Coresets for Relational Data and The Applications

Anonymous Author(s)
Affiliation
Address
email

1 The Proof of Theorem 1

For the sake of simplicity, we use $\mathtt{PC}_i$ to denote the pseudo-cube $\mathtt{PC}_{I_{\nu_0}}(c_{\nu_0,i}, L_{\lceil \log s \rceil})$ and L to denote $L_{\lceil \log s \rceil}$ below. Suppose there is no overlap between the $\mathtt{PC}_j$s. We set the weight $w_j = |P \cap \mathtt{PC}_j|$ for each $1 \leq j \leq k$, and each center $c_{\nu_0,j} \in C_{\nu_0}$ is a representative for the set $P \cap \mathtt{PC}_j$ (we can also view each $c_{\nu_0,j}$ as a set of w_j overlapping points in the space). Since we assume the pseudo-cubes are disjoint, we have $\sum_{j=1}^{k} w_j = n$. Through Assumption 1, we have

$$\begin{aligned} & n\left|\tilde{F}(\theta) - F(\theta)\right| \\ = & \left|\sum_{c_{\nu_0,j} \in C_{\nu_0}} w_j f(\theta, c_{\nu_0,j}) - \sum_{p_i \in P} f(\theta, x_i)\right| \\ \leq & \sum_{j=1}^{k} \sum_{p_i \in P \cap \mathtt{PC}_j} |f(\theta, c_{\nu_0,j}) - f(\theta, p_i)| \\ \leq & n\alpha L^z + n\beta F(\theta). \end{aligned} \tag{1}$$

For a given ϵ_2, through Claim 1, if we set $k = |C_{\nu_0}| = \left((\frac{\alpha}{\epsilon_2})^{\frac{1}{z}} 3^{\lceil \log s \rceil} \cdot 2^{\frac{\lceil \log s \rceil^2 + 3\lceil \log s \rceil + 8}{4}}\right)^{\rho}$, we have the radius

$$r_0 = \frac{\Delta}{k^{1/\rho}} = \frac{(\frac{\epsilon_2}{\alpha})^{\frac{1}{z}}\Delta}{3^{\lceil \log s \rceil} \cdot 2^{\frac{\lceil \log s \rceil^2 + 3\lceil \log s \rceil + 8}{4}}}. \tag{2}$$

Together with Lemma 1, the above radius directly implies $L \leq (\frac{\epsilon_2}{\alpha})^{\frac{1}{z}}\Delta$. Based on (1), we have

$$\left|\tilde{F}(\theta) - F(\theta)\right| \leq \beta F(\theta) + \epsilon_2 \Delta^z. \tag{3}$$

So the set C_{ν_0} with the weights $\{w_1, \cdots, w_k\}$ yields a $(\beta, \epsilon_2)_z$-coreset.

2 The Proof of Theorem 2

Without loss of generality, we assume all the k pseudo-cubes are not empty (otherwise, we can directly remove the empty pseudo-cubes). Then we consider the pseudo-cubes one by one. For $\mathtt{PC}_1$, we directly set $w'_1 = w_1 = |P \cap \mathtt{PC}_1|$ (in the following analysis, we use w_i and w'_i to denote the exact and approximate weights for $c_{\nu_0,i}$, respectively). Suppose currently we have already obtained the values $w'_1, \cdots, w'_{i_0}$, and try to determine the value for w'_{i_0+1}. We define the following notations first.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

$$I_{i_0} = \{i \mid 1 \leq i \leq i_0, w'_i > 0\}; \tag{4}$$
$$S_{i_0} = \cup_{i \in I_{i_0}} \mathtt{PC}_i; \tag{5}$$
$$\tau_{i_0+1} = \frac{|P \cap (\mathtt{PC}_{i_0+1} \setminus S_{i_0})|}{|P \cap \mathtt{PC}_{i_0+1}|}; \tag{6}$$
$$w_{i_0+1} = |P \cap (\mathtt{PC}_{i_0+1} \setminus S_{i_0})|. \tag{7}$$

Obviously, we have $I_1 = \{1\}$ and $S_1 = \mathtt{PC}_1$.

Our algorithm for computing the approximate weight w'_{i_0+1} is as follows. We take a uniform sample of m points from $P \cap \mathtt{PC}_{i_0+1}$ by using the sampling technique for relational data [4]. Each sampled point corresponds a binary random variable x: if it belongs to $\mathtt{PC}_{i_0+1} \setminus S_{i_0}$, $x = 1$; otherwise, $x = 0$. Let g be the sum of these m random variables. Suppose τ is a fixed value $\leq 1/2$ (the exact values of m and τ will be determined in the following analysis).

- If $g/m \geq 2\tau$, set $w'_{i_0+1} = g/m \cdot |P \cap \mathtt{PC}_{i_0+1}|$.
- Else, set $w'_{i_0+1} = 0$.

Informally speaking, if $w'_{i_0+1} > 0$, we call $\mathtt{PC}_{i_0+1}$ as a "heavy pseudo-cube"; if $w'_{i_0+1} = 0$, we call $\mathtt{PC}_{i_0+1}$ as a "light pseudo-cube".

Lemma 1 *Let $\delta, \lambda \in (0, 1)$ and the sample size $m \geq \frac{3}{\delta^2 \tau} \log \frac{2}{\lambda}$. Then with probability at least $1 - \lambda$, $|w'_{i_0+1} - w_{i_0+1}|$ is no larger than either $\delta \cdot w_{i_0+1}$ or $\frac{2}{1-\delta}\tau \cdot |P \cap \mathtt{PC}_{i_0+1}|$.*

Proof. We consider two cases: (1) $\tau_{i_0+1} \geq \tau$ and (2) $\tau_{i_0+1} < \tau$.

For case (1), since $m \geq \frac{3}{\delta^2 \tau} \log \frac{2}{\lambda}$, from the Chernoff bound we know

$$(1 - \delta)\tau_{i_0+1} \leq g/m \leq (1 + \delta)\tau_{i_0+1} \tag{8}$$

with probability at least $1 - \lambda$. If the obtained ratio $g/m \geq 2\tau$, according to our algorithm, we have $w'_{i_0+1} = g/m \cdot |P \cap \mathtt{PC}_{i_0+1}|$, i.e.,

$$(1 - \delta)w_{i_0+1} \leq w'_{i_0+1} \leq (1 + \delta)w_{i_0+1}. \tag{9}$$

If the obtained ratio $g/m < 2\tau$, according to our algorithm, we have $w'_{i_0+1} = 0$. Moreover, from the left-hand side of (8), we know

$$(1 - \delta)\tau_{i_0+1} \leq 2\tau. \tag{10}$$

Therefore, $\tau_{i_0+1} \leq \frac{2\tau}{1-\delta}$. That means

$$|w'_{i_0+1} - w_{i_0+1}| = w_{i_0+1} \leq \frac{2}{1 - \delta}\tau \cdot |P \cap \mathtt{PC}_{i_0+1}|. \tag{11}$$

For case (2), from the additive form of the Chernoff bound, we have $g/m \leq 2\tau$ with probability at least $1 - \lambda$. Then we have

$$|w'_{i_0+1} - w_{i_0+1}| = w_{i_0+1} \leq \tau \cdot |P \cap \mathtt{PC}_{i_0+1}| \leq \frac{2}{1 - \delta}\tau \cdot |P \cap \mathtt{PC}_{i_0+1}|. \tag{12}$$

Combining (9), (11), and (12), we complete the proof. □

Lemma 2 *Suppose $\frac{2}{1-\delta}\tau < 1/k$ and $\tau_{i_0+1} \leq \frac{2\tau}{1-\delta}$. There exists at least one $\hat{i} \in I_{i_0}$, such that*

$$|P \cap \left(\mathtt{PC}_{i_0+1} \cap (S_{\hat{i}} \setminus S_{\hat{i}-1})\right)| \geq \frac{1}{k}|P \cap \mathtt{PC}_{i_0+1}|. \tag{13}$$

If $\hat{i} = 1$, we set $S_0 = \emptyset$.

Proof. From the definition of the S_is, we have

$$S_0 \subset S_1 \subset \cdots \subset S_{i_0}. \tag{14}$$

So we have

$$\texttt{PC}_{i_0+1} = (\texttt{PC}_{i_0+1} \setminus S_{i_0}) \bigcup \Big(\cup_{i=1}^{i_0} \big(\texttt{PC}_{i_0+1} \cap (S_i \setminus S_{i-1})\big)\Big). \tag{15}$$

It implies

$$|P \cap \texttt{PC}_{i_0+1}| = |P \cap \texttt{PC}_{i_0+1} \setminus S_{i_0}| + \sum_{i=1}^{i_0} |P \cap \texttt{PC}_{i_0+1} \cap (S_i \setminus S_{i-1})|. \tag{16}$$

Since $i_0 + 1 \leq k$, from the Pigeonhole principle, we know there exists at least one $\hat{i} \in I_{i_0}$, such that $|P \cap \big(\texttt{PC}_{i_0+1} \cap (S_{\hat{i}} \setminus S_{\hat{i}-1})\big)| \geq \frac{1}{k}|P \cap \texttt{PC}_{i_0+1}|$. □

Below, we analyze the error induced by the approximate weight w'_{i_0+1}. If $g/m \geq 2\tau$, from Lemma 1, we know that $w'_{i_0+1} \in (1 \pm \delta) w_{i_0+1}$. So it only induces an extra factor $1 \pm \delta$ to the objective value. Thus, we should require $(1 + \delta)(1 + \beta) \leq 1 + \epsilon_1$, i.e.,

$$\delta \leq \frac{\epsilon_1 - \beta}{1 + \beta}. \tag{17}$$

Then we focus on the other case, w'_{i_0+1} is set to be 0, i.e., $\texttt{PC}_{i_0+1}$ is a "light pseudo-cube". From Lemma 2 we know there exists at least one $\hat{i} \in I_{i_0}$, such that $|P \cap \big(\texttt{PC}_{i_0+1} \cap (S_{\hat{i}} \setminus S_{\hat{i}-1})\big)| \geq \frac{1}{k}|P \cap \texttt{PC}_{i_0+1}|$. Since $\texttt{PC}_{i_0+1} \cap (S_{\hat{i}} \setminus S_{\hat{i}-1}) \subset S_{\hat{i}} \setminus S_{\hat{i}-1}$, we have

$$|P \cap (S_{\hat{i}} \setminus S_{\hat{i}-1})| \geq \frac{1}{k}|P \cap \texttt{PC}_{i_0+1}|. \tag{18}$$

Actually, the set $S_{\hat{i}} \setminus S_{\hat{i}-1} = \texttt{PC}_{\hat{i}} \setminus S_{\hat{i}-1}$, so (18) implies

$$w_{\hat{i}} = |P \cap (\texttt{PC}_{\hat{i}} \setminus S_{\hat{i}-1})| \geq \frac{1}{k}|P \cap \texttt{PC}_{i_0+1}|. \tag{19}$$

Also, since $\texttt{PC}_{\hat{i}}$ and $\texttt{PC}_{i_0+1}$ are neighbors, we have

$$||c_{\nu_0,\hat{i}} - c_{\nu_0,i_0+1}|| \leq 2L. \tag{20}$$

As a consequence, for any θ in the hypothesis space, the error induced by setting $w'_{i_0+1} = 0$ is

$$\begin{aligned} w_{i_0+1} \cdot f(\theta, c_{\nu_0,i_0+1}) &\leq \frac{2}{1-\delta}\tau \cdot |P \cap \texttt{PC}_{i_0+1}| \cdot f(\theta, c_{\nu_0,i_0+1}) \\ &\leq \frac{2}{1-\delta}\tau k \cdot w_{\hat{i}}\big((1+\beta) f(\theta, c_{\nu_0,\hat{i}}) + \alpha(2L)^z\big) \\ &= \boxed{\frac{2\tau k}{1-\delta}(1+\beta)} \cdot w_{\hat{i}} f(\theta, c_{\nu_0,\hat{i}}) + \boxed{\frac{2\tau k}{1-\delta}\alpha \cdot (2L)^z} \cdot w_{\hat{i}}, \end{aligned}$$

where the first inequality comes from Lemma 1, and the second inequality comes from Assumption 1, (19), and (20). To guarantee the total multiplicative error no larger than ϵ_1 and the additive error no larger than ϵ_2, we need the following two inequalities for setting the value of τ:

$$\frac{2\tau k}{1-\delta}(1+\beta) \cdot \mathbf{k} \leq \epsilon_1 \tag{21}$$

$$\frac{2\tau k}{1-\delta}\alpha \cdot (2L)^z \cdot \mathbf{k} \leq \epsilon_2 \Delta^z. \tag{22}$$

Note that we add an extra factor k in the above two inequalities, because there are at most k light pseudo-cubes. Based on the fact $(\frac{\Delta}{2L})^z \geq (\frac{1}{2}(\frac{\alpha}{\epsilon_2})^{1/z})^z$ from Theorem 1 (usually z is a fixed constant), it is sufficient to set

$$\tau \leq \Theta(\frac{\epsilon_1}{k^2}). \tag{23}$$

Together with (17), we obtain the sample size

$$m \geq \Theta(\frac{k^2}{(\epsilon_1 - \beta)^2 \epsilon_1} \log \frac{k}{\lambda}), \tag{24}$$

where we replace the probability parameter λ by λ/k to take the union bound over all the k pseudo-cubes.

3 Assumption 1 for The Applications

k_c-Clustering[1]. Let $k_c \in \mathbb{Z}^+$. A feasible solution θ for the k_c-clustering problem is a set of k_c centers in $\mathbb{R}^d$, and each data point is assigned to the nearest center. The objective function of the k_c-means clustering problem is as follows:

$$F(\theta) = \frac{1}{n}\sum_{i=1}^{n} \min_{c\in\theta} \|p_i - c\|_2^2. \tag{25}$$

Similarly, the objective function of the k_c-center clustering is

$$F(\theta) = \max_{p\in P} \min_{c\in\theta} ||p - c||_2. \tag{26}$$

And the objective function of the k_c-median clustering is

$$F(\theta) = \frac{1}{n}\sum_{i=1}^{n} \min_{c\in\theta} ||p - c||_2. \tag{27}$$

Obviously for the k_c-center and k_c-median problems, we have $\alpha = 1, \beta = 0$, and $z = 1$. For the k_c-means problem, we consider any two points $p, q \in \mathbb{R}^d$. Denote by $c_p, c_q \in \theta$ the nearest centers to p and q, respectively. Without loss of generality, we can assume $f(\theta, p) \geq f(\theta, q)$. Let $\epsilon \in (0, 1)$. Then we have

$$\begin{aligned}
|f(\theta,p) - f(\theta,q)| &= \|p - c_p\|_2^2 - \|q - c_q\|_2^2 \\
&\leq \|p - c_q\|_2^2 - \|q - c_q\|_2^2 \\
&= \|p - q + q - c_q\|_2^2 - \|q - c_q\|_2^2 \\
&= \|p - q\|_2^2 + 2\langle p - q, q - c_q\rangle \\
&= \|p - q\|_2^2 + 2\left\langle \frac{1}{\sqrt{\epsilon}}(p - q), \sqrt{\epsilon}(q - c_q)\right\rangle \\
&\leq \|p - q\|_2^2 + \frac{1}{\epsilon}\|p - q\|_2^2 + \epsilon\|q - c_q\|_2^2 \\
&= (1 + \frac{1}{\epsilon})\|p - q\|_2^2 + \epsilon f(\theta, q).
\end{aligned} \tag{28}$$

Therefore, we have $\beta = \epsilon$, $\alpha = O(\frac{1}{\epsilon})$, and $z = 2$ for the k_c-means clustering problem.

Logistic Regression. Logistic regression is a widely used binary classification model with each data point p_i having the label $y_i \in \{0, 1\}$ [2]. Denote $g(t) := \frac{1}{1+e^{-t}}$, and the objective function of logistic regression is:

$$F(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\Big(y_i \log g\left(\langle p_i, \theta\rangle\right) + (1 - y_i)\log\left(1 - g\left(\langle p_i, \theta\rangle\right)\right)\Big). \tag{29}$$

Note that we compute the coresets for two classes separately, i.e., the label can be viewed as a fixed number (either 1 or 0). Denote $f'(\theta, t)$ as the derivative of $f(\theta, t)$ (where $t = \langle p, \theta\rangle$). Obviously, $|f'(\theta, t)| < 1$. We consider arbitrary two points $p, q \in \mathbb{R}^d$. Also we assume that they have the same label. Thus, we have

$$\begin{aligned}
|f(\theta, \langle p, \theta\rangle) - f(\theta, \langle q, \theta\rangle)| &\leq |\langle p, \theta\rangle - \langle q, \theta\rangle| \\
&= |\langle p - q, \theta\rangle| \\
&\leq \|\theta\|_2 \cdot \|p - q\|_2.
\end{aligned} \tag{30}$$

Therefore we have $\alpha = O(\|\theta\|_2), \beta = 0$, and $z = 1$.

SVM with Soft Margin. The objective function of the soft margin SVM [1] is as follows:

$$\begin{aligned}
\min_{\boldsymbol{\omega}, b, \xi_i} \quad & \tfrac{1}{2}\|\boldsymbol{\omega}\|^2 + \lambda\textstyle\sum_{i=1}^{n}\xi_i \\
s.t. \quad & y_i(\boldsymbol{\omega}^T x_i + b) \geq 1 - \xi_i \\
& \xi_i \geq 0, i \in [n].
\end{aligned} \tag{31}$$

Specifically, λ is a constant number and ξ_i can be set to be hinge loss $\ell_{hinge}(z) = \max(0, 1 - z)$ or logistic loss $\ell_{log}(z) = \log(1 + \exp(-z))$, where $z = y_i(\boldsymbol{\omega}^T x_i + b)$. Through the similar calculations as the logistic regression, we have $\alpha = O(\|\theta\|_2), \beta = 0, z = 1$.

[1] We use "k_c" instead of "k" to avoid being confused with the coreset size k

4 Experimental results

We evaluate the performance of our relational coreset on three popular machine learning problems, the k_c-means clustering, SVM with soft margin, and logistic regression. The experimental results suggest that our method can achieve promising performances on large-scale data sets, where the coreset size is significantly smaller than the design matrix. Moreover, our coreset can be constructed very efficiently with low runtime. Our approach is in general at least 40 times faster than the end-to-end time of the traditional relational learning framework. In terms of the k_c-means clustering problem, comparing with the recently proposed Rk-means[3] algorithm, our coreset method can achieve a better solution with lower construction time and coreset size, especially when the number of tables is relatively larger. All the experimental results were obtained on a server equipped with 3.0GHz Intel CPUs and 384GB main memory. Our algorithms were implemented in Python with PostgreSQL 12.10.

Data sets and Queries. We design three different join queries on the following two real relational data sets.

(1)HOME CREDIT[2] is a relational data set used for credit forecasting. It contains 7 tables including the historical credit and financial information for each applicant. The dataset has the binary labels. We use 5 of these tables to design two different queries to extract the design matrix.

- QUERY 1 (Q1) is a multi-way acyclic join that involves 5 tables, and the returned design matrix contains about 4.0×10^8 rows with 19 features. The total size is about 60GB.
- QUERY 2 (Q2) is a multi-way acyclic join that involves 4 tables, and the returned design matrix contains 8.0×10^7 rows with 17 features. The total size is about 11GB.

(2)YELP[3] is a relational data set that contains the information of user reviews in business. The data set has no label so we just use it for the clustering task. We use 3 main tables to design a join query that forms the design matrix.

- QUERY 3 (Q3) is a chain acyclic join that involves 3 tables, and the returned design matrix contains about 5.7×10^6 rows with 24 features. The total size is about 1.1GB.

These three queries are designed as follows:

$$\begin{aligned} \text{Q1} = \; & \texttt{SELECT} * \texttt{FROM Application as App, Bureau as Bur} \\ & \texttt{Previous as Pre, CreditCard as Cre, Installments as Ins} \\ & \texttt{WHERE App.sid} = \texttt{Bur.sid AND App.sid} = \texttt{Pre.sid} \\ & \texttt{AND Cre.pid} = \texttt{Pre.pid AND Ins.pid} = \texttt{Pre.pid;} \end{aligned} \tag{32}$$

$$\begin{aligned} \text{Q2} = \; & \texttt{SELECT} * \texttt{FROM Application as App, Previous as Pre,} \\ & \texttt{CreditCard as Cre, Installments as Ins} \\ & \texttt{WHERE App.sid} = \texttt{Pre.sid AND Cre.pid} = \texttt{Pre.pid} \\ & \texttt{AND Ins.pid} = \texttt{Pre.pid;} \end{aligned} \tag{33}$$

$$\begin{aligned} \text{Q3} = \; & \texttt{SELECT} * \texttt{FROM Review as Rev, Usr, Business as Bus} \\ & \texttt{WHERE Rev.uid} = \texttt{Usr.uid AND Rev.bid} = \texttt{Bus.bid.} \end{aligned} \tag{34}$$

Baseline methods. We consider two baseline methods for comparison. (1) ORIGINAL: construct the complete design matrix P by preforming the join query, and run the training algorithm directly on P; (2) Rk-MEANS: the relational k_c-means algorithm [3]. It first performs the κ-means ($\kappa \in (0, k_c]$) on each table and then constructs a grid coreset of size κ^s; (3) RCORE: our proposed relational coreset approach. The experimental results of RCORE and Rk-MEANS are averaged over 10 trials.

[2] https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset

[3] https://tianchi.aliyun.com/dataset/dataDetail?dataId=89722

Results. We consider both the running time and optimization quality. We record the runtime that includes the design matrix/coreset construction time and the training time. We define "**Speedup**" as the ratio of the ORIGINAL's end-to-end time to RCORE's time. For the optimization quality, we take the objective value $F(\theta^*)$ obtained by ORIGINAL as the baseline; we define "**Approx.**"$= \frac{F(\theta)-F(\theta^*)}{F(\theta^*)}$, where $F(\theta)$ is the objective function value obtained by RCORE.

Coreset size		200	400	600	800	1000
Construction time of P(s)				2403		
Construction + training time of RCore(s)		208	288	363	446	531
SVM	**training time on** P(s)			> 21600		
	Speedup	$> 115.10\times$	$> 83.32\times$	$> 65.96\times$	$> 53.80\times$	$> 45.20\times$
	Approx.	0.92	0.31	0.27	0.16	0.02
LR	**training time on** P(s)			686		
	Speedup	106.86	77.36	61.24	49.95	41.97
	Approx.	0.43	0.52	0.42	0.24	0.13

Table 1: The results of SVM and logistic regression on Q2.

Coreset size		200	400	600	800	1000
Construction time of P(s)				10687		
Construction + training time of RCore(s)		228	335	430	529	630
SVM	**training time on** P(s)			-		
	Speedup	$> 46.71\times$	$> 31.86\times$	$> 24.81\times$	$> 20.19\times$	$> 16.94\times$
	Loss	0.40 ± 0.18	0.33 ± 0.14	0.36 ± 0.14	0.32 ± 0.08	0.24 ± 0.03
LR	**training time on** P(s)			-		
	Speedup	$> 46.71\times$	$> 31.86\times$	$> 24.81\times$	$> 20.19\times$	$> 16.94\times$
	Loss	4.13 ± 1.70	3.21 ± 0.90	3.39 ± 1.53	3.17 ± 0.56	3.07 ± 0.31

Table 2: The results of SVM and logistic regression on Q3.

We compare ORIGINAL and RCORE on Q2 and Q3. We consider the SVM and logistic regression (LR) models, and the results are shown in Table 1 and Table 2, respectively. Note that the runtimes for training SVM and logistic regression on our coreset are always less than 1 second, and thus the end-to-end runtimes of RCORE on both the models are almost the same. For Q3, we cannot obtain $F(\theta^*)$ due to the memory limit, so we only report the losses.

For the k_c-means clustering problem, we compare the performances of RCORE and Rk-MEANS under the three queries. According to the setting in [3], κ can be less than k_c. In our experiment, we set $k_c = 10$, and set $\kappa = \{6, 7, 8, 9, 10\}$ for Rk-MEANS. Figure 1 and 2 illustrate the obtained coreset construction times and corresponding losses. The results suggest that when the number of tables is small ($s = 3$), our RCORE has a similar coreset construction time with Rk-MEANS but a lower loss. When s is relatively larger, the advantages of our RCORE in terms of the runtime and optimization quality become more significant. Moreover, as the increasing of coreset size, our RCORE has a decreasing trend in loss, while the performance of Rk-MEANS is relatively unstable.

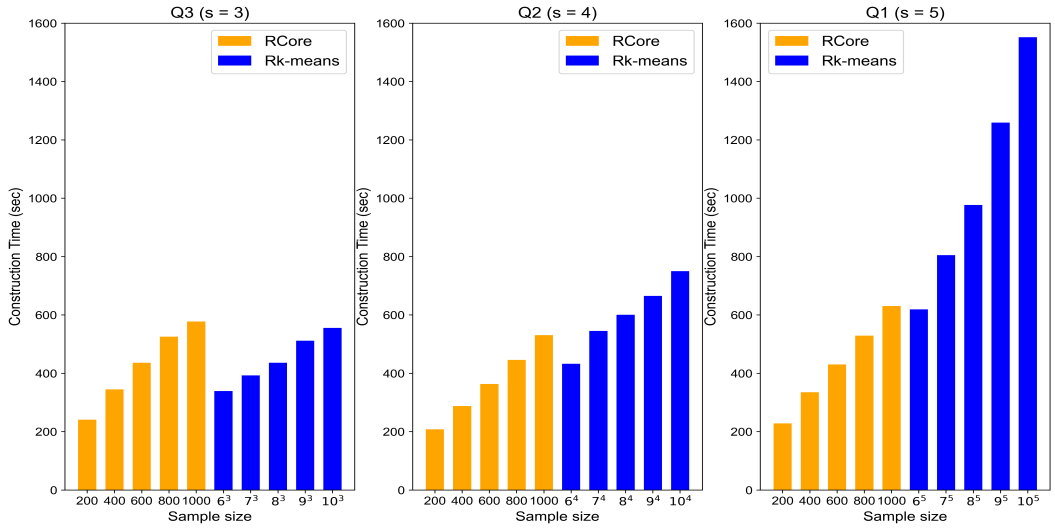

Figure 1: Construction time of RCORE and Rk-MEANS

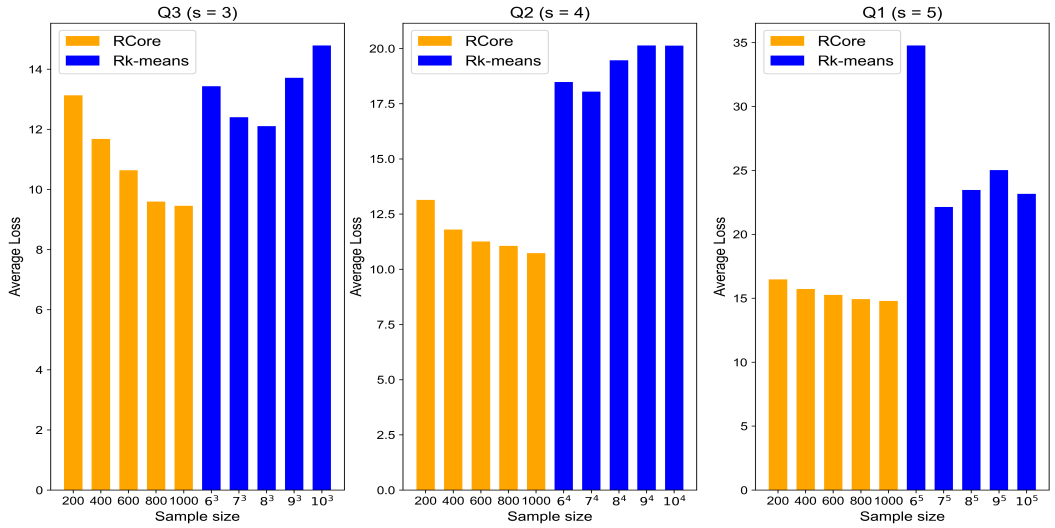

Figure 2: Average loss of RCORE and Rk-MEANS

References

[1] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[2] Jan Salomon Cramer. The early origins of the logit model. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, 35(4):613–626, 2004.

[3] Ryan R. Curtin, Benjamin Moseley, Hung Q. Ngo, XuanLong Nguyen, Dan Olteanu, and Maximilian Schleich. Rk-means: Fast clustering for relational data. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 2742–2752. PMLR, 2020.

[4] Zhuoyue Zhao, Robert Christensen, Feifei Li, Xiao Hu, and Ke Yi. Random sampling over joins revisited. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1525–1539, 2018.