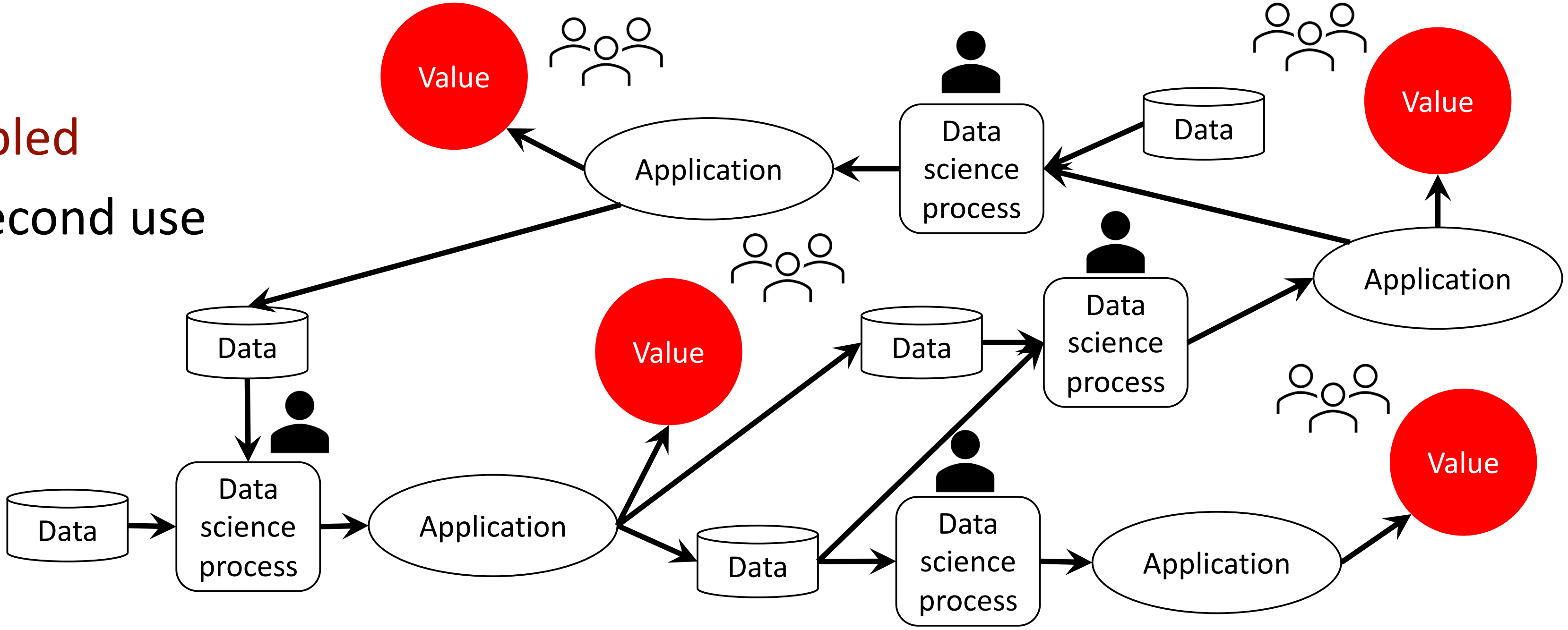
On Shapley Value in Data Assemblage Under Independent Utility

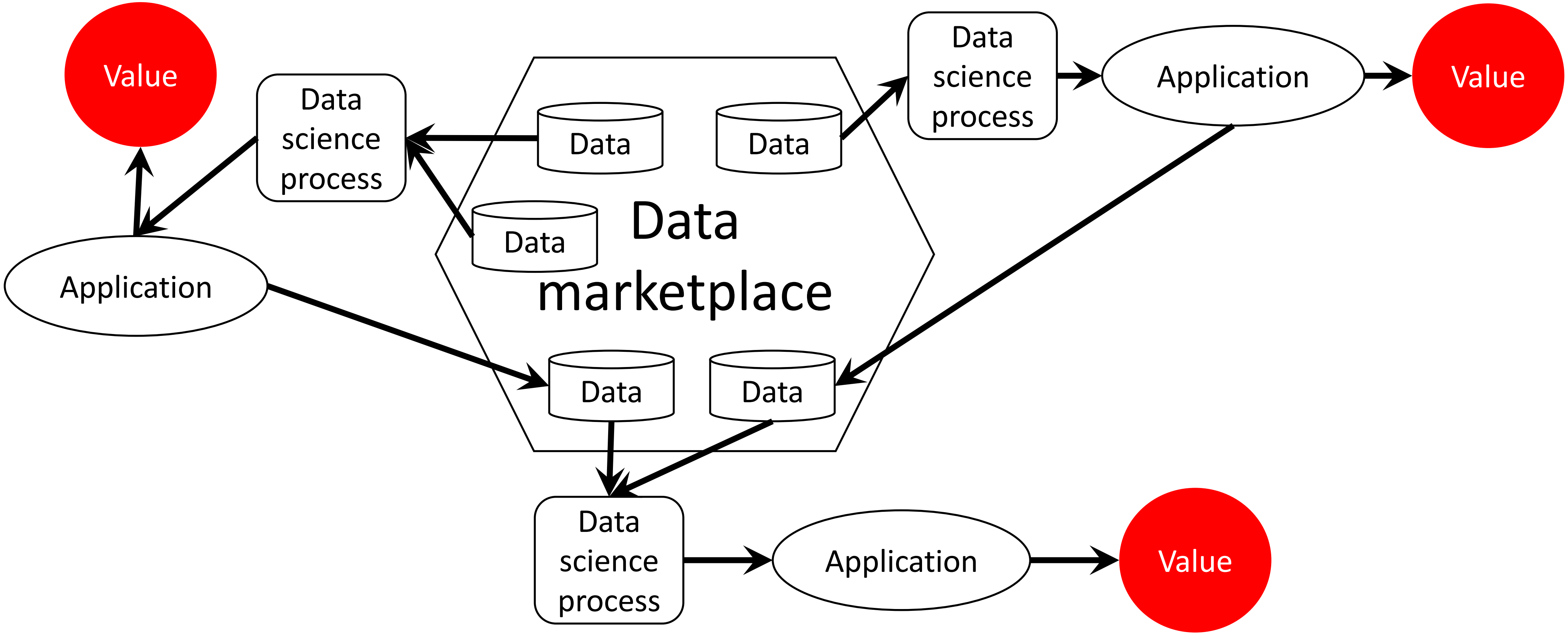
Xuan Luo[1], Jian Pei[1, 2], Zicun Cong[1], and Cheng Xu[1]
[1]Simon Fraser University, Canada
[2]Duke University, United States

Transforming Data into Value

- Challenges:
  - Data are distributed
  - Data may be assembled
  - Data have diverse second use



Data Marketplace



Problem Formulation

- Given a set of data owners $||\mathcal{O}|| = \{o_1, o_2, ..., o_n\}$, a coalition plan $\mathcal{P}$ and a reward from a data buyer. Then how to distribute the reward to the data owners?

o_1

id	name
1	Alice

o_2

id	name
1	Alice

o_3

id	name
2	Kate

o_4

id	department
1	CS
2	Math

o_5

name	department
Kate	Math

Coalition plan $\mathcal{P}$:

$Proj_{name,department}(o_1 \bowtie o_4) \cup$

$Proj_{name,department}(o_2 \bowtie o_4) \cup$

$Proj_{name,department}(o_3 \bowtie o_4) \cup$

o_5

Data Assemblage

Coalition Set

name	department
Alice	CS
Kate	Math

Existing Method

- Shapley Value: given a set of data owners $\|\mathcal{O}\| = \{o_1, o_2, \ldots, o_n\}$,
  - $$\psi(o_i) = \frac{1}{\|\mathcal{O}\|} \sum_{\mathcal{S} \subseteq \mathcal{O} \setminus \{o_i\}} \frac{Utility(\mathcal{S} \cup \{o_i\}) - Utility(\mathcal{S})}{\binom{n-1}{\|\mathcal{S}\|}}$$

o_1

id	name
1	Alice

o_2

id	name
1	Alice

o_3

id	name
2	Kate

o_4

id	department
1	CS
2	Math

o_5

name	department
Kate	Math

Coalition plan $\mathcal{P}$:

$Proj_{name,department}(o_1 \bowtie o_4) \cup$

$Proj_{name,department}(o_2 \bowtie o_4) \cup$

$Proj_{name,department}(o_3 \bowtie o_4) \cup$

o_5

Data Assemblage

Coalition Set

name	department
Alice	CS
Kate	Math

[1] Lloyd S. Shapley. A Value for n-Person Games. Technical Report, RAND Corporation, 1952.

Existing Method

- Shapley Value: given a set of data owners $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$,
  - $$\psi(o_i) = \frac{1}{\|\mathcal{O}\|} \sum_{\mathcal{S} \subseteq \mathcal{O} \setminus \{o_i\}} \frac{Utility(\mathcal{S} \cup \{o_i\}) - Utility(\mathcal{S})}{\binom{n-1}{\|\mathcal{S}\|}}$$
  - E.g. ,
    - $Utility(\varnothing \cup \{o_1\}) - Utility(\varnothing) = 0$
    - $Utility(\{o_2\} \cup \{o_1\}) - Utility(\{o_2\}) = 0$
    - $Utility(\{o_3\} \cup \{o_1\}) - Utility(\{o_3\}) = 0$
    - $Utility(\{o_4\} \cup \{o_1\}) - Utility(\{o_4\}) = 1$
    - $Utility(\{o_5\} \cup \{o_1\}) - Utility(\{o_5\}) = 0$
    - $\ldots$

o_1

id	name
1	Alice

o_2

id	name
1	Alice

o_3

id	name
2	Kate

o_4

id	department
1	CS
2	Math

o_5

name	department
Kate	Math

Coalition plan $\mathcal{P}$:

$Proj_{name,department}(o_1 \bowtie o_4) \cup$

$Proj_{name,department}(o_2 \bowtie o_4) \cup$

$Proj_{name,department}(o_3 \bowtie o_4) \cup$

o_5

Data Assemblage

Coalition Set

name	department
Alice	CS
Kate	Math

[1] Lloyd S. Shapley. A Value for n-Person Games. Technical Report, RAND Corporation, 1952.

SFU SIMON FRASER UNIVERSITY Duke UNIVERSITY

Existing Method

- Shapley Value: given a set of data owners $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$,
  - $$\psi(o_i) = \frac{1}{\|\mathcal{O}\|} \sum_{\mathcal{S} \subseteq \mathcal{O} \setminus \{o_i\}} \frac{Utility(\mathcal{S} \cup \{o_i\}) - Utility(\mathcal{S})}{\binom{n-1}{\|\mathcal{S}\|}}$$
  - E.g. ,
    - $Utility(\varnothing \cup \{o_1\}) - Utility(\varnothing) = 0$
    - $Utility(\{o_2\} \cup \{o_1\}) - Utility(\{o_2\}) = 0$
    - $Utility(\{o_3\} \cup \{o_1\}) - Utility(\{o_3\}) = 0$
    - $Utility(\{o_4\} \cup \{o_1\}) - Utility(\{o_4\}) = 1$
    - $Utility(\{o_5\} \cup \{o_1\}) - Utility(\{o_5\}) = 0$
    - ...

$$\boxed{\psi(o_1) = \frac{1}{6}}$$

o_1

id	name
1	Alice

o_2

id	name
1	Alice

o_3

id	name
2	Kate

o_4

id	department
1	CS
2	Math

o_5

name	department
Kate	Math

Coalition plan $\mathcal{P}$:
$Proj_{name,department}(o_1 \bowtie o_4) \cup$
$Proj_{name,department}(o_2 \bowtie o_4) \cup$
$Proj_{name,department}(o_3 \bowtie o_4) \cup$
o_5

Data Assemblage

Coalition Set

name	department
Alice	CS
Kate	Math

[1] Lloyd S. Shapley. A Value for n-Person Games. Technical Report, RAND Corporation, 1952.

Challenges

Combinatoric nature → Exponential with respect to the number of data owners

SFU SIMON FRASER UNIVERSITY Duke UNIVERSITY

Challenges

Combinatoric nature → Exponential with respect to the number of data owners

Utility evaluation → Potentially high computational cost in evaluating utility

Our Method: IUSV

Independent Utility Assumption

SFU SIMON FRASER UNIVERSITY
Duke UNIVERSITY

Our Method: IUSV

Independent Utility Assumption

Special Case → Closed Form Solution

General Case → Fast Algorithms

Independent Utility Assumption

The **independent utility assumption** holds on a data set $D = \{t_1, \ldots, t_l\}$ if the utility of the data set $Utility(D) = \sum_{i=1}^{l} Utility(t_i)$, and for any $1 \leq i, j \leq l, Utility(t_i)$ and $Utility(t_j)$ are non-negative and independent from each other.

Independent Utility Assumption

The **independent utility assumption** holds on a data set $D = \{t_1, \ldots, t_l\}$ if the utility of the data set $Utility(D) = \sum_{i=1}^{l} Utility(t_i)$, and for any $1 \leq i, j \leq l$, $Utility(t_i)$ and $Utility(t_j)$ are non-negative and independent from each other.

o_1

id	name
1	Alice

o_2

id	name
1	Alice

o_3

id	name
2	Kate

o_4

id	department
1	CS
2	Math

o_5

name	department
Kate	Math

Coalition plan $\mathscr{P}$:

$Proj_{name,department}(o_1 \bowtie o_4) \cup$

$Proj_{name,department}(o_2 \bowtie o_4) \cup$

$Proj_{name,department}(o_3 \bowtie o_4) \cup$

o_5

Data Assemblage

Coalition Set

name	department
Alice	CS
Kate	Math

Let $t_1 =$ (Alice, CS), $Utility(t_1) = 1$

Let $t_2 =$ (Kate, Math), $Utility(t_2) = 1$

$Utility(D) = Utility(t_1) + Utility(t_2) = 2$

SFU SIMON FRASER UNIVERSITY Duke UNIVERSITY

Independent Utility Assumption

o_1

id	name
1	Alice

o_2

id	name
1	Alice

o_3

id	name
2	Kate

o_4

id	department
1	CS
2	Math

o_5

name	department
Kate	Math

Coalition plan $\mathscr{P}$:

$Proj_{name,department}(o_1 \bowtie o_4) \cup$

$Proj_{name,department}(o_2 \bowtie o_4) \cup$

$Proj_{name,department}(o_3 \bowtie o_4) \cup$

o_5

Data Assemblage

Coalition Set

tuple_id	name	department
t_1	Alice	CS
t_2	Kate	Math

$$\boxed{\psi(o_1) = \psi_{t_1}(o_1) + \psi_{t_2}(o_1)}$$

Independent Utility Assumption

o_1

id	name
1	Alice

o_2

id	name
1	Alice

o_3

id	name
2	Kate

o_4

id	department
1	CS
2	Math

o_5

name	department
Kate	Math

Coalition plan $\mathscr{P}$:

$Proj_{name,department}(o_1 \bowtie o_4) \cup$

$Proj_{name,department}(o_2 \bowtie o_4) \cup$

$Proj_{name,department}(o_3 \bowtie o_4) \cup$

o_5

Data Assemblage

Coalition Set

tuple_id	name	department
t_1	Alice	CS
t_2	Kate	Math

$$\psi(o_1) = \psi_{t_1}(o_1) + \psi_{t_2}(o_1)$$

Problem of calculating $\psi(o_i)$ with respect to the coalition set D

Under Independent Utility Assumption

Problem of calculating $\psi_t(o_i)$ with respect to a tuple $t \in D$

Synthesis

- Synthesis:
  - For a given tuple t, a **synthesis** is a set of data owners that can produce t according to the coalition plan $\mathscr{P}$.
  - E.g., for t_2, $\{o_3, o_4\}, \{o_1, o_3, o_4\}$
- Minimal Synthesis
  - E.g., for t_2, $\{o_3, o_4\}$
- Synthesis types:
  - Single-owner Synthesis
    - E.g., for t_2, $\{o_5\}$
  - Multi-owner Synthesis
    - E.g., for t_2, $\{o_3, o_4\}$

o_1

id	name
1	Alice

o_2

id	name
1	Alice

o_3

id	name
2	Kate

o_4

id	department
1	CS
2	Math

o_5

name	department
Kate	Math

Coalition plan $\mathscr{P}$:

$Proj_{name,department}(o_1 \bowtie o_4) \cup$
$Proj_{name,department}(o_2 \bowtie o_4) \cup$
$Proj_{name,department}(o_3 \bowtie o_4) \cup$
o_5

Data Assemblage

Coalition Set

tuple_id	name	department
t_1	Alice	CS
t_2	Kate	Math

Synthesis

- Synthesis:
  - For a given tuple t, a **synthesis** is a set of data owners that can produce t according to the coalition plan $\mathscr{P}$.
  - E.g., for t_2, $\{o_3, o_4\}$, $\{o_1, o_3, o_4\}$
- Minimal Synthesis
  - E.g., for t_2, $\{o_3, o_4\}$
- Synthesis types:
  - Single-owner Synthesis
    - E.g., for t_2, $\{o_5\}$
  - Multi-owner Synthesis
    - E.g., for t_2, $\{o_3, o_4\}$

o_1

id	name
1	Alice

o_2

id	name
1	Alice

o_3

id	name
2	Kate

o_4

id	department
1	CS
2	Math

o_5

name	department
Kate	Math

Coalition plan $\mathscr{P}$:

$Proj_{name,department}(o_1 \bowtie o_4) \cup$
$Proj_{name,department}(o_2 \bowtie o_4) \cup$
$Proj_{name,department}(o_3 \bowtie o_4) \cup$
o_5

Data Assemblage

Coalition Set

tuple_id	name	department
t_1	Alice	CS
t_2	Kate	Math

Synthesis

- Synthesis:
  - For a given tuple t, a **synthesis** is a set of data owners that can produce t according to the coalition plan $\mathscr{P}$.
  - E.g., for t_2, $\{o_3, o_4\}$, $\{o_1, o_3, o_4\}$
- Minimal Synthesis
  - E.g., for t_2, $\{o_3, o_4\}$
- Synthesis types:
  - Single-owner Synthesis
    - E.g., for t_2, $\{o_5\}$
  - Multi-owner Synthesis
    - E.g., for t_2, $\{o_3, o_4\}$

o_1

id	name
1	Alice

o_2

id	name
1	Alice

o_3

id	name
2	Kate

o_4

id	department
1	CS
2	Math

o_5

name	department
Kate	Math

Coalition plan $\mathscr{P}$:

$Proj_{name,department}(o_1 \bowtie o_4) \cup$
$Proj_{name,department}(o_2 \bowtie o_4) \cup$
$Proj_{name,department}(o_3 \bowtie o_4) \cup$
o_5

Data Assemblage

Coalition Set

tuple_id	name	department
t_1	Alice	CS
t_2	Kate	Math

Synthesis

o_1

id	name
1	Alice

o_2

id	name
1	Alice

o_3

id	name
2	Kate

o_4

id	department
1	CS
2	Math

o_5

name	department
Kate	Math

Coalition plan $\mathscr{P}$:

$Proj_{name,department}(o_1 \bowtie o_4) \cup$

$Proj_{name,department}(o_2 \bowtie o_4) \cup$

$Proj_{name,department}(o_3 \bowtie o_4) \cup$

o_5

Data Assemblage

Coalition Set

tuple_id	name	department
t_1	Alice	CS
t_2	Kate	Math

Minimal Syntheses

$\{\{o_1, o_4\}, \{o_2, o_4\}\}$

$\{\{o_3, o_4\}, \{o_5\}\}$

- $\|\mathcal{O}_t\| \leq \|\mathcal{O}\|$, where $\mathcal{O}_t$ is the number of data owners contributing to t

Synthesis

o_1

id	name
1	Alice

o_2

id	name
1	Alice

o_3

id	name
2	Kate

o_4

id	department
1	CS
2	Math

o_5

name	department
Kate	Math

Coalition plan $\mathscr{P}$:

$Proj_{name,department}(o_1 \bowtie o_4) \cup$
$Proj_{name,department}(o_2 \bowtie o_4) \cup$
$Proj_{name,department}(o_3 \bowtie o_4) \cup$
o_5

Data Assemblage

Coalition Set

tuple_id	name	department
t_1	Alice	CS
t_2	Kate	Math

Minimal Syntheses

$\{\{o_1, o_4\}, \{o_2, o_4\}\}$

$\{\{o_3, o_4\}, \{o_5\}\}$

- $\|\mathcal{O}_t\| \leq \|\mathcal{O}\|$, where $\mathcal{O}_t$ is the number of data owners contributing to t
- Given a tuple $t, \forall \mathcal{S} \subseteq \mathcal{O}_t, Utility_t(\mathcal{S}) = Utility(t) \Longleftrightarrow \mathcal{S}$ is a synthesis of t

Special Case

- Case 1: only single-owner synthesis exists
  - Closed-form solution in constant time $\psi_t(o_i) = \dfrac{Utility(t)}{\|\mathcal{O}_t\|}$
  - E.g., assume a tuple t with minimal syntheses: $\{\{o_6\}, \{o_7\}, \{o_8\}\}$

- Case 2: there is a unique multi-owner synthesis (UMOS)
  - Closed-form solution in linear time $\psi_t(o_i) = \dfrac{Utility(t)}{\|\mathcal{O}_t\| \times \binom{\|\mathcal{O}_t\| - 1}{m-1}}$ for o_i in the UMOS, where m is the number of data owners in the UMOS
  - E.g., for t_2, $\{\{o_3, o_4\}, \{o_5\}\}$

SFU SIMON FRASER UNIVERSITY Duke UNIVERSITY

Special Case

- Case 1: only single-owner synthesis exists
  - Closed-form solution in constant time $\psi_t(o_i) = \frac{Utility(t)}{\|\mathcal{O}_t\|}$
  - E.g., assume a tuple t with minimal syntheses: $\{\{o_6\}, \{o_7\}, \{o_8\}\}$

- Case 2: there is a unique multi-owner synthesis (UMOS)
  - Closed-form solution in linear time $\psi_t(o_i) = \frac{Utility(t)}{\|\mathcal{O}_t\| \times \binom{\|\mathcal{O}_t\| - 1}{m-1}}$ for o_i in the UMOS, where m is the number of data owners in the UMOS
  - E.g., for t_2, $\{\{o_3, o_4\}, \{o_5\}\}$

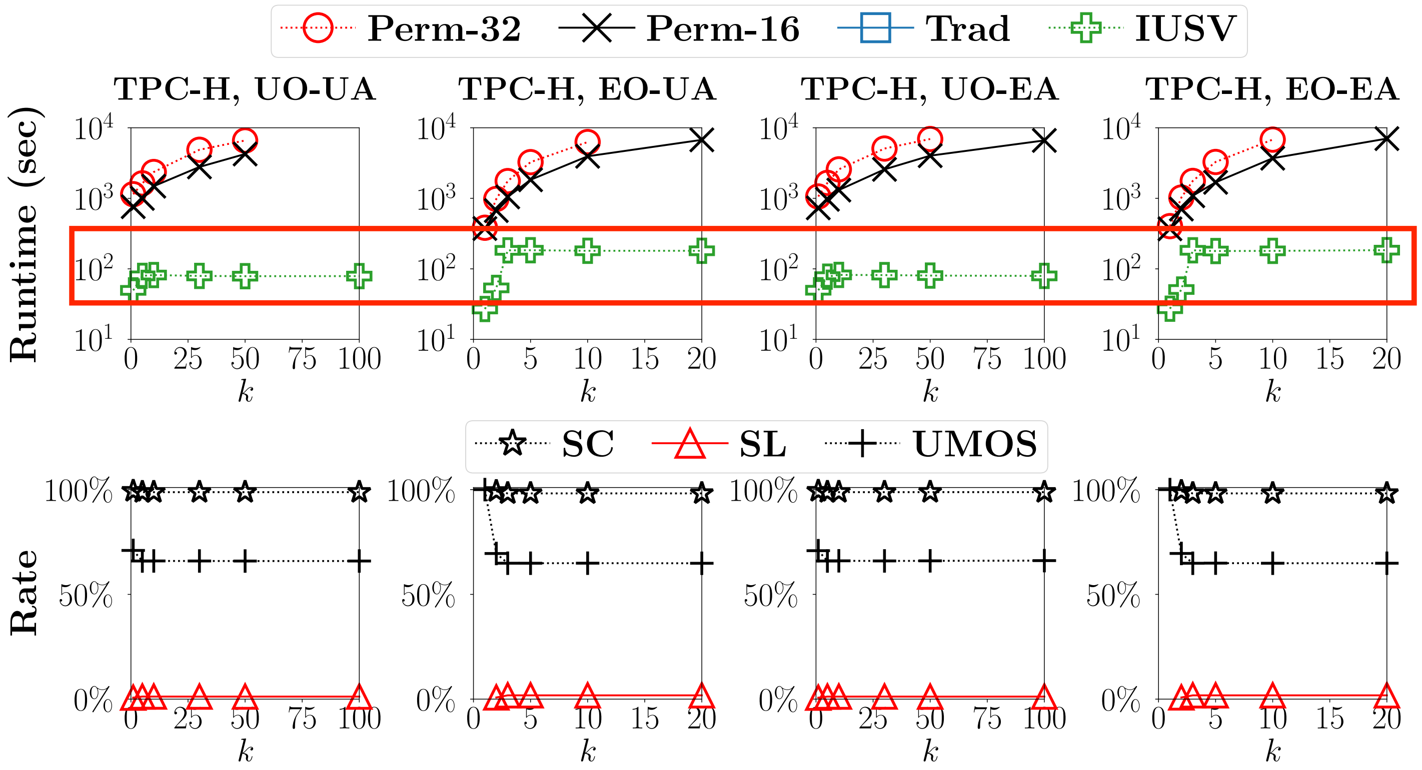
General Case

- SL Algorithm
  - General idea: $\forall \mathcal{S} \subseteq \mathcal{O}_t \backslash \{o_i\}$, enumerate $\mathcal{S}$ and evaluate $Utility_t(\mathcal{S})$ by checking whether $\mathcal{S}$ is a synthesis of t
  - Drawback: high computational cost when $\|\mathcal{O}_t\|$ is large
- SC Algorithm
  - General idea: $\forall \mathcal{S} \subseteq \mathcal{O}_t \backslash \{o_i\}$, use the combination of minimal syntheses to find all such $\mathcal{S}$ that $Utility_t(\mathcal{S} \cup \{o_i\}) - Utility_t(\mathcal{S}) = Utility(t)$
  - Drawback: high computational cost when the number of minimal syntheses is large
- A heuristic method to choose between SL and SC algorithms

General Case

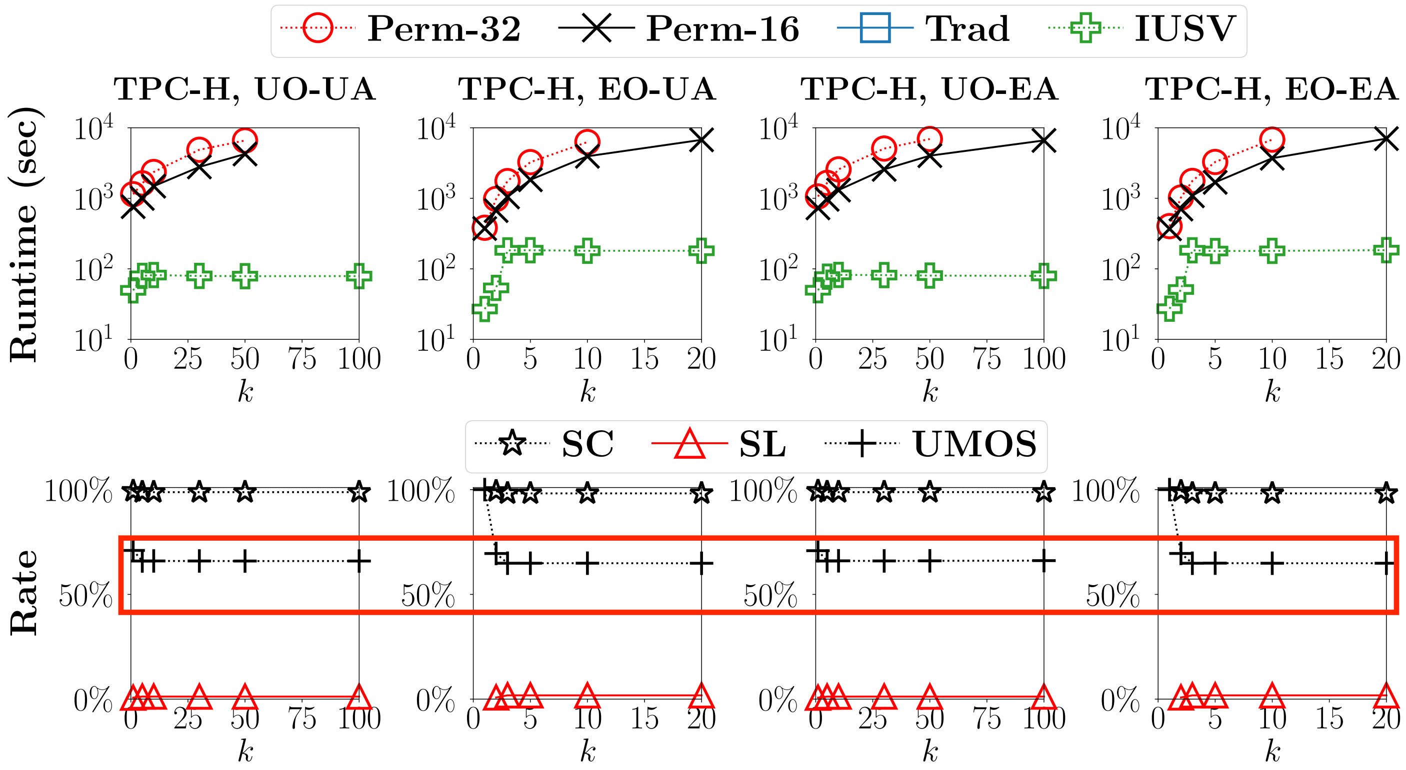
- SL Algorithm
  - General idea: $\forall \mathcal{S} \subseteq \mathcal{O}_t \backslash \{o_i\}$, enumerate $\mathcal{S}$ and evaluate $Utility_t(\mathcal{S})$ by checking whether $\mathcal{S}$ is a synthesis of t
  - Drawback: high computational cost when $\|\mathcal{O}_t\|$ is large
- SC Algorithm
  - General idea: $\forall \mathcal{S} \subseteq \mathcal{O}_t \backslash \{o_i\}$, use the combination of minimal syntheses to find all such $\mathcal{S}$ that $Utility_t(\mathcal{S} \cup \{o_i\}) - Utility_t(\mathcal{S}) = Utility(t)$
  - Drawback: high computational cost when the number of minimal syntheses is large
- A heuristic method to choose between SL and SC algorithms

Experimental Results



Scalability with respect to number of data owners

Experimental Results



Scalability with respect to number of data owners

Closed form solution in the majority cases

Conclusion

- We identify independent utility assumption
- We develop an exact Shapley value computation method under the assumption
  - Closed form solution in the majority cases
  - Fast algorithms in general case
- Experiments show improving performance by orders of magnitudes

Thanks
Q&A