M.L. Lee, K.L. Tan, and V. Wuwongse (Eds.): In Proc. 11th International Conference on Database Systems for Advanced Applications (DASFAA 2006), LNCS 3882, pp. 295–309, 2006.
DASFAA 2006 BEST PAPER

Probabilistic Similarity Join on Uncertain Data

Hans-Peter Kriegel, Peter Kunath, Martin Pfeifle, Matthias Renz
University of Munich, Germany
{kriegel, kunath, pfeifle, renz}@dbs.ifi.lmu.de

Abstract. *An important database primitive for commonly used feature databases is the similarity join. It combines two datasets based on some similarity predicate into one set such that the new set contains pairs of objects of the two original sets. In many different application areas, e.g. sensor databases, location based services or face recognition systems, distances between objects have to be computed based on vague and uncertain data. In this paper, we propose to express the similarity between two uncertain objects by probability density functions which assign a probability value to each possible distance value. By integrating these probabilistic distance functions directly into the join algorithms the full information provided by these functions is exploited. The resulting probabilistic similarity join assigns to each object pair a probability value indicating the likelihood that the object pair belongs to the result set. As the computation of these probability values is very expensive, we introduce an efficient join processing strategy exemplarily for the distance-range join. In a detailed experimental evaluation, we demonstrate the benefits of our probabilistic similarity join. The experiments show that we can achieve high quality join results with rather low computational cost.*

1 Introduction

In many modern application ranges, e.g. spatio-temporal query processing of moving objects [9], sensor databases [8] or personal identification systems [28], usually only uncertain data is available. For instance, in the area of mobile services, the objects continuously change their positions so that exact positional information is almost impossible to obtain. In the area of multimedia databases, e.g. image or music databases, or in the area of personal identification systems based on face recognition and fingerprint analysis, there often exists the problem that a feature value cannot exactly be determined. This uncertain data can be handled by assigning confidence intervals to the feature values, or by specifying probability density functions indicating the likelihoods of certain feature values. In other application areas such as the clustering of distributed feature vectors [13], only approximated (uncertain) information is transmitted to a central server site due to security aspects or limited bandwidth. Let us note that the distance-range join can act as a preprocessing step to speed up clustering.

In order to join these uncertain object representations by traditional join methods, the similarity between the objects has to be measured by one numerical value, i.e. the complete probabilistic distance information is aggregated by only one distance value. Obviously, aggregation goes hand in hand with information loss. For instance, we have no information about the degree of uncertainty of such a single distance value. Even if we

had one, it would be of no use because traditional join algorithms cannot handle this additional information.

In this paper, we propose to use probabilistic distance functions to measure the similarity between uncertain objects. Contrary to traditional approaches, we do not extract aggregated values from the probabilistic distance functions but enhance the join algorithms so that they can exploit the full information provided by these functions. The resulting probabilistic similarity join assigns a probability value to each object pair indicating the likelihood that the pair belongs to the result set, i.e. these probably values reflect the trustability of the result. In applications where wrong results have fatal consequences, e.g. medical treatment, users might only look at very certain results, whereas in commercial advertising, for instance, all results might be interesting. In this paper, we propose a solution for a probabilistic similarity join which is practically very important, the problematic distance-range join.

Probabilistic distance-range joins can be used in the area of location based services but also in many different other areas. For instance, like their non-probabilistic counterparts, they can serve as basic operations for data mining algorithms. Based on the result set of the probabilistic distance-range join, we can efficiently generate a density based clustering of uncertain data. We could group those objects together into one cluster which have a probability value higher than 0.5 that their distance is lower than a certain threshold value.

In this paper, we first present the theoretical foundations of probabilistic similarity joins, and then show how to compute them based on the generally applicable concept of monte-carlo sampling. Thereby, each uncertain object is described by a set of sample points. In order to guarantee efficient join processing, we group the sample points of one uncertain object into k clusters. Minimal bounding boxes of these clusters are then used to identify and skip unnecessary distance computations in a filter step. For the distance-range join, the filter step has an additional advantage. Often an incremental processing of the join query is desired which returns the results in descending order of their probabilities, i.e. the most promising results are returned first. Our approach allows us to determine an upper-bound probability value for each object pair in the filter step which can then be used to return the first results very early.

The remainder of this paper is organized as follows: In Section 2, we present the related work in the area of similarity join processing and query processing of uncertain data. In Section 3, we show how we can carry out a non-probabilistic similarity join on uncertain data. In Section 4, we propose our probabilistic similarity join, which is evaluated in detail in Section 5. We conclude this paper in Section 6 with a short summary and a few remarks on future work.

2 Related Work

In the past decade, a lot of work has been done in the field of similarity join processing. Recently some researchers have focused on the area of query processing of uncertain data. However, to the best of our knowledge no work has been done in the area of join processing of uncertain data. In the following, we present related work on both topics, similarity join processing and query processing of uncertain data.

2.1 Similarity Join

A join groups tuples of two relations R and S into pairs if a *join predicate* is fulfilled. In a *similarity join*, the join predicate is based on the similarity between the objects stored in the relations. This similarity is measured by a distance function d: $O \times O \rightarrow IR_0^+$, e.g. the Euclidean distance between two feature vectors. The most popular similarity join operation is the distance range join. The distance range join $R \underset{\varepsilon}{\bowtie} S$ of two multidimensional or metric sets R and S is the set of pairs where the distance of the objects does not exceed a given parameter ε:

Definition 1 distance-range join (ε-join)
The distance range join $R \underset{\varepsilon}{\bowtie} S$ of two finite sets R and S is the set $R \underset{\varepsilon}{\bowtie} S := \{(r, s) \in R \times S: d(r, s) \leq \varepsilon\}$.

The distance range join can be applied in density-based clustering algorithms which often define the local data density as the number of objects in the ε-neighborhood of some data object. These clustering algorithms can beneficially be expressed by a self-join using the distance-range paradigm [3].

Most related work on efficient join processing is related to the spatial intersection join. These algorithms which are often based on multidimensional index structures can easily be adapted to distance based predicates for multidimensional point databases instead of the intersection of polygons. The most common technique is the *R-tree Spatial Join* (*RSJ*) [4] which processes R-tree like index structures built on both relations R and S. The RSJ algorithm traverses the indexes of R and S synchronously. When a pair of directory pages (P_R, P_S) is under consideration, the algorithm forms all pairs of the child pages of P_R and P_S having distances of at most ε. For these pairs of child pages, the algorithm is called recursively, i.e. the corresponding indexes are traversed in a depth-first order. Various optimizations of RSJ have been proposed such as the *BFRJ-algorithm* [12] which traverses the indexes according to a breadth-first strategy.

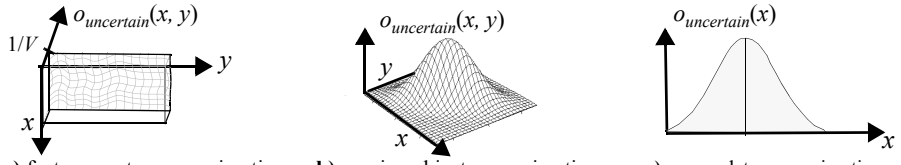
If no multidimensional index is available, it is possible to construct the index on the fly before starting the join algorithm. Several techniques for bulk-loading multidimensional index structures exist [5, 14]. The *seeded tree method* [19] joins two point sets provided that only one is supported by an R-tree. The partitioning of this R-tree is used for a fast construction of the second index on the fly. The *spatial hash-join* [20, 23] decomposes the set R into a number of partitions which is determined according to given system parameters.

A join algorithm particularly suited for similarity self joins is based on the ε-*kdB-tree* [25]. Koudas and Sevcik proposed the *Size Separation Spatial Join* [16] and the *Multidimensional Spatial Join* [15] which make use of space filling curves to order the points in a multidimensional space.

2.2 Query Processing of Uncertain Data

Many studies have focused on the management of uncertain data and on providing probabilistic queries on databases with uncertain data. A survey of the research area concerning uncertainty and incomplete information in databases is given in [1] and [22].

Recently, a lot of work has been published in the area of management and query processing of uncertain data in sensor databases [8] and especially in moving object envi-



a) feature-vector approximation **b)** moving object approximation **c)** sensor data approximation

Fig. 1. Uncertain object descriptions.

ronments [9, 26]. Similar to the approach presented in this paper, the approaches in [8, 9, 26] model uncertain data by means of probabilistic density functions (*pdf*s). In [26], for instance, moving objects send their new positions to the server, iff their new positions considerably vary from their last sent positions. Thus, the server always knows that an object can only be a certain threshold value away from the last sent position. The server, then assigns a pdf to each object reflecting the likelihood of the objects possible positions. Based on this information the server performs probabilistic range queries. Likewise, in [9] an approach is presented for probabilistic nearest neighbor queries. Note that both approaches assume non-uncertain query objects. Thus, they cannot be used as foundation for a join on uncertain objects where both query and database objects are uncertain.

Furthermore, most recently [10] an approach was proposed dealing with spatial query processing not on positionally uncertain data but on existentially uncertain data. This kind of data naturally occurs, if, for instance, objects are extracted from uncertain satellite images. The approach presented in this paper does not deal with existentially uncertain data but with positionally uncertain data which can be modelled by probability density functions.

Definition 2 uncertain object representation
Let $o \in D \subseteq IR^d$ be an object from a database. An *uncertain object representation* is a function $o_{uncertain}$: $IR^d \rightarrow IR_0^+ \cup \infty$, for which the following condition holds:

$$\int\limits_{IR^d} o_{uncertain}(v)dv = 1$$

In the following, we will show that the above definition is a generalization of existing object description techniques used to describe uncertain data.

Modelling Distributed Feature Vectors. In [13], feature vectors were grouped together to small clusters at client site. Then each cluster is represented by a feature vector and a covering-radius and this information was transmitted to the server. If we assume that V is the volume of the hyper-sphere belonging to the micro-cluster of object o, the uncertain object representation $o_{uncertain}$ assigns to each feature vector contained in the hyper-sphere a value of $1/V$ and to each feature vector outside of the hyper-sphere a value of 0. Note that all objects within such a micro-cluster have the same uncertain object representation.

In [18], an approach for distributed clustering of high-dimensional feature vectors was introduced. In order to save transmission cost, only certain dimensions of a feature vector were transmitted to the server. For the dimensions which were not transmitted, the server can limit the possible values by an interval. Thus, the server can individually

generate for each feature vector a conservative approximating box. If we assume that V is the volume of the box belonging to object o, the uncertain object representation $o_{uncertain}$ assigns to each feature vector contained in the box a value of $1/V$ and to each feature vector outside of the box a value of 0 (cf. Figure 1a). In this case, the uncertain object descriptions are different for the different objects.

Modelling Moving Objects. Technical problems with the GPS system, or outdated positional information force the server to approximate moving objects by one- or two-dimensional Gaussian probability density functions $o_{uncertain}$ (cf. Figure 1b). If we assume that the exact positions of the moving objects are available [27], the probability density functions $o_{uncertain}$ correspond to dirac-delta functions which assign to the exact position a value of infinity and to all other positions a value of 0.

Modelling Sensor Data. Many applications use sensors for monitoring values like wind speed, pressure or temperature. Due to continuous changes, a central database has at each time only approximated information of each of these attributes. In [8], it was suggested to model each of these values by appropriate density functions, which corresponds to a 1-dimensional uncertain object representation according to Definition 2 (cf. Figure 1c).

If clear from the context, we simply write o for the uncertain object representation $o_{uncertain}$ from now on. As already mentioned there exists a lot of work in the area of query processing on these uncertain object representations, but, to the best of our knowledge, there does not exist any work in the literature which tackles the complex problem of joining these uncertain objects.

3 Non-Probabilistic Similarity Join on Uncertain Data

Traditional join algorithms require distance functions which express the similarity between two objects by exactly one numerical value. Based on these traditional distance functions, the join algorithms decide for each object pair unambiguously whether it belongs to the result set or not. Usually, this decision is based on 'sharp' object representations, i.e. the objects are assumed to be certain.

In this section, we introduce distance functions which do not express the similarity between two objects by a single numerical value. Instead, we propose distance functions expressing the similarity between two objects by means of a probability density function which we call *probabilistic distance function*. This function describes the probability distribution of all possible distances between two objects. A one-dimensional example is depicted in Figure 2. Figure 2a shows two uncertain objects o and o' according to Definition 2. The distance between these two objects is described by a probabilistic distance function.

Definition 3 probabilistic distance function
Let $d: D \times D \rightarrow IR_0^+$ be a distance function, and let $P(a \leq d(o, o') \leq b)$ denote the probability that $d(o, o')$ is between a and b. Then a probabilistic density function $f_d: D \times D \rightarrow (IR_0^+ \rightarrow IR_0^+ \cup \infty)$ is called a *probabilistic distance function* if the following condition holds:

$$P(a \leq d(o, o') \leq b) = \int_a^b f_d(o, o')(x)dx$$

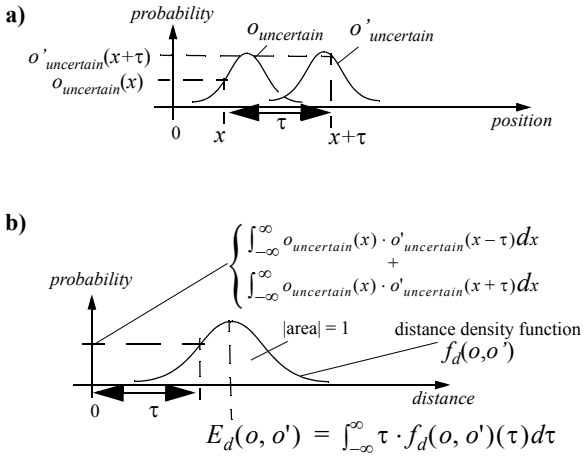


Fig. 2. Probabilistic distance function in an uncertain feature space
b) probabilistic distance function
a) reflecting the distance between two one-dimensional uncertain objects.

If the distance $\tau = d(o,o')$ between two objects can exactly be determined, the probabilistic distance function f_d is equal to the dirac-delta-function δ, i.e. $f_d(o, o')(x) = \delta(x-\tau)$ [7]. Thus, the traditional approach can be regarded as a special case of Definition 3. Let us point out that the probability distribution of each uncertain data item is considered independent.

As traditional join algorithms can only handle distance functions which yield a unique distance value, we propose to extract the expected distance value from these probabilistic distance functions. The expected distance value E_d: $O \times O \rightarrow IR_0^+$ represents the probabilistic distance function by one single value $E_d(o, o') = \int_{-\infty}^{\infty} x \cdot f_d(o, o')(x)dx$ (cf. Figure 2b).

Although, this expected distance value expresses the distance between two uncertain objects in an appropriate way, similarity joins based on this distance measure might be misleading. Look at the example shown in Figure 3 depicting 4 uncertain objects o_A, o_B, o_C and o_D having different uncertainties. On the right hand side of Figure 3 the corresponding probabilistic distance functions of the object pairs are shown. This example demonstrates that both objects o_B and o_D are within the ε-range of object o_A, when simply using the expected distances. Although, the probabilities that the objects o_B and o_C are within the ε-range of o_A are very similar, o_B belongs to the result set and o_C not. Furthermore, although the expected distance between the uncertain objects o_A and o_B is lower than the expected distance between the objects o_A and o_D, the probability that o_B is within the ε-range of o_A is much smaller than the probability for o_D. To sum up, similarity joins based on the expected distances are not able to take the uncertainty of the object representations into account and thus fail to produce meaningful results.

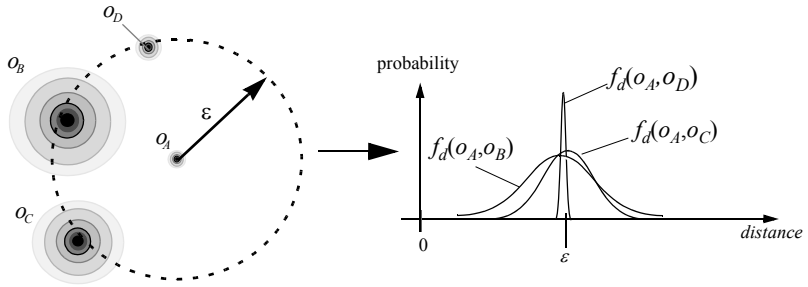


Fig. 3. Distance-range join based on the expected distance.

4 Probabilistic Similarity Join on Uncertain Data

As outlined in Section 3, a non-probabilistic similarity join on uncertain data has some limitations which are overcome by the probabilistic similarity join introduced in this section. The probabilistic similarity join is based on a direct integration of the probabilistic distance functions rather than using only aggregated values. Our new *probabilistic similarity join* assigns to each object pair a probability value reflecting the likelihood that the object pair belongs to the join result set.

Definition 4 probabilistic similarity join
Let R and S denote two relations, and let θ_d denote any similarity join predicate based on a given distance function d. Furthermore, let $P(r\ \theta_d\ s)$ denote the probability that $r\ \theta_d\ s$ is true for an object pair $(r, s) \in R \times S$. Then, the *probabilistic similarity join* $R \overset{\text{prob}}{\underset{\theta}{\bowtie}} S$ consists of object pairs $(r, s) \in R \times S$ for which $P(r\ \theta_d\ s) > 0$ holds, i.e.

$$R \overset{\text{prob}}{\underset{\theta}{\bowtie}} S = \{(r, s, P(r\ \theta_d\ s)) \mid P(r\ \theta_d\ s) > 0\} \subseteq R \times S \times [0,1]$$

4.1 Theoretical Foundations

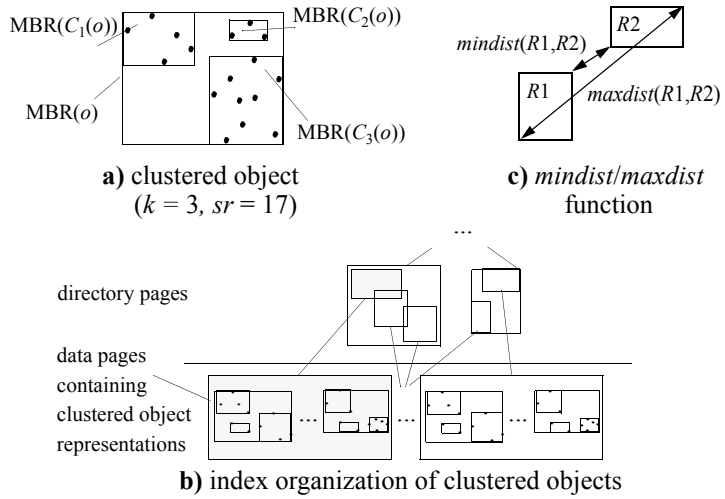
In this section, we shortly show how we can theoretically compute the probability value $P(o\ \theta_d^{dr}\ o')$[1] underlying the probabilistic distance-range join.

Lemma 1. Let $\varepsilon \in IR_0^+$ and let d be an arbitrary distance function between feature vectors. For each pair of uncertain object representations (o, o'), we can compute the probability $P(o\ \theta_d^{dr}\ o')$ based on their probabilistic distance function $f_d(o, o')$ as follows:

$$P(o\ \theta_d^{dr} o'\ ) = \int_{-\infty}^{\varepsilon} f_d(o, o')(x)dx$$

Proof: Lemma 1 directly follows from the definition of the distance-range join (cf. Definition 1) and the definition of the probabilistic distance function (cf. Definition 3). □

1. In the remainder of the paper θ_d^{dr} denotes the join predicate of the distance-range join.



a) clustered object (k = 3, sr = 17)

c) *mindist*/*maxdist* function

b) index organization of clustered objects

Fig. 4. Database Integration of Uncertain Data.

4.2 Computational Aspects

Although for some uncertain object representations it would be possible to compute the probabilistic similarity join directly on Lemma 1, we propose to compute it based on the generally applicable concept of monte-carlo sampling. In many applications the uncertain objects might already be described by a discrete probability density function, i.e. we have the sample set already. If the uncertain object is described by a continuos probability density function, we can easily sample according to this function and derive a set of samples. In the following, we assume that each object o is represented by a set of sr sample points, i.e. o is represented by sr different representations $\{o_1, ..., o_{sr}\}$. After having described how to organize these discrete object representations within a database (cf. Section 4.2.1), we show how to compute the probabilistic distance-range join (cf. Section 4.2.2) based on these discrete object representations.

4.2.1. Database Integration of Uncertain Data. In order to reduce the complexity of the join computations, we introduce efficient join variants which are based on groups of samples. Thereby two samples o_i and o_j of the same object o are grouped together to one cluster, if they are close to each other. We can generate such a clustering on the object samples by applying the partitioning clustering algorithm k-means [21] individually to each sample set $\{o_1, ..., o_{sr}\}$. Thus, an object is no longer approximated by sr samples, but by k clusters containing all the sr sample points of the object (cf. Figure 4a).

Definition 5 Clustered Object Representation

Let $\{o_1, ..., o_{sr}\}$ be a discrete object representation. Then, we call the set $\{\{o_{1,1}, ..., o_{1,n_1}\},..., \{o_{k,1}, ..., o_{k,n_k}\}\}$ a *clustered object representation* where $\bigcup_{i=1...k, j=1...n_i} o_{i,j} = \{o_1, ..., o_{sr}\}$ and $n_1+...+n_k = sr$.

We store these clustered object representations in R-tree [11] like index structures. Thereto, we determine the minimum bounding rectangle MBR($C_i(o)$) of each cluster

$C_i(o) = \{o_{i,1}, ..., o_{i,n_i}\}$, and the minimum bounding rectangle MBR(o) of $o = \{o_1, ..., o_{sr}\}$. Then, we store the clustered object representations as depicted in Figure 4b in a standard index structure suitable for managing spatially extended objects. In the following section, we assume that there exist two functions *mindist* and *maxdist* which return the minimal and the maximal distance between two rectangles, between two points, or between a point and a rectangle (cf. Figure 4c).

4.2.2. Distance Range Join. Managing the uncertain objects in R-tree like index structures (cf. Figure 4) enables us to carry out a distance-range join based on a parallel R-tree run as described in [4]. In general, we can use this approach without any changes regarding the way we use the hierarchical directory structure for pruning branches in the R-tree. The only difference is on the leaf level where we assign a probability value to each object pair. Figure 5a shows the algorithm for computing such a probability value.

Definition 4 requires that the result set of the probabilistic similarity join contains all objects having a probability value higher than 0. Sometimes, it is desirable that this result set is sorted in descending order of the probability values. A straightforward approach would be to determine the complete result set, and then sort it. The disadvantage of this approach is that we have to wait rather long for getting the first element of the result set. In the following, we present an approach which allows us to determine the first element of the sorted result set very efficiently. The basic idea is to adapt the optimal multi-step k-nearest-neighbor approach presented in [24] to our needs.

First, we carry out an approximated probabilistic distance-range join based on a parallel R-tree run and on the probability function presented in Figure 5b. Note that, especially for high sample rates s, this function can be computed much more efficiently than the one presented in Figure 5a. Obviously, for the result set of this approximated join the following lemma holds.

Lemma 2. Let $R \overset{\text{prob}}{\underset{dr,exact}{\bowtie}} S$ denote the result set of a probabilistic similarity join based on the probability function presented in Figure 5a, and let $R \overset{\text{prob}}{\underset{dr,filter}{\bowtie}} S$ denote the result set of a probabilistic similarity join based on the probability function presented in Figure 5b. Then the following statement holds:

$$(r, s, p_{dr,exact}(r,s)) \in R \overset{\text{prob}}{\underset{dr,exact}{\bowtie}} S \Rightarrow \exists (r, s, p_{dr,filter}(r,s)) \in R \overset{\text{prob}}{\underset{dr,filter}{\bowtie}} S: p_{dr,exact}(r,s) \leq p_{dr,filter}(r,s)$$

Proof. As both join variants run through the R-tree directory in the same way, they only differ in the computation of the probability values of object pairs. If we assume an object pair (r,s), the value $p_{dr,filter}(r,s)$ (cf. Figure 5b) is always equal to or larger than $p_{dr,exact}(r,s)$ (cf. Figure 5a), as *mindist* (MBR($C_i(s)$), MBR($C_{i'}(r)$)) ≤ *maxdist* (MBR($C_i(s)$), MBR($C_{i'}(r)$)) holds. Furthermore, if *mindist* (MBR($C_i(s)$), MBR($C_{i'}(r)$)) > ε holds, then for all sample points $s_{i,j}$ and $r_{i',j'}$ the distance $dist(s_{i,j}, r_{i',j'})$ is also larger than ε. □

Next, we sort the set $R \overset{\text{prob}}{\underset{dr,filter}{\bowtie}} S$ in descending order according to the filter probability values. In the refinement step, we incrementally walk through this sorted list and compute the exact probability values $p_{dr,exact}(r,s)$ as shown in Figure 5a. If the filter probability value $p_{dr,filter}$ of the currently considered join candidate pair is smaller or equal to the maximal exact probability value computed so far, we can immediately report the object pair having the maximal exact probability value. If we have already refined further object pairs for which the exact probability value is equal or higher than $p_{dr,filter}$, then all these

a)

```
FUNCTION p_dr,exact /* computes the exact probability */
INPUT:
o = {{o_1,1, ..., o_1,n_1},...,{o_k,1, ..., o_k,n_k}}clustered_uncertain_object,
o' = {{o'_1,1, ..., o'_1,n'_1},..., {o'_k,1, ..., o'_k,n'_k}}clustered_uncertain_object
OUTPUT: numerical value p∈[0..1];

BEGIN
  IF mindist (MBR(o), MBR(o')) > ε THEN
    RETURN 0
  ELSE IF maxdist (MBR(o), MBR(o')) ≤ ε THEN
    RETURN 1
  ELSE BEGIN
    probability := 0;
    FOR i = 1 TO k DO
      FOR i' = 1 TO k DO
        IF maxdist (MBR(C_i(o)), MBR(C_i'(o'))) ≤ ε THEN
          probability := probability + n_i · n'_i';
        ELSE
          FOR j = 1 TO n_i DO
            FOR j' = 1 TO n'_i' DO
              IF dist (o_i,j , o'_i',j') ≤ ε THEN
                probability := probability + 1;
    RETURN probability / sr^2;
  END;
END.
```

b)

```
FUNCTION p_dr,filter /* computes the filter probability */
INPUT:
o = {{o_1,1, ..., o_1,n_1},...,{o_k,1, ..., o_k,n_k}}clustered_uncertain_object,
o' = {{o'_1,1, ..., o'_1,n'_1},..., {o'_k,1, ..., o'_k,n'_k}}clustered_uncertain_object
OUTPUT: numerical value p∈[0..1];

BEGIN
  IF mindist (MBR(o), MBR(o')) > ε THEN
    RETURN 0
  ELSE IF maxdist (MBR(o), MBR(o')) ≤ ε THEN
    RETURN 1
  ELSE BEGIN
    filter_probability := 0;
    FOR i = 1 TO k DO
      FOR i' = 1 TO k DO
        IF mindist (MBR(C_i(o)), MBR(C_i'(o'))) ≤ ε THEN
          filter_probability := filter_probability + n_i · n'_i';
    RETURN filter_probability / sr^2;
  END;
END.
```

Fig. 5. Probability functions underlying the distance-range join.
a) Computation of the exact probability b) Computation of the filter probability

pairs can be reported before starting the next refinement. Obviously, this process can iteratively be continued until the user decides that he has received enough object pairs.

5 Experimental Evaluation

In this section, we examine the effectiveness, i.e. the quality, and the efficiency of our proposed probabilistic similarity join approach. The efficiency of our approach was measured by the number of required distance computations which dominate the overall runtime cost. The depicted cost concerning the *probabilistic distance-range join* experiments reflect the overall number of required distance computations.

5.1 Experimental Setup

The following experiments are based on artificial datasets, each consisting of a set of 3- and 10-dimensional uncertain feature vectors. Additionally, we also applied our approaches to two distributed real-world datasets PLANE and PDB where the feature vectors were described by multi-dimensional boxes according to [18].

ART*d*(*u*) datasets. Each of these artificial datasets contains 1000 uncertain objects distributed equally in a *d*-dimensional normalized data space. Thereby the parameter *u* denotes the grade of uncertainty of the objects in the dataset. The uncertainty of the objects, i.e. the maximal variance of the feature values, is measured relatively to the data space. In our experiments, we used two different settings for the uncertainty. For the ART3 data set, *u* = *'low'* denotes an uncertainty of 3% of the data space and *u* = *'high'* denotes an uncertainty of 5% of the data space. For the ART10 data set, *u* = *'low'* denotes an uncertainty of 3% of the data space and *u* = *'high'* denotes an uncertainty of 4% of the data space.

PLANE dataset. The real world dataset PLANE consists of 1000 high-resolution 3D CAD objects provided by our industrial partner, an American airplane manufacturer. Each object is represented by a 42-dimensional feature vector which is derived from the cover sequence model as described in [17]. The average uncertainty of the PLANE data set is 1% of the data space.

PDB dataset. This 3D protein structure dataset is a real world dataset derived from the Brookhaven Protein Data Bank (PDB) [6]. The 1000 objects are represented by 3D shape histograms [2] resulting in a 120-dimensional feature vector per object. The average uncertainty of the PDB data set is 4% of the data space.

For the sampling of the possible object positions we assumed an equal distribution within the corresponding uncertainty areas. All *d*-dimensional datasets are normalized w.r.t. the unit space $[0,1]^d$. As distance measure we used the L_1-distance (Manhattan distance). We performed a self-join on the datasets where the ε-distance was set to 3% of the dataspace for all datasets, except of the PDB dataset for which we set ε = 1%. If not stated otherwise, the size of the sample set of each uncertain object is initially set to 25 samples which are approximated by 7 clusters.

5.2 Experiments on the Sample Rate

In the first experiments, we examined the quality of our similarity join approaches by varying the number of used samples per object. We noticed that for sample rates higher than 100 the resulting probability values do not change any more considerably. Therefore, we used the probabilistic similarity join result $R_{exact} = \{(r, s, P_{exact}(r\ \theta_d\ s))| P_{exact}(r\ \theta_d\ s) > 0\}$ (cf. Definition 4) based on 100 samples as reference join result for measuring the error of the probabilistic similarity join results $R_{approx} = \{(r, s, P_{approx}(r\ \theta_d\ s)) |$

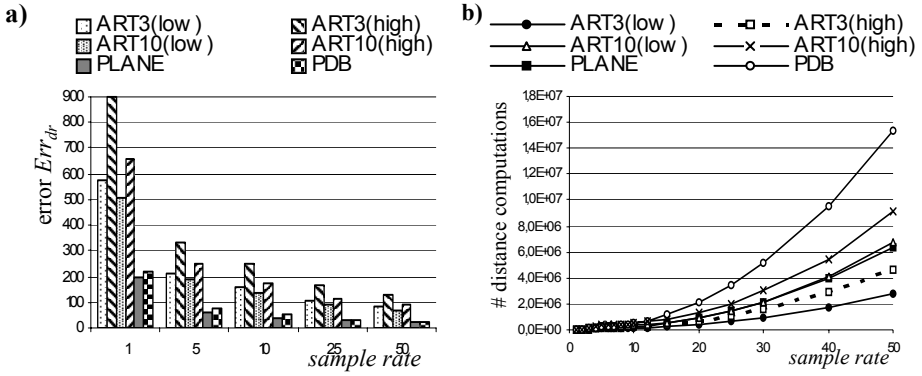


Fig. 6. Influence of the sample rate *sr*
a) Error b) Number of distance computations

$P_{approx}(r\ \theta_d\ s) > 0\}$ based on sample rates $sr < 100$. The used error measure Err_{dr} for the distance-range joins is defined as follows: $Err_{dr}(R_{approx}, R_{exact}) =$

$$\sum_{(r,s) \in R \times S} \left| P_{approx}(r\ \theta_d^{dr}\ s) - P_{exact}(r\ \theta_d^{dr}\ s) \right|$$

Figure 6a shows the error of the *probabilistic distance-range join* for a varying sample rate *sr*. The figure shows clearly that the error decreases rapidly with increasing sample rates. At a sample rate $sr = 10$ the error is less than half the size compared to the error at $sr = 1$ for all datasets. Furthermore, comparing the artificial datasets with high uncertainties (ART*d*(*high*)) to those with low uncertainties (ART*d*(*low*)), we can observe that a higher uncertainty leads to a higher error.

In the next experiment, we investigated how the sample rate influences the cost of the join processing. Figure 6b shows the number of distance computations required to perform the join for varying sample rates. We set the number *k* of clusters to 5 for a sample rate *sr* higher than 5, otherwise we set $k = sr$. The cost increase superlinear with increasing sample rates *sr*. For high sample rates, the good quality (cf. Figure 6a) goes along with high join cost (cf. Figure 6b). In particular, the join processing on datasets with high uncertainty (ART*d*(*high*)) does not only lead to a lower quality of the results but is also more expensive than the processing on more accurate datasets (ART*d*(*low*)). Altogether, we achieve a good trade-off between the quality of the results and the required cost when using a sample rate of $sr = 25$.

5.3 Experiments on the Efficiency

In this subsection, we examine the runtime performance of our probabilistic join approach. At first, we consider the runtime behavior for different sample rates *sr* and varying number of clusters *k*. The experimental results are depicted in Figure 7. On the one hand, when using only one cluster per object ($k = 1$), we have only a few clusters for which we must compute the distances between them. On the other hand, the refinement of these clusters is very expensive. When using one cluster per object, the cluster covers the entire uncertain object, i.e. it has a large extension. The probability that the ε-range value is between the *mindist* value and the *maxdist* value of a pair of such clusters is very

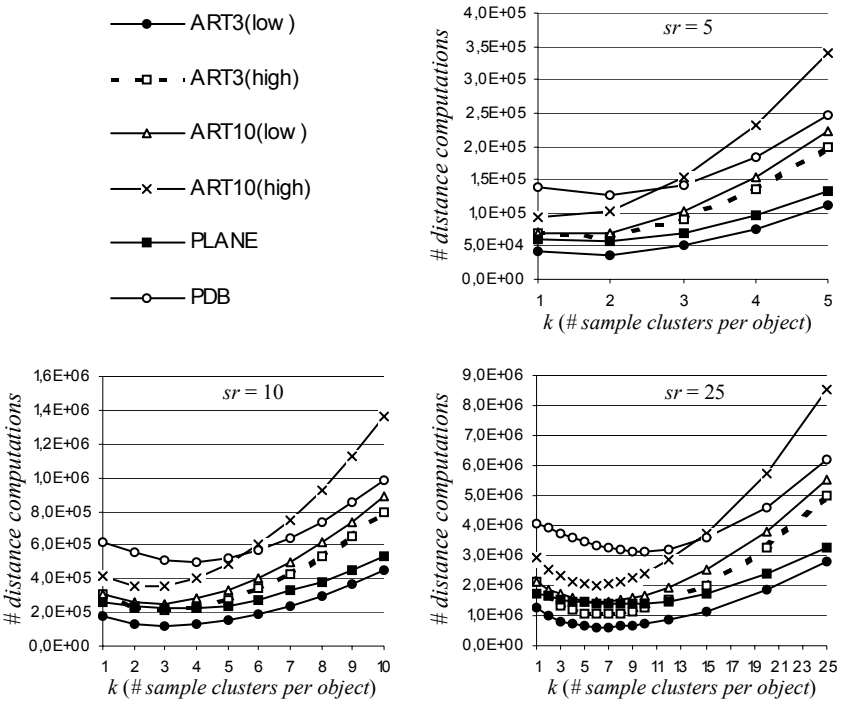


Fig. 7. Runtime performance for varying number of sample clusters

high, i.e. a lot of cluster pairs have to be refined. Very small clusters ($k = s$) also lead to an expensive join processing, because we have to compute a lot of distances between pairs of clusters when refining the object pairs. The best trade-off for k can be achieved somewhere in between these two extremes. As depicted in Figure 7, the optimal setting for k depends on the used sample rate. Generally, the higher the used sample rate sr, the higher is the optimal value for k.

In the next experiments, we demonstrate the advantages of the filter step when enabling a ranked output of the results in descending order of their probabilities. As mentioned in Section 4.2.2, the proposed filter step enables an early output of the first join results. Figure 8a depicts the performance of the ranked distance-range join w.r.t. the probabilities of the results. Only 25% of the distance computations are required to output all certain results, i.e. results having a probability higher than 95%. Only 70% of the distance computations are required to output all results having a probability higher than 50%. The join cost w.r.t. the number of returned results are depicted in Figure 8b. Only 45% of the distance computations are required to return the first 10% of the result set. The proposed incremental join processing is particularly useful when the user wants to stop the query after getting either the most significant results or a small portion of the result set.

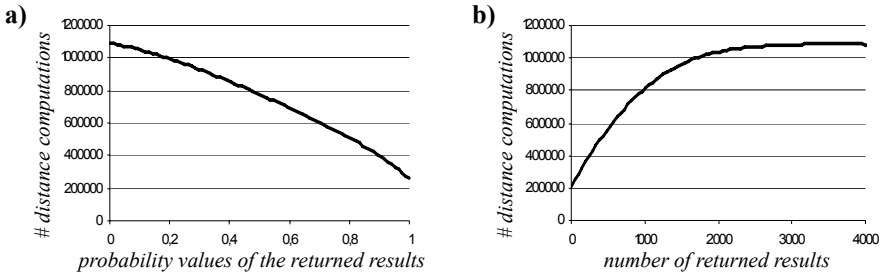


Fig. 8. Runtime ranked distance-range join (ART3(high))
a) performance with respect to the result probability
b) performance with respect to the number of returned results

6 Conclusions

Similarity query processing on uncertain data is an important emerging topic in many modern database application areas. In this paper, we introduced the general concept of *probabilistic similarity joins* on uncertain objects which assign to each object pair a probability value indicating the likelihood that it belongs to the result set. In particular, we introduced in detail how to compute these probability values for the distance-range join. We showed how this similarity join can effectively be carried out based on the generally applicable concept of monte-carlo sampling. In order to improve the efficiency of the proposed probabilistic similarity join, we determined appropriate approximations of the object samples by means of clustering. Based on these approximations, the proposed probabilistic distance-range join algorithm also supports an incremental report of the join results ranked in descending order of their probability values. In a detailed experimental evaluation based on artificial and real-world data sets, we demonstrated that the incremental probabilistic distance-range join allows to report the most significant join results very early.

In our future work, we plan to extend our probabilistic algorithm to further similarity join predicates, e.g. the nearest-neighbor and reverse-nearest neighbor predicates. Furthermore, we will show that probabilistic similarity joins can beneficially be used as a basic operation for various data mining algorithms, e.g. clustering and classification algorithms, which have to process uncertain data.

References

1. Abiteboul S., Hull R., Vianu V.: *Foundations of Databases.* Addison Wesley, 1995.
2. Ankerst M., Kastenmüller G., Kriegel H.-P., Seidl T.: *3D Shape Histograms for Similarity Search and Classification in Spatial Databases*. SSD'99.
3. Böhm C., Braunmüller B., Breunig M., Kriegel H.-P.: *High Performance Clustering Based on the Similarity Join*. CIKM'00.
4. Brinkhoff T., Kriegel H.P., Seeger B.: *Efficient Processing of Spatial Joins Using R-trees.* SIGMOD '93.

5. van den Bercken J., Seeger B., Widmayer P.: *A General Approach to Bulk Loading Multidimensional Index Structures.* VLDB'97.
6. Bernstein F. C., Koetzle T. F., Williams G. J., Meyer E. F., Brice M. D., Rodgers J. R., Kennard O., Shimanovichi T., Tasumi M.: *The Protein Data Bank: a Computer-based Archival File for Macromolecular Structures.* Journal of Molecular Biology, Vol. 112 (1977).
7. Bracewell R.: *The Impulse Symbol*. Ch. 5 in The Fourier Transform and Its Applications, 3rd ed.: McGraw-Hill, 1999.
8. Cheng R., Kalashnikov D.V., Prabhakar S.: *Evaluating probabilistic queries over imprecise data.* SIGMOD'03.
9. Cheng R., Kalashnikov D. V., Prabhakar S.: *Querying imprecise data in moving object environments.* IEEE Transactions on Knowledge and Data Engineering, 2004.
10. Dai X., Yiu M., Mamoulis N., Tao Y., Vaitis M.: *Probabilistic Spatial Queries on Existentially Uncertain Data.* SSTD'05.
11. Guttman A.: *R-trees: A Dynamic Index Structure for Spatial Searching.* SIGMOD'84.
12. Huang Y.-W., Jing N., Rundensteiner E. A.: *Spatial Joins Using R-trees: Breadth-First Traversal with Global Optimizations.* VLDB'97.
13. Januzaj E., Kriegel H.-P., Pfeifle M.: *Scalable Density-Based Distributed Clustering.* PKDD'04.
14. Kamel I., Faloutsos C.: *Hilbert R-tree: An Improved R-tree using Fractals.* VLDB'94.
15. Koudas N., Sevcik K.: *High Dimensional Similarity Joins: Algorithms and Performance Evaluation.* ICDE'98.
16. Koudas N., Sevcik K.: *Size Separation Spatial Join.* SIGMOD'97.
17. Kriegel H.-P., Brecheisen S., Kröger P., Pfeifle M., Schubert M.: *Using Sets of Feature Vectors for Similarity Search on Voxelized CAD Objects.* SIGMOD'03.
18. Kriegel H.-P., Kunath P., Pfeifle M., Renz M.: *Approximated Clustering of Distributed High Dimensional Data.* PAKDD'05.
19. Lo M.-L., Ravishankar C. V.: *Spatial Joins Using Seeded Trees.* SIGMOD'94.
20. Lo M.-L., Ravishankar C. V.: *Spatial Hash Joins.* SIGMOD'96.
21. McQueen J.: *Some Methods for Classification and Analysis of Multivariate Observations*. In 5th Berkeley Symp. Math. Statist. Prob., volume 1, 1967.
22. Motro A.: *Management of Uncertainty in Database Systems.* In Modern Database Systems, Won Kim (Ed.), Addison Wesley, 1995.
23. Patel J.M., DeWitt D.J.: *Partition Based Spatial-Merge Join*. SIGMOD'96.
24. Seidl T., Kriegel H.-P: *Optimal Multi-Step k-Nearest Neighbor Search.* SIGMOD'98.
25. Shim K., Srikant R., Agrawal R.: *High-Dimensional Similarity Joins.* ICDE'97.
26. Wolfson O., Sistla A. P. , Chamberlain S., Yesha Y.: *Updating and Querying Databases that Track Mobile Units.* Distributed and Parallel Databases, 7(3), 1999.
27. Yiu M. L., N. Mamoulis N.: Clustering Objects on a Spatial Network. SIGMOD'04, pp. 443-454.
28. Zhao W., Chellappa R., Phillips P.J., Rosenfeld A.: *Face Recognition: A literature survey.* ACM Computational Survey, 35(4), 2000.