Differentially Private Release of Synthetic Graphs

Marek Eliáš

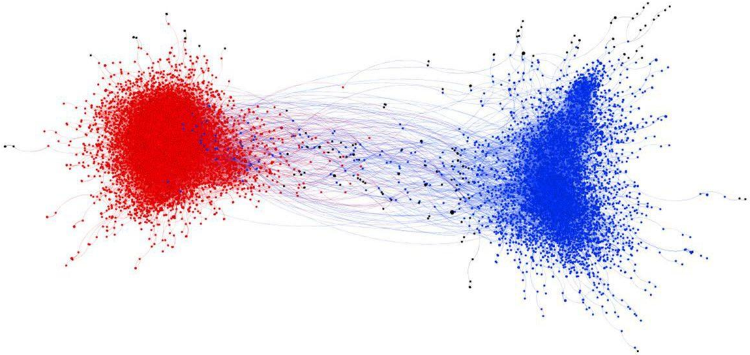
EPFL

Joint work with

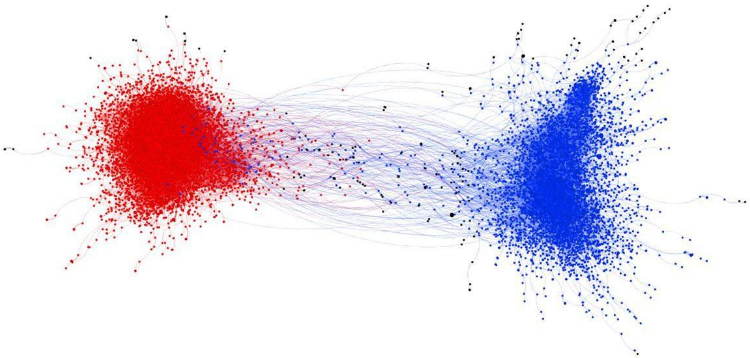
Michael Kapralov, Janardhan Kulkarni, Yin Tat Lee

European Research Council
Established by the European Commission

Private network analysis



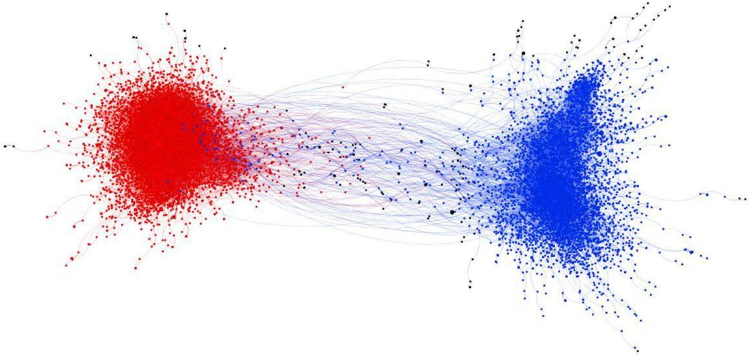
Private network analysis



Social networks:

- contain valuable information about our societies
- stability of the society, information spread

Private network analysis



Social networks:

- contain valuable information about our societies
- stability of the society, information spread

Network analysis in a private manner?

A synthetic graph approximating all cuts

Input:

- graph $G(V, E)$ with edge-weights w

Output:

- differentially private graph G' with weights w'
- for any $I, J \subset V$: $w'(I, J) \approx w(I, J)$
  - i.e., preserving weight of (I, J)-cuts

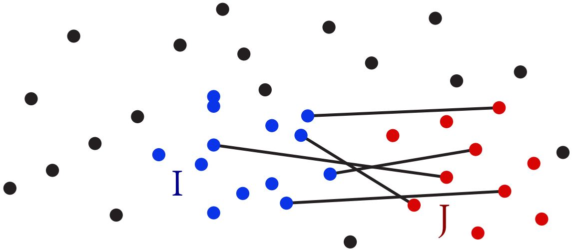
A synthetic graph approximating all cuts

Input:

- graph $G(V, E)$ with edge-weights w

Output:

- differentially private graph G' with weights w'
- for any $I, J \subset V$: $w'(I, J) \approx w(I, J)$
  - i.e., preserving weight of (I, J)-cuts



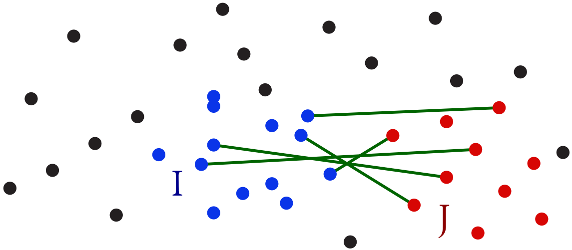
A synthetic graph approximating all cuts

Input:

- graph $G(V, E)$ with edge-weights w

Output:

- differentially private graph G' with weights w'
- for any $I, J \subset V$: $w'(I, J) \approx w(I, J)$
  - i.e., preserving weight of (I, J)-cuts



A synthetic graph approximating all cuts

Input:

- graph $G(V, E)$ with edge-weights w

Output:

- differentially private graph G' with weights w'
- for any $I, J \subset V$: $w'(I, J) \approx w(I, J)$
  - i.e., preserving weight of (I, J)-cuts

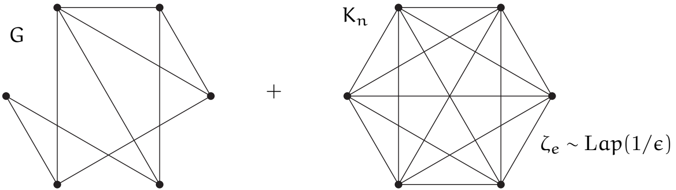
Edge-level privacy:

- neighboring graphs differ by a single edge

Known results

Randomized response

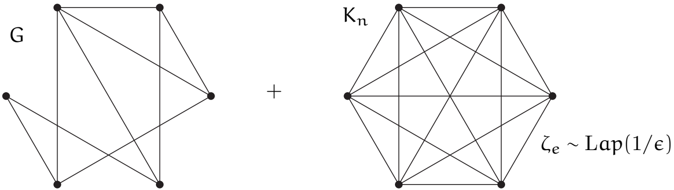
- Gupta, Roth, Ullman'12
- $w'_e = w_e + \zeta_e$, where $\zeta_e \sim \mathrm{Lap}(1/\epsilon)$ i.i.d.
- additive error: $O(n^{3/2})$
- useful only for graphs with $\gg n^{3/2}$ edges



Known results

Randomized response

- Gupta, Roth, Ullman'12
- $w'_e = w_e + \zeta_e$, where $\zeta_e \sim \mathrm{Lap}(1/\epsilon)$ i.i.d.
- additive error: $O(n^{3/2})$
- useful only for graphs with $\gg n^{3/2}$ edges



Other results

- Blocki, Blum, Datta, Sheffet '12; Upadhyay '13

Known results

Exponential mechanism: Naïve version

- score $\Theta(\exp(n^2))$ possible output graphs by their error
- return a sample from this distribution
- error proportional to n^2

[1] Only for cuts of type $(S, V \setminus S)$

Known results

Exponential mechanism: Naïve version

- score $\Theta(\exp(n^2))$ possible output graphs by their error
- return a sample from this distribution
- error proportional to n^2

Exponential mechanism: Improved version

- fundamental result: existence of sparsifiers
  - preserve cut sizes[1] with a small multiplicative error
  - number of edges: $O(n)$

[1] Only for cuts of type $(S, V \setminus S)$

Known results

Exponential mechanism: Naïve version

- score $\Theta(\exp(n^2))$ possible output graphs by their error
- return a sample from this distribution
- error proportional to n^2

Exponential mechanism: Improved version

- fundamental result: existence of sparsifiers
  - preserve cut sizes[1] with a small multiplicative error
  - number of edges: $O(n)$
  - only $\exp(O(n \log n))$ possible sparsifiers!
- additive error: $n \log n$, multiplicative error due to sparsification
- Drawback: exponential time!

[1] Only for cuts of type $(S, V \setminus S)$

Our result

Input:

- graph G^* s.t. $\sum_e w_e^* = m$

Output:

- (ϵ, δ)-DP synthetic graph G with weights w
- with probability $(1 - \gamma)$:
  - for all $I, J \subset V$: $|w(I, J) - w^*(I, J)| \leqslant \tilde{O}\left(\sqrt{mn}\right)$
- i.e. purely additive error

Our result

Input:

- graph G^* s.t. $\sum_e w_e^* = m$

Output:

- (ϵ, δ)-DP synthetic graph G with weights w
- with probability $(1 - \gamma)$:
  - for all $I, J \subset V$: $|w(I, J) - w^*(I, J)| \leqslant O\left(\sqrt{mn/\epsilon} \cdot \log^2(n/\delta)\right)$
- i.e. purely additive error

Our result

Input:

- graph G^* s.t. $\sum_e w_e^* = m$

Output:

- (ϵ, δ)-DP synthetic graph G with weights w
- with probability $(1 - \gamma)$:
  - for all $I, J \subset V$: $|w(I, J) - w^*(I, J)| \leqslant O\left(\sqrt{mn/\epsilon} \cdot \log^2(n/\delta)\right)$
- i.e. purely additive error
- first polytime alg. with non-trivial guarantee for sparse graphs

Our result

Input:

- graph G^* s.t. $\sum_e w_e^* = m$

Output:

- (ϵ, δ)-DP synthetic graph G with weights w
- with probability $(1 - \gamma)$:
  - for all $I, J \subset V$: $|w(I, J) - w^*(I, J)| \leqslant O\left(\sqrt{mn/\epsilon} \cdot \log^2(n/\delta)\right)$
- i.e. purely additive error
- first polytime alg. with non-trivial guarantee for sparse graphs

Lower bounds for purely additive error

$$\Omega(\sqrt{mn/\epsilon})$$

Should we use sparsification?

Algorithm by Spielman and Srivastava

- sample edges by their effective resistance
- number of edges: $O(\alpha^{-2} n \log n)$
- multiplicative error: $(1 + \alpha)$

Should we use sparsification?

Algorithm by Spielman and Srivastava

- sample edges by their effective resistance
- number of edges: $O(\alpha^{-2} n \log n)$
- multiplicative error: $(1 + \alpha)$

Problem:

- only existing edges are sampled
- edge e in the output $\Rightarrow$ e was present in the input!
- not private

Our approach

Find cut approximator using convex optimization

- mirror descent
- iterative technique
- we can choose target precision

Our approach

Find cut approximator using convex optimization

- mirror descent
- iterative technique
- we can choose target precision

Make each iteration private

- mirror descent only needs gradient as an input
- sanitize each gradient evaluation

Our approach

Find cut approximator using convex optimization

- mirror descent
- iterative technique
- we can choose target precision

Make each iteration private

- mirror descent only needs gradient as an input
- sanitize each gradient evaluation

Bound the total privacy

- Advanced composition theorem

Convex objective

- input graph G^*: weights w^*, adjacency matrix A^*
- current solution G: weights w, adjacency matrix A
- let $D = A - A^*$

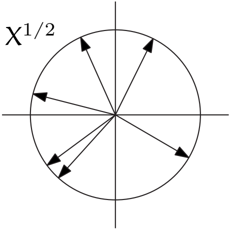
Convex objective

- input graph G^*: weights w^*, adjacency matrix A^*
- current solution G: weights w, adjacency matrix A
- let $D = A - A^*$

Grothendieck problem:

$$F(D) = \max \left\{ \begin{pmatrix} 0 & D \\ D & 0 \end{pmatrix} \bullet X; \quad X \text{ is symmetric}, X \succeq 0, X_{ii} = 1 \ \forall i \right\}$$

- constant-factor approximation of $\max_{I,J \subset V} \left| w(I,J) - w^*(I,J) \right|$
- $X_{i,j} \in [-1, 1]$ for each i, j



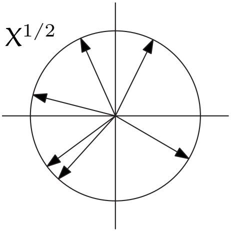
Convex objective

- input graph G^*: weights w^*, adjacency matrix A^*
- current solution G: weights w, adjacency matrix A
- let $D = A - A^*$

Grothendieck problem:

$$F(D) = \max \left\{ \begin{pmatrix} 0 & D \\ D & 0 \end{pmatrix} \bullet X; \quad X \text{ is symmetric}, X \succeq 0, X_{ii} = 1 \; \forall i \right\}$$

- constant-factor approximation of $\max_{I,J \subset V} \left| w(I,J) - w^*(I,J) \right|$
- $X_{i,j} \in [-1, 1]$ for each i, j



Properties:

- $F(D)$ is convex
- $\nabla F(D) = X^*$

Minimization problem

Optimization problem:

$$\min \left\{ F\big(A(w) - A^*\big);\ \ \sum_e w_e = m \right\}$$

- minimization of convex function
- bounded gradient: $(\nabla F(D))_{i,j} \in [-1, 1]$

Minimization problem

Optimization problem:

$$\min \left\{ F\big(A(w) - A^*\big);\ \sum_e w_e = m \right\}$$

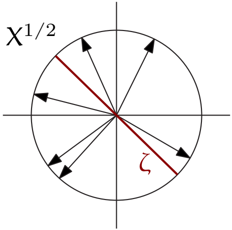
- minimization of convex function
- bounded gradient: $(\nabla F(D))_{i,j} \in [-1, 1]$

Mirror descent theorem:

- after $T = m/n$ iterations:

$$F\big(A(w) - A^*\big) \leqslant \tilde{O}(\sqrt{mn})$$

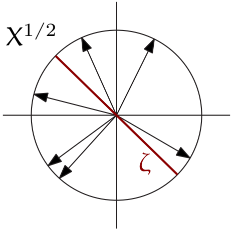
Stochastic gradient



Stochastic gradient: JL transform

- release $X^{1/2}\zeta$, where $\zeta \sim N(0, I)$
- stochastic gradient: $S_X = X^{1/2}\zeta\zeta^T X^{1/2}$
- $\mathbb{E}[S_X] = X$

Stochastic gradient



Stochastic gradient: JL transform

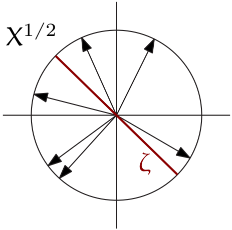
- release $X^{1/2}\zeta$, where $\zeta \sim N(0, I)$
- stochastic gradient: $S_X = X^{1/2}\zeta\zeta^T X^{1/2}$
- $\mathbb{E}[S_X] = X$

Privacy of the gradient at iteration t:

$$X = \nabla F(A(w^{(t)}) - A^*) \text{ and } \tilde{X} = \nabla F(A(w^{(t)}) - \tilde{A}^*)$$

Stochastic gradient

Stochastic gradient: JL transform



- release $X^{1/2}\zeta$, where $\zeta \sim N(0, I)$
- stochastic gradient: $S_X = X^{1/2}\zeta\zeta^T X^{1/2}$
- $\mathbb{E}[S_X] = X$

Privacy of the gradient at iteration t:

$$X = \nabla F(A(w^{(t)}) - A^*) \text{ and } \tilde{X} = \nabla F(A(w^{(t)}) - \tilde{A}^*)$$

- $X^{1/2}\zeta$ and $\tilde{X}^{1/2}\zeta$ have similar distribution:

$$\mathrm{pdf}_X(x) \leqslant e^{\epsilon_0} \cdot \mathrm{pdf}_{\tilde{X}}(x) \text{ w.p. } (1 - \delta_0)$$

$$\epsilon_0 = O(\log \frac{1}{\delta_0}) \cdot \sqrt{\mathrm{tr}\, X^{-1}(\tilde{X} - X)X^{-1}(\tilde{X} - X)}$$

- this implies (ϵ_0, δ_0)-DP

Regularization

$$F(D) = \max \left\{ \begin{pmatrix} 0 & D \\ D & 0 \end{pmatrix} \bullet X + \Psi(X); \quad X \text{ is symmetric}, X \succeq 0, X_{ii} = 1 \right\}$$

Regularization

$$F(D) = \max\left\{ \begin{pmatrix} 0 & D \\ D & 0 \end{pmatrix} \bullet X + \Psi(X); \quad X \text{ is symmetric}, X \succeq 0, X_{ii} = 1 \right\}$$

- $\Psi(X) = \lambda \log \det X$
- λ determines the stability but also error

Regularization

$$F(D) = \max\left\{ \begin{pmatrix} 0 & D \\ D & 0 \end{pmatrix} \bullet X + \Psi(X);\ \ X \text{ is symmetric}, X \succeq 0, X_{ii} = 1 \right\}$$

- $\Psi(X) = \lambda \log\det X$
- λ determines the stability but also error

Claim:

- If A^* and $\tilde{A}^*$ differ in a single edge, then

$$\sqrt{\operatorname{tr} X^{-1}(\tilde{X} - X)X^{-1}(\tilde{X} - X)} \leqslant O(1/\lambda)$$

- crucial property of Ψ: $D^2\Psi(X)[E, E] = -\lambda \operatorname{tr} X^{-1}EX^{-1}E$

Summing up

To get (ϵ, δ)**-DP:**

- we choose

$$\lambda \approx \epsilon^{-1} \sqrt{m/n}$$

Summing up

To get (ϵ, δ)-DP:

- we choose

$$\lambda \approx \epsilon^{-1}\sqrt{m/n}$$

We solve

$$F(D) = \max\left\{ \begin{pmatrix} 0 & D \\ D & 0 \end{pmatrix} \bullet X + \lambda \log\det X;\quad X \text{ symmetric PSD}, X_{ii} = 1 \right\}$$

$$\min\left\{ F\big(A - A(w)\big);\ \sum_e w_e = m \right\}$$

- using $T = m/n$ iterations of mirror descent

Summing up

To get (ϵ, δ)**-DP:**

- we choose

$$\lambda \approx \epsilon^{-1}\sqrt{m/n}$$

We solve

$$F(D) = \max\left\{ \begin{pmatrix} 0 & D \\ D & 0 \end{pmatrix} \bullet X + \lambda \log\det X; \quad X \text{ symmetric PSD}, X_{ii} = 1 \right\}$$

$$\min\left\{ F\big(A - A(w)\big); \; \sum_e w_e = m \right\}$$

- using $T = m/n$ iterations of mirror descent
- privacy (by Advanced composition thm): $\frac{1}{\lambda}\sqrt{T} = \epsilon$
- error due to low number of iterations: $\tilde{O}(\sqrt{mn})$
- error due to regularization: $\lambda n \log n \leqslant \tilde{O}(\epsilon^{-1}\sqrt{mn})$

Open problems

Matching the guarantee of the exponential mechanism

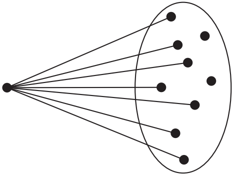
- multiplicative error $(1 + \eta)$, additive error $O(n \log n)$
- in polynomial time?

Open problems

Matching the guarantee of the exponential mechanism

- multiplicative error $(1 + \eta)$, additive error $O(n \log n)$
- in polynomial time?

Node level privacy



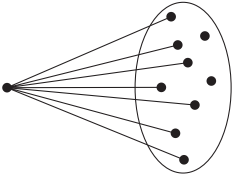
- neighboring graphs differ in whole vertex neighborhoods
- any upper or lower bounds?

Open problems

Matching the guarantee of the exponential mechanism

- multiplicative error $(1+\eta)$, additive error $O(n \log n)$
- in polynomial time?

Node level privacy



- neighboring graphs differ in whole vertex neighborhoods
- any upper or lower bounds?

Is our result implementable?

- using some convex optimization tool

Questions?

European Research Council
Established by the European Commission

https://elias.ba30.eu/