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

{xuan_luo, zicun_cong, cheng_xu_3}@sfu.ca jpei@duke.edu

Background and Problem Formulation

- **Transforming Data into Value**:
  - Data are distributed.
  - Data may be assembled.
  - Data have diverse second use.
  - Data marketplaces enable end-to-end data science as a dynamic Eco-system.
- **Problem Formulation**:
  - Given a set of data owners $\mathcal{O} = \{o_1, \ldots, o_n\}$, a coalition plan $\mathcal{P}$ specifying how data from data owners can be assembled, and a reward from a data buyer. How to distribute the data buyer's reward to data owners?
- **Existing Method**:
  - Shapley value: the expectation of marginal contribution made by the data owner in all possible coalitions with subsets of other data owners.

$$\psi(o_i) = \frac{1}{\|\mathcal{O}\|} \sum_{\mathcal{S} \subseteq \mathcal{O} \setminus \{o_i\}} \frac{Utility(\mathcal{S} \cup \{o_i\}) - Utility(\mathcal{S})}{\binom{n-1}{\|\mathcal{S}\|}}$$

- **Challenges**:
  - Combinatoric nature
    ⇒ Exponential with respect to the number of data owners
  - Utility evaluation
    ⇒ Potentially high computational cost in evaluating utility

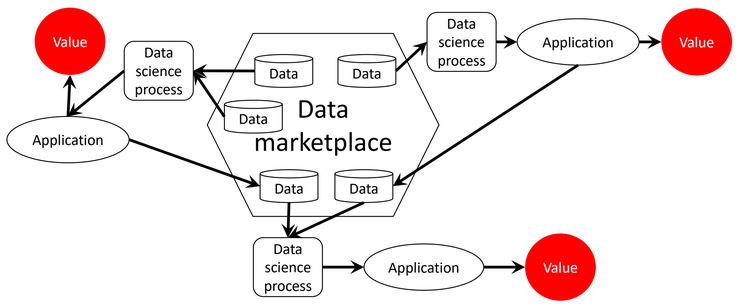

Fig. 1: Data Marketplaces: Enabling End-to-end Data Science as a Dynamic Eco-system

Independent Utility

- **Independent Utility Assumption**: it holds on a data set $D = \{t_1, \ldots, t_l\}$ if the utility of the data set $Utility(D) = \sum_{j=1}^{l} Utility(t_j)$ 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 Shapley Value**: let $D = \{t_1, \ldots, t_l\}$ be a coalition set produced by a coalition by data owners $\mathcal{O} = \{o_1, \ldots, o_n\}$. Under the independent utility assumption, for every data owner o_i $(1 \leq i \leq n)$, the Shapley value of o_i is $\psi(o_i) = \sum_{j=1}^{l} \psi_{t_j}(o_i)$, where $\psi_{t_j}(o_i)$ is the Shapley value of o_i in producing tuple t_j by coalition.

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 $t \in D$

Running Example

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 of t_1: $\{\{o_1, o_4\}, \{o_2, o_4\}\}$
Minimal syntheses of t_2: $\{\{o_3, o_4\}, \{o_5\}\}$

Fig. 2: Example of Data Assemblage

Synthesis

- **Synthesis**: for a tuple in the coalition set $t \in D$, if data owners $o_{i_1}, \ldots, o_{i_m}$ in coalition produce instance of t according to the coalition plan $\mathcal{P}$, then $O = \{o_{i_1}, \ldots, o_{i_m}\}$ is called a synthesis of t.
  ⇒ E.g., for t_2, $\{o_3, o_4\}, \{o_1, o_3, o_4\}$
- **Minimal Synthesis**: a synthesis O is a minimal synthesis of tuple $t \in D$ if no proper subset of O is still a synthesis of t.
  ⇒ E.g., for t_2, $\{o_3, o_4\}$
- **Synthesis Type**:
  - Single-owner synthesis: a synthesis $O = \{o_{i_j}\}$ with only one data owner.
    ⇒ E.g., for t_2, $\{o_5\}$
  - Multi-owner synthesis: a synthesis $O = \{o_{i_1}, \ldots, o_{i_m}\}$ with more than one data owner.
    ⇒ E.g., for t_2, $\{o_3, o_4\}$
- **Observations from Synthesis**:
  - $\|\mathcal{O}_t\| \leq \|\mathcal{O}\|$, where $\mathcal{O}_t$ is the number of data owners contributing to t.
  - Given a tuple t, $\forall \mathcal{S} \subseteq \mathcal{O}_t, Utility_t(S) = Utility(t) \iff \mathcal{S}$ is a synthesis of t.

Special Case

- **Case 1: only single-owner synthesis exists**:
  - Closed form solution in constant time $\psi_t(o_i) = \frac{Utility(t)}{\|\mathcal{O}_t\|}$
  - 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: $\forall \mathcal{S} \subseteq \mathcal{O}_t \setminus \{o_i\}$, 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: use the combination of minimal syntheses to find all such $\mathcal{S} \subseteq \mathcal{O}_t \setminus \{o_i\}$ that $Utility_t(\mathcal{S} \cup \{o_i\}) - Utility_t(\mathcal{S}) = Utility(t)$.
  - Drawback: high computational cost when there is a large number of minimal syntheses.
- A heuristic method to choose between SL and SC algorithms.

Performance Evaluation

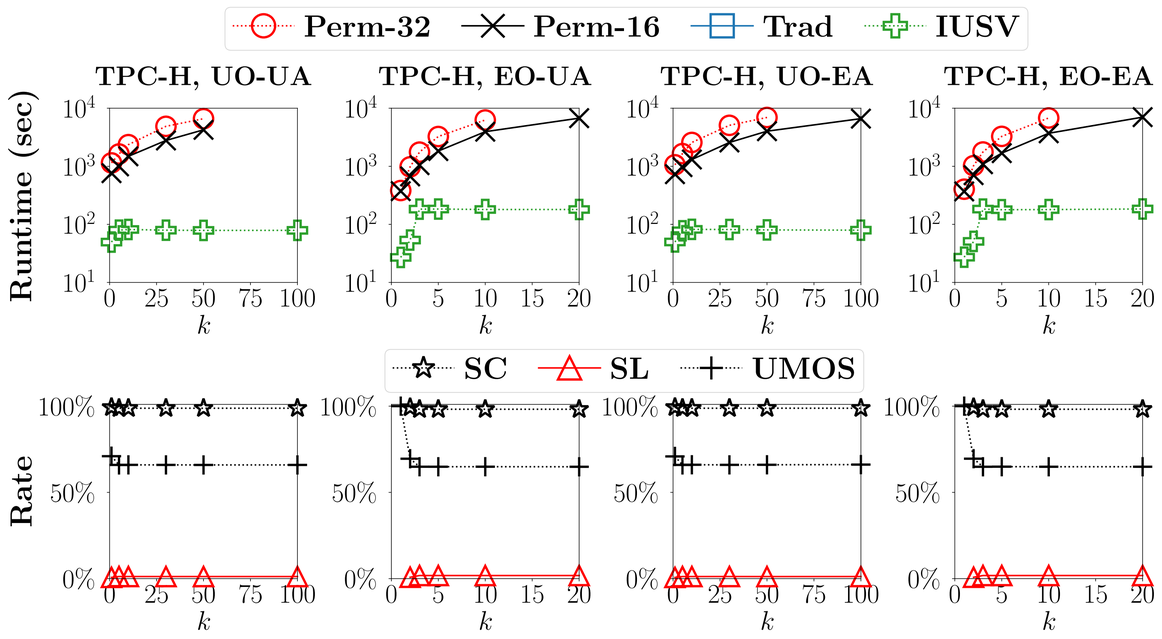

Fig. 3: Scalability on Number of Data Owners

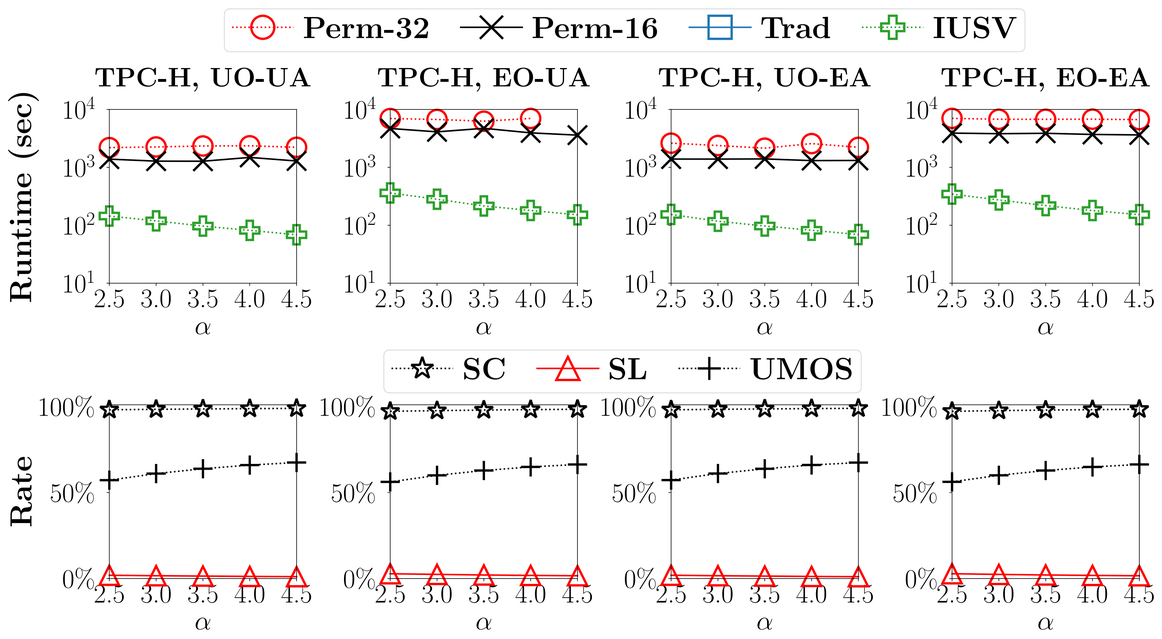

Fig. 4: Effect of Record Assignment Distribution