1 Project outline

The class project is a research project and as such it is aimed at learning something new or exploring some unknown algorithm’s performance. A good project should be composed out of the following three components. The extent to which you choose to focus on one aspect is, as ever, up to you. You may even disregard the other two completely (e.g., a theory-oriented project that focuses on designing a new algorithm and proving formally its performance guarantees need not implement that algorithm). Regardless of your focus, it should be evident that you are conducting a methodological research: present a “high-level” question from which you derive your research question(s), study relevant literature, plan towards achieving your goals, adjust your research questions and analysis when needed, and (hopefully) obtain interesting results and conclusions.

1.1 Design and Analysis

Design an algorithm for a specific task that you are interested in. This could mean that you’re taking a canonical problem and trying to come up with an algorithm whose guarantee outperforms known algorithm’s guarantees. Alternatively, you may wish to design an algorithm which is a privacy-preserving version of an existing algorithm. In either case, it should be evident what is the problem your algorithm tackles, and why is it guaranteed to preserve differential privacy.

In addition, you should provide an analysis of your algorithm. You may choose to take an existing algorithm and improve its analysis (or even achieve exact constants in stead of big-$O$ notation). It could be that tightening down the algorithm’s constant may cause you to alter the algorithm. If you design an algorithm for which you cannot give exact bounds, you should aim for showing some performance guarantees (e.g. under assumptions as to the nature of the input). Alternatively, you may choose to prove a matching lower bound to the performance of an algorithm.

1.2 Implementation

You may choose to implement your algorithm or implement an algorithm presented/discussed in class. If your focus is on implementation, than we ask for a clean and modular implementation, preferably in $\mathcal{R}$, with a demonstration of your implemented algorithm’s ability to run over a multitude of datasets.

In addition, we do note that we have existing (tested) code for basic differentially private algorithms (such as releasing the means, the variance and noisy histogram) which you may borrow and leverage on. That is, you may for example want to compare the performance between two
algorithms, one implemented by yourselves and one we supply. Alternatively, you might wish to implement a more complicated algorithm that invokes any of our existing algorithm.

Existing code we have is for:

- means and variance
- noisy histogram
- quantiles
- outputting a CDF of a variable

You can also search the web for other implementations.

1.3 Experimentation

Upon implementation of a differentially privacy algorithm, we ask that you experiment and run your algorithm on a variety of datasets. In addition to synthetic datasets, the IQSS in Harvard has a variety of datasets for you to test your implementation on. You may wish to take an existing code and extensively test it on many datasets in an attempt to characterize the algorithm’s performance. Whatever you choose to do, you should decide on specific evaluation criteria, and present an experimentation plan towards this evaluation.

2 Project Schedule

Your project proposals are due by the beginning of Tuesday’s class on Oct 14th. The proposal should include your high-level research question, a nonempty-list of papers you aim to read and base your project on, and a list of lower-level questions based on the material you observed from the papers. Proposing a line of research does not necessarily mean you are obligated to provide answers to all of the questions you pose. However, the proposal is an important step in assuring you are conducting methodical research.

Initial project presentations will take place in class during the end of October / beginning of November. These are perhaps the most important step in your projects. You are to give a 10-15 mins talk about:

- your project’s research question.
- the papers you have read – what were the problems tackled in the paper? what algorithms do the authors propose? what are they guaranteeing? where does your project fits in with those works?
- the plan of attack for the rest of the project.

After presenting your project and your “plan of attacked” is agreed upon, you are to submit a time-line for your project.

Final presentations are due at the end of the semester (first week of December). This does not leave you with a lot of time, so we advise you begin as early as possible and communicate regularly with the course team. These presentations too are expected to be 15 mins long. The presentation should include your results and conclusions. Following the presentations, and potentially additional suggestions from us, you are to write a final report summarizing the project, due December 7th.
Note. Needless to say, full attendance of the entire class during projects’ mid-way presentations and final presentations is mandatory. We expect you to listen, participate and contribute to your fellow classmates’ projects, especially regarding suggested research questions or “plans of attack”.
3 Topic Suggestions

Below we have written several suggestions for topics we believe are interesting and merit further investigation. However, we encourage you to have your project related to a field that excites you. Ultimately, the subject of your work is up for you to decide.

3.1 CDFs, Threshold Queries and Quantiles

There are multiple privacy-preserving algorithms for answering threshold queries, computing a variable’s CDF and its quantiles. Possible suggested algorithms include


We already have a partial implementation of the tree-based algorithm. However, you may choose to implement other algorithms and/or check whether post-processing heuristics indeed improve the algorithms’ accuracy significantly.

3.2 Theory of Machine Learning

Computational learning theory is a great source of interesting problems that would be nice to solve in a differentially private way, and there have been several works exploring whether there are efficiency gaps (statistical or computational) between private and non-private learning.


Potential project topics: (i) explore some of the remaining gaps between private learning and non-private learning (e.g. in sample complexity, challenging). (ii) Design private learners for specific tasks.
3.3 Statistics/Machine Learning

There has been a lot of interesting work designing differentially private algorithms for data analysis tasks commonly used in machine learning and statistics, e.g. logistic regression, SVMs, maximum likelihood estimation. Many of these algorithms have strong theoretical guarantees and good practical performance.


Potential project topic: implementing one of such algorithms and optimize its analysis; attempt to tackle the same learning problem with multiple differentially private types of learners.

3.4 Privacy in Social Networks/Graph-Analysis

Many datasets do not naturally fit the model of individual records that we have used in class. One such example is when the data is represented by a graph, e.g. a social network. So far there have been some interesting attempts to look at natural data analysis tasks on graphs but there are a lot of open problems.


Potential Project Topics: 1) Differentially private algorithms for other kinds of graph or social network analyses, beyond those that have been studied so far. 2) Are there efficient algorithms for releasing a private synthetic graph approximately preserving the size of every cut? What about on natural families of graphs? What about the related problem of detecting communities in a social network? 3) How about privatizing a learner / sampler that is based on a social network, designed by studying a multitude of personal networks? (i.e., Given G and a set of node v1, ..., vl we make a learning observation based on l sub-graphs, each generated by restricting G to vj and all nodes within distance d = O(1) from vj.)
3.5 Mechanism Design and Privacy

Differential privacy has many interesting connections to mechanism design. On one hand, differentially private algorithms lead to a certain kind of truthfulness that can be difficult to achieve otherwise. On the other hand, interesting new questions in mechanism design arise when we incorporate privacy concerns into agents’ utility functions. This problem has been looked at from many angles and there are many interesting remaining technical and modeling questions.


3.6 Programming Frameworks and System Design for Differential Privacy

Since differentially private algorithms are often difficult to design and analyze, an attractive possibility would be to design a “differentially private programming language” – one in which we could ensure that every program written satisfies differential privacy, even if the programmer has never heard of differential privacy. There have been several interesting attempts at designing such a language, as well as designing larger systems that incorporate differential privacy.


Potential project topic: present a general design for a larger system that incorporates differential privacy for some setting not addressed by the existing work, analyze the system-level privacy guarantees, and the tradeoffs between privacy, utility, usability, and efficiency your system offers.
3.7 Experimental Evaluation of Differential Privacy

There have been some efforts to experimentally study and heuristically improve both the utility and computational efficiency of differentially private algorithms.

- An Adaptive Mechanism for Accurate Query Answering under Differential Privacy. Li, Miklau, 2012. (And related papers by Miklau et al.)
- Privately Solving Linear Programs. Hsu, Roth, Roughgarden, Ullman, 2014.

Project Ideas: Either try to improve an existing implementation of an algorithm, or thoroughly test the performance of a given (already coded) algorithm.