Database Uncertainty
for Computational Social Choice

Benny Kimelfeld
Technion Data & Knowledge Lab
tdk.net.technion.ac.il
Outline

- Background: Database Uncertainty
- Background: Voting Theory
- Databases for Social Choice
- Analysis of Computational Complexity
- Directions
Database System
Query language (logic) for high-level what, system for low-level how

Enterprise Data

Public Data

No explicit structure
Uncontrolled natural lang.
Basic needs require NLP/ML/AI/statistics
Incomplete, imprecise, inconsistent

Cloud
Opensource
Crowd
Uncertainty in Databases

Inconsistent Data
- Models of repairs
- Consistent query answering
- Entity resolution
- Data cleaning

Probabilistic Data Exchange
- Models of probabilistic DB
- Query inference
- Randomness in queries

Probabilistic Databases

Incomplete Information
- Models of null
- Certain answers (vs. 3VL)
- Data exchange

Common theoretical ground: possible worlds
Incomplete Information

- [Lipski 79]
- [Imielinski and Lipski 84]

<table>
<thead>
<tr>
<th>Registrations</th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>student</td>
<td>course</td>
</tr>
<tr>
<td>Erin</td>
<td>Logic</td>
</tr>
<tr>
<td>Oliver</td>
<td>⊥₁</td>
</tr>
<tr>
<td>Emma</td>
<td>⊥₂</td>
</tr>
</tbody>
</table>

Incomplete DB

Impossible answers

Certain answers

Completed DB
Inconsistent Data

[Arenas, Bertossi, Chomicki 99]

- Who got at least 85 in Logic?
- … at least 90?
- … at least 80?

Functional Dependency:

\[ \text{[student course]} \rightarrow \text{grade} \]

Consistent answers

\[ \text{Repairs} \]

<table>
<thead>
<tr>
<th>student</th>
<th>course</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Erin</td>
<td>Logic</td>
<td>90</td>
</tr>
<tr>
<td>Oliver</td>
<td>Logic</td>
<td>86</td>
</tr>
<tr>
<td>Oliver</td>
<td>Logic</td>
<td>81</td>
</tr>
</tbody>
</table>
Probabilistic Databases

[Sato, Kameya 97]
[Suciu, Dalvi 05]

- Who got at least 85?
  - Erin (1.0); Oliver (0.3)

<table>
<thead>
<tr>
<th>student</th>
<th>course</th>
<th>grade</th>
<th>Pr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Erin</td>
<td>Logic</td>
<td>90</td>
<td>1.0</td>
</tr>
<tr>
<td>Oliver</td>
<td>Logic</td>
<td>86</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>81</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Probabilistic DB

```
Pr[\{t \in a\} = \sum_{t \in a_i} p_i
```

Tuple confidence
• Inconsistent data
  – Find the “best” repairs [PODS’18]
  – Count & generate repairs [PODS’17]
  – Repair with tuple preferences [ICDT’17]

• Probabilistic databases
  – Probabilistic data cleaning [Corr’18]
  – Probabilistic programming with Datalog [TODS’17]
  – Databases for probabilistic preferences [PODS’17, SIGMOD’18, AAAI’18]

• Incomplete information
  – Databases for social choice [IJCAI’18] (next…)
Outline

- Background: Database Uncertainty
- Background: Voting Theory
- Databases for Social Choice
- Analysis of Computational Complexity
- Directions
• Intersecting CS & Economics
• Computational aspects of collective decision making
  – Automated reasoning about election outcomes
  – Fair division of public commodities under individual preferences
  – Market design: demand-supply matching
  – Collaborative ranking

• C.f. Handbook of Computational Social Choice
  – Brandt, Conitzer, Endriss, Lang, Procaccia (2016)
Votes and Incomplete Preferences

Candidates

Voting policy, e.g., scoring rule

Voters

Is Trump surely winning? (necessary winners)

Can Clinton win? (possible winners)

“2-approval”
Formalism

• Candidates: $c_1, \ldots, c_m$
• Voters: $v_1, \ldots, v_n$
• A **preference profile** is a vector $(\succ_1, \ldots, \succ_n)$ of linear orders over the candidates by the voters $v_1, \ldots, v_n$
• A **voting rule** maps a preference profile to a set of winners among the candidates
  – E.g., *plurality* – winners are those 1st place most frequently
• A **positional scoring rule** is a special case of voting rule
Positional Scoring Rule

- Candidates: $c_1, \ldots, c_m$; Voters: $v_1, \ldots, v_n$
- A scoring vector is a sequence $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m$ of $m$ natural numbers ($\sigma_1 > \sigma_m$, gcd = 1)
- Voter $v_i$ assigns to candidates scores according to $\succ_i$

\[
\begin{array}{cccccccc}
& c_{i_1} & \succ_i & c_{i_2} & \succ_i & c_{i_3} & \succ_i & c_{i_4} & \succ_i & c_{i_5} & \succ_i & \cdots & \succ_i & c_{i_m} \\
\sigma_1 & \sigma_2 & \sigma_3 & \sigma_4 & \sigma_5 & \cdots & \sigma_m
\end{array}
\]

- Winners: maximal sum of scores
  - Various variants: single winner, multi winner, tie breaking, …
- Scoring rule $r(m,i)$ defines $i$th number for $m$ candidates
  - We assume that $r(m,i)$ is computable in time $\text{poly}(m,i)$
Popular Scoring Rules

- **Plurality**
  \[
  \begin{array}{cccccc}
  1 & 0 & 0 & 0 & 0 & \cdots & 0 \\
  \sigma_1 & \sigma_2 & \sigma_3 & \sigma_4 & \sigma_5 & \sigma_m \\
  \end{array}
  \]

- **Borda**
  \[
  \begin{array}{cccc}
  m-1 & m-2 & \cdots & 1 & 0 \\
  \end{array}
  \]

- **Veto**
  \[
  \begin{array}{cccccc}
  1 & \cdots & 1 & 1 & 1 & 1 & 0 \\
  \end{array}
  \]

- **k-Approval**
  \[
  \begin{array}{cccccc}
  1 & \cdots & 1 & 0 & 0 & \cdots & 0 \\
  \end{array}
  \]

- **k-Veto**
  \[
  \begin{array}{cccccc}
  1 & \cdots & 1 & 1 & 0 & \cdots & 0 \\
  \end{array}
  \]
Necessary & Possible Winners in Partial Profiles

• Recall: a **preference profile** is a vector \((\succ_1, \ldots, \succ_n)\) of linear orders over the candidates by the voters \(v_1, \ldots, v_n\)

• In a **partial preference profile**, each \(\succ_i\) is a partial order

• A **completion** of a partial preference profile is obtained by completing each \(\succ_i\) to a linear order \(\succ'_i\)
  
  – Hence, \((\succ'_1, \ldots, \succ'_n)\) is a complete preference profile

• [Konczak & Lang 05] A candidate \(c_i\) is a:
  
  – **necessary winner** if wins in *every* completion
  
  – **possible winner** if wins in *at least one* completion

• Naively, nw/pw computation requires all completions
Complexity of Necessary/Possible Winners

- The **necessary** winners can be computed in **polynomial time** under all positional scoring rules
  - [XiaConitzer2011]

- Computing the **possible** winners is:
  - In polynomial time for **plurality** and **veto**
  - **NP-complete** for every other pure* rule
    - [KonczakLang05], [BetzleDorn2010], [XiaConitzer2011], [BaumeisterRothe2012]

**pure:** (vector of m) = (vector of m-1) + (new inserted value)
Outline

- Background: Database Uncertainty
- Background: Voting Theory
- Databases for Social Choice
- Analysis of Computational Complexity
- Directions
Votes and Incomplete Preferences

Candidates

Voters

Voting policy, e.g., scoring rule

Is Trump surely winning? (necessary winners)

Can Clinton win? (possible winners)

“2-approval”
### Candidates

<table>
<thead>
<tr>
<th>cand</th>
<th>party</th>
<th>net w</th>
<th>spouse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinton</td>
<td>D</td>
<td>$45M</td>
<td>Bill</td>
</tr>
<tr>
<td>Trump</td>
<td>R</td>
<td>#1.3B</td>
<td>Melania</td>
</tr>
<tr>
<td>Cruz</td>
<td>R</td>
<td>$3.5M</td>
<td>Heidi</td>
</tr>
<tr>
<td>Sanders</td>
<td>D</td>
<td>$0.8M</td>
<td>Jane</td>
</tr>
</tbody>
</table>

### Under which scoring rules do Democrats necessarily win?

- Is it possible to have a winner of net worth < $10M?
- Is it possible to have co-winners from the same city?
- Can the winner be low-ranked by all young voters?

### Voters

<table>
<thead>
<tr>
<th>person</th>
<th>born</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinton</td>
<td>Chicago</td>
</tr>
<tr>
<td>Trump</td>
<td>NYC</td>
</tr>
<tr>
<td>Cruz</td>
<td>Calgary</td>
</tr>
<tr>
<td>Sanders</td>
<td>NYC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>voter</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Susan</td>
<td>45</td>
</tr>
<tr>
<td>David</td>
<td>62</td>
</tr>
<tr>
<td>James</td>
<td>29</td>
</tr>
</tbody>
</table>
Context Matters!

Is a Republican winning every completion?

<table>
<thead>
<tr>
<th>cand</th>
<th>party</th>
<th>net w</th>
<th>spouse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinton</td>
<td>D</td>
<td>$45M</td>
<td>Bill</td>
</tr>
<tr>
<td>Trump</td>
<td>R</td>
<td>#1.3B</td>
<td>Melania</td>
</tr>
<tr>
<td>Cruz</td>
<td>R</td>
<td>$3.5M</td>
<td>Heidi</td>
</tr>
<tr>
<td>Sanders</td>
<td>D</td>
<td>$0.8M</td>
<td>Jane</td>
</tr>
</tbody>
</table>

Voters

No one is a “necessary winner” … but the winner is necessarily R

1 0 0 0 0

“plurality”
Q(v) :- Pref('p1', v, x, y), BornIn(x, 'NYC'), BornIn(y, 'LA')
Hard Inference Problems!

- Framework & complexity analysis  [Jacob, K, Stoyanovich VLDB14]  [Kenig, Ping, K, Stoyanovich PODS17]
- System implementation  [Cohen, Kenig, Ping, K, Stoyanovich SIGMOD18]  [Kenig, Ilijasic, Ping, K, Stoyanovich AAAI18]
# Simpler View for Voting

[K, Kolaitis, Stoyanovich, IJCAI 2018]

## Candidates

<table>
<thead>
<tr>
<th>cand</th>
<th>party</th>
<th>net w</th>
<th>spouse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinton</td>
<td>D</td>
<td>$45M</td>
<td>Bill</td>
</tr>
<tr>
<td>Trump</td>
<td>R</td>
<td>#1.3B</td>
<td>Melania</td>
</tr>
<tr>
<td>Cruz</td>
<td>R</td>
<td>$3.5M</td>
<td>Heidi</td>
</tr>
<tr>
<td>Sanders</td>
<td>D</td>
<td>$0.8M</td>
<td>Jane</td>
</tr>
</tbody>
</table>

## BornIn

<table>
<thead>
<tr>
<th>person</th>
<th>born</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinton</td>
<td>Chicago</td>
</tr>
<tr>
<td>Trump</td>
<td>NYC</td>
</tr>
<tr>
<td>Cruz</td>
<td>Calgary</td>
</tr>
<tr>
<td>Sanders</td>
<td>NYC</td>
</tr>
</tbody>
</table>

## Voters

<table>
<thead>
<tr>
<th>voter</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Susan</td>
<td>45</td>
</tr>
<tr>
<td>David</td>
<td>62</td>
</tr>
<tr>
<td>James</td>
<td>29</td>
</tr>
<tr>
<td>cand</td>
<td>party</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>Clinton</td>
<td>D</td>
</tr>
<tr>
<td>Trump</td>
<td>R</td>
</tr>
<tr>
<td>Cruz</td>
<td>R</td>
</tr>
<tr>
<td>Sanders</td>
<td>D</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>person</th>
<th>born</th>
<th>voter</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinton</td>
<td>Chicago</td>
<td>Susan</td>
<td>45</td>
</tr>
<tr>
<td>Trump</td>
<td>NYC</td>
<td>David</td>
<td>62</td>
</tr>
<tr>
<td>Cruz</td>
<td>Calgary</td>
<td>James</td>
<td>29</td>
</tr>
<tr>
<td>Sanders</td>
<td>NYC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Completion 1**

<table>
<thead>
<tr>
<th>cand</th>
<th>party</th>
<th>net w</th>
<th>spouse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinton</td>
<td>D</td>
<td>$45M</td>
<td>Bill</td>
</tr>
<tr>
<td>Trump</td>
<td>R</td>
<td>$1.3B</td>
<td>Melania</td>
</tr>
<tr>
<td>Cruz</td>
<td>R</td>
<td>$3.5M</td>
<td>Heidi</td>
</tr>
<tr>
<td>Sanders</td>
<td>D</td>
<td>$0.8M</td>
<td>Jane</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>person</th>
<th>born</th>
<th>voter</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinton</td>
<td>Chicago</td>
<td>Susan</td>
<td>45</td>
</tr>
<tr>
<td>Trump</td>
<td>NYC</td>
<td>David</td>
<td>62</td>
</tr>
<tr>
<td>Cruz</td>
<td>Calgary</td>
<td>James</td>
<td>29</td>
</tr>
<tr>
<td>Sanders</td>
<td>NYC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Completion 2**

<table>
<thead>
<tr>
<th>cand</th>
<th>party</th>
<th>net w</th>
<th>spouse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinton</td>
<td>D</td>
<td>$45M</td>
<td>Bill</td>
</tr>
<tr>
<td>Trump</td>
<td>R</td>
<td>$1.3B</td>
<td>Melania</td>
</tr>
<tr>
<td>Cruz</td>
<td>R</td>
<td>$3.5M</td>
<td>Heidi</td>
</tr>
<tr>
<td>Sanders</td>
<td>D</td>
<td>$0.8M</td>
<td>Jane</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>person</th>
<th>born</th>
<th>voter</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinton</td>
<td>Chicago</td>
<td>Susan</td>
<td>45</td>
</tr>
<tr>
<td>Trump</td>
<td>NYC</td>
<td>David</td>
<td>62</td>
</tr>
<tr>
<td>Cruz</td>
<td>Calgary</td>
<td>James</td>
<td>29</td>
</tr>
<tr>
<td>Sanders</td>
<td>NYC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Completion 3**

<table>
<thead>
<tr>
<th>cand</th>
<th>party</th>
<th>net w</th>
<th>spouse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinton</td>
<td>D</td>
<td>$45M</td>
<td>Bill</td>
</tr>
<tr>
<td>Trump</td>
<td>R</td>
<td>$1.3B</td>
<td>Melania</td>
</tr>
<tr>
<td>Cruz</td>
<td>R</td>
<td>$3.5M</td>
<td>Heidi</td>
</tr>
<tr>
<td>Sanders</td>
<td>D</td>
<td>$0.8M</td>
<td>Jane</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>person</th>
<th>born</th>
<th>voter</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinton</td>
<td>Chicago</td>
<td>Susan</td>
<td>45</td>
</tr>
<tr>
<td>Trump</td>
<td>NYC</td>
<td>David</td>
<td>62</td>
</tr>
<tr>
<td>Cruz</td>
<td>Calgary</td>
<td>James</td>
<td>29</td>
</tr>
<tr>
<td>Sanders</td>
<td>NYC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Completion 4**

<table>
<thead>
<tr>
<th>cand</th>
<th>party</th>
<th>net w</th>
<th>spouse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinton</td>
<td>D</td>
<td>$45M</td>
<td>Bill</td>
</tr>
<tr>
<td>Trump</td>
<td>R</td>
<td>$1.3B</td>
<td>Melania</td>
</tr>
<tr>
<td>Cruz</td>
<td>R</td>
<td>$3.5M</td>
<td>Heidi</td>
</tr>
<tr>
<td>Sanders</td>
<td>D</td>
<td>$0.8M</td>
<td>Jane</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>person</th>
<th>born</th>
<th>voter</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinton</td>
<td>Chicago</td>
<td>Susan</td>
<td>45</td>
</tr>
<tr>
<td>Trump</td>
<td>NYC</td>
<td>David</td>
<td>62</td>
</tr>
<tr>
<td>Cruz</td>
<td>Calgary</td>
<td>James</td>
<td>29</td>
</tr>
<tr>
<td>Sanders</td>
<td>NYC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Completion 5**

- 
- 
-
Incomplete Preference DB

Completions

\( Q \rightarrow a_1 \)
\( Q \rightarrow a_2 \)
\( Q \rightarrow a_3 \)
\( Q \rightarrow a_4 \)

\( \text{na}_i \) Necessary answers

\( \text{ua}_i \) Possible answers
Necessary & Possible Answers

- Let $D$ be a preference database
- Let $Q$ be a query, may involve the $\text{WINNER}$ relation
- A necessary answer is a tuple that belongs to $Q(C)$ for every completion $C$
- A possible answer is a tuple that belongs to $Q(C)$ for at least one completion $C$
Examples of Queries

- Is there surely a Republican winner?
  \[ Q() : \textbf{WINNER}(x), \text{Party}(x,'R') \] [necessary]

- Which cities are guaranteed to have winners from?
  \[ Q(c) : \textbf{WINNER}(x), \text{LivesIn}(x,c) \] [necessary]

- Possible to have a winner of net worth < $10M?
  \[ Q() : \textbf{WINNER}(x), \text{NetW}(x,w), w<10M \] [possible]

- Can co-winners disagree on pro-choice?
  \[ Q() : \textbf{WINNER}(x), \textbf{WINNER}(y), \text{Yes}(x,'pc'), \text{No}(y,'pc') \] [possible]

- Can we avoid having a couple of married co-winners?
  \[ Q() : \textbf{WINNER}(x), \textbf{WINNER}(y), \text{Married}(x,y) \] [¬ necessary]
Outline

- Background: Database Uncertainty
- Background: Voting Theory
- Databases for Social Choice
- Analysis of Computational Complexity
- Directions
Data Complexity

Incomplete Preference DB

Input (partial profile + database)

Fixed!

Algorithm

Every Q is a separate computational problem

Necessary answers

Possible answers

Q → a1
Q → a2
Q → a3
Q → a4

exp!

na_i

ua_i

Algorithm

tuples

yes/no (Boolean)
Conjunctive Queries

- We focus on 2 classes of queries:
  - Conjunctive Query (CQ)
    \[ y : \exists x [ \varphi_1(x, y) \land \cdots \land \varphi_k(x, y) ] \]
    - Each \( \varphi_i(x, y) \) is atomic: DB schema or \textsc{Winner}
    - A.k.a. \texttt{SELECT-PROJECT-JOIN}
  - Union of Conjunctive Queries (UCQ)
    \[ y : Q_1(y) \lor \cdots \lor Q_l(y) \]
    - Each \( Q_i(y) \) is a CQ
    - A.k.a. \texttt{UNION-SELECT-PROJECT-JOIN} or positive relational algebra
Outline

- Background: Database Uncertainty
- Background: Voting Theory
- Databases for Social Choice
- Analysis of Computational Complexity
  - Necessary Answers
  - Possible Answers
- Directions
THEOREM:

The following holds under plurality & veto rules:

• If the $\text{Winner}$ atoms are disconnected, then the necessary answers can be found in poly. Time
  – Generalizes to unions of disconnected CQs

• For every CQ with 2 connected $\text{Winner}$ atoms and no repeated ordinary relations, coNP-complete to determine whether necessary answers exist
Connectivity of Winner Atoms

Connected Winner atoms

\[ Q() \leftarrow \text{Winner}(x), \text{Winner}(y), \text{Relative}(x,y) \]

\[ \text{Winner}(x) \quad \text{Relative}(x,y) \quad \text{Winner}(y) \]

Shared \( \exists \) variables

Disconnected Winner atoms

\[ Q() \leftarrow \text{Winner}(x), \text{Winner}(y), \text{Yes}(x,'proC'), \text{No}(y,'proC') \]

\[ \text{Winner}(x) \quad \text{Yes}(x,'proC') \quad \text{Winner}(y) \quad \text{No}(y,'proC') \]

\[ Q() \leftarrow \text{Winner}(x), \text{Party}(x,'D') \]

\[ \text{Winner}(x) \quad \text{Party}(x,'D') \]

\[ Q(c) \leftarrow \text{Winner}(x), \text{Winner}(y), \text{Works}(x,c), \text{Board}(y,c) \]

\[ \text{Winner}(x) \quad \text{Works}(x,c) \quad \text{Board}(y,c) \]

\[ \text{Winner}(y) \quad \text{Works}(x,c) \quad \text{Board}(y,c) \]

Shared free (output) variable
Connectivity of Winner Atoms

**Connected Winner atoms**

\[ Q() \, : \, \text{WINNER}(x) \, , \, \text{WINNER}(y) \, , \, \text{Relative}(x,y) \]

coNP-complete

\[ Q() \, : \, \text{WINNER}(x) \, , \, \text{WINNER}(y) \, , \, \text{Relative}(x,y) \]

**Disconnected Winner atoms**

\[ Q(c) \, : \, \text{WINNER}(x) \, , \, \text{WINNER}(y) \, , \, \text{Works}(x,c) \, , \, \text{Board}(y,c) \]

Poly. time
Beyond Plurality & Veto

Q() := \text{WINNER}(x), \text{Republican}(x)

\begin{align*}
\text{Plurality: poly.-time} & \quad \text{k-approval: coNP-complete} \\
\begin{array}{cccccc}
1 & 0 & 0 & 0 & 0 & \cdots & 0 \\
\end{array} & \quad
\begin{array}{cccccc}
1 & \cdots & 1 & 0 & 0 & \cdots & 0 \\
\end{array} \\
\begin{array}{cccccc}
1 & \cdots & 1 & 1 & 1 & 1 & 1 & 0 \\
\end{array} & \quad
\begin{array}{cccccc}
1 & \cdots & 1 & 1 & 0 & \cdots & 0 \\
\end{array} \\
\text{Veto: poly.-time} & \quad \text{k-veto: coNP-complete} \\
\end{align*}

\begin{align*}
\text{Borda: coNP-complete} & \\
\begin{array}{cccc}
m-1 & m-2 & \cdots & 1 & 0 \\
\end{array} \\
\end{align*}

\textbf{THEOREM:} Necessity of Q() is \text{coNP-complete} under every pure positional scoring rule except plurality and veto!

\textit{Necessary answers inherit the hardness of the possible winners!}
Hardness beyond Plurality

\(Q() \iff \text{Winner}(x), \text{Winner}(y), \text{Winner}(z), \text{Team}(x,y,z)\)

**Plurality**: \text{coNP-complete}

\[
\begin{array}{cccccc}
1 & 0 & 0 & 0 & 0 & \cdots & 0
\end{array}
\]

**Veto**: \text{coNP-complete}

\[
\begin{array}{cccccc}
1 & \cdots & 1 & 1 & 1 & 1 & 0
\end{array}
\]

**k-approval**: \text{coNP-complete}

\[
\begin{array}{cccccc}
1 & \cdots & 1 & 0 & 0 & \cdots & 0
\end{array}
\]

**k-veto**: \text{coNP-complete}

\[
\begin{array}{cccccc}
1 & \cdots & 1 & 1 & 0 & \cdots & 0
\end{array}
\]

**Borda**: \text{coNP-complete}

\[
\begin{array}{cccc}
m-1 & m-2 & \cdots & 1 & 0
\end{array}
\]

**THEOREM**: The necessity of \(Q()\) is \text{coNP-complete} under every positional scoring rule (pure or not)!

Harder than possible winners!
### Voters in the Reduction

<table>
<thead>
<tr>
<th>Voter</th>
<th>Partial order</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v^1_1$</td>
<td>$x_2 &gt; \neg x_2 &gt; x_1 &gt; \neg x_1 &gt; x_3 &gt; \neg x_3 &gt; x_0$</td>
</tr>
<tr>
<td>$\neg v^1_1$</td>
<td>$\neg x_2 &gt; x_3 &gt; \neg x_1 &gt; x_1 &gt; \neg x_3 &gt; x_0 &gt; x_2$</td>
</tr>
<tr>
<td>$v^3_1$</td>
<td>$x_3 &gt; \neg x_3 &gt; x_1 &gt; \neg x_1 &gt; x_0 &gt; x_1 &gt; \neg x_2$</td>
</tr>
<tr>
<td>$\neg v^3_1$</td>
<td>$\neg x_3 &gt; x_0 &gt; x_2 &gt; \neg x_2 &gt; x_3 &gt; \neg x_3$</td>
</tr>
<tr>
<td>$\neg v^4_1$</td>
<td>${x_1, \neg x_1}$</td>
</tr>
<tr>
<td>$\neg x_2 &gt; x_2 &gt; {x_2, \neg x_2}$</td>
<td></td>
</tr>
<tr>
<td>${x_3, \neg x_3}$</td>
<td></td>
</tr>
<tr>
<td>$\neg x_3 &gt; x_0 &gt; x_1 &gt; \neg x_1 &gt; x_2$</td>
<td></td>
</tr>
<tr>
<td>$\neg x_1 &gt; x_3 &gt; \neg x_3 &gt; x_0$</td>
<td></td>
</tr>
<tr>
<td>$u_1$</td>
<td>$x_1 &gt; \neg x_1 &gt; x_2 &gt; \neg x_2 &gt; x_3 &gt; \neg x_3 &gt; x_0$</td>
</tr>
<tr>
<td>$\neg u_1$</td>
<td>$\neg x_1 &gt; x_2 &gt; \neg x_2 &gt; x_3 &gt; \neg x_3 &gt; x_1 &gt; x_0$</td>
</tr>
<tr>
<td>$u_3$</td>
<td>$x_2 &gt; \neg x_2 &gt; x_3 &gt; \neg x_3 &gt; x_1 &gt; \neg x_1 &gt; x_0$</td>
</tr>
<tr>
<td>$\neg u_3$</td>
<td>$\neg x_2 &gt; x_3 &gt; \neg x_3 &gt; x_1 &gt; \neg x_1 &gt; x_2 &gt; x_0$</td>
</tr>
<tr>
<td>$u_4$</td>
<td>$\neg x_2 &gt; x_3 &gt; \neg x_3 &gt; x_1 &gt; \neg x_1 &gt; x_2 &gt; x_0$</td>
</tr>
<tr>
<td>$\neg u_4$</td>
<td>$\neg x_2 &gt; x_3 &gt; \neg x_3 &gt; x_1 &gt; \neg x_1 &gt; x_2 &gt; x_0$</td>
</tr>
<tr>
<td>$u_5$</td>
<td>$x_3 &gt; \neg x_3 &gt; x_1 &gt; \neg x_1 &gt; x_2 &gt; \neg x_2 &gt; x_0$</td>
</tr>
<tr>
<td>$u_6$</td>
<td>$\neg x_3 &gt; x_1 &gt; \neg x_1 &gt; x_2 &gt; \neg x_2 &gt; x_0$</td>
</tr>
</tbody>
</table>
Outline

- Background: Database Uncertainty
- Background: Voting Theory
- Databases for Social Choice
- Analysis of Computational Complexity
  - Necessary Answers
  - Possible Answers
- Directions
Possible Answers for Plurality and Veto

• Recall: the possible winners can be computed in polynomial time under plurality and veto
  – Hence, the “best” we can hope for is retaining the tractability of plurality and veto
  – We have it for UCQs:

THEOREM: For every UCQ, and under the plurality and veto rules, the possible answers can be computed in polynomial time
Overcoming Hardness

- Small number of candidates (many voters)
- Approximate / probabilistic relaxation (later)
- *Assumptions on the database?*
Outline

- Background: Database Uncertainty
- Background: Voting Theory
- Databases for Social Choice
- Analysis of Computational Complexity
- Directions
Naturally extending the framework, we can include additional voting aspects

- **Under which voting rules Democrats necessarily win?**
  
  \[ Q(r) \text{ :- Rule}(r), \text{Winner}(r,x), \text{Party}(x,'D') \]  
  
  [necessary]

- **Under which scoring rules one can win 2 elections?**
  
  \[ Q(r) \text{ :- Rule}(r), \text{Winner}(r,x,e_1), \text{Winner}(r,x,e_2), e_1 \neq e_2 \]  
  
  [possible]

- **Can a winner be low-ranked by all young voters?**
  
  \[ T(x) \text{ :- Intop}(10,v,x), \text{AgeGroup}(v,'Y') \]  
  
  \[ Q() \text{ :- Winner}(x), \neg T(x) \]  
  
  [possible]
The necessary/possible-winner problems are special cases of the following problem:

What is the probability of winning/losing in a uniformly sampled completion?

Approximations devised
- Additive [Hazon+ 2012], multiplicative [Kenig&K 2018]

Future direction: probabilistic query answering over elections in preference databases
Concluding Remarks

• A framework for extending ComSoc with a DB context
• From poss/nec **winners** to poss/nec **answers**
• Preliminary analysis shows that the extension entails **different complexities**
• Many directions for future research:
  – Richer analysis (query languages, voting rules, data-based tractability properties)
  – Richer modelling (tie-breaking mechanisms, multiple elections, multiple voting rules, combining voter preferences and outcomes)
  – Probabilistic inference (probabilistic voters and databases, approximate inference)
  – Control and bribery (voter manipulation to sat a Boolean query)
Thanks collaborators!

Phokion Kolaitis  Julia Stoyanovich

Uzi Cohen  Batya Kenig  Haoyue Ping  Muhammad Tibi