COPYRIGHT NOTICE

This is the author's version of the work. The definitive version was published in *IEEE Transactions on Knowledge and Data Engineering* (TKDE), 2011.

DOI: `http://dx.doi.org/10.1109/TKDE.2011.93`.

Indexing Uncertain Data in General Metric Spaces

Fabrizio Angiulli and Fabio Fassetti

Abstract—In this work we deal with the problem of efficiently answering range queries over *uncertain objects* in a *general metric space*. In this study, an uncertain object is an object that always exists but its actual value is uncertain and modeled by a multivariate probability density function. As a major contribution, this is the first work providing an effective technique for indexing uncertain objects coming from general metric spaces. We generalize the reverse triangle inequality to the probabilistic setting in order to exploit it as a discard condition. Then, we introduce a novel pivot-based indexing technique, called UP–index, and show how it can be employed to speed up range query computation. Importantly, the candidate selection phase of our technique is able to noticeably reduce the set of candidates with little time requirements. Finally, we provide a criterion to measure the quality of a set of pivots and study the problem of selecting a good set of pivots according to the introduced criterion. We report some intractability results and then design an approximate algorithm with statistical guarantees for selecting pivots. Experimental results validate the effectiveness of the proposed approach and reveal that the introduced technique may be even preferable to indexing techniques specifically designed for the Euclidean space.

Index Terms—Indexing, Metric spaces, Uncertain data

1 Introduction

In this work we deal with the problem of efficiently answering *range queries* over *uncertain objects*. In this study, an uncertain object is an object that always exists but its actual value is uncertain and modeled by a multivariate probability density function [1], [2], [3], [4], [5], [6], [7].

In past years, many efforts for providing efficient algorithms for *similarity search* in metric spaces have been made, due to the variety of fields where this kind of operation is useful [8], [9], [10]. These algorithms search for objects in a given data collection which are similar, or close, to an input object, also called *query object*. In particular, *range queries*, which are of interest here, take as input the query object q and a radius R, and return all the objects of the collection lying within distance R from q. Most of the similarity search approaches proposed in the literature rely on the strategy of building an *index*, that is a data structure aimed to reduce the number of distance computations at query time. Basically, these algorithms distinguish between *indexing time* and *query time*. Loosely speaking, at indexing time, the given collection of objects is partitioned into a set of equivalence classes. At query time, some criteria are exploited in order to discard as many irrelevant classes as possible, while the objects of the non-discarded classes are exhaustively compared with the query object.

As far as the uncertain setting is concerned, to perform a range query a further parameter is required, that is a probability threshold τ. Thus, generally, the query retrieves all the uncertain objects that lie within a given region with probability at least τ [1], [4], [6].

F. Angiulli and F. Fassetti are with DEIS, University of Calabria, Via P. Bucci 41C, 87036 Rende (CS), Italy, email: {f.angiulli, f.fassetti}@deis.unical.it

Uncertainty arises in real data in many ways, resulting from the limitations of the equipment, repeated measurements, continuously changing data values, or in other ways. Despite a lot of applications deal with data modeled as elements of the Euclidean space or, in more general, of a vector space, there are also relevant applications involving data coming from non-vector spaces which are indeed metric ones. Examples of notable metrics involving non-vector data are the Edit distance [11] defined over strings and the distances induced by graph structures.

As an example, consider a graph modelling a cellular network, where graph nodes represents cellular antennas (and their associated cells) and edges represent wired connections between cells. In such a scenario, mobile wireless devices can be represented by means of uncertain objects having associated the probability to stay at a particular node. The distance between two devices is then given by the number of links composing the smallest path joining the cells they belong to, also said number of hops. This distance is a metric, but the involved space is not a vector one. A further example, concerning uncertain strings, is provided later in the paper.

The problem of searching over uncertain data was first introduced in [1] where the authors considered the problem of querying one-dimensional real-valued uniform pdfs. [12] introduced the probabilistic similarity join operator and took into account the problem of its computation when each uncertain object is represented by a finite set of sample points. In order to improve efficiency of the UK-means algorithm [13], that is a clustering algorithm based on the expected distance (ED) [14] from uncertain objects, in [15] various pruning methods to avoid the expensive ED calculation are introduced.

Since ED is a metric, the triangle inequality, involving some pre-computed expected distances between a set of anchor objects and the uncertain data set objects, can be straightforwardly employed in order to prune unfruitful distance computations. [16] considered the problem of indexing categorical uncertain data. Specifically, they presented an extension of the usual equality operator for uncertain data that can be used to define operations such as joins over uncertain attributes. Moreover, they proposed two index structures for uncertain categorical data, the first based on R-trees and the second based on an inverted index structure, to support probabilistic equality queries. To answer uncertain queries [4] introduced the concept of probabilistic constrained rectangles (PCR) of an object o, which, loosely speaking, are families of m-dimensional rectangles surrounding the region which the object o is most likely to belong to, and presented some pruning rules based on the use of these PCRs. Intuitively, PCRs can be considered a generalization of the concept of MBR and are stored in an index data structure, called U-tree, which shares a common rationale with the R-tree. In [17] it is pointed out that the U-tree does not always provide a good support to range queries with non-rectangular regions or rectangular regions not aligning to axes. Motivated by these facts, the UI-tree, an R-tree-like inverted index structure which is based on the partitions of uncertain objects, is introduced. [6] considered the problem of indexing uncertain objects in $\mathbb{R}$ whose pdfs can be modeled as histograms, providing various indexing schemes with linear and near-linear space and logarithmic time.

In many applications the metric space is indeed a vector space. A vector space allows to use geometric and coordinate information which is unavailable in general metric spaces. Among the similarity search method proposed in the literature there are k-d-trees [18], R*-trees [19], and X-trees [20], and many others. Moreover, since these techniques make extensive use of coordinate information to group and classify points, they have an exponential dependency on the number of dimensions.

One of the main techniques to overcome the curse of dimensionality, and which is employed for indexing objects from a general metric space, is the *pivoting* based one [21], [22], [23], [8], [24], [10], [25]. At indexing time, a pivot based algorithm selects a certain number of objects, called *pivots*, and stores in the index all the pairwise distances among objects of the collection and pivots. Due to the reverse triangle inequality, given two generic objects x and y, their distance $d(x, y)$ cannot be smaller than $D_p(x, y) = |d(x, p) - d(p, y)|$, for any other object p. Hence, the value $D_p(x, y)$ is a lower bound for $d(x, y)$. With a set of k pivots $\mathcal{P} = \{p_1, \ldots, p_k\}$, a better lower bound $D_{\mathcal{P}}(x, y)$ to the distance $d(x, y)$ can be obtained as $\max_{1 \leq i \leq k} D_{p_i}(x, y)$. Note that the value of $D_{\mathcal{P}}(x, y)$ is computed by exploiting only the pre-calculated distances among the objects x and y and all the pivots, and without the need of computing the distance between x and y. At query time, first of all, the distances between the query object q and all the pivots are computed. Then, the set of *candidate* objects belonging to the actual query outcome are obtained by selecting only the objects x such that $D_{\mathcal{P}}(q, x) \leq R$. Unfortunately, by using this strategy some spurious objects may be captured, namely objects x such that $D_{\mathcal{P}}(q, x) \leq R$ but $d(q, x) > R$. Hence, the true neighbors of q are eventually retrieved by a *filtering phase* consisting in computing the actual distances among q and each candidate object. Ideally, the set of candidate objects should coincide with the outcome of the query. However, minimizing the number of the spurious objects is a hard challenge. Usually, the greater the number of pivots, the smaller the number of spurious objects in the candidate set [24], [25], [26].

Given a set of uncertain objects DS, an *uncertain range query* of center q, radius R, and probability threshold τ, retrieves all the uncertain objects of DS that lie within distance R from q with probability at least τ, that are the objects x of DS such that $Pr(d(x, y) \leq R) \geq \tau$ holds. In the setting considered here the center q of the query may be either a certain or an uncertain object.

As a major contribution, to the best of our knowledge this is the first work providing an *effective technique for indexing uncertain objects coming from general metric spaces*, while previously introduced techniques are applicable only to objects coming from a vector space. In particular, we introduce a novel indexing technique, called UP–index, which makes use of pivots in order to index uncertain data. Next, we summarize the other contributions of the work:

- we generalize the reverse triangle inequality to the probabilistic setting in order to exploit it as a discard condition, that is to establish if $Pr(d(x, y) \leq R) < \tau$, without the need of computing the actual value of the probability $Pr(d(x, y) \leq R)$;
- we introduce UP–index and show how it can be employed to speed up range query computation on uncertain data. Importantly, the candidate selection phase of our technique is able to noticeably reduce the set of candidates with little time requirements. In particular, to compute the probability $Pr(d(x, q) \leq R)$, and hence to decide if x belongs to the answer of the query, a $(2m)$-dimensional integral has to be evaluated. In contrast, our discard condition costs only $O(h)$ elementary operations, for h a fixed small value, and hence it is independent of the data dimensionality. This means that our filtering phase permits to save a vast amount of time;
- we conduct extensive experiments to validate the proposed approach. In particular, we analyzed scalability with respect to data set size and dimensionality, performances on real scattered data with respect to the degree of uncertainty associated with data, and the behavior on non vector spaces, specifically on a string domain. The experiments pointed out that our method is able to greatly reduce the number of candidate objects and to significantly improve time performances, and that the

technique exhibits effectiveness also on the string domain. Moreover, we compared the UP-index with indexing techniques specifically designed to perform range queries on uncertain objects in the multi-dimensional Euclidean space. The UP-index revealed itself as more efficient and effective than competitors and also to be preferable when the dimensionality of the space increases;
- we provide a criterion to measure the quality of a set of pivots and study the problem of selecting a good set of pivots, reporting some intractability results. Specifically, we prove that selecting a set of pivots that minimizes the expected error in a general metric space is NP-hard, and, moreover, we generalize the result to the Euclidean space. We note that this result is still valid for certain data, for which the study of the complexity of the problem of selecting a good set of pivots has been neglected in the literature. Also, we introduce an estimation algorithm with statistical guarantees for selecting a good quality set of pivots and, then, experimentally show that the introduced criterion is effective in enhancing the performances of UP–index.

Summarizing, this work establishes the foundations for dealing with general metric spaces in the uncertain scenario and, under this perspective, we believe it opens other interesting research directions.

The rest of the work is organized as follows. In Section 2 the notion of distance between uncertain objects is introduced and some of its properties are studied. Section 3 details UP–index building and uncertain range query answering. Section 4 reports experimental results. Section 5 considers the problem of selecting an optimal set of pivots. Finally, Section 6 draws conclusions of the work.

2 Uncertain Distance

In this section, we deal with the distance between uncertain objects. We start by providing some preliminary definitions.

An *attribute* (or *dimension*) a is an identifier with an associated domain denoted as $\mathbb{D}(a)$. Given a set of attributes $A = \{a_1, \ldots, a_m\}$, $\mathbb{D}(A)$ denotes the domain associated with A, namely the set $\mathbb{D}(a_1) \times \cdots \times \mathbb{D}(a_m)$. Whenever the set of attributes is clear by the context the domain associated with A is denoted as $\mathbb{D}$.

A *certain object* (or *value*) v on A is an m-ple $\langle v_1, \ldots, v_m \rangle$, where $v_i \in \mathbb{D}(a_i)$. An *uncertain object* p on A is a random variable having domain $\mathbb{D}(A)$ with associated probability density function (*pdf* for short) f^p, where $f^p(v_1, \ldots, v_m)$ denotes the density for p in $\langle v_1, \ldots, v_m \rangle$, and cumulative distribution function (*cdf* for short) F^p. An *uncertain data set* DS on the domain $\mathbb{D}$ is a set of uncertain objects on $\mathbb{D}$.

In the following, we assume that $\mathbb{D}$ is a metric space, namely a space equipped with a distance function $d : \mathbb{D} \times \mathbb{D} \mapsto \mathbb{R}$ satisfying the following properties: non-negativity, symmetry, reflexivity, and triangle inequality.

Given an object v of $\mathbb{D}$, $\mathcal{B}_v^R$ denotes the set of values $\{w \in \mathbb{D} \mid d(w, v) \leq R\}$, namely the *hyperball* having center v and radius R, while $\mathcal{S}_v^R$ denotes the set of values $\{w \in \mathbb{D} \mid d(w, v) = R\}$, namely the *hypershpere* having center v and radius R.

Given two uncertain objects p and q, a non negative real number R, and a probability threshold τ, if it holds

$$Pr(d(p, q) \leq R) \geq \tau$$

then the objects p and q are at distance not greater than R with probability at least τ. Next, we show how the probability $Pr(d(p, q) \leq R)$ can be computed.

Definition 2.1: Given two uncertain objects p and q, we denote by $\Delta^{p,q}$ the continuous random variable representing the distance between p and q. In particular, for $R \geq 0$, the probability density function of $\Delta^{p,q}$ is:

$$f_\Delta^{p,q}(R) = \int_{\mathbb{D}} \int_{\mathcal{S}_v^R} f^p(v) f^q(w) \, dw \, dv, \tag{1}$$

which corresponds to the probability that the distance between p and q is R; while, for $R < 0$, $f_\Delta^{p,q}(R)$ evaluates to zero. Conversely, for $R \geq 0$, the cumulative distribution function is:

$$F_\Delta^{p,q}(R) = \int_{\mathbb{D}} \int_{\mathcal{B}_v^R} f^p(v) f^q(w) \, dw \, dv, \tag{2}$$

which corresponds to the probability that the distance between p and q is lower than or equal to R; while, for $R < 0$, $F_\Delta^{p,q}(R)$ evaluates to zero.
By definition of $F_\Delta^{p,q}$, it holds that

$$Pr(d(p, q) \leq R) \geq \tau \iff F_\Delta^{p,q}(R) \geq \tau.$$

Definition 2.1 is given for two uncertain objects p and q. If q is a certain object then Equation (1) reduces to

$$f_\Delta^{p,q}(R) = \int_{\mathcal{S}_q^R} f^p(v) \, dv, \tag{3}$$

and Equation (2) reduces to

$$F_\Delta^{p,q}(R) = \int_{\mathcal{B}_q^R} f^p(v) \, dv. \tag{4}$$

We note that the integrals in Equations (1) and (2) are $(2m)$-dimensional, while the integrals in Equations (3) and (4) are m-dimensional.

Next, we generalize the *reverse triangle inequality* to the introduced uncertain context.

Theorem 2.1 (Uncertain Reverse Triangle Inequality): For uncertain objects x, y and z, if $Pr(|d(x, z) - d(z, y)| \leq R) < \tau$ then $Pr(d(x, y) \leq R) < \tau$, for any R and τ.

Proof: In order to prove the theorem, it is enough to prove that for each R and for each x, y, z:

$$Pr(d(x, y) \leq R) \ \leq \ Pr(|d(x, z) - d(z, y)| \leq R).$$

Consider the second term of the inequality. It corresponds to the probability that x assumes value $u \in \mathbb{D}$, y assumes value $v \in \mathbb{D}$, and z assumes value $w \in \mathcal{A}$, where

$\mathcal{A}$ is the locus of objects w such that $|d(u,w) - d(w,v)| \leq R$. Namely:

$$Pr(|d(x,y) - d(y,z)| \leq R) = \int_{\mathbb{D}} \int_{\mathbb{D}} \int_{\mathcal{A}} f^x(u) f^y(v) f^z(w)\, dw\, dv\, du.$$

The domain $\mathbb{D}$ of v can be split in two components according to the value assumed by u. The first component concerns v ranging in the hyperball having center u and radius R, namely $\mathcal{B}_u^R$; whereas, the second component concerns v ranging outside the hyperball having center u and radius R, denoted as $\overline{\mathcal{B}_u^R}$. Then,

$$Pr(|d(x,y)-d(y,z)| \leq R) = \int_{\mathbb{D}} f^x(u) \int_{\mathcal{B}_u^R} f^y(v) \int_{\mathcal{A}} f^z(w)\, dw\, dv\, du + \\ + \int_{\mathbb{D}} f^x(u) \int_{\overline{\mathcal{B}_u^R}} f^y(v) \int_{\mathcal{A}} f^z(w)\, dw\, dv\, du.$$

Consider, now, the first term of the sum. If v lies in the hyperball $\mathcal{B}_u^R$, the distance between v and u is not greater than R. Since d is a metric, $|d(u,w) - d(w,v)| \leq d(u,v) \leq R$ for each w, then $\mathcal{A}$ in this case corresponds to $\mathbb{D}$. As a consequence, $\int_{\mathcal{A}} f^z(w)\, dw$ is the probability for z to belong to the whole domain and then it is equal to 1. Therefore, the previous Equation can be rewritten as:

$$Pr(|d(x,y) - d(y,z)| \leq R) = \int_{\mathbb{D}} f^x(u) \int_{\mathcal{B}_u^R} f^y(v)\, dv\, du + \\ + \int_{\mathbb{D}} f^x(u) \int_{\overline{\mathcal{B}_u^R}} f^y(v) \int_{\mathcal{A}} f^z(w)\, dw\, dv\, du.$$

Since, by Equation (2):

$$Pr(d(x,y) \leq R) = \int_{\mathbb{D}} \int_{\mathcal{B}_u^R} f^x(u) f^y(v)\, du\, dv,$$

it holds that

$$Pr(|d(x,z) - d(z,y)| \leq R) = \\ Pr(d(x,y) \leq R) + \int_{\mathbb{D}} f^x(u) \int_{\overline{\mathcal{B}_u^R}} f^y(v) \int_{\mathcal{A}} f^z(w)\, du\, dv\, dw.$$

Being the second term a probability, it is always non negative and, then, this concludes the proof. □

3 Indexing Uncertain Data

In this section, we introduce our discard condition (Section 3.1), detail the implementation of the UP–index (Section 3.2), analyze the temporal cost of building it and its space occupancy (Section 3.3), and the temporal cost of answering a range query (Section 3.4).

3.1 Range Queries and Discard Condition

Definition 3.1 (Uncertain Range Query): Given an uncertain data set DS, an uncertain object q called the *query object*, a distance value R, and a probability threshold τ, the *range query with center q and radius R* retrieves all the objects x in DS such that $Pr(d(x,q) \leq R) \geq \tau$.

In order to efficiently answer uncertain range queries in general metric spaces, we generalize the pivot based approach to the case of uncertain data. In particular, we exploit a set of certain objects (the *pivots*) in order to single out uncertain objects x in DS that do not belong to the answer of the uncertain range query without explicitly computing $Pr(d(x,q) \leq R)$.

Let p be a certain object, let x and q be two uncertain objects, and let

$$D_p(x,q) = |d(x,p) - d(p,q)|.$$

If $Pr(D_p(x,q) \leq R) < \tau$, then due to Theorem 2.1, it is known that the probability $Pr(d(x,q) \leq R)$ is lower than τ, without the need of computing the actual value of $Pr(d(x,q) \leq R)$.

Thus, the property stated in Theorem 2.1 can be exploited to discard objects. The condition

$$Pr(D_p(x,q) \leq R) < \tau,$$

is referred to as *discard condition* and is used to filter out data set objects.

Specifically, given a set $\mathcal{P} = \{p_1, \ldots, p_k\}$ of k certain objects, the *pivots*, the *candidate* objects are selected as the objects x of the data set such that

$$\forall p \in \mathcal{P},\ Pr(D_p(x,q) \leq R) \geq \tau.$$

Before detailing the implementation of the uncertain pivot based index, we show how the probability $Pr(D_p(x,q) \leq R)$ can be formulated in terms of the pdfs and the cdfs of x and q.

Theorem 3.1: The probability $Pr(D_p(x,q) \leq R)$ can be computed as

$$\int_0^\infty (F_\Delta^{x,p}(r+R) - F_\Delta^{x,p}(r-R)) \cdot f_\Delta^{q,p}(r)\, dr. \qquad (5)$$

Proof: We note that

$$Pr(D_p(x,q) \leq R) = Pr\big(|d(x,p) - d(q,p)| \leq R\big) = \\ = \int_0^\infty Pr\big(|d(x,p) - r| \leq R\big) \cdot Pr\big(d(q,p) = r\big)\ dr = \\ = \int_0^\infty Pr\big(r - R \leq d(x,p) \leq r + R\big) \cdot Pr\big(d(q,p) = r\big)\ dr = \\ = \int_0^\infty \Big(Pr\big(d(x,p) \leq r + R\big) - Pr\big(d(x,p) \leq r - R\big)\Big) \cdot \\ \cdot Pr\big(d(q,p) = r\big)\ dr.$$

Since $Pr\big(d(q,p) = r\big) = f_\Delta^{q,p}(r)$ and $Pr\big(d(x,p) \leq r\big) = F_\Delta^{x,p}(r)$, the result follows. □

Note that if q is a certain object, then the above integral reduces to the difference:

$$F_\Delta^{x,p}(d(q,p) + R) - F_\Delta^{x,p}(d(q,p) - R). \qquad (6)$$

3.2 Building the UP–index

Let $\mathcal{P}$ be the set of pivots. In order to check the discard condition, the integral reported in Equation (5) has to be computed.

We note that the function $F_\Delta^{x,p}$ depends only on the data set object x and the pivot p, and does not depend on the query object q. Thus, if we pre-compute the function $F_\Delta^{x,p}$, then the discard condition check can be accelerated. Indeed, the term $(F_\Delta^{x,p}(r+R) - F_\Delta^{x,p}(r-R))$ can be

obtained by using the pre-computed function $F_\Delta^{x,p}$, while it remains to evaluate the function $f_\Delta^{q,p}$, which depends only on the pivot p and the query object q. Since the function $f_\Delta^{q,p}$ does not depend on the data set objects, once the range query with center q is submitted, for each pivot p the function $f_\Delta^{q,p}$ can be pre-computed, and then used to check all the discard conditions.

Clearly enough, since both $F_\Delta^{x,p}$ and $f_\Delta^{q,p}$ are real functions, in general it is not possible to pre-compute and store all the values they assume. However, we note that, since we have to verify that $Pr(D_p(x,q) \leq R) < \tau$, if we knew an upper bound $\widehat{F}_\Delta^{x,p}(r)$ for $F_\Delta^{x,p}(r)$ and an upper bound $\widehat{f}_\Delta^{x,p}(r)$ for $f_\Delta^{q,p}(r)$, then the discard condition could be still correctly applied.

Thus, the idea is to provide for any uncertain object y a function $G^{y,p}$ which is easy to store and to be employed for computing both an upper bound of $f_\Delta^{y,p}$ and an upper bound of $F_\Delta^{y,p}$.

In particular, we first compute the histogram $g^{y,p}$, consisting of h slots, of the function $f_\Delta^{y,p}$, where h is a fixed parameter. In order for $g^{y,p}$ to represent an upper bound to the function $f_\Delta^{y,p}$, in each slot the maximum value assumed by $f^{y,p}$ in the interval associated with the slot is stored. Then, the function $G^{y,p}$ is obtained as the cumulative histogram of the function $g^{y,p}$.

Next, we detail the building of the function $G^{y,p}$ and then show how to use it in order to compute an upper bound of both $F_\Delta^{y,p}(r)$ and $f_\Delta^{y,p}(r)$.

Let p be a pivot and y be an uncertain object. Let r_m and r_M denote, respectively, the minimum and maximum value for the distance between p and y, and let $s = \frac{(r_M - r_m)}{h}$, where h is the number of histogram slots.

In particular, the values r_m and r_M can be obtained from the pdf of y, by considering the region which y belongs to with non-negligible probability. With this aim, w.l.o.g. it is assumed that each uncertain object y is associated with a finite region $\mathrm{SUP}(y)$, containing the support of y, namely the region such that $Pr(y \notin \mathrm{SUP}(y)) = 0$ holds. For example, $\mathrm{SUP}(y)$ could be defined as an hyper-ball or an hyper-rectangle.

If the support of y is infinite, then $\mathrm{SUP}(y)$ is such that $Pr(y \notin \mathrm{SUP}(y)) \leq \pi$, for a fixed small value π, and the probability for y to exist outside $\mathrm{SUP}(y)$ is considered negligible. In this case the error ε involved in the calculation of the probability $Pr(d(x,y) \leq R)$, with x and y two uncertain objects, is the square of π.

For example, assume that the data set objects y are normally distributed with mean μ_y and standard deviation σ_y. If the region $\mathrm{SUP}(y)$ is defined as $[\mu_y - 4\sigma_y, \mu_y + 4\sigma_y]$ then the probability $\pi = Pr(y \notin \mathrm{SUP}(y))$ is $\pi = 2 \cdot \Phi(-4) \approx 0.00006$ and the maximum error is $\varepsilon = \pi^2 \approx 4 \cdot 10^{-9}$.

Figure 1 reports an example of $f_\Delta^{y,p}$ function (dotted line). We considered objects in $\mathbb{R}$; the object p is 0 and the uncertain object x is distributed according to a normal distribution with $\mu = 4$ and $\sigma = 1$, over the interval $[0, 8]$. Thus, $r_m = 0$, $r_M = 8$. The dashed line shows the function $g^{y,p}$ with $h = 8$ slots and, hence, $s = 1$.

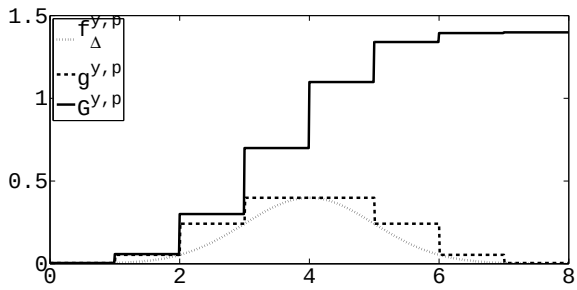


Fig. 1: $f_\Delta^{y,p}$ and its upper bound

The value $g^{y,p}$ in the generic slot i ($1 \leq i \leq h$) is obtained as $g^{y,p}(i) = \max_{r \in \mathcal{I}_i} f_\Delta^{y,p}(r)$ where $\mathcal{I}_i$ denotes the interval $(r_m + s \cdot (i-1), r_m + s \cdot i]$. [1]

Then, the histogram $G^{y,p}$ is built by associating with each slot i ($1 \leq i \leq h$) the value

$$G^{y,p}(i) = \sum_{j=1}^{i} g^{y,p}(j).$$

In the following, we assume that $G^{y,p}(0) = 0$. The values of $G^{y,p}$ for the example function are reported in Figure 1 with a solid line.

An upper bound $\widehat{f}_\Delta^{y,p}(r)$ of $f_\Delta^{y,p}(r)$ is then obtained as:

$$\widehat{f}_\Delta^{y,p}(r) = G^{y,p}(i) - G^{y,p}(i-1),$$

where $i = \lceil \frac{r - r_m}{s} \rceil$ (with $r \in (r_m, R_M]$) is the slot such that $r \in \mathcal{I}_i$, while an upper bound $\widehat{F}_\Delta^{y,p}(r)$ of $F_\Delta^{y,p}(r)$ is given by:

$$\widehat{F}_\Delta^{y,p}(r) = s \cdot G^{y,p}(i-1) + (G^{y,p}(i) - G^{y,p}(i-1)) \cdot (r - (r_m + s \cdot (i-1))).$$

Before concluding, we note that the cost to compute both $\widehat{f}_\Delta^{y,p}$ and $\widehat{F}_\Delta^{y,p}$ by means of $G^{y,p}$ is $O(1)$.

Hence, an upper bound to the probability $Pr(D_p(x,q) \leq R)$ of Equation (5) can be obtained by computing the following integral:

$$\int_0^\infty (\widehat{F}_\Delta^{x,p}(r+R) - \widehat{F}_\Delta^{x,p}(r-R)) \cdot \widehat{f}_\Delta^{q,p}(r)\ dr. \qquad (7)$$

Moreover, the following property holds.

Proposition 3.2: The exact value of Equation (7) can be computed by performing $O(h)$ operations.

Proof: We exploit the fact that functions $\widehat{F}_\Delta^{x,p}$ and $\widehat{f}_\Delta^{q,p}$ are stored in the histograms $G^{x,p}$ and $G^{q,p}$ of h slots each. Let r_m^q and r_M^q (r_m^x and r_M^x, resp.) be the minimum and the maximum value of the distance between p and q (p and x, resp.), and let $s^q = \frac{r_M^q - r_m^q}{h}$ and $s^x = \frac{r_M^x - r_m^x}{h}$.

1. Assume that the interval $\mathcal{I}_1$ includes also the left endpoint r_m.

We can write:

$$\int_{r_m^q}^{r_M^q} (\widehat{F}_\Delta^{x,p}(r+R) - \widehat{F}_\Delta^{x,p}(r-R)) \cdot \widehat{f}_\Delta^{q,p}(r)\ dr =$$
$$= \sum_{i=1}^{h} (G^{q,p}(i) - G^{q,p}(i-1)) \int_{r_{i-1}}^{r_i} (\widehat{F}_\Delta^{x,p}(r+R) - \widehat{F}_\Delta^{x,p}(r-R))\ dr =$$
$$= \sum_{i=1}^{h} (G^{q,p}(i) - G^{q,p}(i-1)) \cdot$$
$$\cdot \left(\int_{r_{i-1}}^{r_i} \widehat{F}_\Delta^{x,p}(r+R)\ dr - \int_{r_{i-1}}^{r_i} \widehat{F}_\Delta^{x,p}(r-R)\ dr \right).$$

where r_i is $r_m^q + s^q \cdot i$.

If we knew the primitive $\mathcal{F}^{x,p}$ of the function $\widehat{F}^{x,p}$, then we could compute the integral

$$\int_a^b \widehat{F}^{x,p}(r)\ dr$$

as $\mathcal{F}^{x,p}(b) - \mathcal{F}^{x,p}(a)$. Thus, let $H^{x,p}$ be the cumulated histogram associated with $\widehat{F}^{x,p}$, namely

$$H^{x,p}(i) = s^x \sum_{j=1}^{i} G^{x,p}(j),\ (1 \leq i \leq h),$$

which can be computed in $O(h)$ time. Then, the function $\mathcal{H}^{x,p}$ can be defined as

$$\mathcal{F}^{x,p}(r) = s^x \cdot H^{x,p}(i-1) + (H^{x,p}(i) - H^{x,p}(i-1)) \cdot (r - (r_m^x + s^x \cdot (i-1)))$$

and can be computed in $O(1)$ time.

We can finally express Equation (7) as the following summation

$$\sum_{i=1}^{h} (G^{q,p}(i) - G^{q,p}(i-1)) \cdot (\mathcal{F}^{x,p}(r_i + R) - \mathcal{F}^{x,p}(r_{i-1} + R) + -\mathcal{F}^{x,p}(r_i - R) + \mathcal{F}^{x,p}(r_{i-1} - R)),$$

composed of h terms, which can be computed by performing $O(h)$ operations. □

3.3 Cost of building the UP–index

In this section, we analyze the spatial and the temporal cost of building the UP–index. Let $\mathcal{P}$ be the set of pivots, let h be the number of histogram slots and let m be the number of dimensions of the domain $\mathbb{D}$.

As for the space occupancy, the UP–index stores the data set DS, the set of pivots $\mathcal{P}$, and, for each pair $x \in DS$ and $p \in \mathcal{P}$ of objects, the cumulative histogram $H^{x,p}$. Each cumulative histogram can be encoded by means of $h+2$ real numbers, that are the two endpoints r_m and r_M, and the values of the histogram in the h slots considered. Overall, to store all the cumulative histograms, $|DS| \cdot |\mathcal{P}| \cdot (h+2)$ floating point numbers are needed.

As far as the temporal cost is concerned, first of all, the cost to evaluate the integral of a multivariate function has to be provided. For a generic function f of m variables, it is known from the *information based complexity theory* [27] that computing a m-dimensional integral has, in the worst case setting, a cost that exponentially depends on the number of variables m. In particular such a cost is $O(\varepsilon^{-m})$, where ε is the maximum error admitted in computing the integral. Nevertheless, when the function f satisfies some properties the curse of dimensionality in the average case setting can be broken [27]. In the following we refer to the cost of computing a m-dimensional integral as $C_{int}(m)$.

In order to build the UP–index, for each uncertain object x in DS and for each pivot p, the histogram $G^{x,p}$ has to be computed. Each histogram can be computed with only one single multidimensional integration by exploiting the following strategy. In order to compute a numerical approximation of the function $F_\Delta^{x,p}$ it is possible to proceed to the integration of the function $f^x(r)$ on the whole domain of x. This corresponds to partitioning the region $\mathcal{R}$ occupied by the pdf f^x in a set $\mathcal{R}_1, \ldots, \mathcal{R}_N$ of (sub-)regions, and then computing the integral value as the summation $\sum_i s_i$, where s_i denotes the volume of the region $\mathcal{R}_i$. During the above integration, the terms s_i pertaining to regions $\mathcal{R}_i$ located at distance not greater than r from p are then accumulated to obtain the value $F_\Delta^{x,p}(r)$. The function $f_\Delta^{x,p}$ can eventually be obtained by exploiting $F_\Delta^{x,p}$. Importantly, the above strategy is independent of the numerical integration method used to calculate integrals.

This leads to the following overall cost for building the index:

$$O\Big(|DS| \cdot |\mathcal{P}| \cdot C_{int}(m)\Big),$$

which does not depend on the resolution h of the histogram.

Finally, we point out that the cost of *inserting* a novel uncertain object x into the UP–index corresponds to the cost of computing the histogram $G^{x,p}$ for each $p \in \mathcal{P}$, and then it is $O(|\mathcal{P}| \cdot C_{int}(m))$. Conversely, the cost of *removing* an uncertain object x from the UP–index corresponds to the cost of removing the $|\mathcal{P}|$ histograms $G^{x,p}$ associated with x, and then it is $O(1)$.

Moreover, the cost of *adding a pivot* p to the UP–index is $O(|DS| \cdot C_{int}(m))$, since it amounts to computing the histogram $G^{x,p}$ for each $x \in DS$. Vice versa, the cost of *deleting a pivot* is $O(1)$, since it amounts to removing the $|DS|$ histograms $G^{x,p}$ associated with p.

3.4 Answering a range query

Assume that the UP–index has been built. A range query with center q and radius R is answered by performing the following steps:

1) (*Initialization*) First, for each pivot p in $\mathcal{P}$, the function $G^{q,p}$ is computed;
2) (*Candidate selection phase*) Then, the candidate objects are selected by evaluating the discard condition. In particular, for each uncertain data set object x, it is checked if the discard condition holds for some pivot p. If it is not the case, x is a *candidate* object;
3) (*Filtering phase*) Finally, for each candidate object x, it is checked whether $Pr(d(x,q) \leq R) < \tau$ holds or not. The query answer consists of the candidate objects x satisfying $Pr(d(x,q) \leq R) \geq \tau$.

The cost to answer a range query is given by: (1) the cost to compute $G^{q,p}$ for each pivot p, (2) the cost to evaluate $Pr(D_p(x,q) \leq R)$ for each object x, and (3) the cost to evaluate $Pr(d(x,q) \leq R)$ for the candidate objects.

The cost of computing $G^{q,p}$ for each $p \in \mathcal{P}$ is $O(|\mathcal{P}| \cdot C_{int}(m))$, as shown in the previous section.

In order to check the discard condition, the upper bound of the integral (5) reported in Equation (7) has to be determined. As stated in Proposition 3.2, this value can be computed in $O(h)$ time.

As for the cost of computing $Pr(d(x,q) \leq R)$, it follows from the discussion in Section 2 that it corresponds to the cost of evaluating a $(2m)$-dimensional integral, namely $C_{int}(2m)$.

Thus, let $cands$ denote the cardinality of the set of candidates, the cost of answering a range query is then:

$$O\Big(\underbrace{|\mathcal{P}| \cdot C_{int}(m)}_{\textit{Initialization}} + \underbrace{|\mathcal{P}| \cdot |DS| \cdot h}_{\textit{Candidate selection phase}} + \underbrace{\textit{cands} \cdot C_{int}(2m)}_{\textit{Filtering phase}}\Big).$$

Clearly, due to the exponential dependence with m of C_{int} in the worst case, the most expensive term of the cost is the last one, that is $\textit{cands} \cdot C_{int}(2m)$. We note that the first two terms correspond to the cost of selecting the set of candidates. This cost is negligible with respect to the cost of the filtering phase. Hence, the candidate selection phase permits to select the (usually small) set of candidates with vast time savings. Compare the above cost, with the cost of the naive brute-force approach to answer a range query, that is $O(|DS| \cdot C_{int}(2m))$.

To conclude, consider a certain query object q. In this case the cost of answering a range query reduces to

$$O\Big(\underbrace{|\mathcal{P}| \cdot C_{dist}}_{\textit{Initialization}} + \underbrace{|\mathcal{P}| \cdot |DS|}_{\textit{Candidate selection phase}} + \underbrace{\textit{cands} \cdot C_{int}(m)}_{\textit{Filtering phase}}\Big),$$

where C_{dist} denotes the cost of computing the distance between two certain objects. Indeed, note that evaluating the integral in Equation (7) for q a certain object reduces to compute the difference between the two real values reported in Equation (6).

4 Experiments

In this section we present results obtained by experimenting the UP–index. The various experiments conducted and their goals are detailed in the following.

Given a query object q and a generic data set object x, if the minimum distance between q and x is greater than R, then x does not belong to the answer of the range query centered in q, and x is called a *far* object, otherwise x is called a *near* object. We denote by n_{near} the number of data set objects which are near.

In order to quantify the savings obtained by using UP–index, we make use of two measures, called *Gain* and *Time gain*, which are detailed next.

The *Gain* measure is defined as

$$\textit{Gain} = 1 - \frac{cands - neighs}{n_{near} - neighs},$$

where $cands$ denotes the size of the set of candidates, that are the uncertain objects which have not been discarded by the candidate selection phase, and $neighs$ denotes the size of the solution set, that is the answer of the range query. Intuitively, if the set of candidates selected by using UP–index coincided with the true outcome of the query, then the $Gain$ would be maximum, that is equal to 1. Differently, the term $\frac{cands-neighs}{n_{near}-neighs}$ represents the (relative w.r.t. the number of objects which are located near the query) number of distances that are required to be computed during the filtering phase in order to discard the candidates that do not belong to the answer of the query.

The *Time gain* measure is defined as

$$\textit{Time gain} = 1 - \frac{t_{pivots} - t_{neighs}}{t_{near} - t_{neighs}},$$

where t_{pivots} denotes the time employed by UP–index to answer the range query, t_{near} denotes the time employed to compute the uncertain distances between the query object and the n_{near} non-far objects, and t_{neighs} is the time needed to compute the uncertain distances between the query object and the $neighs$ objects.

The difference between the gain and the time gain is that the latter measure depends on the implementation, and, particularly, on the number of pdf evaluations employed during integral computations, while the former measure depends only on the number of dominating operations to be performed, that are the multi-dimensional integral computations.

In the experiments we exploited the *Montecarlo* integration method [28], [29].[2] The resolution h of the histograms was set to 100. The parameter τ was set to 0.75. Moreover, pivots were selected by picking at random some data set objects x and then taking the mean value $\overline{x}$ of each of them. Curves shown are then obtained by averaging results on ten runs.

In order to test the method, for each data set considered, one thousands uncertain objects were randomly generated as query objects according to the law used to generate the other data set objects. Since query objects are uncertain, $(2m)$-dimensional integrals have to be evaluated in order to compute distance probabilities.

The *selectivity* of a range query is the percentage of data set objects which belong to the answer of the query, namely the ratio $100 \cdot \frac{neighs}{n}$, where n is the data set size.

4.1 Scalability with respect to data set size and dimensionality

Experiments described in this section (see Figure 2) are designed to study the scalability with respect to the selectivity and the number of pivots employed and, moreover, the performances of the method with respect to the data dimensionality.

2. The number of pdf evaluations to compute m-dimensional integrals was set to $2^{m+1} \cdot 1{,}000$.

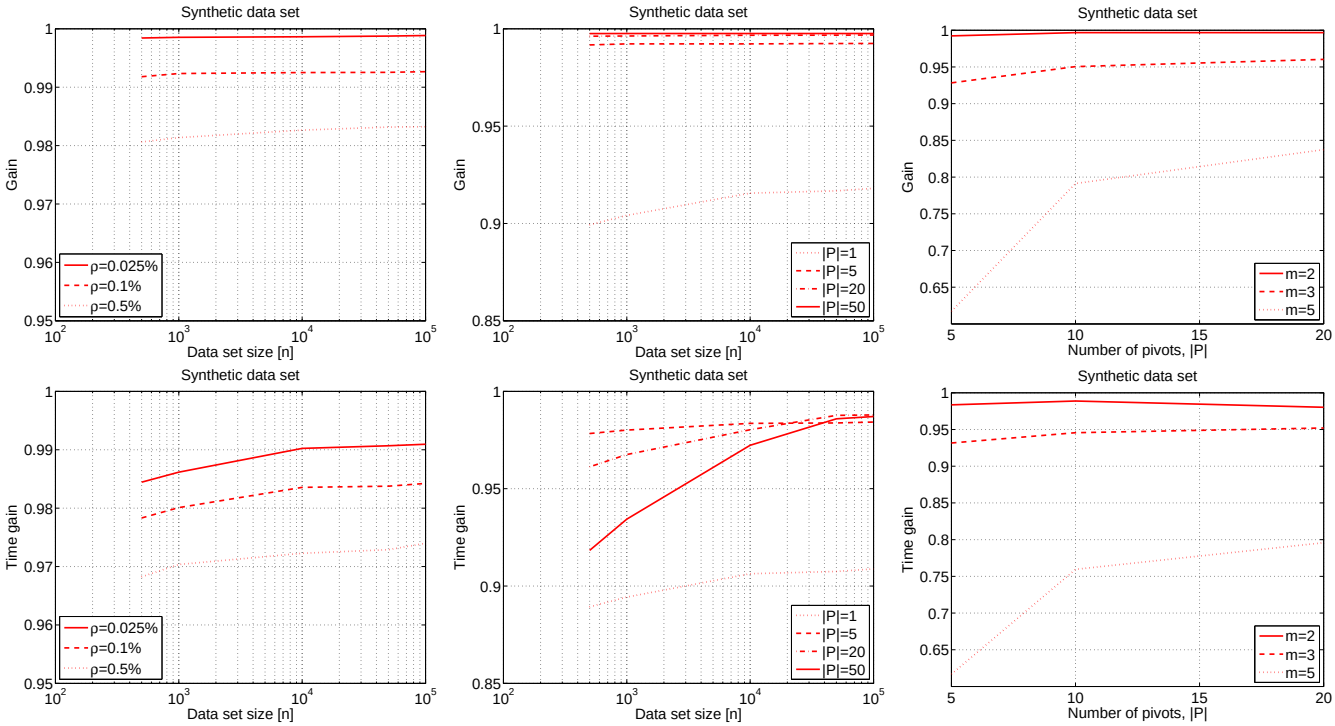


Fig. 2: Experiments on synthetic data sets.

With this aim, we employed the *Synthetic* data set family, consisting of m-dimensional uncertain objects in $\mathbb{R}^m$. Each uncertain object x of the data set has associated the pdf $f^x(v_1, \ldots, v_m) = f_1^x(v_1) \cdot \ldots \cdot f_m^x(v_m)$, where each f_i^x is either a uniform or a normal pdf having mean μ_i, with μ_i randomly selected in the interval $[-10, +10]$, and support $[\mu_i - r_i, \mu_i + r_i]$, with r_i a randomly selected number in $(0.01, r_{\max}]$ (for f_i^x being a normal pdf, its standard deviation is equal to $r_i/4$). The value $r_{\max}$ is such that the maximum volume of the domain associated with a pdf corresponds to the 1% of the volume of the domain $[-10, +10]^m$. We considered data sets up to $100{,}000$ objects and to 5 dimensions.

Selectivity. The first column of Figure 2 shows how the gain and the time gain depend on the selectivity when the number of pivots $|\mathcal{P}|$ is held fixed (specifically $|\mathcal{P}|$ was set to 5). In this experiment the number of dimensions m was set to two. We varied the parameter R in a suitable range and measured the corresponding selectivity ρ. The plot on the first row shows the gain for ρ equal to 0.025% (upper curve), 0.1% (middle curve), and 0.5% (lower curve). The smaller the selectivity, the higher the gain: In all cases the gain is very high, being greater than 0.98 for $\rho = 0.5\%$, and even better in the other cases. The plot on the second row shows the corresponding time gain. The time gain is generally smaller than the gain, since while the gain takes into account the multidimensional integral evaluations needed to compute distance probabilities, the time gain depends also on the other operations performed by the UP–index in order to answer the range query. The curves show that despite the time gain is smaller than the gain, it anyway remains considerable, practically above 0.97 for $\rho = 0.5\%$. As far as the absolute execution time is concerned[3], the time required by UP-index to answer a query on the $100{,}000$ sized data set is 1.3 seconds for $\rho = 0.025\%$, 3.1 seconds for $\rho = 0.1\%$, and 8.2 seconds for $\rho = 0.5\%$, while the brute force method requires 144.8 seconds for $\rho = 0.025\%$, 170.5 seconds for $\rho = 0.1\%$, and 215.1 seconds for $\rho = 0.5\%$.

Pivots. The second column of Figure 2 shows how the gain and the time gain depend on the number of pivots employed when the selectivity is held fixed (specifically, in this experiment ρ was set to 0.1%). Also in this experiment the number of dimensions m was set to two. The plots report the gain and time gain for $|\mathcal{P}| = 1$, $|\mathcal{P}| = 5$, $|\mathcal{P}| = 20$, and $|\mathcal{P}| = 50$. These plots show that the greater the number of pivots employed, the higher the gain. Noticeably, the method performs well also when only one pivot is employed. For $|\mathcal{P}| = 1$, the gain (time gain, resp.) is above 0.9 (0.85, resp.). For greater number of pivots, the gain further improves, being about 0.998 for $|\mathcal{P}| = 50$. As far as the gain is concerned, performances of the method when the number of pivots is held fixed appear to be little sensitive to the data set size. Differently, as for the time gain, there is a trade off between the number of pivots employed and the size of the data set. Indeed, for $n = 10{,}000$, among the different values of $|\mathcal{P}|$ reported in figure, the best time gain is attained for $|\mathcal{P}| = 5$, while for $n = 100{,}000$, the time

3. Experiments have been performed on a Intel Xeon 2.33GHz based computer.

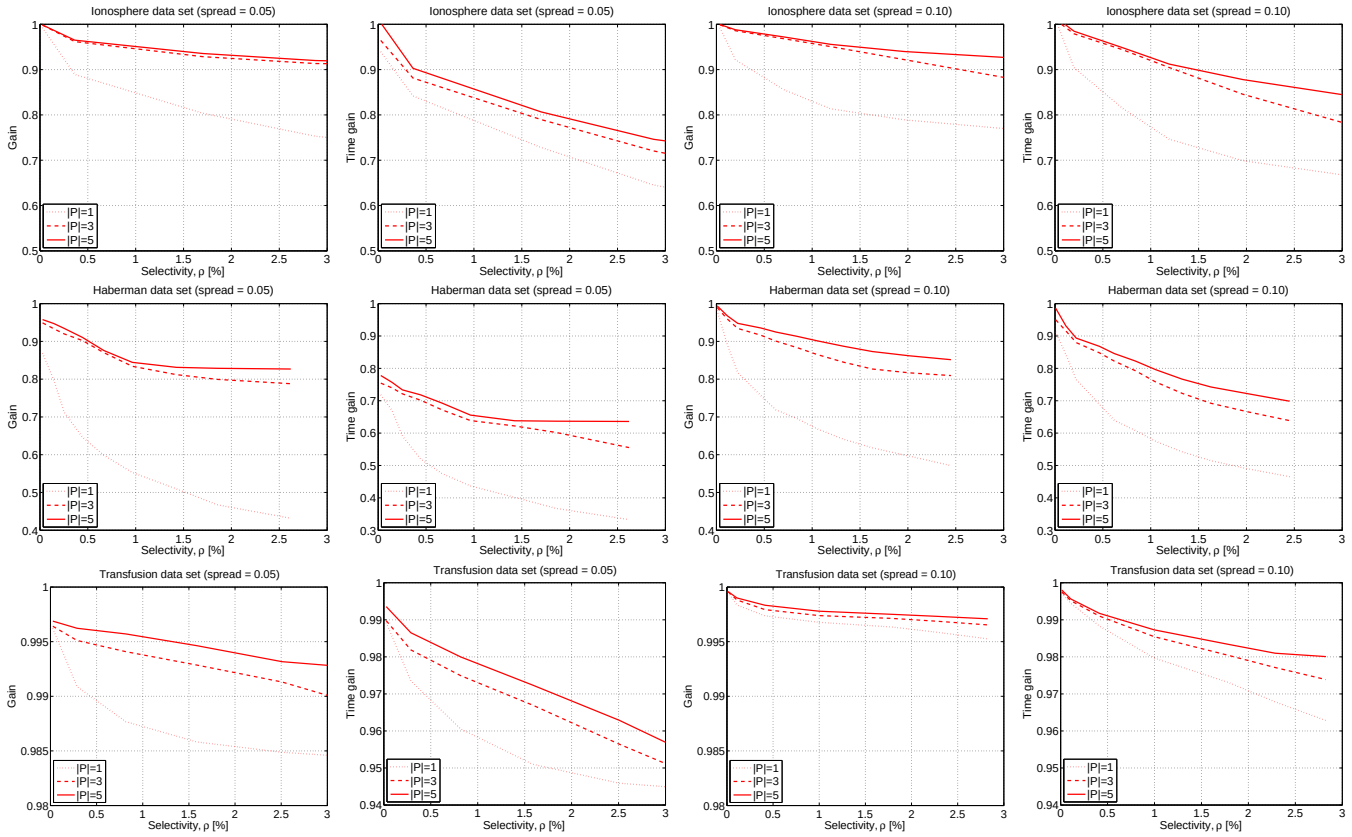


Fig. 3: Sensibility to data uncertainty.

gain performs better for $|\mathcal{P}| = 20$. When $|\mathcal{P}| = 5$ pivots are employed, the time gain worsens for small data set sizes (it is about 0.92 for $n = 500$), but rapidly increases for larger ones. Moreover, the associated time gain is expected to surpass the others for million-sized data sets. Thus, the larger the data set size, the larger the number of pivots that can be profitably employed to improve both the gain and the time gain of the method.

Dimensionality. The third column of Figure 2 shows how the gain and the time gain depend on the number of pivots employed when the number of dimensions m is varied. Specifically, in this experiment the data set size was set to $10{,}000$, the dimensionality m ranges from 2 to 5, and results shown are those corresponding to selectivity ρ approximatively equal to 1%. The number $|\mathcal{P}|$ of pivots varies from 5 to 20.

The plots confirm that the gain increases with the number of pivots employed. However, they also highlight that the greater the data dimensionality the greater the advantage of having more pivots. The time gain is informative. Indeed, while for $m = 2$ the gain increases a few for $|\mathcal{P}| \geq 10$, the corresponding time gain is slightly decreasing, as already observed in the previous experiment concerning the number of pivots employed. The time gain worsens since the additional cost paid to evaluate the discard condition on the augmented set of pivots is not rewarded by a decrease of the number of candidates (as just witnessed by the almost stationary gain).

As for $m = 3$, the slight increase of the gain is compensated by the time spent to evaluate the discard condition on the additional pivots, and, hence, the time gain tends to stabilize in the interval $k \in [10, 20]$. Finally, as for $m = 5$, the gain is always sensibly increasing and, hence, the time gain is increasing as well up to $k = 20$.

It is clear from the time gain that in the uncertain setting the pivots may produce time savings even when their number exceeds the dimensionality m of the data domain. We recall that in the literature the temporal cost associated with pivot based indexes is usually expressed in terms of number of distances computed in order to filter out candidate objects that are not neighbors of the query object. This is a fair assumption only in settings where distance computations are expensive, which is actually the setting considered in this paper.

In particular, it must be noted that in the certain case in order to take advantage of pivots when the m-dimensional real-valued domain $\mathbb{R}^m$ is considered, it should be guaranteed that each query object q is compared with a number k_q of pivots which is smaller than the actual number of dimensions m. Indeed, compare the cost of evaluating the discard condition for the certain object x_j:

$$\min_{i=1}^{k_q} |d(q, p_i) - d(p_i, x_j)|,$$

where $d(q, p_i)$ and $d(p_i, x_j)$ are pre-computed real values,

with the cost of computing the actual distance between q and x_j:

$$\left[\sum_{i=1}^{m} |q_i - x_{j,i}|^t\right]^{\frac{1}{t}},$$

where q_i and $x_{j,i}$ denote the i-th coordinate value of q and x_j, respectively, and t denotes the Minkowski's metric of interest ($t > 0$).

Interestingly, as pointed out beforehand, the above limitation does not hold when pivots are employed to answer range queries on uncertain objects even when the domain is $\mathbb{R}^m$.

4.2 Sensibility to data uncertainty

The experiments presented in the following are designed to study the performances of the method on real scattered data with respect to the selectivity and the number of pivots, and to the degree of uncertainty associated with data set objects.

The data sets considered are from the UCI ML Repository [30]: *Ionosphere* is a two dimensional real-valued data set composed of 351 objects, which has been obtained by projecting the ionosphere data set on the two principal components, *Haberman* is a three dimensional real-valued data set composed of 306 objects, and *Transfusion* is a four dimensional real-valued data set composed of 748 objects.

For each data set above listed, a family of uncertain data sets has been obtained. Each data set is characterized by a parameter, called *spread*, used to determine the degree of uncertainty associated with data set objects. In particular, with each certain object $x_i = (x_{i,1}, \ldots, x_{i,m})$ in the original data set, an uncertain object x_i' having pdf $f^i(v_1, \ldots, v_m) = f_1^i(v_1) \cdot \ldots \cdot f_m^i(v_m)$ is associated. Each one dimensional pdf f_j^i is randomly set to a normal or a uniform distribution, with mean $x_{i,j}$ and support $[a, b]$ depending on the value of the spread. In particular, let r be a randomly generated number in the interval $[0.01s\sigma_j, s\sigma_j]$, where σ_j denotes the standard deviation of the data set along the jth coordinate, then $a = x_{i,j} - 4 \cdot r$ and $b = x_{i,j} + 4 \cdot r$.

Figure 3 reports experiments on the data sets *Ionosphere* (first row), *Haberman* (second row), and *Transfusion* (third row). In particular, in order to determine the behavior of the method in correspondence of various levels of uncertainty in the data, we considered two different values of spread, namely 0.05 and 0.10. Moreover, we considered selectivities ρ up to about 3% by varying the radius R in a suitable range.

The curves displayed confirm that both the gain and the time gain are directly proportional to the selectivity and to the number of pivots employed. For $|\mathcal{P}| = 5$ and spread 0.10, when the maximum value of selectivity is considered, the gain is always remarkable, approximately 0.925 for *Ionosphere*, 0.850 for *Haberman*, and 0.997 for *Transfusion*. Though the larger the number of pivots, the better the method performances, interestingly, curves

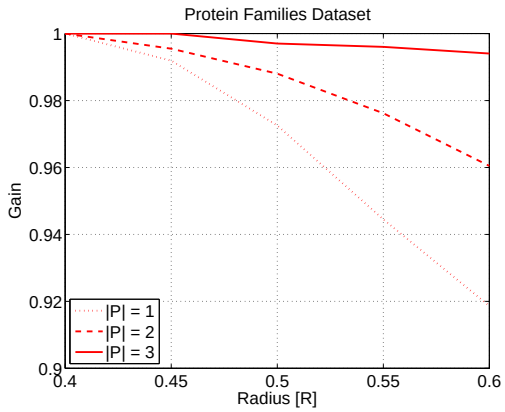


Fig. 4: Experiments on the *Protein Families* data set.

show that even when only one single pivot is considered a sensible improvement is achieved.

Moreover, it can be observed that curves for spread 0.10 are above those for spread 0.05, since the (time) gain increases with data uncertainty. This is due to the fact that when the uncertainty increases, the number of near objects increase as well. Thus, the higher the uncertainty, the larger the number of multi-dimensional integrals in charge of the brute force algorithm.

Also, noticeably, the method performs better on the higher dimensional data set here considered, that is *Transfusion*. This can be explained by noticing the following fact. It is well known that for high dimensional data the variance associated with pairwise distances tends to decrease; hence, the number n_{near} of near objects tends to increase with the dimensionality. This accounts for the increased computational effort (number of multidimensional integrals to be evaluated) in charge of the brute force method. Nonetheless, the very high value of gain reveals that the UP–index is able to sensibly reduce the number of objects $cands$ to be considered as candidate neighbors.

To conclude, it can be observed that the time gain tends to the gain when the dimensionality increases. Indeed, the higher the dimensionality, the smaller the cost of evaluating the discard condition as compared with the cost of computing the distance probability.

4.3 Behaviour on a non-vector space

The experiment detailed in this section has been designed to test the UP–index on a general (non-vector) metric space. Specifically, we analyzed the performances of the method on a uncertain string domain. The metric employed to measure string similarity is the Edit distance [11], defined as the minimum number of edits (i.e. insertions, deletions, or substitutions of a single character) needed to transform one string into the other.

The experiment was conducted on a data set from the PFAM database [31], a large collection of families of protein domains, which are amino acid sequences representing functional regions of a protein. For each domain

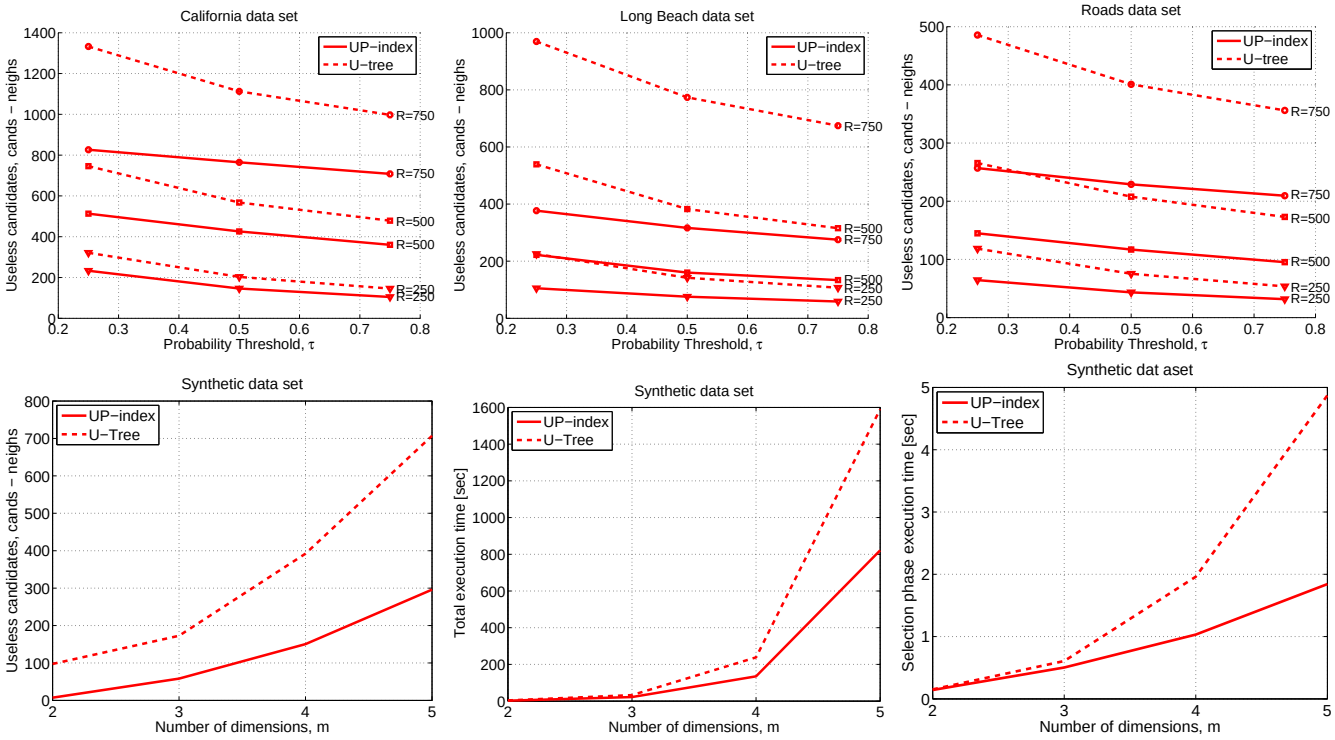


Fig. 5: Comparison between the UP-index and the U-tree.

family, the PFAM database stores a set of representative sequences which contain "don't care" symbols denoting unknown or unimportant amino acids.

Specifically, we considered a data set consisting of about three thousands sequences coming from different domains. Amino acid symbols in the original sequences have been replaced by associated nucleotide sequences, thus obtaining a set of sequences on the alphabet A, C, G, U, plus the "don't care" symbol. In order to simulate uncertain strings, each "don't care" symbol has been modelled as a discrete random variable whose outcomes are the symbols of the alphabet with probabilities set at random.

Each query object is generated by randomly selecting a sequence from the data set and by perturbing it. We varied the search radius R, which corresponds to the similarity value according to the normalized Edit distance, from 0.4 to 0.6 (for larger R, almost all the objects are to be returned as neighbors), and measured the gain of the method. We were not able to measure the time gain since the brute force method was too slow. Practically, there was an enormous difference between the execution time of the pivot based index and the brute force method.

Figure 4 reports the result of the experiment. We varied the number of pivots from 1 to 3. Three pivots guaranteed a very high gain in all cases, though also a smaller number of pivots offered good time savings. The behavior of the method was very similar to that observed in previous experiments, thus confirming the effectiveness of the UP–index in general metric spaces.

4.4 Comparison with the U-tree index

In this section, we compare UP-index with the U-tree indexing technique [2], [4] which has been specifically designed to perform range queries on uncertain objects in the multi-dimensional Euclidean space.[4]

The uncertain data sets *California* ($62{,}556$ objects), *Long Beach* ($53{,}145$ objects), and *Roads* ($30{,}674$ objects) have been obtained from the homonym spatial data sets[5] as described in [2]. In particular, the first two data set have been already employed in [2] to experiment the U-tree index. All the data sets consist of *Constrained-Gaussian* uncertain objects, that are Gaussian-like distributions having a finite domain, whose pdf is reported in [2].[6] Results are averaged on one hundred uncertain queries.

Figure 5 shows on the first row the number of useless candidates, that is the difference $cands - neighs$, selected on the three above mentioned data sets at the end of the candidate selection phase by the two methods, for various values of τ ($\tau \in [0.25, 0.75]$) and R (R in $\{250, 500, 750\}$). Number of pivots used by UP-index in this experiment is 5. From the figure it is clear that the UP-index in more effective than the U-tree, since the number of useless candidates selected by UP-index is sensibly smaller.

In order to compare the behavior of the two methods

4. We employed the source code of U-tree available at `http://www.cse.cuhk.edu.hk/~taoyf/paper/tods07.html`.

5. See `http://www.census.gov/geo/www/tiger/` and `http://www.rtreeportal.org`.

6. The U-tree code available actually supports only this kind of distributions.

when the dimensionality of the space increases, we considered a family of synthetic data sets similar to described in Section 4.1, but including only Constrained-Gaussian uncertain objects. The number of data set objects has been held fixed to $10{,}000$ and their dimensionality m has been varied from 2 to 5. The radius R employed is so that the selectivity ρ is approximatively 1.5%, while τ has been set to 0.75. The number of pivots employed by UP-index has been set to $|\mathcal{P}| = 10 \cdot m$.

Figure 5 shows on the second row the result of the experiment. On the left it is reported the number of useless candidates. Results show that UP-index performs better also in this experiment. Moreover, the difference between the number of useless candidates of the two methods even increases with the dimensionality. The total execution time is reported on the center. For homogeneity, the same integration procedure has been employed by the two methods. As far as the execution time is concerned, UP-index performs better, due to the ability of selecting a smaller set of candidates, and the relative speed of U-tree worsens with dimensionality. Moreover, the figure on the right, reporting the execution time of the initialization and candidate selection phases of the two methods, shows that the UP-index is able to determine a smaller set of candidates with fewer time requirements with respect to the U-tree.

5 Pivot Selection Problem

It follows from the cost analysis of Section 3.4 that the advantages of answering a range query by means of the UP–index rely on the effectiveness of the discard condition, that is in the size *cands* of the set of candidates. Since the cardinality of this set depends on the pivots used, it makes sense to define a criterion to measure the quality of the pivots at hand. In this section we provide a definition suited for the uncertain data setting.

Let $\mathcal{P}$ be a set of pivots. It is know from Theorem 2.1 that

$$\exists p \in \mathcal{P}, Pr(D_p(x,y) \leq R) < \tau \Longrightarrow Pr(d(x,y) \leq R) < \tau.$$

However, if the left side of the implication does not hold, then we must explicitly compute the probability $Pr(d(x,y) \leq R)$. Thus, the best set $\mathcal{P}^*$ of pivots should satisfy the following desideratum:

$$\exists p \in \mathcal{P}^*, Pr(D_p(x,y) \leq R) < \tau \Longleftarrow Pr(d(x,y) \leq R) < \tau.$$

Hence, next we define a criterion to evaluate the quality of a set of pivots whose underlying rationale is to measure the error committed when $d(x,y)$ is replaced with $D_p(x,y)$.

Given a set $\mathcal{P}$ of certain objects, let $D_{\mathcal{P}}(x,y)$ denote the random variable defined as

$$D_{\mathcal{P}}(x,y) = \max_{p \in \mathcal{P}} |d(x,p) - d(p,y)|,$$

and let $\epsilon_{\mathcal{P}}(x,y)$, also called *error*, denote the random variable defined as

$$\epsilon_{\mathcal{P}}(x,y) = d(x,y) - D_{\mathcal{P}}(x,y) = \min_{p \in \mathcal{P}} d(x,y) - |d(x,p) - d(p,y)|.$$

Note that $\epsilon_{\mathcal{P}}(x,y)$ is always non negative. The expected value of $\epsilon_{\mathcal{P}}$ is

$$\mathbf{E}[\epsilon_{\mathcal{P}}(x,y)] = \int_0^\infty u \cdot Pr(\epsilon_{\mathcal{P}}(x,y) = u)\, du.$$

The problem of finding the best set of pivots, referred to as *pivot selection problem* in the following, is defined as follows. *Given $k \geq 1$, find the set $\mathcal{P}^*$ of certain objects such that*

$$\mathcal{P}^* = \operatorname*{arg\,min}_{\mathcal{P}:|\mathcal{P}|=k} Q\left(\{\mathbf{E}[\epsilon_{\mathcal{P}}(x,y)] \mid x,y \in DS\}\right), \qquad (8)$$

where $Q : \wp(\mathbb{R}_0^+) \mapsto \mathbb{R}_0^+$ is a *penalty function* suitable to measure the error distribution, with $\wp(S)$ denoting the power set of the set S.

In order to estimate the error distribution we should know in advance the query objects to be employed in the computation of the error. However, since this information cannot be available, we make the assumption that the queries follow the distribution of the data set objects (as already done in other contexts [24]).

Moreover, we point out that in the literature the set of pivots is singled out among the data set objects, and in the following we consider this setting.

However, since pivots are certain objects, for the purpose of singling out the optimal set of pivots, we need to replace the uncertain objects in DS with a set of representative certain objects, say this set $cert(DS)$. We assume that each object x in DS provides one representative $cert(x)$ to the set $cert(DS)$, that is $cert(DS) = \{cert(x) \mid x \in DS\}$. An example of set $cert(DS)$ is that composed of the means $\overline{x}$ of the objects x in DS.

Moreover, we assume that the objects in the set $cert(DS)$ are already explicitly encoded in the description of the pdfs associated with uncertain objects (in their parameters, the region of definition, some histograms, or other).

5.1 Complexity Analysis

In this section, we investigate the computational complexity of the pivot selection problem.

With this aim, we introduce the decision version of the pivot selection problem, also called *pivot decision problem*, defined as follows: *given a set DS of uncertain objects, a penalty function Q, a positive integer $k \geq 1$ and a non negative real number t, decide whether there exists a set of certain objects $\mathcal{P} \subseteq \mathrm{cert}(DS)$ such that $|\mathcal{P}| = k$ and*

$$Q\left(\{\mathbf{E}[\epsilon_{\mathcal{P}}(x,y)] \mid x,y \in DS\}\right) \leq t. \qquad (9)$$

We denote an instance of this problem by $\langle DS, Q, k, t\rangle$.

The following theorem provides a lower bound to the computational complexity of the pivot decision problem.

Theorem 5.1: The pivot decision problem $\langle DS, Q, k, t\rangle$ is NP-hard.

The contributions of Theorem 5.1 can be summarized as follows:

- First of all, it states that the pivot selection is a *source of complexity* on its own;

- second, it implies that the pivot decision problem is NP-complete whenever the *integration is tractable* [32] or in any *fixed dimensionality*;
- last, but not the least, it captures the *complexity of the certain case*.

Next, we provide the formal proof of the theorem.

Proof of Theorem 5.1: The proof is by reduction from the Vertex Cover problem [33]: given an undirected graph $G = (V, E)$ and a positive integer $k \leq |V|$, is there a vertex cover of size k or less for G, i.e., a subset $V' \subseteq V$ with $|V'| \leq k$ such that for each edge $\{x, y\} \in E$ at least one of x and y belongs to V'?

Let $G = (V, E)$ be an undirected graph. Without loss of generality assume that the graph G is connected.

Let a_V denote an attribute whose domain $\mathbb{D}(A_V)$ is V, and let A_V denote the (singleton) set of attributes $\{a_V\}$.

For each node $x \in V$, let o_x denote the certain object on A_V having value x. For each $W \subseteq V$, let O_W denote the set of certain objects $\{o_x \mid x \in W\}$. Moreover, for each $x \in V$, let u_x denote the uncertain object having associated the pdf f_x defined on $\mathbb{D}(A_V)$ and such that $f_x(y)$ is $\delta(0)$ if $y = x$ and 0 otherwise, where $\delta(t)$ denotes the Dirac delta function. Let U_V denote the set of uncertain objects $\{u_x \mid x \in V\}$.

Let a and b be two positive real numbers such that $b = 2a$, and let d_G denote the distance metric on the objects in O_V defined as follows: $d_G(o_x, o_y) = b$ if $(x, y) \in E$, and $d_V(o_x, o_y) = a$ if $(x, y) \notin E$.

Let DS_G denote the uncertain data set consisting of the set of objects U_V and such that the distance metric associated with the certain objects in the domain $\mathbb{D}(A_V)$ is d_G.

Given a graph $G = (V, E)$, let $G^* = (V^*, E^*)$ be the graph obtained as follows. For each node $x \in V$, the set V^* contains the node x and two new nodes, named x' and x''. Let $V' = \{x' \mid x \in V\}$ and $V'' = \{x'' \mid x \in V\}$. Then, $V^* = V \cup V' \cup V''$. The set of edges E^* is $E \cup E'$, where E' is the following set of new edges: $\{\{x, x'\}, \{x', x''\} \mid x \in V\}$.

Figure 6b shows an example of graph G^* which is associated with the input graph reported in Figure 6a.

Let Q_m be a penalty function such that, for each set S of non negative real numbers it holds that

$$Q_m(S) > 0 \text{ if and only if } \max(S) > 0. \tag{10}$$

Given a graph $G = (V, E)$ and a positive integer number k, next we prove that G has a vertex cover of size equal to k if and only if

$$\langle DS_{G^*}, Q_m, n + k, 0 \rangle$$

is a "yes" instance, where $n = |V|$.

We assume that for each uncertain object u_x of DS_{G^*}, it holds that $cert(u_x) = o_x$. Hence, the set $cert(U_{V^*})$ is O_{V^*}.

We note that, for each pair u_x, u_y of uncertain objects of DS_{G^*}, it holds that

$$\mathbf{E}[\epsilon_{\mathcal{P}}(u_x, u_y)] = d_{G^*}(o_x, o_y) - \max_{p \in \mathcal{P}} |d_{G^*}(o_x, p) - d_{G^*}(p, o_y)|.$$

Also, we note that, for each $\mathcal{P} \subseteq O_{V^*}$ and for each pair u_x and u_y of uncertain objects such that either $o_x \in \mathcal{P}$ or $o_y \in \mathcal{P}$, it holds that $\mathbf{E}[\epsilon_{\mathcal{P}}(u_x, u_y)] = 0$. In fact, in this case the error $\epsilon_{\mathcal{P}}(u_x, u_y)$ is minimized for $p = o_y$ (or, equivalently, $p = o_x$) and it evaluates to $d_{G^*}(o_x, o_y) - d_{G^*}(o_x, o_y) + d_{G^*}(o_y, o_y) = 0$.

In order to complete the proof, we need first to prove the following two claims.

Claim 5.2: For each $\mathcal{P} \subseteq O_{V^*}$ and for each pair u_x and u_y of uncertain objects such that $\{x, y\} \in E^*$, it holds that $\mathbf{E}[\epsilon_{\mathcal{P}}(u_x, u_y)] = 0$, if either $o_x \in \mathcal{P}$ or $o_y \in \mathcal{P}$. Otherwise $\mathbf{E}[\epsilon_{\mathcal{P}}(u_x, u_y)] \geq a$.

Proof: Since $\{x, y\} \in E^*$, the distance $d_{G^*}(o_x, o_y)$ is b by definition. Assume that neither $o_x \in \mathcal{P}$ nor $o_y \in \mathcal{P}$. Then, for each $p = o_z \in \mathcal{P}$, we have three cases:

- both $\{x, z\} \in E^*$ and $\{z, y\} \in E^*$. Then, $d_{G^*}(o_x, o_y) - |d_{G^*}(o_x, o_z) - d_{G^*}(o_z, o_y)| = b - |b - b| = b$;
- both $\{x, z\} \notin E^*$ and $\{z, y\} \notin E^*$. Then, $d_{G^*}(o_x, o_y) - |d_{G^*}(o_x, o_z) - d_{G^*}(o_z, o_y)| = b - |a - a| = b$;
- $\{x, z\} \in E^*$ and $\{z, y\} \notin E^*$ (or, vice versa). Then, $d_{G^*}(o_x, o_y) - |d_{G^*}(o_x, o_z) - d_{G^*}(o_z, o_y)| = b - |b - a| = a$.

Thus, $\mathbf{E}[\epsilon_{\mathcal{P}}(u_x, u_y)] \geq \min\{a, b\} = a$. □

Claim 5.3: For each $\mathcal{P} \subseteq O_{V^*}$ and for each pair u_x and u_y of uncertain objects such that $\{x, y\} \notin E^*$, it holds that $\mathbf{E}[\epsilon_{\mathcal{P}}(u_x, u_y)] \leq 2a - b$ if and only if (1) either $o_x \in \mathcal{P}$ or $o_y \in \mathcal{P}$, or (2) there exists $o_z \in \mathcal{P}$ such that $\{x, z\} \in E^*$ and $\{z, y\} \notin E^*$. Otherwise $\mathbf{E}[\epsilon_{\mathcal{P}}(u_x, u_y)] = a > 2a - b$.

Proof: Since $\{x, y\} \in E^*$, the distance $d_{G^*}(o_x, o_y)$ is a by definition. Assume that neither $o_x \in \mathcal{P}$ nor $o_y \in \mathcal{P}$. Then, for each $p = o_z \in \mathcal{P}$, we have three cases:

- both $\{x, z\} \in E^*$ and $\{z, y\} \in E^*$. Then, $d_{G^*}(o_x, o_y) - |d_{G^*}(o_x, o_z) - d_{G^*}(o_z, o_y)| = a - |b - b| = a$;
- both $\{x, z\} \notin E^*$ and $\{z, y\} \notin E^*$. Then, $d_{G^*}(o_x, o_y) - |d_{G^*}(o_x, o_z) - d_{G^*}(o_z, o_y)| = a - |a - a| = a$;
- $\{x, z\} \in E^*$ and $\{z, y\} \notin E^*$ (or, vice versa). Then, $d_{G^*}(o_x, o_y) - |d_{G^*}(o_x, o_z) - d_{G^*}(o_z, o_y)| = a - |b - a| = 2a - b$.

□

Now, we can resume the main proof.

($\Rightarrow$) Assume that G has a vertex cover C of size k. Let C be $\{x_1, \ldots, x_k\}$. Now we show that $\mathcal{P}_C = \{o_{x_1}, \ldots, o_{x_k}\} \cup \{o_{x'} \mid x \in V\}$ is a set of pivots of size $n + k$ such that

$$Q_m(\{\mathbf{E}[\epsilon_{\mathcal{P}_C}(u_x, u_y)] \mid u_x, u_y \in DS_{G^*}\}) = 0.$$

With this aim, for each $u_x, u_y \in DS_{G^*}$ it must be verified that $\mathbf{E}[\epsilon_{\mathcal{P}_C}(u_x, u_y)] = 0$. Two cases are to be considered, that are $\{x, y\} \in E^*$ and $\{x, y\} \notin E^*$.

Assume that $\{x, y\} \in E^*$. Since C is a vertex cover, then it is the case that either $x \in C$, hence $o_x \in \mathcal{P}_C$, or $y \in C$, hence $o_y \in \mathcal{P}_C$. Thus, $\mathbf{E}[\epsilon_{\mathcal{P}_C}(u_x, u_y)] = 0$.

Conversely, assume that $\{x, y\} \notin E^*$. If either $o_x \in \mathcal{P}_C$ or $o_y \in \mathcal{P}_C$, then $\mathbf{E}[\epsilon_{\mathcal{P}_C}(u_x, u_y)] = 0$. Vice versa, if $o_x \notin \mathcal{P}_C$ and $o_y \notin \mathcal{P}_C$:

- Assume that both x and y belong to V. Recall that for each x' in V' the object $o_{x'}$ is in $\mathcal{P}_C$. Thus, by definition of E^*, $\{x, x'\} \in E^*$ and $\{y, x'\} \notin E^*$, and, by Claim 5.3, $\mathbf{E}[\epsilon_{\mathcal{P}_C}(u_x, u_y)] \leq 2a - b = 0$;
- If both x and y belong to V'', then $\{x, x'\} \in E^*$ and $\{y, x'\} \notin E^*$, and $\mathbf{E}[\epsilon_{\mathcal{P}_C}(u_x, u_y)] = 0$;
- If x belongs to V, y belongs to V'', and $y \neq x''$, then $\{x, x'\} \in E^*$ and $\{y, x'\} \notin E^*$, and $\mathbf{E}[\epsilon_{\mathcal{P}_C}(u_x, u_y)] = 0$;
- Consider, finally, the case $y = x''$. Since $o_x \notin \mathcal{P}_C$ (i.e. $x \notin C$), the graph G is connected, and C is a vertex cover for G^*, it is the case that there exists $z \in V$ such that $\{x, z\} \in E$ and $z \in C$, and, thus, $o_z \in \mathcal{P}_C$. Moreover, $\{x, z\} \in E^*$ and $\{z, x''\} \notin E^*$, and $\mathbf{E}[\epsilon_{\mathcal{P}_C}(u_x, u_y)] = 0$.

($\Leftarrow$) Assume that there exists a subset $\mathcal{P}$ of $cert(DS_{G^*})$ having size $n + k$ such that

$$Q_m(\{\mathbf{E}[\epsilon_{\mathcal{P}}(u_x, u_y)] \mid u_x, u_y \in DS_{G^*}\}) = 0.$$

Then, by Claim 5.2, it is the case that for each $u_x, u_y \in DS_{G^*}$ such that $\{x, y\} \in E^*$, it holds that either $o_x \in \mathcal{P}$ or $o_y \in \mathcal{P}$. This means that the set $C_{\mathcal{P}} = \{x \mid o_x \in \mathcal{P}\}$ is a vertex cover for the graph G^*.

Consider the set of edges $E'' = \{\{x', x''\} \mid x \in V\}$ of E^*. For each edge $\{x', x''\} \in E''$, either x' or x'' have to be in $C_{\mathcal{P}}$. Thus, $C_{\mathcal{P}} \setminus (V' \cup V'')$ is a vertex cover of G whose cardinality is not greater than k. □

It can be easily verified that the *arithmetic mean* and the *maximum* are penalty functions complying with the property of the function Q_m employed in the reduction of Theorem 5.1 (see Equation (10)). Hence, the following result follows from Theorem 5.1.

Theorem 5.4: Let $Q \in \{\max, \text{mean}\}$. Then the problem $\langle DS, Q, k, t\rangle$ is NP-hard.

In particular, as far as the maximum function is concerned, the following property can be checked.

Remark 5.5: If Q_m is set to $\max$ in the reduction of Theorem 5.1, then the reduction still holds for any value $a \in [\frac{b}{2}, b)$, provided that the threshold t is set to a.

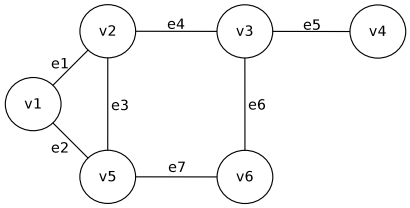
Based on this property, the complexity result stated in Theorem 5.1 can be extended to the Euclidean space, as accounted for in the following Theorem.

Theorem 5.6: Let DS be an uncertain data set on the domain $\mathbb{R}^m$ equipped with the Euclidean distance. Then the $\langle DS, Q, k, t\rangle$ problem is NP-hard.

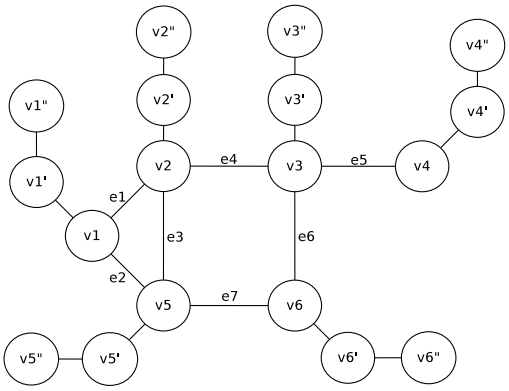
Proof: Let $G = (V, E)$ be an indirect graph. Assume that an arbitrary order has been established on the nodes in V, so that each edge $e \in E$ of the graph can be denoted as a pair $e = (u, v)$, with u preceding v according to the order.

Next, we report a procedure to map each node v of G to a point p_v of $\mathbb{R}^m$, in a way so that $d(p_u, p_v) = b$ iff $(u, v) \in E$ and $d(u, v) = a$ iff $(u, v) \notin E$, where d denotes the Euclidean distance and a and b are positive real numbers with $a \in [\frac{b}{2}, b)$.

Consider the domain $\mathbb{R}^m$ with $m = |V| + |E|$. In particular, $|E|$ dimensions, denoted as $e_1, \ldots, e_{|E|}$, are



(a) An example of input graph G.



(b) The graph G^* described in Theorem 5.1.

	e1	e2	e3	e4	e5	e6	e7	$\mathbf{a_{v1}}$	$\mathbf{a_{v2}}$	$\mathbf{a_{v3}}$	$\mathbf{a_{v4}}$	$\mathbf{a_{v5}}$	$\mathbf{a_{v6}}$
v1	0	0	1	1	1	1	1	$\sqrt{4}$	0	0	0	0	0
v2	2	1	0	0	1	1	1	0	$\sqrt{3}$	0	0	0	0
v3	1	1	1	2	0	0	1	0	0	$\sqrt{3}$	0	0	0
v4	1	1	1	1	2	1	1	0	0	0	$\sqrt{5}$	0	0
v5	1	2	2	1	1	1	0	0	0	0	0	$\sqrt{3}$	0
v6	1	1	1	1	1	2	2	0	0	0	0	0	$\sqrt{4}$

(c) The points p_v associated with the graph G (see Theorem 5.6).

Fig. 6: Examples concerning reductions described in Theorems 5.1 and 5.6.

associated with the edges in E. Moreover, for each node $v \in V$, there is one dimension a_v, associated with v.

Let m_v be the difference between $|V|$ and the number $\deg(v)$ of edges incident the node v. Each node $v \in V$ is mapped to a point p_v of $\mathbb{R}^m$ as follows:

- for each e_i ($1 \leq i \leq |E|$):
  - $p_v[e_i] = 0$ if $e_i = (v, w)$;
  - $p_v[e_i] = 2$ if $e_i = (w, v)$;
  - $p_v[e_i] = 1$ if e_i does not join v;
- $p_v[a_v] = \sqrt{m_v}$;
- $p_v[a] = 0$, for each other dimension a.

Figure 6c reports an example of this mapping on the nodes of the graph G in Figure 6a.

Assume that v and w are not joined by edges. In this case: (i) on $\deg(v)$ dimensions in the set $\{e_1, \ldots, e_{|E|}\}$, p_v assumes value 0 or 2, while p_w assumes value 1; (ii) on $\deg(w)$ dimensions in the set $\{e_1, \ldots, e_{|E|}\}$, p_w assumes value 0 or 2, while p_v assumes value 1; (iii) on all the other dimensions in the set $\{e_1, \ldots, e_{|E|}\}$, both p_v and p_w assume value 1; (iv) on the a_v dimension, p_v assumes value $\sqrt{m_v}$, while p_w assumes value 0; (v) on the a_w dimension, p_w assumes value $\sqrt{m_w}$, while p_v assumes value 0; (vi) on all the other dimensions a_u, with $u \neq v$

and $u \neq w$, both p_v and p_w assume value 0. Then, the distance between p_v and p_w is

$$a = \sqrt{\deg(v) + \deg(w) + \sqrt{m_v}^2 + \sqrt{m_w}^2} = \sqrt{2|V|}.$$

Assume now that v and w are joined by the edge e_i. In this case, on the dimension e_i, either $p_v[e_i] = 0$ and $p_w[e_i] = 2$ or vice versa. As for the other dimensions: (i) on $\deg(v) - 1$ dimensions in the set $\{e_1, \ldots, e_{|E|}\}$, p_v assumes value 0 or 2, while p_w assumes value 1; (ii) on $\deg(w) - 1$ dimensions in the set $\{e_1, \ldots, e_{|E|}\}$, p_w assumes value 0 or 2, while p_v assumes value 1; (iii) on all the other dimensions in the set $\{e_1, \ldots, e_{|E|}\}$, both p_v and p_w assume value 1; (iv) on the a_v dimension, p_v assumes value $\sqrt{m_v}$, while p_w assumes value 0; (v) on the a_w dimension, p_w assumes value $\sqrt{m_w}$, while p_v assumes value 0; (vi) on all the other dimensions a_u, with $u \neq v$ and $u \neq w$,both p_v and p_w assume value 0. Then, the distance between v and w is

$$b = \sqrt{2^2 + \deg(v) - 1 + \deg(w) - 1 + \sqrt{m_v}^2 + \sqrt{m_w}^2} = \\ = \sqrt{2 + 2|V|}.$$

It can be concluded that the distance b between pair of points associated with nodes joined by an edge is strictly larger than the distance a between pair of points associated with nodes not joined by any edge. In particular, it holds that the ratio between a and b is:

$$\frac{a}{b} = \frac{\sqrt{2|V|}}{\sqrt{2|V| + 2}} = \sqrt{\frac{|V|}{|V| + 1}} \in [0.5, 1).$$

Thus, letting Q be $\max$, by Theorem 5.1 and Remark 5.5 the result follows. □

5.2 Pivot-Selection Algorithm

Due to Theorem 5.1, no polynomial time algorithm is conjectured to be able to determine an optimal set of pivots. In this section we describe an heuristic algorithm with limited time bounds for selecting a good quality set of pivots when uncertain objects are taken into account.

The algorithm, reported in Figure 7, is based on a greedy selection of pivots guided by the optimality criterion introduced in Section 5. In the following, we consider as penalty function Q the mean. In particular, the algorithm performs k iterations. It starts with an empty set of pivots $\mathcal{P}^*$ and, then, at each iteration, augments $\mathcal{P}^*$ with the object p^* from $cert(DS)$ that minimizes the penalty function when $\mathcal{P} = \mathcal{P}^* \cup \{p^*\}$ is considered as set of pivots.

Random sampling is exploited in order to estimate the value of the penalty function $Q\left(\{\mathbf{E}[\epsilon_{\mathcal{P}_i}(x, y)] \mid x, y \in DS\}\right)$. With this aim, the algorithm randomly selects T pairs (x_j, y_j) of uncertain objects from DS and, for each of them, a pair (v_j^x, v_j^y) of values distributed according to f^{x_j} and f^{y_j}.

For each set of pivots, the value of the penalty function is estimated by taking into account the T randomly

```
   Input: the uncertain data set DS = {x_1, ..., x_n}
          the number k of pivots
  Output: the set P* of pivots
  Method:
1.  P* = ∅
2.  Q* = ∞
3.  for l = 1 to k do
4.    for i = 1 to n do
5.      P = P* ∪ {cert(x_i)}
6.      S = 0
7.      for j = 1 to T do
8.        randomly pick an uncertain object x_j from DS
9.        randomly pick an uncertain object y_j from DS
10.       randomly pick a value v_j^x distributed according to f^{x_j}
11.       randomly pick a value v_j^y distributed according to f^{y_j}
12.       S = S + min_{p∈P}{d(v_j^x, v_j^y) − |d(v_j^x, p) − d(v_j^y, p)|}
13.     endfor
14.     Q = S/T
15.     if Q < Q* then
16.       Q* = Q
17.       p* = cert(x_i)
18.     endif
19.   endfor
20.   P* = P* ∪ {p*}
21. endfor
22. return P*
```

Fig. 7: Pivot selection algorithm.

selected pairs (v_j^x, v_j^y). In particular, the variable S accumulates the values $\min_{p \in \mathcal{P}}\{d(v_j^x, v_j^y) - |d(v_j^x, p) - d(v_j^y, p)|\}$ associated with the pairs (v_j^x, v_j^y).

Estimating the penalty function value. Next we show how the value of the parameter T can be set to obtain a safe estimation of the penalty function value Q. Let $\epsilon_{\mathcal{P}}$ be the random variable associated with the value $\mathbf{E}[\epsilon_{\mathcal{P}}(x, y)]$, then the best set of pivots $\mathcal{P}^*$ is such that

$$\mathcal{P}^* = \arg\min_{\mathcal{P}:|\mathcal{P}|=k} \mathbf{E}[\epsilon_{\mathcal{P}}].$$

In order to estimate the value $\mathbf{E}[\epsilon_{\mathcal{P}}]$, we exploit random sampling and compute the penalty function value Q as the ratio $\frac{S}{T}$, as shown in Figure 7 (lines 6-14).

Let λ denote the maximum value associated with the error $\epsilon_{\mathcal{P}}(x, y)$, that is to say the maximum distance between two data set objects x and y, let $\xi \in [0, 1]$ be a relative error threshold, and let δ be a probability threshold. Our goal is that

$$Pr\left(\left|\frac{Q}{\lambda} - \frac{\mathbf{E}[\epsilon_{\mathcal{P}}]}{\lambda}\right| < \xi\right) > 1 - \delta, \tag{11}$$

namely, that the probability that the difference between the estimated and the true value of the penalty function is lower that ξ is greater than $1 - \delta$.

The above goal can be reached by properly setting the size T of the sample used to determine the value Q. In particular, an upper bound on the probability for the sum of random variables to deviate from its expected value can be obtained from the *Hoeffding's inequality* [34]:

$$Pr\left(|Q - \mathbf{E}[\epsilon_{\mathcal{P}}]| \geq t\right) \leq 2 \cdot \exp\left(-2T\left(\frac{t}{\lambda}\right)^2\right),$$

$\delta \setminus \xi$	0.1	0.05	0.01
0.1	150	600	14,979
0.05	185	738	18,445
0.01	265	1,060	26,492

TABLE 1: Size T of the sample needed for estimating the penalty function value.

or, equivalently:

$$Pr\left(|Q - \mathbf{E}[\epsilon_{\mathcal{P}}]| < t\right) > 1 - 2 \cdot \exp\left(-2T\left(\frac{t}{\lambda}\right)^2\right).$$

From the above equation, it follows that:

$$Pr\left(\left|\frac{Q}{\lambda} - \frac{\mathbf{E}[\epsilon_{\mathcal{P}}]}{\lambda}\right| < \frac{t}{\lambda}\right) > 1 - 2 \cdot \exp\left(-2T\left(\frac{t}{\lambda}\right)^2\right),$$

and, by setting ξ equal to $\frac{t}{\lambda}$, we finally obtain the following expression:

$$Pr\left(\left|\frac{Q}{\lambda} - \frac{\mathbf{E}[\epsilon_{\mathcal{P}}]}{\lambda}\right| < \xi\right) > 1 - 2 \cdot \exp\left(-2T\xi^2\right).$$

By comparing the last inequality with Equation (11), it follows that our goal is reached for

$$T \geq \frac{1}{2\xi^2} \log\left(\frac{2}{\delta}\right).$$

Table 1 reports the size T of the sample needed to correctly estimate the penalty function in correspondence of various values of ξ and δ.

Cost of the algorithm. As for the temporal cost of the algorithm, it corresponds to the cost of estimating $n \cdot k$ times the penalty function. The cost of computing the error on a single pair (v_i^x, v_i^y) of values, at the l-th iteration, is $O(l)$. Thus, the cost of estimating the penalty function value is $O(T \cdot l)$. However, since $l - 1$ pivots are held fixed during the estimation, the above cost can be reduced to $O(T)$ by storing the randomly selected pairs (v_j^x, v_j^y) are reusing them throughout the execution of the algorithm. Summarizing, the temporal cost of the algorithm is linear, that is $O(n \cdot k \cdot T)$.

We note that criteria different from that used could be adopted to select the sets of pivots $\mathcal{P}$ to be tested for optimality; among them all the strategies suitable to face intractable problems, such as heuristic search, local search, genetic algorithms, and so on. We point out that our goal here is not to explore the behavior of different pivot selection policies, but rather to show that (i) the algorithm together with the estimation technique is able to select a set of pivots scoring a value for the penalty function Q sensibly better than those associated with randomly selected set of pivots, and that (ii) the criterion here introduced is effective in enhancing the quality of the selected set of pivots, namely to improve the gain measure. These aspects will be investigated in the subsequent section devoted to experimental results.

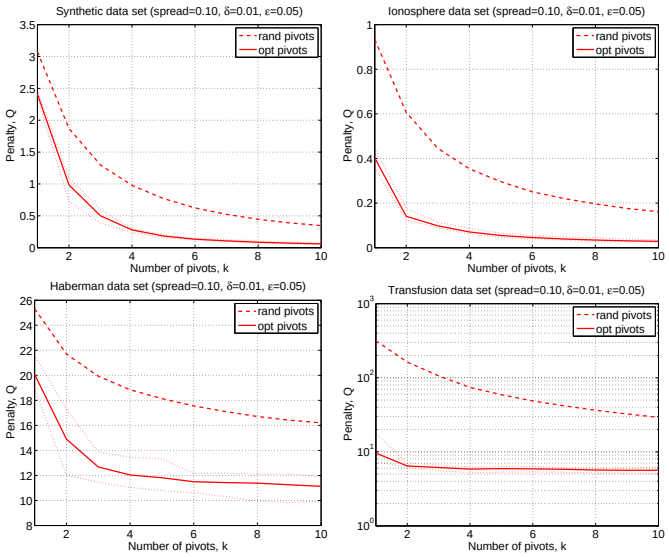


Fig. 8: Optimized vs random method: penalty function.

5.3 Experiments

In this section, we test the effectiveness of the quality criterion introduced at the beginning of Section 5 and of the algorithm introduced in Section 5.2.

In order to check the validity of the criterion for measuring the goodness of a set of pivots formalized in Equation (8), we compared the behavior of the UP–index using the pivots returned by the algorithm described in Section 5.2 (also referred to as *optimized* algorithm in the following) with the behavior of the UP–index using random pivots (also referred to as *random* algorithm in the following).

In particular, in order to quantify advantages of the optimized method over the random one, we measured the *improvement* and the *time improvement*. The *improvement* is defined as

$$Improvement = 1 - \frac{cands_{opt} - neighs}{cands - neighs},$$

where $cands_{opt}$ ($cands$, resp.) is the number of candidate objects selected by the optimized (random, resp.) method. Intuitively, the term $\frac{cands_{opt} - neighs}{cands - neighs}$ represents the relative (w.r.t. the number of candidates selected by the random method) number of probability distances that are required to be computed during the filtering phase of the optimized method out of those involving the true neighbors. The *time improvement* is defined as

$$Time\ improvement = 1 - \frac{t_{opt} - t_{neighs}}{t_{pivots} - t_{neighs}},$$

where t_{opt} (t_{pivots}, resp.) denotes the time employed by the optimized (random, resp.) algorithm to answer the range query, and t_{neighs} is as defined in Section 4.

Figure 8 compares the penalty value associated with pivots returned by the *optimized* algorithm with those associated with the *random* one. In particular, the algorithm of Figure 7 has been executed 100 times on the *Syntethic*

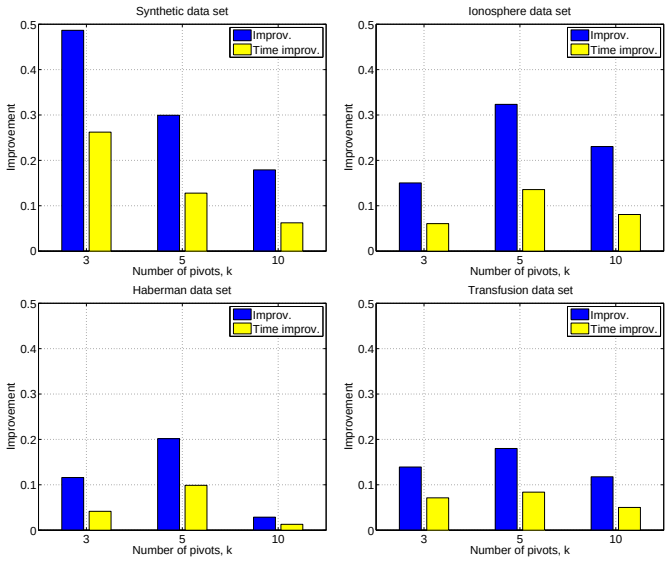


Fig. 9: Optimized vs random method: gain improvement.

(with $n = 10{,}000$ and $m = 2$), *Ionosphere*, *Haberman*, and *Transfusion* data sets (with spread 0.10), with parameters $\delta = 0.01$ and $\xi = 0.05$ (corresponding to $T = 1{,}060$), and k ranging from 1 to 10. The *random* algorithm has been executed $1{,}000$ times on the same data sets. We ran the optimized algorithm different times in order to take into account variability associated with the estimation technique.

The solid curve reports the average value of the penalty function associated with optimized pivots, while dotted curves show their minimum and maximum penalty value. The dashed curve reports the average penalty value associated with random pivots. The plot shows that the penalty associated with optimized pivots is always sensibly smaller than the average penalty associated with random ones, thus confirming that algorithm of Figure 8 is effective. Moreover, it appears that the gap between the random and optimized pivots tends to decrease with their number. This can be explained by noticing that the greater the size of the set of pivots $\mathcal{P}$, the greater the chance of reducing the error $\mathbf{E}[\epsilon_{\mathcal{P}}(x, y)]$ associated with a generic pair (x, y) of data set objects.

Moreover, to show how the penalty function Q varies with respect to the parameters δ and ξ, we next report the average value of Q on the *Synthetic* data set for $k = 3$ optimized pivots:

$\delta \setminus \xi$	0.1	0.05	0.01
0.1	0.6135	0.5589	0.4352
0.05	0.6134	0.5339	0.4179
0.01	0.6085	0.5190	0.4102

In order to compute improvement and time improvement, we performed one hundred uncertain queries, using $k = 3$, $k = 5$, and $k = 10$ pivots, with radius R corresponding to the maximum selectivity ρ considered in Section 4. Figure 9 reports the results of these experiments. As far as the *Synthetic* data set is concerned, the plot on the upper left corner of the figure shows that the improvement is monotonically decreasing with the number of pivots employed, though in any case noticeable, being always approximatively above the 20% and close to the 50% for three pivots. As for the other three data sets, the improvement first ameliorates and then worsens, thus reaching its maximum for $k = 5$, in correspondence of which the improvement is always approximatively above the 20%. The smaller improvement is scored by the *Transfusion* data set, which is also the data set that performs better when random pivots are employed.

All the plots show that the optimized pivots are effective in reducing both the number of useless candidates and the query execution time. Particularly, it follows from the discussion on the curves shown in Figure 8, that the improvement is expected to decrease with the number of pivots and, indeed, this kind of trend is confirmed by experimental results in Figure 9. As for the different behavior of the synthetic data set family with respect to the other data sets, we notice that objects in the first data set are randomly placed, while objects in the other data sets comply with real scattered data. Thus, for the latter data sets, the fact that the improvement presents a maximum for $k = 5$, reveals that there is a sort of trade off between placement of pivots $\mathcal{P}$ with respect to the overall data distribution and expected reduction of the error $\mathbf{E}[\epsilon_{\mathcal{P}}(x, y)]$ associated with a set $\mathcal{P}$ having size k.

6 Conclusions

In this work we dealt with the problem of quickly answering range queries over uncertain objects in a general metric space, and introduced a novel indexing technique, called UP–index. To the best of our knowledge, this is the first work providing an effective technique for indexing uncertain objects coming from general metric spaces.

We generalized the reverse triangular inequality to the probabilistic setting in order to exploit it to recognize non-neighbor objects without performing the heavy operation of computing the distance probability. Then, we introduced the UP–index and showed how to employ it in order to speed up range query computation. Importantly, the cost of the introduced discard condition is independent of the data dimensionality, while deciding whether an object belongs or not to the answer of the query amounts to evaluate a multi-dimensional integral. As a results, our technique permits to save a vast amount of time.

The experimental campaign validated the effectiveness of the proposed approach, in that it pointed out that our method is able to greatly reduce the number of candidate objects and to significantly improve time performances, and revealed that the introduced technique is even preferable to indexing techniques specifically designed for the Euclidean space.

We provided a criterion to measure the quality of a set of pivots and studied the problem of selecting a good

set of pivots. Moreover, we proved that selecting a set of pivots that minimizes the expected error in a general metric space is NP-hard, and introduced an estimation algorithm with statistical guarantees for selecting a good quality set of pivots.

Summarizing, this work establishes the foundations for indexing general metric spaces in the uncertain scenario and, as such, we believe it opens other interesting research directions.

REFERENCES

[1] R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J. S. Vitter, "Efficient indexing methods for probabilistic threshold queries over uncertain data," in *Proceedings of the Thirtieth international conference on Very large data bases (VLDB)*, 2004, pp. 876–887.

[2] Y. Tao, R. Cheng, X. Xiao, W. K. Ngai, B. Kao, and S. Prabhakar, "Indexing multi-dimensional uncertain data with arbitrary probability density functions," in *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, 2005, pp. 922–933.

[3] T. Green and V. Tannen, "Models for incomplete and probabilistic information," *IEEE Data Eng. Bull.*, vol. 29, no. 1, pp. 17–24, 2006.

[4] Y. Tao, X. Xiao, and R. Cheng, "Range search on multidimensional uncertain data," *ACM Transactions on Database Systems*, vol. 32, no. 3, p. 15, 2007.

[5] C. Aggarwal and P. Yu, "A survey of uncertain data algorithms and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 5, pp. 609–623, 2009.

[6] P. K. Agarwal, S.-W. Cheng, Y. Tao, and K. Yi, "Indexing uncertain data," in *PODS '09: Proceedings of the twenty-eighth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2009, pp. 137–146.

[7] C. C. Aggarwal, *Managing and Mining Uncertain Data*, ser. Advances in Database Systems. Springer, 2009, vol. 35.

[8] E. Chávez, G. Navarro, R. Baeza-Yates, and J. Marroquín, "Searching in metric spaces." *ACM Computing Surveys*, vol. 33, no. 3, pp. 273–321, 2001.

[9] H. Samet, *Foundations of Multidimensional and Metric Data Structures (The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling)*. Morgan Kaufmann Publishers Inc., 2005.

[10] P. Zezula, G. Amato, V. Dohnal, and M. Batko, *Similarity Search: The Metric Space Approach*, ser. Advances in Database Systems. Springer, 2006, vol. 32.

[11] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, pp. 707–710, 1966.

[12] H.-P. Kriegel, P. Kunath, M. Pfeifle, and M. Renz, "Probabilistic similarity join on uncertain data," in *International Conference on Database Systems for Advanced Applications*, 2006, pp. 295–309.

[13] M. Chau, R. Cheng, B. Kao, and J. Ng, "Uncertain data mining: An example in clustering location data," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2006, pp. 199–204.

[14] S. Łukaszyk, "A new concept of probability metric and its applications in approximation of scattered data sets," *Computational Mechanics*, vol. 33, no. 4, pp. 299–304, 2004.

[15] W. Ngai, B. Kao, C. Chui, R. Cheng, M. Chau, and K. Yip, "Efficient clustering of uncertain data," in *International Conference on Data Mining*, 2006, pp. 436–445.

[16] S. Singh, C. Mayfield, S. Prabhakar, R. Shah, and S. Hambrusch, "Indexing uncertain categorical data," in *Proceedings of the IEEE 23rd International Conference on Data Engineering (ICDE)*, April 2007, pp. 616–625.

[17] Y. Zhang, X. Lin, W. Zhang, J. Wang, and Q. Lin, "Effectively indexing the uncertain space," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 9, pp. 1247–1261, 2010.

[18] J. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, pp. 509–517, 1975.

[19] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The r*-tree: An efficient and robust access method for points and rectangles," in *Proc. of the SIGMOD Conference*, 1990, pp. 322–331.

[20] S. Berchtold, D. Keim, and H.-P. Kriegel, "The x-tree: An index structure for high-dimensional data," in *Proc. of the Conf. on VLDB*, 1996, pp. 28–39.

[21] P. N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces," in *ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 1993, pp. 311–321.

[22] L. Micó, J. Oncina, and E. Vidal, "A new version of the nearest-neighbour approximating and eliminating search algorithm (aesa) with linear preprocessing time and memory requirements," *Pattern Recognition Letters*, vol. 15, no. 1, pp. 9–17, 1994.

[23] E. Chávez, J. L. Marroquín, and R. A. Baeza-Yates, "Spaghettis: An array based algorithm for similarity queries in metric spaces," in *Symp. on String Processing and Information Retrieval (SPIRE)*, 1999, pp. 38–46.

[24] B. Bustos, G. Navarro, and E. Chávez, "Pivot selection techniques for proximity searching in metric spaces," *Pattern Recognition Letters*, vol. 24, no. 14, pp. 2357–2366, 2003.

[25] B. Bustos, O. Pedreira, and N. R. Brisaboa, "A dynamic pivot selection technique for similarity search," in *ICDE Workshops*, 2008, pp. 394–401.

[26] L. Ares, N. Brisaboa, M. Esteller, O. Pedreira, and A. Places, "Optimal pivots to minimize the index size for metric access methods," in *International Workshop on Similarity Search and Applications (SISAP)*, 2009, pp. 74–80.

[27] J. F. Traub, G. W. Wasilkowski, and H. Woźniakowski, *Information-based complexity*. San Diego, CA, USA: Academic Press Professional, Inc., 1988.

[28] N. Metropolis and S. Ulam, "The monte carlo method," *Journal of the American Statistical Association*, vol. 44, no. 247, pp. 335–341, 1949.

[29] P. Davis and P. Rabinowitz, *Methods of Numerical Integration*. Dover, 1984.

[30] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[31] R. Finn, J. Mistry, J. Tate, P. Coggill, A. Heger, J. Pollington, O. Gavin, P. Gunesekaran, G. Ceric, K. Forslund, L. Holm, E. Sonnhammer, S. Eddy, and A. Bateman, "The pfam protein families database," *Nucleic Acids Research*, vol. 38, no. Database Issue, pp. D211–D222, 2010.

[32] F. J. Hickernell and H. Wozniakowski, "Integration and approximation in arbitrary dimensions," *Advances in Computational Mathematics*, vol. 12, no. 1, pp. 25–58, 2000.

[33] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[34] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.