Tailoring the Shapley Value for In-context Example Selection towards Data Wrangling

Zheng Liang[1], Hongzhi Wang[1,✉], Xiaoou Ding[1], Zhiyu Liang[1], Chen Liang[1], Yafeng Tang[1], Jianzhong Qi[2]
[1] School of Computer Science, Harbin Institute of Technology, Harbin, China
[2] School of Computing and Information Systems, University of Melbourne, Melbourne, Australia
{lz20, wangzh, dingxiaoou,zyliang,chenliang}@hit.edu.cn, tangyf@stu.hit.edu.cn, jianzhong.qi@unimelb.edu.au

I. Proof for the Propositions

Proposition 1. *(Constrained Shapley Value Uniqueness): For any "game" (D, U), where U is a utility function that maps a subset S of players $D = \{d_1, d_2, \ldots, d_n\}$ to a real number: $U(S) \to \mathbb{R}$, if U can only take coalitions (i.e., subset S of D) containing at most k players (i.e., candidate examples) as input, $CSV(\cdot)$ is the* only *value function $F(d_i)$ that satisfies the following properties for reward allocation:*

Symmetry: Any two candidate examples with equal marginal contributions to every subset S receive the same reward. Formally, $\forall d_i, d_j \in D$, if $\forall S \subset D, |S| < k : U(S \cup \{d_i\}) = U(S \cup \{d_j\})$, then $F(d_i) = F(d_j)$, where $F(d_i)$ and $F(d_j)$ are the rewards of d_i and d_j.

Additivity: The utility function value on all players $U(D)$ can be fully divided among the candidate examples, i.e., $U(D) = \sum_{d_i \in D} F(d_i)$.

Balance: For any player $d_i \in D$ playing any two games (D, F_1) and (D, F_2) getting reward $F_1(d_i)$ and $F_2(d_i)$, respectively; its reward allocation for the game $(D, F_1 + F_2)$ is $F_1(d_i) + F_2(d_i)$.

Zero element: A candidate example with zero contribution to the reward of every subset of D with up to k elements has a reward of 0. Formally, $\forall d_i \in D$, if $\forall S \subset D, |S| < k : U(S \cup \{d_i\}) = U(S)$, then $F(d_i) = 0$.

Proof. We show the uniqueness of the CSV based on its definition. Note that this is our corollary of the theorem in a previous work [1]. We give this proof to make our paper self-contained.

We use $r, s, n, \ldots$ to represent the size of sets $R, S, N, \ldots$, respectively. The sets will be introduced below when they are needed.

Let P be the universe of players. Define a game to be any set function $v : P \to \mathbb{R}$ that maps from a subset of U to a real number, where a superadditive game satisfies:

1) $v(\emptyset) = 0$;
2) $v(S) \geq v(S \cap T) + v(S - T)$, $\forall S, T \subset U \wedge |S| \leq k \wedge |T| \leq k$;
3) A carrier of v is any subset $N \subset U$ with $v(S) = v(N \cap S), \forall S \subset U$. $F_i[v] = 0$ for **zero elements** $\forall i \in S \setminus N$;

✉ Hongzhi Wang is the corresponding author.

Our goal is to compute $F_i[v]$, the valuation of i in game v, which is supposed to satisfy the four properties. Following the previous work [2], we compute $F_i[v]$ in three steps:

Step 1: Decompose v into the weighted sum of certain symmetric games.

Step 2: Compute the weight and the valuation function in the symmetric games.

Step 3: Compute $F_i[v]$.

Step 1: We first consider certain symmetric games. For any $R \subset U, R \neq \emptyset$, we define v_R:

$$v_R(S) = \begin{cases} 0 & \text{if } R \subset S, |S| \leq k \\ 1 & \text{if } R \not\subset S, |S| \leq k \\ 0 & \text{if } |S| > k \end{cases} \tag{1}$$

A immediate corollary to the **Additivity property** is that $F[v - w] = F[v] - F[w]$ if v, w, and $v - w$ are all games. Therefore, according to the previous work [2], any game v is a linear combination of symmetric games v_R:

$$v = \sum_{R \subset N, R \neq \emptyset} c_R(v) v_R, \tag{2}$$

where the coefficients are given by

$$c_R(v) = \sum_{T \subset R} (-1)^{r-t} v(T) \tag{3}$$

Step 2: Suppose a projection $\pi : R \to R, \pi(i) = j$, also $\pi(R) = R$, by the **Balance property**, we have:

$$F_i[v_R] = F_{\pi(i)}[v_{\pi(R)}] = F_j[v_R]. \tag{4}$$

Further, based on the **Symmetry property**, we have:

$$1 = v_R(R) = \sum_{j \in R} F_j[v_R] = r F_i[v_R]. \tag{5}$$

Therefore,

$$F_i[v_R] = \begin{cases} \frac{1}{r} & \text{if } i \in R, \\ 0 & \text{if } i \notin R. \end{cases} \tag{6}$$

Step 3: We now apply Equation 6 to Equation 2 and obtain:

$$F_i[v] = \sum_{R\subset N, i\in R} \frac{c_R(v)}{r}, \forall i \in N \tag{7}$$

$$= \sum_{R\subset N, i\in R} \frac{\sum_{T\in R}(-1)^{r-t}v(T)}{r}, \forall i \in N \tag{8}$$

$$= \sum_{S\subset N, i\in S, |S|\leq k} \frac{(s-1)!(n-s)!}{n!}v(S) \tag{9}$$

$$- \sum_{S\subset N, i\notin S, |S|\leq k-1} \frac{(s)!(n-s-1)!}{n!}v(S), \forall i \in N \tag{10}$$

$$= CSV(d_i) \tag{11}$$

Therefore, a unique value function F satisfying Balance, Symmetry, and Additivity, for games with finite carriers, is given by the definition of the Constrained Shapley Value. □

Proposition 2. *(Monte Carlo Marginal Contribution Approximation Quality [3]) According to Hoeffding's inequality, given the range $r = max(CSV(d_i)) - min(CSV(d_i))$ of $CSV(d_i)$, a CSV estimation error bound ϵ, and a confidence level $1-\delta$, the MCSV algorithm takes $\frac{mr^2 avl(D)k^2}{4\epsilon^2}\log\frac{2n}{\delta}$ API token costs and $\frac{mr^2}{2\epsilon^2}k\log\frac{2n}{\delta}$ queries to an LLM to ensure $P(|\overline{CSV(d_i)} - CSV(d_i)| \geq \epsilon) \leq \delta$, where $avl(D)$ denotes the average number of tokens in serialized EM examples.*

Proof. For any random variable $S_{min} \leq S \leq S_{max}$, according to the Hoeffding's inequality we have:

$$P(|S - E(S)| \geq t) \leq 2\cdot\exp(-\frac{2t^2}{\sum_{i=1}^{m}(S_{max}-S_{min})^2}) \tag{12}$$

For $\forall d_i \in D$, let $S = \sum_{i=1}^{n} CSV(d_i)$, r be the difference of the maximum and minimum values of $CSV(d_i)$, c be the number of 'permutations' [4] in Line 3 of the MCSV Algorithm, the Hoeffding's inequality entails that:

$$P(|S - cE(S)| \geq t) = P(|CSV(d_i) - \overline{CSV(d_i)}| \geq \frac{t}{c}) \tag{13}$$

$$P(|CSV(d_i) - \overline{CSV(d_i)}| \geq \epsilon) \leq 2\cdot\exp(-\frac{2c^2\epsilon^2}{cr^2}) \tag{14}$$

Our aim is to make the right hand side to be at most δ:

$$2\cdot\exp(\frac{-2c\epsilon^2}{r^2}) \leq \delta \tag{15}$$

$$c \geq \frac{r^2\cdot\log\frac{2}{\delta}}{2\epsilon^2} \tag{16}$$

As each sample can only be used by one CSV approximation, the total number of utility function computation is $\frac{nc}{k} \geq \frac{nr^2\cdot\log\frac{2}{\delta}}{2k\epsilon^2}$. Each permutation requires examining k candidate examples, which means km questions to the LLM. Thus, the number of QA turns is at least $\frac{mnr^2\cdot\log\frac{2}{\delta}}{2\epsilon^2}$. On average, $\frac{k}{2}$ candidate examples are used for LLM in-context prompting, and hence the MCSV Algorithm consumes $\frac{avl(D)kmnr^2\cdot\log\frac{2}{\delta}}{4\epsilon^2}$ API tokens. This completes the proof. □

Proposition 3. *(Effectiveness of Activated Contribution Approximation) Given a set of candidate examples $D = \{d_1, d_2, \ldots, d_n\}$, the constrained Shapley value of d_i can be computed by:*

$$CSV(d_i) = \frac{1}{n}\sum_{S\subset D, 0\leq|S|\leq k} \frac{AC(S,d_i)}{\binom{n-1}{|S|}} \tag{17}$$

Proof. We rewrite the definition of CSV to:

$$CSV(d_i) = \frac{1}{n}\sum_{\substack{S\subset D\setminus\{d_i\},\\ 0\leq|S|\leq k-1}} \frac{U(S\cup\{d_i\}) - U(S)}{\binom{n-1}{|S|}}$$
$$= \frac{1}{n}\sum_{\substack{S\subset D\setminus\{d_i\},\\ 0\leq|S|\leq k-1}} \frac{U(S\cup\{d_i\})}{\binom{n-1}{|S|}} - \frac{1}{n}\sum_{\substack{S\subset D\setminus\{d_i\},\\ 0\leq|S|\leq k-1}} \frac{U(S)}{\binom{n-1}{|S|}} \tag{18}$$

Let $S' = S\cup\{d_i\}$. Since $d_i \notin S$, we have $S = S'\setminus\{d_i\}$. Putting S' into the equation above yields:

$$CSV(d_i) = \frac{1}{n}\sum_{\substack{d_i\in S',\\ 1\leq|S'|\leq k,\\ S'\subset D}} \frac{U(S')}{\binom{n-1}{|S'|-1}} - \frac{1}{n}\sum_{\substack{d_i\notin S,\\ 0\leq|S|\leq k-1,\\ S\subset D}} \frac{U(S)}{\binom{n-1}{|S|}} \tag{19}$$

Next, we use a shared variable S^* to replace S' and S. Since $S' \neq S$ always holds, they can be viewed as two cases of variable S^*.

$$CSV(d_i) = \frac{1}{n}\sum_{\substack{d_i\in S^*,\\ 1\leq|S^*|\leq k,\\ S^*\subset D}} \frac{U(S^*)}{\binom{n-1}{|S^*|-1}} - \frac{1}{n}\sum_{\substack{d_i\notin S^*,\\ 0\leq|S^*|\leq k-1,\\ S^*\subset D}} \frac{U(S^*)}{\binom{n-1}{|S^*|}}$$
$$= \frac{1}{n}\sum_{\substack{d_i\in S^*,\\ 1\leq|S^*|\leq k,\\ S^*\subset D}} \frac{U(S^*)\cdot(\frac{n}{|S^*|}-1)}{\binom{n-1}{|S^*|}} - \frac{1}{n}\sum_{\substack{d_i\notin S^*,\\ 0\leq|S^*|\leq k-1,\\ S^*\subset D}} \frac{U(S^*)}{\binom{n-1}{|S^*|}} \tag{20}$$

According to the definition of Activated Contribution and the fact that $\left|S^*\right| \neq 0$ when $d_i \in S^*$, we can rewrite the weight term into a unified weight as follows.

$$CSV(d_i) = \frac{1}{n}\sum_{\substack{d_i\in S^*,\\ 0\leq|S^*|\leq k,\\ S^*\subset D}} \frac{U(S^*)f(S^*,d_i)}{\binom{n-1}{|S^*|}} + \frac{1}{n}\sum_{\substack{d_i\notin S^*,\\ 0\leq|S^*|\leq k-1,\\ S^*\subset D}} \frac{U(S^*)f(S^*,d_i)}{\binom{n-1}{|S^*|}}$$
$$= \frac{1}{n}\sum_{\substack{0\leq|S^*|\leq k,\\ S^*\subset D}} \frac{U(S^*)f(S^*,d_i)}{\binom{n-1}{|S^*|}} - \frac{1}{n}\sum_{\substack{d_i\notin S^*,\\ |S^*|=k,\\ S^*\subset D}} \frac{U(S^*)f(S^*,d_i)}{\binom{n-1}{|S^*|}}$$
$$= \frac{1}{n}\sum_{\substack{0\leq|S^*|\leq k,\\ S^*\subset D}} \frac{AC(S^*,d_i)}{\binom{n-1}{|S^*|}} \tag{21}$$

Replacing S^* with S completes the proof.

□

Proposition 4. *(Unbiased Estimation of Activated Contribution Approximation) Given a set of players $D = \{d_1, d_2, \ldots, d_n\}$, the ACSV Algorithm gives an unbiased estimation of the constrained Shapley value for every player, that is, $E(\overline{CSV(d_i)}) = CSV(d_i), 1 \leq i \leq n$.*

Proof. Let $CSV_{i,j}$ be the CSV of a coalition of size j calculated as follows:

$$CSV_{i,j} = \frac{1}{n} \sum_{\substack{S \subset D, \\ |S|=j}} \frac{AC(S, d_i)}{\binom{n-1}{|S|}} \tag{22}$$

By the definition of CSV, we have the following immediately:

$$\begin{aligned} CSV(d_i) &= \frac{1}{n} \sum_{1 \leq j \leq n} CSV_{i,j} \\ CSV_{i,j} &= E(AC(S, d_i)) \end{aligned} \tag{23}$$

In the ACSV Algorithm, all possible S is sampled from N (Line 3) with sample allocation. Thus, according to Theorem 4.5 of a previous work [5], $\frac{\overline{CSV_{i,j}}}{m_{i,j}}$ is an unbiased estimation of $AC(S, d_i)$. Therefore, the proposition holds.

$$\begin{aligned} CSV(d_i) &= \sum_{j=1}^{n} CSV_{i,j} = \sum_{j=1}^{n} E(AC(S, d_i)) \\ &= \sum_{j=1}^{n} E(\frac{\overline{CSV_{i,j}}}{m_{i,j}}) = E(\overline{CSV_i}) \end{aligned} \tag{24}$$

□

Proposition 5. *(AC-based Minimized Deviation Approximation Quality) With sample allocation towards deviation minimization, the ACSV Algorithm takes $\frac{2r^2 \log \frac{2}{\delta} avl(D)\sqrt{n}}{\epsilon^2} \sum_{j=1}^{k} \frac{(j+1)}{\sqrt[3]{j}}$ API token costs to ensure $P(|\overline{CSV(d_i)} - CSV(d_i)| \geq \epsilon) \leq \delta$.*

Proof. According to the optimal solution to the relaxed Deviation Minimization problem, within the probability of at least $1 - \delta$, we have:

$$|\overline{CSV(d_i)} - CSV(d_i)| \leq 2r \sqrt{\frac{\log \frac{2}{\delta} \cdot avl(D) \cdot n}{2B} \sum_{j=1}^{k} \frac{(j+1)}{\sqrt[3]{j}}} \tag{25}$$

By setting the right hand side as ϵ, we have:

$$B = \frac{2r^2 \log \frac{2}{\delta} avl(D)\sqrt{n}}{\epsilon^2} \sum_{j=1}^{k} \frac{(j+1)}{\sqrt[3]{j}} = O(\frac{mk^2\sqrt{n}}{\epsilon^2} \log \frac{1}{\delta}) \tag{26}$$

□

Proposition 6. *(AC-based Regret Minimizing Approximation Quality) With sample allocation towards regret minimization, the error probability of the ACSV Algorithm satisfies the following inequality:*

$$e_n = P(\cup_{i \leq k \leq j} CSV(d_i) < CSV(d_j)) \leq 2k^2 exp(-\frac{n-k}{8\overline{\log} k \cdot H}),$$

where $H = max_{i \in \{1,\ldots,K\}} i \cdot (|CSV(d_i) - CSV(d_{i+1})|)^{-2}$ and $\overline{\log} K = \frac{1}{2} + \sum_{i=2}^{K} \frac{1}{i}$.

Proof. Consider the event ξ defined by

$$\xi = \{j \in \{1, ..., K\}, |\frac{1}{n_k} \sum_{s=1}^{n_k} X_s - f_j| \leq \frac{1}{2} \Delta_{K+1-k}\}. \tag{27}$$

By Hoeffding's Inequality and with the sample allocation strategy in Equation 13, the probability of the complementary event $\bar{\xi}$ can be bounded as follows:

$$\begin{aligned} P(\bar{\xi}) &\leq \sum_{j=1}^{K} \sum_{k=1}^{K-1} P(|\frac{1}{n_k} \sum_{s=1}^{n_k} X_s - f_j| \leq \frac{1}{2} \Delta_{K+1-k}) \\ &\leq \sum_{j=1}^{K} \sum_{k=1}^{K-1} 2exp(2n_k(\Delta_{K+1-k})/2)^2) \\ &\leq 2K^2 exp(-\frac{n-K}{2\overline{\log} K \cdot H}). \end{aligned} \tag{28}$$

Here, the last inequality comes from the fact that:

$$\begin{aligned} & n_k(\Delta_{K+1-k})^2 \\ \geq & \frac{n-K}{\overline{\log}(K)(K+1-H)(\Delta_{K+1-k})^{-2}} \\ \geq & \frac{n-K}{\overline{\log}(K) \cdot H}. \end{aligned} \tag{29}$$

Thus, it suffices to show that on event ξ, the algorithm makes no error. We prove this by induction on k. Let $k \geq 1$. Assume that the algorithm makes no error in all previous $k-1$ stages, i.e., no bad arm $\mu_i < \theta$ has been accepted and no good arm $\mu_i \geq \theta$ has been rejected. Event ξ implies that at the end of stage k, all empirical means are within $\frac{1}{2}(\Delta_{K+1-k})^{-2}$ of the respective true means.

Let $A_k = \{a_1, \ldots, a_{K+1-k}\}$ be the set of active arms during stage k. We order the a_i's such that $\mu_{a_1} > \mu_{a_2} > \ldots > \mu_{a_{K+1-k}}$. Let $m' = m(k)$ be the number of arms left to find in stage k. The fact that no error has occurred in the first $k-1$ stages implies:

$$a_1, a_2, \ldots, a_{m'} \in \{1, \ldots, m\} \tag{30}$$

and

$$a_{m'+1}, \ldots, a_{K+1-k} \in \{m+1, \ldots, K\} \tag{31}$$

If an error is made at stage k, it can be one of the following two types:

(1) The algorithm accepts a_j at stage k for some $k \geq m'+1$.

(2) The algorithm rejects a_j at stage k for some $j \leq m'$.

Let $\sigma = \sigma_k$ be the bijection (from $\{1, \ldots, K+1-k\}$ to A_k) such that $\overline{\mu}_{\sigma(1),n_k} \geq \overline{\mu}_{\sigma(2),n_k} \geq \ldots \geq \overline{\mu}_{\sigma(K+1-k),n_k}$. Suppose Type 1 error has occurred. Then $a_j = \sigma(1)$, since if the algorithm accepts, it must accept the empirical best arm. Furthermore, we have:

$$\overline{\mu}_{a_j,n_k} - \theta \geq \theta - \overline{\mu}_{\sigma(K+1-k),n_k}, \tag{32}$$

since otherwise the algorithm would rather reject arm $\sigma(K+1-k)$. The condition $a_j = \sigma(1)$ and the event ξ implies that:

$$\begin{aligned} \overline{\mu}_{a_j,n_k} &\geq \overline{\mu}_{a_j,n_k}, \\ \mu_{a_j} + \frac{1}{2}(\Delta_{K+1-k}) &\geq \mu_{a_1} - \frac{1}{2}(\Delta_{K+1-k}), \\ (\Delta_{K+1-k}) &\geq \mu_{a_1} - \mu_{a_j} \geq \mu_{a_1} - \theta \end{aligned} \tag{33}$$

We then look at Condition (40). In the event of ξ, for all $i \leq m'$, we have:

$$\begin{aligned} \overline{\mu}_{a_j,n_k} &\geq \mu_{a_j} - \frac{1}{2}\Delta_{(K+1-k)} \\ &\geq \mu_{a_{m'}} - \frac{1}{2}\Delta_{(K+1-k)} \\ &\geq \theta - \frac{1}{2}\Delta_{(K+1-k)} \end{aligned} \tag{34}$$

On the other hand, $\overline{\mu}_{\sigma(K+1-k),n_k} \leq \overline{\mu}_{a_{K+1-k},n_k} \leq \overline{\mu}_{a_{K+1-k},n_k} + \frac{1}{2}\Delta_{(K+1-k)}$. Therefore, using those two observations and (40), we deduce:

$$\begin{aligned} (\mu_{a_j} + \frac{1}{2}\Delta_{(K+1-k)}) - \theta &\geq \theta - (\mu_{a_{K+1-k}} + \frac{1}{2}\Delta_{(K+1-k)}), \\ \Delta_{(K+1-k)} &\geq 2\theta - \mu_{a_j} - \mu_{a_{K+1-k}} > \theta - \mu_{a_{K+1-k}}. \end{aligned} \tag{35}$$

Thus, we proved that if there is a Type 1 error, then:

$$\Delta_{(K+1-k)} > max(\mu_{a_1} - \theta, \theta - \mu_{a_{K+1-k}}) \tag{36}$$

However, at stage k, only $k-1$ arms have been accepted or rejected, and hence $\Delta_{(K+1-k)} \leq max(\mu_{a_1} - \theta, \theta - \mu_{a_{K+1-k}})$. By contradiction, we conclude that Type 1 error cannot have occurred.

The reasoning process for Type 2 error is similar and omitted for conciseness. This completes the induction and the proof.

□

Proposition 7. *The probability of error of PS satisfies:*

$$e_N \leq 2\alpha K^2 \exp(-\frac{n-\alpha K}{2\alpha \cdot \overline{\log} K \cdot H}) \tag{37}$$

where $H(\alpha) = \max_{i\in\{1,2,\ldots,n\}} i \cdot (|CSV_{\pi_i} - CSV_{\pi_{i+1}}|)^{-2}$, $H = \max_{1\leq j\leq \alpha} H(j)$.

Proof. Consider events ξ_{d_i} for the i-th pre-trained MAB.

$$\xi_{d_i} = \{j \in \{1,2,\ldots,K\}, |\frac{1}{n_k}\sum_{s=1}^{n_k} X_{s,d_1} - f_{j,d_i}| \leq \frac{1}{2}\Delta_{K+1-k}\}$$

Also, consider an event ξ defined as follows.

$$\xi = \{j \in \{1,2,\ldots,K\}, |\frac{1}{n_k}\sum_{s=1}^{n_k} X_s - f_j| \leq \frac{1}{2}\Delta_{K+1-k}\} \tag{38}$$

where f_j is defined as follows.

$$f_j = \frac{\sum_{sim_j \in t, A' \in D} \cos\langle \vec{D}, \vec{D'}\rangle \cdot p(t, A')}{\sum_{sim_j \in t, A' \in D} \cos\langle \vec{D}, \vec{D'}\rangle} = \frac{\sum_d \cos\langle \vec{D}, \vec{d}\rangle \cdot f_{j,d}}{\sum_d \cos\langle \vec{D}, \vec{d}\rangle}$$

Letting $w_d = \cos\langle \vec{D}, \vec{d}\rangle$, we can rewrite Equation 38 as follows.

$$\begin{aligned} \xi = \{1 \leq j \leq K, |\frac{1}{n_k}\sum_{s=1}^{n_k}\sum_d w_d \cdot X_{s,d} - \sum_d w_d \cdot f_{j,d}| \\ \leq \frac{1}{2}\sum_{d\in D} w_d \Delta_{K+1-k}\} \end{aligned}$$

Suppose Equation 38 is true. Using the absolute value inequality, for any $1 \leq j \leq K$, we have:

$$|\frac{1}{n_k}\sum_{s=1}^{n_k}\sum_d w_d \cdot X_{s,d} - \sum_d w_d \cdot f_{j,d}| \leq \frac{1}{2}\sum_{d\in D} w_d \Delta_{K+1-k}$$

This implies that when $\overline{\xi}$ is true, $\overline{\xi_{d_1}} \cup \ldots \cup \overline{\xi_{d_{|D|}}}$ must be true regardless of w_d. By the law of total probability, we have:

$$P(\overline{\xi_{d_1}} \cup \ldots \cup \overline{\xi_{d_{|D|}}}) \geq P(\overline{\xi_{d_1}} \cup \ldots \cup \overline{\xi_{d_{|D|}}}|\overline{\xi}) \cdot P(\overline{\xi}) = P(\overline{\xi})$$

From the conclusion of Proposition 6, for each $MAB_i \in \{MAB_1, MAB_2, \ldots, MAB_\alpha\}$, we have:

$$P(\overline{\xi_d}) \leq 2K^2 \exp(-\frac{\frac{n}{a} - K}{2\overline{\log} K \cdot H(i)})$$

where $\overline{\log} K = \frac{1}{2} + \sum_{i=2}^{K} \frac{1}{i}$. By union bound, we come to the conclusion that:

$$P(\overline{\xi}) \leq P(\overline{\xi_{d_1}} \cup \ldots \cup \overline{\xi_{d_{|D|}}}) \leq 2aK^2 \exp(-\frac{\frac{n}{a} - K}{2\overline{\log} K \cdot H(a)}) \tag{39}$$

□

II. Pre-processing and Prompt Templates

Preprocessing. Blocking is a well-known effective technique to improve entity matching result quality. It filters the entity pair candidates that are unlikely to match. We perform blocking with DL-Block [6], the SOTA blocking algorithm, which follows DeepMatcher [7] and leverages deep learning techniques to generate embeddings for similarity measuring over entity pairs. We use *at most 200 examples for validation.*

Prompt template. We show below the prompt templates used for the four DW tasks.

Entity Matching: This is a deduplication task. Use domain knowledge to decide if E1 and E2 are the same.
Q1: E1 is A1:V1,.... E2 is A1:V1',.... Are E1 and E2 the same? A1: Yes.
Q2: E1 is A1:V2,.... E2 is A1:V2',.... Are E1 and E2 the same? A2: No.
...
Qk: E1 is A1:Vk,.... E2 is A1:Vk',.... Are E1 and E2 the same? Ak: No.
Answer the following question:
Q: E1 is A1:V,.... E2 is A1:V',.... Are E1 and E2 the same?

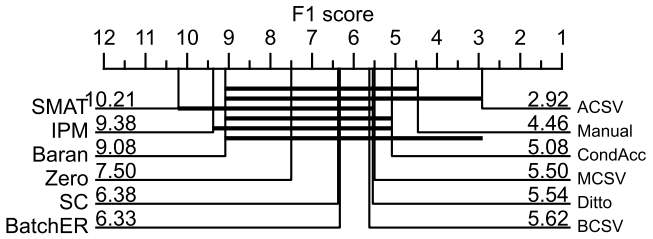
Schema Mapping: This is a schema mapping task. Use domain knowledge to decide if A1 and A2 are the same.
Q1: A1 is $name_1$. A2 is $name_1$. Are A1 and A2 the same?
A1: Yes.
Q2: A1 is $name_3$. A2 is $name_4$. Are A1 and A2 the same?
A2: No.
...
Qk: A1 is $name_{2k-1}$. A2 is $name_{2k}$. Are A1 and A2 the same? Ak: No.
Answer the following question:
Q: A1 is $name_{2k+1}$. A2 is $name_{2k+2}$. Are A1 and A2 the same?

Error Detection: This is a error detection task. Use domain knowledge to decide if E1 and E2 are the same.
Q1: E1 is A1:V1,·. E2 is A1:V1',·. Are E1 and E2 the same? A1: Yes.
Q2: E1 is A1:V2,·. E2 is A1:V2',·. Are E1 and E2 the same? A2: No.
...
Qk: E1 is A1:Vk,·. E2 is A1:Vk',·. Are E1 and E2 the same? Ak: No.
Answer the following question:
Q: E1 is A1:V,·. E2 is A1:V',·. Are E1 and E2 the same?

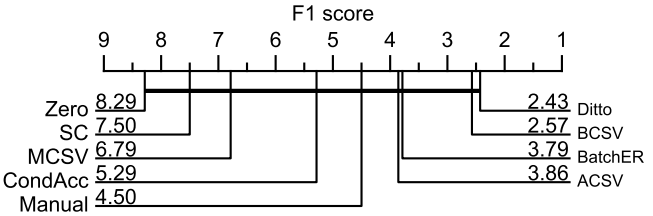
Missing Value Imputation: This is a Data Imputation task. Use domain knowledge to decide the value of $attr_j$.
Q1: $attr_1^1$:v_1^1,$attr_2^1$:v_2^1,...,$attr_n^1$:? A1: v_n^1.
Q2: $attr_1^2$:v_1^2,$attr_2^2$:v_2^2,...,$attr_n^2$:? A2: v_n^2.
...
Qk: $attr_1^k$:v_1^k,$attr_2^k$:v_2^k,...,$attr_n^k$:? Ak: v_n^k.
Answer the following question:
Q: $attr_1^0$:v_1^0,$attr_2^0$:v_2^0,...,$attr_n^0$:? A: v_n^0.

III. Additional Experiments

Critical difference diagrams. With statistical level of 0.1, we run a critical difference diagram based on wilcoxon test in Figure 5(a). Notably, on all data wrangling tasks, ACSV outperforms the manual method by 1.5 in terms of average rank. Although BCSV looks even worse than MCSV, it is severely affected by tasks other than entity matching(where the performance is 0). So we ran another wilcoxon test on 7 entity matching datasets shown in Figure 5(b). The results show that BCSV is the only method comparable to Ditto, and outperforms BatchER significantly. ACSV also approaches BatchER, and both methods(ACSV and BSCV) is much better than the naive adaption of marginal contribution, MCSV.



(a) Critical difference diagram on all data wrangling tasks



(b) Critical difference diagram on entity matching task

Fig. 1: Critical difference diagram under the statistical level of 0.1.

References

[1] L. S. Shapley, "A value for n-person games," in *Classics in Game Theory*. Princeton University Press, 1953, pp. 307–317.

[2] S. Schoch, H. Xu, and Y. Ji, "CS-Shapley: Class-wise Shapley values for data valuation in classification," in *NeurIPS*, 2022.

[3] S. Maleki, "Addressing the computational issues of the Shapley value with applications in the smart grid," Ph.D. dissertation, University of Southampton, 2015.

[4] S. Schoch, R. Mishra, and Y. Ji, "Data selection for fine-tuning large language models using transferred Shapley values," in *ACL Student Research Workshop*, 2023.

[5] J. Zhang, Q. Sun, J. Liu, L. Xiong, J. Pei, and K. Ren, "Efficient sampling approaches to Shapley value approximation," *Proceedings of the ACM on Management of Data*, vol. 1, no. 1, pp. 1–24, 2023.

[6] S. Thirumuruganathan, H. Li, N. Tang, M. Ouzzani, Y. Govind, D. Paulsen, G. M. Fung, and A. Doan, "Deep learning for blocking in entity matching: A design space exploration," *Proceedings of the VLDB Endowment*, vol. 14, no. 11, pp. 2459–2472, 2021.

[7] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, and V. Raghavendra, "Deep learning for entity matching: A design space exploration," in *SIGMOD*, 2018.