Query-Based Data Pricing

Dan Suciu – U. of Washington

Joint with M. Balazinska, B. Howe, P. Koutris, Daniel Li, Chao Li, G. Miklau, P. Upadhyaya
Data Has Value

And it is increasingly being sold/bought on the Web

• Big data vendors
• Data Markets
• Private data

Pricing digital goods is challenging [Shapiro&Varian]
Pricing Data

Pricing data lies at the intersection of several areas:

• Data management
• Mechanism design
• Economics

This talk
1. Big Data Vendors

High value data

- Gartner report: $5k, even if you need only one chart
- Navteq Maps
- Factual
- A few others [Muschalle]:

Expensive datasets, available only to major customers
2. Data Markets

- Azure DataMarkets – 100+ data sources
- Infochimps – 15,000 data sets
- Xignite – financial data
- Aggdata
- Gnip – social media data
- PatientsLikeMe

These datasets are available to the little guy. The markets themselves are struggling, because they are just facilitators; no innovation
3. Private Data

- Private data has value
  - A unique user: $4 at FB, $24 at Google [JPMorgan]

- Today’s common practice:
  - Companies profit from private data without compensating users

- New trend: allow users to profit financially
  - Industry: personal data locker
    https://www.personal.com/, http://lockerproject.org/
  - Academia: mechanisms for selling private data
    [Ghosh11,Gkatzelis12,Aperjis11,Roth12,Riederer12]
Sample Data Markets
Different price by business type
$699 for 885,976 teacher names & emails!
Cheaper just for Washington
A Criticism of Today’s Pricing Schemes

• Small buyers want to purchase only a tiny amount of data: if they can’t, they give up

• Large buyers have specific needs: price is often negotiated in a room-full-of-lawyers

• Sellers can’t easily anticipate all possible queries that buyers might ask

Needed: more flexible pricing scheme, parameterized by queries
Outline

• Framework and examples

• Results so far

• Conclusions
Query-based Pricing

• Seller defines *price-points*: 
  \((V_1, p_1), (V_2, p_2), \ldots\)  Meaning: price\((V_i)=p_i\). 

• Buyer may buy any *query* \(Q\)

• System will determine \(price_D(Q)\) based on:
  – The price points
  – The current database instance \(D\)
  – The query \(Q\)

How should a “*good*” price function be?
Arbitrage Freeness

**Arbitrage-free Axiom:**
For all queries $Q_1, \ldots, Q_k, Q$, if $Q_1, \ldots, Q_k$ determine $Q$, then:

$$\text{price}_D(Q) \leq \text{price}_D(Q_1) + \ldots + \text{price}_D(Q_k)$$

“$Q_1, \ldots, Q_k$ determine $Q$” means that $Q(D)$ can be answered from $Q_1(D), \ldots, Q_k(D)$, without accessing the database instance $D$. 
# Example 1: Pricing Relational Data

S(Shape, Color, Picture)

<table>
<thead>
<tr>
<th>Shape</th>
<th>Color</th>
<th>Picture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swan</td>
<td>White</td>
<td><img src="" alt="Swan Image" /></td>
</tr>
<tr>
<td>Swan</td>
<td>Yellow</td>
<td><img src="" alt="Swan Image" /> . . . . . .</td>
</tr>
<tr>
<td>Dragon</td>
<td>Yellow</td>
<td><img src="" alt="Dragon Image" /></td>
</tr>
<tr>
<td>Car</td>
<td>Yellow</td>
<td>. . . . . .</td>
</tr>
<tr>
<td>Fish</td>
<td>White</td>
<td>. . . . . .</td>
</tr>
</tbody>
</table>

Price list

<table>
<thead>
<tr>
<th>Price list</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_1 = \sigma_{\text{Shape}='Swan'}(S)$</td>
<td>$2$</td>
</tr>
<tr>
<td>$V_2 = \sigma_{\text{Shape}='Dragon'}(S)$</td>
<td>$2$</td>
</tr>
<tr>
<td>$V_3 = \sigma_{\text{Shape}='Car'}(S)$</td>
<td>$2$</td>
</tr>
<tr>
<td>$V_4 = \sigma_{\text{Shape}='Fish'}(S)$</td>
<td>$2$</td>
</tr>
<tr>
<td>$W_1 = \sigma_{\text{Color}='White'}(S)$</td>
<td>$3$</td>
</tr>
<tr>
<td>$W_2 = \sigma_{\text{Color}='Yellow'}(S)$</td>
<td>$3$</td>
</tr>
<tr>
<td>$W_3 = \sigma_{\text{Color}='Red'}(S)$</td>
<td>$3$</td>
</tr>
</tbody>
</table>

Price($\sigma_{\text{Shape}}$) = $2$

Price($\sigma_{\text{Color}}$) = $3$

Example 1: Pricing Relational Data

S(Shape, Color, Picture)

<table>
<thead>
<tr>
<th>Shape</th>
<th>Color</th>
<th>Picture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swan</td>
<td>White</td>
<td><img src="swan.png" alt="" /></td>
</tr>
<tr>
<td>Swan</td>
<td>Yellow</td>
<td>. . . .</td>
</tr>
<tr>
<td>Dragon</td>
<td>Yellow</td>
<td><img src="dragon.png" alt="" /></td>
</tr>
<tr>
<td>Car</td>
<td>Yellow</td>
<td>. . . .</td>
</tr>
<tr>
<td>Fish</td>
<td>White</td>
<td>. . . .</td>
</tr>
</tbody>
</table>

Price list

- \( V_1 = \sigma_{\text{Shape}=\text{Swan}}(S) \) \( \text{Price} = \$2 \)
- \( V_2 = \sigma_{\text{Shape}=\text{Dragon}}(S) \) \( \text{Price} = \$2 \)
- \( V_3 = \sigma_{\text{Shape}=\text{Car}}(S) \) \( \text{Price} = \$2 \)
- \( V_4 = \sigma_{\text{Shape}=\text{Fish}}(S) \) \( \text{Price} = \$2 \)
- \( W_1 = \sigma_{\text{Color}=\text{White}}(S) \) \( \text{Price} = \$3 \)
- \( W_2 = \sigma_{\text{Color}=\text{Yellow}}(S) \) \( \text{Price} = \$3 \)
- \( W_3 = \sigma_{\text{Color}=\text{Red}}(S) \) \( \text{Price} = \$3 \)

Price(\(\sigma_{\text{Shape}}\)) = $2
Price(\(\sigma_{\text{Color}}\)) = $3

Get all Dragons for $2
Get all Red Origami for $3

Example 1: Pricing Relational Data

S(Shape, Color, Picture)

<table>
<thead>
<tr>
<th>Shape</th>
<th>Color</th>
<th>Picture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swan</td>
<td>White</td>
<td><img src="image" alt="Swan" /></td>
</tr>
<tr>
<td>Swan</td>
<td>Yellow</td>
<td><img src="image" alt="Swan" /></td>
</tr>
<tr>
<td>Dragon</td>
<td>Yellow</td>
<td><img src="image" alt="Dragon" /></td>
</tr>
<tr>
<td>Car</td>
<td>Yellow</td>
<td><img src="image" alt="Car" /></td>
</tr>
<tr>
<td>Fish</td>
<td>White</td>
<td><img src="image" alt="Fish" /></td>
</tr>
</tbody>
</table>

Price list

<table>
<thead>
<tr>
<th>V_1</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\sigma_{\text{Shape}=\text{Swan}}(S))</td>
<td>$2</td>
</tr>
<tr>
<td>(\sigma_{\text{Shape}=\text{Dragon}}(S))</td>
<td>$2</td>
</tr>
<tr>
<td>(\sigma_{\text{Shape}=\text{Car}}(S))</td>
<td>$2</td>
</tr>
<tr>
<td>(\sigma_{\text{Shape}=\text{Fish}}(S))</td>
<td>$2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>W_1</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\sigma_{\text{Color}=\text{White}}(S))</td>
<td>$3</td>
</tr>
<tr>
<td>(\sigma_{\text{Color}=\text{Yellow}}(S))</td>
<td>$3</td>
</tr>
<tr>
<td>(\sigma_{\text{Color}=\text{Red}}(S))</td>
<td>$3</td>
</tr>
</tbody>
</table>

$1\,?\,\quad$4\,?\,\quad$8\,?\,\quad$20\,?

Price(\(\sigma_{\text{Shape}}\))=$2\quad$Price(\(\sigma_{\text{Color}}\))=$3

Find the price of the entire db

$1\,?\,\quad$4\,?\,\quad$8\,?\,\quad$20\,?

Get all Dragons for $2

Get all Red Origami for $3

## Example 1: Pricing Relational Data

### Price list

<table>
<thead>
<tr>
<th>Shape</th>
<th>Color</th>
<th>Picture</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swan</td>
<td>White</td>
<td><img src="image" alt="Swan" /></td>
<td>$2</td>
</tr>
<tr>
<td>Swan</td>
<td>Yellow</td>
<td>. . . . . .</td>
<td>$2</td>
</tr>
<tr>
<td>Dragon</td>
<td>Yellow</td>
<td><img src="image" alt="Dragon" /></td>
<td>$2</td>
</tr>
<tr>
<td>Car</td>
<td>Yellow</td>
<td>. . . . . .</td>
<td>$2</td>
</tr>
<tr>
<td>Fish</td>
<td>White</td>
<td>. . . . . .</td>
<td>$2</td>
</tr>
</tbody>
</table>

### Price calculation

- \(V_1 = \sigma_{\text{Shape}='Swan'}(S)\) at $2
- \(V_2 = \sigma_{\text{Shape}='Dragon'}(S)\) at $2
- \(V_3 = \sigma_{\text{Shape}='Car'}(S)\) at $2
- \(V_4 = \sigma_{\text{Shape}='Fish'}(S)\) at $2
- \(W_1 = \sigma_{\text{Color}='White'}(S)\) at $3
- \(W_2 = \sigma_{\text{Color}='Yellow'}(S)\) at $3
- \(W_3 = \sigma_{\text{Color}='Red'}(S)\) at $3

### Find the price of the entire db

- \(V_1, V_2, V_3, V_4\) determine \(Q\), \(\text{price}(Q) \leq 8\)
- \(W_1, W_2, W_3\) determine \(Q\), \(\text{price}(Q) \leq 9\)

To ensure arbitrage-freeness, we can charge only $8 for the entire database.

Example 1: Pricing Relational Data

<table>
<thead>
<tr>
<th>Shape</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swan</td>
<td>Fold, fold, fold, ...</td>
</tr>
<tr>
<td>Dragon</td>
<td>Cut, cut, cut, ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shape</th>
<th>Color</th>
<th>Picture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swan</td>
<td>White</td>
<td><img src="image" alt="Swan" /></td>
</tr>
<tr>
<td>Swan</td>
<td>Yellow</td>
<td>. . . . .</td>
</tr>
<tr>
<td>Dragon</td>
<td>Yellow</td>
<td><img src="image" alt="Dragon" /></td>
</tr>
<tr>
<td>Car</td>
<td>Yellow</td>
<td>. . . . .</td>
</tr>
<tr>
<td>Fish</td>
<td>White</td>
<td>. . . . .</td>
</tr>
</tbody>
</table>

Find the price of the full join: \( Q = R \bowtie S \bowtie T \)

Find the price of the full join: 

- \( \text{Price}(\sigma_{\text{Shape}}) = \$99 \)
- \( \text{Price}(\sigma_{\text{Shape}}) = \$2 \)
- \( \text{Price}(\sigma_{\text{Color}}) = \$3 \)
- \( \text{Price}(\sigma_{\text{Color}}) = \$55 \)

Example 1: Pricing Relational Data

Find the price of the full join:  \( Q = R \bowtie S \bowtie T \)

<table>
<thead>
<tr>
<th>Shape</th>
<th>Instructions</th>
<th>Color</th>
<th>Picture</th>
<th>PaperSpecs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swan</td>
<td>Fold, fold, fold…</td>
<td>White</td>
<td><img src="image" alt="Swan" /></td>
<td>15g/100</td>
</tr>
<tr>
<td>Dragon</td>
<td>Cut, cut, cut…</td>
<td>Yellow</td>
<td><img src="image" alt="Dragon" /></td>
<td></td>
</tr>
<tr>
<td>Car</td>
<td>Yellow</td>
<td>. . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fish</td>
<td>White</td>
<td>. . . .</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Robustness and Accuracy:
- Price(\( \sigma_{\text{Shape}} \)) = $99
- Price(\( \sigma_{\text{Shape}} \)) = $2
- Price(\( \sigma_{\text{Color}} \)) = $3
- Price(\( \sigma_{\text{Color}} \)) = $55

### Example 1: Pricing Relational Data

Find the price of the full join: \( Q = R \bowtie S \bowtie T \)

<table>
<thead>
<tr>
<th>Shape</th>
<th>Instructions</th>
<th>Color</th>
<th>Picture</th>
<th>PaperSpecs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swan</td>
<td>Fold,fold,fold,...</td>
<td>White</td>
<td><img src="https://example.com/swan.png" alt="Swan.png" /></td>
<td>15g/100</td>
</tr>
<tr>
<td>Dragon</td>
<td>Cut,cut,cut,...</td>
<td>Yellow</td>
<td><img src="https://example.com/dragon.png" alt="Dragon.png" /></td>
<td></td>
</tr>
<tr>
<td>Car</td>
<td></td>
<td>Yellow</td>
<td><img src="https://example.com/car.png" alt="Car.png" /></td>
<td></td>
</tr>
<tr>
<td>Fish</td>
<td></td>
<td>White</td>
<td><img src="https://example.com/fish.png" alt="Fish.png" /></td>
<td></td>
</tr>
</tbody>
</table>

Not obvious! E.g. no Yellow Cars in the join.

What to pay for?
- \( \sigma_{\text{Shape='car'}}(R) \) or
- \( \sigma_{\text{Color='yellow'}}(T) \)

Why not charge per row in the answer?

- $Q_1(x,y) = \text{Fortune500}(x,y)$
  - $Q(x,y) = \text{Fortune500}(x,y), \text{StrongBuyRec}(x)$
- $Q \subseteq Q_1$, yet $\text{Price}(Q) \gg \text{Price}(Q_1)$
- “Containment” is unrelated to pricing
- “Determinacy” is the right concept for studying pricing
Example 2: Pricing Private Data

- **Buyer:** query \( c = x_1 + x_2 + \ldots + x_{1000} \)
- **User compensation:** $10
- **Price for the buyer:** $10,000

<table>
<thead>
<tr>
<th>UID</th>
<th>User</th>
<th>Rating (0..5)</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alice</td>
<td>3</td>
<td>$10</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
<td>0</td>
<td>$10</td>
</tr>
<tr>
<td>3</td>
<td>Carol</td>
<td>1</td>
<td>$10</td>
</tr>
<tr>
<td>4</td>
<td>Dan</td>
<td>0</td>
<td>$10</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>1000</td>
<td>Zoran</td>
<td>2</td>
<td>$10</td>
</tr>
</tbody>
</table>

1. Raw data is too expensive!
Example 2: Pricing Private Data

Differential privacy
- Perturbation is necessary for privacy [Dwork’2011]

Selling private data
- Perturbation is a cost saving feature
- Two extremes:
  - Raw data = no perturbation = high price
  - Differentially private = high perturbation = low price
Example 2: Pricing Private Data

- Buyer: \( c = x_1 + x_2 + \ldots + x_{1000} \)
  - Tolerates error \( \pm 300 \)
  - Equivalently: variance \( v = 5000 \)

- Answer: \( \hat{c} = c + \text{Lap}(\sqrt{(v/2)}) \)

- User compensation: $10, $0.001 (query is 0.1-DP)

- Price for the buyer: $10,000, $1

\*Probability(|\( \hat{c} - c \)| \( \geq 3 \sqrt{2} \) \( \sigma \) < 1/18=0.056 (Chebyshev), where \( \sigma = \sqrt{v} = 50\sqrt{2} \)

\*\* \( \varepsilon = \sqrt{2} \) sensitivity(q)/\( \sigma = 5\sqrt{2} / 50\sqrt{2} = 0.1 \)
Example 2: Pricing Private Data

- Another buyer: $c = x_1 + x_2 + \ldots + x_{1000}$
  - Zero error, error $\pm 300$ error $\pm 30$
  - Variance $= 0$, variance $= 5000$ variance $= 50$
- User compensation: $\$10/\text{item}, \$0.001/\text{item}$ $\$0.1/\text{item?}$ $\$1/\text{item?}$
- Price for the buyer: $\$10000, \$1$ $\$100? \$1000?$
  - If price $> \$100 \rightarrow$ arbitrage!
    Buy $100 \times$ queries with variance $5000$, take average. Cost $= 100 \times \$1$.

3. Multiple queries: must be arbitrage-free.
Outline

• Framework and examples

• Results so far

• Conclusions
Price of Relational Queries

**Given:** Price points \((V_1, p_1), \ldots, (V_k, p_k)\)

Database D

Arbitrary query Q.

**Compute:** \(\text{Price}_D(Q)\)

Must ensure this:

**Arbitrage-freeness:** For all queries, if \(Q_1, \ldots, Q_k\) determine Q
then \(\text{price}_D(Q) \leq \text{price}_D(Q_1) + \ldots + \text{price}_D(Q_k)\)
Price of Relational Queries

- Simple algorithm for computing $\text{price}_D(Q)$ given an oracle for checking determinacy.
- Two options for determinacy:
  - Instance-independent: used by RDBMS today in query-answering using views; undecidable!
  - Instance-dependent: seems more natural for pricing; $\Pi^p_2$ in the database.
- If (a) price-points $(V_i, p_i)$ are selection queries, and (b) $Q$ is a Union of Conjunctive Queries then $\text{price}_D(Q)$ is NP-complete in the database.
- Reduction to ILP makes pricing (almost) practical.
Price of Relational Queries

ILP construction time (100)  Total time (100)
ILP construction time (1000) Total time (1000)
## Compensation for Private Data

<table>
<thead>
<tr>
<th>UID</th>
<th>User</th>
<th>Rating (0..5)</th>
<th>Pay</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alice</td>
<td>3</td>
<td>$10</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
<td>0</td>
<td>$10</td>
</tr>
<tr>
<td>3</td>
<td>Carol</td>
<td>1</td>
<td>$10</td>
</tr>
<tr>
<td>4</td>
<td>Dan</td>
<td>0</td>
<td>$10</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1000</td>
<td>Zoran</td>
<td>2</td>
<td>$10</td>
</tr>
</tbody>
</table>

Query $c = x_1 + x_2 + \ldots + x_{1000}$
Variance $v = 50$

How much should we pay Carol?
Compensation for Private Data

<table>
<thead>
<tr>
<th>UID</th>
<th>User</th>
<th>Rating (0..5)</th>
<th>Compensation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alice</td>
<td>3</td>
<td>$10</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
<td>0</td>
<td>$10</td>
</tr>
<tr>
<td>3</td>
<td>Carol</td>
<td>1</td>
<td>$10</td>
</tr>
<tr>
<td>4</td>
<td>Dan</td>
<td>0</td>
<td>$10</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1000</td>
<td>Zoran</td>
<td>2</td>
<td>$10</td>
</tr>
</tbody>
</table>

Query $c = x_1 + x_2 + \ldots + x_{1000}$

Variance $v = 50$

How much should we pay Carol?

Differential Privacy

**Def.** [Dwork’11] **Fix** $\varepsilon$. Mechanism $\hat{\ell}$ is called $\varepsilon$-differential private, if for all $D, D'$ that differ in one item, and any set $S$

$$P[\hat{\ell}(D) \in S] \leq \exp(\varepsilon) \times P[\hat{\ell}(D') \in S]$$
Compensation for Private Data

<table>
<thead>
<tr>
<th>UID</th>
<th>User</th>
<th>Rating (0..5)</th>
<th>Compensation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alice</td>
<td>3</td>
<td>$10</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
<td>0</td>
<td>$10</td>
</tr>
<tr>
<td>3</td>
<td>Carol</td>
<td>1</td>
<td>$10</td>
</tr>
<tr>
<td>4</td>
<td>Dan</td>
<td>0</td>
<td>$10</td>
</tr>
<tr>
<td>1000</td>
<td>Zoran</td>
<td>2</td>
<td>$10</td>
</tr>
</tbody>
</table>

How much should we pay Carol?

Query $c = x_1 + x_2 + \ldots + x_{1000}$
Variance $v = 50$

Differential Privacy

**Def.** [Dwork’11] Fix $\varepsilon$. Mechanism $\hat{c}$ is called $\varepsilon$-differential private, if for all $D, D'$ that differ in one item, and any set $S$

$P[\hat{c}(D) \in S] \leq \exp(\varepsilon) \times P[\hat{c}(D') \in S]$

**Thm.** The mechanism $\hat{c}(D) = c(D) + \text{Lap}(\Delta c/\varepsilon)$ is $\varepsilon$-differential private

Variance $v = 2(\Delta c/\varepsilon)^2$

Carol gets no money!
Compensation for Private Data

<table>
<thead>
<tr>
<th>UID</th>
<th>User</th>
<th>Rating (0..5)</th>
<th>Pay</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alice</td>
<td>3</td>
<td>$10</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
<td>0</td>
<td>$10</td>
</tr>
<tr>
<td>3</td>
<td>Carol</td>
<td>1</td>
<td>$10</td>
</tr>
<tr>
<td>4</td>
<td>Dan</td>
<td>0</td>
<td>$10</td>
</tr>
<tr>
<td>1000</td>
<td>Zoran</td>
<td>2</td>
<td>$10</td>
</tr>
</tbody>
</table>

Thm. The mechanism $\hat{\epsilon}(D) = c(D) + \text{Lap}(\Delta c/\epsilon)$ is $\epsilon$-differential private.

How much should we pay Carol?

Data Pricing
Fix variance $v$

Def. Carol’s privacy loss is $\epsilon(v) = \sup_S \log(P[\hat{\epsilon}(D) \in S]/P[\hat{\epsilon}(D') \in S])$

W($\epsilon$) = Carol’s valuation function

Variance $v=2(\Delta c/\epsilon)^2$

Carol gets no money!
Compensation for Private Data

Incentivizing Carol to reveal her valuation $W(\varepsilon)$ is difficult! [Ghosh’11, Gkatzelis’12, Riederer’12]
We use an idea from [Aperjis&Huberman’11]:

- **Option A**: risk neutral
- **Option B**: risk averse
- **Option C**: opt-out

Incentivizing Carol to reveal her valuation $W(\varepsilon)$ is difficult! [Ghosh’11, Gkatzelis’12, Riederer’12]
We use an idea from [Aperjis&Huberman’11]:

- **Option A**: risk neutral
- **Option B**: risk averse
- **Option C**: opt-out
Compensation for Private Data

Incentivizing Carol to reveal her valuation $W(\varepsilon)$ is difficult! [Ghosh’11, Gkatzelis’12, Riederer’12]

We use an idea from [Aperjis&Huberman’11]:

- **Option A**: risk neutral
- **Option B**: risk averse
- **Option C**: opt-out

Risk-averse users count on the fact that most queries will have low privacy leak
Compensation for Private Data

Incentivizing Carol to reveal her valuation $W(\varepsilon)$ is difficult! [Ghosh’11, Gkatzelis’12, Riederer’12]

We use an idea from [Aperjis&Huberman’11]:

- **Option A**: risk neutral
- **Option B**: risk averse
- **Option C**: opt-out

Risk-neutral users want full compensation at the risk of never being paid.

Risk-averse users count on the fact that most queries will have low privacy leak.
Outline

• Framework and examples

• Results so far

• Conclusions
The Third Wave of Computing

• First wave = hardware
  – IBM, DEC, Sun, …
  – 1950 – 1980

• Second wave = software
  – Microsoft, Borland, Fox Software, Oracle, …
  – 1980 -- 2010

• Third wave = data!
  – Google maps v.s. IOS maps
  – Facebook’s users
Conclusions

• Data has (lots of) value!
• Pricing data: at the intersection of three areas:
  – Data management
  – Mechanism design
  – Economics
• Key concepts:
  – Arbitrage-free
  – Compensation = function of privacy loss
References

• Koutris et al., PODS, 2012

• Li et al., ICDT, 2013

• Koutris et al, under review