must show up on time to give an in-person presentation.

• **Project presentation and report** (40 points): We will run a lightweight peer-review process to grade your final projects. We will ask each student to review and rate all projects.

  The presentation and the report each count for 20 points. The following criteria are provided as guidelines. You can discuss/post your criteria on Ed.

  – Is the project relevant to machine learning or theoretical computer science?
  – How original and creative is the idea and the execution of the project?
  – Does the project show a deep understanding of related research areas and topics?
  – How useful is the project to students and researchers in the ML/TCS community?
  – Does the report clearly state the authors’ main contributions?
  – Does the report discuss the challenges incurred and highlight the novelty of the results?
  – Is the report well-structured, well-written, and technically sound?
  – Does the report discuss existing literature and provide a comparison with related work?
  – Does the presentation give a good overview of the project’s motivation and main results?
  – Does the presentation make the audience want to learn more about the project?
  – Are the speakers well-prepared? Is the presentation delivered at an appropriate pace?
  – Are the slides well-organized and effective?

In the case of building on existing research projects, the evaluation process should focus on the new components that are developed while taking the course.

• **Peer review** (up to 15 bonus points):

  – Each student can submit a score (22, 28, 34, or 40 points) or declare conflicts of interest for each project. The instructor will participate as a reviewer (with twice the weight).
  – Scores will be normalized to have the same mean and variance across reviewers. The final score of a project is the average of the normalized scores of all reviewers.
  – (Top projects) The top 20% of projects will receive 5 bonus points. The top 10% of projects will receive 5 additional bonus points (so 10 points in total).
  – (Top reviewers) If your evaluation is among the top 20% most accurate evaluations (measured by $\ell_2$-norm of the difference), you will receive 5 bonus points.
  – Names, titles, and reports of the top projects and reviewers will be announced on the public course homepage.
Examples of Project Topics. One possible approach is to (1) choose a machine learning problem that you are excited about, (2) implement state-of-the-art or widely-used algorithms for this problem, (3) explore various adversarial attacks and see whether these attacks compromise the performance of existing algorithms, and (4) design new robust algorithms against these attacks and prove/explain why your algorithms can defend against such attacks.

We provide examples of project topics from previous years to illustrate the variety and possibilities of what you can do. You are highly encouraged to choose a topic that excites you the most, which may not necessarily come from this list.

- Adversarial Attacks in CNN Gender Classification
- Bayesian Neural Networks
- Can We Distinguish Real and Generated Images?
- Computer Vision Techniques for Game Flow Tracking
- Diffusion Models in Computer Vision
- Digit Classification with the MNIST Database
- Directed Acyclic Graph Discovery With Perturb-Seq Data
- Flow Navigation via Reinforcement Learning
- Linear Reinforcement Learning
- Machine Translation From Speech
- Non-Negative Matrix Factorization for Single-Cell RNA Sequencing Data
- Nonparametric Density Estimation under Distribution Drift
- Perturbed Gradient Descent for Variational Quantum Algorithms
- Preventing AI Collusion Through Platform Intervention
- Privacy-Preserving Machine Learning
- Redundancy-Reduction Proxies for Self-Supervised Learning
- Robust Mean Estimation for Few-Shot Classification Meta Learning
- Sparsification in Graph Neural Networks
- Stability in Adversarial Reach-Avoid Games
- Tensor PCA and Its Empirical Study
- Visualizing Non-Convex Optimization Landscapes
Open Problems. We provide a list of open problems related to the course material.

- Open problems for semi-random non-convex matrix completion:
  - Can we recover the ground-truth matrix exactly (rather than approximately)?
  - Can semi-random attacks compromise the performance of state-of-the-art or widely-used algorithms (e.g., neural networks, gradient descent) for matrix completion?
  - Does the reweighting preprocessing algorithm improve the performance of state-of-the-art algorithms for matrix completion?
  - What is the best way to solve the reweighting problem in practice?
  - Can we efficiently solve other low-rank matrix problems in the semi-random model?

- Example open problems for high-dimensional robust statistics:
  - Are there practical applications in which replacing empirical mean or median with robust mean estimators would yield significantly better results?
  - Are there nearly-linear time robust mean estimation algorithms that can take advantage of existing deep learning frameworks (e.g., gradient-descent style)?
  - How can we visualize the non-convex optimization landscape of robust mean estimation?
  - Is there a nearly-linear time algorithm for robust sparse mean estimation?
  - Is there a nearly-linear time algorithm for robustly learning fixed-structure Bayesian networks without additional assumptions (i.e., balancedness and lower bound on parental configuration probabilities)?

- Example open problems for spectral graph theory:
  - Implement different graph sparsification algorithms and compare their performances and runtimes.
  - Implement different packing/covering SDP solvers and compare them with solvers from existing optimization packages.
  - Implement different algorithms for solving (undirected or directed) electric flow and maximum flow.
  - Give better or simpler algorithms for constructing low-stretch spanning trees.
  - Is there an $O(n \log^3 n)$ algorithm for undirected approximate maximum flow?

- Example open problems for algorithmic game theory:
  - Are there practical applications that would benefit greatly from recent results on strategic classification, Bayesian persuasion, or planning with strategic agents.
  - Is there a polynomial-time algorithm for socially optimal private signaling in non-atomic Bayesian network routing games?
  - Is there an fully polynomial-time approximation scheme for computing Nash equilibria in anonymous games?