Random Projections, Graph Sparsification, and Differential Privacy

Jalaj Upadhyay

Center for Applied Cryptographic Research
University of Waterloo

December 02, 2013

The Peril of Last Talk on Monday

This Paper in One Slide

Random Projections (JL transform)

$\Downarrow$

Differential privacy

This Paper in One Slide

Graph Sparsification

$+$

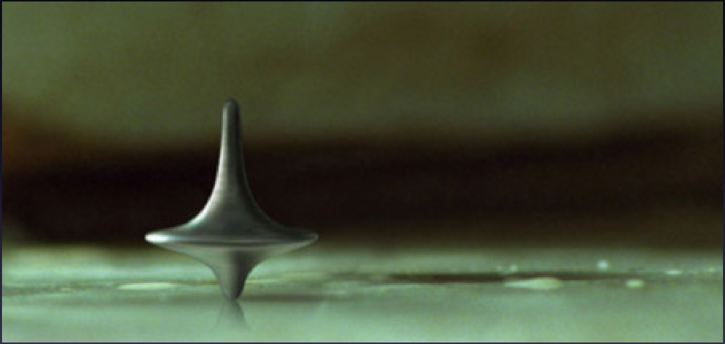
Random Projections (JL transform)

$\Downarrow$

Differential privacy

with improved sanitization time, and comparable utility and privacy guarantee

Hope We are All Not There Anymore



Differential Privacy: The Mathematical Formulation

- The idea is that absence or presence of an individual entry should not change the output “by much"

Differential Privacy: The Mathematical Formulation

- The idea is that absence or presence of an individual entry should not change the output “by much"
- A sanitization algorithm, $\mathcal{K}$, gives ϵ-differential privacy if, for all “neighboring data," D_1 and D_2, and for all range S,

$$\frac{\mathsf{Pr}\left[\mathcal{K}(D_1) \in S\right]}{\mathsf{Pr}\left[\mathcal{K}(D_2) \in S\right]} \leq \exp(\epsilon)$$

Differential Privacy: The Mathematical Formulation

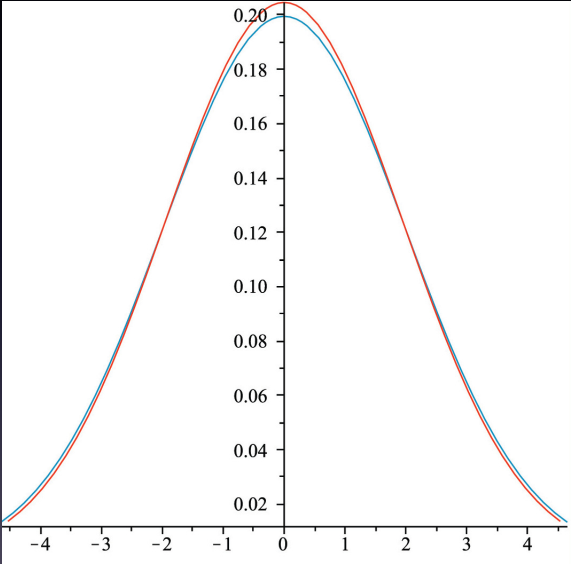
- The idea is that absence or presence of an individual entry should not change the output “by much"
- A sanitization algorithm, $\mathcal{K}$, gives ϵ-differential privacy if, for all “neighboring data," D_1 and D_2, and for all range S,

$$\frac{\Pr\left[\mathcal{K}(D_1) \in S\right]}{\Pr\left[\mathcal{K}(D_2) \in S\right]} \leq \exp(\epsilon)$$

- A sanitization algorithm, $\mathcal{K}$, gives (ϵ, δ)-differential privacy if, for all “neighboring data," D_1 and D_2, and for all range S,

$$\Pr\left[\mathcal{K}(D_1) \in S\right] \leq \exp(\epsilon)\Pr\left[\mathcal{K}(D_2) \in S\right] + \delta.$$

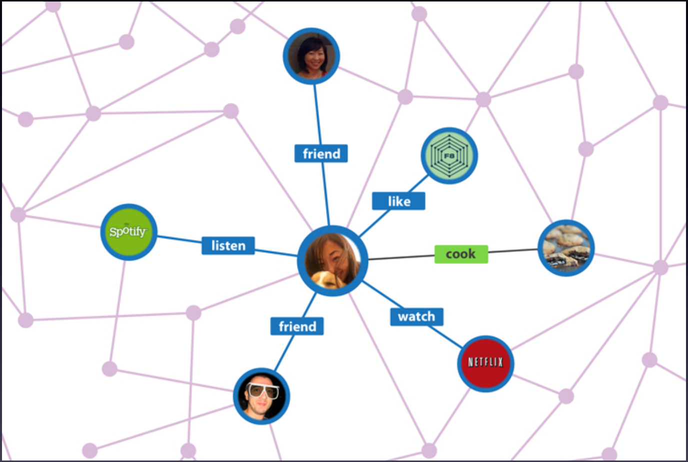
Differential Privacy: The Pretty (Common) Picture



Why Should We Care About Cut Queries?

- A natural question in social networking
- How many people have friends outside their circle?

Why Should We Care About Cut Queries?



Why Should We Care About Cut Queries?

- A natural question in social networking
- How many people have friends outside their circle?
- The answer is the number of edges crossing the border of the set of the vertices corresponding to those people
- This number is called cut corresponding to the set of vertices

Why Should We Care About Cut Queries?

- A natural question in social networking
- How many people have friends outside their circle?
- The answer is the number of edges crossing the border of the set of the vertices corresponding to those people
- This number is called cut corresponding to the set of vertices

Question: Why would you really care about the privacy?

Friendships or "What You May Call" Between People

Suppose Facebook decides to reveal the friendship graph

Friendships or "What You May Call" Between People

Suppose Facebook decides to reveal the friendship graph

There might be some people who might end up in trouble

But Spare a Thought for a Few Celebrities

But Spare a Thought for a Few Celebrities

But Spare a Thought for a Few Celebrities

But Spare a Thought for a Few Celebrities

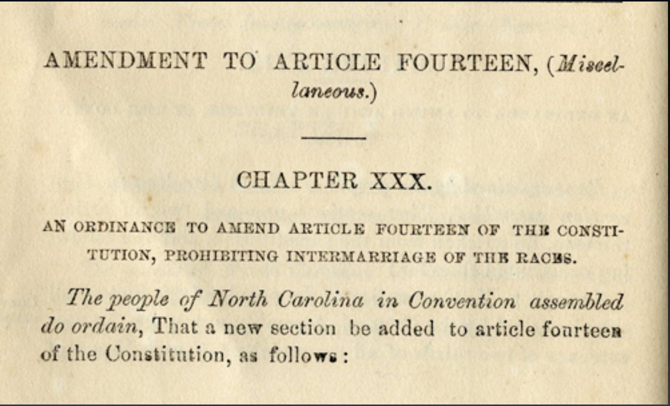
Disclaimer

The speaker does not support any of the above infidelity

None of this work should be used in any of the above cited or related scenarios

Mr. Kennedy, Mr. Clinton, or NSA did not fund this research

Scenarios Where You Can Use This Work...



AMENDMENT TO ARTICLE FOURTEEN, (*Miscellaneous.*)

CHAPTER XXX.

AN ORDINANCE TO AMEND ARTICLE FOURTEEN OF THE CONSTITUTION, PROHIBITING INTERMARRIAGE OF THE RACES.

The people of North Carolina in Convention assembled do ordain, That a new section be added to article fourteen of the Constitution, as follows:

Scenarios Where You Can Use This Work...

The Starting Point of This Work

- Blocki et al. (BBDS) showed that Johnson-Lindenstrauss (JL) transform preserves DP

The Starting Point of This Work

- Blocki et al. (BBDS) showed that Johnson-Lindenstrauss (JL) transform preserves DP
- JL transform says that using special choice of projection matrix, projecting a set of vectors to a lower dimensional space preserves their pairwise distance

The Starting Point of This Work

- Blocki et al. (BBDS) showed that Johnson-Lindenstrauss (JL) transform preserves DP
- The idea of BBDS is to use random projection of the column entries of the representative matrix

The Starting Point of This Work

- Blocki et al. (BBDS) showed that Johnson-Lindenstrauss (JL) transform preserves DP
- The idea of BBDS is to use random projection of the column entries of the representative matrix
- For a graph G, a reasonable choice is Laplacian, $L_G := D_G - A_G$

The Starting Point of This Work

- Blocki et al. (BBDS) showed that Johnson-Lindenstrauss (JL) transform preserves DP
- The idea of BBDS is to use random projection of the column entries of the representative matrix
- For a graph G, a reasonable choice is Laplacian, $L_G := D_G - A_G$
- For a set of vertices, S, $\Phi(S, \bar{S}) = \chi_S^T L_G \chi_S = \|\sqrt{L_G}\chi_S\|$

BBDS Mechanism Step by Step

The utility guarantee comes from JL-lemma

BBDS Mechanism Step by Step

The utility guarantee comes from JL-lemma

If we apply JL transform on $\sqrt{L_G}$, then

$$\Phi(S, \bar{S}) = \|M\sqrt{L_G}\chi_S\| = (1 \pm \epsilon)\|\sqrt{L_G}\chi_S\|$$

BBDS Mechanism Step by Step

The utility guarantee comes from JL-lemma

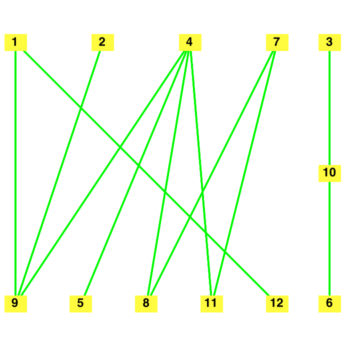
If we apply JL transform on $\sqrt{L_G}$, then

$$\Phi(S, \bar{S}) = \|M\sqrt{L_G}\chi_S\| = (1 \pm \epsilon)\|\sqrt{L_G}\chi_S\|$$

BBDS showed that it also preserves differential privacy when M is Gaussian

What about DP?

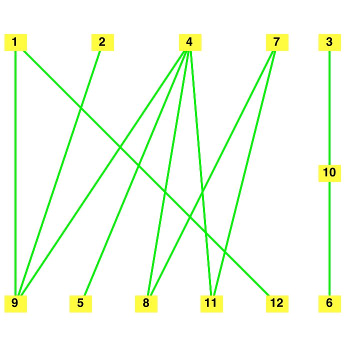
Just multiplying $\sqrt{L_G}$ by M does not give DP guarantee



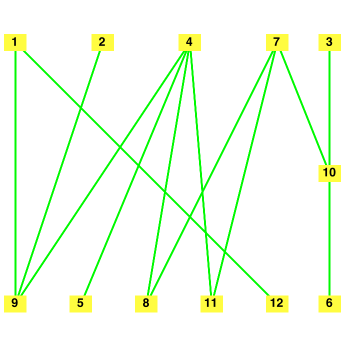
$S = \{3, 6, 10\}$ gives answer 0

What about DP?

Just multiplying $\sqrt{L_G}$ by M does not give DP guarantee

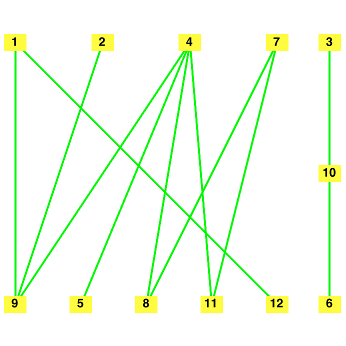


$S = \{3, 6, 10\}$ gives answer 0

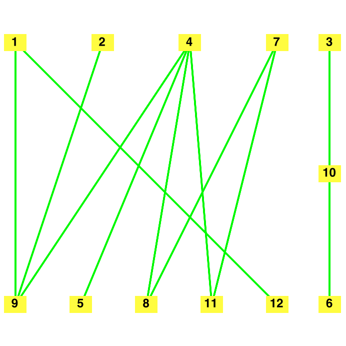


$S = \{3, 6, 10\}$ gives a non-zero answer

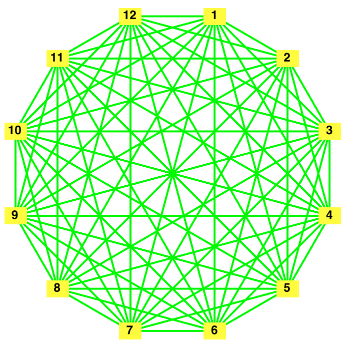
The Elegant Idea Used in BBDS



The Elegant Idea Used in BBDS



is reweighted and transformed to



This makes the graph connected and increases its second smallest eigenvalue

The Two Faces of Complete Graph

Algorithmic Disadvantage of a Complete Graph

On the negative side, overlaying a complete graph destroys any structural property of the graph

Algorithmic Disadvantage of a Complete Graph

On the negative side, overlaying a complete graph destroys any structural property of the graph

Why do we care about this?

- Most of the graphs are sparse or have some structure
- Sparsity and structure helps a lot in algorithmic design

Algorithmic Disadvantage of a Complete Graph

On the negative side, overlaying a complete graph destroys any structural property of the graph

Why do we care about this?

- Most of the graphs are sparse or have some structure
- Sparsity and structure helps a lot in algorithmic design

Question: Can we instead use a sparse graph?

Differential Privacy on Sparse Graphs

Crucial observations

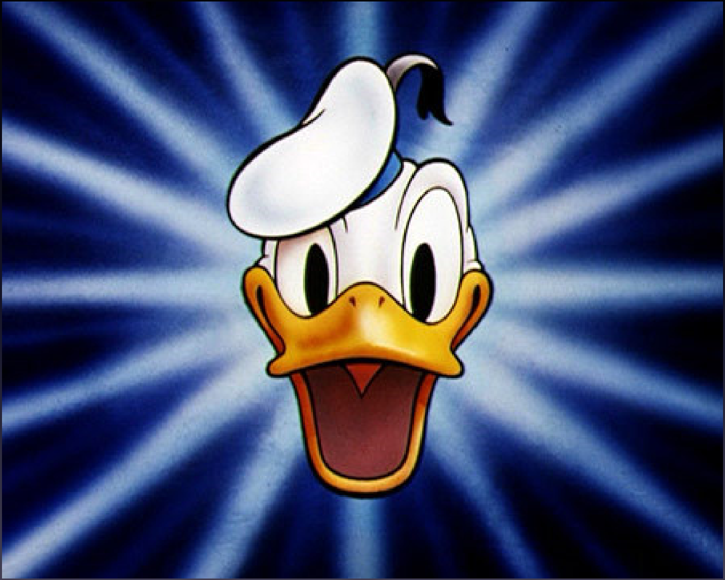
- Second smallest eigenvalue gives an estimate of connectivity (Cheeger's theorem and Fielder's result)
- Eigenvalue of a graph is at least the eigenvalue of any of its subgraph (Fielder's result)

Differential Privacy on Sparse Graphs

Crucial observations

- Second smallest eigenvalue gives an estimate of connectivity (Cheeger's theorem and Fielder's result)
- Eigenvalue of a graph is at least the eigenvalue of any of its subgraph (Fielder's result)

> An expander graph is a sparse graph with high second smallest eigenvalue



Basic Construction

Input: An n-vertices sparse graph G

- Pick a sparse expander graph, E

Basic Construction

Input: An n-vertices sparse graph G

- Pick a sparse expander graph, E
- Set $L_{\tilde{G}} = \frac{w}{d} L_E + \left(1 - \frac{w}{d}\right) L_G$

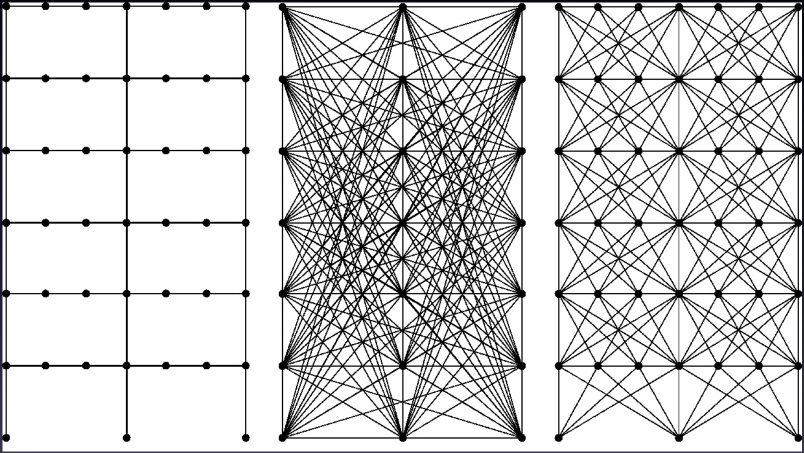
Basic Construction

Input: An n-vertices sparse graph G

- Pick a sparse expander graph, E
- Set $L_{\tilde{G}} = \frac{w}{d} L_E + \left(1 - \frac{w}{d}\right) L_G$
- Pick a random projection matrix M with Gaussian noise, and multiply with $L_{\tilde{G}}$

Utility follows by comparing the spectral property of expander with complete graph

Pictorial View of the Difference in Approaches



Original Graph

BBDS
(Not complete picture)

This Work

What About Dense Graphs?

When graph has high conductance, then apply sparsification techniques followed by random projection

Can use local sparsification techniques or Global Sparsification Techniques

What About Dense Graphs?

When graph has low conductance, overlay a high conductance graph (complete or sparse graph), and then apply sparsification techniques followed by random projections

Can use local sparsification techniques or Global Sparsification Techniques

What About Dense Graphs?

Main Lemma: The above sparsification techniques followed by JL transform that uses Gaussian matrix also preserves differential privacy

Run Time of Sanitization Algorithms

- Sparsification techniques uses time $\tilde{O}(m)$, where m is the number of edges
- For dense weighted graphs, $m = O(n^2)$, so sparsification requires time $\tilde{O}(n^2)$
- Number of entries in the Laplacian of a sparse graph is $\tilde{O}(n)$
- Multiplying the Laplacian of the graph by a Gaussian matrix takes $\tilde{O}(n^2)$
- Total run time of sanitization is $\tilde{O}(n^2)$

A Comparative Study

Abbreviations: k : total number of queries, ε : privacy parameter, n : number of vertices, δ : spectral approximation parameter, s: set of vertices in a query

Method	Noise for any k	Run Time
Randomized Response	$O(\sqrt{sn \log k/\varepsilon})$	$O(n^2)$
Exponential Sanitizer	$O(n \log n/\varepsilon)$	Intractable
Multiplicative Weight	$\tilde{O}(\sqrt{\lvert\mathcal{E}\rvert} \log k/\varepsilon)$	$O(n^2)$
JL transform	$O(s\sqrt{\log k}/\varepsilon)$	$O(rn^{2.38})$

A Comparative Study

Abbreviations: k : total number of queries, ε : privacy parameter, n : number of vertices, δ : spectral approximation parameter, s: set of vertices in a query

Method	Noise for any k	Run Time
Randomized Response	$O(\sqrt{sn \log k/\varepsilon})$	$O(n^2)$
Exponential Sanitizer	$O(n \log n/\varepsilon)$	Intractable
Multiplicative Weight	$\tilde{O}(\sqrt{\lvert\mathcal{E}\rvert} \log k/\varepsilon)$	$O(n^2)$
JL transform	$O(s\sqrt{\log k}/\varepsilon)$	$O(rn^{2.38})$
Basic Scheme	$O(s\sqrt{\log k}/\varepsilon)$	$O(n^{2+o(1)})$
Using δ-Sparsifier	$O(s\delta\sqrt{\log k}/\varepsilon)$	$O(n^{2+o(1)})$

Conclusion

- In this talk, we showed an algorithmic improvement over the sanitization techniques
- We achieve the best of both the world: efficient sanitization and almost the same privacy and utility guarantee

Conclusion

- In this talk, we showed an algorithmic improvement over the sanitization techniques
- We achieve the best of both the world: efficient sanitization and almost the same privacy and utility guarantee

We also do the following in the paper:

- A combinatorial analysis to answer (S, T)-cut queries
- Further optimization: Fast-JL transform of Ailon-Chazelle preserves differential privacy