Efficient Approximate Algorithms for Empirical Variance with Hashed Block Sampling

Xingguang Chen
The Chinese University of Hong Kong
China
xgchen@link.cuhk.edu.hk

Fangyuan Zhang
The Chinese University of Hong Kong
China
fzhang@se.cuhk.edu.hk

Sibo Wang
The Chinese University of Hong Kong
China
swang@se.cuhk.edu.hk

ABSTRACT

Empirical variance is a fundamental concept widely used in data management and data analytics, e.g., query optimization, approximate query processing, and feature selection. A direct solution to derive the empirical variance is scanning the whole data table, which is expensive when the data size is huge. Hence, most current works focus on approximate answers by sampling. For results with approximation guarantees, the samples usually need to be uniformly independent random, incurring high cache miss rates especially in compact columnar style layouts. An alternative uses block sampling to avoid this issue, which directly samples a block of consecutive records fitting page sizes instead of sampling one record each time. However, this provides no theoretical guarantee. Existing studies show that the practical estimations can be inaccurate as the records within a block can be correlated.

Motivated by this, we investigate how to provide approximation guarantees for empirical variances with block sampling from a theoretical perspective. Our results shows that if the records stored in a table are 4-wise independent to each other according to keys, a slightly modified block sampling can provide the same approximation guarantee with the same asymptotic sampling cost as that of independent random sampling. In practice, storing records via hash clusters or hash organized tables are typical scenarios in modern commercial database systems. Thus, for data analysis on tables in the data lake or OLAP stores that are exported from such hash-based storage, our strategy can be easily integrated to improve the sampling efficiency. Based on our sampling strategy, we present an approximate algorithm for empirical variance and an approximate top-k algorithm to return the k columns with the highest empirical variance scores. Extensive experiments show that our solutions outperform existing solutions by up to an order of magnitude.

CCS CONCEPTS

• **Theory of computation → Design and analysis of algorithms**; **Approximation algorithms analysis**.

KEYWORDS

Empirical Variance; Block Sampling; Approximate Algorithms

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539377

ACM Reference Format:
Xingguang Chen, Fangyuan Zhang, and Sibo Wang. 2022. Efficient Approximate Algorithms for Empirical Variance with Hashed Block Sampling. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3534678.3539377

1 INTRODUCTION

Empirical variance is a fundamental concept and is frequently used in data management and data analytics. For instance, existing databases maintain statistics like average, empirical variance, count, distinct values, and provide them to the query optimizers [23, 29, 41, 42], e.g., the empirical variance of the salary for the female in an employees database. In data mining, empirical variance is also an important method in analyzing the dispersion of the data and has been used in feature selections [5, 8, 25, 30].

A direct solution to derive the exact empirical variance, say for a column or an attribute X of the table, is to make a full scan of the table. This solution, however, incurs high computational costs when the data size becomes huge as pinpointed in [9]. Another straightforward solution is to maintain the mean $\mathbb{E}[X]$ of attribute X and the mean $\mathbb{E}[X^2]$ of the square of the attribute values. These results can be easily maintained during the updates of records. Then, the empirical variance of attribute X can be directly computed as $\mathbb{E}[X^2] - (\mathbb{E}[X])^2$. However, as indicated in [41], statistics on individual columns/attributes are insufficient when processing complicated queries or data analysis tasks that may involve predicates, e.g., considering records of employees only falling into a certain age group. In such cases, maintaining the mean and mean of squares for each attribute no longer works.

In the literature, most existing works focus on approximate answers via sampling records. However, uniformly random sampling incurs extremely high cache miss rates especially in compact columnar style layouts. For example, if we want to sample a record in the columnar layout, we then need to sample the value stored in each column, and record values reside in different pages for different columns. For the next records to be sampled, the accessed pages could be totally different, resulting in high cache miss rates. To alleviate such an issue, existing studies, e.g., [9, 16, 32], turn to block sampling, which samples all the records that fall into the same block. Clearly, block sampling makes full use of each accessed record and significantly reduces the cache miss rates. It also brings randomness in the block level. However, if the records within a block are highly correlated, the estimation with block sampling becomes inaccurate compared to the uniform random samples given the same number of sampled records as indicated in [9]. Thus, Chaudhuri et al. [9] investigate the correlation within each block and present solutions to adaptively adjust the sample size according to the cross-validation

error within blocks. Nevertheless, their solution still provides no theoretical guarantee on the returned approximate result.

Motivated by the limitations of existing solutions, we investigate how to derive approximation guarantees for empirical variance by block sampling from a theoretical perspective. In particular, our theoretical results show that if the records stored in a table are 4-wise independent to each other according to the keys, then we can randomly sample a position p in the table and then retrieve consecutive rows following p that fit into the same page. For instance, if a table has n records, then we first randomly sample a position p. Next, if the records are stored in a columnar layout, each column takes 4 bytes, and the block size is equal to a page size (4096 bytes). Then we will retrieve the p-th to the $((p+1023)\%n)$-th entry for each column as a block. Notice that 4-wise independence on the keys in the block does not necessarily indicate 4-wise independence on other attribute values in the block since the attribute values hashed to the same slot may have the same value. To tackle this issue, we apply a slightly modified sampling trick to the block to guarantee that for each attribute, the sampled values are still 4-wise independent. Given a block of b records, our analysis shows that if we sample b records from the block with our modified block sampling, the estimation error of the derived empirical variance can be bounded by a small value depending on b with a probability p. Meanwhile, the larger the block size b is, the smaller the estimation error is. With such properties, we design an estimation framework as follows: Firstly, we sample $O(\log n)$ blocks to boost the success probability from p to $1 - 1/n^c$ ($c \geq 1$) using the median-of-means estimator. Then, if the estimation accuracy is insufficient, we double the block size b, re-sample $O(\log n)$ blocks, and re-estimate the empirical variance. This strategy guarantees that the estimated result is becoming more and more accurate with a high success probability. The estimation process of the empirical variance stops when the required accuracy is achieved.

In practice, hash-based storage is a typical scenario in modern commercial databases. For example, Oracle 12C supports storing data with hash clusters [2] and DB2 supports storing data with hash organized tables [1]. If the tables in the data lake or the OLAP data stores are exported from such hash-based storage, then our sampling strategies can be easily integrated to improve the practical efficiency without sacrificing the approximation guarantees. Moreover, in data analysis, there exist many mining tables [13, 43] that typically contain hundreds or even thousands of columns. An important task on such mining tables is to select the appropriate features for the downstream tasks. The top-k query on empirical variance has been widely used in feature selection [8, 25, 30] and is the built-in function in existing frameworks, like Ski-learn [30]. Thus, we further devise efficient approximate algorithms for the top-k query on empirical variance. We experimentally evaluate our proposed algorithms against alternatives on large-scale real-world datasets. Experiments show that our proposed algorithms outperform existing alternatives by up to an order of magnitude.

2 PRELIMINARIES

2.1 Problem Definition

Given a sequence of n values $x_1, \cdots, x_n$, let μ denote the mean of these n values. The empirical variance σ^2 and empirical standard deviation σ are defined as:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2, \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}.$$

We only consider the empirical variance since the empirical standard deviation can be easily derived given the empirical variance score. In data analysis like approximate query processing [29] and feature selection [6, 25], we are usually given a table and are interested in the empirical variance of records in the table that satisfy specific predicates, e.g., records of employees with age group falling into 30 to 40, on each column or on certain columns. Formally, the given table consists of a set $\mathcal{D}$ of n' records where each record includes $h+1$ attributes $\alpha_0, \alpha_1, \cdots, \alpha_h$. One special attribute $\alpha_{key} = \alpha_0$ is the key of the records. Here we assume that the key is a single attribute while it is easy to extend to the case when the key is a set of attributes. In addition, the value x of a record on each attribute α_i is assume to be normalized to the range of $[0, 1]$ by $(x - min)/(max - min)$, a.k.a, the min-max normalization, where min (resp. max) is the min value (resp. max value) among all records on attribute α_i. Without normalization, the comparison of different empirical variance scores might be meaningless since the score might be mainly affected by the range of attribute values. To clarify the queries with predicates, e.g., the age range as mentioned above, we define the subset $\mathcal{D}_p$ of $\mathcal{D}$ as the set of n records satisfying predicate p and $\mathcal{D}_p(\alpha)$ as attribute values of these records with respect to attribute α. In addition, we define the selectivity θ as the ratio of the number n of records satisfying the predicate to the total number n' of records, i.e., $\theta = n/n'$. We have $n = n'$ if all records in $\mathcal{D}$ satisfy the predicate. To derive the exact empirical variance of $\mathcal{D}_p(\alpha)$, the whole table needs to be scanned, which is too expensive on massive datasets. With column store layouts, it is possible to scan related attributes only, but the cost can still be high on massive datasets. To avoid high computational costs, we consider the ϵ-approximate empirical variance defined as follows.

Definition 1 (ϵ-approximate empirical variance). *Given an error bound ϵ and a failure probability p_f, the ϵ-approximate empirical variance returns an estimation $\hat{\sigma}^2$ of empirical variance such that $|\hat{\sigma}^2 - \sigma^2| \leq \epsilon$ holds with $1 - p_f$ probability.*

To explain, we aim to provide an ϵ-absolute guarantee so that the returned estimation is close to the exact value and differs by at most ϵ. We will consider how to design algorithms for ϵ-approximate empirical variance in Section 3.1.

As pinpointed in Section 1, many mining tables include a large number of columns [13] and feature selection is usually required to handle downstream tasks. A standard approach is to select the k attributes/columns with the top-k highest empirical variance scores. Denote $\sigma^2(\alpha_i)$ as the empirical variance over the i-th attribute α_i ($0 < i \leq h$) of the records satisfying predicate p, i.e., $\mathcal{D}_p(\alpha_i)$. Then the top-k query on empirical variance is defined as follows.

Definition 2 (Top-k query). *Given the set $\mathcal{D}$ of records and a positive integer k, the top-k query on empirical variance returns the k attributes with the k highest empirical variance scores.*

Deriving exact top-k answers is still expensive. Luckily, in feature selection, approximate solutions are shown to be sufficient [18, 31]. Thus, we consider the approximate top-k query defined as follows.

Definition 3 (Approximate top-k query). *Given the set $\mathcal{D}$ of records, a positive integer k, an error bound ϵ, and a failure probability p_f, the approximate top-k query on empirical variance returns k attributes $\alpha'_1, \cdots, \alpha'_k$ such that,*

- $\left|\hat{\sigma}^2(\alpha'_i) - \sigma^2(\alpha'_i)\right| \leq \epsilon$ *for any* $i \in [1, k]$
- $\left|\hat{\sigma}^2(\alpha'_k) - \sigma^2(\alpha^*_k)\right| \leq \epsilon$

hold with at least $1 - p_f$ probability, where α^*_k is the attribute whose actual empirical variance $\sigma^2(\alpha^*_k)$ is the k-th largest.

In the above definition, the first condition indicates that the estimated empirical variance of α'_i ($1 \leq i \leq k$) is close to its exact score. For the second condition, it indicates that the estimated empirical variance of α'_k is close to the real k-th largest one. Note that when $\epsilon < (\sigma^2(a^*_k) - \sigma^2(a^*_{k+1}))/2$, the above definition returns the exact top-k answer. When ϵ is getting closer and closer to $(\sigma^2(a^*_k) - \sigma^2(a^*_{k+1}))/2$, the returned approximate top-k answers are becoming more and more accurate. Thus the quality of the approximate top-k query is guaranteed.

In our proposed solution, we will make use of 4-wise independence to obtain approximation guarantees. The formal definition of k-wise independence (4-wise is the case when $k = 4$) and k-wise independent hash functions are defined as follows.

Definition 4 (k-Wise Independence). *[19] Consider a set of discrete random variables $X_1, \cdots, X_n$. The random variables are k-wise independent if for any set $I \subseteq \{1, \cdots, n\}$ with $|I| \leq k$ and any values x_i we have:*

$$\Pr(\wedge_{i \in I} X_i = x_i) = \prod_{i \in I} \Pr(X_i = x_i).$$

Definition 5 (k-Wise Independent Hash Functions). *[36] A hash function $\hbar$ is k-wise independent if for any distinct keys $u_1, \cdots, u_k \in \{1, \cdots, s\}$ and hash values $v_1, \cdots, v_k \in \{1, \cdots, t\}$:*

$$\Pr(\hbar(u_1) = v_1 \wedge \cdots \wedge \hbar(u_k) = v_k) = \frac{1}{t^k}.$$

Notice that 4-wise independent hash function can be easily designed and takes only $O(1)$ cost to compute, which is simple and efficient. It is widely used in big data analysis, such as query optimization [4], streaming processing [14], and cardinally estimation [20]. Table 1 lists the frequently used notations in this paper.

2.2 Existing Approximate Solutions

In this section, we discuss existing approximate solutions for empirical variance that provide approximation guarantees.

Baseline solution by Chernoff bound. We first present a baseline solution that derives the empirical variance by exploiting the Chernoff bound [12]. Let X_α and Y_α be random variables generated as follows: *(i)* a record r_i is first randomly sampled from $\mathcal{D}$; *(ii)* assign the value x_i on attribute α to X_α and x_i^2 to Y_α. Let μ_α be the mean of the values on attribute α. Then, it is clear that $\mathbb{E}[X_\alpha] = \mu_\alpha$. Given X_α and Y_α, we have the following equation.

$$\mathbb{E}[Y_\alpha] - (\mathbb{E}[X_\alpha])^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^2 = \sigma^2(\alpha). \quad (1)$$

Then, we can uniformly sample random records from $\mathcal{D}$ and apply Chernoff bound to derive an estimation (with upper and lower bound) for $\mathbb{E}[Y_\alpha]$ and $\mathbb{E}[X_\alpha]$. Next, by Equation 1, we can derive the estimation (with upper and lower bound) for $\sigma^2(\alpha)$. We have the following lemma for the lower and upper bounds.

Table 1: Frequently used notations.

Notation	Description
$\mathcal{D}$	The input set of records
$\mathcal{D}_p$	The subset of records in $\mathcal{D}$ satisfying predicate p
n	The number of records satisfying predicate p in $\mathcal{D}$
n'	the number of records in $\mathcal{D}$
θ	the ratio of n to n'
h	the number of attributes in $\mathcal{D}$
m	the number of samples from $\mathcal{D}$
b, r	the block size and the number of blocks
α, A	attribute α from the set A of attributes in $\mathcal{D}$
p_f	the failure probability of the algorithm
$\sigma^2(\alpha)$	the empirical variance of α in $\mathcal{D}$
$\hat{\sigma}^2(\alpha)$	the estimation of $\sigma^2(\alpha)$
$\underline{\sigma}^2(\alpha), \overline{\sigma}^2(\alpha)$	a lower and upper bound of $\sigma^2(\alpha)$
ϵ	the error bound for approximate queries

Lemma 1. *Given a random subset $S = \{X_1, \ldots, X_m\}$ of m values from $\mathcal{D}(\alpha)$, a failure probability p'_f, and $a = \ln(4/p'_f)$, we have the lower bound $\underline{\sigma}^2$ and upper bound $\overline{\sigma}^2$ of σ^2:*

$$\underline{\sigma}^2 = \frac{\left(\sqrt{\sum_{i=1}^m X_i^2 + \frac{2a}{9}} - \sqrt{\frac{a}{2}}\right)^2 - \frac{a}{18}}{m} - \frac{\left(\sqrt{\sum_{i=1}^m X_i + \frac{a}{2}} + \sqrt{\frac{a}{2}}\right)^4}{m^2},$$

$$\overline{\sigma}^2 = \frac{\left(\sqrt{\sum_{i=1}^m X_i^2 + \frac{a}{2}} + \sqrt{\frac{a}{2}}\right)^2}{m} - \frac{\left(\left(\sqrt{\sum_{i=1}^m X_i + \frac{2a}{9}} - \sqrt{\frac{a}{2}}\right)^2 - \frac{a}{18}\right)^2}{m^2},$$

with probability at least $1 - p'_f$.

We omit the proof as it stands as our baseline and is not our focus.

State-of-the-art empirical variance bounds. The state-of-the-art empirical variance bounds are proposed by Maurer et al. [28]. Based on a concentration inequality for self-bounding random variables (Theorem 13 in [27]) and conditional expectations, they establish concentration bounds as follows.

Lemma 2. *Given a random subset $S = \{X_1, \ldots, X_m\}$ of m values from $\mathcal{D}(\alpha)$, a failure probability p'_f, and $a = \ln(2/p'_f)$, we have the lower bound $\underline{\sigma}^2$ and upper bound $\overline{\sigma}^2$ of σ^2:*

$$\underline{\sigma}^2 = \left(\sqrt{\frac{1}{m(m-1)}\sum_{i,j=1}^{m}\frac{(X_i - X_j)^2}{2}} - \sqrt{\frac{2a}{m-1}}\right)^2$$

$$\overline{\sigma}^2 = \left(\sqrt{\frac{1}{m(m-1)}\sum_{i,j=1}^{m}\frac{(X_i - X_j)^2}{2}} + \sqrt{\frac{2a}{m-1}}\right)^2$$

with probability at least $1 - p'_f$.

Deficiency of existing solutions. Unfortunately, the approximate algorithms based on the above bounds require the samples to be uniformly random and independent. This severely degrades the performance since uniformly random sampling incurs extremely

high cache miss rates especially in compact columnar style layouts. Motivated by the deficiency of existing solutions, we next present our proposed solution which achieves high efficiency while preserving the same approximation guarantee.

3 OUR SOLUTION

In this section, we investigate how to derive approximation guarantees for empirical variance by block sampling from a theoretical perspective. We show that when the records are 4-wise independent between each other according to the keys, we can sample with a slightly modified block sampling strategy. This avoids the high cache miss rates and our theoretical analysis shows that it provides the same approximation guarantee with the same time complexity as random sample based methods. As we mentioned in Section 1, in practice, hash-based storage are typical scenarios in modern commercial databases, e.g., Oracle 12C with hash clusters [2] and DB2 with hash organized tables [1]. If the tables in the data lake or the OLAP data stores are exported from such hash-based storage, then our analysis here can stand as the backbones for an efficient and effective block sampling strategy with approximation guarantees. In the following, we will focus on column-stores layout, which suffers from high cache miss rates during the random sampling.

The high-level idea of our estimation framework is as follows. We adopt block sampling by leveraging the 4-wise independence property and present an error bounded estimation. However, the probability that the estimation falls inside the correct range of ϵ' cannot be bounded with high probability, which motivates us to use the Median-of-Mean (MoM) estimator to boost the success probability. This ensures that the estimation is both efficient and accurate. Based on the MoM estimator, we derive lower and upper bounds of the empirical variance, which stands as the backbone for ϵ-approximate and approximate top-k algorithms to be introduced later. The main idea behind these algorithms is doubling the block size b adaptively and having a more accurate estimation until termination conditions are met.

3.1 Estimator of Empirical Variance

Recap that we use a 4-wise independent hash function to hash the records of the original database based on their keys. With the property of 4-wise independence, we will show that even when we sample a block of records in consecutive slots, dubbed as block sampling, to derive the estimation of the empirical variance, it still has bounded error with probability guarantees.

Consider a non-key attribute α in table $\mathcal{D}$. The position of the attribute is consistent with the key. In particular, if the position of the key attribute of a record r' is in position i in the table, then the position of the non-key attribute α of record r' is also in position i. Next, we start a block l by randomly sample a position pos_s in $\mathcal{D}$. From this position, we load a block of b consecutive records. In this block, we sample records with replacement if the consecutive records have the same hash values and keep only the sampled records satisfying the predicate in block l. If not, we directly sample them and keep the records satisfying the predicate in block l. After that, if there are less than b records in block l, we load one more block, use the same sampling method and add records satisfying the predicate to block l. We terminate the sampling process for block l when there are b records in it. This sampling strategy is still very cache-friendly since consecutive records are maintained in continuous memories.

We define a random variable X_i as the attribute value with respect to some key which is the i-th one added to the block, where $1 \le i \le b$. We will show that any four X_i, X_j, X_k, X_l are 4-wise independent using the following lemma.

Lemma 3. *Given that records are four-wise independent with respect to keys, using above block sampling strategy, any four X_i sampled from a given attribute α in a block are still 4-wise independent.*

All omitted proofs in this section are deferred to Appendix B. Define $\hat{\mu}_l$ as the average of $X_1, \ldots, X_b$, i.e., $\hat{\mu}_l = \frac{1}{b}\sum_{i=1}^{b} X_i$. Then we can calculate an unbiased estimation $\hat{\sigma}_l^2$ of the empirical variance σ^2 as $\hat{\sigma}_l^2 = \frac{1}{b-1}\sum_{i=1}^{b}(X_i - \hat{\mu}_l)^2$. We prove that $\hat{\sigma}_l^2$ is an unbiased estimation of σ^2 during the proof of Lemma 4. Thanks to the 4-wise independence of the attribute values sampled within a block, we have the following lemma to show that the estimation has a probability guarantee to approach the actual empirical variance σ^2.

Lemma 4. *If the variables $X_1, \ldots, X_b \in [0,1]$ are 4-wise independent, $\hat{\mu}_l = \frac{1}{b}\sum_{i=1}^{b} X_i$ and $\hat{\sigma}_l^2 = \frac{1}{b-1}\sum_{i=1}^{b}(X_i - \hat{\mu}_l)^2$, then*

$$\Pr(|\hat{\sigma}_l^2 - \sigma^2| > \epsilon') \le \frac{\sigma^2\left(1 - \frac{b-3}{b-1}\sigma^2\right)}{b\epsilon'^2}.$$

According to the above lemma, the more accurate estimation of σ^2 we require, the smaller ϵ' we should set. To keep the same failure probability, the block size b should be enlarged. The concentration bound derived above in Lemma 4 is not tight enough. Intuitively, when requiring a $1/n^c$ failure probability where c is a constant no smaller than 1, the block size b should be linear to the number n of records, which is impractical and motivates us to use the MoM (median-of-means) estimator [11, 22, 24, 26]. By taking the median of $O(\log n)$ sub-estimators, MoM can boost the success probability to $1 - 1/n^c$. For example, we have a sub-estimator that estimates in the correct range of ϵ' with a probability no smaller than 0.6. When using $r = O(\log n)$ such sub-estimators and taking the median of these r answers, the final result will be in the correct range of ϵ' with probability $1 - 1/n^c$. Next, we show the MoM estimator of the empirical variance as follows.

We use r sub-estimators of the empirical variance, where the determination of r is shown in Section 3.4. For each sub-estimator, we sample a block of b records as discussed before. With these r blocks, we have r estimations $\hat{\sigma}_1^2, \ldots, \hat{\sigma}_r^2$ of the empirical variance. The MoM estimator takes the median of all these sub-estimators, i.e., $\tilde{\sigma}^2 = \text{median}(\hat{\sigma}_1^2, \ldots, \hat{\sigma}_r^2)$. Then we want to show that the MoM estimator gets close to the actual empirical variance σ^2 with a probability guarantee. We define a variable Z_l for each sampling block as $Z_l = \mathbb{1}(|\hat{\sigma}_l^2 - \sigma^2| > \epsilon')$ where $\mathbb{1}$ is an indicator function. We also have $\mathbb{E}(Z_l) = \Pr(|\hat{\sigma}_l^2 - \sigma^2| > \epsilon')$. Let $p_{\epsilon',b} = \sigma^2(1 - \frac{b-3}{b-1}\sigma^2)/(b\epsilon'^2)$. Then we have the following theorem, where the proof is deferred to Appendix B.

Lemma 5. *[11, 24] MoM estimator has the following property:*

$$\Pr(|\tilde{\sigma}^2 - \sigma^2| > \epsilon') \le \exp\left(-2r\left(\frac{1}{2} - p_{\epsilon',b}\right)^2\right)$$

Algorithm 1: ϵ-approximate Empirical Variance

Input: Set $\mathcal{D}$, attribute α, probability p_f, error bound ϵ
Output: ϵ-approximate empirical variance with respect to α

1 $b \leftarrow b_0$, $i_{\max} \leftarrow \lceil \log_2 \frac{n'}{b_0} \rceil$, $p'_f \leftarrow p_f / i_{\max}$;
2 **while** $numVisitedRecords < n'$ **do**
3 Randomly sample r blocks with block size b from $\mathcal{D}$;
4 Calculate $\underline{\sigma}^2(\alpha), \overline{\sigma}^2(\alpha)$ by Lemma 1 with p'_f;
5 $\hat{\sigma}^2(\alpha) \leftarrow (\underline{\sigma}^2(\alpha) + \overline{\sigma}^2(\alpha))/2$;
6 **if** $\overline{\sigma}^2(\alpha) - \underline{\sigma}^2(\alpha) \leq 2\epsilon$ **then**
7 **return** $\hat{\sigma}^2(\alpha)$;
8 **else**
9 $b \leftarrow 2b$;
10 **return** $\sigma^2(\alpha)$;

when we have $p_{\epsilon',b} < \frac{1}{2}$.

Deriving lower and upper bounds of the empirical variance. With Lemma 5, we can derive lower and upper bounds of the empirical variance by the MoM estimator as follows.

Theorem 1. *Given a sample of r blocks where each block contains b records, a failure probability p'_f, parameters $a = \ln(1/p'_f)$ and $d = \frac{1}{2} - \sqrt{\frac{a}{2r}}$, we have the lower and upper bounds of σ^2:*

$$\sigma^2 \geq \underline{\sigma}^2 = \frac{2bd\tilde{\sigma}^2 + 1 - \sqrt{1 + 4bd\tilde{\sigma}^2\left(1 - \frac{b-3}{b-1}\tilde{\sigma}^2\right)}}{2\left(bd + \frac{b-3}{b-1}\right)}$$

$$\sigma^2 \leq \overline{\sigma}^2 = \frac{2bd\tilde{\sigma}^2 + 1 + \sqrt{1 + 4bd\tilde{\sigma}^2\left(1 - \frac{b-3}{b-1}\tilde{\sigma}^2\right)}}{2\left(bd + \frac{b-3}{b-1}\right)}$$

with probability at least $1 - p'_f$.

3.2 ϵ-Approximate Query

ϵ-approximate algorithm. We design an algorithm to return the ϵ-approximate empirical variance given an attribute α in set $\mathcal{D}$. Algorithm 1 shows the pseudo-code of this algorithm. It runs in iterations. First, we randomly sample r blocks where each block contains b records (Algorithm 1 Line 3). The block number r is fixed according to the analysis in Section 3.4. With these records, we calculate the lower and upper bounds, i.e., $\underline{\sigma}^2(\alpha)$ and $\overline{\sigma}^2(\alpha)$, of empirical variance $\sigma^2(\alpha)$ by Theorem 1 and then use the average of them as the estimated empirical variance $\hat{\sigma}^2(\alpha)$ (Algorithm 1 Lines 4-5). If the difference between $\overline{\sigma}^2(\alpha)$ and $\underline{\sigma}^2(\alpha)$ is no larger than 2ϵ, we return $\hat{\sigma}^2(\alpha)$ for the query (Algorithm 1 Lines 6-7). Else, we double the block size b (Algorithm 1 Lines 8-9). If the algorithm does not terminate when the number of visited records (we visit records and add them into blocks only if they satisfy the predicates) is no smaller than the total number n' of records, we calculate the exact empirical variance $\sigma^2(\alpha)$ by scanning the related attributes of all records and return it for the query (Algorithm 1 Line 10).

Algorithm 2: Approximate Top-k Empirical Variance

Input: Dataset $\mathcal{D}$, k, p_f, ϵ
Output: An approximate top-k query answer

1 $C \leftarrow A$, $b \leftarrow b_0$, $R \leftarrow \emptyset$, $i_{\max} \leftarrow \lceil \log_2 \frac{n'}{b_0} \rceil$, $p'_f \leftarrow \frac{p_f}{i_{\max} \cdot h}$;
2 **while** $numVisitedRecords < n'$ **do**
3 Randomly sample r blocks with block size b from $\mathcal{D}$;
4 **for** $\alpha \in C$ **do**
5 Calculate $\underline{\sigma}^2(\alpha), \overline{\sigma}^2(\alpha)$ by Lemma 1 with p'_f;
6 $R \leftarrow$ top-k attributes from C according to $\overline{\sigma}^2(\alpha)$;
7 Sort $\alpha \in R$ as $R = \{\alpha'_1, \ldots, \alpha'_k\}$ in a non-increasing order by their lower bounds;
8 **if** $\overline{\sigma}^2(\alpha'_i) - \underline{\sigma}^2(\alpha'_i) \leq 2\epsilon$ *for* $i \in [1, k]$ **then**
9 **return** R;
10 **else**
11 $b \leftarrow 2b$;
12 $\underline{\sigma}^2(\alpha''_k) \leftarrow$ the k-th largest $\underline{\sigma}^2(\alpha)$ for $\alpha \in C$;
13 **for** $\alpha \in C$ **do**
14 **if** $\overline{\sigma}^2(\alpha) < \underline{\sigma}^2(\alpha''_k)$ **then**
15 $C \leftarrow C \setminus \{\alpha\}$;
16 $R \leftarrow$ top-k attributes from C according to $\sigma^2(\alpha)$;
17 **return** R;

Theoretical Analysis. We will use the following theorem to show that Algorithm 1 returns an answer satisfying the definition of ϵ-approximate empirical variance with high probability.

Theorem 2. *Algorithm 1 returns an ϵ-approximate query of the given attribute satisfying Definition 1 with $1 - p_f$ probability.*

Theorem 3 shows the expected running time of Algorithm 1.

Theorem 3. *The expected running time of Algorithm 1 to return ϵ-approximate empirical variance is:*

$$O\left(\min\left\{n', \frac{\log(\log n / p_f)}{\epsilon^2 \theta}\right\}\right).$$

Remark. With a similar analysis, we can prove that if we use Chernoff bound or the bounds proposed by Maurer et al. [28] to derive upper and lower bounds (Algorithm 1 Line 4), the expected time complexity of these methods is the same as ours. We omit their proofs for the interest of space. This shows that the time complexity of our proposed algorithm with block sampling is asymptotically the same as state-of-the-art solutions via random sampling.

3.3 Approximate Top-k Query

Approximate top-k algorithm. Algorithm 2 shows the pseudo-code of the algorithm to answer the approximate top-k query on empirical variance. Initially, we include all attributes $\alpha \in A$ in a candidate set C (Algorithm 2 Line 1). Then, the algorithm runs in iterations. At the beginning of each iteration, we randomly sample r blocks with block size b from $\mathcal{D}$, where the total sample size is $m = r \cdot b$. The setting of r will be discussed in the next subsection. Then we calculate the lower and upper bounds of the empirical

variance, i.e., $\underline{\sigma}^2(\alpha)$ and $\overline{\sigma}^2(\alpha)$ for attributes in C (Algorithm 2 Lines 4-5) by Theorem 1. According to $\overline{\sigma}^2(\alpha)$, we find the top-$k$ attributes and put them into the result set R. We then sort attributes in R by their lower bounds $\underline{\sigma}^2(\alpha)$ and have $R = \{\alpha'_1, \ldots, \alpha'_k\}$. If $\overline{\sigma}^2(\alpha'_i) - \underline{\sigma}^2(\alpha'_i) \leq 2\epsilon$ for $i \in [1, k]$, we return R as the answer to the query (Algorithm 2 Lines 8-9). Else, we double the block size b (Algorithm 2 Lines 10-11). Besides, we prune the attributes whose upper bound $\overline{\sigma}^2(\alpha)$ is smaller than the k-th largest lower bound $\underline{\sigma}^2(\alpha''_k)$, which are impossible to become the top-k attributes (Algorithm 2 Lines 12-15). If we still cannot return R when the number of records we have visited is no smaller than the total number n' of records, we return top-k attributes from C according to their exact empirical variances (Algorithm 2 Lines 16-17).

Theoretical analysis. The following theorem will show that Algorithm 2 returns the answer satisfying the definition of the approximate top-k query with high probability.

Theorem 4. *Algorithm 2 returns a set $R = \{\alpha'_1, \ldots, \alpha'_k\}$ of k attributes. Then R will be an approximate top-k answer satisfying Definition 3 with at least $1 - p_f$ probability.*

Theorem 5 shows the expected running time of Algorithm 2.

Theorem 5. *The expected running time of Algorithm 2 to return the approximate top-k query is:*

$$O\left(\min\left\{h \cdot n', \frac{h \cdot \log(h \cdot \log n/p_f)}{\epsilon^2\theta}\right\}\right).$$

Remark. Let $a' = \log(h \cdot \log n/p_f)$, it can be proved that the expected time complexity is $O\left(\min\left\{h \cdot n', \frac{h\cdot(a'+a'^2\epsilon+\sqrt{a'^3\epsilon})}{\epsilon^2\theta}\right\}\right)$ if we use Chernoff bound to derive upper and lower bounds (Algorithm 2 Line 5) and integrate it into our top-k algorithm. The time complexity of the approximate top-k algorithm with the state-of-the-art empirical variance bounds [28] is $O\left(\min\left\{h \cdot n', \frac{h\cdot\log(h\cdot\log n/p_f)}{\epsilon^2\theta}\right\}\right)$. We omit the details for the interest of space. This shows that the time complexity of our algorithm is asymptotically the same as state-of-the-art solutions via random sampling.

3.4 Block Number

Now we discuss how to set the block number r. According to the proof of Theorem 1, we bound $\Pr(|\sigma^2 - \tilde{\sigma}^2| > \sqrt{\sigma^2(1 - \frac{b-3}{b-1}\sigma^2)/(bd)})$ with failure probability p'_f. The smaller $\sqrt{\sigma^2(1 - \frac{b-3}{b-1}\sigma^2)/(bd)}$ is, the closer σ^2 and $\tilde{\sigma}^2$ are. Given the number m of sampling records satisfying the predicates, failure probability p'_f and $a = \ln(1/p'_f)$, we require bd to be as large as possible. Since $m = r \cdot b$ and $d = \frac{1}{2} - \sqrt{\frac{a}{2r}}$, we rewrite bd as $\frac{m}{r}(\frac{1}{2} - \sqrt{\frac{a}{2r}})$. To make sure that $\frac{1}{2} - \sqrt{\frac{a}{2r}}$ is positive, $r > 2a$. Defining $\beta(r) = \frac{1}{r}(\frac{1}{2} - \sqrt{\frac{a}{2r}})$, we need to derive the maximum of $\beta(r)$ when $r > 2a$. The first derivative $\beta'(r)$ of $\beta(r)$, is $\beta'(r) = -\frac{1}{2}r^{-2} + \sqrt{\frac{9a}{8}}r^{-\frac{5}{2}}$. $\beta'(r)$ is positive when $2a < r < 4.5a$ and negative when $r > 4.5a$, and it gets 0 at $r = 4.5a$, which means that we get the maximum of $\beta(r)$ at $r = 4.5a$. Since block number r should be a positive integer, we use $\lceil 4.5a \rceil$ as the default value of r in our solution.

3.5 Optimization: Estimating the Expectation

Notice that in the derivation of empirical variance bounds in Lemma 4, we use the fact that $\mu_4 = \frac{1}{n}\sum_{i=1}^n (x_i - \mu)^4 \leq \frac{1}{n}\sum_{i=1}^n (x_i - \mu)^2 = \sigma^2$. Is there a tighter bound of μ_4? We observe that $\frac{1}{n}\sum_{i=1}^n (x_i - \mu)^4 \leq \frac{1}{n}\sum_{i=1}^n ((x_i - \mu)^2 \cdot (\max_{x_i} |x_i - \mu|)^2) = (\max_{x_i} |x_i - \mu|)^2 \frac{1}{n}\sum_{i=1}^n (x_i - \mu)^2 = (\max_{x_i} |x_i - \mu|)^2\sigma^2$. However, we cannot have μ and the x_i with max distance to μ before scanning all records. So we need an upper bound c' of $(\max_{x_i} |x_i - \mu|)^2$ based on the estimation of μ. For the lower (resp. upper) bound $\underline{\mu}$ (resp. $\overline{\mu}$) of μ, we use another MoM estimator based on r estimators $\hat{\mu}_1, \ldots, \hat{\mu}_r$, i.e., $\tilde{\mu} = \text{median}(\hat{\mu}_1, \ldots, \hat{\mu}_r)$. We will show the details of deriving $\underline{\mu}$ and $\overline{\mu}$ later. Since $x_i \in [0, 1]$, we can define c' as $c' = (\max\{\overline{\mu}, 1 - \underline{\mu}\})^2$. With the definition of c', we have $\mu_4 \leq c'\sigma^2$. Notice that when c' gets its maximum value 1, we still have $\mu_4 \leq \sigma^2$ as before.

Similar to the derivation of the empirical variance bounds, we first derive the concentration inequality to bound the gap between the estimator $\hat{\mu}$ and the expectation μ.

Lemma 6. *If the variables $X_1, \ldots, X_b \in [0, 1]$ are 4-wise independent, $\hat{\mu}_l = \frac{1}{b}\sum_{i=1}^b X_i$, then*

$$\Pr(|\hat{\mu}_l - \mu| > \epsilon') \leq \frac{\mu(1-\mu) + 3b\mu^2(1-\mu)^2}{b^3\epsilon'^4}.$$

Define a variable Z'_l for each sampling block as $Z'_l = \mathbb{1}(|\hat{\mu}_l - \mu^2| > \epsilon')$ where $\mathbb{1}$ is an indicator function. Then we have $\mathbb{E}(Z'_l) = \Pr(|\hat{\mu}_l^2 - \mu^2| > \epsilon')$. When setting $p'_{\epsilon',b}$ as $p'_{\epsilon',b} = \frac{\mu(1-\mu)+3b\mu^2(1-\mu)^2}{b^3\epsilon'^4}$, we have the following lemma.

Lemma 7. *[11, 24] MoM estimator has the following property:*

$$\Pr(|\tilde{\mu} - \mu| > \epsilon') \leq \exp\left(-2r\left(\frac{1}{2} - p'_{\epsilon',b}\right)^2\right)$$

when we have $p'_{\epsilon',b} < \frac{1}{2}$.

Deriving lower and upper bounds of the expectation. We can derive lower and upper bounds of the expectation by the MoM estimator as the following lemma.

Lemma 8. *Given a sample of r blocks where each block contains b records, a failure probability p'_f, parameters $a = \ln(2/p'_f)$ and $d = \frac{1}{2} - \sqrt{\frac{a}{2r}}$, we have the lower bound of μ:*

$$\mu \geq \underline{\mu} = \frac{2\tilde{\mu} + \frac{\sqrt{3}}{b\sqrt{d}} - \sqrt{\frac{3}{b^2 d} + \frac{4\sqrt{3}\tilde{\mu}(1-\tilde{\mu})}{b\sqrt{d}} + \frac{2}{b^2\sqrt{3d}} + \frac{2}{b^3 d}}}{2\left(1 + \frac{\sqrt{3}}{b\sqrt{d}}\right)}$$

and the upper bound:

$$\mu \leq \overline{\mu} = \frac{2\tilde{\mu} + \frac{\sqrt{3}}{b\sqrt{d}} + \sqrt{\frac{3}{b^2 d} + \frac{4\sqrt{3}\tilde{\mu}(1-\tilde{\mu})}{b\sqrt{d}} + \frac{2}{b^2\sqrt{3d}} + \frac{2}{b^3 d}}}{2\left(1 + \frac{\sqrt{3}}{b\sqrt{d}}\right)}$$

with probability at least $1 - p'_f/2$.

With the lower and upper bounds of the expectation, we have the value of c', i.e., $c' = (\max\{\overline{\mu}, 1 - \underline{\mu}\})^2$. Using $\mu_4 \leq c'\sigma^2$, we derive the tighter bounds of the empirical variance as follows.

Table 2: Summary of datasets

Dataset	Rows	Columns
Enem	49,609,963	110
Census American Housing	79,728,345	130
Census American Population	109,869,032	142
Airline Reporting Carrier On-Time	199,874,820	56

Lemma 9. *Given a sample of r blocks where each block contains b records, a failure probability p'_f, parameters $a = \ln(2/p'_f)$ and $d = \frac{1}{2} - \sqrt{\frac{a}{2r}}$, we have the lower and upper bounds of σ^2:*

$$\sigma^2 \geq \underline{\sigma}^2 = \frac{2bd\tilde{\sigma}^2 + c' - \sqrt{c'^2 + 4bd\tilde{\sigma}^2\left(c' - \frac{b-3}{b-1}\tilde{\sigma}^2\right)}}{2\left(bd + \frac{b-3}{b-1}\right)}$$

$$\sigma^2 \leq \overline{\sigma}^2 = \frac{2bd\tilde{\sigma}^2 + c' + \sqrt{c'^2 + 4bd\tilde{\sigma}^2\left(c' - \frac{b-3}{b-1}\tilde{\sigma}^2\right)}}{2\left(bd + \frac{b-3}{b-1}\right)}$$

with probability at least $1 - p'_f$.

The time complexity of the algorithms with the optimization is omitted for conciseness since when setting c' as its upper bound 1, the above bounds return back to Theorem 1.

4 EXPERIMENTS

In this section, we experimentally evaluate our proposed algorithms against alternatives. All experiments are conducted on a Linux machine with an Intel Xeon 3.7GHz CPU and 256GB memory.

4.1 Experimental Settings

Datasets. We use four large real datasets: Enem, Census American Housing (*hus*), Census American Population (*pus*) and Airline Reporting Carrier On-Time (*airline*), where each includes more than 10M records. These four datasets are publicly available and tested in [37, 38]. Each attribute in the datasets is linearly scaled to the interval $[0, 1]$ using the min-max normalizer. Table 2 shows the summary of these four large real datasets. We use a 4-wise independent hash function to hash records into hash clusters following Oracle 12C [2] and then export the tables to a compact columnar layout for each dataset. Then, we test all methods on these exported tables. In the experiments, each metric is averaged over 10 cases.

Algorithms. We compare the ϵ-approximate empirical variance and approximate top-k empirical variance algorithms using different bounds, i.e., Chernoff bound (dubbed as *Baseline*), the state-of-the-art empirical variance bounds [28] (dubbed as *COLT-Bound*), and bounds derived with our hashed block sampling (dubbed as *Hash-BS*). For algorithms with *Baseline* (resp. *COLT-Bound*), the steps are the same as Algorithms 1 and 2 except how upper and lower bounds are derived and how records are sampled. For *Baseline* (resp. *COLT-Bound*), it uses Lemma 1 (resp. 2) rather than Theorem 1 to derive $\underline{\sigma}^2(\alpha)$ and $\overline{\sigma}^2(\alpha)$; both methods sample records randomly while ours samples by blocks. In addition, we include the exact solution by scanning all records (dubbed as *Exact*). All algorithms are implemented with C++ and compiled with full optimization.

Parameter settings. All approximate algorithms include a failure probability p_f. We set $p_f = 1/n'$ for the ϵ-approximate and approximate top-k algorithms using different bounds, i.e., Chernoff bound, the state-of-the-art empirical variance bounds, and bounds derived by our hashed block sampling. These three approximate algorithms for approximate top-k empirical variance queries include an error parameter ϵ to have a trade-off between the query efficiency and accuracy. We tune ϵ in Appendix A. Experimental results show that $\epsilon = 0.01$ achieves the best trade-off between the query efficiency and accuracy on top-k queries in our algorithms. So we set $\epsilon = 0.01$ as the default value in the rest of the experiments for top-k queries. As for initial block size b_0, we set $b_0 = 512$ corresponding to the number of values in a memory page, where the page size is 4KB and each double value takes 8 bytes. Also, the initial sample size of the approximate alternatives is set as $m_0 = 512$.

4.2 Effectiveness of ϵ-Approximate Query

In the first set of experiments, we vary ϵ from 0.0025 to 0.05 and report the results when ϵ is $\{0.0025, 0.005, 0.01, 0.025, 0.05\}$ to validate the query efficiency and accuracy of the ϵ-approximate empirical variance queries on four datasets. We fix the selectivity $\theta = 0.25$ and examine the impact of ϵ. Figure 1 shows the running time of our *Hash-BS* and the alternatives. We observe that our *Hash-BS* outperforms the alternatives in all cases. Notably, our *Hash-BS* is up to 569× faster than *Exact* when $\epsilon = 0.05$ on dataset *airline*. Compared with *Baseline*, *Hash-BS* achieves up to 75× speedup when $\epsilon = 0.01$ on dataset *enem*. Also, our method is up to 16× faster than *COLT-Bound* when $\epsilon = 0.0025$ on dataset *enem*.

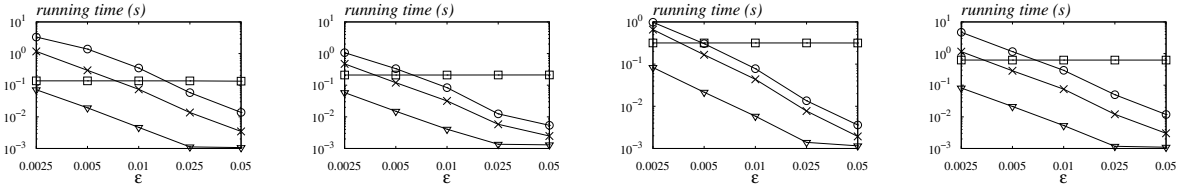
In the second set of experiments, we vary the selectivity θ from 0.1 to 1 and report the results when θ is $\{0.1, 0.25, 0.5, 1\}$ to validate the impact of θ on the query. At the same time, ϵ is fixed as 0.01. As Figure 2 shows, our algorithm consistently outperforms the alternatives in all cases again. Remarkably, our algorithm *Hash-BS* has an up to 373× speedup over *Exact* when $\theta = 0.5$ on dataset *airline*. Comparing with *Baseline*, our method has an up to 75× speedup when $\theta = 0.25$ on dataset *enem*. Besides, our *Hash-BS* is up to 16× faster than *COLT-Bound* when $\theta = 0.25$ on dataset *enem*.

We further examine the average absolute error of all methods in above two sets of experiments. Interested readers are referred to the appendix for the details. The experiments show that our solution achieves the best efficiency and smallest average absolute error in all experiments, demonstrating the effectiveness of our *Hash-BS*.

4.3 Effectiveness of Top-k Query

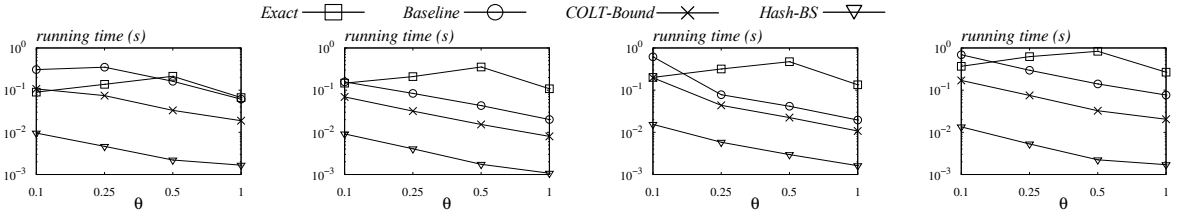
Firstly, we evaluate the query efficiency of the empirical variance top-k queries on all four datasets by varying k from 1 to 16 and fix θ as 0.25. We show the results when k is equal to $\{1, 2, 4, 8, 16\}$. Figure 3 shows the running time of our *Hash-BS* against the alternatives. We observe that *Hash-BS* outperforms the alternatives in all cases, which is up to 373× faster than *Exact* when $k = 1$ on dataset *pus*. In the comparison with *Baseline*, our *Hash-BS* is up to 74× faster when $k = 4$ on dataset *enem*. Besides, *Hash-BS* achieves an up to 23× speedup over *COLT-Bound* when $k = 1$ on dataset *pus*.

Next, we vary the selectivity θ from 0.1 to 1 fixing k as 4 and report the results when θ is $\{0.1, 0.25, 0.5, 1\}$. Figure 4 compares the running time of each algorithm. Our *Hash-BS* outperforms all



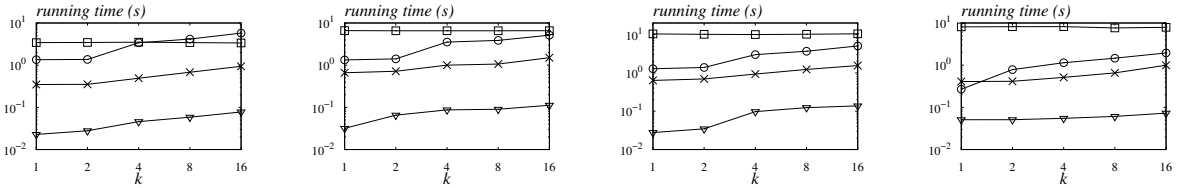
(a) enem (b) hus (c) pus (d) airline

Figure 1: Varying ϵ: Running time of ϵ-approximate empirical variance algorithms.



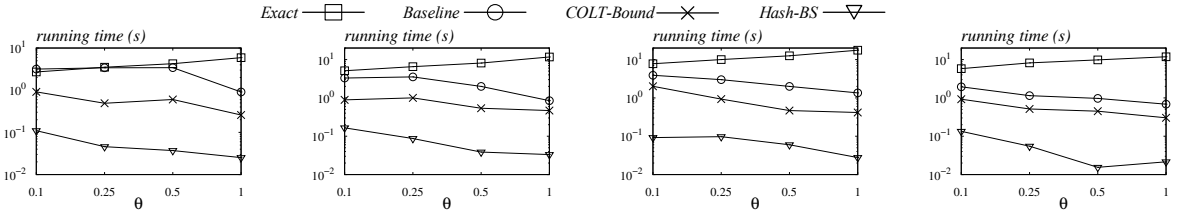
(a) enem (b) (c) pus (d) airline

Figure 2: Varying θ: Running time of ϵ-approximate empirical variance algorithms.



(a) enem (b) hus (c) pus (d) airline

Figure 3: Varying k: Running time of empirical variance top-k algorithms.



(a) enem (b) hus (c) pus (d) airline

Figure 4: Varying θ: Running time of empirical variance top-k algorithms.

Table 3: Cache miss statistics (cache-misses/10^3 records)

Method	enem	hus	pus	airline
Exact	0.560	0.541	0.593	0.659
Baseline	16.8	17.6	18.7	25.2
COLT-Bound	17.5	17.1	18.3	25.2
Hash-BS	1.17	1.04	1.09	1.49

alternatives in all cases. Remarkbly, our method is up to 636× (resp. 29×) faster than *Exact* (resp. *COLT-Bound*) when $\theta = 0.5$ on dataset *airline*. Compared with *Baseline*, *Hash-BS* has an up to 92× speedup when $\theta = 0.5$ on dataset *enem*.

In addition, we evaluate the precision for two sets of experiments about the top-k query. Due to the space limit, we defer the results in Appendix A. Experiments show that our *Hash-BS* have the best efficiency and report exact top-k answers with 100% precision.

4.4 Cache Miss Analysis

Finally, we conduct a set of experiments to do cache miss analysis on ϵ-approximate empirical variance query using different algorithms with all four datasets. We fix ϵ as 0.01 and θ as 0.25 and report the number of cache misses for every 10^3 records we retrieve in Table 3. Compared with *Exact*, our method has a little bit more cache misses every 10^3 records since the sequential scan used in *Exact* is cache-friendly. Significantly, *Hash-BS* has one order of magnitude less cache miss rates than *Baseline* and *COLT-Bound* among all datasets, which contribute to our improvements over these alternatives.

5 RELATED WORK

In feature selection, some techniques are proposed to select a small number of relevant features [8, 25]. According to whether the labels are available or not, feature selection methods are divided into supervised and unsupervised methods. Typical supervised feature selection methods include information gain [15], gini index [25] and Fisher score [17]. Without the labels, unsupervised feature selection methods are Laplacian Score [25], empirical variance [5, 8, 25, 30] and so on. In particular, a plethora of work use empirical variance in feature selection for text mining [5] and a discriminated feature gets high empirical variance.

Finding the confidence interval for the variance of a population has been studied for a rich history. If $X_1, \ldots, X_n$ are normally

distributed following $N(\mu, \sigma^2)$, an $(1-\alpha)$ confidence interval for σ^2 is $(n-1)s^2/\chi^2_{1-\alpha/2,n-1} \leq \sigma^2 \leq (n-1)s^2/\chi^2_{\alpha/2,n-1}$ proposed by Tate and Klett [35] in 1959, where $s^2 = \sum_{i=1}^{n}(X_i - \hat{\mu})^2/(n-1)$, $\hat{\mu} = \sum_{i=1}^{n} X_i/n$ and χ^2_{p,d_f} is the point on a central chi-squares distribution with d_f degrees of freedom exceeded with probability p. Burch [7] derives distribution-dependent and distribution-free confidence intervals for the variance of a population, which are asymptotically consistent. Also, some variance estimators [33, 34] without confidence intervals are proposed. Since we need to calculate the confidence interval with sample guarantees for the empirical variance, the methods above are not proper. The state-of-the-art empirical variance bounds [28] are proposed by Maurer et al. based on a concentration inequality for self-bounding random variables [27], which is one of our main competitors.

Also, there exists a plethora of work focusing on approximate top-k queries, such as [10, 21, 39, 40]. Chen et al. [10] propose efficient algorithms for approximate top-k empirical entropy and mutual information queries. Kim et al. [21] take advantage of q-grams and inverted q-gram indexes available to find the approximate top-k substring matches. Wang et al. [39, 40] consider the approximate top-k personalized PageRank queries.

6 CONCLUSION

This paper presents a hashed block sampling framework leveraging the 4-wise independence property to estimate the empirical variance efficiently. Experimental results show that our proposed solution gains up to an order of magnitude speedup over the state-of-the-art solution while providing more accurate results.

ACKNOWLEDGMENTS

This research is supported by Hong Kong RGC ECS (Grant No. 24203419), Hong Kong RGC CRF (Grant No. C4158-20G), Hong Kong ITC ITF (Grant No. MRP/071/20X), CUHK Direct Grant (Grant No. 4055181) and NSFC of China (Grant No. U1936205).

REFERENCES

[1] 2022. DB2 Hashing and Hash Organized Tables. https://www.oreilly.com/library/view/db2-developers-guide/9780132836470/ch06.html.
[2] 2022. Managing Hash Clusters. https://docs.oracle.com/database/121/ADMIN/hash.htm#ADMIN019.
[3] 2022. Technical Report. https://github.com/xgchen8/Hash-BS.
[4] Noga Alon, Phillip B. Gibbons, Yossi Matias, and Mario Szegedy. 1999. Tracking Join and Self-Join Sizes in Limited Storage. In *PODS*. 10–20.
[5] Christopher M Bishop et al. 1995. *Neural networks for pattern recognition*.
[6] Christos Boutsidis, Michael W. Mahoney, and Petros Drineas. 2008. Unsupervised feature selection for principal components analysis. In *SIGKDD*. 61–69.
[7] Brent D. Burch. 2017. Distribution-dependent and distribution-free confidence intervals for the variance. *Stat. Methods Appl.* 26, 4 (2017), 629–648.
[8] Deng Cai, Chiyuan Zhang, and Xiaofei He. 2010. Unsupervised feature selection for multi-cluster data. In *SIGKDD*. 333–342.
[9] Surajit Chaudhuri, Gautam Das, and Utkarsh Srivastava. 2004. Effective Use of Block-Level Sampling in Statistics Estimation. In *SIGMOD*. 287–298.
[10] Xingguang Chen and Sibo Wang. 2021. Efficient Approximate Algorithms for Empirical Entropy and Mutual Information. In *SIGMOD*. 274–286.
[11] Yen-Chi Chen. 2020. A short note on the median-of-means estimator. http://faculty.washington.edu/yenchic/short_note/note_MoM.pdf.
[12] Fan R. K. Chung and Lincoln Lu. 2006. Survey: Concentration Inequalities and Martingale Inequalities: A Survey. *Internet Math.* 3, 1 (2006), 79–127.
[13] John Clear, Debbie Dunn, Brad Harvey, Michael L. Heytens, Peter Lohman, Abhay Mehta, Mark Melton, Lars Rohrberg, Ashok Savasere, Robert M. Wehrmeister, and Melody Xu. 1999. NonStop SQL/MX Primitives for Knowledge Discovery. In *SIGKDD*. 425–429.
[14] Graham Cormode and Minos N. Garofalakis. 2005. Sketching Streams Through the Net: Distributed Approximate Query Tracking. In *PVLDB*. 13–24.
[15] Thomas M. Cover and Joy A. Thomas. 2001. *Elements of Information Theory*.
[16] Arnaud Doucet, Mark Briers, and Stéphane Sénécal. 2006. Efficient block sampling strategies for sequential Monte Carlo methods. *Journal of Computational and Graphical Statistics* 15, 3 (2006), 693–711.
[17] Richard O. Duda, Peter E. Hart, and David G. Stork. 2001. *Pattern classification, 2nd Edition*.
[18] Pablo A. Estévez, M. Tesmer, Claudio A. Perez, and Jacek M. Zurada. 2009. Normalized Mutual Information Feature Selection. *IEEE Trans. Neural Networks* 20, 2 (2009), 189–201.
[19] Nick Harvey. 2014. k-wise independence. https://www.cs.ubc.ca/~nickhar/W15/Lecture13Notes.pdf.
[20] Yesdaulet Izenov, Asoke Datta, Florin Rusu, and Jun Hyung Shin. 2021. COMPASS: Online Sketch-based Query Optimization for In-Memory Databases. In *SIGMOD*. 804–816.
[21] Younghoon Kim and Kyuseok Shim. 2013. Efficient top-k algorithms for approximate substring matching. In *SIGMOD*. 385–396.
[22] Guillaume Lecué and Matthieu Lerasle. 2020. Robust machine learning by median-of-means: theory and practice. *The Annals of Statistics* 48, 2 (2020), 906–931.
[23] Viktor Leis, Andrey Gubichev, Atanas Mirchev, Peter A. Boncz, Alfons Kemper, and Thomas Neumann. 2015. How Good Are Query Optimizers, Really? *PVLDB* 9, 3 (2015), 204–215.
[24] Matthieu Lerasle. 2019. Lecture Notes: Selected topics on robust statistical learning theory. arXiv:1908.10761 [stat.ML]
[25] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. 2018. Feature Selection: A Data Perspective. *ACM Comput. Surv.* 50, 6 (2018), 94:1–94:45.
[26] Gábor Lugosi and Shahar Mendelson. 2019. Sub-Gaussian estimators of the mean of a random vector. *The annals of statistics* 47, 2 (2019), 783–794.
[27] Andreas Maurer. 2006. Concentration inequalities for functions of independent variables. *Random Struct. Algorithms* 29, 2 (2006), 121–138.
[28] Andreas Maurer and Massimiliano Pontil. 2009. Empirical Bernstein Bounds and Sample-Variance Penalization. In *COLT*.
[29] Yongjoo Park, Barzan Mozafari, Joseph Sorenson, and Junhao Wang. 2018. VerdictDB: Universalizing Approximate Query Processing. In *SIGMOD*. 1461–1476.
[30] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12 (2011), 2825–2830.
[31] Hanchuan Peng, Fuhui Long, and Chris H. Q. Ding. 2005. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 8 (2005), 1226–1238.
[32] Smriti R. Ramakrishnan, Garret Swart, and Aleksey Urmanov. 2012. Balancing reducer skew in MapReduce workloads using progressive sampling. In *SOCC*. 16.
[33] Richard M Royall and William G Cumberland. 1978. Variance estimation in finite population sampling. *J. Amer. Statist. Assoc.* 73, 362 (1978), 351–358.
[34] Rajesh Singh and Sachin Malik. 2014. Improved estimation of population variance using information on auxiliary attribute in simple random sampling. *Appl. Math. Comput.* 235 (2014), 43–49.
[35] Robert F Tate and Gerald W Klett. 1959. Optimal confidence intervals for the variance of a normal distribution. *Journal of the American statistical Association* 54, 287 (1959), 674–682.
[36] Mikkel Thorup and Yin Zhang. 2012. Tabulation-Based 5-Independent Hashing with Applications to Linear Probing and Second Moment Estimation. *SIAM J. Comput.* 41, 2 (2012), 293–331.
[37] Chi Wang and Kaushik Chakrabarti. 2018. Efficient Attribute Recommendation with Probabilistic Guarantee. In *SIGKDD*. 2387–2396.
[38] Chi Wang and Bailu Ding. 2019. Fast Approximation of Empirical Entropy via Subsampling. In *SIGKDD*. 658–667.
[39] Sibo Wang, Renchi Yang, Runhui Wang, Xiaokui Xiao, Zhewei Wei, Wenqing Lin, Yin Yang, and Nan Tang. 2019. Efficient Algorithms for Approximate Single-Source Personalized PageRank Queries. *ACM Trans. Database Syst.* 44, 4 (2019), 18:1–18:37.
[40] Sibo Wang, Renchi Yang, Xiaokui Xiao, Zhewei Wei, and Yin Yang. 2017. FORA: Simple and Effective Approximate Single-Source Personalized PageRank. In *SIGKDD*. 505–514.
[41] Mohamed Zaït, Sunil Chakkappen, Suratna Budalakoti, Satyanarayana R. Valluri, Ramarajan Krishnamachari, and Alan Wood. 2017. Adaptive Statistics in Oracle 12c. *PVLDB* 10, 12 (2017), 1813–1824.
[42] Kai Zeng, Shi Gao, Jiaqi Gu, Barzan Mozafari, and Carlo Zaniolo. 2014. ABS: a system for scalable approximate queries with accuracy guarantees. In *SIGMOD*. 1067–1070.
[43] Chaoqun Zhan, Maomeng Su, Chuangxian Wei, Xiaoqiang Peng, Liang Lin, Sheng Wang, Zhe Chen, Feifei Li, Yue Pan, Fang Zheng, and Chengliang Chai. 2019. AnalyticDB: Real-time OLAP Database System at Alibaba Cloud. *PVLDB* 12, 12 (2019), 2059–2070.

A SUPPLEMENTARY EXPERIMENTS

Accuracy of ϵ-Approximate Query. For the accuracy of the ϵ-approximate query with various ϵ, we measure the average absolute error (dubbed as *AAE*) of the returned values shown in Figure 5. We can observe that the average relative error is remarkably smaller than the ϵ we set and our solution has a smaller average absolute error than the alternative approximate solutions. As for the accuracy of ϵ-approximate query with various θ, all approximate methods provide similarly accurate results as shown in Figure 6.

Accuracy of Top-k Query. As shown in Figure 7, our *Hash-BS* reports the exact top-k answers with 100% precision for the approximate top-k query with k from 1 to 16. In the set of experiments analysing the impact of θ, *Hash-BS* still reports the exact top-k answers with 100% precision as shown in Figure 8.

Tuning Error Bound ϵ of Top-k Query. We also examine the trade-off between the query efficiency and accuracy of our *Hash-BS* for the approximate top-k query on empirical variance. We vary ϵ with $\{0.0025, 0.005, 0.01, 0.025, 0.05\}$ when fixing $k = 4$ and $\theta = 0.25$) on all datasets. Figure 9 shows the results when tuning ϵ. As ϵ increases, the running time decreases on all datasets. Besides, ϵ influences the accuracy, e.g., precision, of the queries. As shown in Figure 9(b), when ϵ increases from 0.01 to 0.025, the precision starts to become smaller than 100% on dataset *hus*. Therefore, we choose $\epsilon = 0.01$ as the default value for top-k queries.

B PROOFS

In this section, we provide some omitted proofs in Section 3. We defer the proofs of Theorems 4-5 and Lemmas 3-9 to the technical report, which is included in the supplementary material [3].

Proof of Theorem 1. We discuss the lower and upper bounds of the empirical variance as follows. For conciseness, we define a temporary variable γ as $\gamma = \sqrt{1 + 4bd\tilde{\sigma}^2\left(1 - \frac{b-3}{b-1}\tilde{\sigma}^2\right)}$. We have

$$\Pr\left(\sigma^2 < \frac{2bd\tilde{\sigma}^2 + 1 - \gamma}{2\left(bd + \frac{b-3}{b-1}\right)}\right) + \Pr\left(\sigma^2 > \frac{2bd\tilde{\sigma}^2 + 1 + \gamma}{2\left(bd + \frac{b-3}{b-1}\right)}\right)$$

$$= \Pr\left(\frac{2bd\tilde{\sigma}^2 + 1}{2\left(bd + \frac{b-3}{b-1}\right)} - \sigma^2 > \frac{\gamma}{2\left(bd + \frac{b-3}{b-1}\right)}\right)$$

$$+ \Pr\left(\sigma^2 - \frac{2bd\tilde{\sigma}^2 + 1}{2\left(bd + \frac{b-3}{b-1}\right)} > \frac{\gamma}{2\left(bd + \frac{b-3}{b-1}\right)}\right)$$

$$= \Pr\left(\left|\sigma^2 - \frac{2bd\tilde{\sigma}^2 + 1}{2\left(bd + \frac{b-3}{b-1}\right)}\right| > \frac{\gamma}{2\left(bd + \frac{b-3}{b-1}\right)}\right)$$

$$= \Pr\left(\left(\sigma^2 - \frac{2bd\tilde{\sigma}^2 + 1}{2\left(bd + \frac{b-3}{b-1}\right)}\right)^2 > \frac{\gamma^2}{4\left(bd + \frac{b-3}{b-1}\right)^2}\right)$$

$$= \Pr\left(\left(bd + \frac{b-3}{b-1}\right)(\sigma^2)^2 - (2bd\tilde{\sigma}^2 + 1)\sigma^2 + bd(\tilde{\sigma}^2)^2 > 0\right).$$

With the transformation, the above formula is equal to

$$\Pr\left((\sigma^2)^2 - 2\sigma^2\tilde{\sigma}^2 + (\tilde{\sigma}^2)^2 > \frac{\sigma^2\left(1 - \frac{b-3}{b-1}\sigma^2\right)}{bd}\right)$$

$$= \Pr\left(|\sigma^2 - \tilde{\sigma}^2| > \sqrt{\frac{\sigma^2\left(1 - \frac{b-3}{b-1}\sigma^2\right)}{bd}}\right).$$

Let $\epsilon' = \sqrt{\frac{\sigma^2\left(1 - \frac{b-3}{b-1}\sigma^2\right)}{bd}}$ and we have $p_{\epsilon',b} = \frac{bd}{b} = d = \frac{1}{2} - \sqrt{\frac{a}{2r}} < \frac{1}{2}$. Applying Lemma 5, we have

$$\Pr(|\tilde{\sigma}^2 - \sigma^2| > \epsilon') \le \exp\left(-2r\left(\sqrt{\frac{a}{2r}}\right)^2\right) = \exp(-a) = p'_f,$$

which completes the proof of Theorem 1. □

Proof of Theorem 2. Since we only answer the query for one attribute α, we omit the bracket including α for conciseness. Consider the first case that we return the estimation $\hat{\sigma}^2$ as the answer when the difference between $\overline{\sigma}^2$ and $\underline{\sigma}^2$ is no larger than 2ϵ. Recall that $\hat{\sigma}^2$ is the average of $\underline{\sigma}^2$ and $\overline{\sigma}^2$. Besides, we have $\underline{\sigma}^2 \le \sigma^2 \le \overline{\sigma}^2$. So the distance between $\hat{\sigma}^2$ and σ^2 is at most half of the difference between $\overline{\sigma}^2$ and $\underline{\sigma}^2$, i.e., $|\hat{\sigma}^2 - \sigma^2| \le (\overline{\sigma}^2 - \underline{\sigma}^2)/2$. Since $\overline{\sigma}^2 - \underline{\sigma}^2 \le 2\epsilon$ in this case, we have $|\hat{\sigma}^2 - \sigma^2| \le \epsilon$, satisfying Definition 1. In the second case we return σ^2 for the query. Obviously, the exact solution will satisfy Definition 1.

The above analysis requires that the derived bounds hold for all iterations. Since the probability of $\sigma^2 \notin [\underline{\sigma}^2, \overline{\sigma}^2]$ is at most p'_f and there are at most $i_{\max} = \lceil \log_2(n'/b_0) \rceil$ iterations, the total failure probability is at most $i_{\max} \cdot p'_f = p_f$. Therefore, Algorithm 1 will return an ϵ-approximate empirical variance answer satisfying Definition 1 with at least $1 - p_f$ probability. □

Proof of Theorem 3. For a given attribute α, the first termination condition of Algorithm 1 is that the difference between the upper and lower bounds of the empirical variance, i.e., $\overline{\sigma}^2 - \underline{\sigma}^2$, is no larger than 2ϵ. Recall the expressions of the lower and upper bounds in Theorem 1. Then we require that

$$\overline{\sigma}^2 - \underline{\sigma}^2 = \frac{\sqrt{1 + 4bd\tilde{\sigma}^2\left(1 - \frac{b-3}{b-1}\tilde{\sigma}^2\right)}}{bd + \frac{b-3}{b-1}} \le \frac{\sqrt{1 + 4bd\tilde{\sigma}^2}}{bd} \le 2\epsilon$$

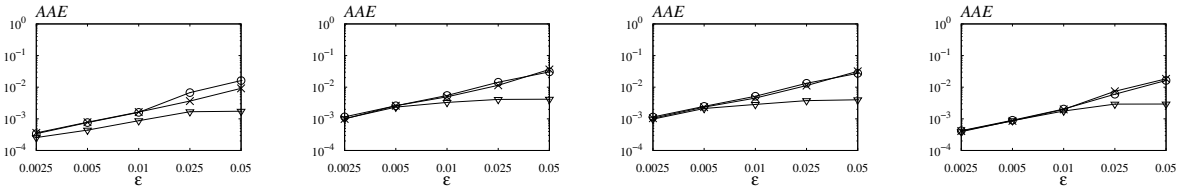
where $d = \frac{1}{2} - \sqrt{\frac{a}{2r}}$ and $a = \ln(1/p'_f)$. The above inequality is equivalent to $4b^2d^2\epsilon^2 - 4bd\tilde{\sigma}^2 - 1 \ge 0$. Solving this inequality, then we require block size

$$b \ge \frac{\tilde{\sigma}^2 + \sqrt{(\tilde{\sigma}^2)^2 + \epsilon^2}}{2d\epsilon^2}.$$

We also have

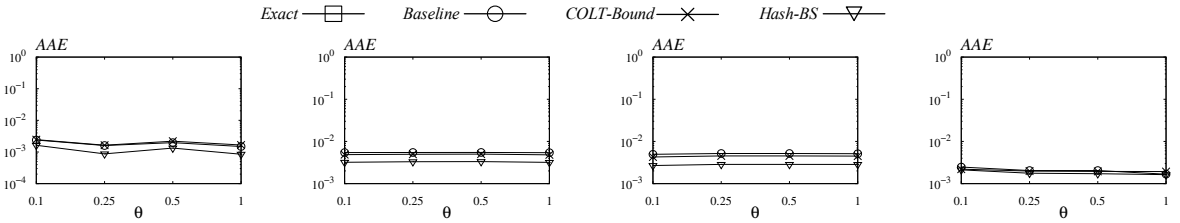
$$\frac{\tilde{\sigma}^2 + \sqrt{(\tilde{\sigma}^2)^2 + \epsilon^2}}{2d\epsilon^2} \le \frac{\tilde{\sigma}^2 + \sqrt{(\tilde{\sigma}^2)^2} + \sqrt{\epsilon^2}}{2d\epsilon^2} \le \frac{\epsilon + 2}{2d\epsilon^2}.$$

When the number $m = r \cdot b$ of sampling records satisfying the predicates is no smaller than $r(\epsilon + 2)/(2d\epsilon^2) \triangleq m^*$, we have $\overline{\sigma}^2 - \underline{\sigma}^2 \le 2\epsilon$ and the stopping condition in this case will be satisfied.



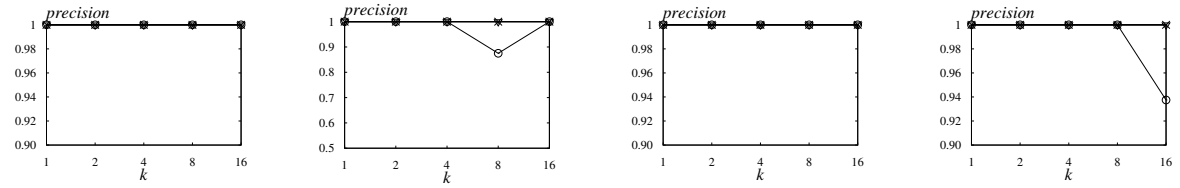
(a) enem (b) hus (c) pus (d) airline

Figure 5: Varying ϵ: Average absolute error of ϵ-approximate empirical variance algorithms.



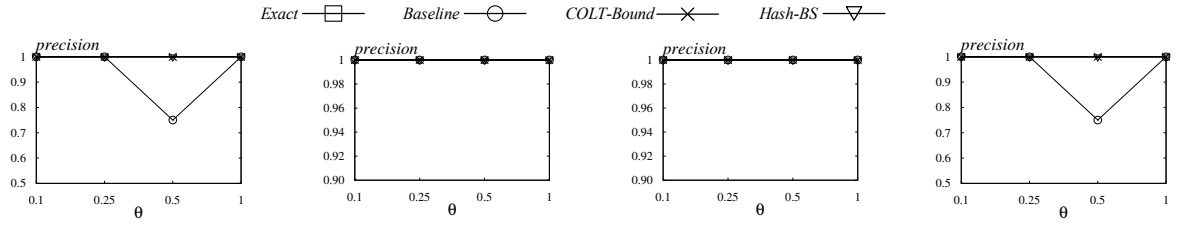
(a) enem (b) hus (c) pus (d) airline

Figure 6: Varying θ: Average absolute error of ϵ-approximate empirical variance algorithms.



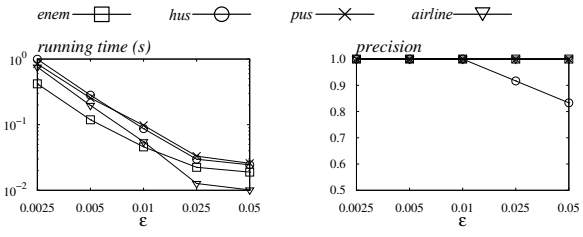
(a) enem (b) hus (c) pus (d) airline

Figure 7: Varying k: Query Precision of empirical variance top-k algorithms.



(a) enem (b) hus (c) pus (d) airline

Figure 8: Varying θ: Query Precision of empirical variance top-k algorithms.



(a) Running Time (b) Precision

Figure 9: Tuning ϵ: approximate top-k.

In the algorithm, the sample size m will double in each iteration and check whether m is large enough to satisfy the termination condition. So the algorithm terminates with $m \leq 2m^*$ with at least $1-p_f$ probability. The total number M^* of records we have ever added to the blocks for an attribute is at most $4m^*$ since we resample and double the number of records for each iteration, which is $O\left(\frac{r}{d\epsilon^2}\right)$. According to the previous analysis, $r = \lceil 4.5a \rceil$ and d is a constant with this setting where $a = \ln(1/p'_f)$. In Algorithm 1, $i_{\max} = \log_2 \lceil n/b_0 \rceil$ and $p'_f = p_f/i_{\max}$. So M^* is $O(\log(\log n/p_f)/\epsilon^2)$.

The second stopping condition indicates that the number of records we have visited is no larger than the total number n' of records. Recall that θ is the ratio of the number of records satisfying the predicates to the total number of records. Since all records are hashed and we use the block sampling strategy, the ratio of M^* to the total number of sampled records is also θ in expectation. Then the expected running time of the ϵ-approximate algorithm is

$$O\left(\min\left\{n', \frac{\log(\log n/p_f)}{\epsilon^2\theta}\right\}\right),$$

which completes the proof of the theorem. □