Cardinality Estimation of Skyline Queries: Harmonics in Data

Parke Godfrey

Technical Report CS-2002-03

October 2002

Department of Computer Science

4700 Keele Street North York, Ontario M3J 1P3 Canada

Cardinality Estimation of Skyline Queries

Parke Godfrey*

York University
4700 Keele Street
Toronto, ON M3J 1P3
CANADA
godfrey@cs.yorku.ca

College of William and Mary
P.O. Box 8795
Williamsburg, VA 23187-8795
U.S.A.
godfrey@cs.wm.edu

Abstract

There is interest to support queries with preferences in relational systems. In accordance, the skyline operator has been proposed as an extension to SQL. The corresponding skyline clause extends on the aggregate operators of max and min. It essentially selects the tuples that are mutually optimal over a number of attributes (by filtering out all tuples with worse values on every skyline attribute compared with the *skyline* tuples).

Recent work has focused on how to compute efficiently skyline queries in relational systems over large datasets. Steps remain, however, before the skyline operator can be incorporated into today's relational systems. A better understanding is needed on how the skyline operator composes with the other relational operators, both logically and algorithmically. The query optimizer must be able to incorporate the skyline operator. This will require extending the cost model for skyline queries. A critical component of the cost model must be a cardinality estimator for skyline queries' result sets.

Skyline cardinality estimation is the focus of this work. Under a basic model of assumptions of sparseness of values on attributes' domains (that is, virtually no duplicate values over an attribute) and statistical independence across attributes, we prove the expected skyline cardinality for two-dimensional skyline queries is the harmonic of the number of input tuples, and we generalize to prove the expected value for multi-dimensional skyline queries is a higher-order harmonic—a particular two parameter extension of the harmonic numbers—with respect to the number dimensions (skyline attributes) and the number of input tuples. We then consider the ramifications on the estimates as we relax the assumptions of the basic model, some of which are counter-intuitive. Our results provide a basis for a cost model for the skyline operator, and general insight into queries over multi-dimensional criteria.

1 Introduction

1.1 About Skyline Queries

Sometimes one wants to query relational data to find a best match. The aggregation operators min and max allow one to retrieve the best—that is, either lowest or highest—tuples with respect to one criterion. The order by clause in SQL allows one to rank order the results, perhaps with respect to many criteria. (The rank ordering will be equivalent to a nested sort over the indicated attributes' values.) Beyond this, relational query languages as SQL provide little else for finding best matches, or for expressing preferences as part of one's queries.

As an example, consider a table of restaurant guide information, as in Figure 1(a). Column **S** stands for *service*, **F** for *food*, and **D** for *decor*. Each is scored from 1 to 30, with 30 as the best. This table is modeled on the Zagat Survey Guides (for example, see [19]). We are interested to choose a restaurant from the guide. We are looking for a best choice, or a set of best choices from which to choose. Ideally, we would like the restaurant chosen to be the best for service, food, *and* decor, *and* be the lowest priced. There is likely no

*Parke Godfrey is presently on leave-of-absence from York University at The College of William and Mary.

restaurant	S	F	D	price
Summer Moon	21	25	19	47.50
Zakopane	24	20	21	56.00
Brearton Grill	15	18	20	62.00
Yamanote	22	22	17	51.50
Fenton & Pickle	16	14	10	17.50
Briar Patch BBQ	14	13	3	22.50

(a) Restaurant guide table, GoodEats.

restaurant	S	F	D	price
Summer Moon	21	25	19	47.50
Zakopane	24	20	21	56.00
Yamanote	22	22	17	51.50
Fenton & Pickle	16	14	10	17.50

(b) The skyline.

Figure 1: The restaurant table and the skyline.

restaurant that is better than all others on all criteria, however, as is usually the case in real life, and in real data. No one restaurant trumps all others. For instance, Summer Moon is best on food, but Zakopane is best on service.

While there is no one best restaurant with respect to our criteria, we can eliminate from consideration at least those restaurants which are worse on all criteria than some other. The Briar Patch BBQ should be eliminated because the Fenton & Pickle is better in comparison across all our criteria. The Brearton Grill is eliminated, in turn, because Zakopane is better than it on all criteria. If Zakopane were not in the table, the Brearton Grill would have remained a consideration. (Note that Summer Moon is not better than the Brearton Grill on D, decor, even though it is better on every other criterion.) Meanwhile the Fenton & Pickle is worse on every criterion than every other (remaining) restaurant, except on price, where it is the best. So it stays in consideration. (If we were to remove price as one of our criteria, the Fenton & Pickle should be eliminated too.) This would result in the choices in Figure 1(b).

In [3], a new relational operator is proposed called the *skyline operator*. They propose an extension to SQL with a skyline of clause as the language counterpart of the operator, which allows easy expression of the restaurant query we imagined above.

```
select ... from ... where ...
    group by ... having ...
    skyline of a1 [min | max | diff], ...,
               an [min | max | diff]
```

(a) Proposed skyline operator for SQL.

```
select * from GoodEats
    skyline of S max, F max,
               D max, price min
```

(b) Query to choose restaurants.

Figure 2: Skyline queries.

The skyline of clause is shown in Figure 2(a). Syntactically, it is similar to an order by clause. The columns a_1, ..., a_n are the attributes over which our preferences range. They must be of domains that have a natural ordering, such as integers, floats, and dates. The directives min and max specify whether we prefer low or high values, respectively. The directive diff states that one wants to retain best choices with respect to each distinct value of that attribute. Let min be the default directive if none is stated. The skyline query in Figure 2(b) over the table GoodEats in Figure 1(a) expresses what we had in mind above for choosing "best" restaurants, and would result in the answer set in Figure 1(b). If the table GoodEats had a column C for *cuisine*, we could add C diff to the skyline of clause to find the best restaurants by each cuisine group.

Skyline queries are not outside the expressive power of current SQL, but it is quite cumbersome to render skyline-like queries.[1] The skyline clause would be a useful syntactic addition to SQL, if skyline-like queries were to become commonplace. More important than ease of expression, however, is the expense of evaluation. A skyline operator in the relational engine would make skyline queries tractable to evaluate.

[1] It involves a self-θ-join over the table, and then an except of the join result against the original table.

1.2 Related Work

The concept of skyline in itself is not new in the least. Of course the search for optimal solutions is a well-established endeavor with a deep literature. Beginning in the 1960's, work focused on optimization with respect to multiple criteria. Techniques have been explored for finding good *utility functions* to combine effectively the multiple criteria of interest into a single score. Then traditional mathematical techniques for finding the optimal solution—with respect to a single criterion, the utility function in this case—could then be applied. Others, however, recognized that it is often difficult, if not virtually impossible, to find a reasonable utility function. Thus work in *multiple criteria optimization* focused on how to find *all* optimal solutions in the space with respect to the multiple criteria [17]. The definition of solution in this context is often the same as our definition for skyline: no other potential solution is better across all the criteria. This is referred to often as *Pareto optimal*.

Multiple-criterion optimization usually assumes an implicit solution space from which the optimal solutions are to be found. Often, this space is quite large, but also has properties that help to devise good techniques. For skyline queries, the space is explicit: it is the input relation of tuples. Also, one knows no particular properties of the space. The skyline idea has been studied before in this context of an explicit solution space as the *maximal vector problem*. In [12], the first algorithm to find the maximal vectors (or skyline tuples) from a set of vectors (or relation) was devised. In [13], the maximal vector problem is addressed in the context of computational geometry. In [1], they established that the average number of skyline tuples, $\hat{\mathsf{s}}_{d,n}$ is $\mathcal{O}((\ln n)^{d-1})$.[2] This is not useful for cardinality estimation, though, which is our endeavor.[3] Note that $(\ln n)^{d-1}$ is exceedingly greater than n, even for relatively small d and large n. It is true that $(\ln n)^{d-1}$ for any fixed d is $\mathcal{O}(n)$, but the n_0 at which the functions cross is very large. It was not established also whether $\mathcal{O}((\ln n)^{d-1})$ is a tight ("Θ") or loose ("o") bound. (We shall demonstrate it is loose.)

In a way, interest has returned to the maximal vector problem recently in the guise of skyline queries. Previous work was main-memory based, however, and not well suited to databases. Progress has been made as of recent on how to compute efficiently such queries in a relational system and over large datasets [3, 5, 11, 18]. In [3], the skyline operator is introduced. They posed two algorithms for it, a block-nested loops style algorithm (and variations) and a divide-and-conquer approach derived from work in [12, 13]. In [18], an algorithm for skyline evaluation is introduced that uses specialized indexing. In [5], we develop a general skyline algorithm based on the "block-nested loops" algorithm of [3] that is faster, pipelinable, and more amenable, we believe, to use in a relational query optimizer.

Steps remain, however, before a skyline operator can be realistically incorporated into today's relational systems. We need to understand better how the skyline operator composes with the other relational operators, both logically and algorithmically. The query optimizer must be able to incorporate the skyline operator without damaging its overall effectiveness, as well as being able to optimize queries that employ the skyline of clause. This will require a cost model for skyline queries, and a critical component of the cost model must be an estimator of the cardinalities of skyline queries' result sets.

A related topic is nearest-neighbors search. This has been studied in the context of relational systems too [16]. In [2], elements of a cost model for nearest-neighbor searches are considered, but for high-dimensional cases. In [11], they employ nearest-neighbors algorithms to pipeline the generation of skyline tuples.

Interest in skyline queries arises in most part from the desire to support queries with preferences in relational systems. In [4], a more general operator called *winnow* is introduced for the purpose of expressing preference queries. Skyline is a special case of winnow. Skyline and related techniques could make it possible to integrate easily certain *cooperative query answering*, *query relaxation*, and *preference query* techniques which have been proposed [6, 7, 9].[4]

Other than [1], there has not been effort to establish the expected cardinality of skyline (or maximal vector) sets, to our knowledge.

[2] For this, they made essentially the same assumptions that we shall make in Definition 1 about attributes' distributions and pair-wise independence.

[3] We should note that cardinality estimation was not the goal of [1]. They used the $\mathcal{O}((\ln n)^{d-1})$ result to establish a theoretical bound on the complexity of computing the "skyline".

[4] See [8] for an older survey of cooperative answering.

1.3 Outline

Skyline cardinality estimation is the focus of this paper. We establish results via both analytical and experimental means. In Section 2, we prove that the expected skyline cardinality is $\Theta((\ln n)^{d-1}/(d-1)!)$, in which n is the number of tuples in the input relation and d is the number of skyline attributes, under a basic model with assumptions of sparseness over attributes' domains (namely that there are virtually no duplicate values) and statistical independence across attributes. We prove more specifically that the expected skyline cardinality for two-dimensional skyline queries is the harmonic of the number of input tuples, and we prove the expected cardinality for multi-dimensional skyline queries is a higher-order harmonic—(a particular two parameter extension of the harmonic numbers—with respect to the number dimensions (skyline attributes) and the number of input tuples. This allows us to derive concrete size estimations that could be used in a cost model. We consider further the distribution of the value of the skyline cardinality itself, and show the distribution as derived by experiment. In Section 3, we consider the ramifications on the estimates, some counter-intuitive, as we relax the assumptions of the basic model. We consider the effects of the distributions of the values over attribute domains, when attributes are restricted to small finite domains, and pair-wise correlation of attributes. In Section 4, we conclude.

2 Skyline Cardinality

2.1 The Basic Model

We want to estimate the cardinality of the output relation of the skyline operator based upon its input relation. The input can be a base table of the database or a virtual table which is the intermediate result in a query's evaluation. Let us establish a basic model of assumptions about the input relation under which it will be possible for us to establish analytically the cardinality.

Definition 1 *Basic model of the input relation and skyline query.*

Let *dimension* refer to an attribute of the relation that participates in the skyline criteria.

(a) *Domain assumption (sparseness).* For each dimension, we assume that there are no duplicate values on the attribute across the tuples of the relation.

(b) *Independence assumption.* The dimensions are pair-wise statistically independent.

Consider a skyline operation with d dimensions over such an input relation of n tuples. (So the relation has at least d attributes, which obey the assumptions above.) Let $\mathsf{s}_{d,n}$ be the random variable which measures the number of tuples (the cardinality) of the output relation (that is, the set of the resulting skyline tuples). Let $\hat{\mathsf{s}}_{d,n}$ denote the expected value of $\mathsf{s}_{d,n}$.

2.2 Expected Cardinality

We are interested to determine $\hat{\mathsf{s}}_{d,n}$. Since under our basic model, no two input tuples share a value over any dimension, the tuples can be ordered totally on any given dimension. It is not necessary then to consider the actual values of the tuples. Instead, we can conceptually replace the value on, say, dimension i of a tuple by its *rank* in the total ordering along dimension i. Without loss of generality, we assume that we are *minimizing* over the dimensions for the skyline. Let rank 1 refer to the tuple with the smallest value (on that dimension), and n the one with the largest (n is the number of input tuples). We can now just refer to a tuple's rank on a dimension and ignore the actual value.

Lemma 2 [1] The skyline expected value $\hat{\mathsf{s}}_{d,n}$ for $d > 1$ and $n > 0$ obeys the following recurrence equation.

$$\hat{\mathsf{s}}_{d,n} = \frac{1}{n}\hat{\mathsf{s}}_{d-1,n} + \hat{\mathsf{s}}_{d,n-1}$$

For $n > 0$, $\hat{\mathsf{s}}_{1,n} = 1$.

Proof Consider $\hat{\mathsf{s}}_{1,n}$. Since no two tuples share the same value on the dimension, only the tuple with rank 1 is in the skyline.

Consider $\hat{\mathsf{s}}_{d,n}$, for $d > 1$. One tuple has rank n on dimension 1. This tuple cannot dominate any other tuple, since it has a higher value on dimension 1 than any other. What is the probability that this tuple itself is a skyline tuple? It is the probability that no other tuple dominates it on dimensions $2, \ldots, d$, given the independence assumption. As $\hat{\mathsf{s}}_{d-1,n}$ is the expected value of the number of skyline tuples out of n tuples on $d-1$ dimensions, then $\frac{1}{n}\hat{\mathsf{s}}_{d-1,n}$ represents the probability that this one tuple is part of the skyline.

Since the *n-th* ranked tuple on dimension 1 cannot dominate any other tuple, the estimated number of skyline tuples of the remaining $n-1$ is $\hat{\mathsf{s}}_{d,n-1}$. □

The recurrence for $\hat{\mathsf{s}}_{d,n}$ is related with harmonic numbers.

Definition 3 *Harmonic numbers.*

(a) The *harmonic of* n, for integers $n > 0$: $\mathsf{H}_n = \sum_{i=1}^{n} \frac{1}{i}$

(b) The *k-th order harmonic of* n, for integers $k > 0$ and integers $n > 0$: $\mathsf{H}_{k,n} = \sum_{i=1}^{n} \frac{\mathsf{H}_{k-1,i}}{i}$

Define $\mathsf{H}_{0,n} = 1$, for $n > 0$. Define $\mathsf{H}_{k,0} = 0$, for $k > 0$.

(c) The *k-th hyper-harmonic of* n, for integers $k > 0$ and integers $n > 0$: $\mathcal{H}_{k,n} = \sum_{i=1}^{n} \frac{1}{i^k}$

Note that $\lim_{k\to\infty} \mathcal{H}_{k,n}$ converges, for all $k > 1$. Let $\mathcal{H}_{k,\infty}$ denote $\lim_{k\to\infty} \mathcal{H}_{k,n}$.

Note that $\mathsf{H}_n = \mathsf{H}_{1,n} = \mathcal{H}_{1,n}$, for all $n > 0$.

The harmonic series in Definition 3(a) is well known. There is no consensus on a two-parameter generalization of the harmonic series. A common extension is that of Definition 3(c). These are less important for our work here, and we call these the hyper-harmonics. The $\mathcal{H}_{k,n}$ converge for $k > 1$. There is a second two-parameter generalization as given in Definition 3(b). This was introduced in [14, 15] in work on the logarithmic binomial formula. While this generalization is not common, it is a natural extension of the harmonic series, and it is useful for us.[5] The $\mathsf{H}_{k,n}$ do not converge for $k > 1$. Let us enumerate some of the properties of these.

Lemma 4 *Properties of harmonic numbers.* Let k and n be positive integers.

(a) [10] $\mathsf{H}_n \doteq \ln n + \gamma + \frac{1}{2n}$ and $\lim_{n\to\infty} \mathsf{H}_n - \ln n - \gamma = 0$

(b) [15] $\mathsf{H}_{k,n} < n$, for all $k > 0$ and $n > 0$, and $\lim_{k\to\infty} \mathsf{H}_{k,n} = n$

(c) $\mathsf{H}_{k,n}$ monotonically increases with respect to k and with respect to n.

(d) [15] $\mathsf{H}_{k,n} = \frac{1}{n}\mathsf{H}_{k-1,n} + \mathsf{H}_{k,n-1}$

Recall $\mathsf{H}_{0,n} = 1$ and $\mathsf{H}_{k,0} = 0$. (Thus, $\mathsf{H}_{1,n} = \mathsf{H}_n$.)

(e) [15] $\mathsf{H}_{k,n} = \sum_{i=1}^{n} \binom{n}{i} (-1)^{i-1} i^{-k}$

Note that γ is the Euler-Mascheroni constant, $\gamma \doteq .577$.

It is possible immediately to relate $\hat{\mathsf{s}}_{d,n}$ and $\mathsf{H}_{k,n}$.

Theorem 5 $\hat{\mathsf{s}}_{d,n} = \mathsf{H}_{d-1,n}$.

Any particular $\mathsf{H}_{k,n}$ can be solved in terms of $\mathcal{H}_{j,n}$'s ($1 \le j \le k$). The H_n and $\mathcal{H}_{k,n}$ are easy to compute, or approximate, and so could be used within a cost model on-the-fly. (Recall that $\mathsf{H}_n = \mathcal{H}_{1,n}$.) For instance, we can derive that

- $\mathsf{H}_{2,n} = \frac{1}{2}\mathsf{H}_n^2 + \frac{1}{2}\mathcal{H}_{2,n}$,
- $\mathsf{H}_{3,n} = \frac{1}{6}\mathsf{H}_n^3 + \frac{1}{2}\mathsf{H}_n\mathcal{H}_{2,n} + \frac{1}{3}\mathcal{H}_{3,n}$, and
- $\mathsf{H}_{4,n} = \frac{1}{24}\mathsf{H}_n^4 + \frac{1}{3}\mathsf{H}_n\mathcal{H}_{3,n} + \frac{1}{8}\mathcal{H}_{2,n}^2 + \frac{1}{4}\mathsf{H}_n^2\mathcal{H}_{2,n} + \frac{1}{4}\mathcal{H}_{4,n}$.

[5] What is effectively a generalization of the $\mathsf{H}_{k,n}$'s appears in [10] (in Section 1.2.9).

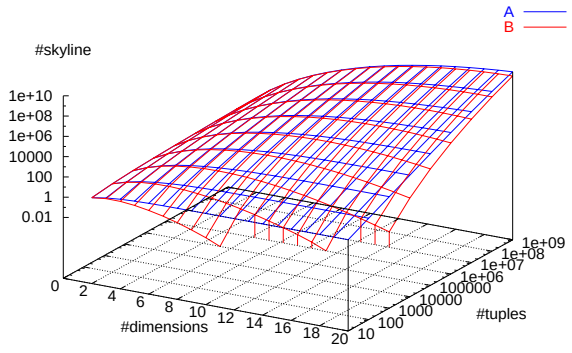


Figure 3: Plot of $\hat{\mathsf{s}}_{d,n}$ (A) and $\mathsf{H}_n^{d-1}/(d-1)!$ (B).

This can be generalized as follows.

Theorem 6 $\mathsf{H}_{k,n} = \sum_{\substack{c_1,\ldots,c_k \geq 0 \ \wedge \\ 1\cdot c_1 + 2\cdot c_2 + \ldots + k\cdot c_k = k}} \prod_{i=1}^{k} \frac{\mathcal{H}_{i,n}^{c_i}}{i^{c_i} \cdot c_i!}$ for $k \geq 1$ and $n \geq 1$, with the c_i's as integers.

For any k, the coefficients of the terms in the summation sum to one. The number of terms to express $\mathsf{H}_{k,n}$ in terms of $\mathcal{H}_{j,n}$'s is $\wp(k)$, the number of ways to partition the positive integer k as a sum of positive integers. Since $\wp(k)$ grows quickly—for instance, $\wp(10) = 42$ and $\wp(20) = 627$—it is not a viable to solve for $\mathsf{H}_{k,n}$'s in this way.

We can study properties of $\mathsf{H}_{k,n}$ to establish bounds on skyline cardinality estimation.[6]

Theorem 7 *Bounds on* $\mathsf{H}_{k,n}$*, and hence,* $\hat{\mathsf{s}}_{d,n}$*.*
(a) $\mathsf{H}_n^k/k! < \mathsf{H}_{k,n} < \min(\mathsf{H}_n^k, n)$
(b) $\mathsf{H}_n^{d-1}/(d-1)! < \hat{\mathsf{s}}_{d,n} < \min(\mathsf{H}_n^{d-1}, n)$
(c) $\mathsf{H}_{k,n}$ is $\Theta((\ln n)^k/k!)$
(d) $\hat{\mathsf{s}}_{d,n}$ is $\Theta((\ln n)^{d-1}/(d-1)!)$

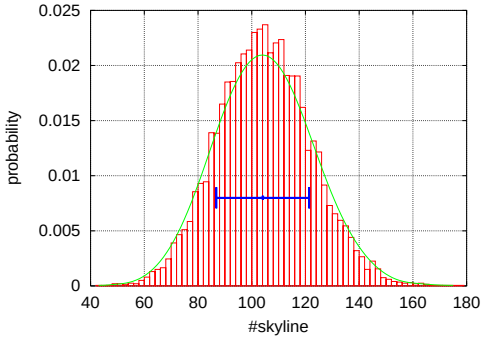
This improves on the asymptotic bound established in [1]. While $\hat{\mathsf{s}}_{d,n}$ is $\Theta((\ln n)^{d-1}/(d-1)!)$, for larger d's, it requires very large n's before $\hat{\mathsf{s}}_{d,n}$ converges closely on $\mathsf{H}_n^{d-1}/(d-1)!$. For $d = 3$ and $n = 10^7$, $\mathsf{H}_{10^7}^2/2 \doteq 139.37$ and $\hat{\mathsf{s}}_{3,10^7} = \mathsf{H}_{2,10^7} \doteq 140.19$, whereas, for $d = 10$ and $n = 10^7$, $\mathsf{H}_{10^7}^9/9! \doteq 277,708$ and $\hat{\mathsf{s}}_{10,10^7} = \mathsf{H}_{9,10^7} \doteq 357,604$. Interestingly, while $\mathsf{H}_n^k/k!$ is monotonically increasing with respect to n, it is not with respect to k. So $\mathsf{H}_n^{d-1}/(d-1)!$ is not a good concrete estimate (but it can be used when no better estimate is available).

Of course, the $\mathsf{H}_{k,n}$ can be numerically approximated off-line, and then a look-up table used at run-time. We computed $\mathsf{H}_{d-1,n}$ for $d = 2,\ldots,20$ and $n = 10,\ldots,10^7$ by factors of ten.[7] Figure 3 plots this, along with $\mathsf{H}_n^{d-1}/(d-1)!$. As with logarithms, interpolation can be safely used to estimate values that are not in the look-up table.

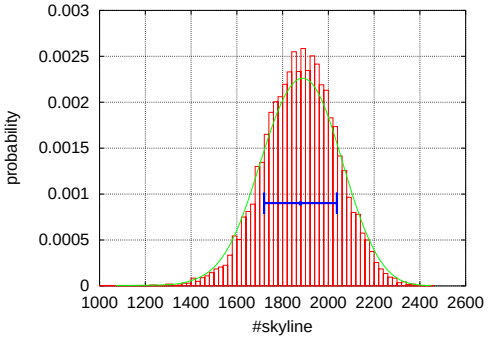
2.3 Skyline Distribution

We have solved for the expected value of skyline cardinality, $\hat{\mathsf{s}}_{d,n}$ (with respect to basic model in Definition 1), but we do not know the distribution of the random variable $\mathsf{s}_{d,n}$. If its variance were huge, for instance,

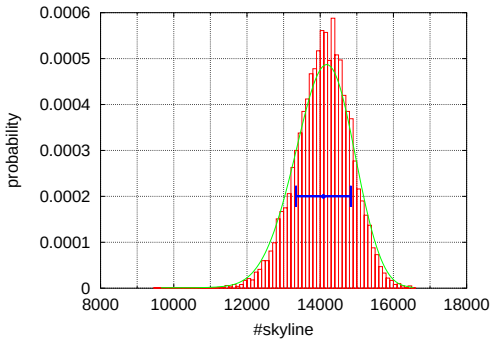
[6] Our notational convention is that superscripts are exponentiation and subscripts are parameters.
[7] One must take care to avoid overflow and accumulated round-off when approximating these.



(a) Distribution of $\mathsf{s}_{3,10^6}$.



(b) Distribution of $\mathsf{s}_{5,10^6}$.



(c) Distribution of $\mathsf{s}_{7,10^6}$.

	$\mathsf{H}_{d-1,10^6}$	simulation		
d	$\hat{\mathsf{s}}_{d,10^6}$	μ	median	σ
3	104	104	104	17.3
5	1,880	1,878	1,882	158.5
7	14,087	14,087	14,120	745.9

(d) Means, medians, and standard deviations.

Figure 4: Distributions of $\mathsf{s}_{d,10^6}$ (for $d = 3$, 5, and 7).

our use of $\hat{\mathsf{s}}_{d,n}$ in a cost model would be of limited utility. It is also possible for the median to be less than the mean, with a long tail towards n. We are really interested in likely values the optimizer will encounter, not the *expected value*, per se, which itself as a value might never occur. In a cost model, it is important to anticipate the chance of being significantly off the estimation, and especially when the actual cardinality is exceedingly large in comparison with the estimate. The query plan can be made to accommodate contingencies, to varying degrees.

It is possible to infer some properties of the distribution just given what we know of $\hat{\mathsf{s}}_{d,n}$. The domain of $\mathsf{s}_{d,n}$ is $1 \ldots n$, of course. For d and n combinations that are likely in skyline queries in practice, $\hat{\mathsf{s}}_{d,n}$ is close to the 1-end of the spectrum. (See Figure 4(d).) This statistically limits how large the variance can be. (There cannot be much probability that $\mathsf{s}_{d,n}$ is near n, since this would serve to inflate $\hat{\mathsf{s}}_{d,n}$.)

It would be interesting—but a major undertaking—to develop a good statistical understanding of the distribution of $\mathsf{s}_{d,n}$. In the meantime, we can study experimentally the distribution. For each dimension (d) of 3, 5, and 7, we ran numerical "simulations" as follows. For each *trial*, we generated one million tuples of d attributes randomly. Each attribute was of type integer and its value was randomly chosen across all values.[8] The number of skyline tuples (minimizing over the d values) was then determined. A simulation then consisted of 10,000 trials. Figures 4(a), (b), and (c) show the distributions of the 10,000 trials for $\mathsf{s}_{d,10^6}$ for $d = 3$, 5, and 7, respectively. The data points were binned into about sixty bins in each case to approximate the distribution via a histogram. We normalize the y-axis to probability so the area covered by the histogram is one. In each case, the error-bar represents the mean and spans one standard deviation to

[8] The values range over 1, ..., 2,147,483,647.

the leaf and to the right of the mean. The super-imposed curve is a bezier fit of the data. The distributions of $\mathsf{s}_{d,n}$ are well behaved and resemble normal (Gaussian) distributions. That the medians are so near the means in the simulations offers evidence that the true distributions have little if no skew.

3 Generalizing from the Basic Model

Next, we explore the effects of relaxing the assumptions of our basic model from Definition 1.

3.1 Domain Distributions

In the basic model, we make no assumption about attributes' distributions, beyond the assumption of sparseness (that there are, virtually, no tuples that share a value on an attribute). Remarkably, the distributions are immaterial. The skyline operator compares tuples always by the same attribute (for each skyline attribute), so the two values being compared at any given point are drawn from the same distribution. Under sparseness, there are no ties; one or the other tuple trumps on any attribute comparison with equal probability.

The skyline is no longer independent of the attributes' distributions when the sparseness condition does not hold, thus allowing tuples to tie on values. We consider relaxing the sparseness condition next.

3.2 Sparseness versus Repeated Values

Many attribute domains are small. For instance, a *Boolean* type only allows the values *true* and *false*. Furthermore, we are really interested in the *range* of values that occur over an attribute, rather than the domain, per se. For a given distribution of tuples, the values may cluster on just a few of the possible values. When we relax the sparseness assumption from Definition 1, it allows for tuples to share values. Furthermore, it allows for duplicate tuples (at least with respect to the skyline attributes). Since most real data is like this, we are interested in how this affects the number of skyline tuples.

Definition 8 Let $\mathsf{s}_{\langle p\times p\rangle,n}$ denote the random variable that measures the number of skyline tuples from a relation of n tuples, with respect to a two-dimensional skyline query. Each of the skyline attributes range over p values, the tuples' values are uniformally distributed over them, and the skyline attributes are pair-wise statistically independent.

Let $\mathsf{s}_{\langle p^d\rangle,n}$ more generally denote the number of skyline tuples with respect to a d-dimensional skyline query under the same conditions. Let $\hat{\mathsf{s}}_{\langle p^d\rangle,n}$ denote the expected value.

It is straightforward to establish $\hat{\mathsf{s}}_{\langle p^d\rangle,n}$'s behavior in the limit, for d, p, and n.

Theorem 9 *Bounds on* $\mathsf{s}_{\langle p^d\rangle,n}$ *and* $\hat{\mathsf{s}}_{\langle p^d\rangle,n}$.

(a) $1 \le \mathsf{s}_{\langle p^d\rangle,n} \le n$

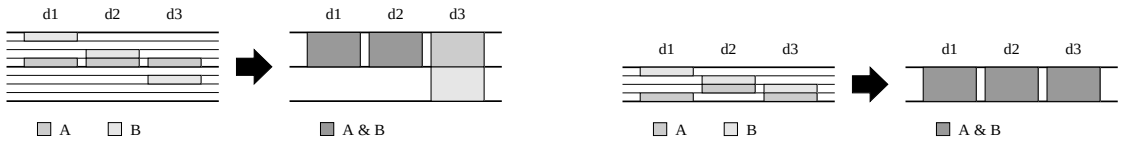
(b) $\lim_{d\to\infty} \hat{\mathsf{s}}_{\langle p^d\rangle,n} = n$

(c) $\lim_{p\to\infty} \hat{\mathsf{s}}_{\langle p^d\rangle,n} = \hat{\mathsf{s}}_{d,n}$

(d) $\lim_{n\to\infty} \dfrac{\hat{\mathsf{s}}_{\langle p^d\rangle,n}}{n} = \dfrac{1}{p^d}$

We are more specifically interested in how $\hat{\mathsf{s}}_{\langle p^d\rangle,n}$ relates to $\hat{\mathsf{s}}_{d,n}$. Does value repetition (non-sparseness) increase the number of expected skyline tuples, or decrease it?

One way to view this is to consider a relation that starts as sparse, but we *bin*, or *partition*, the tuple's values for each attribute into just a few bins (values). So the tuples in the initial relation share no values, but after *binning*, they do. Also, after binning, there can be duplicate tuples. Figure 5 shows two effects that occur. In some cases, a pair of tuples that were incomparable before can be comparable after binning. Figure 5(a) shows this. Tuples A and B were incomparable before. (They both might be skyline.) But after, A trumps B. (So only A could be in the skyline of the binned relation.) It is possible to lose skyline tuples by this effect. In other cases, all the skyline attribute values of the two tuples become the same. Figure 5(b) shows this. It is possible to gain skyline tuples (due to duplicates) by this effect.



(a) A pair of tuples become comparable.

(b) A pair of tuples become identical.

Figure 5: Binning effects.

(a) Two edges with no repeats. $\hat{\mathsf{s}}_{2,2} = 3/2$.

(b) Two edges with repeats. $\hat{\mathsf{s}}_{2\times 2,2} = 11/8$.

Figure 6: Choose two edges.

The probability of value sharing occurring, as in Figure 5(a), is much greater generally than that of the tuples becoming identical, as in 5(b).[9] That is, there is a higher probability that tuples will share values on some dimensions, but not on all. This gap increases with the number of dimensions. So we would expect the first effect to dominate the second, and for the number of skyline to go down due to binning.

There is the case when there are many more tuples than possible value combinations (p^d). In this case, duplicates reign. By uniformity, there is with high probability, for each possible value combination, a tuple that matches it. So there is a tuple with the best p on each dimension, and so this is the only possibility for skyline. How many skyline tuples there are depends on how many duplicates of this there are. The limit in Theorem 9(d) shows this effect.

Consider the case of two tuples of two dimensions, with each dimension as Boolean. This is the same as considering two edges placed on a 2×2 bipartite grid, as in Figure 6. If no vertex (value) sharing is allowed, the only possibilities are as in Figure 6(a). Allowing vertex sharing, there are sixteen possibilities as in Figure 6(b) (choosing two possible edges, with replacement). By our assumptions of uniform distributions and independence, all sixteen possibilities are equally likely.

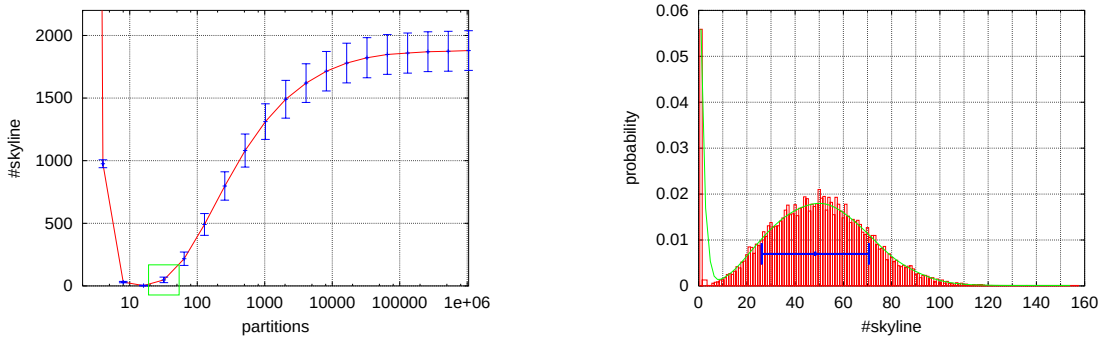
The dark-hued diagonal in Figure 6(b) represents the cases in which there are no repeated values. These behave exactly as $\mathsf{s}_{2,2}$. The light-hued diagonal are the duplicate cases, so $\hat{\mathsf{s}} = 2$ over these. The rest are cases of repeats, but no duplicates. For these, $\hat{\mathsf{s}} = 1$. Note there are twice as many cases of repeats (but no duplicates) than of duplicates.

We can solve via the probabilities for $\hat{\mathsf{s}}_{\langle p\times p\rangle,2}$ (and for higher values of n, although it becomes increasingly cumbersome).

Lemma 10 $\hat{\mathsf{s}}_{\langle p\times p\rangle,2} = \mathsf{H}_2 - \dfrac{1}{p} + \dfrac{3}{2p^2}$

For $p > 1$, $\hat{\mathsf{s}}_{\langle p\times p\rangle,2} < \hat{\mathsf{s}}_{2,2}$.

[9] Except in the very extreme, of course. When we bin all values to a single bin for each dimension, all the tuples become identical.



(a) Plotting "$\hat{\mathsf{s}}_{\langle p^5\rangle,10^6}$" at different p's.

(b) Distribution of "$\mathsf{s}_{\langle 32^5\rangle,10^6}$".

Figure 7: Varying p, the partition degree (for $d = 5$.)

Conjecture 11 For $n < p^d \mathsf{H}_{d,n}$, $\hat{\mathsf{s}}_{\langle p^d\rangle,n} < \hat{\mathsf{s}}_{d,n}$.

This conjecture may be hard to prove, especially for cases of small n's and p's. For instance, while $\hat{\mathsf{s}}_{\langle p\times p\rangle,2} < \hat{\mathsf{s}}_{2,2}$ (for $p > 1$), $\hat{\mathsf{s}}_{\langle p\times p\rangle,2}$ is not monotone with respect to p. Of course, for our purposes, we are only interested in relatively large n. To ascertain experimentally $\hat{\mathsf{s}}_{\langle p^d\rangle,n}$ and the distribution of $\mathsf{s}_{\langle p^d\rangle,n}$, as in Section 2.3, we ran 10,000 trials for each of $p = 2$ to 2^{20} (by doubling) for $d = 5$ and $n = 10^6$. Figure 7(a) shows this. The error-bars represent the standard deviations. Figure 7(b) shows the distribution of $\mathsf{s}_{\langle 32^5\rangle,10^6}$, and is constructed in the same way as those in Figure 4.

To the left of the nadir in Figure 7(a), the number of tuples dominates the possible value combinations, and the distribution counts the number of duplicates of that best scoring combination. These distributions are true Gaussian, by the central limit theorem. Around the nadir is the balance point between the duplicate effect and value sharing. Here, the distributions are odder, as in Figure 7(b). That distribution is bimodal, in which many trials hit the "best" value combination once (and so have a skyline of one) and many do not. As we move to the right, the distributions become well behaved and Gaussian-like again. The number of skyline is diminished due to the value-sharing effect, which acts as dimensional reduction. The distributions converge on that of $\mathsf{s}_{5,10^6}$ as p grows and the relation becomes virtually sparse.

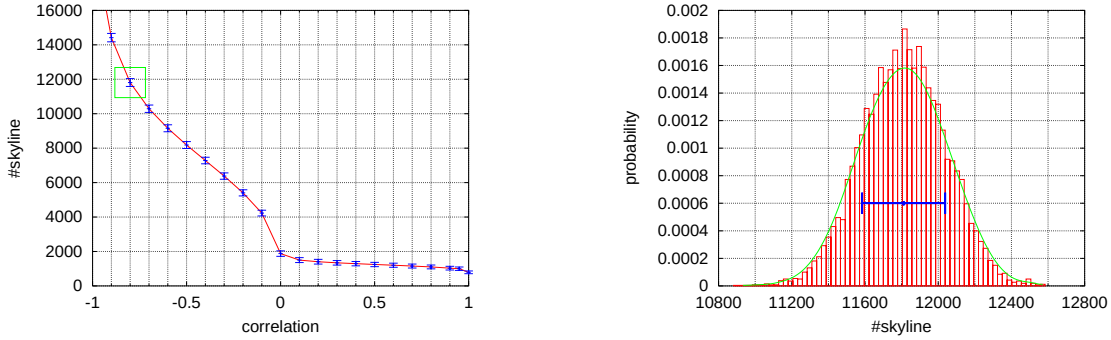
This discovery that $\hat{\mathsf{s}}$ diminishes as p does has ramifications for users of skyline queries, not just for the cost model. A tempting strategy when the skyline query returns too few results (for the user's liking) is to bin the dimensions' values further. For instance, we might see little difference between a restaurant at which the average meal is \$25 and one at which it is \$22. So we might not want one restaurant trumping another on cost unless it were at least \$5 dollars less expensive. This could lead us to bin price into five dollar brackets. However, this reasonable-seeming strategy backfires. As we have just seen, binning would only reduce further the number of skyline.

3.3 Correlation and Anti-correlation

Our other assumption in the basic model in Definition 1 is that of statistical independence of the dimensions (skyline attributes). This is rarely true for real data, and the skyline operator is sensitive to correlation.

Definition 12 Let $\mathsf{s}^{(r)}_{d,n}$ denote the random variable that measures the number of skyline tuples of an n-tuple relation with respect to a d-dimensional skyline query, for which the d skyline attributes are pair-wise statistically independent, save for one pair that are correlated at r.

The skyline is affected quite differently by correlation and anti-correlation. High correlation between two skyline attributes acts as dimensional reduction. If tuple A has a better value on one of the dimensions than B, with high probability, it also does on the other dimension. At $r = 1.0$, that probability is one, so one of



(a) Plotting "$\hat{\mathsf{s}}^{(r)}_{5,10^6}$" at different correlations.

(b) Distribution of $\mathsf{s}^{(-.8)}_{5,10^6}$.

Figure 8: Varying r, the correlation (for $d = 5$.)

the dimensions is effectively eliminated. Anti-correlation is the antithesis of skyline, however. If A is better than B on one dimension, it is likely that B is better than A on the other. At $r = -1.0$, *all* tuples are in the skyline. In a way, skyline queries with highly anti-correlated skyline attributes do not make much sense. One is trying to optimize two values that are strictly trade-offs. Nevertheless, any realistic implementation of the skyline operator would have to accommodate such cases.

To ascertain experimentally $\hat{\mathsf{s}}^{(r)}_{d,n}$ and the distribution of $\mathsf{s}^{(r)}_{d,n}$, as in Section 2.3, we ran 10,000 trials for each of $r = -0.9$ to 1.0 (and -0.95) by steps of 0.1 for $d = 5$ and $n = 10^6$. Figure 8(a) shows this. The error-bars represent the standard deviations. Figure 8(b) shows the distribution of $\mathsf{s}^{(-.8)}_{5,10^6}$, and is constructed in the same way as those in Figure 4. The correlated cases behave as expected. Note that a simple linear interpolation between $d = 5$ and $d = 4$ would not be accurate to model the correlation. However, it would be easy to fit a function for the interpolation. The $\hat{\mathsf{s}}$ grow rapidly under anti-correlation, but not as fast as people expected, and the distributions remain remarkably well behaved and Gaussian-like. The standard deviations also do not diverge rapidly. Only in extreme anti-correlation does $\hat{\mathsf{s}}$ become really large.

4 Conclusions

Skyline queries offer a natural extension to min and max aggregation, and allow for one to query for nearest matches to one's objectives. The skyline operator, and potential extensions, may offer support for richer classes of queries with preferences. It may soon be worthwhile to add the skyline clause—and the underlying skyline operator—to relational systems.

Progress has been made recently to develop good methods for evaluating skyline queries in a relational setting. In this paper, we have made some initial progress in understanding the cardinality of the results of skyline queries. This offers a basis for what would be required in the optimizer's cost model to accommodate the skyline operator. To be sure, work remains, and many details to resolve, to extend adequately the cost model for skyline. We need a fuller understanding of skyline under correlation, for instance how multiple correlations affect the skyline, and how the attributes' distributions affect the skyline's cardinality when their domains are small (that is, p is small).

Our analyses and insights should help us to understand better the uses of skyline queries, and help us in this next stage to build better, more robust algorithms for skyline's computation. We hope this work also yields more general insights into the nature of multi-dimensional data. In the longterm, skyline may provide us, in turn, tools for the relational model and systems. Once integrated, the skyline operator might provide the query optimizer sometimes with more efficient plans for evaluating queries with self-joins. Since the skyline has the property that it is independent of attributes' domains (for large p), skyline statistics may provide a new type of useful database statistics that could yield better cost estimations for queries generally.

Skyline work opens up many new avenues.

References

[1] J. L. Bentley, H. T. Kung, M. Schkolnick, and C. D. Thompson. On the average number of maxima in a set of vectors and applications. *JACM*, 25(4):536–543, 1978.

[2] S. Berchtold, C. Böhm, D. A. Keim, and H.-P. Kriegel. A cost model for nearest neighbor search in high-dimensional data space. In *Proceedings of the Sixteenth PODS*, pages 78–86, 1997.

[3] S. Börzsönyi, D. Kossmann, and K. Stocker. The skyline operator. In *Proceedings of the 17th ICDE*, pages 421–430, 2001.

[4] J. Chomicki. Querying with intrinsic preferences. In *Proceedings of EDBT*, 2002.

[5] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang. On skyline computation, 2002. Submitted.

[6] W. W. Chu, H. Yang, K. Chiang, M. Minock, G. Chow, and C. Larson. CoBase: A scalable and extensible cooperative information system. *Journal of Intelligent Information Systems (JIIS)*, 6(2/3):223–259, May 1996.

[7] T. Gaasterland, P. Godfrey, and J. Minker. Relaxation as a platform for cooperative answering. *Journal of Intelligent Information Systems (JIIS)*, 1(3/4):293–321, 1992.

[8] T. Gaasterland, P. Godfrey, and J. Minker. An overview of cooperative answering. In R. Demolombe and T. Imielinski, editors, *Nonstandard Queries and Nonstandard Answers*, Studies in Logic and Computation 3, chapter 1, pages 1–40. Clarendon Press, Oxford, 1994.

[9] T. Gaasterland and J. Lobo. Qualifying answers according to user needs and preferences. *Fundamenta Informaticæ*, 32(2):121–137, 1997.

[10] D. E. Knuth. *Fundamental Algorithms: The Art of Computer Programming*, volume 1. Addison Wesley, Reading, MA, second edition, 1973.

[11] D. Kossmann, F. Ramask, and S. Rost. Shooting stars in the sky: An online algorithm for skyline queries. In *Proceedings of 28th International Conference on Very Large Data Bases (VLDB-2002)*, Aug. 2002. To appear.

[12] H. T. Kung, F. Luccio, and F. P. Preparata. On finding the maxima of a set of vectors. *JACM*, 22(4):469–476, 1975.

[13] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 1985.

[14] S. Roman. The logarithmic binomial formula. *American Mathematics Monthly*, 99:641–648, August-September 1992.

[15] S. Roman. The harmonic logarithms and the binomial formula. *Journal of Combinatorial Theory, Series A*, 63:143–163, 1993.

[16] N. Roussopoulos, S. Kelley, and F. Vincent. Nearest neighbor queries. In M. J. Carey and D. A. Schneider, editors, *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 71–79. ACM Press, 1995.

[17] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation, and Application*. John Wiley & Sons, New York, 1986.

[18] K.-L. Tan, P.-K. Eng, and B. C. Ooi. Efficient progressive skyline computation. In *Proc. of 27th VLDB*, pages 301–310, 2001.

[19] *Zagat Toronto Restaurants*. Zagat Survey, 2002.