K-Dominance in Multidimensional Data: Theory and Applications

Thomas Schibler[1] and Subhash Suri[2]

1 **University of California, Santa Barbara, CA 93106, USA.**
`tschibler@gmail.com`
2 **University of California, Santa Barbara, CA 93106, USA.**
`suri@cs.ucsb.edu`

Abstract

We study the problem of k-dominance in a set of d-dimensional vectors, prove bounds on the number of maxima (skyline vectors), under both worst-case and average-case models, perform experimental evaluation using synthetic and real-world data, and explore an application of k-dominant skyline for extracting a small set of top-ranked vectors in high dimensions where the full skylines can be unmanageably large.

1998 ACM Subject Classification Information Systems, Mathematics of Computing

Keywords and phrases Dominance, skyline, database search, average case analysis, random vectors

Digital Object Identifier 10.4230/LIPIcs.ESA.2017.1

1 Introduction

In multi-criteria optimization and decision-making applications, there is often no single best answer, and a popular approach is to use pareto optimality. The set of pareto optimal points, which are the coordinate-wise *undominated* solutions, is called the *skyline*. Unfortunately as the dimension of the data grows,[1] the size of the skyline tends to explode and most, if not all, of the input vectors can appear on the skyline [4, 7, 8]. A database query for a car or a smart phone, for instance, can easily produce an overwhelming number of incomparable choices. In the National Basketball Association's (NBA) database of 21,961 players in 17 dimensions (scoring attributes), more than 1400 players appear on the skyline (see Figure 2 for real-world datasets with large skylines). The problem is even more pronounced in crowdsourced data such as movies or consumer product ratings—each input vector is the rating profile of a product by the users—where virtually every product can be highly ranked by some user, potentially elevating it to the skyline. While the classical result of Bentley et al [4] shows that the expected size of the skyline of n random vectors, whose components are chosen independently, is $O((\log n)^{d-1})$ in d-dimensions, the exponential dependence on d renders the skyline useless even in theory except in very low dimensions.

The *k-dominant skyline* $\mathcal{KDS}$ was introduced recently as a way to tame this curse of dimensionality, where by relaxing d-dominance to k-dominance, for $k < d$, many more points can be eliminated from the skyline, resulting in a smaller, more manageable, set of maxima. Formally, given a finite set of points $\mathcal{V}$ in R^d, a point u is said to *k-dominate* another point

[1] Although in theory all input points can appear on the skyline even in two dimensions, this pathological behavior is rarely observed in low-dimensional real-world data.

v if $u_i \geq v_i$ holds for k of the dimensions, $i = 1, 2, \ldots, d$, with strict inequality in at least one dimension. We use the notation $u \succ_k v$ to indicate k-domination of v by u, and write $u \not\succ_k v$ when u does not k-dominate v. The *k-dominant skyline* of $\mathcal{V}$, denoted $\mathcal{KDS}(\mathcal{V}, k)$, is the set of points that are not k-dominated:

$$\mathcal{KDS}(\mathcal{V}, k) \;=\; \{v \in \mathcal{V} \mid u \;\not\succ_k\; v,\; \forall\, u \in \mathcal{V} \setminus \{v\}\}.$$

In this paper we make both theoretical and applied contributions to the study of k-dominant skylines.

1.1 Main Contributions

We derive the first non-trivial (poly-logarithmic) upper bound on the *average* size of $\mathcal{KDS}$ for random vectors, and in the process generalize the result of Bentley et al. [4]. Our experiments on synthetic and real-world data show that $\mathcal{KDS}$ size conforms to these bounds. We then use movie ratings and NBA basketball datasets to show that $\mathcal{KDS}$ is an effective tool for high-dimensional ranking queries. The full-dimensional skylines of these data sets are unmanageably large, but $\mathcal{KDS}$ consistently finds top vectors using purely pareto property, without the need for ad hoc preference (utility) functions. (We discuss these issues further in Section 6.) Specifically, our paper makes the following contributions:

1. [*Average Case Bound:*] Let $\mathcal{V}$ be a set of n vectors in d dimensions where the components of each vector are distributed independently of each other, and for each component the magnitudes form a random permutation of $\{1, 2, \ldots, n\}$. We show that the expected number of vectors appearing on the k-dominant skyline is

$$\begin{cases} O\left((\log n)^{2k-d-1}\right) & \text{for } k > \frac{d+1}{2} \\ O(1) & \text{otherwise} \end{cases}$$

   Our result smoothly interpolates the cardinality of $\mathcal{KDS}$ for all values of k, and subsumes the classical result of Bentley et al. [4] as a special case for $k = d$.

2. [*Efficient Algorithm:*] The k-dominance relation does not obey transitivity, which is needed for the efficient (sub-quadratic) computation of skylines. We show that $\mathcal{KDS}$ can be computed from the (traditional) skylines of all k-dimensional projections of the input vectors. Our algorithm runs in worst-case time $O\left(d^{d-k} n \log^{k-1} n\right)$, which is subquadratic even for non-constant dimensions as long as $d \leq c \log n / \log\log n$, for some constant c.

3. [*Experiments:*] Our average case analysis assumes attribute distinctness and statistical independence of input vectors. While there is no reason to believe that real-world datasets meet these conditions, our experiments on a variety of data show that the $\mathcal{KDS}$ size follows the exponential growth predicted by our analysis.

4. [*Applications:*] We show that $\mathcal{KDS}$ is a useful tool for *robust ranking* in high dimensions. For instance, which players should be called the "10 most dominant NBA players" when more than 1400 are pareto optimal (undominated)? Our experiments show that the vectors (NBA players or movies) that are the *first ones* to populate the k-dominant skyline, and therefore have the most longevity because k-dominance is a monotone property, are indeed the natural candidates for top ranking.

1.2 Related Work

The skyline or maxima problem has a long history in computational geometry, databases, and mathematical optimization [3, 4, 13, 6, 5, 9, 15, 19, 20, 21, 27, 28, 14, 22, 25], and the quest for efficient algorithms that scale to high dimensions and large input continues to this day. It has also been observed that as the dimension grows, the skylines can quickly lose their data-reduction utility because most of the input vectors may appear on the skyline [7, 10].

A number of approaches have been considered for identifying a smaller size representation of the full skyline in high dimensions. One simple method aggregates all the dimensions into a single *utility function*, and uses the ranking defined by this function. This approach while computationally attractive is problematic because it simply transfers all the burden to the "designer" of the utility function. Indeed, one of the important benefits of skyline-based approaches is to let the users *explore* various tradeoffs across different components such as price, location, brand etc. [7, 8]. Another approach is to compute a small set of input vectors that approximates the skyline over *all* preference functions. Unfortunately, these approaches are computationally intractability even for linear preference functions, as in the case of k-regret set [10, 24, 26, 1], or *approximate*-skyline [18].

The focus of our paper is to both prove an upper bound on the size of the $\mathcal{KDS}$ and to explore its applications for ranking high dimensional data. There exists a substantial and rich literature on estimating the cardinality of (conventional) skylines [4, 6, 14, 15, 21, 28], under a variety of data models, including in-memory, distributed, and data streams. However, very little is known about the cardinality of k-dominant skylines. The k-dominant skyline was introduced in [7] but the primary goal of that paper is to design efficient heuristics for computing the $\mathcal{KDS}$ scalably in practice. (In a related work, Chan et al. [8] compute vectors that appear in many k-dominant skylines, but again the paper is concerned with the design and evaluation of an efficient heuristic.) In [17], Hwang et al. consider certain threshold phenomena in k-dominant skylines under a continuous probability model, and derive limit bounds as $n, d \to \infty$. By contrast, we establish parametric upper bounds on the cardinality of $\mathcal{KDS}$ under both random and worst-case inputs, similar to those for the conventional skylines obtained by Bentley et al. [4] or Buchta [6].

2 K-Dominance and the Worst-Case Bound

Let $\mathcal{V}$ be a set of n vectors in d-dimensional space. We will use the letters u and v to denote generic vectors of $\mathcal{V}$, and v_i as the ith coordinate of $v \in \mathcal{V}$. That is, $v = (v_1, v_2, \ldots, v_d)$ is the coordinate representation of vector v. We will use the terms vectors and points interchangeably since vectors are commonly viewed as points in d-dimensional space. Given two vectors $u, v \in \mathcal{V}$, we say that u *dominates* v, denoted $u \succ v$, if $u_i \geq v_i$ for $i = 1, 2, \ldots, d$, with strict inequality in at least one dimension. We say that a vector u *k-dominates* v if $u_i \geq v_i$ for at least k of the d possible dimensions, with strict inequality in at least one. We write this as $u \succ_k v$, generalizing the original definition of domination, which is equivalent to d-domination $\succ_d$. The *k-dominant skyline* of $\mathcal{V}$ is the set of vectors that are not k-dominated by any other vector. That is,

$$\mathcal{KDS}(\mathcal{V}, k) \;=\; \{v \in \mathcal{V} \mid u \;\not\succ_k\; v,\; \forall\, u \in \mathcal{V} \setminus \{v\}\}.$$

The k dimensions involved in k-dominance $u \succ_k v$ need not be the same as those involved in $u' \succ_k v'$, and so the k-dominant skyline does not equal the skyline of $\mathcal{V}$ after projection onto some fixed k-dimensional subspace. It also means that the k-dominance relation is *not transitive*: if $u \succ_k v$ and $v \succ_k w$, then we do not necessarily have $u \succ_k w$.

Given a set of input vectors $\mathcal{V}$ in d dimensions, we define a *directed graph* G_k, called the *k-dominant graph*, as follows: the vertices of G_k correspond to the vectors of $\mathcal{V}$, and its edges correspond to k-dominance relationships between pairs of vectors. That is, the graph G_k has a *directed* edge (u, v) whenever $u \succ_k v$, for $u, v \in \mathcal{V}$. The *in-degree* of a node v in G_k is the number of edges directed into v, namely, in-degree$(v) = |\{(u, v) \in G_k\}|$. The following two facts are easy to establish, whose proofs are omitted from this abstract due to lack of space.

▶ **Lemma 1.** *A vector v belongs to $\mathcal{KDS}(\mathcal{V}, k)$ if and only if v has in-degree zero in G_k.*

▶ **Lemma 2.** *$\mathcal{KDS}(\mathcal{V}, k') \subseteq \mathcal{KDS}(\mathcal{V}, k)$, for $k' < k$.*

With these preliminaries, we can now prove the following bound for the size of the $\mathcal{KDS}$ in the worst-case. Due to lack of space, the proof is omitted from this abstract.

▶ **Theorem 3.** *The worst-case cardinality of $\mathcal{KDS}$ obeys the following bound:*

1. *Given any set of n vectors in d-space and any k with $k \leq (d+1)/2$, $|\mathcal{KDS}(\mathcal{V}, k)| \leq 1$.*
2. *For any $n \geq 1$, and k, d such that $k > (d+1)/2$, there* exists *a set $\mathcal{V}$ of n vectors in d-space for which $|\mathcal{KDS}(\mathcal{V}, k)| = n$.*

3 Average Case Analysis

We now analyze the size of the k-dominant skylines when the input vectors are drawn randomly from a distribution. Our analysis uses the standard "attribute distinctness and statistical independence" model [4, 6, 14], which only assumes that the vector components are distributed independently of each other, and for each component the magnitudes form a random permutation of $\{1, 2, \ldots, n\}$, namely, a total rank ordering. While not all data sets necessarily satisfy these assumptions, the model is sufficiently general, elegant and mathematically tractable for deriving non-trivial bounds.

We interpret the input set of n vectors as an $n \times d$ array, whose rows are the vectors and whose columns are permutations of $\{1, 2, \ldots, n\}$. The set of all possible permutations produces exactly $(n!)^d$ distinct vectors. We analyze the complexity of the k-dominant skyline for an input array $\mathcal{V}$ chosen uniformly at random from this set. What is the expected size of the k-dominant skyline for such an input $\mathcal{V}$?

We analyze the average size of $\mathcal{KDS}$ by setting up a recurrence. In order to aid that derivation, let $A(n, d, k)$ denote the average size of the k-dominant skyline for a set of n vectors in d dimensions, where the vectors are chosen under the random model described above. We assume, without loss of generality, that the first column of the input $n \times d$ array is sorted in the ascending order $(1, 2, \ldots, n)$—that is, the first coordinate of the ith vector is i, which if necessary can be realized by simply relabeling the vectors. See Figure 1 for illustration.

Let us focus on a single but arbitrary vector v, and derive the probability that it belongs to the k-dominant skyline. Suppose the vector v is represented as the ith row, which means its first coordinate is $v_1 = i$. We partition the remaining set of input vectors into two groups:

$$\mathcal{V}_a = \{u \in \mathcal{V} \mid u_1 < v_1\} \qquad \text{and} \qquad \mathcal{V}_b = \{u \in \mathcal{V} \mid u_1 > v_1\} \tag{1}$$

That is, $\mathcal{V}_a$ is the set of vectors that v dominates on the first coordinate, and $\mathcal{V}_b$ is the set of vectors that dominate v on the first coordinate. The key observation is that for v to be a $\mathcal{KDS}$ vector both of the following two events must occur:

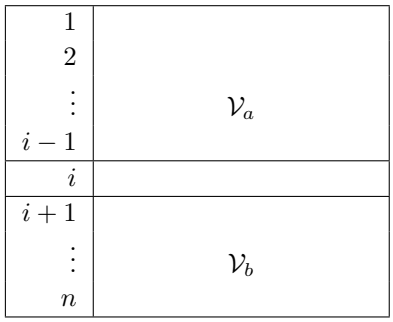

Figure 1 Average case analysis of $\mathcal{KDS}$. The vector v is the ith vector. The first $(i-1)$ vectors form the subset $\mathcal{V}_a$, and the last $(n-i)$ vectors form the subset $\mathcal{V}_b$.

1. None of the vectors in $\mathcal{V}_a$ k-dominates v among dimensions $\{2, 3, \ldots, d\}$, and
2. None of the vectors in $\mathcal{V}_b$ $(k-1)$-dominates v among dimensions $\{2, 3, \ldots, d\}$.

This follows because v can fail to be on the $\mathcal{KDS}$ only if either some vector in $\mathcal{V}_a$ or some vector in $\mathcal{V}_b$ k-dominates it. If some vector in $\mathcal{V}_a$ were to k-dominate v, it has to do so using all k dimensions from the set $\{2, 3, \ldots, d\}$ since v already dominates each vector of $\mathcal{V}_a$ on dimension 1. Condition (1) computes the probability that no $u \in \mathcal{V}_a$ k-dominates v. On the other hand, since each vector $u \in V_b$ already dominates v on the first coordinate, it needs to only find $k-1$ other dimensions among $\{2, 3, \ldots, d\}$ to achieve k-domination of v. Condition (2) computes the probability that no $u \in \mathcal{V}_b$ k-dominates v. The two events are independent because the remaining $d-1$ dimensions are independent of the first, and so the probability that v belongs to the $\mathcal{KDS}$ is the *product* of these two probabilities. The following two lemmas derive these probabilities.

▶ **Lemma 4.** *The probability of event (1) is* $\frac{A(i,d-1,k)}{i}$.

Proof. Consider the $i \times (d-1)$ array consisting of the first i rows and the last $(d-1)$ columns of $\mathcal{V}$. This is a random set of i vectors in $(d-1)$-dimensional space, which for convenience we call the *reduced space*. By induction, the expected $\mathcal{KDS}$ size for this set is $A(i, d-1, k)$. The probability that v is one of these skyline vectors is $A(i, d-1, k)/i$, by symmetry. Since v already dominates all the vectors of $\mathcal{V}_a$ in the first coordinate, it is on the $\mathcal{KDS}$ of $\mathcal{V}_a \cup \{v\}$ with the same probability. ◀

▶ **Lemma 5.** *The probability of event (2) is* $\frac{A(n-i+1,d-1,k-1)}{n-i+1}$.

Proof. The proof is similar to (1), and omitted from the abstract due to space. ◀

By combining the preceding two lemmas, we get the probability that v lies on the $\mathcal{KDS}$:

$$\Pr[v \in \mathcal{KDS}] \;=\; \frac{A(i, d-1, k)}{i} \times \frac{A(n-i+1, d-1, k-1)}{n-i+1}.$$

By summing over all n points, we get

$$A(n, d, k) \;=\; \sum_{i=1}^{n} \left(\frac{A(i, d-1, k)}{i} \times \frac{A(n-i+1, d-1, k-1)}{n-i+1} \right) \tag{2}$$

Since $\frac{A(i,d-1,k)}{i}$ is a probability in this recurrence, we can replace it with 1 and derive the following upper bound with a change of index:

$$A(n,d,k) \;\leq\; \sum_{i=1}^{n} \left(\frac{A(n-i+1,d-1,k-1)}{n-i+1} \right) \;\leq\; \sum_{i=1}^{n} \frac{A(i,d-1,k-1)}{i} \tag{3}$$

The function $A(n,d,k)$ is monotone non-decreasing with n because

$$A(n,d,k) \;=\; A(n-1,d,k) \;+\; \frac{A(n,d-1,k)}{n} \times \frac{A(1,d-1,k-1)}{1} \tag{4}$$

where the second term is non-negative. Therefore, we have the following upper bound:

$$A(n,d,k) \;\leq\; \sum_{i=1}^{n} \frac{A(i,d-1,k-1)}{i} \;\leq\; A(n,d-1,k-1) \times \sum_{i=1}^{n} \frac{1}{i} \tag{5}$$

Since the harmonic number $H_n = \sum_{i=1}^{n} \frac{1}{i} \approx \ln n$ [11], after j iterations, we get:

$$A(n,d,k) \;\leq\; A(n,d-j,k-j) \times (H_n)^j$$

The stopping condition for the recurrence is reached when $2(k-j)$ becomes less than or equal to $(d-j)+1$, at which point the skyline size drops to at most 1 by Theorem 3. We can show that the maximum number of iterations j is $j = 2k-(d+1)$, which leads to the following theorem.

▶ **Theorem 6.** *Let $\mathcal{V}$ be a set of n random points in d-dimensional space under the attribute distinctness and statistical independence model. Then, the expected cardinality of their k-dominant skyline is*

$$\begin{cases} O\left((\log n)^{2k-d-1}\right) & \text{for } k > \frac{d+1}{2} \\ O(1) & \text{otherwise} \end{cases}$$

Remarks

Theorem 6 smoothly interpolates between the two extreme cardinality bounds for $\mathcal{KDS}$ previously known, namely, $O(1)$ for $k \leq (d+1)/2$, and $O(\log^{d-1} n)$ for $k = d$, and the classical result of Bentley et al. [4] emerges as a special case of this theorem for $k = d$. In database systems, a constructive way to utilize this result could be this: by reducing the value of k by one, we can expect the $\mathcal{KDS}$ to shrink by a significant fraction, namely, a factor of $O(\log^2 n)$. A query engine can therefore tune the parameter k to predictably control the (expected) number of skyline vectors that appear on $\mathcal{KDS}$.

4 Computing KDS in Subquadratic Time

Unlike full-dimensional dominance, the k-dominance relation is *not transitive*: that is, $a \succ_k b$ and $b \succ_k c$ does not guarantee $a \succ_k c$. The failure of transitivity means that the algorithms for traditional skylines [3, 15] cannot be used for computing $\mathcal{KDS}$. In fact, to the best of our knowledge, no subquadratic time algorithm is known for k-dominant skylines even for a constant dimension.

Let $I_k \subset \{1,2,\ldots,d\}$ be an index set of size k, namely, $|I_k| = k$. There are $\binom{d}{k}$ size k distinct index sets, and each such set defines a subset of k dimensions. The projection of the

input set of points $\mathcal{V}$ along any particular set I_k is called a *k-projection* of $\mathcal{V}$. A k-projection is a set of k-dimensional points, and we refer to its skyline as the skyline of the k-projection. Then, the following simple observation is the key to our algorithm.

▶ **Lemma 7.** *A point $v \in \mathcal{V}$ belongs to $\mathcal{KDS}(\mathcal{V}, k)$ if and only if v belongs to the skylines of all distinct k-projections of $\mathcal{V}$.*

We can, therefore, compute $\mathcal{KDS}(\mathcal{V}, k)$ by computing the skylines of all distinct k-projections of $\mathcal{V}$ and taking their common intersection.

▶ **Theorem 8.** *Let $\mathcal{V}$ be a set of n points in d dimensions, where $d = O(\log n / \log\log n)$. The k-dominant skyline $\mathcal{KDS}(\mathcal{V}, k)$ is precisely the common intersection of the skylines of all possible k-projections of $\mathcal{V}$, and it can be computed in worst-case time $O(n \log^{d-1} n)$.*

Proof. The number of distinct k-projections of $\mathcal{V}$ is $\binom{d}{k} = \binom{d}{d-k}$. We can compute the skyline of each of these projections in time $O(n \log^{k-1} n)$ using the divide-and-conquer algorithm of [3]. The size of each of these skylines is at most n, and therefore we can compute their common intersection in $O(nd)$ time. The total running time of the algorithm is therefore $O\left(\binom{d}{d-k}\, n \log^{k-1} n\right)$. Since $\binom{d}{d-k} \leq (\frac{ed}{d-k})^{d-k} = O(d^{d-k})$, we can upper bound the running time as $O\left(d^{d-k} n \log^{k-1} n\right)$, which is sub-quadratic as long as $d \leq c \log n / \log\log n$, for an appropriate constant c. ◀

Finding a worst-case subquadratic algorithm in dimensions higher than $\Theta(\log n)$ remains a challenging open problem, even for conventional skylines.

5 Experimental Evaluation

Our theoretical bounds on the expected size of the $\mathcal{KDS}$ are predicated on certain properties of random data (attribute distinctness and statistical independence), which may or may not be satisfied by real-world datasets. In our experiments, we chose a number of diverse data sets to evaluate how their $\mathcal{KDS}$ size behaves in practice. We also generated synthetic data sets, following our theoretical distribution, to establish a baseline. (Since time complexity was not an important concern, we implemented a simple algorithm, which constructs an $O(dn)$ space data structure in $O(dn^2)$ time from which the $\mathcal{KDS}$ for any value of $k \leq d$ can be extracted in $O(n)$ time explicitly.)

Data Sets

The synthetic data sets are produced by generating random permutations with $n \approx 10^5$ and $d = 20$ or 30. For the real-world data, we use a number of popular high-dimensional data sets, as shown in Figure 2. The NBA basketball data is available at [2], the Movie Lens data at [23, 16], and all the remaining datasets are available from the UCI repository [12]. They vary in size from a few thousand points to more than hundred thousand points in dimensions ranging from 10 to several thousand. Altogether these data sets allow us to evaluate the behavior of $\mathcal{KDS}$ under highly diverse conditions. In addition, for many of these sets, nearly all the input vectors show up on the full-dimensional skyline, providing a useful test for the utility of the $\mathcal{KDS}$. In all the experimental plots, the x-axis is the value of k and y-axis the size of $\mathcal{KDS}$.

Data	Size n	Dim d	Full Skyline
NBA Career	4051	17	80
NBA Season	21961	17	1466
Movie Lens 1	1682	943	1330
Movie Lens 2	3952	6040	3475
Wine Quality	4898	12	3094
Human Activity	7352	561	7352
Internet Ads	3279	1558	1976
Particle Signal	130065	50	129596

Figure 2 Data sets used in experiments and the size of their d-dimensional skyline.

5.1 KDS Size for Synthetic Data

We generated a number of random data sets in moderate dimensions $d \leq 30$, which proved sufficient to show the exponential growth rate of the k-dominant skylines. Figure 3 shows the results for $n = 100K$, with $d = 20$ and $d = 30$; the plots for other values of n and d were found to be similar. (When the plots flatten out near the end, it means that the $\mathcal{KDS}$ size has reached the total number of input vectors.)

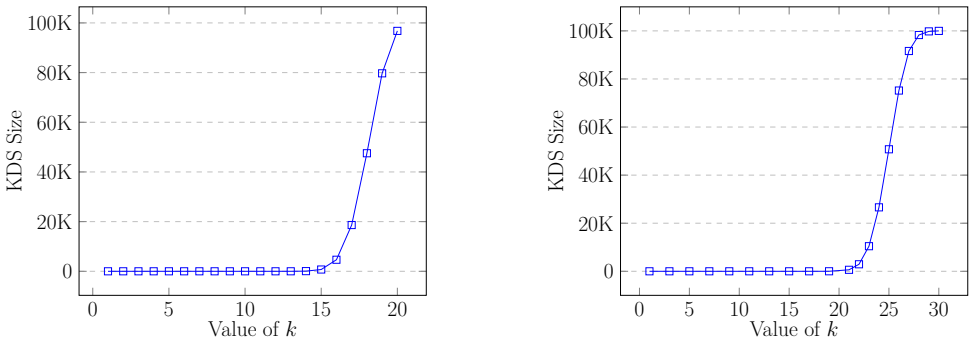

Figure 3 $\mathcal{KDS}$ size scaling for $n = 10^5$ random vectors in 20 (left) and 30 (right) dimensions.

5.2 KDS Size for Real-World Data

The NBA data sets [2] include scores in 17 different categories for 4051 basketball players. A higher score indicates better performance in each skill (dimension). The NBA-Career set has data aggregated over each player's entire career, while the NBA-Season set has a separate vector for each season in which a player was active (a total of 21961 vectors). Figure 4 shows the $\mathcal{KDS}$ size as a function of increasing k, for both NBA data sets, confirming a clear exponential rate of growth.

Figure 5 shows the results for the Movie Lens data [23, 16], in which each entry is a movie rating. We used first $100,000$ and first $1,000,000$ ratings to generate two different data sets. The former (Movie Lens 1) has $n = 1682$ distinct movies with $d = 943$ distinct reviewers, and the latter (Movie Lens 2) has $n = 3952$ movies with $d = 6040$ reviewers. (Each movie is rated on the scale of 1 (worst) to 5 (best), and we use the default value of 3 (average) for all blank entries.)

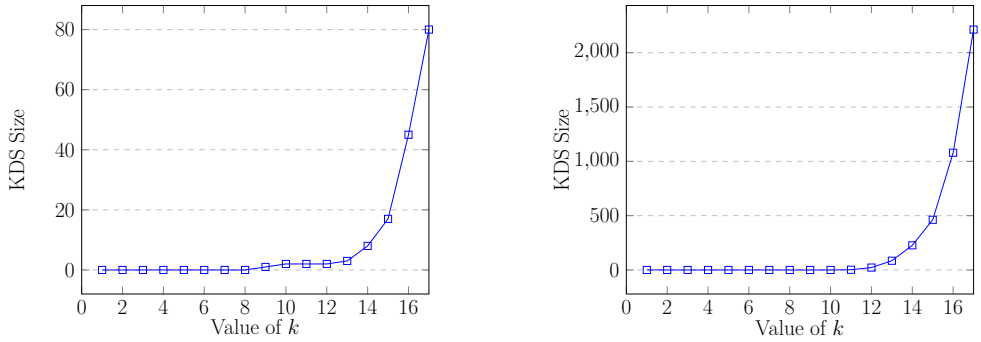

Figure 4 Results for the NBA player data sets, NBA-Career (left) and NBA-Season (right).

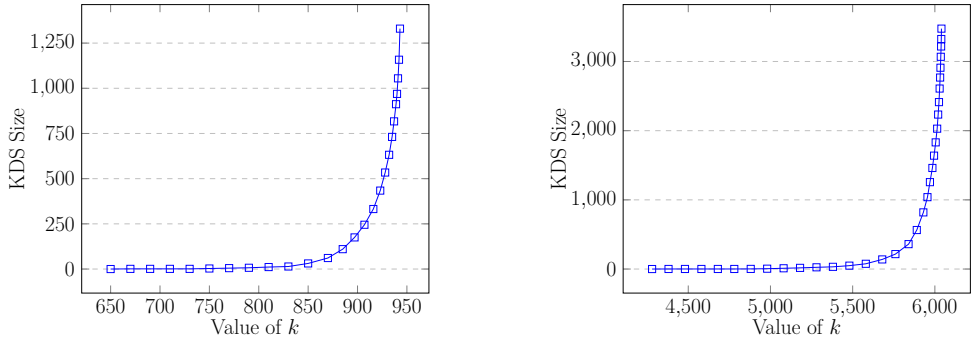

Figure 5 Results for Movie Lens 1 (left) and Movie Lens 2 (right) data sets.

Finally, the following figure shows the results for the remaining four data sets: wine quality, human activity, Internet ads, and particle signal.

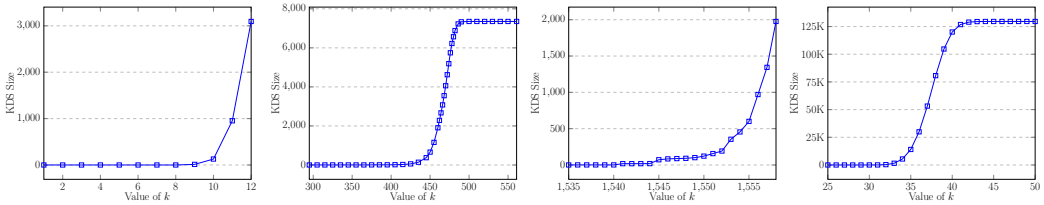

Figure 6 Results for wine quality, human activity, Internet Ads, and particle signal data sets.

In summary, across a diverse collection of data sets, the k-dominant skylines follow the exponential curve shown by our average case analysis, suggesting that the model of random data with attribute distinctness and statistical independence is potentially useful in practice.

6 Application of KDS in Ranking of High-Dimensional Vectors

In multi-criteria decision problems, there is typically no single best answer and often too many incomparable answers are possible. For instance, in the NBA basketball data, almost 1500 players are *undominated*, and thus arguably the best players. One approach, exemplified by the top-k operator in databases, is to define a *utility function* that aggregates the user preferences across multiple dimensions, for instance, as a linear combination. Formulating an appropriate utility function, however, is quite challenging because it requires

users to accurately quantify their preferences, and also requires different dimensions to be similar in scale. (In Section 6.3, we also compare simple aggregation-type rankings with the $\mathcal{KDS}$-based ranking.) The alternative approach of *skylines* uses the easier-to-apply principle of pareto optimality—no rational user prefers a solution that is dominated by another on all dimensions—but is stymied by the explosion of skyline size in higher dimensions.

The $\mathcal{KDS}$ suggests a natural approach for *pruning* the skyline using the control knob of parameter k, based on the following insight: many vectors may be undominated in high dimensions, but some are undominated only because they score highly in one or few dimensions, while others are undominated because they score highly in most of the dimensions. Intuitively, the second kind are the more significant ones. The $\mathcal{KDS}$ based approach automatically selects the vectors that remain undominated on most dimensions, and thus appear on the *early* skyline when k is small. Using the asymptotic expected growth rate of $\mathcal{KDS}$ (cf. Theorem 6), a user can *control* the skyline to a desired size with parameter k.

In order to test this hypothesis, we carried out the following experiment. Suppose we consider a *random subset of dimensions* for the data, compute the $\mathcal{KDS}$ of the reduced set, and repeat this trial multiple times. How similar are the $\mathcal{KDS}$ in these trials? Intuitively, if $\mathcal{KDS}$ vectors lacked any intrinsic significance, then these skylines should not have much in common. On the other hand, if $\mathcal{KDS}$ vectors were fundamentally dominant, then they would persist in many subsets of dimensions, and show a high Jaccard index of similarity. We chose the Movie Lens and NBA data sets for these experiments because their "top vectors" are familiar names, and easy to interpret.

6.1 Top Movies and NBA Players

For the movies, we use the Movie Lens 2 data set, and drop about 10% of the dimensions at random, obtaining a set with 5426 dimensions out of the original 6040. We chose $k = 4800$, for which we are assured to get a small $\mathcal{KDS}$ size, based on Figure 5. We repeated this experiment 5 times, producing 5 different k-dominant skyline sets. The resulting skylines had size between 27 and 29. We then counted how many times each $\mathcal{KDS}$ vector (movie) appeared among these 5 skylines. Figure 7 shows the result: remarkably, the top 27 movies are common to all five $\mathcal{KDS}$ sets. Arguably, all of these movies have claims to be considered "top fan favorites," demonstrating the consistency and utility of $\mathcal{KDS}$ as an automatic data exploration tool.

For the basketball data, we use the NBA-Season data where we randomly dropped 2 of the 17 dimensions, and chose $k = 12$. We performed 5 trials of this experiment, and counted how many times each $\mathcal{KDS}$ vector appears among these 5 skylines. Figure 8 shows the results. Once again, most of the names found by the algorithm are all-time greats.

6.2 Jaccard Similarity

Among the five random trials of the Movie Lens data, the smallest $\mathcal{KDS}$ had 27 vectors, and the largest one had 29 vectors. To quantify the pairwise similarity among these 5 sets, we use the Jaccard similarity measure, which is defined as $\frac{\|A \cap B\|}{\|A \cup B\|}$, for two sets A and B. The Jaccard index of 1 indicates a perfect match. As the following table shows, the similarity index is between 0.93 and 1 in all cases. Among the five random trials of the NBA-Season data, the smallest $\mathcal{KDS}$ for $k = 12$ had 49 vectors, and the largest one had 68 vectors. The table below shows the Jaccard index of similarity among these 5 sets.

Movie	Occur
Toy Story	5
The Usual Suspects	5
Braveheart	5
Star Wars: Ep. IV	5
Pulp Fiction	5
Shawshank Redemption	5
Forrest Gump	5
Schindler's List	5
Terminator 2	5
Silence of the Lambs	5
Fargo	5
The Godfather	5
Casablanca	5
One Flew Over Cuckoo's Nest	5
Star Wars: Episode V	5
The Princess Bride	5
Raiders of the Lost Ark	5
Groundhog Day	5
Back to the Future	5
L.A. Confidential	5
Saving Private Ryan	5
Shakespeare in Love	5
The Matrix	5
The Sixth Sense	5
American Beauty	5
Being John Malkovich	5
Gladiator	5

Figure 7 Occurrence frequency of top movies in 5 random trials of $\mathcal{KDS}$

NBA Player	Occur
Wilt Chamberlain 1961	5
Artis Gilmore 1974	5
Bob Mcadoo 1974	5
George Mcginnis 1974	5
K. Abdul-jabbar 1975	5
Julius Erving 1975	5
Artis Gilmore 1975	5
Moses Malone 1978	5
Michael Jordan 1984	5
Michael Jordan 1986	5
Charles Barkley 1987	5
Michael Jordan 1987	5
Michael Jordan 1988	5
Karl Malone 1988	5
Hakeem Olajuwon 1988	5
Karl Malone 1989	5
David Robinson 1990	5
Hakeem Olajuwon 1992	5
Shaquille O'neal 1993	5
David Robinson 1993	5
Dwight Howard 2007	5
Allen Iverson 2007	5
LeBron James 2007	5
Al Jefferson 2007	5
Amare Stoudemire 2007	5
Dwight Howard 2008	5
Dwyane Wade 2008	5
Kevin Durant 2009	5
Dwight Howard 2009	5
David Lee 2009	5
Charles Barkley 1987	4
Karl Malone 1988	4

Figure 8 Occurrence frequency of top NBA players in 5 random trials

	1	2	3	4	5
1	1	0.931	0.966	0.966	0.931
2	0.931	1	0.964	0.964	1
3	0.966	0.964	1	1	0.964
4	0.966	0.964	1	1	0.964
5	0.931	1	0.964	0.964	1

Figure 9 Jaccard Similarity for top movies

	1	2	3	4	5
1	1	0.603	0.716	0.638	0.776
2	0.603	1	0.459	0.493	0.568
3	0.716	0.459	1	0.718	0.658
4	0.638	0.493	0.718	1	0.585
5	0.776	0.568	0.658	0.585	1

Figure 10 Jaccard Similarity for top NBA players

6.3 KDS vs. Aggregation-based Ranking

As a point of comparison, we now discuss some pitfalls suffered by alternative ranking methods based on simple utility functions. We use the movie database because the results are easier to interpret. We recall that in this dataset each vector represents a movie, with each dimension being one user's ratings on a 1–5 scale. The movie data, however, is incomplete since not all users rate all movies, and so we use the following two (natural) scoring functions to compare different vectors.

1. the weighted average of each movie's ratings, and
2. the raw sum of each movie's ratings.

We observe that, as expected, the first method tends to highly rank those movies that have high scores but *very few ratings*. In fact, none of the movies ranked in the top 10 would be considered popular—each had an average score of 5 but rated by at most five users! Our second method corrects for this bias, and steers the ranking towards more popular movies by summing the scores of all the users for each movie. However, this method leads to movies that are widely known but not necessarily top rated candidates for many users. For instance, the original 3 Star Wars movies appeared in the top 4, however the lowest had an average rating of roughly 4/5, which is significantly lower than that of many movies appearing much later in the list. The NBA data set reveals another problem with aggregation-based ranking: different dimensions have vastly different scales, making it difficult to combine them into a single meaningful utility function.

By comparison, as the preceding experiments have shown, the $\mathcal{KDS}$-based ranking gives sensible and robust results without the need for a domain-specific and difficult to formulate utility function. When the full skyline has far too many points, the tunable parameter k of the $\mathcal{KDS}$ automatically acts as a proxy for the importance of skyline vectors: the earlier a point appears on $\mathcal{KDS}$ the more significant it is.

7 Concluding Remarks

In this paper, we made both theoretical and applied contributions to the study of k-dominance in multidimensional data. On the theoretical front, we derived an upper bound on the average case complexity of $\mathcal{KDS}$, which generalizes a classical result of Bentley et al. [4]. We also showed that while k-dominance does not satisfy transitivity, one can still compute the $\mathcal{KDS}$ in roughly the same time as the conventional skyline (worst-case sub-quadratic) as long as the dimension is $d = O(\log n / \log\log n)$. Our experiments show that the size of $\mathcal{KDS}$ in many real-world multi-dimensional data sets follows the same exponential trend of our average case analysis, suggesting that our theoretical bounds can be helpful predictors in practice. Finally, we validate the usefulness of k-dominant skylines as a tool for selecting a small set of top-ranked vectors when the full skyline contains far too many.

References

1 P. Agarwal, N. Kumar, S. Sintos, and S. Suri. Efficient algorithms for k-regret minimizing sets. In *Proc. of 16th Int. Symp. on Experimental Algorithms*, 2017, to appear.

2 Basketball Data. http://databasebasketball.com/.

3 J. L. Bentley. Multidimensional Divide and Conquer. *Communications of the ACM*, 23(4):214–229, 1980.

4 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. *Journal of the ACM*, 25(4):536–543, 1978.

5 S. Börzsönyi, D. Kossmann, and K. Stocker. The skyline operator. In *Proceedings of the 17th International Conference on Data Engineering*, pages 421–430, 2001.

6 C. Buchta. On the average number of maxima in a set of vectors. *Information Processing Letters*, 33(2):63 – 65, 1989.

7 C. Y. Chan, H. V. Jagadish, K. L. Tan, A. K. H. Tung, and Z. Zhang. Finding K-dominant Skylines in High Dimensional Space. In *Proceedings of the ACM SIGMOD International Conference on Management*, pages 503–514, 2006.

8 C. Y. Chan, H. V. Jagadish, K. L. Tan, A. K. H. Tung, and Z. Zhang. On High Dimensional Skylines. In *Proc. 10th International Conference on Extending Database Technology (EDBT)*, pages 478–495, 2006.

9 S. Chester and I. Assent. Explanations for Skyline Query Results. In *Proc. International Conference on Extending Database Technology (EDBT)*, pages 349–360, 2015.

10 S. Chester, A. Thomo, S. Venkatesh, and S. Whitesides. Computing k-regret Minimizing Sets. *PVLDB*, 7(5):389–400, 2014.

11 T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, 2nd edition*. MIT Press, McGraw-Hill Book Company, 2000.

12 U. Data Repository. http://archive.ics.uci.edu/ml/.

13 M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition, 2008.

14 P. Godfrey. Skyline cardinality for relational processing. In *Proc. Foundations of Information and Knowledge Systems (FoIKS)*, pages 78–97, 2004.

15 P. Godfrey, R. Shipley, and J. Gryz. Algorithms and analyses for maximal vector computation. *VLDB J.*, 16(1):5–28, 2007.

16 F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):1–19, 2015.

17 H. K. Hwang, T. H. Tsai, and W. M. Chen. Threshold phenomena in k-dominant skylines of random samples. *SIAM Journal on Computing*, 42(2):405–441, 2013.

18 V. Koltun and C. H. Papadimitriou. Approximately Dominating Representatives. *Theoretical Computer Science*, 371(3):148–154, 2007.

19 D. Kossmann, F. Ramsak, 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)*, pages 275–286, 2002.

20 H. T. Kung, F. L., and F. P. Preparata. On Finding the Maxima of a Set of Vectors. *Journal of the ACM*, 22(4):469–476, 1975.

21 Y. Lu, J. Zhao, L. Chen, B. Cui, and D. Yang. Effective skyline cardinality estimation on data streams. In *19th International Conference on Database and Expert Systems Applications*, pages 241–254, 2008.

22 J. Matousek. Computing dominances in E^n. *Information Processing Letters*, 38(5):277–278, 1991.

23 Movie Lens Data. http://grouplens.org/datasets/movielens/.

24 D. Nanongkai, A. D. Sarma, A. Lall, R. J. Lipton, and J. Xu. Regret-Minimizing Representative Databases. *PVLDB*, 3(1):1114–1124, 2010.

25 D. Papadias, Y. Tao, G. Fu, and B. Seeger. An optimal and progressive algorithm for skyline queries. In *Proc. ACM SIGMOD*, pages 467–478, 2003.

26 P. Peng and R. C. Wong. Geometry approach for k-regret query. In *Proc. International Conference on Data Engineering (ICDE)*, pages 772–783, 2014.

27 F. P. Preparata and M. I. Shamos. *Computational Geometry—An Introduction*. Springer, 1985.

28 E. Tiakas, A. N. Papadopoulos, and Y. Manolopoulos. On estimating the maximum domination value and the skyline cardinality of multi-dimensional data sets. *Int. J. Knowledge-Based Organ.*, 3(4):61–83, 2013.