EN.601.774 Theory of Replicable ML Spring 2025

Project Ideas

Instructor: *Jess Sorrell* Due Friday, Mar 7

1 Replicable learning (or other algorithms) over real-world distributions

We've already seen one algorithm for replicable learning of finite hypothesis classes, and will soon see another Bun et al. [2023] (Section 5.3), which is much more sample-efficient. For a simple hypothesis class (something where you can computationally afford to iterate over all hypotheses in the class), evaluate whether replicable learning can be achieved *on average* with fewer samples than required by the worst-case theoretical guarantees. Note that since these worst-case bounds are asymptotic, you'll need to evaluate how the number of samples required to hit a target replicability failure parameter ρ grows as $\rho \to 0$.

You can consider versions of this project for other existing replicable algorithms as well, such as heavy hitters, reinforcement learning, clustering, etc. If you have a learning algorithm in mind, just email me and I can send you references to any formally replicable or stable versions that I'm aware of.

2 "Easy" replicability for strongly convex optimization

For optimization problems in which we try to optimize a strongly convex function over a dataset drawn from a target distribution, there's a simple approach to obtain replicable outputs via post-processing. Optimize the loss on the dataset nonreplicably and then round the model to a randomly chosen canonical representative (like a high-dimensional statistical query). The rounding process can hurt utility quite a bit, however. Empirically (and/or theoretically) investigate the cost of replicability (e.g. sample overhead) in your favorite strongly convex optimization problem.

3 Hyperparameter tuning as adaptive statistical queries

Hyperparameter tuning is a common step in many model training workflows, where a model's loss on "unseen" data is tested using a holdout set. If model loss is tested repeatedly on the entirety of the holdout data, this process of hyperparameter tuning can be modeled as a sequence of adaptive statistical queries evaluated on reused data. We've seen in lecture that data reuse for adaptive statistical queries can lead to badly overfitting ones data.

- Are there natural examples of hyperparameter tuning workflows that can be shown to provably lead to overfitting (like our query learner from lecture 9)?
- Empirically investigate whether hyperparameter tuning leads to overfitting ones holdout data in practice (by splitting holdout data into a set to reuse for hyperparameter tuning and a set to validate loss of the tuned model).

References

Mark Bun, Marco Gaboardi, Max Hopkins, Russell Impagliazzo, Rex Lei, Toniann Pitassi, Satchit Sivakumar, and Jessica Sorrell. Stability is stable: Connections between replicability, privacy, and adaptive generalization. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 520–527, 2023.