Privacy of Correlated Data & Relaxations of Differential Privacy

CompSci 590.03
Instructor: Ashwin Machanavajjhala

Duke UNIVERSITY

Outline

- Recap: Pufferfish Privacy Framework [Kifer-**M** PODS'12]
- Defining Privacy for Correlated Data [Kifer-**M** PODS'12 & Ding-**M** '13]
  - Induced Neighbor Privacy
- Relaxing differential privacy for utility
  - Crowd Blending Privacy [Gehrke et al CRYPTO '12]
  - E-privacy [**M** et al VLDB '09]

Recap: No Free Lunch Theorem

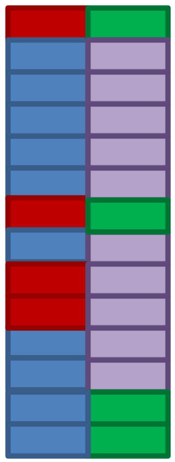
It is not possible to guarantee *any* utility in addition to privacy, *without making assumptions about*

- *the data generating distribution* [Kifer-Machanavajjhala SIGMOD '11]
- *the background knowledge available to an adversary* [Dwork-Naor JPC '10]

Correlations & Differential Privacy

- When an adversary knows that individuals in a table are correlated, then (s)he can learn sensitive information about individuals even from the output of a differentially private mechanism.
- Example 1: Contingency tables with pre-released exact counts
- Example 2: Social Networks

Marginal counts



D

	2 + Lap(1/ε)	2 + Lap(1/ε)	4
	2 + Lap(1/ε)	8 + Lap(1/ε)	10
	4	10	

Mean : 8
Variance : $2/ke^2$

can reconstruct the table with high precision for large k

Recap: Why we need domain specific privacy?

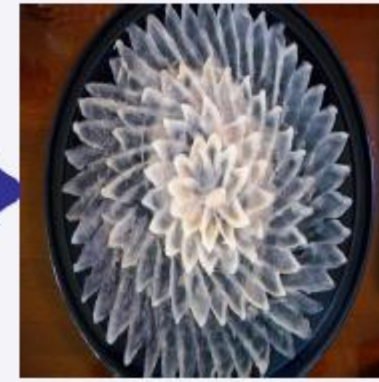
- For handling correlations
  - Prereleased marginals & Social networks [Kifer-**M** SIGMOD '11]
- Utility driven applications
  - For some applications existing privacy definitions do not provide sufficient utility [**M** et al PVLDB '11]
- Personalized privacy & aggregate secrets [Kifer-**M** PODS '12]

Qn: How to design principled privacy definitions customized to such scenarios?

Recap: Pufferfish Framework

- Pufferfish (data):
  - contains tetrodotoxin (sensitive information).
- Toxin is everywhere:
  - Liver
  - Intestines
  - Skin / Muscles
- Removing all toxin = removing fish

- Chef (algorithm):
  - Processes the fish.
- Certification and license (privacy definition):
  - Rules chef must follow / restrictions on algorithm
  - Guarantees output is (relatively) safe.



- Fugu (sanitized data):
  - Tasty (high utility)
  - Minimal toxins
  - Minimal leakage of sensitive information

Recap: Pufferfish Semantics

- What is being kept secret?
- Who are the adversaries?
- How is information disclosure bounded?

Recap: Sensitive Information

- **Secrets**: S be a set of potentially sensitive statements
  - “individual j’s record is in the data, and j has Cancer”
  - “individual j’s record is not in the data”

- **Discriminative Pairs**: Spairs is a subset of SxS. Mutually exclusive pairs of secrets.
  - (“Bob is in the table”, “Bob is not in the table”)
  - (“Bob has cancer”, “Bob has diabetes”)

Recap: Adversaries

- An adversary can be completely characterized by his/her prior information about the data
  - We do not assume computational limits

- **Data Evolution Scenarios**: set of all probability distributions that could have generated the data.
  - *No assumptions*: All probability distributions over data instances are possible.

  - *I.I.D.*: Set of all f such that: $P(data = \{r_1, r_2, \ldots, r_k\}) = f(r_1) \times f(r_2) \times \ldots \times f(r_k)$

Recap: Pufferfish Framework

- Mechanism M satisfies ε-Pufferfish(S, Spairs, D), if for every
  - w ε Range(M),
  - (s_i, s_j) ε Spairs
  - Θ ε D, such that $P(s_i \mid \theta) \neq 0$, $P(s_j \mid \theta) \neq 0$

$$P(M(data) = w \mid s_i, \theta) \leq e^{\varepsilon} P(M(data) = w \mid s_j, \theta)$$

Recap: Pufferfish Semantic Guarantee

$$e^{-\epsilon} \leq \frac{P(s_i \mid \mathfrak{M}(\mathfrak{Data}) = \omega, \theta)}{P(s_j \mid \mathfrak{M}(\mathfrak{Data}) = \omega, \theta)} \Big/ \frac{P(s_i \mid \theta)}{P(s_j \mid \theta)} \leq e^{\epsilon}$$

Posterior odds of s_i vs s_j

Prior odds of s_i vs s_j

Recap: Pufferfish & Differential Privacy

- Spairs:
  - "record j is in the table" vs "record j is not in the table"
  - "record j is in the table with value x" vs "record j is not in the table"
- Data evolution:
  - For all $\theta = [f_1, f_2, f_3, \ldots, f_k, \pi_1, \pi_2, \ldots, \pi_k]$
  - $P[\text{Data} = D \mid \theta] = \prod_{rj \text{ not in } D} (1-\pi_j) \times \prod_{rj \text{ in } D} \pi_j \times f_j(r_j)$

A mechanism M satisfies differential privacy

if and only if

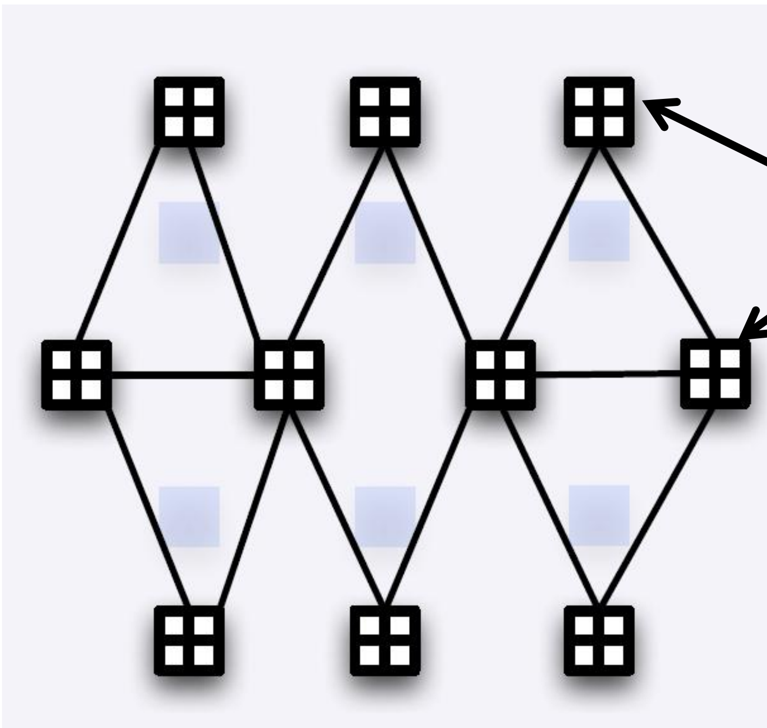
it satisfies Pufferfish instantiated using Spairs and {θ}

(as defined above)

Outline

- Recap: Pufferfish Privacy Framework [Kifer-**M** PODS'12]
- Defining Privacy for Correlated Data [Kifer-**M** PODS'12 & Ding-**M** '13]
  - Current research
- Relaxing differential privacy for utility
  - Crowd Blending Privacy [Gehrke et al CRYPTO '12]
  - E-privacy [**M** et al VLDB '09]

Reason for Privacy Breach

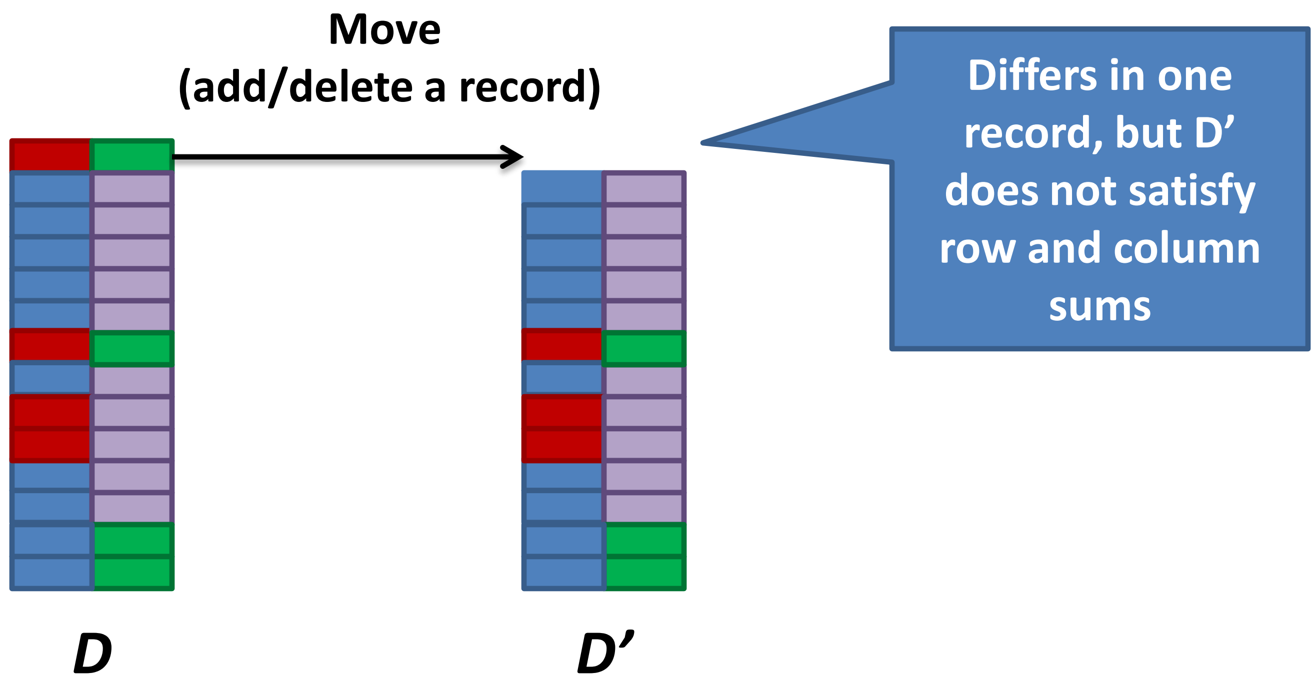


 can distinguish between every pair of these tables based on the output

Space of all possible tables

Induced Neighbor Privacy

- *Differential Privacy*: Neighboring tables differ in one value
  … But one or both the neighbors may not satisfy the constraints.



Duke UNIVERSITY

Induced Neighbor Privacy

Induces Neighbors (Q) [Kifer-**M** ’11 & Pan]

- Pick an individual j
- Consider 2 tables D_a, D_b that differ in j’s record
  - $D_a(j) = a$, and $D_b(j) = b$
- D_a and D_b are induced neighbors if they are **minimally different**
  - D_a and D_b satisfy the constraints in Q
  - Let M = {m1, m2, …, mk} be the smallest set of *moves* that change D_a to D_b
  - There does not exist a D_c which satisfies the constraints and can be constructed from D_a using a subset of moves from D_b

Example 1

Table A

a1,b1
a2,b2
a3,b3

	a1	a2	a3	
b1	1			1
b2		1		1
b3			1	1
	1	1	1	

Table B

a1,b2
a2,b2
a3,b3

Is Table B an Induced Neighbor of Table A given the row and column sums?

Ans: NO

Example 1

Table B

a1,b2
a2,b2
a3,b3

Table B does not satisfy row and column sums.

	a1	**a2**	**a3**	
b1				~~1~~
b2	**1**	**1**		~~1~~
b3			**1**	**1**
	1	**1**	**1**	

Example 2

Table A

a1,b1
a2,b2
a3,b3
a1,b1
a2,b2
a3,b3

	a1	a2	a3	
b1	2			2
b2		2		2
b3			2	2
	2	2	2	

Is Table B an Induced Neighbor of Table A given the row and column sums?

Ans: No

Table B

a1,b2
a2,b3
a3,b1
a1,b2
a2,b3
a3,b1

	a1	a2	a3	
b1			2	2
b2	2			2
b3		2		2
	2	2	2	

Example 2

Table A	Table C	Table B
a1,b1	a1,b2	a1,b2
a2,b2	a2,b3	a2,b3
a3,b3	a3,b1	a3,b1
a1,b1	a1,b1	a1,b2
a2,b2	a2,b2	a2,b3
a3,b3	a3,b3	a3,b1

Table C can be generated from Table A using a subset of moves.

	a1	a2	a3	
b1	1		1	2
b2	1	1		2
b3		1	1	2
	2	2	2	

Duke UNIVERSITY

Example 3

Table A

a1,b1
a2,b2
a3,b3
a1,b1
a2,b2
a3,b3

Table C

a1,b2
a2,b3
a3,b1
a1,b1
a2,b2
a3,b3

Table C and Table A are induced neighbors.

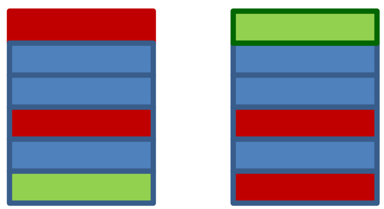
	a1	a2	a3	
b1	2			2
b2		2		2
b3			2	2
	2	2	2	

	a1	a2	a3	
b1	1		1	2
b2	1	1		2
b3		1	1	2
	2	2	2	

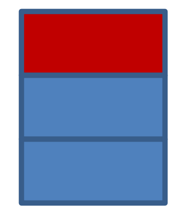
Induced Neighbor Privacy

For every pair of induced neighbors

For every output …



D_1 D_2



O

Adversary should not be able to distinguish between any D_1 and D_2 based on any O

$$\log\left(\frac{\Pr[A(D_1) = O]}{\Pr[A(D_2) = O]}\right) < \varepsilon \qquad (\varepsilon > 0)$$

Induced Neighbors Privacy and Pufferfish

- Given a set of count constraints Q,
- Spairs:
  - "record j is in the table" vs "record j is not in the table"
  - "record j is in the table with value x" vs "record j is not in the table"
- Data evolution:
  - For all $\theta = [f_1, f_2, f_3, \ldots, f_k, \pi_1, \pi_2, \ldots, \pi_k]$
  - $P[\text{Data} = D \mid \theta] \propto \prod_{r_j \text{ not in } D} (1-\pi_j) \times \prod_{r_j \text{ in } D} \pi_j \times f_j(r_j)$, if D satisfies Q
  - $P[\text{Data} = D \mid \theta] = 0$, if D does not satisfy Q

Conjecture: A mechanism M satisfies induced neighbors privacy
if and only if
it satisfies Pufferfish instantiated using Spairs and {θ}

Laplace Mechanism for Induced Neighbors Privacy

Thm: If **induced-sensitivity** of the query is **S_{in}(q)**, then adding Lap(λ) noise guarantees **ε-participation privacy**.

$$\lambda = S_{in}(q)/\varepsilon$$

S_{in}(q): Smallest number s.t. for any **induced-neighbors** d, d',

$$|| q(d) - q(d') ||_1 \leq S_{in}(q)$$

Induced Sensitivity

- $q_{a1,b1}$: The number of records with A = a1 and B = b1?
  - Sensitivity = ?

- q_{b1}: The number of records with B=b1?
  - Sensitivity = ?

- q_{all}: All the counts in the contingency table?
  - Sensitivity = ?

?	?	?	2
?	?	?	2
?	?	?	2
2	2	2	

Duke UNIVERSITY

Induced Sensitivity

- $q_{a1,b1}$: The number of records with A = a1 and B = b1?
  - Sensitivity = 1

- q_{b1}: The number of records with B=b1?
  - Sensitivity = 0

- q_{all}: All the counts in the contingency table?
  - Sensitivity = 6

?	?	?	2
?	?	?	2
?	?	?	2
2	2	2	

Duke UNIVERSITY

Induced Sensitivity

What is the sensitivity if all counts in the contingency table are released?

- Sensitivity ≥ 6

2			2
	2		2
		2	2
2	2	2	

Table A

\-

1		1	2
1	1		2
	1	1	2
2	2	2	

Table C

=

+1		-1	2
-1	+1		2
	-1	+1	2
2	2	2	

Diff

Induced sensitivity

- The Diff between two induced neighbors represents the moves
  - + means addition and – means deletion.
  - +1 in each cell must be offset by a -1 in the same row and another -1 in the same column (degree = 2)
  - Hence, if we have an edge between every +1 and -1 in the same row or column, we get a graph which is a collection of cycles!.

+	−	
−		+
	+	−

+	−		
−		+	
	+		−
		−	+

+	−			
−		+		
	+		−	
		−		+
			+	−

Induced Sensitivity

2			2
	2		2
		2	2
2	2	2	

Table A

\-

1		1	2
1	1		2
	1	1	2
2	2	2	

Table C

=

+1		-1	2
-1	+1		2
	-1	+1	2
2	2	2	

Diff

Simple cycle can have at most min(2r, 2c) nodes

where r = number of rows

c = number of columns

Duke UNIVERSITY

Induced Sensitivity

2			2
	2		2
		2	2
2	2	2	

Table A

-

1	1		2
1		1	2
	1	1	2
2	2	2	

Table D

(NOT an induced neighbor of A)

=

+1	-1		2
-1	+2	-1	2
	-1	+1	2
2	2	2	

Diff

2			2
	2		2
		2	2
2	2	2	

Table A

-

1	1		2
1	1		2
		2	2
2	2	2	

Table E

(is an induced neighbor of A)

=

+1	-1		2
-1	+1		2
			2
2	2	2	

Diff

Computing induced sensitivity

2D case:

q_{all} : outputs all the counts in a 2-D contingency table.
Marginals: row and column sums.
The induced-sensitivity of q_{all} = *min(2r, 2c)*.

General Case: Deciding whether $S_{in}(q) > 0$ is NP-hard.

Conjecture: Computing $S_{in}(q)$ is hard (and complete) for the second level of the polynomial hierarchy.

Summary

- Correlations in the data can allow adversaries to learn sensitive information even from a differentially private release.
- Induced Neighbors Privacy helps limit this disclosure when correlations are due constraints that are publicly known about the data.
- Algorithms for differential privacy can be used to ensure induced neighbor privacy by using the appropriate sensitivity.

Open Questions

- Induced neighbor privacy for general count constraints
  - Are ways to approximate the sensitivity?
- Answering queries using noisy data + exact knowledge
- Privacy of social networks
  - Adversaries may use social network evolution models to infer sensitive information about edges in a network [Kifer-**M** SIGMOD '11]
  - Can correlations in a social network be generatively described?

Outline

- Recap: Pufferfish Privacy Framework [Kifer-**M** PODS'12]
- Defining Privacy for Correlated Data [Kifer-**M** PODS'12 & Ding-**M** '13]
  - Current research
- Relaxing differential privacy for utility
  - Crowd Blending Privacy [Gehrke et al CRYPTO '12]
  - E-privacy [**M** et al VLDB '09]

Recap: Pufferfish & Differential Privacy

- Spairs:
  - "record j is in the table" vs "record j is not in the table"
  - "record j is in the table with value x" vs "record j is not in the table"
- Data evolution:
  - For all $\theta = [f_1, f_2, f_3, \ldots, f_k, \pi_1, \pi_2, \ldots, \pi_k]$
  - $P[\text{Data} = D \mid \theta] = \prod_{r_j \text{ not in } D} (1-\pi_j) \times \prod_{r_j \text{ in } D} \pi_j \times f_j(r_j)$

An adversary may know an arbitrary distribution about each individual

A mechanism M satisfies differential privacy if and only if it satisfies Pufferfish instantiated using Spairs and {θ}

Duke UNIVERSITY

Need for relaxed notions of privacy

- In certain applications, differentially private mechanisms do not provide sufficient utility

- How to define privacy while guarding against restricted forms of attackers?
  - Need to be resistant to attacks: Previous definitions were susceptible to composition, minimality, and other attacks.

Duke
UNIVERSITY

Approaches to Relax Privacy

- Computationally Bounded Adversaries [Groce et al TCC ‘11]
- Allowing certain disclosures [Gehrke et al CRYPTO ‘12]
- Considering “realistic” adversaries with bounded prior knowledge [M et al VLDB ‘09]

Restricting the Adversary's computational power

- Consider attackers who can execute a polynomial time Turing machine (e.g., only use algorithms in P)

- [Groce et al TCC '11]
  "… for queries with output in R^d (for a constant d) and a natural class of utilities, **any computationally private mechanism can be converted to a statistically private mechanism** that is roughly as efficient and achieves almost the same utility …"

Crowd-blending Privacy

[Gehrke et al CRYPTO '12]

Definition: Individuals t and t' in a database D are ***indistinguishable*** with respect to mechanism M if, for all outputs w

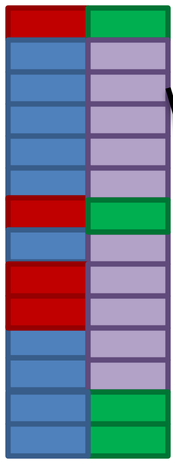
$$P[M(D) = w] \leq e^{\varepsilon} P[M(D_{t,t'}) = w]$$

where, $D_{t,t'}$ is the database where t is replaced with t'

Blending in a Crowd:

An individual t in D is said to ***ε-blend*** in a crowd of k people with respect to mechanism M if t is indistinguishable from k-1 other individuals in the data.

Crowd Blending Privacy

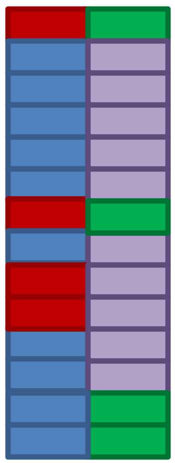


D

2	2	4
2	8	10
4	10	

This individual 0-blends in a crowd of size 8

Crowd Blending Privacy



D

	2 + Lap(2/ε)	2+ Lap(2/ε)
	2+ Lap(2/ε)	8+ Lap(2/ε)

Every tuple ε-blends in a crowd of size N = 14

Crowd Blending Privacy

Definition:

A mechanism M is (k, ε)-crowd blending private if for every database D and every individual t,

- either, t ε-blends in a crowd of size k

- or, for all w, $P(M(D) = w) \leq e^{\varepsilon} P(M(D - \{t\}) = w)$

Mechanisms

- Release a histogram by suppressing all counts less than k
  - Satisfies (K,0)-crowd blending privacy
- Release a histogram by adding Laplace noise to counts less than k
  - Satisfies (K, ε)-crowd blending privacy

Weaker than differential privacy

- Adversary can infer a sensitive property of an individual. But it will be shared by at least k other people
  - This looks like a property of the population rather than that of the individual.

- The definition does not satisfy composability.

Sampling + Crowd-blending => Differential Privacy

- Let M_p be a mechanism that:
  - Constructs a sample S by picking each record in the data with probability p
  - Executes mechanism M on S.

Theorem:

If M is (k,ε)-crowd-blending private (for k > 1). Then M_p satisfies:

$\forall\, D, D'$ *that differ in one record,*
$\forall\, w \in Range(M)$

$$P\left(M_p(D) = w\right) \leq e^{\varepsilon} P\left(M_p(D') = w\right) + \delta$$

$$\varepsilon = \ln\left(pe^{\varepsilon} \cdot \left(\frac{2-p}{1-p}\right) + 1 - p\right) \qquad \delta = e^{-\Omega(k\cdot(1-p)^2)}$$

Duke UNIVERSITY

Open Questions

- What other mechanisms satisfy Crowd-blending privacy?
- Given a privacy budget, can we answer a workload of queries with minimum error by using the sampling + crowd-blending approach?
- Sampling + k-anonymity => Differential Privacy
  - What other mechanisms in addition to sampling give sufficient privacy?
- How big should K be?
  - K is the boundary between individual-specific and population level properties.

Next Class

- E-privacy
  - Relaxation of differential privacy which limits the adversaries considered.

- Application of privacy technology to US Census

References

[Groce et al TCC '11]

A. Groce, J. Katz, A. Yerukhimovic, "*Limits on computational privacy in the client/server setting*", TCC 2011

[Gehrke et al CRYPTO '12]

J. Gehrke, M. Hay, E. Liu, R. Pass, "*Crowd Blending Privacy*", CRYPTO 2012

[**M** et al VLDB '09]

A. Machanavajjhala, J. Gehrke, M. Gotz, "*Data Publishing against Realistic Adversaries",* PVLDB 2(1) 2009

[Kifer-**M** SIGMOD'11]

D. Kifer, A. Machanavajjhala, "*No Free Lunch in Data Privacy*", SIGMOD 2011

[Kifer-**M** PODS'12]

D. Kifer, A. Machanavajjhala, "*A Rigorous and Customizable Framework for Privacy*", PODS 2012

[Ding-**M** '12]

B. Ding, A. Machanavajjhala, "Induced Neighbors Privacy(*Work in progress)*", 2012