**Introduction**

**Database with binary attributes**

- Data analysts need statistical information about the database.
- Analysts ask counting queries.
- E.g., What fraction of the patients are smokers?
- Most queries from the analysts focus on a small number of attributes.
- What fraction of the records have a specific attribute value?
- Goal of the server: Return accurate answers without revealing information about individual records.

**Differential Privacy**

- Database $D$ contains $n$ records, each containing $d$ attributes, $D \in ([0,1]^d)^n$.
- Query family $Q$.
- Server employs a randomized algorithm $A: ([0,1]^d)^n \to [0,1]^d$.
- $(c,\varepsilon)$-Privacy: For any database $D$, every record $x \in D$, $\Pr[A(D) \neq x] \leq e^\varepsilon$.
- For any subset $S \subseteq [d]$, the output of $A$ on $D$ has Hamming weight at most $c \cdot \log_2 n$.

**Related Work**

- Accuracy and Privacy are conflicting goals.
- Converting data into lossless summaries is especially crucial for databases with small number of records.

**Results**

**Question:** Can we construct a k-way marginal query to design faster private algorithms that are accurate on databases of size $O(n^k)$?

- **Results for k-way Disjunction Query**
  - Since a subset $S \subseteq [d]$ of size at most $k$.
  - Answer is the fraction of records $r$ in the database for which at least one of the attributes of the set $S$ is TRUE.
  - The function value $f_S(r)$ is the answer to the query $S$ on the database $D$.

- **Simplifying the problem**
  - Suppose we have a polynomial $p_D$ for each function $f_D: [0,1]^d \to \{0,1\}$.
  - Sum of absolute values of coefficients of $p_D(x)$ is at most $\kappa$.

**Polynomial Approximations**

- Let $h_k \subseteq \{0,1\}^d$ be the subset of inputs with at most $k$ bits set to TRUE.
- Goal: Find a polynomial $p_h: [0,1]^d \to \{0,1\}$ such that $p_h(x) = f_{h_k}(x)$ for every input $x \in h_k$.
- Sum of absolute values of the coefficients of $p_h(x)$ is at most $W$.
- Degree of $p_h$ is $\leq t$.

**Question:** What is the least possible $W$ and $t$?

**Approach**

**(1) Learning Approach to Designing Private Algorithms**

- For each $D \in ([0,1]^d)^n$, there is an underlying function $f_D: [0,1]^d \to \{0,1\}$.
- Input to the function is the indicator vector $v$ of the query set $S$.
- $k$-way function $dq_S$ inputs to the function have a most $k$ bits set to TRUE.

**Differentially Private Algorithms**

- Maximum database size $n$.
- Number of monomials.
- Polynomial $p_D(x)$ is at most $\kappa$.
- Accuracy $\epsilon$. Appear to be the subset $\{x \in [0,1]^d : |L(x)| \leq \kappa\}$.

**Polynomial Approximations**

- Suppose we have a polynomial $p_D$ for each function $f_D: [0,1]^d \to \{0,1\}$.
- Sum of absolute values of coefficients of $p_D(x)$ is at most $\kappa$.
- Degree of $p_D$ is $\leq t$.

- Can derive a learning algorithm:
  - Given samples $\{g_D(x)v_D\}$, need to learn a hypothesis $h$ satisfying $|L(h) - L(g_D)| < \delta$ for every input $x$.
- The polynomial $p$ is a function of the polynomial $f_D$.
  - Use Multiplicative Weights Method.
  - Each monomial is an expert.
  - The weight on the expert is the coefficient of the monomial.

- Construct: View disjunction as a function of $m$ disjunctions.
- Optimize the final parameters.
- Input to the outer level disjunction has Hamming weight at most $k$.

**Lower bounds:**

- Exponentially strong lower bound on the linear size of the linear decision list guarantee constant accuracy for a k-way query.

**Future Directions**

- Use polynomial approximations for $(OR)_{k=0}^m$ to derive goal online algorithms and turn faster private and accurate algorithms.
- The polynomial approximations can be viewed as linear combination of monomials.
- A linear combination of some other small set of functions with similar properties can be used by the same approach to improve run-time.
- There is a collection of functions $\{f_{\Gamma_1}, \ldots, f_{\Gamma_m}\} \subseteq \{1,-1\}$ such that for each $x \in [0,1]^d$, there exists a linear combination $\sum_{\Gamma \in C} \alpha_{\Gamma} f_{\Gamma}(x)$.

**References**


