Proceedings of the 16th International Conference on Extending Database Technology (EDBT), Genoa, Italy, 2013

SkyDiver: A Framework for Skyline Diversification

G. Valkanas
Dept. of Informatics and Telecommunications
University of Athens
Athens, Greece
gvalk@di.uoa.gr

A. N. Papadopoulos
Dept. of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
papadopo@csd.auth.gr

D. Gunopulos
Dept. of Informatics and Telecommunications
University of Athens
Athens, Greece
dg@di.uoa.gr

ABSTRACT

Skyline queries have attracted considerable attention by the database community during the last decade, due to their applicability in a series of domains. However, most existing works tackle the problem from an efficiency standpoint, i.e., returning the skyline as quickly as possible. The user is then presented with the entire skyline set, which may be in several cases overwhelming, therefore requiring manual inspection to come up with the most informative data points. To overcome this shortcoming, we propose a novel approach in selecting the k most *diverse* skyline points, i.e., the ones that best capture the different aspects of both the skyline *and* the dataset they belong to. We present a novel formulation of *diversification* which, in contrast to previous proposals, is intuitive, because it is based solely on the domination relationships among points. Consequently, additional artificial distance measures (e.g., L_p norms) among skyline points are not required. We present efficient approaches in solving this problem and demonstrate the efficiency and effectiveness of our approach through an extensive experimental evaluation with both real-life and synthetic data sets.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining ; H.3.3 [**Information Search and Retrieval**]: Selection process

General Terms

Algorithms

Keywords

Skyline, Diversity, MinHashing, Approximation.

1. INTRODUCTION

Skyline queries, in the context of databases, were initially proposed in [4] and since then, they have attracted considerable attention by the database and data analysis community, as they perform multi-objective optimization without the requirement for user-defined scoring functions. The only input required by the user is the *preferences* regarding the minimization / maximization of attribute values. For example, if *price* and *quality* are two of the attributes, then users prefer to minimize price and maximize quality, selecting items which are (objectively) better than (i.e., *dominate*) others.

Despite the research attention that this field has attracted, most of the efforts focus on the efficiency perspective of the problem, i.e., how to retrieve the skyline as quickly as possible (minimizing I/O and CPU time). However, depending on the data distribution and dimensionality, it is very likely that the skyline will contain a significantly high number of points for the user to inspect manually. More formally, the expected skyline cardinality of a set of n randomly generated points in d dimensions is $m = O((\ln n)^{d-1})$ [3]. In a data set containing 10^9 points, having about 10^3 skyline points may not be that much compared to the data set cardinality, but it is impractical for the user to inspect manually.

To overcome the skyline cardinality explosion problem, two main directions have been followed, both of which focus on the selection of a fixed-size subset of k skyline points. The first alternative considers the entire dataset, selecting a set of k *skyline representatives*, which collectively dominate as many distinct points as possible [21]. The second alternative considers the skyline set alone, and selects k skyline points that best describe the skyline contour. The state-of-the-art techniques include [38, 32] and rely solely on the L_p distance.

In this research, we also address the skyline cardinality explosion problem, and propose the SkyDiver (Skyline Diversification) framework, which outputs efficiently a subset of k skyline points with high diversity. Our measure of diversity is defined in a meaningful and intuitive way, based on the most fundamental skyline concept: the *dominance* relation. In particular, each skyline point is related to its *dominated set* $\Gamma(p)$, i.e., the set of points that it dominates, which is an established approach for dominance-based ranking [21, 36], thereby making it suitable as a building block for our diversification model. More specifically, the diversity between two skyline points p and q is defined as the *Jaccard distance* J_d of their corresponding dominated sets, i.e., $J_d(p,q) = 1 - \frac{|\Gamma(p) \cap \Gamma(q)|}{|\Gamma(p) \cup \Gamma(q)|}$. When $\Gamma(p)$ and $\Gamma(q)$ largely overlap, the diversity score will be small; conversely, sharing few dominated points results in high diversity. Finally, our diversification model inherently encourages large domination sets, because for a fixed number of commonly dominated points, the selected pair will be the one that collectively maximizes the domination score. The choice of J_d arises naturally, taking

EDBT/ICDT '13, March 18 - 22 2013, Genoa, Italy.

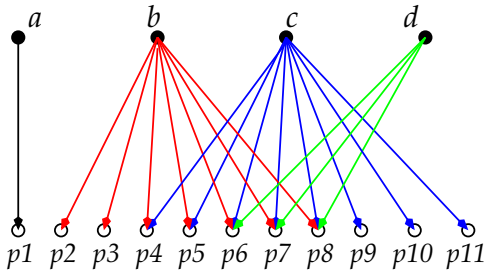


Figure 1: Graph with dominance relations.

into consideration that it is the most widely used similarity measure for sets.

The need for diversification arises in any context where there are users with varying tastes, e.g., web search [1, 2], or (manual) post-processing is necessary to fulfill their needs, e.g., exploratory search [8] and data analysis [25, 31]. The skyline setting is no exception, considering that some users may be looking for a "cheap" buy, whereas others for a "quality" one. Moreover, there may be some users who are interested in having an investigative look before proceeding with their purchase. Without knowing the user's true interests, our best bet would be to diversify the skyline result, to fulfill the needs of as many users as possible.

To motivate our approach further, consider Figure 1, where a set of points has been split into its skyline (left) and the set of dominated points (right), and an edge signifies that the skyline point dominates the corresponding point on the right. This representation abstracts a multitude of domains: *i*) nodes are product reviews and an edge exists when a product is at least as good as another, *ii*) nodes are web pages and a node dominates another if it contains at least as much information on a topic of interest, *iii*) nodes are web search results and an edge exists if a user selected one result over the others. The selected document becomes part of the skyline, whereas the rest belong to the dominated set.

Note that the entire representation only relies on the *dominance* relation because this may be all we have. For instance, in our third example, we only know that a user preferred some documents over the rest, without explicitly knowing "why". Similarly, the data may belong to a 3rd party who has anonymized or obfuscated it and we are only presented with this dominance graph, but not the actual data values. This practically translates into an inability to build and use a multi-dimensional index.

A max-coverage approach with $k = 2$ would return the pair (b, c). However, their domination sets largely overlap, meaning that little *new* information will be provided. Similarly, d discusses topics already covered by both b and c. On the contrary, a may provide truly fresh information that none other does, despite the fact that it dominates a single point. Our proposed approach would return the pair (c, a): c dominates the most points and addresses a lot of the information found in b and d; a provides truly new information compared with c, and will attract users with a varied taste better than any other combination.

Overall, our contributions are briefly described as follows:

- We define a novel and intuitive measure of skyline diversity, which is solely based on the dominance property. In particular, the diversity between skyline points is computed as the Jaccard distance of their associated dominated sets, making our technique suitable for settings, such as partially-ordered domains or data with categorical feature. Therefore, we advocate this measure as an intuitive approach for *(dis)similarity* computation between skyline points.
- Given our similarity measure, the problem of k-most diverse skyline points is mapped naturally to the k-dispersion problem. Since k-dispersion is NP-hard [19], we apply a greedy-based heuristic that offers a 2-approximation with respect to the optimal solution.
- We propose the index-independent SkyDiver framework, to efficiently approximate the optimal solution, employing *MinHash* signatures and we provide theoretical guarantees for the effectiveness of our approach. Alternatively, *Locality Sensitive Hashing* (LSH) can be used, as a space-efficient approximation to the MinHashing.
- We provide an extensive and comprehensive experimental evaluation of SkyDiver, using both real-life and synthetic datasets to verify our theoretical study.

The rest of the work is organized as follows. Related work is summarized in Section 2. Section 3 presents some fundamental concepts with respect to the problem and our solution. Our algorithms are studied in detail in Section 4 whereas Section 5 discusses performance evaluation results based on real-life and synthetic data sets. Finally, Section 6 summarizes our work and discusses future work briefly.

2. RELATED WORK

Diversity is a topic studied in different disciplines. Since the literature is very rich, we followingly discuss briefly the most basic contributions that are closely related to ours.

Operations Research. Diversification in Operations Research has been used as a means of dispersion in optimization problems. In particular, this concept has applications in *facility location*, where the locations of k new stores or warehouses must be determined in order to be convenient to deliver products to clients. The intractability of the problem was first investigated in [19], where it was shown to be NP-Hard. Some heuristic-based algorithms are studied in [14], whereas [28] discusses problem variations in detail. The authors also show that a 2-approximation is the best that we can get when the distance measure respects the triangular inequality. In [16] the authors provide a uniform treatment of the different dispersion problems and experiment with randomized heuristics. Finally, we note the work of [26] which studies upper bounds and exact algorithms for dispersion problems.

Information Retrieval. It has become evident in the IR community [5], that the returned results of a query should provide some sort of diversification, thereby covering various tastes or resolving query ambiguity [30], even after taking personalization factors into account. Such items can also serve as a good basis for further query refinement or exploratory search [8]. In [1] a systematic approach to diversifying results is presented, aiming to minimize the risk of dissatisfaction of the average user. The problem is shown to be NP-hard, thus, approximation algorithms are used to solve it efficiently. Finally, [2] investigates new scoring functions that take into consideration both the relevance and the diversity of the result set.

Database Management. The diversification problem has been also addressed by the database community. Diversification of XML results is studied in [22]. In [17], diversification is studied for points in Euclidean space and access methods are used. In [34], the DivDB system is developed, which provides result diversification by using an SQL interface, whereas [13] studied the dynamic case of the problem.

Our research contributes to the data management discipline, by investigating diversification issues in the result of a skyline query[4], which is widely used to reveal the best items based on maximization or minimization preferences. Among the different algorithms proposed in the literature for skyline query processing, BBS [24] is the most preferred, because it has two significant properties, namely result progressiveness and I/O optimality. However, in cases where indexing cannot be applied, one must resort to other alternatives. Two efficient algorithms for skyline computation without the use of an index are proposed in [11] and [29]. The first one is designed for the streaming case and performs multiple passes over the data returning approximate results. In contrast, the second one is designed for the I/O model and always provides correct results. In the sequel, we take a closer look to existing techniques that are mostly related to our work.

Skyline Diversity. Representing the skyline contour [32] has been suggested as an alternative for skyline diversification [38]. Both techniques use an L_p norm (Euclidean distance in particular) as the measure of diversity between skyline points. This choice may be problematic in the following cases: *i*) the dimensions correspond to attributes that are difficult to combine (e.g., price and quality), *ii*) the skyline is computed over a partially-ordered domain [37], and *iii*) the points' attributes are non-numerical values, e.g., when operating over a document collection where the attributes may be terms, q-grams or topics. Under such circumstances, a multidimensional index can not be used, rendering the techniques infeasible or even inapplicable. Additionally, the Euclidean distance is sensitive to dimension scaling, meaning that a weighted distance measure might be more appropriate. Therefore, by selecting an off-the-shelf distance measure, the scale independence property of skylines is disregarded. More notable is the fact that only the skyline set $\mathcal{S}$ is used to determine a solution, disregarding the rest of the points. Note that this is in contrast with existing literature that accepts dominance power, i.e., $|\Gamma(p)|$, as a predominant quality characteristic of a skyline point [24, 36].

Coverage-based techniques. The techniques in [21, 15, 10] also consider the problem of reducing the skyline size and suggest to select a subset of k skyline points according to a maximum coverage criterion. In particular, the optimization goal is to maximize the number of distinct non-skyline points dominated by at least one of the k selected skyline points. Despite its set-oriented nature, this technique essentially solves a *different* problem, aiming to maximize the dominated set of the selected skyline points, and not to diversify them in any way. Note that such a solution would have been highly attractive in conjunction with a greedy heuristic, as shown by the following lemma.

Lemma 1. *([7]) The greedy algorithm on a set-cover problem with finite VC-dimension v, yields an approximation ratio of $O(v \log vc)$, where c is the optimal solution.*

The set system of such a max-coverage instance has a finite VC-dimension [33] of d (d being the dimensionality of the problem) due to the axis-aligned hyper-rectangles of dominating regions, anchored to the upper right corner of the d-dimensional space [23]. From Lemma 1 and a reduction of max-coverage to set-cover, we can also expect a better approximation than the $1 - 1/e$ of the general case. To the best of our knowledge, such a remark has been largely overlooked in the skyline literature.

Table 1: k-max-coverage vs k-dispersion

		k-max-coverage		k-dispersion	
data	k	coverage	diversity	coverage	diversity
IND5M4D	2	98.4%	0.064	95.5%	1.000
	10	99.9%	0.064	95.8%	0.916
	50	100%	0.018	98.3%	0.553
FC5D	2	93.7%	0.304	88.6%	1.000
	10	98.9%	0.088	88.9%	0.941
	50	99.8%	0.032	93.2%	0.714
REC5D	2	70%	0.634	56.2%	1.000
	10	93.1%	0.328	56.7%	0.997
	50	98.6%	0.142	68.6%	0.864

To illustrate the difference between our objective and the one in coverage-based techniques, we have performed the following experiment: We computed the diversity and coverage scores, both by a k-dispersion and a k-max-coverage algorithm, for various data sets (see Section 5 for details). Table 1 contains the results of this experiment. We draw the following conclusions: *i)* Clearly, we can not solve the diversity problem through coverage. Coverage selects points with high overlap in their dominating regions, which sharply reduces diversity. *ii)* When the objective is diversity, coverage is not as high as when aiming for coverage per se, but it is still high enough. This was expected, since the diversity measure tends to select points that cover a good portion of the dataset from their own viewpoint.

The SkyDiver framework, proposed in this paper, bases diversification on the concept of domination and thus, it does not depend on additional distance functions, in contrast to existing techniques. Moreover, SkyDiver is index-independent, in the sense that given the skyline set, diversification is provided by performing a single pass over the data. Therefore, even if there exist categorical attributes, skyline diversification can still be performed. Finally, we note that, in contrast to previously proposed techniques, our approach considers not only the skyline set, but also the dominated subsets in order to facilitate diversification.

3. BASIC CONCEPTS

In this section, we present some basic concepts and definitions regarding the focus of our research, in order to keep the work self-contained. Table 2 depicts the basic notations that are used frequently.

3.1 Problem Definition

Let $\mathcal{D}$ be a d-dimensional dataset, where w.l.o.g. smaller values are preferred.[1] We say that $p = (p.x_1, ..., p.x_d) \in \mathcal{D}$ *dominates* $q = (q.x_1, ..., q.x_d) \in \mathcal{D}$ (and write $p \prec q$), when: $\forall i \in \{1, ..., d\}, p.x_i \leq q.x_i \wedge \exists j \in \{1, ..., d\} : p.x_j < q.x_j$. The skyline $\mathcal{S}$ of $\mathcal{D}$, is composed of all points in $\mathcal{D}$ that are not dominated by any other point.

Given a data set $\mathcal{D}$, the skyline set $\mathcal{S}$ and an integer k, $k \geq$

[1]We focus on numerical attributes for ease of presentation. Our approach applies to categorical ones equally well.

Table 2: Frequently used symbols

Symbol	Description
$\mathcal{D}$, $n = \|\mathcal{D}\|$	the data set and its cardinality
$\mathcal{S}$, $m = \|\mathcal{S}\|$	the skyline set and its cardinality
s_j	the j-th skyline point
k	number of diverse skyline points
t	size of each signature
M, $\widehat{M}$	domination and signature matrix
ξ	LSH similarity threshold
ζ	number of zones for LSH
$\Gamma(p)$	set of points dominated by p
$J_s(p,q)$	Jaccard similarity between p, q
$J_d(p,q)$	Jaccard distance between p, q
$\widehat{J_d}(p,q)$	Jaccard distance for signatures

2, the goal of the diversification process is to return a subset $S_k \subseteq S$ containing k skyline points, aiming to maximize their diversity, i.e., the *dissimilarities* among the skyline points. To quantify the diversity between two skyline points we need a distance function $d : \mathcal{S} \times \mathcal{S} \to \Re$.

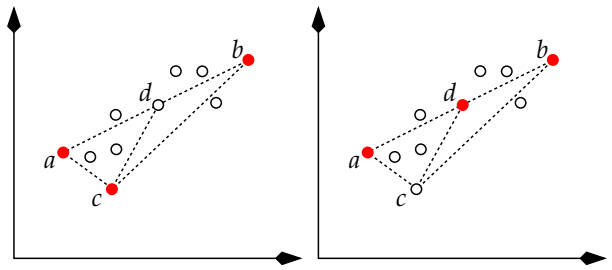
To overcome the limitations discussed in Section 2, we propose to use the *Jaccard distance* for diversity computation. With each skyline point p there is an associated subset of $\mathcal{D}$ containing all points dominated by p, denoted as $\Gamma(p)$. The *domination score* of p is the cardinality of $\Gamma(p)$. The similarity between p and q is defined as the Jaccard similarity between the sets $\Gamma(p)$ and $\Gamma(q)$, i.e., $J_s(p,q) = \frac{|\Gamma(p) \cap \Gamma(q)|}{|\Gamma(p) \cup \Gamma(q)|}$ and ranges between 0 and 1. The corresponding distance measure is thus $J_d(p,q) = 1 - J_s(p,q)$ and it is well known that it satisfies all metric properties. The selection of the Jaccard distance as a measure of diversity was based on the following rationale: *i*) it solely relies on the domination relationships among points, and thus, no user-defined distance function or additional input is required, *ii*) the quality of the resulting set of points does not depend on the skyline $\mathcal{S}$ alone, but on the characteristics of $\mathcal{D}$ as well, *iii*) it leads to elegant ways of diversity computation by means of min-wise independent permutations and *iv*) it is the most widely accepted measure for set similarity/dissimilarity.

To facilitate diversification, we take the perspective of [13], viewing k-diversity as a *dispersion* problem. In k-dispersion, the goal is to find k objects such that an objective function of their distance is optimized. The optimal solution of the k-dispersion problem is given by:

$$OPT = \arg \max_{\substack{\mathcal{A} \subseteq \mathcal{S} \\ |\mathcal{A}| = k}} f(\mathcal{A})$$

There are two basic alternatives for the objective function: *i*) in the k-MSDP (Max-Sum Dispersion Problem) the goal is to maximize the *sum of the pair-wise distances*, and *ii*) in the k-MMDP (Max-Min Dispersion Problem) the goal is to maximize *the minimum pair-wise distance*. Although either alternative can be employed, we choose to work with k-MMDP because it leads to 2-approximation algorithms, instead of the 4-approximation of k-MSDP [28].

EXAMPLE 1. *Figure 2 illustrates the output of a k-MMDP and a k-MSDP for k=3. For simplicity, assume that objects are 2D points and the L_2 distance is used as the measure of diversity. By inspecting the two solutions, we observe that the solution for k-MMDP returns points that are more distant to each other than k-MSDP. Both solutions contain the objects a and b. However, k-MMDP returns d as the third point, whereas k-MSDP returns c. Observe that the distance between a and c is smaller than the distance between a and d. Thus, in k-MSDP, although the sum of distances between the returned points is maximized, small distances may still occur, because they are compensated by larger ones.*



(a) solution for 3-MSDP (b) solution for 3-MMDP

Figure 2: Solutions to dispersion problems.

3.2 Straight-Forward Techniques

Before studying our framework, we describe briefly some straight-forward approaches and we report on their efficiency and effectiveness.

Brute-Force. This algorithm generates all pair-wise distances between skyline points, evaluates all $\binom{m}{k}$ alternatives and selects the optimal solution. Clearly, this method suffers from performance degradation by increasing the number of skyline points or the value k. In addition, there is a $O(m^2)$ cost to compute all pair-wise distances of the skyline points.

Simple Greedy. This method avoids the computation of all pair-wise distances among skyline points, by employing a heuristic-based algorithm, which guarantees a 2-approximation of the optimal solution. The main drawback of this approach is that in order to compute the Jaccard distance of two skyline points p and q, range queries must be executed to determine the cardinalities of the dominating sets $\Gamma(p)$, $\Gamma(q)$ and $\Gamma(p) \cap \Gamma(q)$. Evidently, the cost of such an approach is prohibitive, both with respect to I/O and CPU time, even when an aggregate multidimensional index is available.

Sampling-Based. One may be inclined to think that sampling $\mathcal{S}$ or $\mathcal{D} - \mathcal{S}$ will lead to a reduction of the cost to compute the k-most diverse skyline points. Taking a sample from $\mathcal{S}$ means that less than m skyline points will participate in the selection process, whereas sampling from $\mathcal{D} - \mathcal{S}$ results in fewer points that will contribute to the computation of the Jaccard distance between skyline points. In our case, sampling is not helpful as we discuss in the sequel.

LEMMA 2. *Let S be a set of m items in a metric space and Δ the diameter (maximum pair-wise distance). Any one-pass deterministic or randomized algorithm, that uses less than or equal to $m/2$ items, will fail with probability at least 1/2 to compute Δ exactly or provide a 2-approximation.*

PROOF. We focus on data sets containing exactly m points. Each point is uniquely identified by its id between 1 and m. Define the data sets $D_1, D_2, ..., D_m$ as follows. Each data set D_i contains a set of $m - 1$ points that are clustered together in a minimum bounding sphere with diameter δ,

whereas the point with id=i, $1 \leq i \leq m$ is located at a distance $2\delta + c$ from the center of the sphere, where c is a small constant. Let A be a deterministic algorithm that uses $s < m$ space. A randomly chosen D_i is selected and given as input to A. Each point of D_i is presented to A as a stream of points. Algorithm A selects s points to maintain in a deterministic manner. Each pair of nodes has the same probability of being the one with the maximum distance. Since we have $\binom{m}{2}$ different distances from which exactly one is the maximum, with s points we can produce $\binom{s}{2}$ pair-wise distances, meaning that the probability of success is $s(s-1)/m(m-1)$ and the probability of failure is $1 - s(s-1)/m(m-1)$. Setting $s = m/2$ we get that the failure probability is $(3m-2)/(4m-4) \geq 1/2$, $\forall m \geq 2$.

For the 2-approximation case, the difference is that the i-th element must be included in the s points selected by the algorithm. The distance between p_i and any of the other points lying inside the sphere is guaranteed to be at least $\Delta/2$. Thus, the success probability in this case equals s/m and consequently, the failure probability is $1 - s/m$. Therefore, even by maintaining $m/2$ elements from S any deterministic algorithm will err with probability $1/2$.

Using Yao's minimax principle [35], the effectiveness of any one-pass randomized algorithm cannot be better. □

The previous result states that if one wants to get a success probability larger than $1/2$ in estimating the diameter Δ (i.e., the optimal solution to the 2-dispersion problem), at least $m/2$ points must be stored by any deterministic or randomized algorithm. The result may be extended for any k. On the other hand, sampling from $\mathcal{D} - \mathcal{S}$ is not an effective solution either, because of the *sparsity* issue. To illustrate this effect, assume that the data set is viewed as a *domination matrix* [2] M with $n - m$ rows and m columns, $m = |\mathcal{S}|$ and $n = |\mathcal{D}|$. In this matrix, the cell in the i-th row and the j-th column is 1 if the j-th skyline point dominates the i-th data point and 0 otherwise. The sparsity of the domination matrix depends heavily on the data distribution as well as on the dimensionality of the data space. As an example, for 10,000 uniformly distributed points, in 3 dimensions the percentage of zeros is 45%, in 5 dimensions it is 84% and in 7 dimensions the percentage of zeros reaches 97%. The percentage of zeros is higher in anticorrelated data sets. It is evident that with the existence of sparsity it is not possible to simply perform random sampling and then try to compute the diversity among skyline points. Such an approach could miss important parts of the columns (containing 1's), resulting in erroneous diversity computation.

4. THE SKYDIVER FRAMEWORK

Given the shortcomings of the aforementioned solutions, more suitable techniques need to be employed. In this section, we present the SkyDiver framework for the skyline diversification problem. Our methodology consists of two consecutive phases, *fingerprinting* and *selection*:

Phase 1: Fingerprinting. This phase generates a reduced size *signature* for each skyline point, based on MinHashing.

Phase 2: Selection. This phase is responsible for selecting the k most diverse skyline points. It is applied either directly to the MinHash signatures or to the modified signatures generated by *Locality Sensitive Hashing* (LSH).

[2]Keep in mind that this matrix is used only for illustration purposes and it is not constructed in practice.

4.1 Phase 1: Fingerprinting with MinHashing

The basic objective of the following method is threefold: *i*) to avoid the execution of range queries, *ii*) to avoid the computation of all $O(m^2)$ pairwise skyline diversities and *iii*) to be able to work either with or without an index. In this respect, we propose the usage of the *MinHashing* technique [6], because it fits nicely with our diversity measure and requires a single pass over the data. Each column of the domination matrix is represented by a *signature* of size t, such that $t << (n - m)$.

Let $\mathcal{H} = \{h_1, ..., h_t\}$ be a set of t min-wise independent hash functions, where each h_i performs a random permutation of the rows. The cardinality of $\mathcal{H}$ (i.e., the number of hash functions used) determines the size of each signature. To generate random permutations of rows, each hash function $h_i \in \mathcal{H}$ is of the form $h_i(x) = a_i \cdot x + b_i \mod P$, where P is a prime number larger than $n - m$ and a_i, b_i are randomly chosen constants taking integer values in $[1, P]$. Although such a family of hash functions does not satisfy the *min-wise independence property*, it is used as an approximation that works very well in practice. Moreover, the MinHash technique has a very nice property that is directly related to the Jaccard similarity. If $J_s(p, q)$ is the Jaccard similarity between skyline points p and q, then for each hash function h_i it holds that $Prob[h_i(p) = h_i(q)] = J_s(p, q)$ [6].

Recall that each row of the domination matrix M corresponds to a bit-array. If the j-th position of the i-th row is 1 then the j-th skyline point dominates the i-th point. Each row is hashed t times using the hash functions in $\mathcal{H}$ and the signature of each skyline point is updated accordingly. Each signature is composed of t integer values, capturing the first row identifier with a non-zero element for each permutation.

4.1.1 Index-Free Signature Generation

There are many reasons why the data set may not be supported by an R-tree-like indexing scheme. Some of them are: *i*) high dimensionality, which deteriorates indexing performance, *ii*) the data set may contain intermediate results and thus no index is available yet, *iii*) operations performed on a projection of the data set in specific dimensions make the index inapplicable and *iv*) the data set contains *categorical* attributes that prevent multi-dimensional indexing.

Figure 3 outlines the index-free signature generation process. The algorithm takes as input the set of skyline points $\mathcal{S}$, the family of hash functions $\mathcal{H}$ and the number t of signature slots. The output is a matrix $\widehat{M}$ with t rows and m columns, where m is the cardinality of the skyline set. Each column of $\widehat{M}$ stores the MinHash signature of the corresponding skyline point. Each data point is scanned once (Line 2) and it is checked against the skyline points to detect dominance relationships. If a skyline point s_j dominates the investigated point (Line 6), then the matrix containing the MinHash signatures is updated accordingly (Line 7). The procedure to update the signature matrix is given in Lines 9–12, where we iteratively apply the hash functions.

Note that the index-free technique requires a single pass over the multidimensional data set, provided that the skyline set is available. The advantage of this method is that no index is required, whereas the sequential scan of the data set is expected to be efficient taking into consideration that

Algorithm SigGen-IF ($\mathcal{D}$, $\mathcal{S}$, $\mathcal{H}$, t)
Input: $\mathcal{D}$ data set, $\mathcal{S}$ skyline set, $\mathcal{H}$ hash functions, t number of slots per signature
Output: $\widehat{M}$ signature matrix

```
1.  initialize all cells of matrix M̂ with ∞;
2.  for (rowcount ← 1 to |D|) /* read data points */
3.      p ← next data point;
4.      if ( p is a skyline point ) then continue;
5.      for j ← 1 to |S|
6.          if (s_j ≺ p) then
7.              UpdateMatrix( rowcount, j );
8.  return(M̂);

9.  Procedure UpdateMatrix( row, column )
10. for i ← 1 to t
11.     v_i ← h_i(row); /* apply hash function */
12.     M̂[i, column] ← min(M̂[i, column], v_i);
```

Figure 3: Index-free signature generation.

usually the data file is stored sequentially on the disk. Most notably, such an approach does not require that attributes are numeric, but can handle categorical attributes as well as partially ordered domains. However, in cases where an index is already available, more efficient processing is possible, which we investigate in the next paragraphs.

4.1.2 Index-Based Signature Generation

More often than not, data points that are close in the multidimensional space are expected to be dominated by the same subset of skyline points. This feature is unique in index-based techniques since the sequential scan of data points does not guarantee any locality of references, unless the data is presorted based on a spatial proximity criterion (e.g., space filling curves). Therefore, when an index is present, we can exploit this property and reduce processing costs by avoiding index probes. We discuss in detail the appropriateness of each approach in Section 5.

Figure 4 outlines the MinHash signature creation in the presence of an aggregate R-tree. A priority queue PQ is used to store R-tree entries that require further consideration. The algorithm removes the top entry e from PQ (Line 11) and checks whether it is partially or fully dominated by a skyline point (Line 14). Full dominance means that the lower left corner of e is dominated, whereas partial dominance means that e is *not* fully dominated, but its upper right corner is. Partial dominance prevails and if both relations exist, we need to visit e's subtree (Line 16) by queuing it in PQ. In case of exclusive full dominance, *UpdateFullDominance* is called(Line 18), which updates the signatures without probing the index, by iterating over the number of enclosed objects in e (Lines 21–24).

Example 2. *Consider the set of points in Figure 5, enclosed by the minimum bounding rectangles R_1, R_2 and R_3. The skyline set is composed of a, b and c. Evidently, R_1 is fully dominated by b, whereas R_2 is fully dominated by a, b and c. Neither MBR is partially dominated. Therefore, for these two MBRs we avoid expanding the search toward the leaf and update the signatures immediately. In contrast, we have to increase the level of detail for R_3, because although it is fully dominated by c, it is partially dominated by b.*

Algorithm SigGen-IB ($\mathcal{D}$, $\mathcal{S}$, $\mathcal{H}$, t, R)
Input: $\mathcal{D}$ data set, $\mathcal{S}$ skyline set, $\mathcal{H}$ hash functions, t number of slots per signature
Output: $\widehat{M}$ signature matrix

```
1.  rowcount ← 1;
2.  initialize all cells of matrix M̂ with ∞;
3.  initialize priority queue PQ;
4.  for entry e in R.root
5.      DominatedRel( e, full, partial );
6.      if ( !partial.isEmpty ) then
7.          PQ.insert( e );
8.          continue;
9.      UpdateFullDominance( e, full );
10. while ( !PQ.empty ) do
11.     e ← PQ.removeTop();
12.     node ← R.read(e.id);
13.     for each entry e' in node
14.         DominatedRel( e', full, partial );
15.         if ( !partial.isEmpty and !node.isLeaf ) then
16.             PQ.insert( e' );
17.             continue;
18.         UpdateFullDominance( e', full );
19. return(M̂);

20. Procedure UpdateFullDominance( e, fullDom )
21. for k ← 1 to e.count
22.     for j ← 1 to |fullDom|
23.         UpdateMatrix( rowcount, S.index(fullDom_j) );
24.     rowcount++;
```

Figure 4: Index-based signature generation.

4.2 Phase 2: Selecting Diverse Skyline Points

The next step involves the selection of k skyline points, aiming to maximize their diversity. To perform this step efficiently, we should avoid the computation of the exact diversity score between pairs of skyline points, since this process involves probing the R-tree for Jaccard distance computations, which leads to significant computation costs due to the execution of range queries of large volume. Thus, in our framework we exploit the signatures only, avoiding access to the raw data. We study two different techniques: the first one is based solely on the MinHash signatures whereas the second applies locality-sensitive hashing aiming to less memory consumption and enabling speed/accuracy trade-offs.

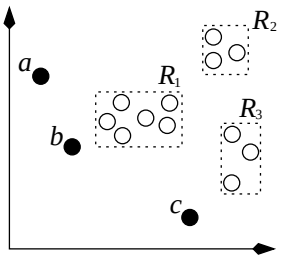


Figure 5: Domination of MBRs.

Algorithm SELECTDIVERSESET ($\mathcal{S}$, k, $F(.)$)
Input: $\mathcal{S}$ skyline set, k number of required points
$F(.)$ distance measure used
Output: $\mathcal{A}$ set of diverse skyline points

1. $m \leftarrow |\mathcal{S}|$; /* cardinality of skyline set */
2. $best \leftarrow -\infty$;
3. $p \leftarrow$ skyline point in $\mathcal{S}$ with max dominance score;
4. $\mathcal{A} \leftarrow \{p\}$
5. **while** ($|\mathcal{A}| < k$) **do**
6. $x \leftarrow \arg\max_{x \in \mathcal{S}-\mathcal{A}} \min_{y \in \mathcal{A}}\{F(x,y)\}$, i.e., find a point $x \in \mathcal{S}$-$\mathcal{A}$ such that the min distance $F(.)$ between x and the points in $\mathcal{A}$ is maximized;
7. resolve ties by selecting the point with the max dominance score;
8. $\mathcal{A} \leftarrow \mathcal{A} \cup \{x\}$;
9. **return**($\mathcal{A}$);

Figure 6: Outline for selecting k diverse points.

4.2.1 The Signature-Based Method

Let $\widehat{J_s}(p,q)$ be the *estimated* Jaccard similarity defined as the fraction of signature positions of p and q where their values agree. The corresponding *estimated* Jaccard distance $\widehat{J_d}(p,q)$ is simply $1-\widehat{J_s}(p,q)$. Since we have t different hash functions, formally we have:

$$\widehat{J_s}(p,q) = \frac{|j : 0 \le j \le t, h_j(p) = h_j(q)|}{t}$$

LEMMA 3. *([27]) The distance function $\widehat{J_d}$ respects the triangular inequality.*

According to the previous discussion, the set of signatures along with the distance measure $\widehat{J_d}$ is a *metric space*. We need this result to apply a greedy heuristic for the k-dispersion problem that guarantees a 2-approximation with respect to the optimal solution [28]. The algorithm first determines the two points with the maximum pair-wise distance, and then greedily adds more points to the result set, trying to maximize the minimum distance. The problem with this approach is that it requires quadratic complexity ($O(m^2)$) to determine the two most distant points. Here, we use a different technique with $O(k^2m)$ complexity without sacrificing the 2-approximation guarantee. The basic difference with respect to the work in [28] is that instead of selecting the two most distant points, we start with the skyline point with the maximum domination score and then use the greedy approach to include more points, until k points are determined. We also resolve ties by selecting the points with highest domination score, thereby treating coverage as a secondary objective.

The algorithm outline for selecting k diverse skyline points is given in Figure 6. Next, we show that SELECTDIVERSESET achieves a 2-approximation with respect to the optimal solution. This can be proved by using the same rationale as the one used for the proof of Theorem 2 of [28]. Here, we give a simpler alternative.

LEMMA 4. *Algorithm* SELECTDIVERSESET *reports a set of k skyline points in $O(k^2m)$ time, achieving a 2-approximation with respect to the optimal solution.*

PROOF. The first point is selected in $O(m)$ time. To select the second one we compute the distance between each of the $m-1$ remaining points and the one selected. To select the third point we need $2(m-2)$ distance computations. Thus, to select all k points we need in total $m+(m-1)+2(m-2)+$ ... $+k(m-k) \in O(k^2m)$ distance computations. Let, S_j denote the set containing the j skyline points selected so far, where $j \le k$. Thus, when the first point is selected we have $|S_1| = 1$, whereas when all k points are selected $|S_k| = k$. Let p_1 be the first point selected. To select the second point p_2, the algorithm scans all skyline points and picks the one that maximizes its distance from p_1. Create a neighborhood $\mathcal{N}(s_i)$ for each skyline point $s_i \in S_j$, such that $\mathcal{N}(s_i) = \{q \in D : F(s_i,q) < OPT/2\}$. We argue that there must be a point in D not belonging to any of the j neighborhoods. If this is not the case, then the optimal solution to the k-MMDP problem could not be OPT, which is a contradiction. Thus, for any $j \le k$ it is guaranteed that the minimum distance between points in S_j is at least $OPT/2$. □

According to [12], if $\Omega(\varepsilon^{-3}\beta^{-1}\log(1/\delta))$ is the signature size, where ε is the maximum allowed error ($0 < \varepsilon < 1$), then with probability at least $1-\delta$ it holds that $(1-\varepsilon)J_s(p,q) + \varepsilon\beta \le \widehat{J_s}(p,q) \le (1+\varepsilon)J_s(p,q) + \varepsilon\beta$, where $0 < \beta < 1$ is the required precision. This essentially means that the 2-approximate greedy heuristic using $\widehat{J_d}$ as the distance measure, is applied on the (ε,δ)-approximation of the Jaccard distances, for which the inequalities are $(1+\varepsilon)J_d(p,q) - \varepsilon - \varepsilon\beta \le \widehat{J_d}(p,q) \le (1-\varepsilon)J_d(p,q) + \varepsilon - \varepsilon\beta$. Consequently, by setting appropriate values for ε and δ, the signature distances can be made very close to the original Jaccard distances [9]. Due to distance distortions, as a result of embedding the distances in lower dimensionality, it is possible to obtain a sub-optimal solution. The following theorem relates the true optimal solution, to the one computed by working with MinHash signatures.

THEOREM 1. *Let OPT be the value of the optimal solution to the k-diversity problem in the original space and let x, y denote the corresponding skyline points, i.e., $J_d(x,y) = OPT$. Also, let $\widehat{OPT}$ be the optimal value if the problem is solved using MinHash signatures and let a, b be the corresponding skyline points, i.e., $\widehat{J_d}(a,b) = \widehat{OPT}$. For a given ε and sufficiently small δ, it holds that: $J_d(a,b) \ge \frac{1+\varepsilon}{1-\varepsilon}OPT - \frac{2\varepsilon}{1-\varepsilon}$.*

PROOF. The optimal solution is given by a set of k points, $S_k = \{o_1, o_2, ..., o_k\}$, forming a k-clique. Each k-clique is *represented* by a single value, i.e., the minimum of the $\binom{k}{2}$ distances among any two points in the clique, $R(S_k) = min_{\forall o_i,o_j}(J_d(o_i,o_j))$. Based on this formulation, OPT is the representative distance of the optimal solution.

Given our formalization of the top-k diversity problem as an instance of k-MMDP, for any set of k points, $P_k = \{p_1, p_2, ..., p_k\}$, $P_k \neq S_k$, it holds that $R(P_k) = min_{\forall p_i,p_j}(J_d(p_i,p_j)) \le R(S_k) = OPT$. In other words, any other k-clique will either be at most as good as S_k but never better, e.g., containing an edge with equal minimum weight, or it will contain at least one distance strictly worse than OPT.

Let $\widehat{S_k}$ be the set of k points that we select as the optimal solution in the MinHash signature space and $\widehat{OPT}$ be their representative distance defined by the skyline points a

and b, i.e. $\widehat{J}_d(a,b) = \widehat{OPT}$. It is not hard to verify that if $\widehat{S}_k = S_k$, or $\widehat{S}_k$ contains $R(S_k)$ but no worse edge, then $J_d(a,b) = \widehat{J}_d(a,b) = OPT$ and the inequality surely holds. The challenging case is when $\widehat{S}_k$ contains some points whose edge is worse in the original space than in the signature space. Specifically, the problem arises when in the signature space $\widehat{J}_d(a,b) \geq \widehat{J}_d(x,y)$, despite that $J_d(a,b) < OPT$ in the original space. Let D_O be any distance from the optimal solution O in the original space, $OPT < D_O$. Since $\varepsilon > 0 \Rightarrow (1+\varepsilon)OPT - \varepsilon < (1+\varepsilon)D_O - \varepsilon$. Simply put, underestimating a higher value than OPT could also yield a sub-optimal result, but the worst case is obtained when we underestimate OPT itself. The same holds for overestimating $J_d(a,b)$ and lower values. On the other hand, overestimating a value $J_d(w,z)$, $J_d(w,z) < J_d(a,b) < OPT$ so that $\widehat{J}_d(w,z) \geq \widehat{J}_d(a,b) \geq \widehat{J}_d(x,y)$ contradicts our assumption that $\widehat{J}_d(a,b)$ is the optimal solution in the signature space; otherwise, $\widehat{J}_d(w,z)$ would have been selected. For the worst case scenario to occur, i.e., $\widehat{J}_d(a,b) \geq \widehat{J}_d(x,y)$, where $J_d(a,b)$ has been overestimated and OPT has been underestimated, it should hold:

$$\widehat{J}_d(a,b) \geq \widehat{J}_d(x,y) \Leftrightarrow (1-\varepsilon)J_d(a,b)+\varepsilon \geq (1+\varepsilon)OPT-\varepsilon \Leftrightarrow$$

$$J_d(a,b) \geq \frac{(1+\varepsilon)}{(1-\varepsilon)}OPT - \frac{2\varepsilon}{1-\varepsilon}$$

□

COROLLARY 1. *Let a and b be the two skyline points defining the solution of the 2-approximation heuristic for the k-diversity problem, when run over the MinHash signatures, where $J_d(a,b)$ is their corresponding distance. Also, let OPT be the optimal distance. Then, the following holds: $J_d(a,b) \geq \frac{1}{2}\frac{(1+\varepsilon)}{(1-\varepsilon)}OPT - \frac{\varepsilon}{1-\varepsilon}$.*

For the previous bounds to work, we have assumed that the parameter δ is very small. This is because if the distances are not well-preserved in the signature space, then we cannot have a guarantee about the solution in the original space.

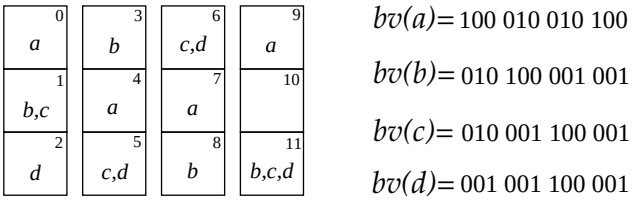
4.2.2 The LSH-Based Method

A potential limitation of the signature-based approach, is that the size of the signatures may be increased significantly in order to reduce the error probability, resulting in: *i*) increased processing costs during distance computation and *ii*) increased memory requirements. Keeping the signature size as small as possible has a direct negative impact on accuracy, due to Theorem 1, a result we also experimentally validate. Instead, we propose to apply *Locality Sensitive Hashing* (LSH) [18].

The signature matrix is split to ζ zones, each containing r rows such that $\zeta \cdot r = m$. For each zone, a hash function is applied and each signature part is hashed to a bucket. Based on this scheme, the probability that two skyline points s_1, s_2, will hash to different buckets in all zones equals [3] $(1 - J_s(s_1,s_2)^r)^\zeta$, whereas the probability that will hash to the same bucket in at least one zone equals $1 - (1 - J_s(s_1,s_2)^r)^\zeta$. Our target is *to select skyline points that are hashed in different buckets in all zones* or if this is not possible, *to minimize the number of skyline points that fall in the same bucket in some of the zones*. The values of r and ζ are controlled by the value of the threshold ξ, which is selected such that $\zeta \cdot r = m$ and $\xi \approx (1/\zeta)^{(1/r)}$. Basically, the threshold ξ controls the shape of the sigmoid function $1-(1-J_s(s_1,s_2)^r)^\zeta$. For example, we may assume that we consider two points similar if their similarity is more than 20%, 50% or 80%.

Let B denote the number of buckets used per zone. Each skyline point is seen as a bit-vector containing $\zeta \cdot B$ dimensions, where a value of 1 (0) denotes that the skyline point is hashed (not hashed) to the corresponding bucket. Consequently, two skyline points s_1, s_2 are dissimilar if they both have a value of 1 in as few bit-vector positions as possible, thus, the diversity is quantified by the *Hamming distance* between their corresponding bit-vector representations. In particular, we observe that the number of buckets where two skyline points disagree equals half the Hamming distance between their corresponding bit-vectors. Note also that since each skyline point is necessarily hashed in exactly one bucket in each hashtable, the L_1 norm of its bit-vector is equal to ζ, i.e., $||bv(s_i)||_1 = \zeta, \forall i \in [1,m]$.

EXAMPLE 3. *Figure 7 depicts a possible distribution of signatures into buckets, when ζ=4 and B=3. The number shown in the upper-right corner of each bucket corresponds to the position in the bit-vectors, which are presented on the right. Each bit-vector contains $\zeta \cdot B$=12 bits, where exactly four of them are set. By observing points a and b we see that they are never hashed to the same bucket. Therefore, their distance should be equal to 4, whereas one can easily verify that the Hamming distance between $bv(a)$ and $bv(b)$ is 8.*



(a) hashtables for the zones (b) skyline bit-vectors

Figure 7: Buckets and bit-vectors of skyline points.

Since the Hamming distance satisfies the triangular inequality, the 2-approximation heuristic is immediately applicable. Thus, to determine the k most diverse skyline points, algorithm SELECTDIVERSESET of Figure 6 is applied by using the Hamming distance of the bit-vectors instead of the signature-based distance that has been used previously. We denote this algorithm as SKYDIVER-LSH.

5. PERFORMANCE EVALUATION

In this section, we report on the results of a comprehensive set of experiments, towards comparing the various techniques covered in the previous sections. First, we present the implemented algorithms as well as the data sets used.

5.1 Algorithms and Data Sets

We have implemented four different algorithms, two of which (SKYDIVER-MH and SKYDIVER-LSH) can be applied

[3]To be precise, we should use $\widehat{J}_s(s_1,s_2)$, but since the distortion of the similarities can be made arbitrarily small, we can safely assume that $\widehat{J}_s(s_1,s_2) \approx J_s(s_1,s_2)$.

regardless of having an index in place. The evaluated algorithms are presented in Table 3.

Table 3: Evaluated algorithms

Algorithm	Reference
BRUTE-FORCE (BF)	brute-force algo. (Sec. 3.2)
SIMPLE-GREEDY (SG)	simple greedy algo. (Sec. 3.2)
SKYDIVER-MH (MH)	MinHash-based algo. (Sec. 4.2.1)
SKYDIVER-LSH (LSH)	LSH-based algo. (Sec. 4.2.2)

We have generated synthetic data sets following the *independent* (IND) and *anticorrelated* (ANT) distributions, using the methodology presented in [4]. In addition, we have used two real-life data sets: *Forest Cover* (FC) downloaded from UCI Machine Learning Repository (http://kdd.ics.uci.edu) and *Recipes* (REC) [20], obtained from Sparkrecipes.com, where each data point is a recipe and its attributes are the nutritional values for several common compounds, e.g., carbohydrates, protein, calcium etc. Table 4 summarizes the basic dataset characteristics, with default values underlined.

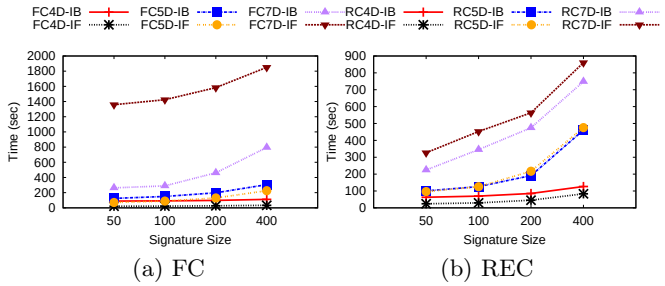
Table 4: Basic data set characteristics

Data set	Cardinality	Dimensionality
Independent (IND)	1M, 2M, 5M, 7M	2, 3, 4, 6
Anticorrelated (ANT)	1M, 2M, 5M, 7M	2, 3, 4, 6
Forest Cover (FC)	$\sim$ 581K	4, 5, 7
Recipes (REC)	$\sim$ 365K	4, 5, 7

The code was written in C++ and all experiments were run on a Quad-Core @3.5GHz machine, with 4Gb RAM, running Linux. Each data set was indexed by an aggregate R*-tree, with a 4Kb page size. An associated cache with 20% of the corresponding R*-tree's blocks was used with every experiment. Timings reported in the graphs are in seconds, measured as CPU processing time and assuming a default value of 8ms per page fault. Unless stated otherwise, all values reported below refer to the 2-step process of finding the k-most diverse skyline points, without the cost of finding the skyline itself as it does not affect the relative performance of the algorithms. Regarding effectiveness, we report the minimum (Jaccard) distance among the pair of points that has been selected by each approach.

5.2 Experiments and Results

We begin by evaluating when signature creation should use an index (IB) or not (IF), in case we have such an option. We then evaluate the efficiency of all techniques compared to various parameters, using the IB approach, since BF and SG use the index as well. We finally report on result quality and memory consumption.



(a) FC (b) REC

Figure 8: Time for generating MinHash signatures vs signature size.

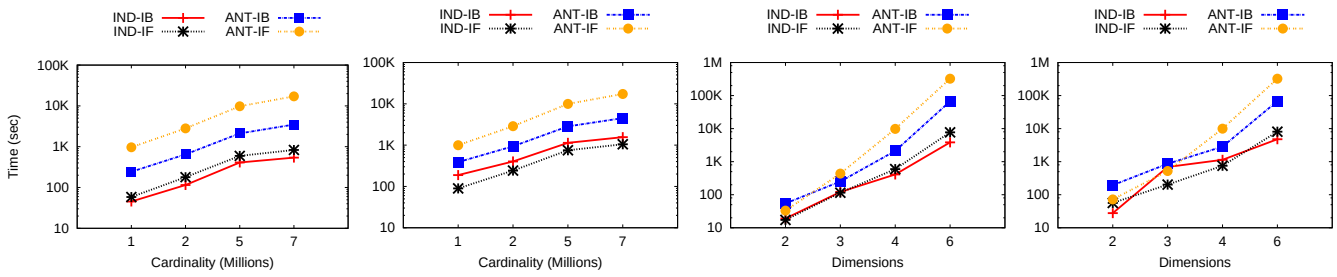
Signature Generation. Our first experiment focuses on the cost for signature generation and how the index affects this step. Figures 8(a)-(b) show the signature generation time as a function of the signature size, for all dimensions of FC and REC data sets, respectively. Clearly, by increasing the signature size, the signature generation phase requires more time. Nevertheless, selecting IB or IF seems to be unrelated to signature size. Similar results have been obtained for IND and ANT, which we omit due to space limitations.

Figures 9(a)-(d) report the time taken to generate the signatures, whether we use an index (IB) or not (IF), for varied cardinalities / dimensionalities of IND and ANT data sets. Figures 9(a) and 9(b) report CPU and total time – I/O's included – respectively, for varying cardinalities and $d = 4$. ANT data consistently favor the IB approach. However, for IND data, the IF method is more efficient when total time is concerned. On the contrary, when taking only CPU time into account, IB is better. This is due to a lot of I/Os on the R-tree, more than what a linear scan on the actual data set requires.

Even more interesting is the case when we vary the dimensionality, as shown in Figures 9(c) and 9(d). In particular, for ANT data sets, low dimensionality favors the IF approach. In this case, the costs are mostly due to I/Os. However, as dimensionality increases, more dominance checks are executed, which IF performs naïvely. On the other hand, IB saves on CPU costs, by utilizing the index. For IND data, IB and IF differ marginally for few dimensions and as d increases, IB is favored. For 2D, the R-tree saves several I/O operations, and the overall cost of IB is much lower. However, for average dimensionality, the I/O cost sharply increases for IB, making it less suitable. Basically, partial dominations of MBRs have dramatically increased, which necessitate that we decompose them further, resulting in additional I/Os. Specifically, we observe an $\sim 70\times$ increase in the number of I/Os from 2D to 3D, but the I/O increase is niche as d increases from that point onwards. A big part of the R-tree has to be traversed when $d \geq 3$, yet several dominance checks are saved, explaining why CPU-costs do not follow this trend. Given the efficiency when the signature size is set to 100 and the fact that we achieve good quality (Figure 12), as we discuss in the next paragraphs, we adopt it as the default signature size for the rest of our evaluation.

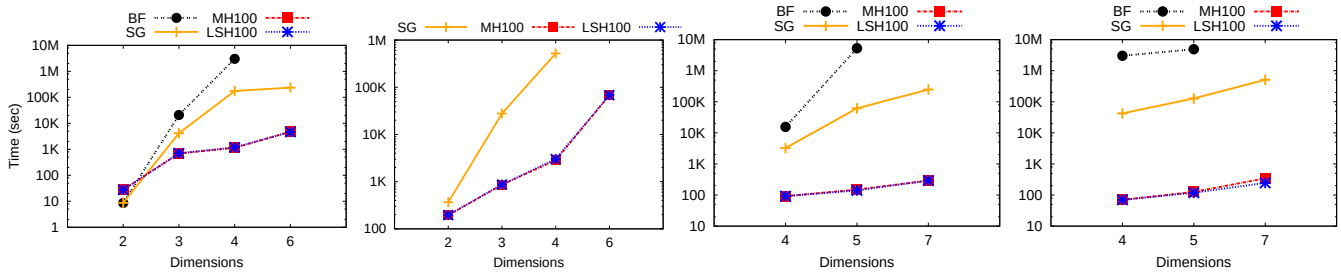
User Guide. We propose the following scheme which is experimentally validated: The IB method should be considered: *i*) when the R-tree can be memory resident, assuming enough resources, whereas for a disk-resident index *ii*) for average and high-dimensional data ($d \geq 4$) and *iii*) when $d = 2$, provided we are dealing with IND data. In the few remaining cases, IF should be favored.

Runtime VS Dimensionality. We now turn to the efficiency of the techniques for selecting the k-most diverse skyline points. Figures 10(a)-(d) demonstrate their performance on all data sets, for varying dimensionalities. In particular, we have plotted the overall time taken to compute the 10-most diverse skyline points, including the time for signature generation (for MH and LSH). As expected, BF shows the worst performance, given that it searches exhaustively for the optimal solution. By increasing the dimensionality, the number of skyline points increases too and, consequently, so does the number of $\binom{m}{k}$ enumerations. Moreover, unlike the other techniques, BF's reported times are for $k = 2$; $k = 10$ yields even more enumerations, since the



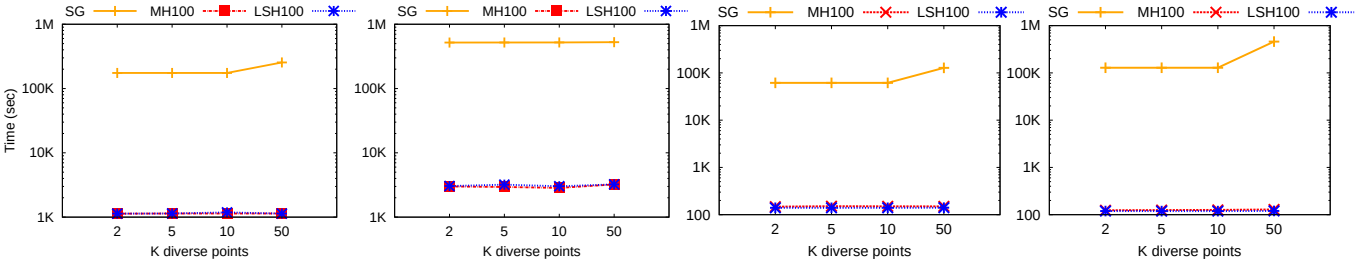
(a) CPU time (b) Total time (c) CPU time (d) Total time

Figure 9: Time for generating MinHash signatures of size 100 for synthetic data sets.



(a) IND (b) ANT (c) FC (d) REC

Figure 10: Runtime for $k = 10$ diverse skyline points vs dimensionality.



(a) IND (b) ANT (c) FC (d) REC

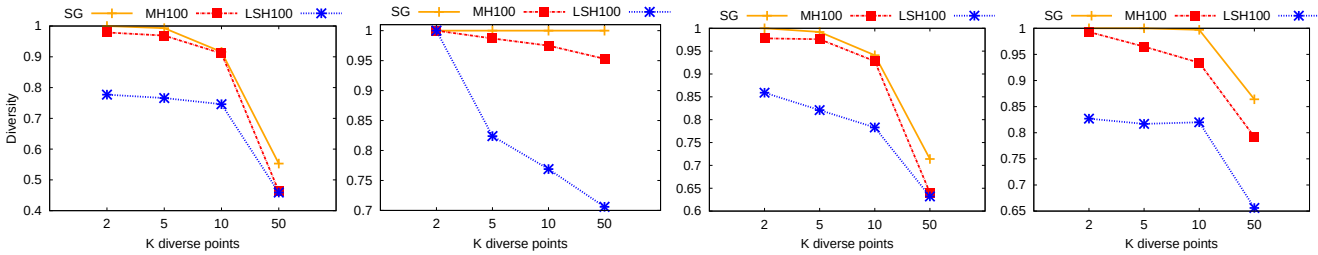
Figure 11: Runtime vs number of diverse points (k).

skyline contains a few hundred points at best. We even ran BF for $k = 5$, but at the time of writing the experiments have not finished yet. Not surprisingly, BF is inappropriate for practical applications, even when dealing with small skyline sets and reasonable values of k. For this reason, we omit it from subsequent experiments.

The greedy algorithm SG is inferior to MH and LSH by, approximately 2-3 orders of magnitude. Note that we have boosted SG, by maintaining in-memory the minimum distance of each non-selected skyline point. Even so, most of the time is spent on I/Os due to range queries for Jaccard distance computations, whereas CPU cost is only a fraction. This validates our goal to keep range queries to a minimum. SG performs better only for the IND data set and $d = 2$, where there are very few skyline points ($\sim 5 - 10$) and signature creation phase places enough overhead to make the signature-based techniques slightly worse. In all other occasions, SG's performance is worse; in fact it did not complete for the ANT 6D setting. Finally, though MH and LSH differ slightly, at this granularity their difference is not discernible.

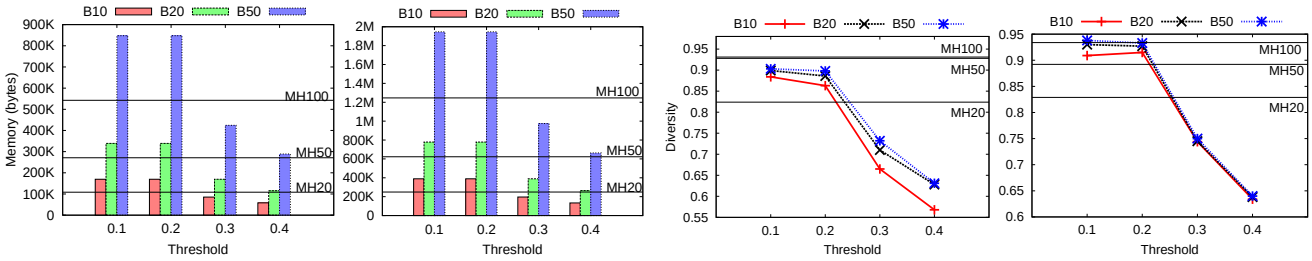
Runtime VS Number of Points (k). Figures 11(a)-(d) portray the efficiency of the techniques with respect to the number of requested points. The graphs clearly support our earlier findings that MH and LSH are superior to SG by orders of magnitude, for reasonable values of k. All three algorithms exhibit a consistent behavior in all data sets and k values: MH and LSH perform almost identically for all k values, with LSH being slightly better as shown in Figure 11(c)-(d), which is one of the main reasons to consider it over MH. CPU costs are minimal for these techniques, accounting for no more than 45 sec for ANT, and at most 2 seconds for the other data sets, with k=50 and default values for the other parameters. On the other hand, for all k values, SG is burdened with an excessive number of I/Os, due to range queries, despite being boosted. The technique also shows a noticeable increase in runtime for k=50, across all data sets, as a result of increased CPU costs. This is because when increasing k, the pair-wise Jaccard distance computations add-up to a more noticeable amount, given that range queries require O(d) checks, and recursively descend the R-tree if needed, to compute the intersection.

Quality of Results. We now turn our attention to the ef-



(a) IND (b) ANT (c) FC (d) REC

Figure 12: Quality vs number of diverse points (k).



(a) FC - Memory (b) REC - Memory (c) FC - Quality (d) REC - Quality

Figure 13: LSH vs MinHashing, for $k = 10$ diverse skyline points, with signature size fixed to 100

fectiveness aspect of our approach. Figures 12(a)-(d) demonstrate the diversity score, i.e., the minimum Jaccard distance in the original space, of the selected set of skyline points, for different values of k (number of selected points). As expected, by increasing k, the minimum Jaccard distance is reduced. SG performs better than MH and LSH in general, however, the latter two achieve very good performance, given their efficiency savings. With the exception of REC data set, MH is only slightly worse than SG for k values up to 10. In constrast, LSH has a steeper decline in the diversity score, but requires less memory, as shown in Figure 13(a)-(b) and explained in the sequel.

MinHashing VS LSH. Figure 13 depicts a comparison between MH and LSH, demonstrating the memory vs accuracy trade-off. In particular, we have performed a series of experiments with a fixed k value (k=10), while varying the parameters ξ (threshold) and B (number of buckets per zone) for LSH, and (varying) the signature size for MinHash. By increasing ξ, the number of zones ζ is reduced, which increases memory savings. In addition, maintaining fewer buckets per zone reduces memory consumption further. The price we pay in this case is a drop in accuracy. As expected, the accuracy of LSH is lower than that of MH as shown in Figure 13(c)-(d), whereas the savings in storage are more sensitive to the value of ξ, due to the high correlation between ξ and ζ as shown in Figure 13(a)-(b). For example, by using LSH with $\xi = 0.2$ and B=20, we need around 300Kb for the FC data set, whereas MH requires almost 600Kb. Moreover, the corresponding diversity score obtained by LSH is 0.88 when MH performs marginally better obtaining a diversity score of 0.93. Overall, the significant reduction in memory requirements make LSH a very attractive alternative, in cases where we are willing to sacrifice accuracy, up to an acceptable level.

Another key observation is that by simply reducing the signature size in MinHashing does not give promising results. For example, using a threshold of 0.2, with 10 buckets per zone, LSH obtains results of similar or better quality, while requiring less memory than MH50. In general, the accuracy of MinHashing drops rapidly by decreasing the signature size, whereas by carefully controlling the threshold and the number of buckets per zone, LSH can be tuned better.

6. CONCLUSIONS

In this article, we have studied the problem of selecting k skyline points that best diversify the skyline result. Our proposal is entirely based on the dominance relationships between points and therefore, no artificial distance functions are required. In particular, we quantify the diversity between two skyline points as the Jaccard distance of their corresponding domination sets, capturing dataset characteristics in the process. To confront the NP-hardness of the problem, we employ MinHash signatures and Locality Sensitive Hashing along with a 2-approximation greedy heuristic. Our techniques work in an index-free or index-based (R-tree-like) case, achieve a controlled error and demonstrate a significant performance improvement compared with straight-forward approaches. In addition, our framework supports a memory consumption vs accuracy trade-off. We also experimentally validated the performance of our approach based on real-life and synthetic data sets, achieving orders of magnitude better runtime performance in comparison to straight-forward techniques.

Interesting future directions include: i) the diversification of a data set $\mathcal{A}$ based on (dominance) relationships over another set $\mathcal{B}$, where $\mathcal{A}$ is not necessarily a Pareto optimal set (as in the skyline case), as well as ii) parallelization aspects of our methodology, aiming for scalable skyline diversification over massive data.

Acknowledgements: This work has been co-financed by

EU and Greek National funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Programs: Heraclitus II fellowship, THALIS - GeomComp, THALIS - DISFER, ARISTEIA - MMD" and the EU funded project INSIGHT.

7. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proc. WSDM '09*, pages 5–14, 2009.

[2] A. Angel and N. Koudas. Efficient diversity-aware search. In *Proc. SIGMOD '11*, pages 781–792, 2011.

[3] 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. *J. ACM*, 25(4):536–543, 1978.

[4] S. Börzsönyi, D. Kossmann, and K. Stocker. The skyline operator. In *Proc. ICDE '01*, pages 421–430, 2001.

[5] B. Boyce. Beyond topicality: A two stage view of relevance and the retrieval process. *Information Processing & Management*, 18(3):105–109, 1982.

[6] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher. Min-wise independent permutations. *J. Computer System Science*, 60(3):630–659, 2000.

[7] H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite vc-dimension: (preliminary version). In *Proc. SCG '94*, pages 293–302, 1994.

[8] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proc. SIGIR '08*, pages 659–666, 2008.

[9] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J. D. Ullman, and C. Yang. Finding interesting associations without support pruning. *IEEE Trans. on Knowl. and Data Eng.*, 13(1):64–78, 2001.

[10] A. Das Sarma, A. Lall, D. Nanongkai, R. J. Lipton, and J. Xu. Representative skylines using threshold-based preference distributions. In *Proc. ICDE '11*, pages 387–398, 2011.

[11] A. Das Sarma, A. Lall, D. Nanongkai, and J. Xu. Randomized multi-pass streaming skyline algorithms. *PVLDB*, 2(1):85–96, 2009.

[12] M. Datar and S. Muthukrishnan. Estimating rarity and similarity over data stream windows. In *Proc. ESA '02*, pages 323–334, 2002.

[13] M. Drosou and E. Pitoura. Dynamic diversification of continuous data. In *Proc. EDBT '12*, pages 216–227, 2012.

[14] E. Erkut, Y. Ülküsal, and O. Yenicerioğlu. A comparison of p-dispersion heuristics. *Comput. Oper. Res.*, 21(10):1103–1113, 1994.

[15] Y. Gao, J. Hu, G. Chen, and C. Chen. Finding the most desirable skyline objects. In *Proc. DASFAA '10*, pages 116–122, 2010.

[16] J. B. Ghosh. Computational aspects of the maximum diversity problem. *Oper. Res. Lett.*, 19(4):175–181, 1996.

[17] J. R. Haritsa. The kndn problem: A quest for unity in diversity. *IEEE Data Eng. Bull.*, 32(4):15–22, 2009.

[18] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. STOC '98*, pages 604–613, 1998.

[19] C.-C. Kuo, F. Glover, and K. S. Dhir. Analyzing and modeling the maximum diversity problem by zero-one programming. *Decision Sciences*, 24(6):1171–1185, 1993.

[20] T. Lappas, G. Valkanas, and D. Gunopulos. Efficient and domain-invariant competitor mining. In *Proc. SIGKDD '12*, pages 408–416, 2012.

[21] X. Lin, Y. Yuan, Q. Zhang, and Y. Zhang. Selecting stars: The k most representative skyline operator. In *Proc. ICDE '07*, pages 86–95, 2007.

[22] Z. Liu, P. Sun, and Y. Chen. Structured search result differentiation. *PVLDB*, 2(1):313–324, 2009.

[23] W. Maass. Efficient agnostic pac-learning with simple hypothesis. In *Proc. COLT '94*, pages 67–75, 1994.

[24] D. Papadias, Y. Tao, G. Fu, and B. Seeger. Progressive skyline computation in database systems. *ACM Trans. Database Syst.*, 30(1):41–82, 2005.

[25] A. N. Papadopoulos, A. Lyritsis, and Y. Manolopoulos. Skygraph: an algorithm for important subgraph discovery in relational graphs. *Data Min. Knowl. Discov.*, 17(1):57–76, 2008.

[26] D. Pisinger. Upper bounds and exact algorithms for p-dispersion problems. *Comput. Oper. Res.*, 33(5):1380–1398, 2006.

[27] A. Rajaraman and J. Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2011.

[28] S. S. Ravi, D. J. Rosenkrantz, and G. K. Tavyi. Heuristic and special case algorithms for dispersion problema. *Operations Research*, 42(2):299–310, 1994.

[29] C. Sheng and Y. Tao. On finding skylines in external memory. In *Proc. PODS '11*, pages 107–116, 2011.

[30] K. Spärck-Jones, S. E. Robertson, and M. Sanderson. Ambiguous requests: implications for retrieval tests, systems and theories. *SIGIR Forum*, 41(2):8–17, 2007.

[31] J. Stoyanovich, W. Mee, and K. A. Ross. Semantic ranking and result visualization for life sciences publications. In *Proc. ICDE '10*, pages 860–871, 2010.

[32] Y. Tao, L. Ding, X. Lin, and J. Pei. Distance-based representative skyline. In *Proc. ICDE '09*, pages 892–903, 2009.

[33] V. Vapnik. *Statistical learning theory*. Wiley, 1998.

[34] M. R. Vieira, H. L. Razente, M. C. N. Barioni, M. Hadjieleftheriou, D. Srivastava, C. T. Jr., and V. J. Tsotras. Divdb: A system for diversifying query results. *PVLDB*, 4(12):1395–1398, 2011.

[35] A. C.-C. Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proc. FOCS '77*, pages 222–227, 1977.

[36] M. L. Yiu and N. Mamoulis. Efficient processing of top-k dominating queries on multi-dimensional data. In *Proc. VLDB '07*, pages 483–494, 2007.

[37] S. Zhang, N. Mamoulis, D. W. Cheung, and B. Kao. Efficient skyline evaluation over partially ordered domains. *PVLDB*, 3(1-2):1255–1266, 2010.

[38] H. Zhenhua, X. Yang, and L. Ziyu. *l*-skydiv query: Effectively improve the usefulness of skylines. *SCIENCE CHINA Information Sciences*, 53(9):1785–1799, 2010.