Finding Simplex Items in Data Streams (Appendices)

Paper #596

January 6, 2023

(Authors' note) Due to ICDE's requirements for the number of pages of the main text, the revised version really does not fit all of this paper, and we have to put a small part in this appendix file.

A Problem Statement: Symbols

Table 1: Symbols frequently used in this paper.

Symbol	**Meaning**
e	A distinct item in data streams
k	The number of polynomial order
p	The number of consecutive windows (the definition to be true simplex items)
w	The current time window
f_w	The frequency of an item in window w
a_k	The coefficient of the highest order
ε	The mean squared error (MSE)
T	The threshold for ε
s	The number of consecutive windows (the Preliminary Condition to be potential simplex items)
$h_i(.)$	i^{th} hash function in Stage 1
q_i	The number of counters in the i^{th} array in Stage 1
u	The number of cells in Stage 2
$\mathcal{W}$	The weight of the item in Stage 2
t	The lasting time (the number of consecutive windows) of simplex items
$\mathcal{P}$	The replacement probability
L	The threshold for the highest order coefficient a_k
G	The threshold for Potential Λ
w_{str}	The starting window

B X-Sketch Design: The Effects of X-Sketch's Key Parameters

We divide the key parameters of X-Sketch into two categories: 1) parameters for problem definition; 2) parameters for algorithm design.

1) Parameters for problem definition: (1) A larger number of consecutive windows p means a stricter definition and requires more memory overhead to store the item frequencies of more windows. In other words, for a given/fixed memory size, this situation is likely to result in a decrease in accuracy. (2) A larger threshold T of MSE ε implies a greater tolerance for higher order polynomial fitting errors, and the number of simplex items satisfying the definition as well as the accuracy will increase appropriately. (3) The effect of the threshold L for the highest order coefficient a_k is similar to T, and the rationale has been shown in Section III-C.

2) Parameters for algorithm design: (1) The larger the window number s used for preliminary filtering in Stage 1, the more accurate the potential items entering Stage 2. However, the memory overhead of X-Sketch also increases as s increases, which in turn reduces the accuracy of the final simplex items. (2) The more the number of cells u in each bucket in Stage 2, the more accurate our algorithm will be (via the replacement strategy) but at the same time the processing time will be longer. (3) The larger the ratio r of the memory size of Stage 1 to the memory size of the whole X-Sketch, the more accurate the final simplex items will be. However, if r is too large, the accuracy will decrease because Stage 2 does not have enough memory to store the simplex items.

C Mathematical Analysis: Error Bound of Stage 1

Since Stage 1 of X-Sketch leverages the data structure of TowerSketch, the error bounds for Stage 1 are the same as those of TowerSketch. Next, we directly provide the error bounds and related proofs of TowerSketch.

Theorem 1. *Let $0 = \delta_0 \leq \delta_1 \leq \delta_2 \leq \delta_d$, where $\delta_i (1 \leq i \leq d)$ is the number of bits in each counter in the i^{th} counter array of Stage 1. Given an arbitrary item e, assume its frequency in window w satisfies $2^{\delta_{t-1}} - 1 \leq f_w \leq 2^{\delta_t} - 1$. Let $\tilde{f}_w$ be its frequency reported by Stage 1, and f be the number of items in window w. Given an arbitrary small positive number γ, when $f_w + \gamma f < 2^{\delta_t} - 1$, the estimation error of f_w is bounded.*

$$Pr(\tilde{f}_w \leq f_w + \gamma f) \geq 1 - \prod_{k=t}^{d} \left(\frac{1}{\gamma q_k} \right). \tag{1}$$

where q_k refers to the number of counters in the k^{th} array in Stage 1.

Proof. We define an indicator variable $I_{w,k,l}$ as

$$I_{w,k,l} = \begin{cases} 1, & h_k(f_w) = h_k(f_l) \ \wedge \ w \neq l \\ 0, & \text{otherwise} \end{cases}$$

As the d hash functions are independent from each other, we have

$$E(I_{w,k,l}) = Pr\left\{h_k(f_w) = h_k(f_l)\right\} = \frac{1}{q_k}$$

We define another variable $R_{w,k} = \sum_{l=1}^{v} f_l \cdot I_{w,k,l}$, indicating the estimation error caused by hash collisions in counter $\mathcal{A}_k[h_k(f_w)]$.

Then for $\forall k \geqslant t$, we have

$$\mathcal{A}_k[h_k(f_w)] = \begin{cases} f_w + R_{w,k}, & f_w + R_{w,k} < 2^{\delta_k} - 1 \\ +\infty, & \text{otherwise} \end{cases}$$

And we have

$$E\left(R_{w,k}\right) = E\left(\sum_{l=1}^{v} f_l \cdot I_{w,k,l}\right) = \sum_{l=1}^{v} f_l \cdot E\left(I_{w,k,l}\right) \leqslant \frac{f}{q_k}$$

Therefore, we have

$$\begin{aligned} Pr\{\hat{f_w} \geqslant f_w + \gamma \cdot f\} &= Pr\left\{\forall k \geqslant t, \mathcal{A}_k[h_k(e_w)] \geqslant f_w + \gamma \cdot f\right\} \\ &= Pr\left\{\forall k \geqslant t, f_w + R_{w,k} \geqslant f_w + \gamma \cdot f\right\} \\ &= Pr\left\{\forall k \geqslant t, R_{w,k} \geqslant \gamma \cdot f\right\} \\ &\leqslant Pr\left\{\forall k \geqslant t, \frac{R_{w,k}}{E(X_{w,k})} \geqslant \gamma \cdot q_k\right\} \end{aligned}$$

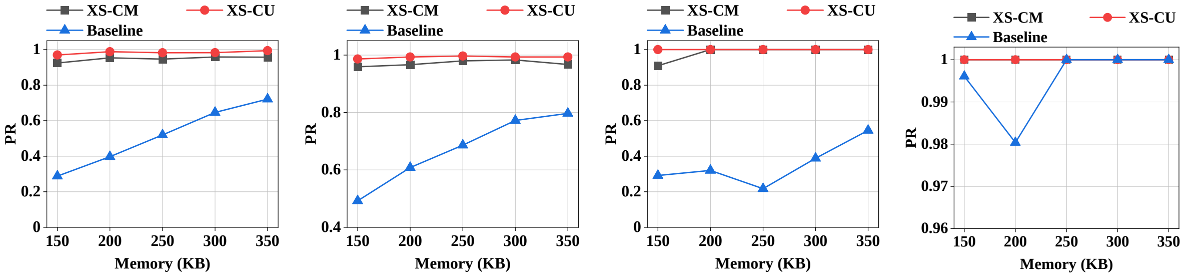
According to the Markov inequality, we can derive that

$$\begin{aligned} Pr\left\{\hat{f_w} \geqslant f_w + \gamma \cdot f\right\} &\leqslant \prod_{k=t}^{d}\left\{E\left(\frac{R_{w,k}}{E(R_{w,k})}\right) / \left(\gamma \cdot q_k\right)\right\} \\ &= \prod_{k=t}^{d}\left(\frac{1}{\gamma \cdot q_k}\right) \end{aligned}$$

Therefore, we have

$$\begin{aligned} Pr\left\{\hat{f_w} \leqslant f_w + \gamma f\right\} &= 1 - Pr\left\{\hat{f_w} \geqslant f_w + \gamma \cdot f_w\right\} \\ &\geqslant 1 - \prod_{k=t}^{d}\left(\frac{1}{\gamma \cdot q_k}\right) \end{aligned}$$

□



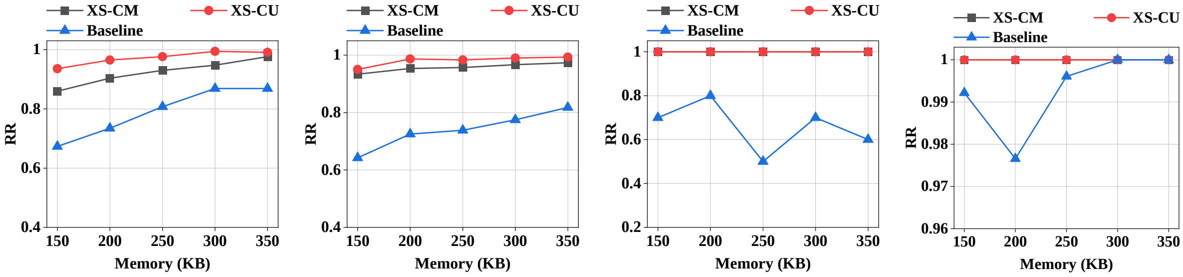
(a) IP Trace (b) MAWI (c) Data Center (d) Synthetic

Figure 24: Precision Rate (PR) on finding 3-simplex items.

D Experimental Results: Experiments on Finding 3-Simplex Items

(1) As shown in Figure 24(a)-24(d), XS-CM and XS-CU achieve 34.2% and 36.1% higher PR than the baseline solution on average, respectively.

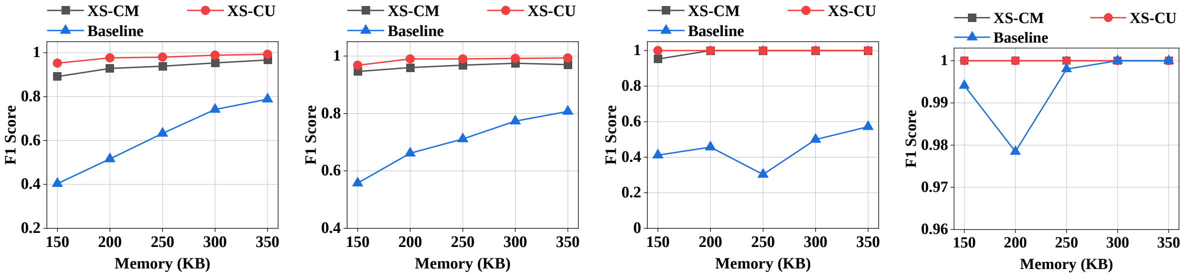
(2) As shown in Figure 25(a)-25(d), XS-CM and XS-CU achieve 17.4% and 19.3% higher RR than the baseline solution on average, respectively.



(a) IP Trace (b) MAWI (c) Data Center (d) Synthetic

Figure 25: Recall Rate (RR) on finding 3-simplex items.

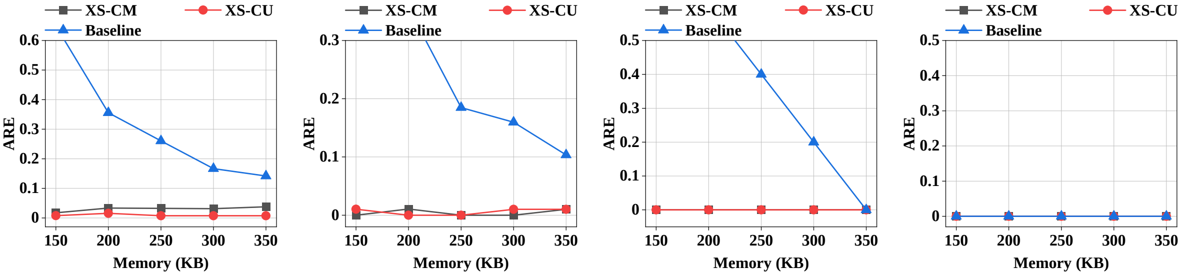
(3) As shown in Figure 26(a)-26(d), XS-CM and XS-CU achieve 28.2% and 30.1% higher F1 Score than the baseline solution on average, respectively.



(a) IP Trace (b) MAWI (c) Data Center (d) Synthetic

Figure 26: F1 Score on finding 3-simplex items.

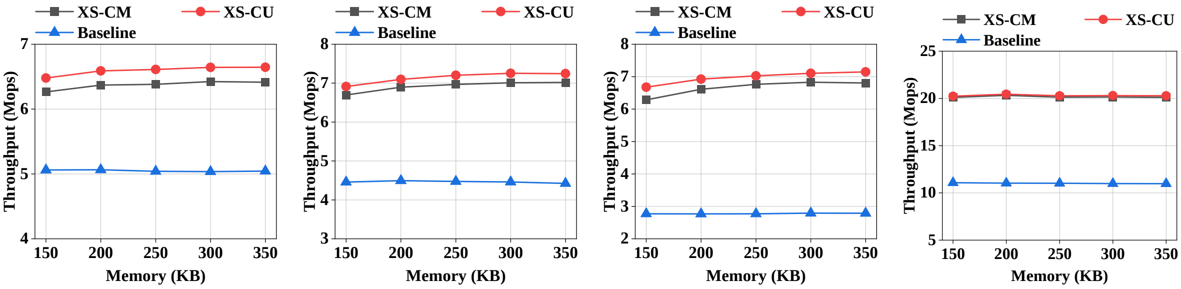
(4) As shown in Figure 27(a)-27(d), XS-CM and XS-CU achieve 32.5 and 35.7 times lower ARE than the baseline solution on average, respectively. In this case, the AREs are all 0 for both XS-CM and XS-CU on the Synthetic and Data Center datasets, and all 0 for the baseline solution on the Synthetic.



(a) IP Trace (b) MAWI (c) Data Center (d) Synthetic

Figure 27: ARE on finding 3-simplex items.

(5) As shown in Figure 28(a)-28(d), XS-CM and XS-CU achieve 1.76 and 1.82 times higher throughput than the baseline solution on average, respectively.



(a) IP Trace (b) MAWI (c) Data Center (d) Synthetic

Figure 28: Throughput on finding 3-simplex items.

E X-Sketch for ML: Experiments on Other Datasets

Here, we present experimental results run on three other datasets, including the MAWI, Data Center and Synthetic datasets introduced in Section V-A. We still implement linear regression model and time series model in Python, and X-Sketch in both C++ and Python.

E.1 Experiments on Two Real-World Datasets

Results on the MAWI Dataset. As shown in Table 4, X-Sketch's running times are 1016.8/162.4×, 1026.7/167.5×, 933.3/158.1× and 7474.74/1193.4×, 8919.5/1455.5×,

Table 4: Experiments on the MAWI Dataset.

	Model	Accuracy (%)	Running Time (s)
$k = 0$	X-Sketch (C++ / py)	99.69	0.038 / 0.238
	Linear Regression	96.62	38.64
	Time Series	99.50	284.04
$k = 1$	X-Sketch (C++ / py)	97.35	0.039 / 0.239
	Linear Regression	87.39	40.04
	Time Series	98.81	347.86
$k = 2$	X-Sketch (C++ / py)	85.14	0.042 / 0.248
	Linear Regression	74.29	39.2
	Time Series	96.60	582.38

Table 5: Experiments on the Data Center Dataset.

	Model	Accuracy (%)	Running Time (s)
$k = 0$	X-Sketch (C++ / py)	100	0.048 / 0.246
	Linear Regression	99.95	87.76
	Time Series	100	1250.99
$k = 1$	X-Sketch (C++ / py)	99.87	0.043 / 0.246
	Linear Regression	98.61	87.98
	Time Series	99.92	888.52
$k = 2$	X-Sketch (C++ / py)	98.16	0.044 / 0.254
	Linear Regression	95.81	87.97
	Time Series	99.67	842.27

Table 6: Experiments on the Synthetic Dataset.

	Model	Accuracy (%)	Running Time (s)
$k = 0$	X-Sketch (C++ / py)	100	0.012 / 0.225
	Linear Regression	98.25	3.95
	Time Series	100	117.32
$k = 1$	X-Sketch (C++ / py)	96.98	0.012 / 0.223
	Linear Regression	89.93	3.96
	Time Series	98.70	141.48
$k = 2$	X-Sketch (C++ / py)	86.47	0.014 / 0.227
	Linear Regression	75.49	3.95
	Time Series	97.94	242.7

13866.2/2348.3× faster than the linear regression and time series for $k = 0, 1, 2$, respectively.

Results on the Data Center Dataset. As shown in Table 5, X-Sketch's running times are 1828.3/356.7×, 2046.0/357.6×, 1999.3/346.3× and 26062.3/5085.3×, 20663.3/3611.9×, 19142.5/3316.0× faster than the linear regression and time series for $k = 0, 1, 2$, respectively.

E.2 Experiments on the Synthetic Dataset

Results on the Synthetic Dataset. As shown in Table 6, X-Sketch's running times are 329.2/17.6×, 330/17.8×, 282.1/17.4× and 9776.7/521.4×, 11790/634.4×, 17335.7/1069.2× faster than the linear regression and time series for $k = 0, 1, 2$, respectively.

E.3 Conclusions of Experiments on Five Datasets

As can be seen from Tables 2 to 6, both X-Sketch and the two machine learning models have shorter running times on the two synthetic datasets (Transactional, Synthetic) and longer running times on the three real-world datasets (IP Trace, MAWI, Data Center). This is related to the item characteristics of the dataset itself. For example, human-generated synthetic datasets generally have more regular item frequencies than collected real-world datasets. However, the running time of the two machine learning models on these three real-world datasets is obviously much longer than that of our X-Sketch, because X-Sketch can pick out those easy-to-predict simplex items from relatively less regular real-world datasets significantly faster and more accurately. As future work, we plan to further explore X-Sketch for machine learning on more datasets and scenarios.