Knowl Inf Syst (2012) 32:281–301
DOI 10.1007/s10115-011-0441-1

REGULAR PAPER

A sampling approach for skyline query cardinality estimation

Cheng Luo · Zhewei Jiang · Wen-Chi Hou · Shan He · Qiang Zhu

Received: 12 July 2010 / Revised: 12 July 2011 / Accepted: 27 August 2011 /
Published online: 16 September 2011

Abstract A skyline query returns a set of candidate records that satisfy several preferences. It is an operation commonly performed to aid decision making. Since executing a skyline query is expensive and a query plan may combine skyline queries with other data operations such as join, it is important that the query optimizer can quickly yield an accurate cardinality estimate for a skyline query. Log Sampling (LS) and Kernel-Based (KB) skyline cardinality estimation are the two state-of-the-art skyline cardinality estimation methods. LS is based on a hypothetical model $A(\log(n))^B$. Since this model is originally derived under strong assumptions like data independence between dimensions, it does not apply well to an arbitrary data set. Consequently, LS can yield large estimation errors. KB relies on the integration of the estimated probability density function (PDF) to derive the scale factor Ψ_{ds}. As the estimation of PDF and the ensuing integration both involve complex mathematical calculations, KB is time consuming. In view of these problems, we propose an innovative purely sampling-based (PS) method for skyline cardinality estimation. PS is non-parametric. It does not assume any particular data distribution and is, thus, more robust than LS. PS does

C. Luo (✉)
Department of Mathematics and Computer Science, Coppin State University,
2500 West North Avenue, Baltimore, MD 21216, USA
e-mail: cluo@coppin.edu

Z. Jiang
Frederick Community College, 7932 Opossumtown Pike, Frederick, MD 21702, USA

W.-C. Hou
Computer Science Department, Southern Illinois University Carbondale,
Carbondale, IL 62901, USA

S. He
School of Economics and Management, Southwest Petroleum University,
Chengdu 610500, Sichuan, People's Republic of China

Q. Zhu
Department of Computer and Information Science, University of Michigan,
Dearborn, MI 48128, USA

not require complex mathematical calculations. Therefore, it is much simpler to implement and much faster to yield the estimates than KB. Extensive empirical studies show that for a variety of real and synthetic data sets, PS outperforms LS in terms of estimation speed, estimation accuracy, and estimation variability under the same space budget. PS outperforms KB in terms of estimation speed and estimation variability under the same performance mark.

Keywords Skyline query · Cardinality estimation · Sampling

1 Introduction

Considering a multi-dimensional data space, a point $p1$ dominates another point $p2$ if $p1$ is better than $p2$ on at least one dimension and not worse than $p2$ on the other dimensions. If we regard smaller values as better, point dominance can be formally defined as follows. Given a d-dimensional data set ds, a point $p1 \in ds$ dominates another point $p2 \in ds$ **iff** $\exists i$, where $1 \leq i \leq d$, such that $p1_i < p2_i$ and $\forall j$ where $1 \leq j \leq d$ and $j \neq i$, such that $p1_j \leq p2_j$. A skyline query on a data set is to find the set of points that are not dominated by any others.

To perform a skyline query, the user specifies whether smaller or larger values are preferred for each attribute of interest. For instance, considering Table 1 that contains the hotel information, suppose the two most important factors for the user to choose a hotel is the price and the distance between the hotel and the conference venue, naturally, the user prefers the price to be lower and the distance to be shorter. With these criteria, performing a skyline query on Table 1 would return hotels h1, h2, h3, and h4. Notice that hotel h5 is eliminated because it is dominated by at least one of the other hotels, for instance, hotel h1. As this example demonstrates, a skyline query returns a set of candidates when multiple preferences cannot be satisfied simultaneously by a single record. The decision maker can decide the relative weight for each preference later and choose the best candidate accordingly.

The skyline problem was first studied in the discipline of computational geometry, where it is known as the Maximal Vector Problem [4]. Later, Börzsönyi et al. [6] introduced the skyline problem to the database community. Since then, it has sparked many researches. For instance [2,20,27], to name a few. Evaluating a skyline query is expensive. Among the many proposed evaluation algorithms [1,4–6,10,14,22,23], the Double Divide-and-Conquer algorithm [22] has the lowest worst-case time complexity of $O(n^2 \log(d-2))$, where n is the data set size and $d \geq 2$ the dimensionality. Moreover, practical applications usually combine skyline queries with join operations. Therefore, it is essential that an accurate cardinality estimate for a skyline query can be derived efficiently. The query optimizer will then be able to use the estimates to choose the best execution plan.

Sampling has been applied extensively to cardinality estimations [8,11,16–19,24]. The popularity of sampling benefits from at least the following four aspects. Firstly, sampling

Table 1 A skyline application

HotelId	Price($)	Distance(mi)
h1	70	6
h2	85	4
h3	127	3
h4	60	7
h5	140	7

evaluates only a fraction of the original data set. It is a cost-effective approach particularly suitable for studying large databases. Secondly, compared with other well-known compression or approximation methods such as histogram [15,26], cosine transform [7], and wavelet [12,25], sampling is the least affected by the increase of dimensionality known as the "curse of dimensionality" [3]. It is, thus, a strong candidate to handle high-dimensional data. Thirdly, sampling is a non-parametric approach. It assumes no particular distribution of the original data set. Hence, it does not rely on the usual assumptions like uniformity and independence, which do not hold true for many real-life data sets. Moreover, it avoids the non-trivial task of a parametric method to model the data set. Lastly, but not leastly, sampling does not require any special evaluation algorithms for cardinality estimation. In fact, the same evaluation algorithm that produces the exact result for the whole data set can be used by sampling to estimate the cardinality. This implies that any research advances on the evaluation algorithms will directly benefit the sampling approach.

Despite the aforementioned advantages, applying sampling to skyline cardinality estimation is non-trivial. The difficulty lies in the non-linear relationship between the skyline cardinality of the sample and that of the whole data set. Log Sampling (LS) [9] and Kernel-Based (KB) skyline cardinality estimation [28] are the two state-of-the-art skyline cardinality estimation methods. LS is an overall parametric approach. Sampling is used only to derive the parameters. Like other parametric approaches, it relies on assumptions like independence. When such assumptions do not hold, its accuracy suffers. KB integrates the estimated probability density function (PDF) to derive the scale factor Ψ_{ds}. As the estimation of PDF and the ensuing integration both involve complex mathematical calculations, KB is complicated and slow to yield estimates.

In view of these problems, we propose an innovative and purely sampling-based skyline cardinality estimation approach (PS). PS carries all the advantages of a pure sampling approach. For instance, it is non-parametric, meaning it does not require any particular data distribution and is, thus, applicable to any data sets. In addition, PS is almost immune to the "curse of dimensionality," meaning it can handle high-dimensional data with ease. Furthermore, PS does not require any expensive mathematical calculations, so it is easy to implement and quick to yield estimates. Extensive empirical studies show that for a variety of real and synthetic data sets, PS outperforms LS in terms of estimation speed, estimation accuracy, and estimation variability using the same sample size. PS outperforms KB in terms of estimation speed and estimation variability under the same performance mark.

The rest of the paper is organized as follows. Section 2 reviews related research in skyline query cardinality estimation. Section 3 discusses how to apply a purely sampling-based approach (PS) to skyline cardinality estimation. Section 4 compares PS with LS and KB. Detailed theoretical analyses and experimental results are presented. Finally, Sect. 5 concludes this paper.

2 Related work

Assuming that for each point, the value on one dimension is distributed independently of the value on another dimension, and on each dimension, the values for each point in the data set are distinct, Bentley et al. [4] derived an asymptotic bound for skyline cardinality as $O((\ln(n))^{d-1})$, where n is the data set size and d the dimensionality. Godfrey et al. [13] relaxed the distinctive value restriction by assuming that the majority of the points have distinct values on any dimension, and the values of the points on any one dimension are uniformly distributed. They derived the cardinality formula as $(\ln(n) + \gamma)^d/d!$.

Chaudhuri et al. [9] relaxed the distinctive value restriction further to allow repeated values on any dimensions. They derived a formula that works with both numerical and categorical values. They also attempted to relax the independence assumption, so that other distributions can be accommodated. By empirical studies, they observed that when the data distribution does not have significant correlations/anti-correlations, the skyline cardinality follows the model: $A(\log(n))^B$, where A and B are two constants.

Log Sampling (LS) is based on this hypothetical model. LS draws a small sample to estimate the two parameters A and B, which are then used to estimate the skyline cardinality of the whole data set. More specifically, it takes the following steps:

1. Draw a small sample from the data set using simple random sampling without replacement.
2. Split the sample into two parts , s_1, s_2, of different sizes, that is, $|s_1| \neq |s_2|$.
3. Evaluate skyline queries on s_1 and s_2 to get their skyline cardinality, $|\text{Sky}_{s_1}|$ and $|\text{Sky}_{s_2}|$, respectively.
4. Solve A and B. According to the model: $|\text{Sky}_{s_1}| = A(\log(|s_1|))^B$ and $|\text{Sky}_{s_2}| = A(\log(|s_2|))^B$, Therefore, $B = \frac{\log(|\text{Sky}_{s_2}|)-\log(|\text{Sky}_{s_1}|)}{\log(\log(|s_2|))-\log(\log(|s_1|))}$ and $A = \frac{|\text{Sky}_{s_1}|}{(\log(|s_1|))^B}$.
5. Estimate the skyline cardinality of the whole data set. $|\text{Sky}_{ds}| = A(\log(n))^B$

The major drawbacks of LS lie in the hypothetical model it is based on. Firstly, the model is proposed according to observations, rather than rigorous mathematical deductions. Therefore, it may not be accurate. Secondly, many data sets do not follow this model. Notable examples include significantly correlated / anti-correlated data.

Kernel-Based skyline cardinality estimation (KB) [28] uses kernels to estimate the probability density function (PDF) at an arbitrary position in the data space from a random sample. Given a random sample s and an arbitrary position q, the PDF at q can be estimated by the formula:

$$\text{PDF}(q) = \sum_{s_i \in s} \frac{1}{|s| h^d \sqrt{\det\left(\sum\right)}} K\left(\frac{\text{dist}_{\sum}(q, s_i)}{h}\right) \tag{1}$$

where h is the kernel bandwidth, $\sum$ (a $d \times d$ matrix) is the kernel orientation, $\det\left(\sum\right)$ is the determinant of $\sum$, $\text{dist}_{\sum}(q, s_i)$ is the Mahalanobis distance between q and the sample point s_i, and K is the kernel function, which in practice is usually implemented as Gaussian Kernel [21], as defined in the equation:

$$K(x) = \frac{1}{\sqrt{2\pi}} \text{e}^{-\frac{x^2}{2}}$$

KB estimates the skyline cardinality generally as follows:

1. Draw a sample s from the data set ds using simple random sampling without replacement
2. Evaluate skyline query on the sample s to get local skyline points sky_s
3. For each local skyline point $p \in \text{sky}_s$, calculate the probability that a random point in the data set falls in its inverse dominance region (IDR). This probability, denoted as Ω_p, can be calculated by the integration of PDF over the whole region of IDR.

$$\Omega_p = \int_{\text{IDR}(p)} \text{PDF}(q) \text{d}q \tag{2}$$

4. Estimate the skyline cardinality of the data set using the formula:

$$|\text{Sky}_{ds}| = |ds| \times \frac{1}{|s|} \sum_{p \in \text{sky}_s} (1 - \Omega_p)^{|ds|-|s|}$$

Since KB uses complex mathematical calculations, for instance, Eqs. 1 and 2, it is comparatively complex and slow. The goal of query optimization is to quickly identify an efficient query plan. With a time-consuming estimation method, query optimization would lose much of its merit.

3 Applying sampling to skyline cardinality estimation

3.1 A naive sampling approach for skyline cardinality estimation

We can easily propose a naive sampling approach to estimate skyline cardinality. The steps are as follows:

1. Draw a small sample using simple random sampling without replacement.
2. Evaluate skyline query on the sample to find its skyline cardinality.
3. Scale the sample's skyline cardinality to estimate the data set's skyline cardinality.

Unfortunately, as noted by Chaudhuri et al. [9], this naive sampling approach yields erroneous estimates. According to them, "this does not work for the reason that the size of the Skyline is not a linear function of the data size".

For the sake of exposition, we define the symbols in Table 2.

Moreover, without loss of generality, we assume that smaller values are preferred.

The following example illustrates this unique feature of skyline query.

Table 2 Symbols

Symbol	Meaning
ds	The data set
n	The data set size
Sky_{ds}	The skyline point set of the data set
s	The sample
m	The sample size
sky_s	The skyline point set of the sample
$\lvert\text{Sky}_{ds}\rvert$	Cardinality of the data set's skyline
$\widetilde{\lvert\text{Sky}_{ds}\rvert}$	Cardinality estimate of the data set's skyline
$\lvert\text{sky}_s\rvert$	Cardinality of the sample's skyline

Table 3 Data set characteristics

Data set	Size	Dimensionality
Household	100,000	6
Corel	60,000	9
NBA	300,000	16
Spatial skyline	500,000	5
IND	300,000	6
COR	300,000	6
ANT	300,000	6

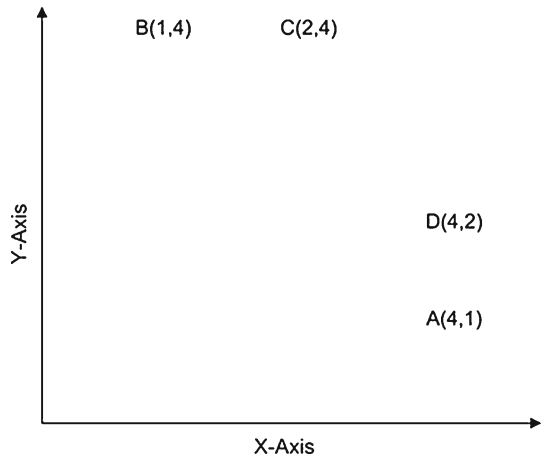


Fig. 1 Naive sampling

Considering a two-dimensional data set ds that has 4 points: $A(4, 1)$, $B(1, 4)$, $C(2, 4)$, and $D(4, 2)$, the points are plotted in Fig. 1. Clearly, the data set's skyline point set $\text{Sky}_{ds} = \{A(4, 1), B(1, 4)\}$. Assume we apply the naive sampling approach to derive $|\widetilde{\text{Sky}_{ds}}|$.

1. We sample two points from ds. Suppose the points drawn are $A(4, 1)$ and $C(2, 4)$, namely $s = \{A(4, 1), C(2, 4)\}$ and $m = 2$.
2. Derive the sample's skyline point set sky_s. Because neither point A nor C is dominated by another point, therefore, $\text{sky}_s = s = \{A(4, 1), C(2, 4)\}$ and $|\text{sky}_s| = 2$.
3. Estimate the cardinality of Sky_{ds}.

$$|\widetilde{\text{Sky}_{ds}}| = \frac{|\text{sky}_s|}{m} \times n = \frac{2}{2} \times 4 = 4$$

As the example demonstrates, point C is a sample's skyline point, but not a data set's skyline point. Scaling the cardinality of the sample's skyline point set will overestimate the data set's skyline cardinality.

3.2 A purely sampling-based approach for skyline cardinality estimation

The naive sampling approach produces erroneous estimates, because skyline points local to the sample are not necessarily a subset of the data set's global skyline point set. Skyline cardinality depends on not only data size but also data scope. Skyline cardinality of the local sample and that of the global data set belong to different scopes. It is technically incorrect to scale the former to estimate the latter.

Based on the above analyses, we realize that the key to devise a purely sampling-based approach is to break the scope boundary. That is to say, the skyline cardinality to be scaled must be in the same data scope as that to be estimated.

Let us define another symbol Sky_s to denote the global skyline point set contained in the sample. Please note that although looks similar, the symbol sky_s that we defined earlier has a totally different meaning. It denotes the local skyline point set of the sample.

We follow three steps to estimate the skyline cardinality:

1. Draw a sample s of size m using simple random sampling without replacement.
2. Determine the global skyline point set contained in the sample Sky_s.

3. Scale $|\text{Sky}_s|$ to estimate the skyline cardinality Sky_{ds}.

$$\widetilde{|\text{Sky}_{ds}|} = \frac{|\text{Sky}_s|}{m} \times n$$

Theorem 1 $\widetilde{|\text{Sky}_{ds}|}$ *is an unbiased estimator of the global skyline point cardinality.*

Proof (sketch). Since the points are sampled randomly, each point is equally likely to be selected. Therefore, when we sample m points, the expected number of global skyline points contained in the sample should be $\frac{|\text{Sky}_{ds}|}{n} \times m$, that is: $E(|\text{Sky}_s|) = \frac{|\text{Sky}_{ds}|}{n} \times m$. Hence, $|\text{Sky}_{ds}| = \frac{E(|\text{Sky}_s|)}{m} \times n$. Because $\widetilde{|\text{Sky}_{ds}|} = \frac{|\text{Sky}_s|}{m} \times n$, therefore $E(\widetilde{|\text{Sky}_{ds}|}) = \frac{E(|\text{Sky}_s|)}{m} \times n = |\text{Sky}_{ds}|$. □

Example Considering a two-dimensional data set ds that has 4 points: $A(4, 1)$, $B(1, 4)$, $C(2, 4)$, and $D(4, 2)$, the points are plotted in Fig. 1. Clearly, the data set's skyline point set $\text{Sky}_{ds} = \{A(4, 1), B(1, 4)\}$.

1. We sample two points from ds. Suppose the points drawn are $A(4, 1)$ and $C(2, 4)$, namely $s = \{A(4, 1), C(2, 4)\}$ and $m = 2$.
2. Derive the set of global skyline points contained in the sample Sky_s. Because point C is dominated by another point B, therefore, it is not a global skyline point. $\text{Sky}_s = \{A(4, 1)\}$ and $|\text{Sky}_s| = 1$.
3. Estimate the cardinality of Sky_{ds}.

$$\widetilde{|\text{Sky}_{ds}|} = \frac{|\text{Sky}_s|}{m} \times n = \frac{1}{2} \times 4 = 2$$

It is straightforward to implement steps 1 and 3; However, step 2 requires further investigation.

3.2.1 Determining the global skyline points in the sample Sky_s

To determine the global skyline points in the sample s, we can simply compare each point $p \in s$ with the rest of the $n - 1$ points in the data set ds. If p is dominated by at least one of the $n - 1$ points, then p is not a global skyline point. If none of the $n - 1$ points dominates p, then p is a global skyline point.

However, this simple approach, let's call it Sky_s1, is time consuming. Firstly, the comparison itself takes time. In addition, Sky_s1 needs to access the whole data set. This implies that very likely it has to access the secondary storage device (e.g., hard disk), because a large data set cannot fit in the memory, thus further slow down the estimation. The whole point of query optimization is to quickly identify an efficient query plan. If the estimation time itself is comparable to the query execution time, we lose the benefit of using query optimization at all.

Alternatively, Sky_s can be determined by performing the following steps.

1. Determine the local skyline point set of the sample sky_s.
2. Determine the local skyline point set of the remaining data set $ds - s$. Let us call it sky_{ds-s}.
3. For each point p in sky_s, check whether it is dominated by at least one point in sky_{ds-s}. p is a global skyline point if no points in sky_{ds-s} dominate it.

The correctness of this approach is based on the following two theorems.

Theorem 2 *The local skyline point set of the sample* sky_s *is a superset of the global skyline points contained in the sample* Sky_s. *In other words,* $\text{sky}_s \supset \text{Sky}_s$.

Proof Theorem 2 can be proved by contradiction. Suppose there exists a point p in the sample s, such that $p \in \text{Sky}_s$ but $p \notin \text{sky}_s$. $p \in \text{Sky}_s$ means p is not dominated by any point in the entire data set ds, while $p \notin \text{sky}_s$ means p is dominated by at least one point in the sample s of the data set. This is a contradiction. Therefore, p does not exist. In other words, if a point in the sample is a global skyline point, it must also be a local skyline point of the sample. □

Theorem 3 *Given a point* p *in* sky_s. *If no points in* sky_{ds-s} *dominate it, then it is a global skyline point.*

Proof Theorem 3 can also be proved by contradiction. Suppose there exists a point $p \in \text{sky}_s$ that is not dominated by any points in Sky_{ds-s}, but it is not a global skyline point. p is not a global skyline point implies that it is dominated by at least one point in the data set. The point that dominates p, let us call it p_d, must come from one of the two sources, the sample or the rest of the data set. Case 1: p_d comes from the sample s. p_d coming from s means p is dominated by another point in the sample, therefore p can not be a local skyline point. This contradicts the condition that $p \in \text{sky}_s$. Case 2: p_d comes from the rest of the data set. p_d coming from $ds - s$ means p is dominated by a point in $ds - s$. If p_d is in sky_{ds-s}, this contradicts the given condition that no point in sky_{ds-s} dominates p. If p_d is not in sky_{ds-s}, there must exist another point in sky_{ds-s} that dominated p_d and in turn also dominates p. This also contradicts the condition that no point in sky_{ds-s} dominated p. Combining these cases, the assumption cannot be true. □

Example Considering a two-dimensional data set ds that has 4 points: $A(4, 1)$, $B(1, 4)$, $C(2, 4)$, and $D(4, 2)$, the points are plotted in Fig. 1. Clearly, the data set's skyline point set $\text{Sky}_{ds} = \{A(4, 1), B(1, 4)\}$.

1. We sample two points from ds. Suppose the points drawn are $A(4, 1)$ and $C(2, 4)$, namely $s = \{A(4, 1), C(2, 4)\}$ and $m = 2$. Since A and C do not dominate each other, $\text{sky}_s = s = \{A(4, 1), C(2, 4)\}$.
2. The rest of the points are in $ds - s$. $ds - s = \{B(1, 4), D(4, 2)\}$. Similarly, $\text{sky}_{ds-s} = ds - s = \{B(1, 4), D(4, 2)\}$.
3. Comparing each point in sky_s with the points in sky_{ds-s}, it is found that point C is dominated by point B. Therefore, the global skyline points in the sample $\text{Sky}_s = \{A(4, 1)\}$.

Although technically correct, this approach, let us call it Sky_s2, is still not efficient. However, Sky_s2 does have one advantage. It divides the data set into two independent parts, s and $ds - s$, which opens the door to devise a more efficient algorithm. More specifically, step 1 focuses exclusively on the sample s. Since s is usually small, its associated time complexity is acceptable. Step 2 focuses exclusively on the remaining data set $ds - s$. It contributes the lion's share to the overall time complexity. The structure of Sky_s2 makes it possible to improve step 2, while leaving the other two steps intact.

As we are *estimating* skyline cardinality, we do not need to derive the actual set of sky_{ds-s}. It is sufficient if we can find an alternative set that is quick to derive, small to store, and approximates sky_{ds-s} well. sky_{ds-s} contains the local skyline points of $ds - s$. These points are not dominated by any other points in $ds - s$. We argue that within $ds - s$, a set of points that have a good chance to dominate others can serve as a good approximation of sky_{ds-s}. The rationale is that because the set $ds - s$ is finite, given a point $p \in ds - s$, p has a good

chance to dominate any other points, which means p has a good chance to be dominated by no other points. Therefore, p is very likely a local skyline point.

According to the definition of point dominance, given a d-dimensional data set ds, a point $p1 \in ds$ dominates another point $p2 \in ds$ **iff** $\exists i$, where $1 \leq i \leq d$, such that $p1_i < p2_i$ and $\forall j$ where $1 \leq j \leq d$ and $j \neq i$, such that $p1_j \leq p2_j$. If we sum up the d inequalities, we have $\sum_{m=1}^{d} p1_m < \sum_{m=1}^{d} p2_m$. In other words, if $p1$ dominates $p2$, then the sum of $p1$'s attribute values is smaller than the sum of $p2$'s attribute values. Therefore, to approximate sky_{ds-s}, we can use a set of points whose sums of attribute values are the top k smallest. The value of k depends on the space budget and performance considerations. Section 4 will discuss this issue in detail.

It should be pointed out that the attribute value on each dimension is expected to play an equally important role in deciding point dominance. It is very likely that the domain of each dimension is different, which will cause a larger domain to have a larger weight in deciding point dominance. To prevent this problem, we need to normalize each dimension before summing up the attribute values. We opt to normalize the domain of each dimension to a pre-determined domain [0, 1]. Let X be the attribute of a given dimension, $maxX$ and $minX$ the maximum and minimum value of X, respectively. Then, each value $x \in X$ is normalized as follows:

$$x^z = \frac{x - \min X}{\max X - \min X}$$

where x^z denotes the normalized value of x. For example, the set of x values, $\{0,\ 1,\ 2,\ 3,\ 4\}$, is normalized to $\{0,\ 1/4,\ 1/2,\ 3/4,\ 1\}$.

4 Experimental results

This section presents the experimental results of our purely sampling-based method (PS) on both synthetic and real-life data sets. We compare its performance with that of LS [9] and KB [28].

4.1 Experimental settings

Techniques All the three methods were implemented in Visual C++ 6.0. Zhang [28] kindly provided the source codes of LS and KB. He also provided the real-life data sets. The experiments were conducted on a dedicated computer with a 2.4 GHz CPU and 768 MB RAM.

Data sets Four real-life data sets, Household, Corel, NBA, and Spatial skyline were selected for the experiments. Household is available at http://www.ipums.org. Each tuple has 6 attributes that record the percentage each American family spend on gas, electricity, water, heating, insurance, and property tax. Corel is available at kdd.ics.uci.edu. Each tuple has 9 attributes that record the mean, standard deviation, and skewness of the H, S, and V values of an image pixel. NBA is available at http://www.basketballreference.com. Each tuple has 16 attributes that record each player's statistics such as number of games played, minutes played, etc. Spatial skyline is available at http://www.rtreeportal.org. Each tuple records the 2D co-ordinates of a place in Los Angles. Zhang et al. [28] used a new version of this database. Each tuple in the new version has 5 attributes, which were randomly chosen from the original data set.

In addition, three synthetic data sets, which have independent (IND), correlated (COR), and anti-correlated (ANT) attributes, respectively, were tested in the experiments. These syn-

thetic data sets are generated by the data set generator based on [6]. The generator is available at pgfoundry.org. Each tuple in these synthetic data sets has 6 attributes.

Table 3 summarizes the major characteristics of the tested data sets.

Parameter configurations The LS method splits a random sample s into two parts s_1 and s_2. There are no guidelines regarding the sizes of $s1$ and $s2$. However, since the formula that estimates B is a fraction,

$$B = \frac{\log(|\text{Sky}_{s_2}|) - \log(|\text{Sky}_{s_1}|)}{\log(\log(|s_2|)) - \log(\log(|s_1|))}$$

the denominator cannot be zero. Therefore, the sizes of s_1 and s_2 cannot be the same. Besides this restriction, the choices of $|s_1|$ and $|s_2|$ also affect the time complexity of LS, although not significantly. Let the space budget be b. In our experiments, we set $|s_1| = 2|s_2|$ and $|s_1| + |s_2| = b$.

KB also draws two samples s_1 and s_2 to calculate the local skyline points and estimate the PDF of the data set. Zhang's implementation has $|s_2| = 2|s_1|$ and $|s_1| + |s_2| = b$.

For PS, we have one random sample s_1 from the data set ds, k points that have the top k smallest sums of attribute values in $ds - s_1$, and another random sample s_2 from $ds - s_1$. The top k points and s_2 together approximate the local skyline points in $ds - s_1$. In our experiments, we set $|s_1| = 2|s_2|$, $k = |s_2|$, and $|s_1| + k + |s_2| = b$.

Error metric The error metric we used is the absolute value of the relative error, which is defined as $RE = 100 * |t - e|/t$, where t and e are the true and estimated skyline cardinalities, respectively.

Reported values The reported estimation speed and estimation accuracy were the averages and medians of every 10 runs, respectively. The reported standard deviations were derived from every 10 runs.

4.2 Comparisons between LS and PS

We first show the comparisons between LS and PS. Since these two methods have comparable estimation speed, we compare them with the same space budget. More specifically, each data set is tested under three space budgets, namely 5, 10, and 15% of the data set size. Under each space budget, the two methods were compared in estimation speed, estimation accuracy, and estimation variability.

4.2.1 Estimation speed

For both LS and PS, local skyline query evaluations and checking point dominance take the lion's share of the total cost. For simplicity's sake, we assume a straightforward algorithm is used to determine whether a point p is dominated by any point in a given data set ds. The algorithm is implemented as a function $checkDominance()$, whose pseudo-code is shown in Fig. 2. The function returns true if p is not dominated by any point in ds, and false otherwise.

For LS, finding the local skyline points for s_1 has time complexity $O\left(\frac{b^2}{9}d\right)$, where b is the space budget and d the dimensionality. Finding the local skyline points for s_2 takes $O\left(\frac{4b^2}{9}d\right)$. Adding them up, the time complexity of LS is roughly $O\left(\frac{5b^2}{9}d\right)$.

For PS, finding the local skyline points for s_1 takes $O\left(\frac{b^2}{4}d\right)$. Checking if each point in the local skyline point set of s_1 is dominated by any points in the top k points or s_2 takes $O\left(\frac{b^2}{4}d\right)$. It should be noted that since the top-k points can be pre-computed, the processing

```
checkDominance(p, ds, d, n)
Input: point p, dataset ds, dimensionality d, and dataset size n
begin
    for (int i=0;i<n;i++) do
        if (ds[i]!=p) then
            int lt=0, eq=0;
            for (int j=0;j<d;j++) do
                if (ds[i][j]<p[j]) then
                    lt++;
                end
                else if (ds[i][j]==p[j]) then
                    eq++;
                end
            end
            if (lt+eq==d && lt>0) then
                return false;
            end
        end
    end
    return true;
end
```

Fig. 2 Algorithm checkDominance

time is not taken into consideration. Adding them up, the time complexity of PS is roughly $O\left(\frac{b^2}{2}d\right)$. Therefore, PS is in general slightly faster than LS.

Table 4 shows the estimation speed on the chosen data sets. The estimation speed is measured in seconds. The experimental data show that the estimation time for both LS and PS is only a small fraction of the time to actually calculate the skyline cardinality. A case in point is the Corel data set; even with the largest 15% space budget, the estimation time is only about 3% of the time to calculate the actual skyline cardinality. In fact, in most cases, the 5% space budget is enough to derive a good estimate, as shown in the next section estimation accuracy. This clearly shows the benefits of efficient skyline cardinality estimation methods.

In addition, Table 4 shows that the estimation speed is not only associated with the data set size and dimensionality but It also largely depends on the data set distribution. For instance, although the data set COR has the same size and dimensionality as IND and ANT, it is much faster to execute a skyline query on COR.

Finally, the experimental data confirm that given the same sample size, PS is slightly faster than LS in most cases.

4.2.2 Estimation accuracy

We show both the cardinality estimates and estimation errors to report estimation accuracy. The error metric we used is the absolute value of the relative error, which is defined as $\text{RE} = 100 * |t - e|/t$, where t and e are the true and estimated skyline cardinalities, respectively. Clearly, the true skyline cardinality, which resides in the denominator, greatly affects the estimator error. For instance, a small true skyline cardinality will magnify the difference between the true and estimated skyline cardinalities. Therefore, the original cardinality estimates are also presented.

Real-life data sets A real-life data set usually consists of data that have intricate correlations. It is difficult to accurately capture its distribution. Consequently, using a simplified data model $A(\log(n))^B$ to summarize a real-life data set will generally result in large estimation errors. In contrast, PS is a non-parametric method. It does not rely on any assump-

Table 4 Estimation speed

Datasets	Techniques	5%	10%	15%
Household	LS	0.7	3	6.1
	PS	0.6	2.3	4.8
	Actual	85		
Corel	LS	1.2	6.9	16.4
	PS	1	6	14.6
	Actual	556		
NBA	LS	16.4	54.8	109.5
	PS	11.9	36.9	78.3
	Actual	2,786		
Spatial	LS	63.1	250.4	560.2
	PS	227.9	890.4	2,058.1
	Actual	10,228		
IND	LS	9.1	27.6	50.7
	PS	6.2	19.4	36.4
	Actual	392		
COR	LS	0.4	1.2	1.8
	PS	0.9	5.8	5.3
	Actual	9		
ANT	LS	37.6	134.5	277.9
	PS	28.9	98	199
	Actual	3,436		

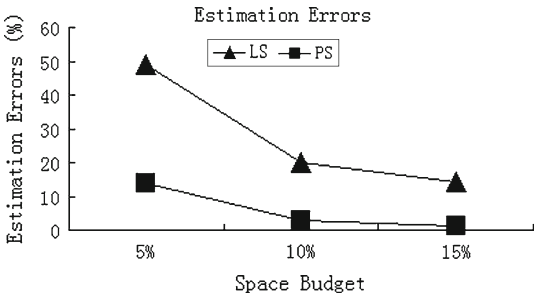


Fig. 3 Estimation errors of household

tions of data distribution and is, therefore, applicable to any data sets. Figures 3, 4, 5, and 6 show that PS outperforms LS in terms of estimation accuracy by wide margins. Particularly, almost all of the estimation errors of PS are below 10%, even under the 5% space budget.

Interestingly, Figs. 7, 8, 9, and 10 show that LS tend to underestimate skyline cardinality, while PS tend to overestimate skyline cardinality.

Synthetic datasets Figures 11 and 12 show that both LS and PS perform well on IND. Under 10% space budget and above, the estimation errors of both methods are well below 10%. Datasets with independent distributions fit well in the $A(\log(n))^B$ model, thus the estimates of LS are quite accurate.

Figures 13 and 14 show that the true skyline cardinality of ANT is large, while that of COR is small. This is because we prefer small values on all dimensions. For COR, a point

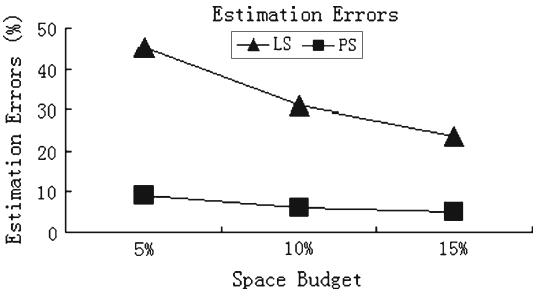


Fig. 4 Estimation errors of corel

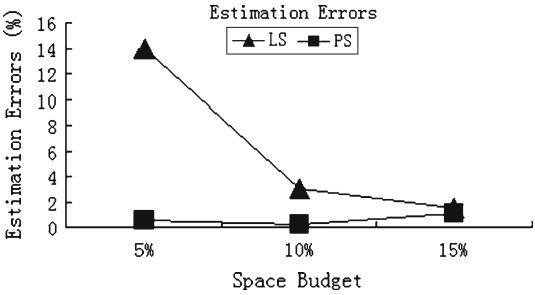


Fig. 5 Estimation errors of NBA

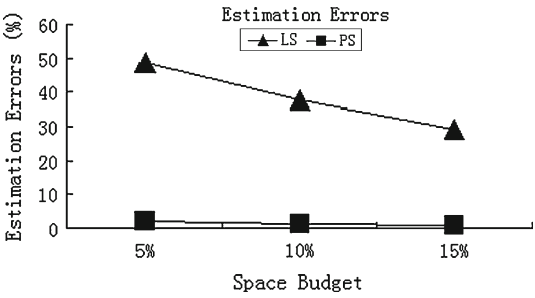


Fig. 6 Estimation errors of spatial skyline

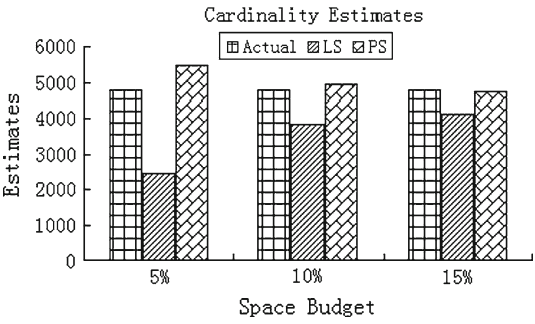


Fig. 7 Estimates of household

that has a small value on a dimension i where $1 \leq i \leq d$ tends to also have small values on other dimensions. Therefore, COR has a small set of skyline points. For similar reasons, ANT has a large set of skyline points.

Figures 15 and 16 show that the estimation errors of PS on COR and ANT are comparable. Although PS performs slightly better on ANT than on COR, the performance edge is not

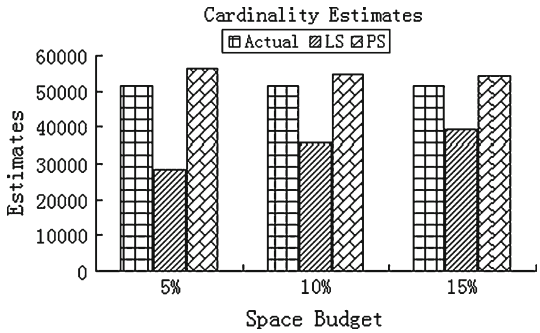


Fig. 8 Estimates of corel

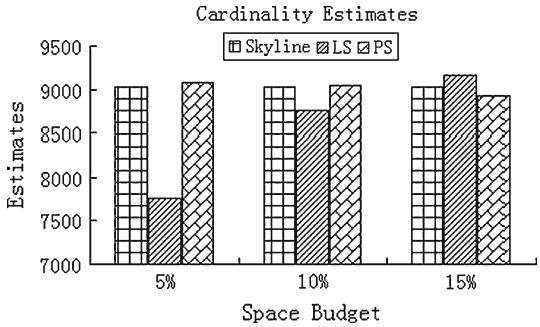


Fig. 9 Estimates of NBA

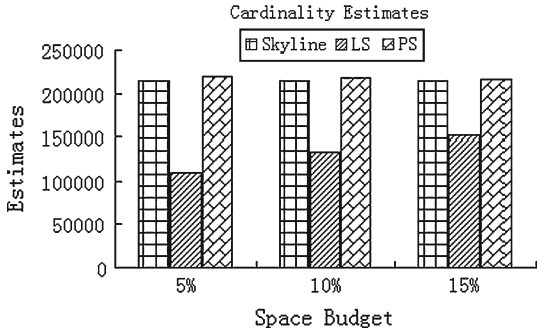


Fig. 10 Estimates of spatial skyline

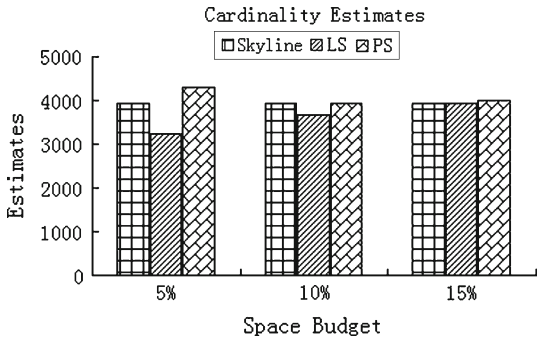


Fig. 11 Estimates of IND

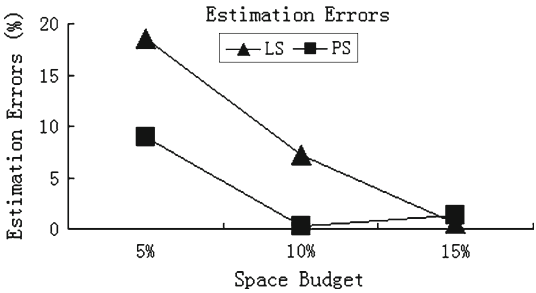


Fig. 12 Estimation errors of IND

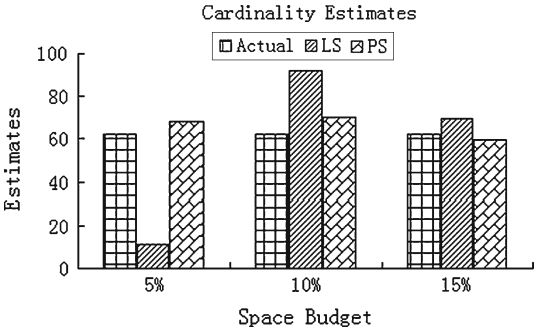


Fig. 13 Estimates of COR

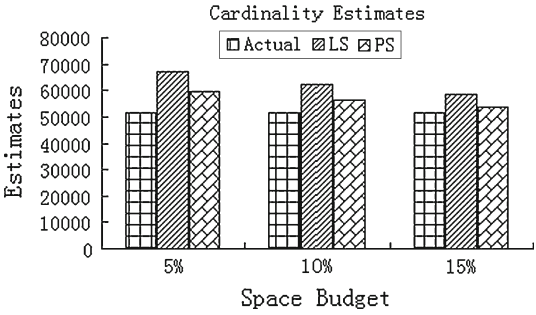


Fig. 14 Estimates of ANT

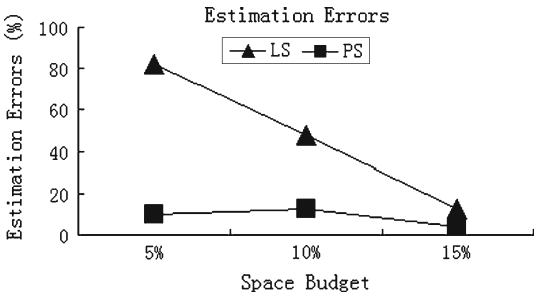


Fig. 15 Estimation errors of COR

significant. As far as LS is concerned, it performs much better on ANT, which suggests $A(\log(n))^B$ models ANT much better than it models COR.

To summarize, PS outperforms LS for all the three synthetic data sets. When the space budget is around 10%, the estimation errors of PS are under 10%.

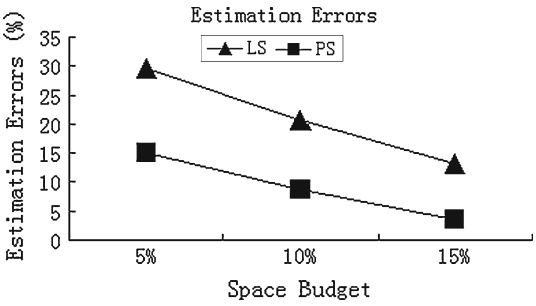


Fig. 16 Estimation errors of ANT

Table 5 Standard deviation of the estimation

Datasets	Techniques	5%	10%	15%
Household	LS	1,418.0	871.7	510.4
	PS	252.4	206.3	190.5
Corel	LS	619.48	594.0	368.8
	PS	453.9	276.6	201.2
NBA	LS	5,879.9	5,759.0	3,035.9
	PS	508.1	405.0	294.1
Spatial	LS	4,866.1	8,505.1	5,175.4
	PS	1,337.8	1,321.0	626.1
IND	LS	1,251.3	834.2	699.8
	PS	382.1	259.2	197.5
COR	LS	424.4	313.2	271.3
	PS	40.2	32.8	28.3
ANT	LS	4,122.5	2,913.7	2,715.4
	PS	1,420.7	917.9	826.7

4.2.3 Estimation variability

Table 5 shows the standard deviations of the estimations for the chosen data sets. The values were derived from 10 runs. Generally speaking, the standard deviation for both methods are small in comparison with the actual skyline cardinality. Still, PS has smaller standard deviations than LS. LS's standard deviations are on average 3 times as large as those of PS. This demonstrates that PS is a more robust method.

4.3 Comparisons between KB and PS

AS preliminary experiments show that given the same sample size, KB is significantly slower than PS, but KB can yield estimates with good accuracy using small sample size, we decided not to compare the performance of KB and PS based on the same sample size. Instead, we compared these two methods based on the same desirable estimation accuracy. More specifically, for each data set, we require the estimation error to be 15% or less. Under this condition, we compared the sample size, estimation speed, and estimation variability of the two methods.

The experiments were conducted by gradually increasing the sample size in each iteration. When the estimation method achieved 85% or better estimation accuracy for three

Table 6 Comparisons between KB and PS

Data set	Technique	Sample size	Estimation speed	Standard deviation
Household	KB	600	1.7	1,378.6
	PS	2,200	0.1	1,126.7
Corel	KB	300	1.4	1,971.9
	PS	460	0	110.0
NBA	KB	2,400	58.9	1,094.0
	PS	2,000	0.3	619.7
Spatial	KB	390	1.9	27,616.2
	PS	350	0	3,532.1
IND	KB	300	0.6	1,760.4
	PS	1,750	0.1	681.9
COR	KB	62,700	70.9	47.6
	PS	25,000	3.3	31.7
ANT	KB	2,000	65.8	4,931.6
	PS	16,000	30.5	1,361.0

consecutive iterations, the experiment stopped and the experimental data for the first of the three consecutive iterations were reported. Again, each iteration consisted of 10 runs.

As Table 6 shows, generally, both KB and PS are able to yield accurate estimates using small samples. Except for the COR data set, both methods achieved 85% or better estimation accuracy using less than 5% of the original data. In the majority of the cases, KB used smaller samples than PS to reach the 85% performance mark. However, for the NBA and COR data sets, KB had to use larger samples. The NBA data set has the largest dimensionality (16). Since KB needs to integrate the PDF over the IDRs, the higher the dimensions, the more samples are needed to achieve a desirable estimation accuracy. This is a classical form of "curse of dimensionality" [3]. The COR data set has a very low ratio of skyline points. Out of the total 300, 000 points, there are only 62 skyline points. It requires relatively large samples to yield good estimates.

As far as estimation speed is concerned, KB was shown to be a much slower method than PS. Under the same 85% performance mark, PS took much less time, sometimes several orders of magnitude less time, than KB to yield the estimates for all the data sets. Considering that KB generally used smaller samples, the estimation speed gap between these two methods can be even wider.

In the case of estimation variability, PS's standard deviations of the estimates were smaller than KB's for all the data sets. In some cases, such as the *Corel* and *Spatial* data sets, the differences were substantial. Although increasing the sample size can help reduce the standard deviation, doing so will further slow down the estimation.

Under the same performance mark, in comparison with KB, PS was able to yield estimates with much higher speed and smaller variability. Although generally it required larger samples, we argue PS is still a better estimation method for the following reasons. Firstly, the sample sizes of PS were well below 5% of the original data set sizes in most cases. A modern computer can easily provide enough memory to store theses samples. Secondly, even with relatively larger samples size, PS outperformed KB in terms of estimation speed by wide margins. Thirdly, PS is more robust because it has much smaller estimation variability than KB. Finally, KB does not work well with high-dimensional data because of "curse of dimen-

sionality". It is not an effective method for data sets with small ratios of skyline points may be because it requires larger sample sizes, which results in even slower estimation speed. In contrast, PS is a clear winner in these two cases.

4.4 Summary

The experimental results show that using the same sample size, PS outperformed LS in terms of estimation speed, estimation accuracy, and estimation variability. Under the same performance mark, namely 85%, PS generally required larger sample size than KB. However, the sample size for KB was still only a small fraction of the original data set size. In most cases, the sampling ratio was well below 5%. Furthermore, PS outperformed KB in terms of estimation speed and estimation variability by wide margins. Lastly, PS was a clear winner for high-dimensional data and data sets with small skyline point ratios.

5 Conclusions

Generally, a real-life data set contains intricate correlations among attribute values as well as points. Given this complexity, it is difficult, if not impossible, to model the skyline cardinality of an arbitrary data set. The state-of-the-art skyline cardinality estimation method Log Sampling (LS) is based on a hypothetical model $A(\log(n))^B$. This model is derived under strong assumptions like data independence between dimensions. Since a data set usually does not satisfy these assumptions, the estimation accuracy of LS suffers.

Kernel-Based (KB) skyline cardinality estimation is another state-of-the-art method. KB involves complex mathematical calculations to estimate the PDF of the data set and the scaling factor. It is slow to yield estimates. Because KB needs to integrate PDF over IDRs, which are essentially spatial regions, it suffers from "curse of dimensionality". KB has to use comparatively large samples for data sets with small skyline point ratios, which further slows down it's estimation speed.

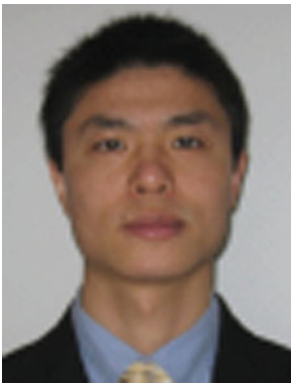
We propose a non-parametric purely sampling-based (PS) method for skyline cardinality estimation. PS does not assume any particular data distribution. It is, thus, more robust and applicable to any data sets. Extensive experiments confirm that PS outperforms LS in estimation speed, estimation accuracy, and estimation variability using the same sample size. PS does not require complex mathematical calculations. It is, thus, significantly faster and simpler than KB. Experimental results demonstrate that given the same performance mark, although the sample size of PS was generally slightly larger than that of KB, its overall sampling ratio was still small—well below 5% in most cases. On the other hand, PS outperformed KB in terms of estimation speed and estimation variability by wide margins. In addition, PS is a clear winner for high-dimensional data sets and data sets with small skyline point ratios.

References

1. Bartolini I, Ciaccia P, Patella M (2008) Efficient sort-based skyline evaluation. ACM Trans Database Syst 33(4):1–49
2. Bartolini I, Ciaccia P, Patella M (2010) Query processing issues in region-based image databases. Knowl Inf Syst 25(2):389–420
3. Bellman R (1961) Adaptive control processes: a guided tour. Princeton University Press, Princeton
4. Bentley J, Kung H, Schkolnick M, Thompson C (1978) On the average number of maxima in a set of vectors and applications. J ACM 25(4):536–543

5. Bentley J, Clarkson K, Levine D (1990) Fast linear expected-time alogorithms for computing maxima and convex hulls. In: SODA '90: proceedings of the first annual ACM-SIAM symposium on discrete algorithms. pp 179–187
6. Börzsönyi S, Kossmann D, Stocker K (2001) The skyline operator. In: Proceedings of the 17th international conference on data engineering. pp 421–430
7. Briggs W, Henson V (1995) DFT: an owner's manual for the discrete Fourier transform. Society for industrial and applied Mathematics Published, Philadelphia
8. Chaudhuri S, Motwani R, Narasayya V (1999) On random sampling over joins. In: Proceedings of ACM SIGMOD conference. pp 263–274
9. Chaudhuri S, Dalvi N, Kaushik R (2006) Robust cardinality and cost estimation for skyline operator. In: ICDE '06: proceedings of the 22nd international conference on data engineering. p 64
10. Chomicki J, Godfrey P, Gryz J, Liang D (2003) Skyline with presorting. In: Proceedings of ICDE 2003. pp 717–816
11. Ganguly S, Gibbons P, Matias Y, Silberschatz A (1996) Bifocal sampling for skew-resistant join size estimation. In: Proceedings of the 1996 ACM SIGMOD international conference on management of data. pp 271–281
12. Gilbert A, Kotidis Y, Muthukrishnan S, Strauss M (2001) Surfing wavelets on streams: one-pass summaries for approximate aggregate queries. In: Proceedings of the 27th international conferrence on VLDB. pp 79–88
13. Godfrey P, Shipley R, Gryz J (2005) Maximal vector computation in large data sets. In: Proceedings of VLDB. pp 229–240
14. Godfrey P, Shipley R, Gryz J (2007) Algorithms and analyses for maximal vector computation. VLDB J 16(1):5–28
15. Gunopulos D, Kollios G, Tsotras J, Domeniconi C (2005) Selectivity estimators for multidimensional range queries over real attributes. VLDB J 14(2):137–154
16. Hass P, Naughton J, Seshadri S, Swami A (1993) Fixed-precision estimation of join selectivity. In: Proceedings of 12th ACM symposium on principles of database systems. pp 190–201
17. Hass P, Naughton J, Seshadri S, Stokes L (1995) Sampling-based estimation of the number of distinct values of an attribute. In: Proceedings of 21st international conference on very large data bases. pp 311–322
18. Hou W-C, Ozsoyoglu G, Taneja, BK (1988) Statistical estimators for relational algebra expression. In: Proceedings of 7th ACM symposium on principles of database systems. pp 276–287
19. Hou W-C, Ozsoyoglu G, Taneja, BK (1989) Processing aggregate relational queries with hard time constraints. In: Proceedings of ACM SIGMOD international conference on management of data. pp 68–77
20. Huang Z, Sun S, Wang W (2010) Efficient mining of skyline objects in subspaces over data streams. Knowl Inf Syst 22(2):159–183
21. Hwang J-N, Lippman S-R (1994) A nonparametric multivariate density estimation: a comparative study. IEEE Trans Signal Process 42(10):2795–2810
22. Kung H, Luccio F, Preparata F (1975) On finding the maxima of a set of vectors. J. ACM 22(4): 469–476
23. Lee K, Zheng B, Li H, Lee, W (2007) Approaching the skyline in Z order. In: VLDB '07: proceedings of the 33rd international conference on very large data bases. pp 279–290
24. Lipton R, Naughton J, Schneider D (1990) Practical selectivity estimation through adaptive sampling. In: Proceedings 1990 ACM SIGMOD international conference managment of data. pp 1–11
25. Matias Y, Vitter J, Wang M (1998) Wavelet-based histograms for selectivity estimation. In: Proceedings of SIGMOD
26. Poosala V, Ioannidis Y (1997) Selectivity estimation without the attribute value independence assumption. In: VLDB '97: proceedings of the 23rd international conference on very large data bases. pp 486–495
27. Sun S, Huang Z, Zhong H, Dai D, Liu H (2010) Efficient monitoring of skyline queries over distributed data streams. Knowl Inf Syst 25(3):575–606
28. Zhang Z, Yang Y, Cai R, Papadias D, Tung A (2009) Kernel-based skyline cardinality estimation. In: SIGMOD '09: proceedings of the 35th SIGMOD international conference on Management of data. pp 509–522

Author Biographies



Cheng Luo received his PhD degree in Computer and Electrical Engineering from Southern Illinois University Carbondale in 2007. He is currently an Assistant Professor of Computer Science in the Department of Mathematics and Computer Science at Coppin State University, Baltimore, MD, USA. His interests are in query optimization, XML, and data mining.

Zhewei Jiang received her PhD degree in Computer Science from Southern Illinois University Carbondale in 2008. Presently, she works at Frederick Community College, Frederick, MD, USA, as an IT analyst. Her research interests lie in XML query evaluation and optimization, multi-dimensional data searches, and data mining.

Wen-Chi Hou received his MS and PhD degrees in computer science and engineering from Case Western Reserve University, Cleveland Ohio, in 1985 and 1989, respectively. He is presently a Professor of Computer Science at Southern Illinois University at Carbondale. His interests include statistical databases, mobile databases, XML databases, and data streams.

Shan He received his first MS degree in management engineering from Southwest Jiaotong University, China, in 1996 and second MS degree in electronic business management from the University of Warwick, UK, in 2003. At present he is an Associate Professor of Management Science at Southwest Petroleum University, China. His interests include data mining and visualization, Web data management, and multi-criteria decision making.

Qiang Zhu received his PhD degree in computer science at the University of Waterloo, Canada, in 1995. He is currently a Professor of Computer and Information Science at the University of Michigan, Dearborn, MI, USA. He is also an IBM CAS Faculty Fellow at the IBM Toronto Laboratory. His research interests include database query optimization, data streams, multi-dimensional indexing, self-managing database systems, data mining, and Web data management.