Privacy in Statistics and Machine Learning Spring 2021
Project Guidelines

Adam Smith and Jonathan Ullman

Overview The course project is an opportunity to do something new with the material we are learning in class. You may form teams of one to three people.

We don't want to place too many constraints on what you do. However, broadly, there are two types of projects: applications projects and foundations projects.

In an applications project, the idea is to take one or more algorithms from the literature and evaluate them on a new dataset or application. Ideally, you might even improve on those algorithms, or adapt them to a slightly different problem.

In a foundations project, the idea is to summarize one or more papers, and present one or two of the main technical ideas in depth. Ideally, a theory project will also discover new results that go beyond what is in the literature, or survey papers that were not previously jointly discussed.

For both types of projects, the expectations for the scope of the project scale linearly with the number of team members.

Milestones

Project Proposal (1 page): Briefly describe the problem you are going to address and your plan for getting started. Include information like the key papers (or other materials) you're reading as part of your project, the tools/languages/datasets you will use (if relevant), and what final outcomes you're hoping for. Research evolves as it goes on, but the more planning you do at this stage, the better!

Progress Report (1 page): Update us on how the project is going so far. How have your goals and research plan changed and why? Presentation(?): Prepare a short talk about your work. Describe the main question you addressed, your methods, and the main outcomes. Your talk should be understandable given what we've seen in the class, and should be accessible to the other students.

Final Report: Prepare a final report describing the problem you worked on and your results. Your report should have roughly the form of a conference paper, describing the problem and its motivation, the key background work, the methods you use, and the results you obtained.

The report should be written in your group's own words—no verbatim copying is allowed except for definitions, algorithms and theorem statements. A good length target is 4–5 pages single-column, single-spaced 12pt text per team member. The exact format isn't a hard constraint; for example, part of the report might be a python notebook. Longer reports are OK as long as the extra length is making the report easier to read, not harder—for example, figures are encouraged, and it is fine to include extra figures beyond the basic page limit.

Draft Final Report A little more than a week before the final report is due, we ask you to submit a draft of the final report to give us an idea of what your report willl cover and what you accomplished. We will give you quick feedback on the report, and notably on whether anything major needs to change for the final report.

Audience for Final Report: Your classmates. The report should be written to be read and understood by other students in the class. You may assume they have understood the material in the class, but you should present whatever additional background is necessary to understand the main ideas of your report. The final reports will be made available via the course web page.

Deadlines

- Project Proposals: Friday March 5th
- Progress Report: Friday April 2th
- Draft Final Report: Friday, April 23
- Presentations: The week of April 26th
- Final Report: Friday April 30th

Project Ideas Here are some initial project ideas to help with brainstorming. This list is just suggestions and we'd be delighted to hear your ideas for projects that have nothing to do with anything on this list. If you are interested in learning more about one or more of these areas, we can provide some relevant background material tailored to your interests. We'll keep updating this list as we think of more projects.

- Reidentification and reconstruction attacks
  - Try claiming a prize from Aircloak for their Diffix Challenge
  - Try extracting user information from the Facebook or Google advertising interface

- Memorization in machine learning
  - Vision models: to what extent can trained models be used to recover specific patches of their inputs?

- Differential privacy in statistics: come up with improved private algorithms for statistical inference problems
  - Hierarchical models
  - Random graph models
  - Confidence intervals and other uncertainty estimates for DP algorithms

- Differential privacy for causal inference

- Differentially private optimization: improve the theoretical understanding of optimization on private data
  - Bounds for optimization of nonconvex functions under DP
  - Improved bounds for differentially private online learning

- Improving accuracy by incorporating public data sources: take differentially private algorithms for some problem and explore how having some small amount of additional public data might help make the algorithm work better

- Differential privacy and streaming algorithms: are there problems that can be solved privately and can be solved by small-space streaming algorithms, but not both at the same time?
  - Example: connected components in graphs under edge DP

- Auditing differentially private algorithms: choose some specific implementation of a differentially private algorithm and “stress test” it, to see whether the theoretical privacy analysis hold for the real code, and how close the analysis is to being tight
- Example: experiment with different DP algorithms using large values of eps (e.g. eps = 5) to see whether any realistic attacks are possible
- Example: identify vulnerabilities arising from floating point arithmetic
- The interplay between DP and other desiderata for learning and statistics:
  - Fairness
  - Robustness to adversarial examples
  - Robustness to training data poisoning
  - Robustness to outliers