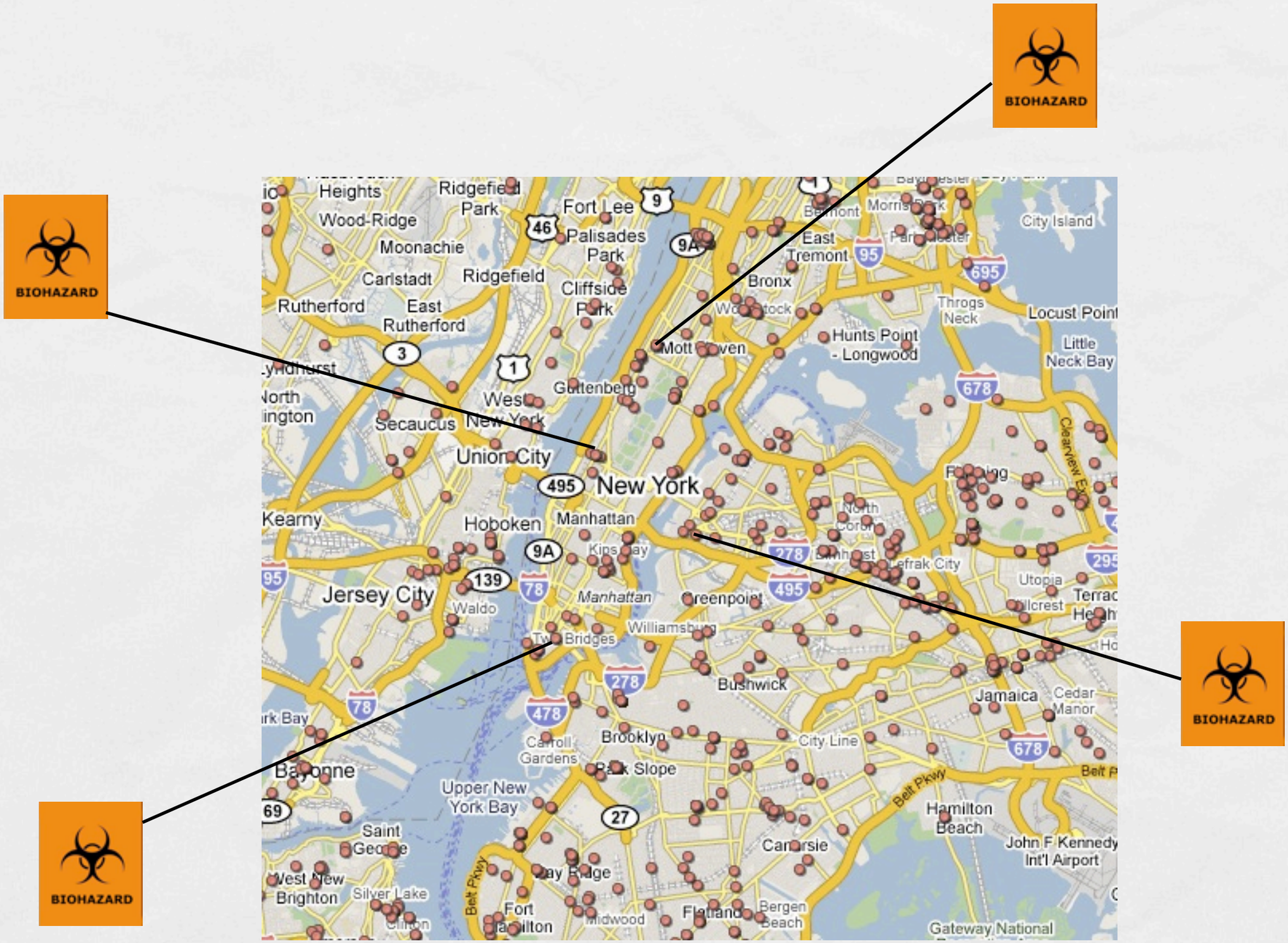
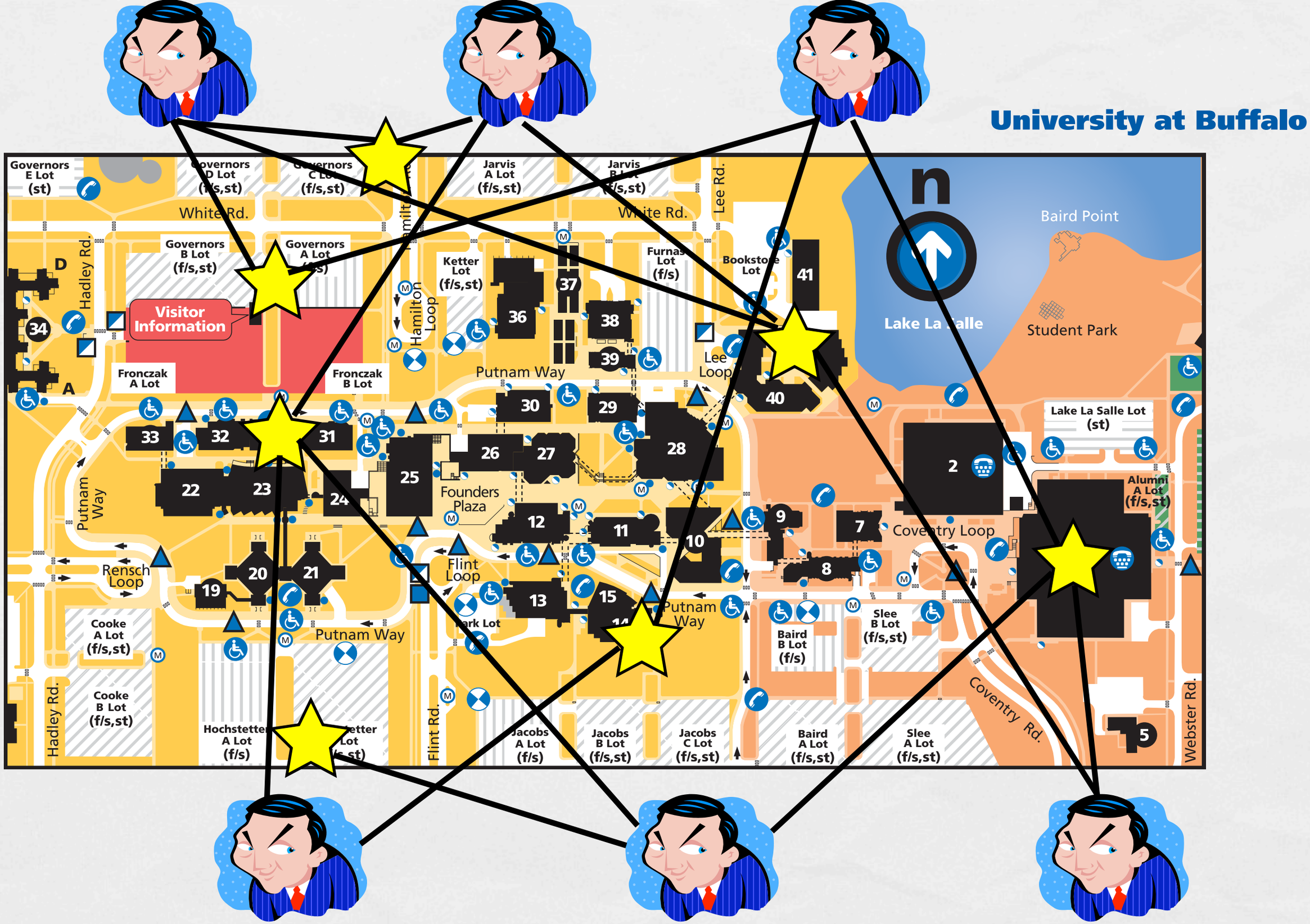
Differentially Private Approximation Algorithms

Katrina Ligett
Cornell University (work done while visiting MSR)

joint work with
Anupam Gupta, Aaron Roth (CMU)
Frank McSherry, Kunal Talwar (Microsoft Research)
to appear, SODA 2010

facebook


BIOHAZARD
BIOHAZARD
BIOHAZARD
BIOHAZARD
Heights
Ridgefield Park
Fort Lee
Wood-Ridge
Palisades Park
Moonachie
Carlstadt
Ridgefield
Cliffside Park
Rutherford
East Rutherford
Bronx
East Tremont
City Island
Throgs Neck
Locust Point
Hunts Point - Longwood
Little Neck Bay
Mott Haven
Guttenberg
West New York
Secaucus
Union City
New York
Kearny
Hoboken
Manhattan
Jersey City
Waldo
Williamsburg
Greenpoint
Bushwick
Jamaica
Cedar Manor
Brooklyn
Carroll Gardens
Park Slope
City Line
Bayonne
Upper New York Bay
Hamilton Beach
John F Kennedy Int'l Airport
Saint George
West New Brighton
Silver Lake
Bay Ridge
Fort Hamilton
Midwood
Flatlands
Bergen Beach
Canarsie
Gateway National
Utopia
Belt Pkwy
Clearview Exp
46
9
9A
95
695
3
1
678
495
139
78
278
478
27
69


University at Buffalo
Governors E Lot (st)
Governors D Lot (f/s,st)
Governors C Lot (f/s,st)
Jarvis A Lot (f/s,st)
Jarvis B Lot (f/s,st)
White Rd.
Lee Rd.
Governors B Lot (f/s,st)
Governors A Lot (f/s,st)
Hadley Rd.
Visitor Information
Hamilton Loop
Ketter Lot (f/s,st)
Furnas Lot (f/s)
Bookstore Lot
Baird Point
Lake La Salle
Student Park
Fronczak A Lot
Fronczak B Lot
Putnam Way
Lee Loop
Lake La Salle Lot (st)
Alumni A Lot (f/s,st)
Founders Plaza
Coventry Loop
Rensch Loop
Flint Loop
Cooke A Lot (f/s,st)
Cooke B Lot (f/s,st)
Hochstetter A Lot (f/s)
Park Lot
Slee B Lot (f/s,st)
Baird B Lot (f/s)
Coventry Rd.
Webster Rd.
Flint Rd.
Jacobs A Lot (f/s)
Jacobs B Lot (f/s,st)
Jacobs C Lot (f/s,st)
Baird A Lot (f/s,st)
Slee A Lot (f/s,st)

mincut

k-median

set cover

mincut

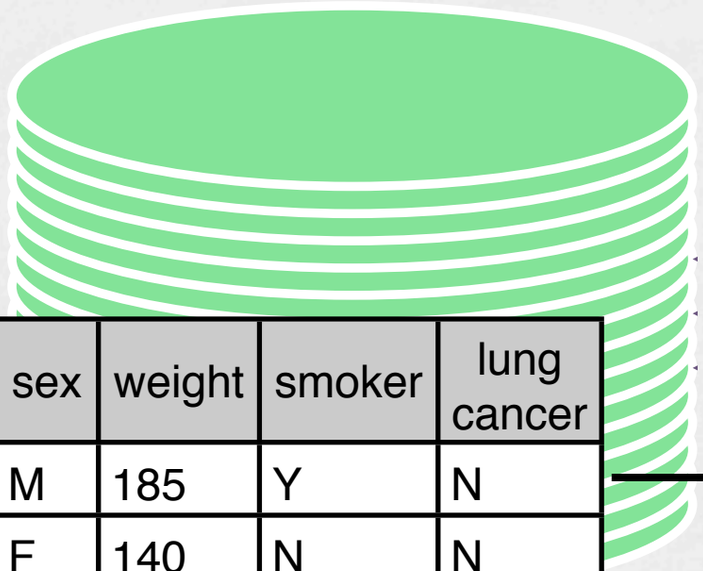
k-median

set cover

- What are the challenges and tradeoffs for useful, private, efficient combinatorial optimization?
- How does approximation for efficiency interact with approximation for privacy?

Database Privacy: The Model

- Database D a set of rows, one per person
- Sanitizing algorithm A probabilistically maps D to event or object



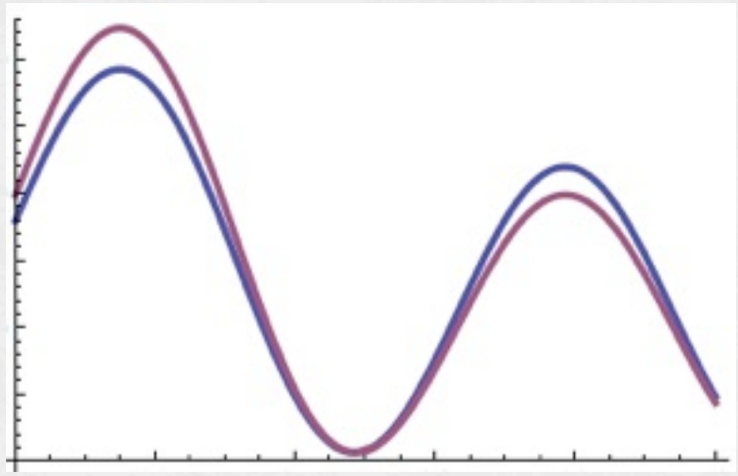
name	DOB	sex	weight	smoker	lung cancer
John Doe	12/1/51	M	185	Y	N
Jane Smith	3/3/46	F	140	N	N
Ellen Jones	4/24/59	F	160	Y	Y
Jennifer Kim	3/1/70	F	135	N	N
Rachel Waters	9/5/43	F	140	N	N

A

Differential Privacy [DwMcNiSm06]

ε-Differential Privacy for mechanism A: Changing a single element in the input doesn't change the probability of obtaining an outcome in any S by much.

For two neighboring data sets D_1, D_2:

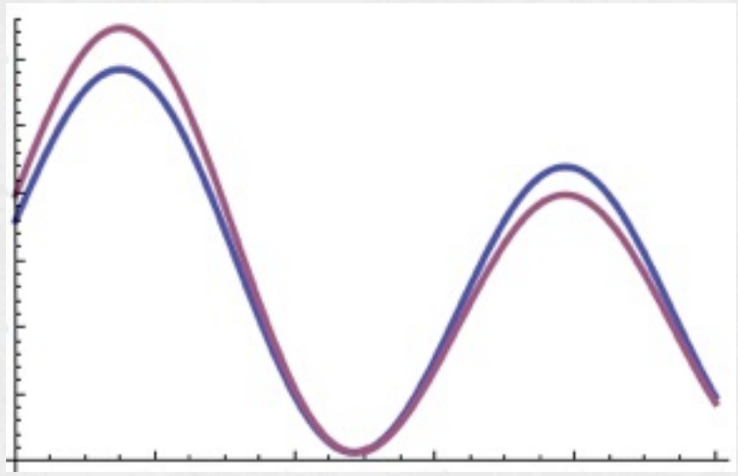


Differential Privacy [DwMcNiSm06]

ε-Differential Privacy for mechanism A: Changing a single element in the input doesn't change the probability of obtaining an outcome in any S by much.

For two neighboring data sets D_1, D_2:

$$\Pr[A(D_1) \in S] \leq e^{\varepsilon} \Pr[A(D_2) \in S]$$

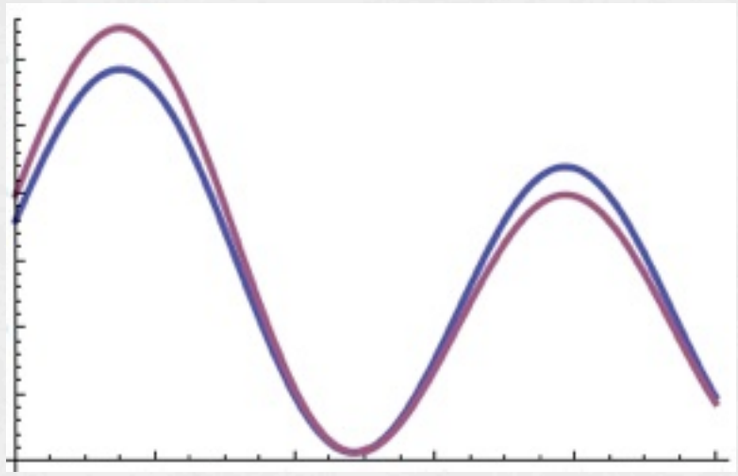


Differential Privacy [DwMcNiSm06]

ε-Differential Privacy for mechanism A: Changing a single element in the input doesn't change the probability of obtaining an outcome in any S by much.

For two neighboring data sets D_1, D_2:

$$\Pr[A(D_1) \in S] \leq e^{\varepsilon} \Pr[A(D_2) \in S]$$



Nobody should be able to learn anything more if you're in the database than if you're not

Differential Privacy [DwMcNiSm06]

- Gives provable privacy guarantee to each individual

- Is a statistical property of *mechanism* behavior
  - unaffected by auxiliary information
  - independent of adversary's computational power

Differential Privacy - (some) Previous Work

- Numerical function evaluation - positive and negative results [DN03, DMNS06, DMT07, NRS07]

- Learning
  - Can privately learn in the SQ-model. [BDMN05]
  - Can privately learn* in the PAC-model. [KLNRS08]
  - Can output* a private data set useful for learning in the PAC model. [BLR08]
  - Info-theoretic vs. efficient separation. [DNRRV09]

Differential Privacy - (some) Previous Work

- "Exponential mechanism" [MT07] allows DP selection for set of discrete options, but
  - may give poor utility guarantees (e.g., min cut)
  - inefficient when set of candidate outcomes is exponential

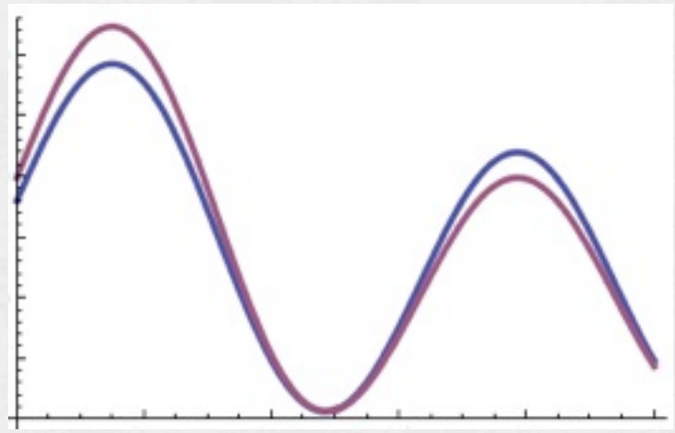
Differential Privacy - (some) Previous Work

- "Exponential mechanism" [MT07] allows DP selection for set of discrete options, but
  - may give poor utility guarantees (e.g., min cut)
  - inefficient when set of candidate outcomes is exponential

- Relatively little work on complex outputs like synthetic databases, coverings, partitions; [FFKN09] an exception

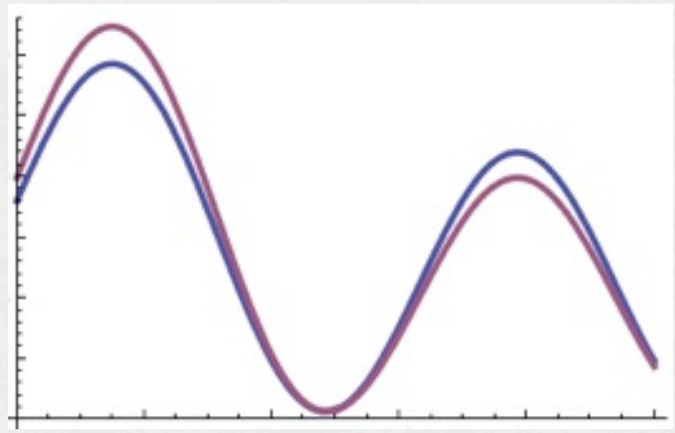
A Tool: Additive Noise [DwMcNiSm06]

- Recall, DP requires probability of any output to be multiplicatively close under any two neighboring databases



A Tool: Additive Noise [DwMcNiSm06]

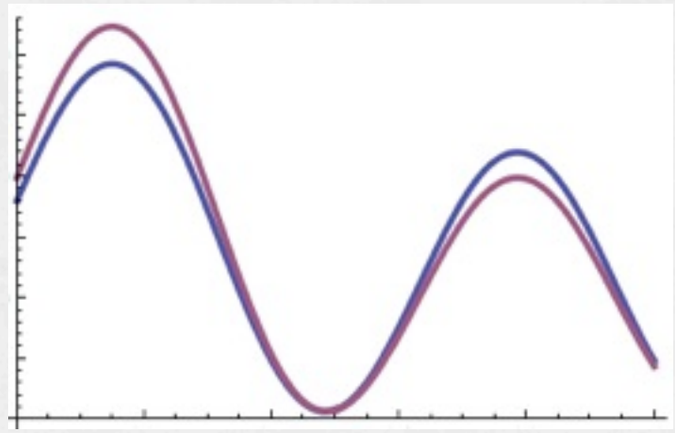
- Recall, DP requires probability of any output to be multiplicatively close under any two neighboring databases



- Idea: if output is a number, add noise

A Tool: Additive Noise [DwMcNiSm06]

- Recall, DP requires probability of any output to be multiplicatively close under any two neighboring databases

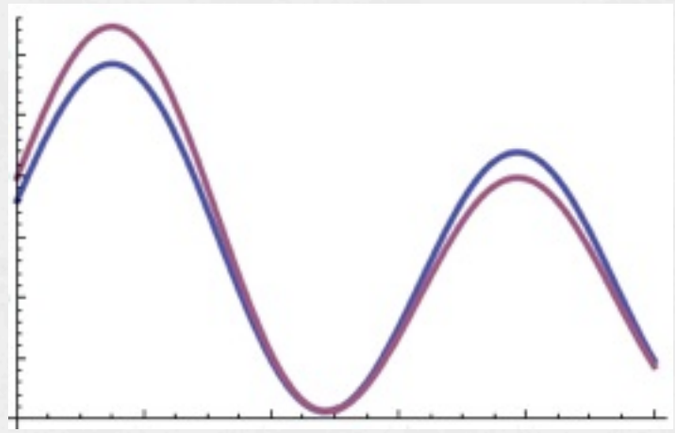


- Idea: if output is a number, add noise

- Query's **sensitivity** measures difference on neighboring databases

A Tool: Additive Noise [DwMcNiSm06]

- Recall, DP requires probability of any output to be multiplicatively close under any two neighboring databases

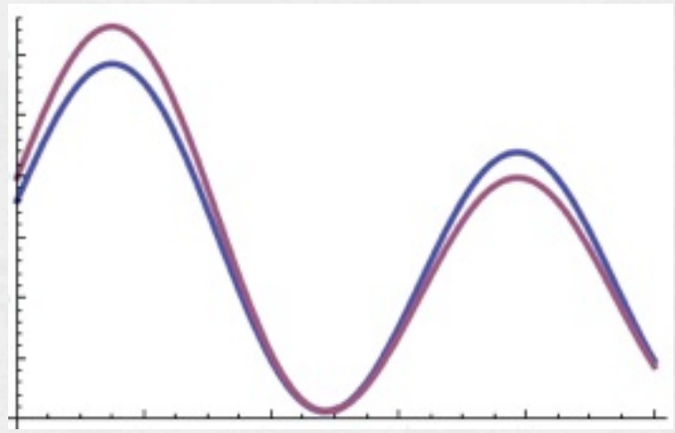


- Idea: if output is a number, add noise
- Query's **sensitivity** measures difference on neighboring databases

$$\Delta Q = \max_{\{D1,\, D2\}} ||Q(D_1) - Q(D_2)||_1$$

A Tool: Additive Noise [DwMcNiSm06]

- Recall, DP requires probability of any output to be multiplicatively close under any two neighboring databases

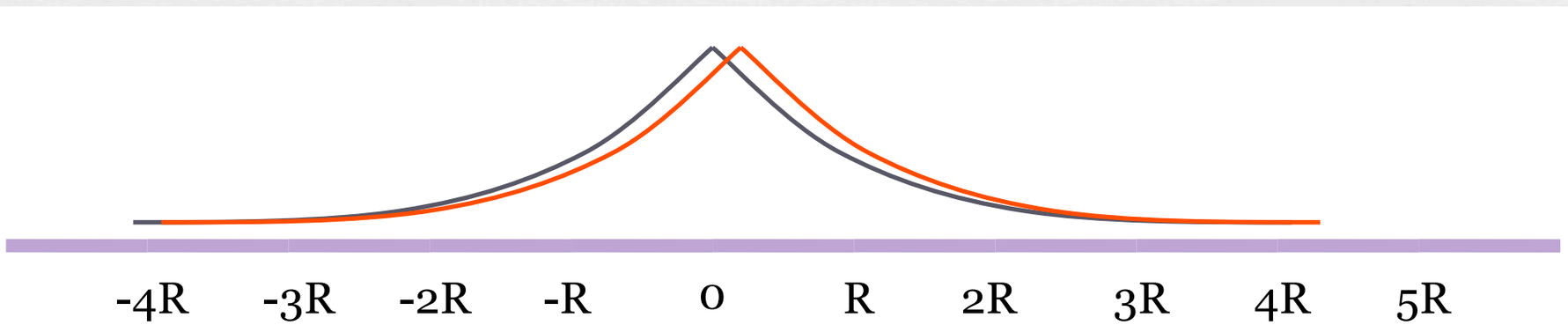


- Idea: if output is a number, add noise

- Query's **sensitivity** measures difference on neighboring databases

$$\Delta Q = \max_{\{D1, D2\}} ||Q(D_1) - Q(D_2)||_1$$

- Adding scaled symmetric noise Lap(|x|/R) with $R = \Delta Q/\varepsilon$ gives DP



$$\frac{\Pr[\,Q(A(D_1)) = t]}{\Pr[\,Q(A(D_2)) = t]} \leq \exp(-\Delta f/R)$$

EXPONENTIAL MECHANSIM [MT07]

- Algorithm:
  - Define a score function on any input D, output S
  - Output each possible S with probability proportional to
    exp(α score(D,S) / (2 sensitivity(score)))

- Preserves α-differential privacy

- (Must then show utility)

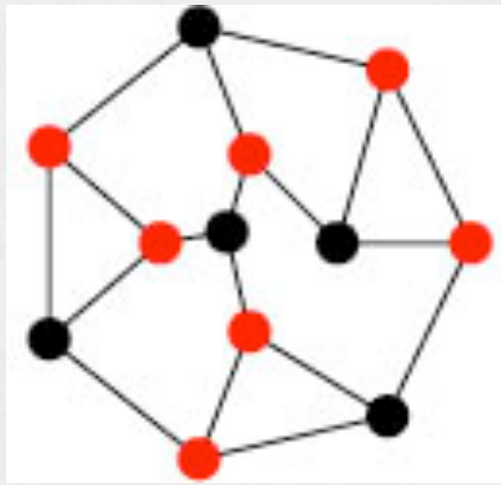
	Non-Private, Efficient Algorithm	Private, Efficient Algorithm	Private, Information Theoretic Algs w/ matching LBs
Vertex Cover	2 x OPT [Pitt85]	(2 + 16/ε) x OPT	Θ(1/ε) x OPT
Weighted Vertex Cover	2 x OPT [Hochbaum82]	(16 + 16/ε) x OPT	Θ(1/ε) x OPT
Set Cover	ln(n) x OPT [Johnson74]	O(ln(n) + ln(m)/ε) x OPT*	Θ(ln (m)/ε) x OPT
Weighted Set Cover	ln(n) x OPT [Chvatal79]	O(ln(n)(ln m + lnln n)/ε) x OPT*	Θ(ln (m)/ε) x OPT
Min Cut	OPT [Ford-Fulkerson56]	OPT + O(ln(n)/ε)*	OPT + Θ(ln(n)/ε)
CPPP	(1-1/e) x OPT [Nemhauser-Wolsey-Fisher78]	(1-1/e) x OPT – O(k ln(m)/ε)*	OPT-Θ(k ln(n/k)/ε)
k-Median	(3 + ε) x OPT [Arya et al. 04]	6 x OPT + O(k^2 $\ln^2$(n/ε))	OPT+Θ(k ln(n/k)/ε)

	Non-Private, Efficient Algorithm	Private, Efficient Algorithm	Private, Information Theoretic Algs w/ matching LBs
Vertex Cover	2 x OPT [Pitt85]	(2 + 16/ε) x OPT	Θ(1/ε) x OPT
Weighted Vertex Cover	2 x OPT [Hochbaum82]	(16 + 16/ε) x OPT	Θ(1/ε) x OPT
Set Cover	ln(n) x OPT [Johnson74]	O(ln(n) + ln(m)/ε) x OPT*	Θ(ln (m)/ε) x OPT
Weighted Set Cover	ln(n) x OPT [Chvatal79]	O(ln(n)(ln m + lnln n)/ε) x OPT*	Θ(ln (m)/ε) x OPT
Min Cut	OPT [Ford-Fulkerson56]	OPT + O(ln(n)/ε)*	OPT + Θ(ln(n)/ε)
CPPP	(1-1/e) x OPT [Nemhauser-Wolsey-Fisher78]	(1-1/e) x OPT – O(k ln(m)/ε)*	OPT-Θ(k ln(n/k)/ε)
k-Median	(3 + ε) x OPT [Arya et al. 04]	6 x OPT + O(k^2 $\ln^2$(n/ε))	OPT+Θ(k ln(n/k)/ε)

VERTEX COVER

- Select a small set of the n vertices to cover the existing edges

- An edge's presence or absence is private information



VERTEX COVER - VALUE

- Computing even the value of vertex cover to within $n^{1-\varepsilon}$ impossible under “functional privacy” [HKKN01]

VERTEX COVER - VALUE

- Computing even the value of vertex cover to within $n^{1-\varepsilon}$ impossible under "functional privacy" [HKKN01]

- Can immediately get *differentially* private 2-approx of VC value (add Laplace noise to non-private 2-approx) [DMNS06]

VERTEX COVER - VALUE

- Computing even the value of vertex cover to within $n^{1-\varepsilon}$ impossible under "functional privacy" [HKKN01]

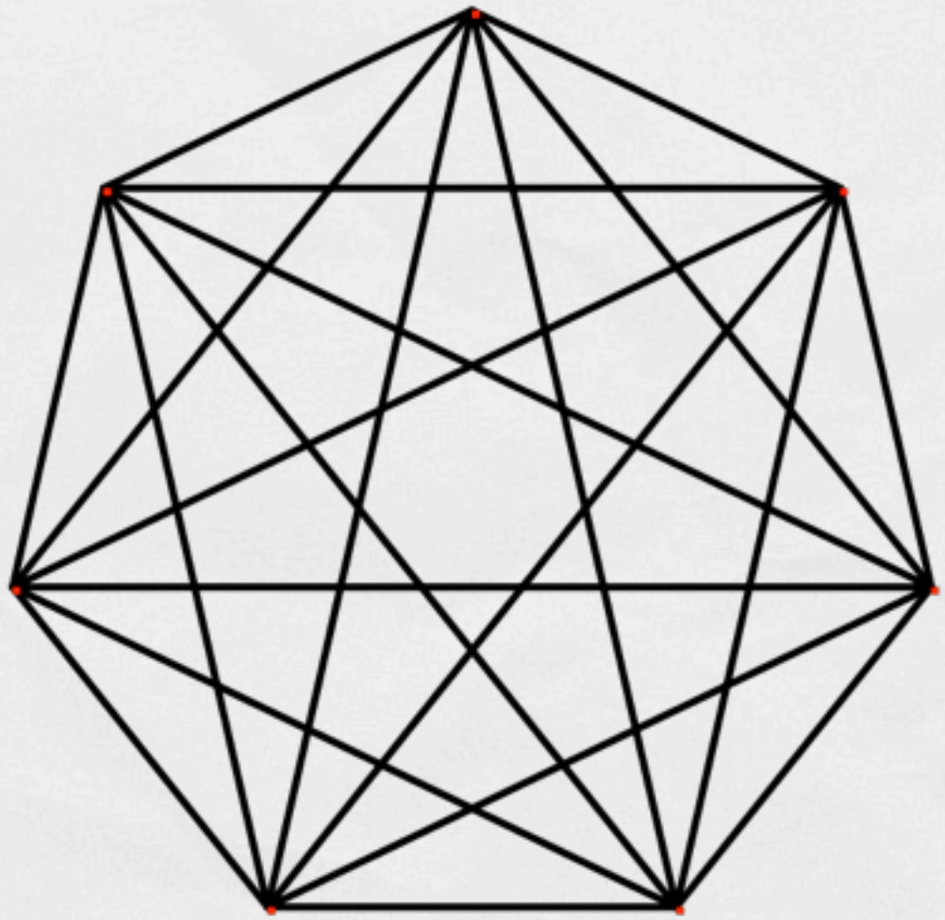
- Can immediately get *differentially* private 2-approx of VC value (add Laplace noise to non-private 2-approx) [DMNS06]

- Orthogonal definitions.

VERTEX COVER - SEARCH

- Obstacle: (private) data not only influences objective function, but imposes hard constraints on feasible solutions.

- Impossible to output explicit cover while preserving DP: any DP algorithm must output n–1 vertices



Output Representation

- Solution: allow algorithm to output an *orientation* for each of the possible edges; implicitly defines a cover for any subset of edges, without needing to reveal the subset itself.

- Output tells each agent which drop-off location to use. An arbitrary set of agents can collude and still not be able to infer the presence of any other agent from the algorithm's output.

- A permutation of the vertices implies such an orientation.

Vertex Cover

We adapt the 2-approximation algorithm of Pitt: Select an uncovered edge; select one of its endpoints

1: **let** $n \leftarrow |V|$, $V_1 \leftarrow V, E_1 \leftarrow E$.
2: **for** $i = 1, 2, \ldots, n$ **do**
3: **let** $w_i \leftarrow (4/\epsilon) \times \sqrt{n/(n-i+1)}$.
4: **pick** a vertex $v \in V_i$ with probability proportional to $d_{E_i}(v) + w_i$.
5: **output** v. **let** $V_{i+1} \leftarrow V_i \setminus \{v\}$, $E_{i+1} \leftarrow E_i \setminus (\{v\} \times V_i)$.
6: **end for**

1: **let** $n \leftarrow |V|$, $V_1 \leftarrow V, E_1 \leftarrow E$.
2: **for** $i = 1, 2, \ldots, n$ **do**
3: **let** $w_i \leftarrow (4/\epsilon) \times \sqrt{n/(n-i+1)}$.
4: **pick** a vertex $v \in V_i$ with probability proportional to $d_{E_i}(v) + w_i$.
5: **output** v. **let** $V_{i+1} \leftarrow V_i \setminus \{v\}$, $E_{i+1} \leftarrow E_i \setminus (\{v\} \times V_i)$.
6: **end for**

Theorem: The algorithm preserves $\sum_i 2/iw_i \leq \epsilon$ -differential privacy

1: **let** $n \leftarrow |V|$, $V_1 \leftarrow V, E_1 \leftarrow E$.
2: **for** $i = 1, 2, \ldots, n$ **do**
3: **let** $w_i \leftarrow (4/\epsilon) \times \sqrt{n/(n-i+1)}$.
4: **pick** a vertex $v \in V_i$ with probability proportional to $d_{E_i}(v) + w_i$.
5: **output** v. **let** $V_{i+1} \leftarrow V_i \setminus \{v\}$, $E_{i+1} \leftarrow E_i \setminus (\{v\} \times V_i)$.
6: **end for**

Theorem: The algorithm preserves $\sum_i 2/iw_i \leq \epsilon$ -differential privacy

Proof:

Let instance A and instance B differ in a single edge: B = A + {e}

1: **let** $n \leftarrow |V|$, $V_1 \leftarrow V, E_1 \leftarrow E$.
2: **for** $i = 1, 2, \ldots, n$ **do**
3: **let** $w_i \leftarrow (4/\epsilon) \times \sqrt{n/(n-i+1)}$.
4: **pick** a vertex $v \in V_i$ with probability proportional to $d_{E_i}(v) + w_i$.
5: **output** v. **let** $V_{i+1} \leftarrow V_i \setminus \{v\}$, $E_{i+1} \leftarrow E_i \setminus (\{v\} \times V_i)$.
6: **end for**

Theorem: The algorithm preserves $\sum_i 2/iw_i \leq \epsilon$ -differential privacy

Proof:

Let instance A and instance B differ in a single edge: B = A + {e}

$$\frac{\mathbf{Pr}[ALG(A) = \pi]}{\mathbf{Pr}[ALG(B) = \pi]} = \prod_{i=1}^{n} \frac{(w_i + d_i(A))/((n-i+1)w_i + 2m_i(A))}{(w_i + d_i(B))/((n-i+1)w_i + 2m_i(B))}$$

1: **let** $n \leftarrow |V|$, $V_1 \leftarrow V, E_1 \leftarrow E$.
2: **for** $i = 1, 2, \ldots, n$ **do**
3: **let** $w_i \leftarrow (4/\epsilon) \times \sqrt{n/(n-i+1)}$.
4: **pick** a vertex $v \in V_i$ with probability proportional to $d_{E_i}(v) + w_i$.
5: **output** v. **let** $V_{i+1} \leftarrow V_i \setminus \{v\}$, $E_{i+1} \leftarrow E_i \setminus (\{v\} \times V_i)$.
6: **end for**

Theorem: The algorithm preserves $\sum_i 2/iw_i \leq \epsilon$ -differential privacy

Proof:

Let instance A and instance B differ in a single edge: B = A + {e}

$$\frac{\mathbf{Pr}[ALG(A) = \pi]}{\mathbf{Pr}[ALG(B) = \pi]} = \prod_{i=1}^{n} \frac{(w_i + d_i(A))/((n-i+1)w_i + 2m_i(A))}{(w_i + d_i(B))/((n-i+1)w_i + 2m_i(B))}$$

Suppose π_j is the first vertex output incident to e

Since $d_i(A) = d_i(B)$ for all $i>j$…

1: **let** $n \leftarrow |V|$, $V_1 \leftarrow V, E_1 \leftarrow E$.
2: **for** $i = 1, 2, \ldots, n$ **do**
3: **let** $w_i \leftarrow (4/\epsilon) \times \sqrt{n/(n-i+1)}$.
4: **pick** a vertex $v \in V_i$ with probability proportional to $d_{E_i}(v) + w_i$.
5: **output** v. **let** $V_{i+1} \leftarrow V_i \setminus \{v\}$, $E_{i+1} \leftarrow E_i \setminus (\{v\} \times V_i)$.
6: **end for**

Theorem: The algorithm preserves $\sum_i 2/iw_i \leq \epsilon$ -differential privacy

Proof:
Let instance A and instance B differ in a single edge: B = A + {e}

$$\frac{\mathbf{Pr}[ALG(A) = \pi]}{\mathbf{Pr}[ALG(B) = \pi]} = \frac{w_j + d_j(A)}{w_j + d_j(B)} \times \prod_{i \leq j} \frac{(n-i+1)w_i + 2m_i(B)}{(n-i+1)w_i + 2m_i(A)}$$

1: **let** $n \leftarrow |V|$, $V_1 \leftarrow V, E_1 \leftarrow E$.
2: **for** $i = 1, 2, \ldots, n$ **do**
3: **let** $w_i \leftarrow (4/\epsilon) \times \sqrt{n/(n-i+1)}$.
4: **pick** a vertex $v \in V_i$ with probability proportional to $d_{E_i}(v) + w_i$.
5: **output** v. **let** $V_{i+1} \leftarrow V_i \setminus \{v\}$, $E_{i+1} \leftarrow E_i \setminus (\{v\} \times V_i)$.
6: **end for**

Theorem: The algorithm preserves $\sum_i 2/iw_i \leq \epsilon$ -differential privacy

Proof:

Let instance A and instance B differ in a single edge: B = A + {e}

$$\frac{\mathbf{Pr}[ALG(A) = \pi]}{\mathbf{Pr}[ALG(B) = \pi]} = \frac{w_j + d_j(A)}{w_j + d_j(B)} \times \prod_{i \leq j} \frac{(n-i+1)w_i + 2m_i(B)}{(n-i+1)w_i + 2m_i(A)}$$

$m_i(B) = m_i(A) + 1$. The ratio is maximized if $m_i(A) = 0$.

1: **let** $n \leftarrow |V|$, $V_1 \leftarrow V, E_1 \leftarrow E$.
2: **for** $i = 1, 2, \ldots, n$ **do**
3: **let** $w_i \leftarrow (4/\epsilon) \times \sqrt{n/(n-i+1)}$.
4: **pick** a vertex $v \in V_i$ with probability proportional to $d_{E_i}(v) + w_i$.
5: **output** v. **let** $V_{i+1} \leftarrow V_i \setminus \{v\}$, $E_{i+1} \leftarrow E_i \setminus (\{v\} \times V_i)$.
6: **end for**

Theorem: The algorithm preserves $\sum_i 2/iw_i \leq \epsilon$ -differential privacy

Proof:

Let instance A and instance B differ in a single edge: B = A + {e}

$$\frac{\mathbf{Pr}[ALG(A) = \pi]}{\mathbf{Pr}[ALG(B) = \pi]} = \frac{w_j + d_j(A)}{w_j + d_j(B)} \times \prod_{i \leq j} \frac{(n-i+1)w_i + 2m_i(B)}{(n-i+1)w_i + 2m_i(A)}$$

$m_i(B) = m_i(A) + 1$. The ratio is maximized if $m_i(A) = 0$.

$$\prod_{i \leq j} \frac{(n-i+1)w_i + 2m_i(B)}{(n-i+1)w_i + 2m_i(A)} \leq \prod_{i \leq j} \frac{(n-i+1)w_i + 2}{(n-i+1)w_i + 0}$$

1: **let** $n \leftarrow |V|$, $V_1 \leftarrow V, E_1 \leftarrow E$.
2: **for** $i = 1, 2, \ldots, n$ **do**
3: **let** $w_i \leftarrow (4/\epsilon) \times \sqrt{n/(n-i+1)}$.
4: **pick** a vertex $v \in V_i$ with probability proportional to $d_{E_i}(v) + w_i$.
5: **output** v. **let** $V_{i+1} \leftarrow V_i \setminus \{v\}$, $E_{i+1} \leftarrow E_i \setminus (\{v\} \times V_i)$.
6: **end for**

Theorem: The algorithm preserves $\sum_i 2/iw_i \leq \epsilon$ -differential privacy

Proof:
Let instance A and instance B differ in a single edge: B = A + {e}

$$\frac{\mathbf{Pr}[ALG(A) = \pi]}{\mathbf{Pr}[ALG(B) = \pi]}$$

$$\leq \prod_{i \leq j} \frac{(n-i+1)w_i + 2}{(n-i+1)w_i + 0} = \prod_{i \leq j} \left(1 + \frac{2}{(n-i+1)w_i}\right) \leq \exp\left(\sum_{i \leq j} \frac{2}{(n-i+1)w_i}\right)$$

using $1 + x \leq \exp(x)$

The w_i are chosen so that $\sum_i 2/(n-i+1)w_i = (\varepsilon/\sqrt{n}) \sum_i 1/2\sqrt{i}$ is at most ε.

1: **let** $n \leftarrow |V|$, $V_1 \leftarrow V, E_1 \leftarrow E$.
2: **for** $i = 1, 2, \ldots, n$ **do**
3: **let** $w_i \leftarrow (4/\epsilon) \times \sqrt{n/(n-i+1)}$.
4: **pick** a vertex $v \in V_i$ with probability proportional to $d_{E_i}(v) + w_i$.
5: **output** v. **let** $V_{i+1} \leftarrow V_i \setminus \{v\}$, $E_{i+1} \leftarrow E_i \setminus (\{v\} \times V_i)$.
6: **end for**

Theorem: For all G: $\mathbb{E}[ALG(G)] \leq \left(2 + 2avg_{i \leq n}(w_i)\right) OPT(G)$

$\leq (2 + 16/\epsilon) OPT(G)$

Proof: by induction. (Omitted)

Vertex Cover - Information Theoretic LB

Consider graph with $1/(2\varepsilon)$ vertices

Vertex Cover - Information Theoretic LB

Consider graph with $1/(2\varepsilon)$ vertices

- DP algorithm on empty graph must put u before v with probab $1/2$

Vertex Cover - Information Theoretic LB

Consider graph with $1/(2\varepsilon)$ vertices

- DP algorithm on empty graph must put u before v with probab $1/2$
- Star graph rooted at u has at most $1/\varepsilon$ more edges, so u is output before v with proab at most $1/2e$

Vertex Cover - Information Theoretic LB

Consider graph with $1/(2\varepsilon)$ vertices

- DP algorithm on empty graph must put u before v with probab $1/2$
- Star graph rooted at u has at most $1/\varepsilon$ more edges, so u is output before v with proab at most $1/2e$
- Expected cost of DP algorithm is $\sim 1/\varepsilon$; OPT is 1

WEIGHTED VERTEX COVER

$$s(v) = \frac{1}{W(v)} \qquad s((u,v)) = s(u) + s(v) \qquad V_j = \{v : W(v) = 2^j\}$$

Input: $G = (V, E), W: V \to \mathbb{R}$ **Output:** $\pi = (\pi_1, \ldots, \pi_n)$

1.while not all vertices have been output **do**:

2. **pick** an uncovered edge e with probability proportional to $s(e)$
3. **output** endpoint $u \in e$ with probability proportional to $s(u)$
4. **while** there exists some weight class V_j such that the number of nodes of class j or higher that we've output is at least $V_j/2$ **do**:
5. **pick** the smallest such value j
6. **output** all remaining vertices in V_j in random order.
7. **end while**

8.end while

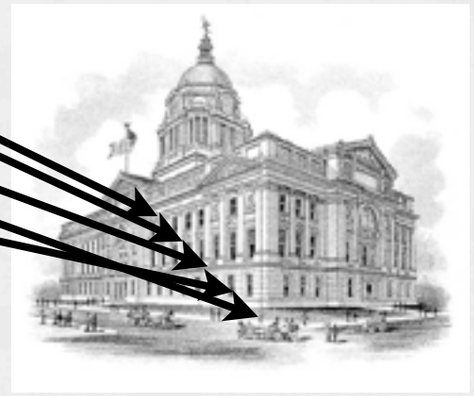
	Non-Private, Efficient Algorithm	Private, Efficient Algorithm	Private, Information Theoretic Algs w/ matching LBs
Vertex Cover	2 x OPT [Pitt85]	(2 + 16/ε) x OPT	Θ(1/ε) x OPT
Weighted Vertex Cover	2 x OPT [Hochbaum82]	(16 + 16/ε) x OPT	Θ(1/ε) x OPT
Set Cover	ln(n) x OPT [Johnson74]	O(ln(n) + ln(m)/ε) x OPT*	Θ(ln (m)/ε) x OPT
Weighted Set Cover	ln(n) x OPT [Chvatal79]	O(ln(n)(ln m + lnln n)/ε) x OPT*	Θ(ln (m)/ε) x OPT
Min Cut	OPT [Ford-Fulkerson56]	OPT + O(ln(n)/ε)*	OPT + Θ(ln(n)/ε)
CPPP	(1-1/e) x OPT [Nemhauser-Wolsey-Fisher78]	(1-1/e) x OPT – O(k ln(m)/ε)*	OPT-Θ(k ln(n/k)/ε)
k-Median	(3 + ε) x OPT [Arya et al. 04]	6 x OPT + O(k^2 $\ln^2$(n/ε))	OPT+Θ(k ln(n/k)/ε)

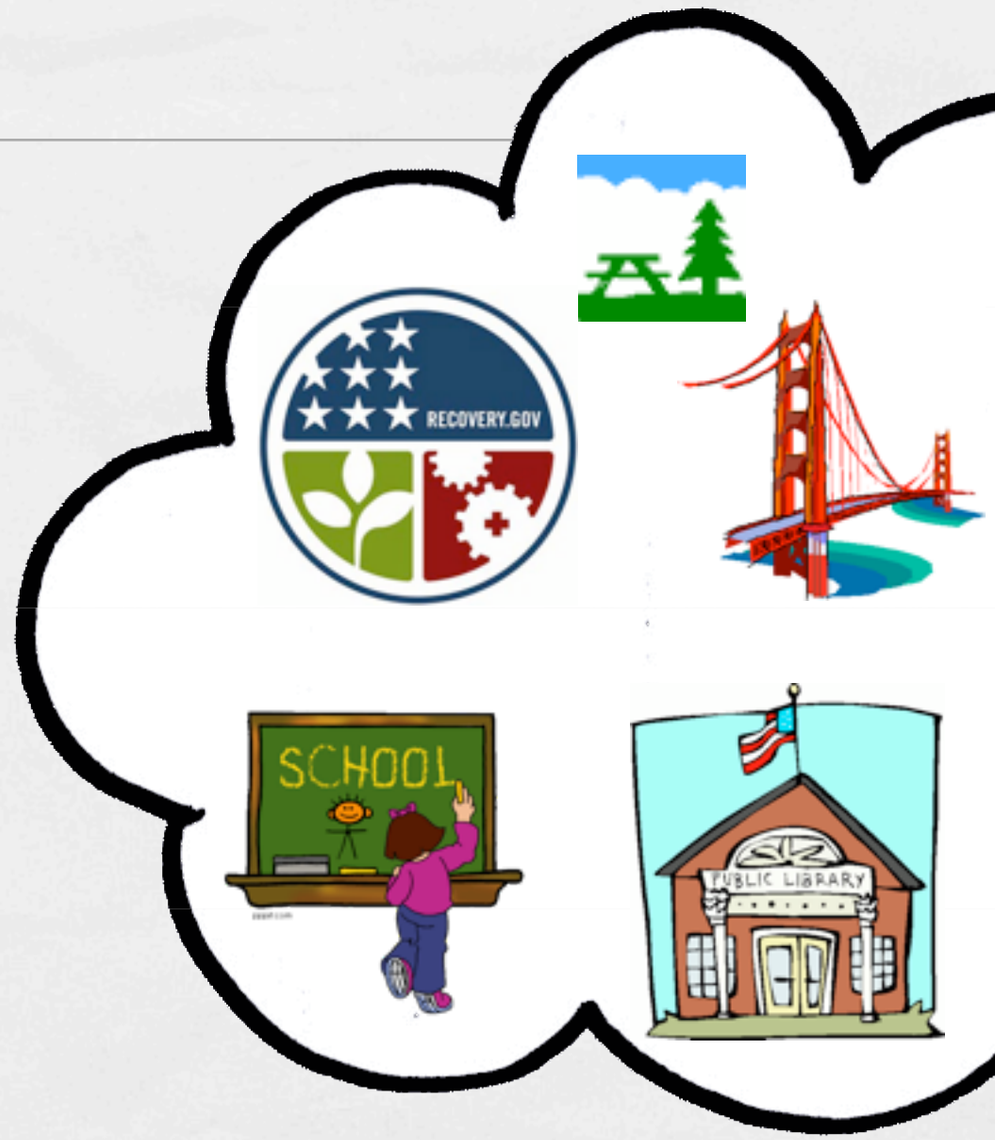
	Non-Private, Efficient Algorithm	Private, Efficient Algorithm	Private, Information Theoretic Algs w/ matching LBs
Vertex Cover	2 x OPT [Pitt85]	(2 + 16/ε) x OPT	Θ(1/ε) x OPT
Weighted Vertex Cover	2 x OPT [Hochbaum82]	(16 + 16/ε) x OPT	Θ(1/ε) x OPT
Set Cover	ln(n) x OPT [Johnson74]	O(ln(n) + ln(m)/ε) x OPT*	Θ(ln (m)/ε) x OPT
Weighted Set Cover	ln(n) x OPT [Chvatal79]	O(ln(n)(ln m + lnln n)/ε) x OPT*	Θ(ln (m)/ε) x OPT
Min Cut	OPT [Ford-Fulkerson56]	OPT + O(ln(n)/ε)*	OPT + Θ(ln(n)/ε)
CPPP	(1-1/e) x OPT [Nemhauser-Wolsey-Fisher78]	(1-1/e) x OPT – O(k ln(m)/ε)*	OPT-Θ(k ln(n/k)/ε)
k-Median	(3 + ε) x OPT [Arya et al. 04]	6 x OPT + O(k^2 $\ln^2$(n/ε))	OPT+Θ(k ln(n/k)/ε)

SUBMODULAR MAXIMIZATION

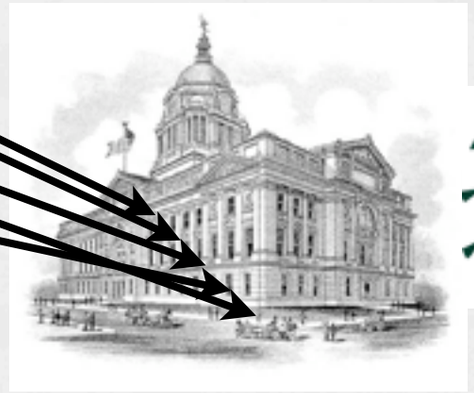
SUBMODULAR MAXIMIZATION

SUBMODULAR MAXIMIZATION

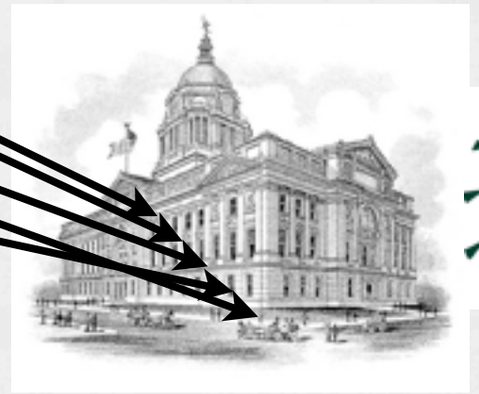




SUBMODULAR MAXIMIZATION



SUBMODULAR MAXIMIZATION



- Combinatorial Public Projects mechanism design problem with set R of m resources, n agents with submodular valuation functions $f_i:2^R->[0,1]$.

- Mechanism elicits valuation functions; outputs a set of k resources.

Private Submodular Maximization

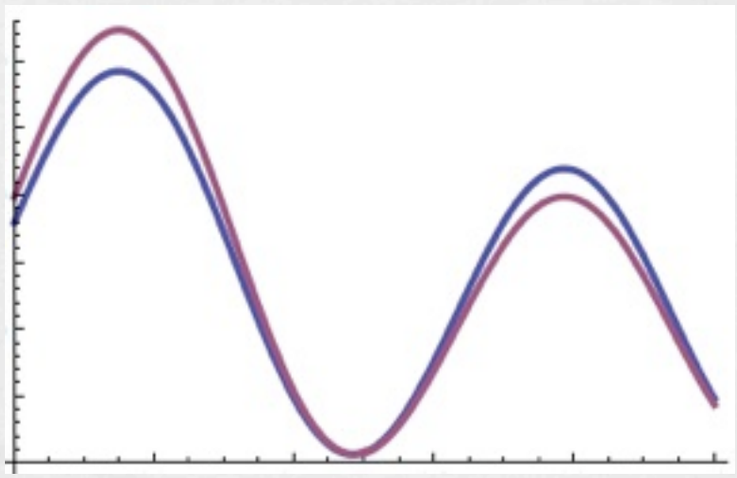
- Combinatorial Public Projects
  mechanism design problem with set R of m resources, n agents with submodular valuation functions $f_i:2^R->[0,1]$; output size k.

- We give a Differentially Private* mechanism that achieves welfare (1-1/e) OPT - O(k ln(m)/ε)
  A constant approximation if OPT is Ω(k ln(m)/ε))

Approximate Differential Privacy [DKM+06]

δ-Approximate ε-Differential Privacy for mechanism A

For any two neighboring data sets D_1, D_2 and potential output set S:

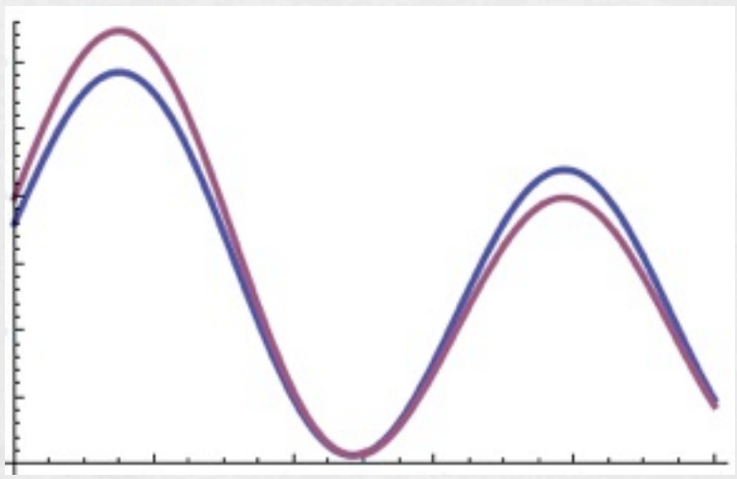


Approximate Differential Privacy [DKM+06]

δ-Approximate ε-Differential Privacy for mechanism A

For any two neighboring data sets D_1, D_2 and potential output set S:

$$Pr[A(D_1) \in S] \leq e^{\varepsilon} Pr[A(D_2) \in S] + \delta$$



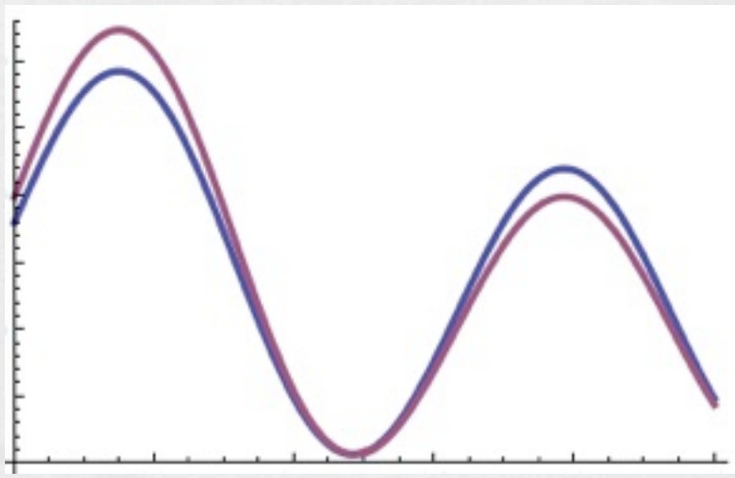
Approximate Differential Privacy [DKM+06]

δ-Approximate ε-Differential Privacy for mechanism A

For any two neighboring data sets D_1, D_2 and potential output set S:

$$Pr[A(D_1) \in S] \leq e^{\varepsilon} Pr[A(D_2) \in S] + \delta$$

Almost all events have their probabilities preserved; those that don't are very low probability



Submodular Maximization

- Take as input a set of resources R and n submodular functions f_i over subsets of R scaled such that $\max_S f_i(S) = 1$

- Output a set of k resources

- Use a probabilistic greedy algorithm

Input: m resources R, $f_1, \ldots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \ldots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) \coloneqq \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp(\epsilon'(F(S_i + \{r\}) - F(S_i)))$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

Theorem: For any δ the algorithm preserves $(8\epsilon'(e-1)\ln\left(\frac{2}{\delta}\right), \delta)$ -differential privacy.

Proof: Let instance A and B differ in the addition of a single user I (to B).

Denote the marginal social utility of r_j at time i by $s_{i,j}(A) = F_A(S_i + \{r_j\}) - F_A(S_i)$

Write $\beta_{i,j} = s_{i,j}(B) - s_{i,j}(A)$ for the additional marginal utility due to user I.

Input: m resources R, $f_1, \dots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \dots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) := \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp(\epsilon'(F(S_i + \{r\}) - F(S_i)))$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

Theorem: For any δ the algorithm preserves $(8\epsilon'(e-1)\ln\left(\frac{2}{\delta}\right), \delta)$ -differential privacy.

Proof: Let instance A and B differ in the addition of a single user I (to B).

Denote the marginal social utility of r_j at time i by $s_{i,j}(A) = F_A(S_i + \{r_j\}) - F_A(S_i)$

Write $\beta_{i,j} = s_{i,j}(B) - s_{i,j}(A)$ for the additional marginal utility due to user I.

$$\prod_{i=1}^{k} \frac{\Pr[M(A) = S_k]}{\Pr[M(B) = S_k]}$$

Input: m resources R, $f_1, \dots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \dots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) := \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp(\epsilon'(F(S_i + \{r\}) - F(S_i)))$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

Theorem: For any δ the algorithm preserves $(8\epsilon'(e-1)\ln\left(\frac{2}{\delta}\right), \delta)$ -differential privacy.

Proof: Let instance A and B differ in the addition of a single user I (to B).

Denote the marginal social utility of r_j at time i by $s_{i,j}(A) = F_A(S_i + \{r_j\}) - F_A(S_i)$

Write $\beta_{i,j} = s_{i,j}(B) - s_{i,j}(A)$ for the additional marginal utility due to user I.

$$\prod_{i=1}^{k} \frac{\Pr[M(A) = S_k]}{\Pr[M(B) = S_k]} = \prod_{i=1}^{k} \frac{\exp(\epsilon' s_{i,r_i}(A)/\sum_j \exp(\epsilon' s_{i,r_j}(A))}{\exp(\epsilon' s_{i,r_i}(B)/\sum_j \exp(\epsilon' s_{i,r_j}(B))}$$

Input: m resources R, $f_1, \ldots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \ldots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) := \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp(\epsilon'(F(S_i + \{r\}) - F(S_i)))$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

Theorem: For any δ the algorithm preserves $(8\epsilon'(e-1)\ln\left(\frac{2}{\delta}\right), \delta)$ -differential privacy.

Proof: Let instance A and B differ in the addition of a single user I (to B).

Denote the marginal social utility of r_j at time i by $s_{i,j}(A) = F_A(S_i + \{r_j\}) - F_A(S_i)$

Write $\beta_{i,j} = s_{i,j}(B) - s_{i,j}(A)$ for the additional marginal utility due to user I.

$$\prod_{i=1}^{k} \frac{\Pr[M(A) = S_k]}{\Pr[M(B) = S_k]} = \prod_{i=1}^{k} \frac{\exp(\epsilon' s_{i,r_i}(A)/\sum_j \exp(\epsilon' s_{i,r_j}(A))}{\exp(\epsilon' s_{i,r_i}(B)/\sum_j \exp(\epsilon' s_{i,r_j}(B))}$$

$$\leq \prod_{i=1}^{k} \frac{\sum_j \exp(\epsilon' s_{i,r_j}(B))}{\sum_j \exp(\epsilon' s_{i,r_j}(A))}$$

Input: m resources R, $f_1, \ldots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \ldots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) := \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp(\epsilon'(F(S_i + \{r\}) - F(S_i)))$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

Theorem: For any δ the algorithm preserves $(8\epsilon'(e-1)\ln\left(\frac{2}{\delta}\right), \delta)$ -differential privacy.

Proof: Let instance A and B differ in the addition of a single user I (to B).

Denote the marginal social utility of r_j at time i by $s_{i,j}(A) = F_A(S_i + \{r_j\}) - F_A(S_i)$

Write $\beta_{i,j} = s_{i,j}(B) - s_{i,j}(A)$ for the additional marginal utility due to user I.

$$\prod_{i=1}^{k} \frac{\Pr[M(A) = S_k]}{\Pr[M(B) = S_k]}$$

Input: m resources R, $f_1, \dots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \dots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) := \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp(\epsilon'(F(S_i + \{r\}) - F(S_i)))$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

Theorem: For any δ the algorithm preserves $\left(8\epsilon'(e-1)\ln\left(\frac{2}{\delta}\right),\delta\right)$ -differential privacy.

Proof: Let instance A and B differ in the addition of a single user I (to B).

Denote the marginal social utility of r_j at time i by $s_{i,j}(A) = F_A(S_i + \{r_j\}) - F_A(S_i)$

Write $\beta_{i,j} = s_{i,j}(B) - s_{i,j}(A)$ for the additional marginal utility due to user I.

$$\prod_{i=1}^{k} \frac{\Pr[M(A) = S_k]}{\Pr[M(B) = S_k]} \leq \prod_{i=1}^{k} \frac{\sum_j \exp(\epsilon'\beta_{i,j})\exp(\epsilon' s_{i,r_j}(A))}{\sum_j \exp(\epsilon' s_{i,r_j}(A))}$$

Input: m resources R, $f_1, \ldots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \ldots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) := \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp(\epsilon'(F(S_i + \{r\}) - F(S_i)))$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

Theorem: For any δ the algorithm preserves $(8\epsilon'(e-1)\ln\left(\frac{2}{\delta}\right), \delta)$ -differential privacy.

Proof: Let instance A and B differ in the addition of a single user I (to B).

Denote the marginal social utility of r_j at time i by $s_{i,j}(A) = F_A(S_i + \{r_j\}) - F_A(S_i)$

Write $\beta_{i,j} = s_{i,j}(B) - s_{i,j}(A)$ for the additional marginal utility due to user I.

$$\prod_{i=1}^{k} \frac{\Pr[M(A) = S_k]}{\Pr[M(B) = S_k]} \leq \prod_{i=1}^{k} \frac{\sum_j \exp(\epsilon' \beta_{i,j}) \exp(\epsilon' s_{i,r_j}(A))}{\sum_j \exp(\epsilon' s_{i,r_j}(A))}$$

$$\leq \prod_{i=1}^{k} \mathbb{E}_i[\exp(\epsilon' \beta_i)]$$

Input: m resources R, $f_1, \ldots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \ldots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) := \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp(\epsilon'(F(S_i + \{r\}) - F(S_i)))$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

Theorem: For any δ the algorithm preserves $(8\epsilon'(e-1)\ln\left(\frac{2}{\delta}\right), \delta)$ -differential privacy.

Proof: Let instance A and B differ in the addition of a single user I (to B).

Denote the marginal social utility of r_j at time i by $s_{i,j}(A) = F_A(S_i + \{r_j\}) - F_A(S_i)$

Write $\beta_{i,j} = s_{i,j}(B) - s_{i,j}(A)$ for the additional marginal utility due to user I.

Input: m resources R, $f_1, \dots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \dots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) := \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp\left(\epsilon'(F(S_i + \{r\}) - F(S_i))\right)$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

Theorem: For any δ the algorithm preserves $(8\epsilon'(e-1)\ln\left(\frac{2}{\delta}\right), \delta)$ -differential privacy.

Proof: Let instance A and B differ in the addition of a single user I (to B).

Denote the marginal social utility of r_j at time i by $s_{i,j}(A) = F_A(S_i + \{r_j\}) - F_A(S_i)$

Write $\beta_{i,j} = s_{i,j}(B) - s_{i,j}(A)$ for the additional marginal utility due to user I.

$$\prod_{i=1}^{k} \frac{\Pr[M(A) = S_k]}{\Pr[M(B) = S_k]}$$

Input: m resources R, $f_1, \ldots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \ldots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) := \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp(\epsilon'(F(S_i + \{r\}) - F(S_i)))$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

Theorem: For any δ the algorithm preserves $(8\epsilon'(e-1)\ln\left(\frac{2}{\delta}\right), \delta)$ -differential privacy.

Proof: Let instance A and B differ in the addition of a single user I (to B).

Denote the marginal social utility of r_j at time i by $s_{i,j}(A) = F_A(S_i + \{r_j\}) - F_A(S_i)$

Write $\beta_{i,j} = s_{i,j}(B) - s_{i,j}(A)$ for the additional marginal utility due to user I.

$$\prod_{i=1}^{k} \frac{\Pr[M(A) = S_k]}{\Pr[M(B) = S_k]} \leq \prod_{i=1}^{k} \mathbb{E}_i[\exp(\epsilon'\beta_i)]$$

Input: m resources R, $f_1, \ldots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \ldots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) := \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp(\epsilon'(F(S_i + \{r\}) - F(S_i)))$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

Theorem: For any δ the algorithm preserves $\left(8\epsilon'(e-1)\ln\left(\frac{2}{\delta}\right),\delta\right)$ -differential privacy.

Proof: Let instance A and B differ in the addition of a single user I (to B).

Denote the marginal social utility of r_j at time i by $s_{i,j}(A) = F_A(S_i + \{r_j\}) - F_A(S_i)$

Write $\beta_{i,j} = s_{i,j}(B) - s_{i,j}(A)$ for the additional marginal utility due to user I.

$$\prod_{i=1}^{k} \frac{\Pr[M(A) = S_k]}{\Pr[M(B) = S_k]} \leq \prod_{i=1}^{k} \mathbb{E}_i[\exp(\epsilon'\beta_i)] \leq \exp((e-1)\epsilon' \sum_{i=1}^{k} \mathbb{E}_i[\beta_i])$$

Input: m resources R, $f_1, \ldots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \ldots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) := \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp(\epsilon'(F(S_i + \{r\}) - F(S_i)))$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

Theorem: For any δ the algorithm preserves $\left(8\epsilon'(e-1)\ln\left(\frac{2}{\delta}\right),\delta\right)$ -differential privacy.

Proof: Let instance A and B differ in the addition of a single user I (to B).

Denote the marginal social utility of r_j at time i by $s_{i,j}(A) = F_A(S_i + \{r_j\}) - F_A(S_i)$

Write $\beta_{i,j} = s_{i,j}(B) - s_{i,j}(A)$ for the additional marginal utility due to user I.

$$\prod_{i=1}^{k}\frac{\Pr[M(A)=S_k]}{\Pr[M(B)=S_k]} \le \prod_{i=1}^{k}\mathbb{E}_i[\exp(\epsilon'\beta_i)] \le \exp\left((e-1)\epsilon'\sum_{i=1}^{k}\mathbb{E}_i[\beta_i]\right)$$

$$\mathbb{E}[e^x] \le e^{(e-1)\mathbb{E}[x]}$$

Input: m resources R, $f_1, \dots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \dots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) := \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp(\epsilon'(F(S_i + \{r\}) - F(S_i)))$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

Theorem: For any δ the algorithm preserves $\left(8\epsilon'(e-1)\ln\left(\frac{2}{\delta}\right),\delta\right)$ -differential privacy.

Proof: Let instance A and B differ in the addition of a single user I (to B).

Denote the marginal social utility of r_j at time i by $s_{i,j}(A) = F_A(S_i + \{r_j\}) - F_A(S_i)$

Write $\beta_{i,j} = s_{i,j}(B) - s_{i,j}(A)$ for the additional marginal utility due to user I.

$$\prod_{i=1}^{k} \frac{\Pr[M(A) = S_k]}{\Pr[M(B) = S_k]} \leq \prod_{i=1}^{k} \mathbb{E}_i[\exp(\epsilon'\beta_i)] \leq \exp((e-1)\epsilon' \sum_{i=1}^{k} \mathbb{E}_i[\beta_i])$$

For x ≤ 1: $\mathbb{E}[e^x] \leq e^{(e-1)\mathbb{E}[x]}$

Input: m resources R, $f_1, \ldots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \ldots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) := \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp(\epsilon'(F(S_i + \{r\}) - F(S_i)))$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

Theorem: For any δ the algorithm preserves $\left(8\epsilon'(e-1)\ln\left(\frac{2}{\delta}\right),\delta\right)$ -differential privacy.

Proof: Let instance A and B differ in the addition of a single user I (to B).

Denote the marginal social utility of r_j at time i by $s_{i,j}(A) = F_A(S_i + \{r_j\}) - F_A(S_i)$

Write $\beta_{i,j} = s_{i,j}(B) - s_{i,j}(A)$ for the additional marginal utility due to user I.

$$\prod_{i=1}^{k} \frac{\Pr[M(A) = S_k]}{\Pr[M(B) = S_k]} \leq \prod_{i=1}^{k} \mathbb{E}_i[\exp(\epsilon'\beta_i)]$$

Input: m resources R, $f_1, \ldots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \ldots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) := \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp(\epsilon'(F(S_i + \{r\}) - F(S_i)))$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

Theorem: For any δ the algorithm preserves $\left(8\epsilon'(e-1)\ln\left(\frac{2}{\delta}\right),\delta\right)$ -differential privacy.

Proof: Let instance A and B differ in the addition of a single user I (to B).

Denote the marginal social utility of r_j at time i by $s_{i,j}(A) = F_A(S_i + \{r_j\}) - F_A(S_i)$

Write $\beta_{i,j} = s_{i,j}(B) - s_{i,j}(A)$ for the additional marginal utility due to user I.

$$\prod_{i=1}^{k} \frac{\Pr[M(A) = S_k]}{\Pr[M(B) = S_k]} \leq \prod_{i=1}^{k} \mathbb{E}_i[\exp(\epsilon'\beta_i)] \leq \exp((e-1)\epsilon' \sum_{i=1}^{k} \mathbb{E}_i[\beta_i])$$

Input: m resources R, $f_1, \ldots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \ldots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) := \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp(\epsilon'(F(S_i + \{r\}) - F(S_i)))$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

Theorem: For any δ the algorithm preserves $(8\epsilon'(e-1)\ln\left(\frac{2}{\delta}\right), \delta)$ -differential privacy.

Proof: Let instance A and B differ in the addition of a single user I (to B).

Denote the marginal social utility of r_j at time i by $s_{i,j}(A) = F_A(S_i + \{r_j\}) - F_A(S_i)$

Write $\beta_{i,j} = s_{i,j}(B) - s_{i,j}(A)$ for the additional marginal utility due to user I.

$$\prod_{i=1}^{k} \frac{\Pr[M(A) = S_k]}{\Pr[M(B) = S_k]} \leq \prod_{i=1}^{k} \mathbb{E}_i[\exp(\epsilon'\beta_i)] \leq \exp((e-1)\epsilon' \sum_{i=1}^{k} \mathbb{E}_i[\beta_i])$$

But the *realized* utility user I achieves is at most 1...

So by a Chernoff bound, except with probability δ:

$$\sum_{i=1}^{k} \mathbb{E}_i[\beta_i] \leq 8\ln(\frac{2}{\delta})$$

Input: m resources R, $f_1, \ldots, f_n \quad k, \epsilon, \delta$ **Output:** $S_k = \{r_1, \ldots, r_k\}$

1. Let $R_1 \leftarrow R, F(x) := \sum_{i=1}^{n} f_i(x), S_1 \leftarrow \emptyset, \epsilon' \leftarrow \frac{\epsilon}{8e\ln(2/\delta)}$
2. **for** i = 1 to k **do**
3. **pick** r from R_i with probability proportional to $\exp(\epsilon'(F(S_i + \{r\}) - F(S_i)))$
4. **let** $R_{i+1} \leftarrow R_i - \{r\}, S_{i+1} \leftarrow S_i + \{r\}$
5. **end for.**
6. **output** S_{k+1}

PRIVACY => TRUTHFULNESS

Definition: A mechanism M is γ-truthful if for all players i, for all valuation functions f_i':

$$E[f_i(M(f_1,\ldots,f_i,\ldots,f_n))] \geq E[f_i(M(f_1,\ldots, f_i',\ldots,f_n))] - \gamma$$

PRIVACY => TRUTHFULNESS

Definition: A mechanism M is γ-truthful if for all players i, for all valuation functions f_i':

$$E[f_i(M(f_1,\ldots,f_i,\ldots,f_n))] \geq E[f_i(M(f_1,\ldots, f_i',\ldots,f_n))] - \gamma$$

- No efficient, 0-truthful mechanism achieves $OPT/m^{1/2-\varepsilon}$ [PSS08]

PRIVACY => TRUTHFULNESS

Definition: A mechanism M is γ-truthful if for all players i, for all valuation functions f_i':

$$E[f_i(M(f_1,\ldots,f_i,\ldots,f_n))] \geq E[f_i(M(f_1,\ldots, f_i',\ldots,f_n))] - \gamma$$

- No efficient, 0-truthful mechanism achieves $OPT/m^{1/2-\varepsilon}$ [PSS08]
- Any ε-Differentially Private mechanism is (2ε)-truthful [MT07]

PRIVACY => TRUTHFULNESS

Definition: A mechanism M is γ-truthful if for all players i, for all valuation functions f_i':

$$E[f_i(M(f_1,\ldots,f_i,\ldots,f_n))] \geq E[f_i(M(f_1,\ldots, f_i',\ldots,f_n))] - \gamma$$

- No efficient, 0-truthful mechanism achieves $OPT/m^{1/2-\varepsilon}$ [PSS08]

- Any ε-Differentially Private mechanism is (2ε)-truthful [MT07]

- This work: a constant-approx (for significant OPT), efficient, truthful mechanism

	Non-Private, Efficient Algorithm	Private, Efficient Algorithm	Private, Information Theoretic Algs w/ matching LBs
Vertex Cover	2 x OPT [Pitt85]	$(2 + 16/\varepsilon)$ x OPT	$\Theta(1/\varepsilon)$ x OPT
Weighted Vertex Cover	2 x OPT [Hochbaum82]	$(16 + 16/\varepsilon)$ x OPT	$\Theta(1/\varepsilon)$ x OPT
Set Cover	$\ln(n)$ x OPT [Johnson74]	$O(\ln(n) + \ln(m)/\varepsilon)$ x OPT*	$\Theta(\ln(m)/\varepsilon)$ x OPT
Weighted Set Cover	$\ln(n)$ x OPT [Chvatal79]	$O(\ln(n)(\ln m + \ln\ln n)/\varepsilon)$ x OPT*	$\Theta(\ln(m)/\varepsilon)$ x OPT
Min Cut	OPT [Ford-Fulkerson56]	OPT + $O(\ln(n)/\varepsilon)$*	OPT + $\Theta(\ln(n)/\varepsilon)$
CPPP	(1-1/e) x OPT [Nemhauser-Wolsey-Fisher78]	(1-1/e) x OPT – $O(k \ln(m)/\varepsilon)$*	OPT-$\Theta(k \ln(n/k)/\varepsilon)$
k-Median	$(3 + \varepsilon)$ x OPT [Arya et al. 04]	6 x OPT + $O(k^2 \ln^2(n/\varepsilon))$	OPT+$\Theta(k \ln(n/k)/\varepsilon)$

Techniques and Themes

- Adding noise to the input

- Hard constraints = challenge for privacy
  - Non-explicit outputs

- Various applications of exponentially weighted sampling
  - Iterative, reducing the sample space

Future Work

- More sophisticated understanding of problem/technique classification

- Further exploration of interactions between game theory and privacy