Differentially Private Streaming Algorithms in PINQ

Lucas Waye

Harvard School of Engineering and Applied Sciences, 2nd year PhD

INTRODUCTION

Most research on programming frameworks for differential privacy is concerned with static data sets. Every computation is a “one-shot” result and nothing needs to be recomputed because the results will be the same up to randomness introduced for privacy. However, there are many circumstances where this model does not make sense, or is simply infeasible. Support for dynamically changing data has been researched in the form of differentially private streaming algorithms. I present a practical framework for which a non-expert can perform differentially private operations on streams. The system is built as an extension to PINQ, a differentially private programming framework for static data sets.

Most of the research and techniques of differential privacy have been with respect to a single static database. This precludes many environments where this approach is not feasible. Situations where one would like to support streaming data sets.

CONTRIBUTIONS

- Extended PINQ to support streaming algorithms
- Support for different properties of streaming algorithms Pan-Privacy Continuous Output User-Level Privacy
- Implemented five differentially private streaming algorithms in the Streaming PINQ framework

STRENGTHS

- Streaming PINQ makes it simple to implement differentially private streaming algorithms. It is implemented in roughly 1000 lines of C# code. The system is meant to be “look and feel” like PINQ. Many of the classes are entirely new and do not rely on the PINQ object model directly, but the same coding style is adopted for the programmer’s ease of use and understanding.
- PINQStreamingAgent - Agents are responsible for enforcing that a differentially private operation is preserved for the stream. There are two inheriting classes that enforce either user-level privacy or event-level privacy. In the user-level privacy agent, privacy can never be returned to the stream. On the other hand, when viewing the stream with event-level privacy, the agent only needs to make sure that at most ε is “learned” for each event.
- The table above shows the implemented algorithms in Streaming PINQ. Note that ε is removed from accuracy measurements. ε and δ are user-defined parameters to the algorithm. Algorithms with an asterisk (*) denote known optimal accuracy for their listed properties. Buffered Average is simply adding just enough Laplace noise to achieve differential privacy. Randomized Response Count’s error match the theoretical lower bound from [2]’s negative result, given its properties (pan-private and continuous observation). Pan-Privacy is with respect to one intrusion.

REFERENCES