Differentially Private Distributed Convex Optimization via Functional Perturbation

Erfan Nozari

Department of Mechanical and Aerospace Engineering
University of California, San Diego
http://carmenere.ucsd.edu/erfan

July 6, 2016

Joint work with Pavankumar Tallapragada and Jorge Cortés
Distributed Coordination

What if local information is sensitive?
What if local information is sensitive?
Motivating Scenario: Optimal EV Charging

[Han et. al., 2014]
Motivating Scenario: Optimal EV Charging
[Han et. al., 2014]

Central aggregator solves:

\[
\begin{align*}
& \text{minimize} & & U\left(\sum_{i=1}^{n} r_i\right) \\
& \text{subject to} & & r_i \in C_i, \quad i \in \{1, \ldots, n\}
\end{align*}
\]

- \( U = \) energy cost function
- \( r_i = r_i(t) = \) charging rate
- \( C_i = \) local constraints
Motivating Scenario: Optimal EV Charging

[Han et. al., 2014]

Central aggregator solves:

$$\text{minimize } \sum_{i=1}^{n} U(r_i)$$

subject to \( r_i \in C_i \quad i \in \{1, \ldots, n\} \)

- \( U = \) energy cost function
- \( r_i = r_i(t) = \) charging rate
- \( C_i = \) local constraints
Myth: Aggregation Preserves Privacy

• Fact: NOT in the presence of side-information

Toy example:

|   |     |
|---|-----|
| 1 | 100 |
| 2 | 120 |
| ... | ... |
| n | 90  |

Database Average = 110

⇒

|   |     |
|---|-----|
| 1 | 100 |
| 2 | 120 |
| ... | ... |
| n | 90  |

Side Information

Real example: A. Narayanan and V. Shmatikov successfully de-anonymized Netflix Prize dataset (2007)
Side information: IMDB databases!
Myth: Aggregation Preserves Privacy

- Fact: NOT in the presence of side-information
Myth: Aggregation Preserves Privacy

- Fact: NOT in the presence of **side-information**

| Database |   |
|----------|---|
| 1        | 100 |
| 2        | 120 |
| ...      |    |
| n        | 90  |

Average = 110

- Toy example:

Real example: A. Narayanan and V. Shmatikov successfully de-anonymized Netflix Prize dataset (2007)
Side information: IMDB databases!
Myth: Aggregation Preserves Privacy

- Fact: NOT in the presence of side-information

Toy example:

| Database | Side Information |
|----------|------------------|
| 1 100    | 2 120            |
| 2 120    | :                |
| :        |                  |
| n 90     |                  |

\[
\text{Average} = 110
\]

\[
\Rightarrow d_1 = 100
\]
Myth: Aggregation Preserves Privacy

- Fact: NOT in the presence of side-information

- Toy example:

  | Database | Side Information |
  |----------|------------------|
  | 1 100    | 2 120            |
  | 2 120    |                  |
  | ...      | ...             |
  | n 90     | 2 120            |
  |          |                  |
  |          | n 90             |

  Average = 110  \Rightarrow d_1 = 100

- Real example: A. Narayanan and V. Shmatikov successfully de-anonymized Netflix Prize dataset (2007)
  Side information: IMDB databases!
Outline

1. DP Distributed Optimization
   - Problem Formulation
   - Impossibility Result

2. Functional Perturbation
   - Perturbation Design

3. DP Distributed Optimization via Functional Perturbation
   - Regularization
   - Algorithm Design and Analysis
Outline

1 DP Distributed Optimization
   - Problem Formulation
   - Impossibility Result

2 Functional Perturbation
   - Perturbation Design

3 DP Distributed Optimization via Functional Perturbation
   - Regularization
   - Algorithm Design and Analysis
Problem Formulation
Optimization

Standard additive convex optimization problem:

\[
\begin{align*}
\text{minimize} & \quad f(x) \triangleq \sum_{i=1}^{n} f_i(x) \\
\text{subject to} & \quad G(x) \leq 0 \\
& \quad Ax = b
\end{align*}
\]

Assumption:
- \( D \) is compact
- \( f_i \)'s are strongly convex and \( C^2 \)
Problem Formulation
Optimization

Standard additive convex optimization problem:

\[
\begin{align*}
\text{minimize} \quad f(x) &= \sum_{i=1}^{n} f_i(x) \\
\text{subject to} \quad G(x) &\leq 0 \\
&\quad Ax = b
\end{align*}
\]

Assumption:
- \(D\) is compact
- \(f_i\)'s are strongly convex and \(C^2\)
Standard additive convex optimization problem:

\[
\text{minimize } \quad f(x) \triangleq \sum_{i=1}^{n} f_i(x)
\]

Assumption:
- \( D \) is compact
- \( f_i \)'s are strongly convex and \( C^2 \)
Problem Formulation
Optimization

Standard additive convex optimization problem:

\[
\begin{align*}
\text{minimize} \quad & f(x) \triangleq \sum_{i=1}^{n} f_i(x) \\
\text{subject to} \quad & x \in X
\end{align*}
\]

- A non-private solution
  [Nedic et. al., 2010]:

\[
x_i(k+1) = \text{proj}_X(z_i(k) - \alpha_k \nabla f_i(z_i(k)))
\]

\[
z_i(k) = \sum_{j=1}^{n} w_{ij} x_j(k)
\]

Assumption:
- \( D \) is compact
- \( f_i \)'s are strongly convex and \( C^2 \)
Problem Formulation
Optimization

Standard additive convex optimization problem:

$$\min_{x \in X} f(x) \triangleq \sum_{i=1}^{n} f_i(x)$$

- A non-private solution
  [Nedic et. al., 2010]:

$$x_i(k+1) = \text{proj}_X(z_i(k) - \alpha_k \nabla f_i(z_i(k)))$$

$$z_i(k) = \sum_{j=1}^{n} w_{ij} x_j(k)$$

Assumption:
- $D$ is compact
- $f_i$’s are strongly convex and $C^2$

Erfan Nozari (UCSD)  Differentially Private Distributed Optimization
Problem Formulation

Privacy

• “Information”: \( F = (f_i)_{i=1}^n \in \mathcal{F}^n \)
Problem Formulation

Privacy

- “Information”: \( F = (f_i)_{i=1}^n \in \mathcal{F}^n \)

- Given \((\mathcal{V}, \| \cdot \|_\mathcal{V})\) with \( \mathcal{V} \subseteq \mathcal{F} \),

### Adjacency

\( F, F' \in \mathcal{F}^n \) are \( \mathcal{V} \)-adjacent if there exists \( i_0 \in \{1, \ldots, n\} \) such that

\[
  f_i = f'_i \text{ for } i \neq i_0 \quad \text{and} \quad f_{i_0} - f'_{i_0} \in \mathcal{V}
\]
Problem Formulation
Privacy

- “Information”: \( F = (f_i)_{i=1}^n \in \mathcal{F}^n \)
- Given \((\mathcal{V}, \| \cdot \|_\mathcal{V})\) with \(\mathcal{V} \subseteq \mathcal{F}\),

Adjacency

\( F, F' \in \mathcal{F}^n \) are \( \mathcal{V} \)-adjacent if there exists \( i_0 \in \{1, \ldots, n\} \) such that

\[
f_i = f'_i \text{ for } i \neq i_0 \quad \text{and} \quad f_{i_0} - f'_{i_0} \in \mathcal{V}
\]

- For a random map \( M : \mathcal{F}^n \times \Omega \rightarrow X \) and \( \epsilon \in \mathbb{R}_{>0}^n \)

Differential Privacy (DP)

\( M \) is \( \epsilon \)-DP if

\[
\forall \text{ \( \mathcal{V} \)-adjacent } F, F' \in \mathcal{F}^n \quad \forall \mathcal{O} \subseteq X
\]

\[
P\{M(F', \omega) \in \mathcal{O}\} \leq e^{\epsilon_{i_0}} \| f_{i_0} - f'_{i_0} \|_\mathcal{V} P\{M(F, \omega) \in \mathcal{O}\}
\]
Case Study
Linear Classification with Logistic Loss Function

- Training records: \( \{(a_j, b_j)\}_{j=1}^N \) where \( a_j \in [0, 1]^2 \) and \( b_j \in \{-1, 1\} \)
- Goal: find the best separating hyperplane \( x^T a = 0 \)
Case Study
Linear Classification with Logistic Loss Function

- Training records: \( \{(a_j, b_j)\}_{j=1}^{N} \) where \( a_j \in [0, 1]^2 \) and \( b_j \in \{-1, 1\} \)
- Goal: find the best separating hyperplane \( x^Ta = 0 \)

Convex Optimization Problem

\[
x^* = \arg\min_{x \in X} \sum_{j=1}^{N} \left( \ell(x; a_j, b_j) + \frac{\lambda}{2} |x|^2 \right)
\]

- Logistic loss: \( \ell(x; a, b) = \ln(1 + e^{-ba^T x}) \)
Case Study
Linear Classification with Logistic Loss Function

- Training records: \( \{(a_j, b_j)\}_{j=1}^{N} \)
  where \( a_j \in [0, 1]^2 \) and \( b_j \in \{-1, 1\} \)

- Goal: find the best separating hyperplane \( x^T a = 0 \)

Convex Optimization Problem

\[
x^* = \arg\min_{x \in X} \sum_{i=1}^{n} \sum_{j=1}^{N_i} \left( \ell(x; a_{i,j}, b_{i,j}) + \frac{\lambda}{2} |x|^2 \right)
\]

- Logistic loss: \( \ell(x; a, b) = \ln(1 + e^{-b a^T x}) \)
Message Perturbation vs. Objective Perturbation

A generic distributed optimization algorithm:

\[ x_i^+ = h_i(x_i, x_{-i}) \]
Message Perturbation vs. Objective Perturbation

Message Perturbation:

Network

Message Passing

\[ i \rightarrow j \]

Local State Update

\[ x_i^+ = h_i(x_i, x_{-i}) \]

Objective Perturbation:

Network

Message Passing

\[ i \rightarrow j \]

Local State Update

\[ x_i^+ = h_i(x_i, x_{-i}) \]
Message Perturbation vs. Objective Perturbation

**Message Perturbation:**

Network

Message Passing

Local State Update

\[ x_i^+ = h_i(x_i, x_{-i}) \]

**Objective Perturbation:**

Network

Message Passing

Local State Update

\[ x_i^+ = h_i(x_i, x_{-i}) \]
Message Perturbation vs. Objective Perturbation

Message Perturbation:

Network

Message Passing

\[
\begin{align*}
  i & \quad \rightarrow \quad j \\
  x_i^+ &= h_i(x_i, x_{-i})
\end{align*}
\]

Local State Update

Objective Perturbation:

Network

Message Passing

\[
\begin{align*}
  i & \quad \rightarrow \quad j \\
  x_i^+ &= h_i(x_i, x_{-i})
\end{align*}
\]

Local State Update
Impossibility Result

Generic message-perturbing algorithm:

\[ x(k + 1) = a_T(x(k), \xi(k)) \]
\[ \xi(k) = x(k) + \eta(k) \]
**Impossibility Result**

Generic message-perturbing algorithm:

\[ x(k + 1) = a_I(x(k), \xi(k)) \]

\[ \xi(k) = x(k) + \eta(k) \]

**Theorem**

If

- The \( \eta \rightarrow x \) dynamics is **0-LAS**
- \( \eta_i(k) \sim \text{Lap}(b_i(k)) \) or \( \eta_i(k) \sim \mathcal{N}(0, b_i(k)) \)
- \( b_i(k) \) is \( O\left(\frac{1}{k^p}\right) \) for some \( p > 0 \)

Then **no \( \epsilon \)-DP** of the information set \( \mathcal{I} \) for any \( \epsilon > 0 \)
Impossibility Result: An Example

Algorithm proposed in [Huang et. al., 2015]:

\[ x_i(k + 1) = \text{proj}_X(z_i(k) - \alpha_k \nabla f_i(z_i(k))) \]

\[ z_i(k) = \sum_{j=1}^{n} w_{ij} \xi_j(k) \]

\[ \xi_j(k) = x_j(k) + \eta_j(k) \]
Algorithm proposed in [Huang et. al., 2015]:

\[
x_i(k+1) = \text{proj}_X(z_i(k) - \alpha_k \nabla f_i(z_i(k)))
\]

\[
z_i(k) = \sum_{j=1}^{n} w_{ij} \xi_j(k)
\]

\[
\xi_j(k) = x_j(k) + \eta_j(k)
\]
Algorithm proposed in [Huang et. al., 2015]:

\[ x_i(k + 1) = \text{proj}_X(z_i(k) - \alpha_k \nabla f_i(z_i(k))) \]

\[ z_i(k) = \sum_{j=1}^{n} w_{ij} \xi_j(k) \]

\[ \xi_j(k) = x_j(k) + \eta_j(k) \]

- \( \eta_j(k) \sim \text{Lap}(\alpha p^k) \)
- \( \alpha_k \propto q^k \quad 0 < q < p < 1 \)
Impossibility Result: An Example

Algorithm proposed in [Huang et. al., 2015]:

\[ x_i(k + 1) = \text{proj}_X(z_i(k) - \alpha_k \nabla f_i(z_i(k))) \]

\[ z_i(k) = \sum_{j=1}^{n} w_{ij} \xi_j(k) \]

\[ \xi_j(k) = x_j(k) + \eta_j(k) \]

- \( \eta_j(k) \sim \text{Lap}(\propto p^k) \) \quad 0 < q < p < 1
- \( \alpha_k \propto q^k \)

Finite sum
Algorithm proposed in [Huang et. al., 2015]:

- Simulation results for a linear classification problem:
Outline

1. DP Distributed Optimization
   - Problem Formulation
   - Impossibility Result

2. Functional Perturbation
   - Perturbation Design

3. DP Distributed Optimization via Functional Perturbation
   - Regularization
   - Algorithm Design and Analysis
State of the Art

- [Chaudhuri et. al., 2011]
  - First proposed “objective perturbation” by adding linear random functions
  - Extended by [Kifer et. al., 2012] to constrained and non-differentiable problems
  - Preserves DP of objective function parameters

- [Zhang et. al., 2012]
  - Proposed objective perturbation by adding sample path of Gaussian stochastic process
  - Preserves DP of objective function parameters

- [Hall et. al., 2013]
  - Proposed objective perturbation by adding quadratic random functions
  - Preserves DP of objective function parameters
State of the Art

- [Chaudhuri et. al., 2011]
  - First proposed “objective perturbation” by adding linear random functions
  - Extended by [Kifer et. al., 2012] to constrained and non-differentiable problems
  - Preserves DP of objective function parameters

- [Zhang et. al., 2012]
  - Proposed objective perturbation by adding sample path of Gaussian stochastic process
  - Preserves DP of objective function parameters
State of the Art

- [Chaudhuri et. al., 2011]
  - First proposed “objective perturbation” by adding linear random functions
  - Extended by [Kifer et. al., 2012] to constrained and non-differentiable problems
  - Preserves DP of objective function parameters

- [Zhang et. al., 2012]
  - Proposed objective perturbation by adding sample path of Gaussian stochastic process
  - Preserves DP of objective function parameters

- [Hall et. al., 2013]
  - Proposed objective perturbation by adding quadratic random functions
  - Preserves DP of objective function parameters
Prelim: Hilbert Spaces

• Hilbert space $\mathcal{H} = \text{complete inner-product space}$

• Orthonormal basis $\{e_k\}_{k \in I} \subset \mathcal{H}$

• If $\mathcal{H}$ is separable:

$$h = \sum_{k=1}^{\infty} \langle h, e_k \rangle e_k$$

• For $D \subseteq \mathbb{R}^d$, $L^2(D)$ is a separable Hilbert space $\Rightarrow F = L^2(D)$
Prelim: Hilbert Spaces

- Hilbert space $\mathcal{H} = \text{complete inner-product space}$

- Orthonormal basis $\{e_k\}_{k \in I} \subset \mathcal{H}$

- If $\mathcal{H}$ is separable:

$$h = \sum_{k=1}^{\infty} \langle h, e_k \rangle e_k$$

- For $D \subseteq \mathbb{R}^d$, $L_2(D)$ is a separable Hilbert space $\Rightarrow \mathcal{F} = L_2(D)$
• $\Phi$: coefficient sequence $\delta \rightarrow$ function $h = \sum_{k=1}^{\infty} \delta_k e_k$

• Adjacency space:

$$\mathcal{V}_q = \{ \Phi(\delta) \mid \sum_{k=1}^{\infty} (k^q \delta_k)^2 < \infty \}$$
Functional Perturbation via Laplace Noise

- $\Phi$: coefficient sequence $\delta \rightarrow$ function $h = \sum_{k=1}^{\infty} \delta_k e_k$

- Adjacency space:
  \[ V_q = \{ \Phi(\delta) \mid \sum_{k=1}^{\infty} (k^q \delta_k)^2 < \infty \} \]

- Random map:
  \[ \mathcal{M}(f, \eta) = \Phi \left( \Phi^{-1}(f) + \eta \right) = f + \Phi(\eta) \]
Functional Perturbation via Laplace Noise

- $\Phi :$ coefficient sequence $\delta \rightarrow$ function $h = \sum_{k=1}^{\infty} \delta_k e_k$

- Adjacency space:

$$\mathcal{V}_q = \{ \Phi(\delta) \mid \sum_{k=1}^{\infty} (k^q \delta_k)^2 < \infty \}$$

- Random map:

$$\mathcal{M}(f, \eta) = \Phi(\Phi^{-1}(f) + \eta) = f + \Phi(\eta)$$

Theorem

For $\eta_k \sim \text{Lap}(\frac{\gamma}{k^p})$, $q > 1$, and $p \in \left(\frac{1}{2}, q - \frac{1}{2}\right)$, $\mathcal{M}$ guarantees $\epsilon$-DP with

$$\epsilon = \frac{1}{\gamma} \sqrt{\zeta(2(q - p))}$$
Outline

1. DP Distributed Optimization
   - Problem Formulation
   - Impossibility Result

2. Functional Perturbation
   - Perturbation Design

3. DP Distributed Optimization via Functional Perturbation
   - Regularization
   - Algorithm Design and Analysis
Resilience to Post-processing

Algorithm sketch:

1. Each agent **perturbs its own** objective function (offline)
2. Agents **participate in an arbitrary** distributed optimization algorithm with perturbed functions (online)
Resilience to Post-processing

Algorithm sketch:

1. Each agent \textbf{perturbs its own} objective function (offline)
2. Agents \textbf{participate in an arbitrary} distributed optimization algorithm with perturbed functions (online)

\[
\mathcal{M} : L^2(D)^n \times \Omega \rightarrow L^2(D)^n \\
\mathcal{F} : L^2(D)^n \rightarrow \mathcal{X}, \text{ where } (\mathcal{X}, \Sigma_{\mathcal{X}}) \text{ is an arbitrary measurable space}
\]

**Corollary (special case of [Ny & Pappas 2014, Theorem 1])**

If \( \mathcal{M} \) is \( \epsilon \)-DP, then \( \mathcal{F} \circ \mathcal{M} : L^2(D)^n \times \Omega \rightarrow \mathcal{X} \) is \( \epsilon \)-DP.
Ensuring Regularity of Perturbed Functions

- $\hat{f}_i = M(f_i, \eta_i)$ may be discontinuous/non-convex/...
Ensuring Regularity of Perturbed Functions

- $\hat{f}_i = M(f_i, \eta_i)$ may be discontinuous/non-convex/...

- $S = \{\text{Regular functions}\} \subset C^2(D) \subset L_2(D)$
Ensuring Regularity of Perturbed Functions

- \( \hat{f}_i = M(f_i, \eta_i) \) may be discontinuous/non-convex/...

- \( S = \{ \text{Regular functions} \} \subset C^2(D) \subset L_2(D) \)

- **Ensuring Smoothness:** \( C^2(D) \) is dense in \( L_2(D) \) so

  \[
  \forall \varepsilon_i > 0 \text{ pick } \hat{f}_i^s \in C^2(D) \text{ such that } \| \hat{f}_i - \hat{f}_i^s \| < \varepsilon_i
  \]
Ensuring Regularity of Perturbed Functions

- $\hat{f}_i = M(f_i, \eta_i)$ may be discontinuous/non-convex/...

- $S = \{\text{Regular functions}\} \subset C^2(D) \subset L_2(D)$

- **Ensuring Smoothness:** $C^2(D)$ is dense in $L_2(D)$ so
  \[
  \forall \varepsilon_i > 0 \text{ pick } \hat{f}^s_i \in C^2(D) \text{ such that } \|\hat{f}_i - \hat{f}^s_i\| < \varepsilon_i
  \]

- **Ensuring Regularity:**
  \[
  \tilde{f}_i = \text{proj}_S(\hat{f}^s_i)
  \]

**Proposition**

$S$ is convex and closed relative to $C^2(D)$
1. Each agent **perturbs** its function:

\[ \hat{f}_i = \mathcal{M}(f_i, \eta_i) = f_i + \Phi(\eta_i), \quad \eta_{i,k} \sim \text{Lap}(b_{i,k}), \quad b_{i,k} = \frac{\gamma_i}{k \rho_i} \]

2. Each agent **selects** \( \hat{f}^s_i \in S_0 \) such that

\[ \| \hat{f}_i - \hat{f}^s_i \| < \varepsilon_i \]

3. Each agent **projects** \( \hat{f}^s_i \) onto \( S \):

\[ \tilde{f}_i = \text{proj}_S(\hat{f}^s_i) \]

4. Agents **participate** in any distributed optimization algorithm with \( (\tilde{f}_i)_{i=1}^n \)
Algorithm

1. Each agent **perturbs** its function:

   \[ \hat{f}_i = \mathcal{M}(f_i, \eta_i) = f_i + \Phi(\eta_i), \quad \eta_{i,k} \sim \text{Lap}(b_{i,k}), \quad b_{i,k} = \frac{\gamma_i}{k \rho_i} \]

2. Each agent **selects** \( \hat{f}_i^s \in S_0 \) such that

   \[ \|\hat{f}_i - \hat{f}_i^s\| < \varepsilon_i \]

3. Each agent **projects** \( \hat{f}_i^s \) onto \( S \):

   \[ \tilde{f}_i = \text{proj}_S(\hat{f}_i^s) \]

4. Agents **participate** in any distributed optimization algorithm with \( (\tilde{f}_i)_{i=1}^n \)
Accuracy Analysis

- Set of “regular” functions:

\[ S = \{ h \in C^2(D) \mid \alpha I_d \leq \nabla^2 h(x) \leq \beta I_d \text{ and } |\nabla h(x)| \leq \overline{u}\} \]

Lemma (K-Lipschitzness of argmin)

For \( f, g \in S \),

\[
\left| \arg\min_{x \in X} f - \arg\min_{x \in X} g \right| \leq \kappa_{\alpha, \beta}(\| f - g \|)
\]
Accuracy Analysis

- Set of “regular” functions:

\[ S = \{ h \in C^2(D) \mid \alpha I_d \leq \nabla^2 h(x) \leq \beta I_d \text{ and } |\nabla h(x)| \leq \bar{u} \} \]

**Lemma (\( \kappa \)-Lipschitzness of \( \arg\min \))**

For \( f, g \in S \),

\[ |\arg\min_{x \in X} f - \arg\min_{x \in X} g| \leq \kappa_{\alpha, \beta}(\|f - g\|) \]

- Define

\[ \tilde{x}^* = \arg\min_{x \in X} \sum_{i=1}^{n} \tilde{f}_i \quad \text{and} \quad x^* = \arg\min_{x \in X} \sum_{i=1}^{n} f_i, \]

**Theorem (Accuracy)**

\[ \mathbb{E} |\tilde{x}^* - x^*| \leq \sum_{i=1}^{n} \kappa_n \left( \frac{\zeta(q_i)}{\epsilon_i} \right) + \kappa_n(\epsilon_i) \]
Simulation Results
Linear Classification with Logistic Loss Function

Theoretical bound
Empirical data
Piecewise linear fit

Theoretical bound
2nd order
6th order
14th order
Conclusions and Future Work

In this talk, we

- Proposed a definition of DP for functions
- Illustrated a fundamental limitation of message-perturbing strategies
- Proposed the method of functional perturbation
- Discussed how functional perturbation can be applied to distributed convex optimization

Future work includes

- Relaxation of the smoothness, convexity, and compactness assumptions
- Comparing the numerical efficiency of different bases for $L_2$
- Characterizing the expected sub-optimality gap of the algorithm and the optimal privacy-accuracy trade-off curve
- Further understanding the appropriate scales of privacy parameters for particular applications
Conclusions and Future Work

In this talk, we

• Proposed a definition of DP for functions
• Illustrated a fundamental limitation of message-perturbing strategies
• Proposed the method of functional perturbation
• Discussed how functional perturbation can be applied to distributed convex optimization

Future work includes

• relaxation of the smoothness, convexity, and compactness assumptions
• comparing the numerical efficiency of different bases for $L_2$
• characterizing the expected sub-optimality gap of the algorithm and the optimal privacy-accuracy trade-off curve
• further understanding the appropriate scales of privacy parameters for particular applications
Questions and Comments

Full results of this talk available in:

E. Nozari, P. Tallapragada, J. Cortés, “Differentially Private Distributed Convex Optimization via Functional Perturbation,” *IEEE Trans. on Control of Net. Sys.*, provisionally accepted, http://arxiv.org/abs/1512.00369
Formal Definition
in original context [Dwork et. al., 2006]

Context:

- \( D \in \mathcal{D} \): A database of records
- Adjacency: \( D_1, D_2 \in \mathcal{D} \) are adjacent if they differ by at most 1 record
- \((\Omega, \Sigma, \mathbb{P})\): Probability space
- \( q : \mathcal{D} \rightarrow X \): (Honest) query function
- \( M : \mathcal{D} \times \Omega \rightarrow X \): Randomized/sanitized query function
- \( \epsilon > 0 \): Level of privacy

**Definition**

\( M \) is \( \epsilon \)-DP if

\[
\forall \text{ adjacent } D_1, D_2 \in \mathcal{D} \quad \forall O \subseteq X \quad \mathbb{P}\{M(D_1) \in O\} \leq e^\epsilon \mathbb{P}\{M(D_2) \in O\}
\]

- Adjacency is symmetric \( \Rightarrow \)

\[
\begin{align*}
\mathbb{P}\{M(D_1) \in O\} &\leq e^\epsilon \mathbb{P}\{M(D_2) \in O\} \\
\mathbb{P}\{M(D_2) \in O\} &\leq e^\epsilon \mathbb{P}\{M(D_1) \in O\}
\end{align*}
\]
Formal Definition: Geometric Interpretation in original context

Definition

\[ \mathcal{M} \text{ is } \epsilon\text{-DP if} \]

\[ \forall \text{ adjacent } D_1, D_2 \in \mathcal{D} \quad \forall \mathcal{O} \subseteq X \quad \mathbb{P}\{\mathcal{M}(D_1) \in \mathcal{O}\} \leq e^\epsilon \mathbb{P}\{\mathcal{M}(D_2) \in \mathcal{O}\} \]
Operational Meaning of DP
A binary decision example [Geng&Pramod, 2013]

- Adversary’s decision = \[
\begin{cases} 
\text{TRUE} & \text{if } M(D,\omega) \in \mathcal{O} \\
\text{FALSE} & \text{if } M(D,\omega) \in \mathcal{O}^c 
\end{cases}
\]

- \( MD = \{ M(D_1,\omega) \in \mathcal{O}^c \} \)
- \( FA = \{ M(D_2,\omega) \in \mathcal{O} \} \)

- If \( M \) is \( \epsilon \)-DP then

\[
\begin{align*}
\mathbb{P}\{M(D_1,\omega) \in \mathcal{O}\} & \leq e^\epsilon \mathbb{P}\{M(D_2,\omega) \in \mathcal{O}\} \\
\mathbb{P}\{M(D_2,\omega) \in \mathcal{O}^c\} & \leq e^\epsilon \mathbb{P}\{M(D_1,\omega) \in \mathcal{O}^c\}
\end{align*}
\]

\[
\Rightarrow p_{MD}, p_{FA} \geq \frac{e^\epsilon - 1}{e^{2\epsilon} - 1}
\]
Generalizing the Definition: Using Metrics
[Chatzikokolakis et. al., 2013]

- If $D_1, D_2$ differ in $N$ elements then
  \[ P\{M(D_1, \omega) \in \mathcal{O}\} \leq e^{N\epsilon} P\{M(D_2, \omega) \in \mathcal{O}\} \]

- $d : \mathcal{D} \times \mathcal{D} \to [0, \infty)$ metric on $\mathcal{D}$

**Definition – revisited**

$M$ gives/preserves $\epsilon$-differential privacy if

\[ \forall D_1, D_2 \in \mathcal{D} \quad \forall \mathcal{O} \subseteq X \text{ we have} \]
\[ P\{M(D_1, \omega) \in \mathcal{O}\} \leq e^{\epsilon d(D_1, D_2)} P\{M(D_2, \omega) \in \mathcal{O}\} \]