PROUD: A Probabilistic Approach to Processing Similarity Queries over Uncertain Data Streams

Mi-Yen Yeh
Dept. of Electrical Engineering
National Taiwan University
miyen@arbor.ee.ntu.edu.tw

Kun-Lung Wu
IBM T.J. Watson Research Center
klwu@us.ibm.com

Philip S. Yu
Dept. of Computer Science
University of Illinois at Chicago
psyu@cs.uic.edu

Ming-Syan Chen
Dept. of Electrical Engineering
National Taiwan University
mschen@cc.ee.ntu.edu.tw

ABSTRACT

We present *PROUD* - A PRObabilistic approach to processing similarity queries over Uncertain Data streams, where the data streams here are mainly time series streams. In contrast to data with certainty, an uncertain series is an ordered sequence of random variables. The distance between two uncertain series is also a random variable. We use a general uncertain data model, where only the mean and the deviation of each random variable at each timestamp are available. We derive mathematical conditions for progressively pruning candidates to reduce the computation cost. We then apply PROUD to a streaming environment where only sketches of streams, like wavelet synopses, are available. Extensive experiments are conducted to evaluate the effectiveness of PROUD and compare it with Det, a deterministic approach that directly processes data without considering uncertainty. The results show that, compared with Det, PROUD offers a flexible trade-off between false positives and false negatives by controlling a threshold, while maintaining a similar computation cost. In contrast, Det does not provide such flexibility. This trade-off is important as in some applications false negatives are more costly, while in others, it is more critical to keep the false positives low.

1. INTRODUCTION

Recently, there is a growing amount of research interest in uncertain data. Explicitly, there are research results reported on the query processing in uncertain database [3, 5, 8, 12, 15, 23, 25, 26, 27, 30, 34], indexing uncertain data [2, 4, 6, 9, 7, 20, 28, 29, 31] , privacy preserving with uncertain data [1], sketch and aggregate processing in probabilistic data streams [10, 17], and so on. Uncertainty in data comes from various sources. To protect privacy, people deliberately introduce disturbance to the confidential data before further processing. In a sensor network, sensor readings are interfered with noise generated by the equipment itself or other exterior influences. The readings here could be the temperature measurements, or the location or speed of moving objects. In this application, a false negative (e.g., not discovering speeding or equipment over-heating) is much less desired than a false positive. On the other hand, in some situation, the false negatives tend to be more tolerable. For example, in mobile network applications, location privacy is an important issue. To protect locations of users in wireless network, the telecommunication companies might blur the true position data of users to other applications. In this case, a false negative (i.e., over-protecting) tends to be more acceptable than a false positive which could lead to privacy leak. Therefore, a proper control threshold on the false positives and the false negatives is indeed application-dependent and very important to achieve the desired functions of such applications.

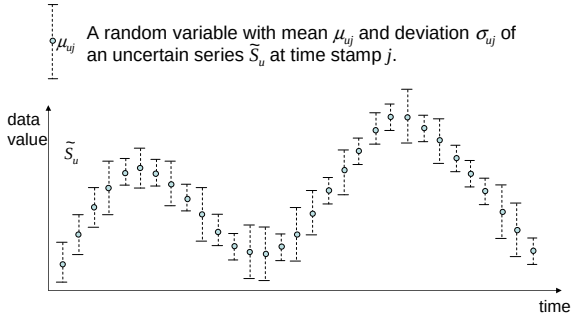

Figure 1: Uncertain time series model.

In this paper, we study the problem of processing similarity queries over uncertain time series. In time series databases, uncertainty also exists. In the process of data collection, the data value at each timestamp can be blurred with uncertainty. The uncertainty at each time point can be modeled as a continuous random variable, in which the exact probability distribution is unknown. In contrast to data with certainty, each time series now is considered as an ordered sequence of random variables. Furthermore, the possible value of each random variable is continuous, which is different from those tuples with limited possible values in probabilistic databases. Hence, it becomes a big challenge to find the distance between two uncertain data series.

This problem was first studied in [19], by Lian et al., in the context of time series. They treated the time series with uncertainty as *cloaked time series*. A general model of the cloaked time se-

EDBT 2009, March 24–26, 2009, Saint Petersburg, Russia.

ries was proposed with a high dimension (e.g. 128), similar to the one shown in Fig. 1. At each time point, no detailed distribution of each random variable is known except the mean and the deviation. Based on this assumption, they defined a new pattern matching query over the cloaked time series databases. They further sped up the matching process by taking advantage of R-tree indexing.

However, many applications have endless data streams, instead of time series with a fixed length. For example, daily stock trading prices, daily temperature observations, and video surveillance data are potentially limitless streams. In this case, on-line summarization, like wavelet synopses, is usually maintained, instead of off-line indexing, like R-trees. Hence, we need to find a new way to deal with uncertainty in processing similarity queries in a streaming environment.

For the above reasons, we present PROUD - A PRObabilistic approach to processing similarity queries over Uncertain Data streams. First we discuss the case when raw uncertain time series are processed. Then we discuss how to apply PROUD in a streaming environment when only synopses are available. Given a reference uncertain series, and a specified time range, we report the series with a high enough probability that their distances to the referenced one are within a given distance bound. The user can decide how high the probability is acceptable by controlling a probability threshold. Based on the same general uncertain data model as shown in Fig. 1 and similar assumptions as those in [19], i.e., only the mean and the deviation of each random variable at each timestamp are available, the statistics of the distance between two uncertain series are computed. The uncertain distance between two sequences of random variables is also a random variable. First we describe how to obtain the expected value and the variance of this distance random variable. Then, by using the central limit theorem, we show how this distance random variable can be modeled as a normal distribution. Finally, based on the normal distribution, we derive mathematical conditions for progressively pruning candidates to reduce the computation cost of PROUD.

In a streaming environment, usually only the summarization of data, instead of raw data, are retained. Therefore, how to compute the statistics of distance between two uncertain streams directly using these synopses is an important problem. Based on the way of processing raw uncertain data, we then apply PROUD to stream synopses, in particular, the Haar wavelet-based synopses. It is noted that the uncertain streams we discussed here are time series data streams with continuous uncertainties, which are different from the streams constituted of probabilistic tuples drawn from a finite domain indicated in [10, 17].

To evaluate the effectiveness of PROUD, we conduct extensive experiments using both real and synthetic data. For comparisons, we implement a deterministic method for computing distances, referred to as *Det*. In Det, the distance is directly computed from uncertain data, treating them as if they were data without uncertainty. We measure both the quality of solution and the computation time cost. The results show that both have similar computation costs. However, PROUD offers a flexible trade-off between false negatives and false alarms by controlling a probability threshold parameter. If the user sets a high probability threshold, only candidates with a very high probability that their distances to the reference stream are closer than the given distance bound are retained. As a result, it may reduce the false alarms and increase the false dismissals. On the contrary, for a low probability threshold, it may reduce the false negatives and increase the false alarms. This is the flexibility that Det does not have. The trade-off is important as in some cases false negatives are more costly, while in others, it is more critical to keep the false positives low. Users can then decide how high the probability threshold should be depending on their specific applications.

Our contributions can be summarized as follows:

- We presented PROUD - a new probabilistic approach to processing similarity queries within an arbitrary time range among multiple uncertain streams.
- We demonstrated how various probabilistic theories can help us deal with similarity query processing over multiple uncertain data streams.
- We showed how to process probabilistic similarity queries using only the stream synopses, for example, the wavelet-based synopses, instead of raw data.
- We conducted extensive studies to show that PROUD provides a flexible trade-off between false negatives and false positives by controlling a probability threshold, while Det does not.

The remainder of this paper is organized as follows. The problem definition is given in Section 2. In Section 3, we describe how to compute various statistics of the distance random variable. Section 4 describes how PROUD is applied directly on the stream synopses. The experimental studies are presented in Section 5. Finally, the work is concluded in Section 6.

2. PROBLEM STATEMENT

In this section, we first describe the problem of range-specified similarity queries over multiple series. The basic notations and definitions are introduced as follows.

- A time series S_u is an ordered sequence of data values: S_u = [d_{u1}, d_{u2},... ,d_{un}], where at time j, $S_u[j] = d_{uj}$.
- An *uncertain* time series $\widetilde{S}_u$ is a time series containing uncertainty at each time point. Given a time j, the value of the uncertain time series $\widetilde{S}_u[j]$ is modeled as:

$$\widetilde{S}_u[j] = d_{uj} + e_{uj},$$

where d_{uj} is the true data value and e_{uj} is the error.

In general, the error e_{uj} could be drawn from any arbitrary probability distribution. Hence, we can treat $\widetilde{S}_u[j]$ as a random variable at time j. However, we do not know the detailed distribution of the random variable $\widetilde{S}_u[j]$. We usually only have its mean μ_{uj} and deviation σ_{uj}. In addition, all the random variables at different timestamps are assumed to be independent.

For the similarity measure between two series, we adopt the commonly used Euclidean distance in this paper. The Euclidean distance between two series S_u and S_v given a specified time range $T = [t_s, t_e]$ is:

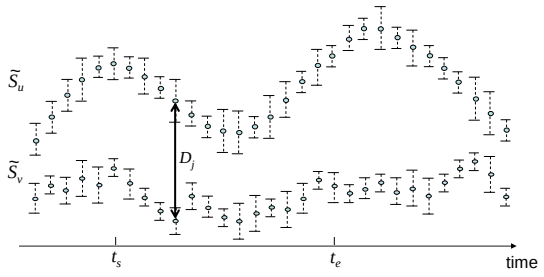

Figure 2: The probabilistic distance model.

$$dst(S_u, S_v)|_{t_s}^{t_e} = \sqrt{\sum_{j=t_s}^{t_e} (S_u[j] - S_v[j])^2}. \quad (1)$$

In deterministic case, given a reference series S_{ref} and a distance bound r, the similarity query is to find those series S_u's that satisfy the following:

$$dst(S_{ref}, S_u)|_{t_s}^{t_e} \leq r. \quad (2)$$

However, we cannot compute an exact value of the Euclidean distance between two uncertain time series. Instead, the uncertain distance between two uncertain series is also a random variable. Therefore, we define a new probabilistic similarity queries over uncertain series as follows.

DEFINITION 1. *Time-range-specified Probabilistic similarity query: Given an uncertain reference time series $\widetilde{S}_{ref}$, a specified time range $T = [t_s, t_e]$, a distance bound r, and a probabilistic threshold τ, we report those series $\widetilde{S}_u$'s that satisfy the following equation:*

$$\Pr(dst(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_e} \leq r) \geq \tau, \text{ where } \tau \in (0, 1]. \quad (3)$$

Instead of looking for similar series using the exact Euclidean distance, we now process the probabilistic similarity query according to the the cumulative distribution information of the distance. According to the given distance bound r, if an uncertain series $\widetilde{S}_u$ has a probability higher than the threshold τ that its distance to the reference one $\widetilde{S}_{ref}$ is not bigger than r, then $\widetilde{S}_u$ is a qualified candidate.

3. SIMILARITY QUERY PROCESSING

To deal with the probabilistic similarity query defined in Definition 1, we can first transform it to the following one:

$$\Pr([dst(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_e}]^2 \leq r^2) \geq \tau, \text{ where } \tau \in (0, 1]. \quad (4)$$

As a result, for ease of exposition, when we use the word "distance" from now on, we mean the square of the Euclidean distance, denoted as $Dst(\widetilde{S}_u, \widetilde{S}_v)$.

DEFINITION 2. *Square of Euclidean distance: We define the square of the Euclidean distance between two uncertain series $\widetilde{S}_u$ and $\widetilde{S}_v$ as $Dst(\widetilde{S}_u, \widetilde{S}_v)$.*

$$Dst(\widetilde{S}_u, \widetilde{S}_v)|_{t_s}^{t_e} = [dst(\widetilde{S}_u, \widetilde{S}_v)|_{t_s}^{t_e}]^2. \quad (5)$$

3.1 Statistics Computation of Uncertain Distance

In this section, we will show how to compute the related statistics, i.e, the expected value and the variance of the uncertain distance between two uncertain streams. To give a clearer illustration, we temporarily omit the notation $|_{t_s}^{t_e}$ from now on in our description, and will add it back later.

As shown in Fig. 2, we can treat the distance between two uncertain streams as a sum of a sequence of random variables:

$$Dst(\widetilde{S}_u, \widetilde{S}_v) = \sum_j D_j^2, \quad (6)$$

where D_j is a random variable representing $(\widetilde{S}_u[j] - \widetilde{S}_v[j])$.

From this formula, $Dst(\widetilde{S}_u, \widetilde{S}_v)$, which is a function of a series of independent random variables D_j, is also a random variable itself. D_j's of different j's are also independent random variables. Suppose the mean and the variance of the random variable D_j^2 is $E(D_j^2)$ and $Var(D_j^2)$, respectively, according to the *Central Limit Theorem* [32], the normal form variate,

$$Dst(\widetilde{S}_u, \widetilde{S}_v)_{norm} \equiv \frac{Dst(\widetilde{S}_u, \widetilde{S}_v) - \sum_j E(D_j^2)}{\sqrt{\sum_j Var(D_j^2)}}, \quad (7)$$

has a limiting cumulative distribution function that approaches a normal distribution. In other words, the distance $Dst(\widetilde{S}_u, \widetilde{S}_v)$ is a random variable which approaches a normal distribution with a corresponding mean and variance

$$Dst(\widetilde{S}_u, \widetilde{S}_v) \sim N(\sum_j E(D_j^2), \sum_j Var(D_j^2)). \quad (8)$$

This makes it possible that, regardless of the probability distributions of the original random variables, we can directly use the normal distribution to model the distance random variable $Dst(\widetilde{S}_u, \widetilde{S}_v)$. Furthermore, using this central limit theorem, we can efficiently decide if a candidate series has a high enough probability of being close to the referenced series within the given distance bound r^2.

The expected value of the distance random variable $Dst(\widetilde{S}_u, \widetilde{S}_v)$ can be computed as follows:

$$\begin{aligned} & E(Dst(\widetilde{S}_u, \widetilde{S}_v)) \\ = & \sum_j E(D_j^2) \\ = & \sum_j (E(\widetilde{S}_u^2[j]) - 2E(\widetilde{S}_u[j] \cdot \widetilde{S}_v[j]) + E(\widetilde{S}_v^2[j])). \end{aligned} \quad (9)$$

Given the computational formula for the variance, the mean value of $E(\widetilde{S}_u^2[j])$ is:

$$\begin{aligned} E(\widetilde{S}_u^2[j]) &= (E(\widetilde{S}_u[j]))^2 + Var(\widetilde{S}_u[j]) \\ &= \mu_{uj}^2 + \sigma_{uj}^2 \end{aligned} \quad (10)$$

Also, by our assumption, the random variables of different timestamps are all independent. Therefore, Eq.(9) becomes:

$$\begin{aligned} & E(Dst(\widetilde{S}_u, \widetilde{S}_v)) \\ = & \sum_j ((\mu_{uj}^2 + \sigma_{uj}^2) - 2(\mu_{uj} \cdot \mu_{vj}) + (\mu_{vj}^2 + \sigma_{vj}^2)). \end{aligned} \quad (11)$$

Next, we need to know how to compute the variance of the distance random variable $Dst(\widetilde{S}_u, \widetilde{S}_v)$. Since D_j^2's of different j's are independent of each other, the variance of $Dst(\widetilde{S}_u, \widetilde{S}_v)$ is $\sum_j Var(D_j^2)$. Hence once we get the value of $Var(D_j^2)$, we can compute the variance of the total distance. However, at certain timestamp j, we only know the variance of each random variable $\widetilde{S}_u[j]$ or $\widetilde{S}_v[j]$. What would be the variance of the function which is composed of these variables?

To solve this problem, we use the *delta method* [22], which is an important technique used in the Statistics field. Essentially, the delta method uses the second-order *Taylor series expansion* to expand a function of a random variable about the mean of that random variable. Then it takes the variance of the expansion result. For example, to compute the variance of function $f(X)$, first we expand $f(X)$ about the mean μ of X:

$$f(X) = f(\mu) + (X - \mu)f'(\mu),$$

where $f'()$ is $df()/dX$.

Therefore, the variance of $f(X)$ is

$$Var(f(X)) \approx Var(X) \cdot (f'(\mu))^2.$$

If we want to compute the variance of a function of more than one variables, it can be computed as follows:

$$Var(f(\vec{X})) \approx [\frac{\partial f(\vec{\mu})}{\partial X_i}] \cdot \Omega \cdot [\frac{\partial f(\vec{\mu})}{\partial X_i}]^T,$$

where $\vec{X}$ is a vector of random variables X_i, $\vec{\mu}$ is the vector of means of those variables, and Ω is the covariance matrix of those random variables.

Back to our case, the variance of $D_j^2 = (\widetilde{S}_u[j] - \widetilde{S}_v[j])^2$ is derived as follows:

$$\begin{aligned} & \begin{bmatrix} 2(\mu_{uj} - \mu_{vj}) & -2(\mu_{uj} - \mu_{vj}) \end{bmatrix} \cdot \begin{bmatrix} \sigma_{uj}^2 & 0 \\ 0 & \sigma_{vj}^2 \end{bmatrix} \cdot \\ & \begin{bmatrix} 2(\mu_{uj} - \mu_{vj}) \\ -2(\mu_{uj} - \mu_{vj}) \end{bmatrix} \\ = & \quad 4(\sigma_{uj}^2 + \sigma_{vj}^2) \cdot (\mu_{uj} - \mu_{vj})^2. \end{aligned} \quad (12)$$

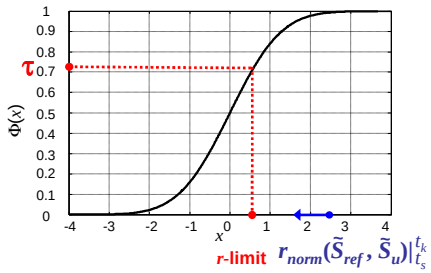

Figure 3: Cumulative distribution function of a normal distribution.

Finally, the variance of $Dst(\widetilde{S}_u, \widetilde{S}_v)$ is:

$$\begin{aligned} & Var(Dst(\widetilde{S}_u, \widetilde{S}_v)) \\ = & \sum_j Var(D_j^2) \\ = & \sum_j 4(\sigma_{uj}^2 + \sigma_{vj}^2) \cdot (\mu_{uj} - \mu_{vj})^2. \end{aligned} \quad (13)$$

In order to provide a better overview of the symbols and its related statistics used in our paper, we summarize them in the Table 1.

3.2 Candidate Selection

To find qualified candidates of Eq. (4), we need to know the cumulative distribution function (cdf) of the random variable $Dst(\widetilde{S}_{ref}, \widetilde{S}_u)$. From the previous section, we discussed how to model the distribution of $Dst(\widetilde{S}_u, \widetilde{S}_v)$ as a normal distribution with corresponding mean and variance. Note that the cdf of a normal distribution can be expressed in terms of the well-known *error function*. Given the mean μ and the deviation σ of a random variable X with normal distribution, its cdf is defined as follows:

$$\begin{aligned} & Pr(X \leq x) \\ = & \Phi_{\mu,\sigma^2}(x) \\ = & \frac{1}{2}(1 + erf(\frac{x - \mu}{\sigma\sqrt{2}})), \end{aligned} \quad (14)$$

where $erf()$ is the error function. Note that the value of this error function and its inverse function can be obtained by looking up from an existing statistics table.

To solve the similarity query problem defined in Eq. (4), we illustrate our idea in Fig. 3. Given a probability threshold τ, and the cdf of the normal distribution, we first compute a corresponding value *r-limit* which satisfied

$$Pr(Dst(\widetilde{S}_{ref}, \widetilde{S}_u)_{norm} \leq \textit{r-limit}) = \tau. \quad (15)$$

DEFINITION 3. *r-limit: Using the standard normal distribution function $N(0,1)$ and its cdf, given a probability threshold τ, we can obtain a corresponding value r-limit:*

$$\textit{r-limit} = \sqrt{2} erf^{-1}(2\tau - 1), \quad (16)$$

where $erf^{-1}(2\tau - 1)$ can be obtained by looking up a statistics table.

Table 1: Main symbols used in this paper.

Symbol	Description
$\widetilde{S}_u$	uncertain time series with identity u
$\widetilde{S}_u[j]$	the value of $\widetilde{S}_u$ at timestamp j
μ_{uj}	the expected value of the uncertainty at timestamp j of $\widetilde{S}_u$
σ_{uj}	the deviation of the uncertainty at timestamp j of $\widetilde{S}_u$
D_j	= $\widetilde{S}_u[j] - \widetilde{S}_v[j]$
D_j^2	= $(\widetilde{S}_u[j] - \widetilde{S}_v[j])^2$
$Dst(\widetilde{S}_u, \widetilde{S}_v)$	= $\sum_j (\widetilde{S}_u[j] - \widetilde{S}_v[j])^2$
$E(Dst(\widetilde{S}_u, \widetilde{S}_v))$	expected value of the uncertain distance between $\widetilde{S}_u$and$\widetilde{S}_v$
$Var(Dst(\widetilde{S}_u, \widetilde{S}_v))$	variance of the uncertain distance between $\widetilde{S}_u$and$\widetilde{S}_v$

The *r-limit* value defined in Definition 3 gives us a limit that, the probability of a normalized distance bound which is smaller than this value would definitely not be higher than τ.

Given a reference series $\widetilde{S}_{ref}$, assume that the expected value and the variance of the uncertain distance of a candidate series $\widetilde{S}_u$ to $\widetilde{S}_{ref}$ are obtained. Then, we can normalize the value of the given distance bound r^2 using the current obtained expectation and the variance of the uncertain distance.

DEFINITION 4. *Given the user defined distance bound r, with the expectation and variance of the uncertain distance between the reference series $\widetilde{S}_{ref}$ and a candidate series $\widetilde{S}_u$, we define the normalized distance bound as r_{norm}, and it is*

$$r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u) = \frac{r^2 - E(Dst(\widetilde{S}_{ref}, \widetilde{S}_u))}{\sqrt{Var(\widetilde{S}_{ref}, \widetilde{S}_u))}}. \quad (17)$$

If a candidate series $\widetilde{S}_u$ meet the following inequality

$$r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u) \geq \textit{r-limit}, \quad (18)$$

then

$$Pr(Dst(\widetilde{S}_{ref}, \widetilde{S}_u)_{norm} \leq r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)) \geq \tau. \quad (19)$$

, which means Eq. (4) is satisfied. As a result, $\widetilde{S}_u$ will be included in the answer set. On the contrary, if any candidate series cannot meet the above inequality, it will be pruned away.

3.3 Progressively Pruning

In real applications, the deviation at each time point usually remains the same for the same time series. For example, in the sensor network, data series obtained from the same sensor will have the same uncertain deviation. It is different sensors in different places that may have different deviations of uncertainties. Hence, we focus on the model that the uncertainty deviation is only time-series dependent hereafter. Explicitly, for a specific uncertain time series $\widetilde{S}_u$, the deviations of uncertainty at different time points are:

$$\sigma_{uj} = \sigma_u, \forall j.$$

In accordance with the model that the uncertain deviation is time-series dependent, the expected value and the variance of the uncertain distance between two streams now are:

$$\begin{aligned} & E(Dst(\widetilde{S}_u, \widetilde{S}_v)|_{t_s}^{t_e}) \\ = \; & (\sigma_u^2 + \sigma_v^2) \cdot (t_e - t_s + 1) + \sum_{j=t_s}^{t_e} (\mu_{uj} - \mu_{vj})^2 \end{aligned} \quad (20)$$

and

$$\begin{aligned} & Var(Dst(\widetilde{S}_u, \widetilde{S}_v)|_{t_s}^{t_e}) \\ = \; & 4(\sigma_u^2 + \sigma_v^2) \cdot \sum_{j=t_s}^{t_e} (\mu_{uj} - \mu_{vj})^2. \end{aligned} \quad (21)$$

Under this assumption, we do not need to compute the final expected value and the variance to decide if a candidate stream is qualified for the given threshold τ. In other words, if we can decide whether to prune a candidate or not with only part of random variables D_j's, it may save a lot of processing time.

Without loss of generality, assume we update the expected value and the variance in Eq. (20) and (21) starting from the very beginning of the given time range t_s. At certain time $t_k \in [t_s, t_e]$, the partial values of Eq. (20) and (21) are:

$$\begin{aligned} & E(Dst(\widetilde{S}_u, \widetilde{S}_v)|_{t_s}^{t_k}) \\ = \; & E(Dst(\widetilde{S}_u, \widetilde{S}_v)|_{t_s}^{t_k-1}) + (\mu_{ut_k} - \mu_{vt_k})^2 \end{aligned} \quad (22)$$

and

$$\begin{aligned} & Var(Dst(\widetilde{S}_u, \widetilde{S}_v)|_{t_s}^{t_k}) \\ = \; & Var(Dst(\widetilde{S}_u, \widetilde{S}_v)|_{t_s}^{t_k-1}) \\ & +4(\sigma_u^2 + \sigma_v^2)(\mu_{ut_k} - \mu_{vt_k})^2 \end{aligned} \quad (23)$$

Accordingly, at time t_k, having the current expected value and the variation from Eq. (22) and Eq. (23), we can compute the current normalized distance bound which is defined in Definition 4 as:

$$r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_k} = \frac{r^2 - E(Dst(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_k})}{\sqrt{Var(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_k})}}. \quad (24)$$

It is clear that both of Eq. (22) and Eq. (23) are non-decreasing as t_k approaches t_e. Furthermore, the value in Eq. (24) will approach the value in Eq. (17). If $r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_k}$ can be monotonically non-increasing as t_k approaches t_e, once

$$r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_k} < r\text{-limit}, \tag{25}$$

as illustrated in Fig. 3, we can stop computing the expected value and the variance for a candidate stream $\widetilde{S}_u$ and prune it away.

Therefore, we want to examine in what condition will Eq. (24) be non-increasing. To do this, we check the difference of Eq. (24) between two consecutive timestamps:

$$\begin{aligned} & r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_k-1} - r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_k} \\ = & \frac{r^2 - E(Dst(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_k-1})}{\sqrt{Var(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_k-1})}} \\ & - \frac{r^2 - E(Dst(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_k})}{\sqrt{Var(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_k})}}. \end{aligned} \tag{26}$$

We will prove in the Appendix that when we update to certain timestamp t_k, if

$$+\sqrt{\sum_{j=t_s}^{t_k-1}(\mu_{refj} - \mu_{uj})^2 \cdot \sum_{j=t_s}^{t_k}(\mu_{refj} - \mu_{uj})^2} \geq r^2 - (\sigma_{ref}^2 + \sigma_u^2) \cdot (t_e - t_s + 1), \tag{27}$$

the difference in Eq. (26) will always be positive when updating to the following timestamps, which means that we can guarantee r_{norm} is non-increasing. Hence, we can safely prune the candidate accordingly.

3.4 The PROUD Algorithm

We summarize the PROUD algorithm in Fig. 4. Given a reference series $\widetilde{S}_{ref}$, a time range $T = [t_s, t_e]$, a distance bound r, and a probability threshold τ, the algorithm will output the desired series that have a probability no smaller than τ where their distances to the reference one are not bigger than r. First, all the sub-series within the time range T are extracted. Without loss of generality, starting from the random variables at time t_s, we incrementally update the expected value and the variance of the uncertain distance between a candidate and the reference series according to Eq. (22) and Eq. (23). We check if the condition in Eq. (27) is satisfied to guarantee that $r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_k}$ is non-increasing. Once this condition is met, we can prune the candidate according to Eq. (25). Note if the condition in Eq. (27) has never been met (only if the given r is really large, which rarely happens), the prune is only performed after the entire data are computed.

4. APPLYING PROUD TO WAVELET SYNOPSES

More and more emerging applications are required to handle a large amount of data in the form of rapidly arriving streams under limited resources. Due to memory constraints, instead of storing the whole data stream, usually only the sketches of the stream are retained. In this section, we will discuss how our PROUD method can be applied when only summarization of streams is available.

Algorithm: PROUD
Input: $\widetilde{S}_{ref}$, $T = [t_s, t_e]$, r, τ
Output: The series which have distance to $\widetilde{S}_{ref}$ not bigger than r with probability not lower than τ

```
1.  Extract relevant random variables of S̃_ref in [t_s,t_e].
2.  for(j = t_s; j ≤ t_e; j++){
3.      t_k=j;
4.      Update E(Dst(S̃_ref, S̃_u)|_{t_s}^{t_k}) and
        Var(Dst(S̃_ref, S̃_u)|_{t_s}^{t_k}) for each candidate S̃_u
5.      if(Eq. (27) is met){
6.          if(Eq. (25) is met){
7.              prune S̃_u away
8.          }
9.      }
10.     else if(j equals to t_e){
11.         if(Eq. (25) is met){
12.             prune S̃_u away
13.         }
14.     }
15. }
```

Figure 4: Algorithm of PROUD.

When the uncertain series we formerly considered become endless streams, we need a way to properly summarize the mean and variance of each coming random variable. Follow the assumption in Sec. 3.3, to deal with similarity queries with uncertainty, we need to compute the expected value and the variance of uncertain distances between streams according to Eq. (20) and Eq. (21). Therefore, we need to know how to compute the above two values with stream synopses.

There are many summarization methods for data streams such as discrete Fourier transform, discrete wavelet transform, singular value decomposition, piecewise linear approximation and so on. Here we mainly focus on how to use PROUD under the Haar wavelet decomposition. One reason is that wavelet transform plays an important role in time series analysis [24]. In addition, it has the multi-resolution property in decomposing the original data. Last, but not least, the Haar wavelet decomposition is simple and can be maintained online. We leave the application of PROUD to other kinds of summarization methods as future work.

4.1 Wavelet Summarization for Uncertain Data Streams

In this section, we discuss how we use the Haar wavelet decomposition to summarize an uncertain data stream. By the assumption that the variance is the same at different j, for a stream $\widetilde{S}_u$, we need to summarize its mean μ_{uj} at different j's and keep one σ_u^2. In other words, to summarize an uncertain stream means to summarize a stream of mean values.

The Haar wavelet decomposition is achieved by averaging two adjacent data values of a sequence of data at different time resolutions. An example is given in Table 2. The final wavelet coefficients are $\{5.875, -0.625, -0.25, 0.5, -1, 1.5, 1, -1\}$.

Table 2: The Haar wavelet decomposition.

	averages	wavelet coefficients
raw data	$\{4, 6, 7, 4, 8, 6, 5, 7\}$	-
high resolution	$\{5, 5.5, 7, 6\}$	$\{-1, 1.5, 1, -1\}$
mid resolution	$\{5.25, 6.5\}$	$\{-0.25, 0.5\}$
low resolution	$\{5.875\}$	$\{-0.625\}$

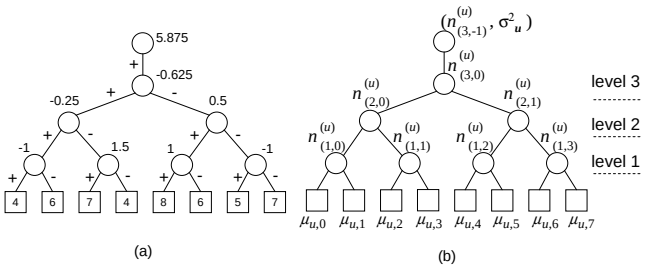

Figure 5: (a)The error tree for Table 2. (b)The notation of an error tree proposed in [16].

To better illustrate the Haar wavelet decomposition, a widely used data structure called *error tree* is proposed in [21]. The error tree for the decomposition in Table 2 is shown in Fig. 5(a). This tree is composed of wavelet coefficients as nodes and signs as edges.

We will similarly use an error tree to illustrate our idea. We slightly modified the node labeling method proposed in [16], which is shown in Fig. 5(b). The leaf nodes represent the sequence of means of an uncertain stream $\widetilde{S}_u$. Each non-leaf node is labeled with an identifier with two attributes as subscripts: *level* and *placement*. A node with a label of $n_{(l,p)}^{(u)}$ means that it is in the p-th placement of level l in the error tree corresponding to stream S_u. This notation can be efficiently maintained when data keep steaming in. Moreover, when not all wavelet coefficients are retained, we can easily find the relative positions of the retained coefficients in the error tree via the node labels.

For an uncertain data stream $\widetilde{S}_u$, the mean values μ_{uj} are used to build this tree. Under the assumption that the variance of uncertainty at each timestamp is the same for a specific stream, we keep the variance for each stream at the root node of the highest level as shown in Fig. 5(b).

Generally, not all the wavelet coefficients in an error tree are retained because memory space is limited. To meet different error requirements between the raw data and the retained coefficients, many online approaches to selecting wavelet synopses have been proposed [11, 13, 14, 18]. Fortunately, our proposed method is dependent only on the retained coefficients, and is independent of how the coefficients are retained. Therefore, we do not further discuss the relationship between our method and any specific on-line approaches to retaining wavelet synopses.

Given the retained coefficients of a stream, we can efficiently extract the relevant coefficients within any time range $[t_s, t_e]$. As suggested in [16], the extraction method is outlined in Fig. 6, where the black nodes represent retained coefficients, while the white ones are those being discarded. Assume the given range is $[t_0, t_{11}]$, which contains the shaded triangular area, we can decompose it into two complete error subtrees where one covers $[t_0, t_7]$ and the other $[t_8, t_{11}]$. For each complete error subtree, the new average node will be computed by traversing from the original root node $n_{(4,-1)}^{(u)}$ to the root of the subtree. For example, in Fig. 6, the new average node $a_{(3,0)}^{(u)}$ equals to $n_{(4,-1)}^{(u)} + n_{(4,0)}^{(u)}$. Finally, the information of the variance σ_u^2 of this stream is the same kept at the root of the complete tree with highest level.

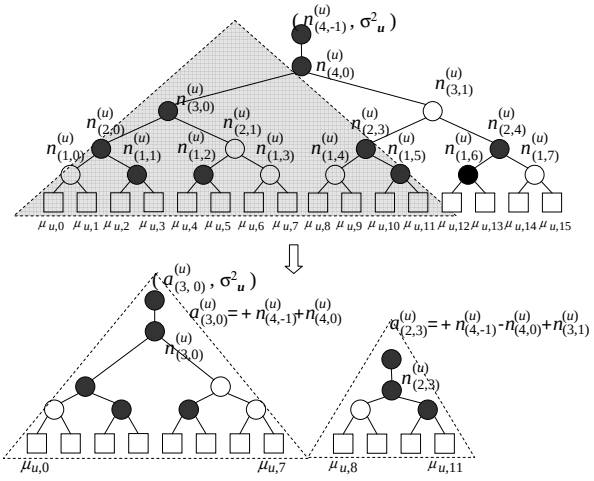

Figure 6: Coefficients extraction.

4.2 Statistic Computation Using Wavelet Coefficients

In this section, we described how to compute the expectation and the variance value of the uncertain distance between two uncertain streams when only partial coefficients are retained. According to Eq. (20) and Eq. (21), we need to compute

$$\sum_{j=t_s}^{t_e} (\mu_{uj} - \mu_{vj})^2 \text{ and } (\sigma_u^2 + \sigma_v^2)$$

to answer the probabilistic similarity queries. Since we do not transform the variance of each stream, $(\sigma_u^2 + \sigma_v^2)$ can be obtained directly. Now, the mean values are summarized as coefficients, and some of them are missing. Under this case, we show how to compute the distance directly from the retained coefficients.

Recently, the authors of [33] provided a way of computing the distance between two streams directly using their retained wavelet coefficients in a level-wise fashion. Given a specified time range $T = [t_s, t_e]$ and the retained wavelet coefficients, the distance between two streams S_u and S_v is:

$$\begin{aligned} & Dst(S_u, S_v)|_{t_s}^{t_e} = Dst(S_u, S_v)|_1^L \\ = & \sum_p [n_{(l,p)}^{(u)} - n_{(l,p)}^{(v)}]^2 \times 2^L + ... + \sum_p [n_{(l,p)}^{(u)} - n_{(l,p)}^{(v)}]^2 \times 2^1 \\ = & \sum_{l=1}^{L} \sum_p [n_{(l,p)}^{(u)} - n_{(l,p)}^{(v)}]^2 \times 2^l, \end{aligned} \tag{28}$$

where $L = \lfloor \lg_2(t_e - t_s + 1) \rfloor$ is the highest level number of the extract subtrees, $n_{(l,p)}^{(u)}$ and $n_{(l,p)}^{(v)}$ are retained coefficients inside the $[t_s, t_e]$ range of S_u and S_v.

Back to our case, according to Eq. (28), the expected value and the variance of uncertain distance shown in Eq. (20) and Eq. (21) can be computed directly using the wavelet coefficients respectively as follows:

$$\begin{aligned} & E(Dst(\widetilde{S}_u, \widetilde{S}_v)) \\ = & \sum_{l=1}^{L} \sum_{p} [n_{(l,p)}^{(u)} - n_{(l,p)}^{(v)}]^2 \times 2^l \\ & + (\sigma_u^2 + \sigma_v^2) \cdot (t_e - t_s + 1), \end{aligned} \tag{29}$$

and

$$\begin{aligned} & Var(Dst(\widetilde{S}_u, \widetilde{S}_v)) \\ = & \ 4(\sigma_u^2 + \sigma_v^2) \cdot \sum_{l=1}^{L} \sum_{p} [n_{(l,p)}^{(u)} - n_{(l,p)}^{(v)}]^2 \times 2^l, \end{aligned} \tag{30}$$

where $L = \lfloor \lg_2(t_e - t_s + 1) \rfloor$ is the highest level number of the extract subtrees, $n_{(l,p)}^{(u)}$ and $n_{(l,p)}^{(v)}$ are retained coefficients inside the $[t_s, t_e]$ range transformed from μ_{uj}'s and μ_{vj}'s.

With the above two equations, we can compute the statistics of the uncertain distance between two uncertain streams directly using the retained wavelet coefficients.

4.3 Pruning Strategy

By means of level-wise computation, and computing the distance from highest level to the lowest, we can gradually get a clearer view of the expected value and the variance of the uncertain distance. Because usually we can first get the rough view at higher levels with few coefficients and then details at lower levels. This is how we can leverage the multi-resolution property of the wavelet decomposition. As a result, starting from highest level L, when we update Eq. (29) between the reference $\widetilde{S}_{ref}$ and some stream $\widetilde{S}_u$ to certain level $\rho \in [1, L]$, it is:

$$\begin{aligned} & E(Dst(\widetilde{S}_{ref}, \widetilde{S}_u)|_\rho^L) \\ = & \sum_{l=\rho+1}^{L} \sum_{p} [n_{(l,p)}^{(ref)} - n_{(l,p)}^{(u)}]^2 \times 2^l \\ & + \sum_{p} [n_{(\rho,p)}^{(ref)} - n_{(\rho,p)}^{(u)}]^2 \times 2^\rho \\ & + (\sigma_{ref}^2 + \sigma_u^2) \cdot (t_e - t_s + 1), \end{aligned} \tag{31}$$

Similarly, the current variance is:

$$\begin{aligned} & Var(Dst(\widetilde{S}_{ref}, \widetilde{S}_u)|_\rho^L) \\ = & \ 4(\sigma_u^2 + \sigma_u^2) \cdot (\sum_{l=\rho+1}^{L} \sum_{p} [n_{(l,p)}^{(ref)} - n_{(l,p)}^{(u)}]^2 \times 2^l \\ & + \sum_{p} [n_{(\rho,p)}^{(ref)} - n_{(\rho,p)}^{(u)}]^2 \times 2^\rho). \end{aligned} \tag{32}$$

After we update the coefficients at level ρ, we can compute the normalized distance bound similar in Eq. (24) as follows:

$$r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_\rho^L = \frac{r^2 - E(Dst(\widetilde{S}_{ref}, \widetilde{S}_u)|_\rho^L)}{\sqrt{Var(\widetilde{S}_{ref}, \widetilde{S}_u)|_\rho^L)}}. \tag{33}$$

If $r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_\rho^L$ can be monotonically non-increasing as ρ approaches to 1, once

$$r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_\rho^L < r\text{-limit}, \tag{34}$$

as illustrated in Fig. 3, we can stop computing the expected value and the variance for a candidate stream $\widetilde{S}_u$ and prune it away.

We will give the proof in the Appendix that at certain level ρ, once

$$\begin{aligned} & r^2 - (\sigma_{ref}^2 + \sigma_u^2) \cdot (t_e - t_s + 1) \\ & + \sqrt{\sum_{l=\rho+1}^{L} \sum_{p} [n_{(l,p)}^{(ref)} - n_{(l,p)}^{(u)}]^2 \times 2^l} \\ & \cdot \sqrt{\sum_{l=\rho}^{L} \sum_{p} [n_{(l,p)}^{(ref)} - n_{(l,p)}^{(u)}]^2 \times 2^l} \\ & \geq 0, \end{aligned} \tag{35}$$

we can guarantee that $r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_\rho^L$ will be monotonically non-increasing as ρ approaches to the following lower levels. Therefore we can do the pruning safely as described in Eq. (34)

5. PERFORMANCE STUDY

We conducted a series of experiments with both real and synthetic data to evaluate PROUD. We compared PROUD with a deterministic approach, referred to as *Det*, where the similarity queries are processed on uncertain data as if they were data with certainty. In addition, the pruning strategy for Det is similar to that used in PROUD. When updating the distance between a candidate stream and the reference one, once it exceeds the given distance bound r, we prune that candidate away. Both approaches were implemented in Visual C++ and the experiments were run on a PC with 2.8GHz CPU and 2GB RAM.

We processed the range-specified probabilistic similarity queries on both real and synthetic data. To generate an uncertain stream $\widetilde{S}_u$, we did the following. First, we had the true data stream S_u. We computed the standard deviation of the entire stream S_u, which is σ_{S_u}. Then, we picked a set of value: [0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2] of σ_{S_u} as the deviation of the uncertainty σ_u. Given a σ_u, for each timestamp j, we randomly drew a number from a normal distribution or uniform distribution with mean equals to $S_u[j]$ and the deviation equals to σ_u to be the uncertain value $\widetilde{S}_u$.

To measure the performance of PROUD and Det, we compared these two methods in terms of quality of query results and computation time cost. In quality, we compared the *false alarm ratio* and the *miss ratio*[1] of both two methods. The ground truth is based on the query result with the same distance bound without uncertainty

[1]The false alarm ratio and miss ratio were computed as the fractions of the number of false alarms and the number of misses over the size of the ground truth, respectively.

$T = 30$				
Deviation Ratio	PROUD			Det
	$\tau = 0.01$	$\tau = 0.5$	$\tau = 0.9$	
0.02	0.0609	0.0000	0.0000	0.0066
0.10	0.2241	0.0000	0.0000	0.0457
0.50	0.0306	0.0000	0.0000	0.0023
2.00	0.0000	0.0000	0.0000	0.0000
$T = 100$				
0.02	0.0262	0.0000	0.0000	0.0045
0.10	0.0821	0.0000	0.0000	0.0026
0.50	0.0037	0.0000	0.0000	0.0000
2.00	0.0000	0.0000	0.0000	0.0000
$T = 300$				
0.02	0.0255	0.0000	0.0000	0.0100
0.10	0.0175	0.0000	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0000
2.00	0.0000	0.0000	0.0000	0.0000

Table 3: False alarm ratio of PROUD and Det (real data.)

on the true data S_u. In computation time cost, we measured the CPU time of processing a query on average.

The following experiments were all conducted on wavelet synopses of the uncertain data. It is noted that PROUD is independent of the way wavelet coefficients are chosen. Without loss of generality, here the wavelet coefficients were retained using the method proposed in [13], which retains the B largest coefficients in terms of absolute normalized coefficient values. We randomly picked a few different streams from our dataset as the reference stream and performed queries. Then the averaged results are reported.

5.1 Experiments with Real Data

The real data we used here were the daily average temperature data of 300 cities around the world, which were obtained from the temperature data archive of the University of Dayton[2]. The data from each city was regarded as a stream, each of which has $3,416$ data points.

Table 3 and Fig. 7 show the quality of query results for the real data. We varied the time range (T) of queries and the magnitude of the uncertain deviation to have a series of subplots. The uncertain deviation were varied from 0.02 to 2 of the deviation of each original stream. Furthermore, for PROUD, we show the results under three different τ's: [0.01, 0.5, 0.9].

First, let us look at the false alarm ratios listed in Table 3. Both Det and PROUD, with $\tau = 0.5$ and $\tau = 0.9$, barely incur any false alarms. This can be explained from Eq. (11) or Eq. (20). The summation of the variances of two uncertain variables at each timestamp are also included. Since this value is always positive, it makes the expected value likely to be larger than the true distance. However, for PROUD, there are more false alarms when $\tau = 0.01$ than when $\tau = 0.5$ or $\tau = 0.9$.

Fig. 7 shows the impact of uncertainty level on the miss ratio, under three different time ranges T's and threshold τ's. The x-axis is the deviation ratio, and the y-axis is the miss ratio. Generally, the miss ratio increases as the uncertain deviation ratio increases. The miss ratio of Det is always around that of PROUD when $\tau = 0.5$. When

[2]http://www.engr.udayton.edu/weather/

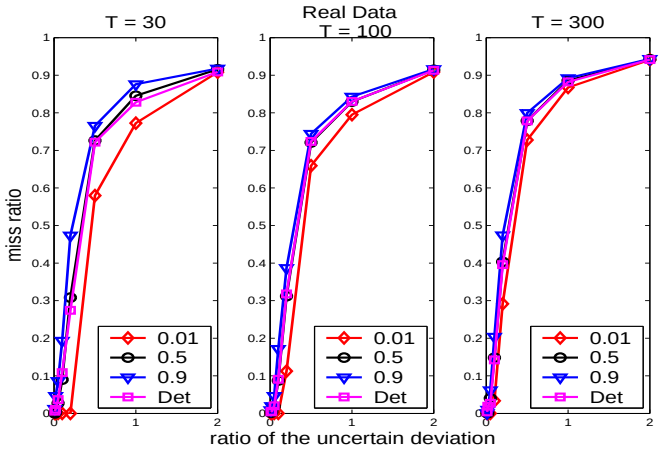

Figure 7: The miss ratio of PROUD and Det (real data.)

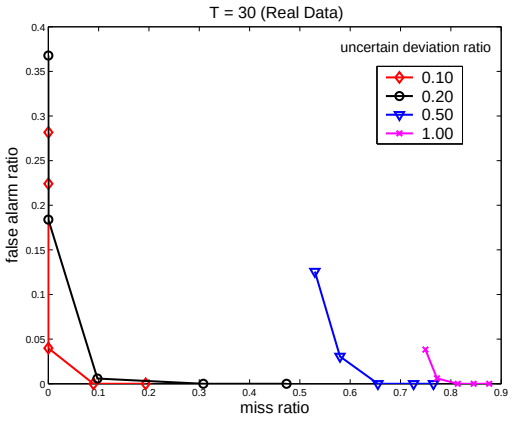

Figure 8: The trade-offs between false alarms and misses offered by PROUD (real data.)

$\tau = 0.01$, PROUD has lower miss ratios than Det. On the other hand, when $\tau = 0.9$, PROUD has higher miss ratios than Det. Contrasting this observation with the false alarm ratios from Table 3, it clearly shows that PROUD offers a flexible trade-off between miss ratios and false alarms by controlling τ. With a smaller τ, PROUD offers a solution with lower miss ratios and higher false alarms. In contrast, with a larger τ, it provides a solution with higher miss ratios and lower false alarms.

To clearly illustrate the trade-offs between false alarms and misses offered by PROUD, we further plot the false alarm ratio versus miss ratio under different τ's and uncertain deviation ratios in Fig. 8. The τ's are [0.001, 0.01, 0.1, 0.5, 0.9] and the deviation ratio ranges from 0.1 to 1. Each line in this figure represents different τ's under the same specific uncertain deviation ratio. Although the false alarm ratios are relatively small compared with the miss ratios, we clearly observe the trade-off between them. For a smaller τ, the false alarm ratio is higher and the miss ratio is lower. In contrast, for a larger τ, the reverse is true. As the uncertain deviation ratio gets larger, the curves move toward the higher miss ratio region.

As the time range becomes bigger, i.e., $T = 100$ or $T = 300$, the miss ratios of PROUD with different τ's become closer to one another. For example, when deviation ratio is 0.2, the miss ratio of PROUD with $\tau = 0.01$ is about 1/2 of that with $\tau = 0.9$ when

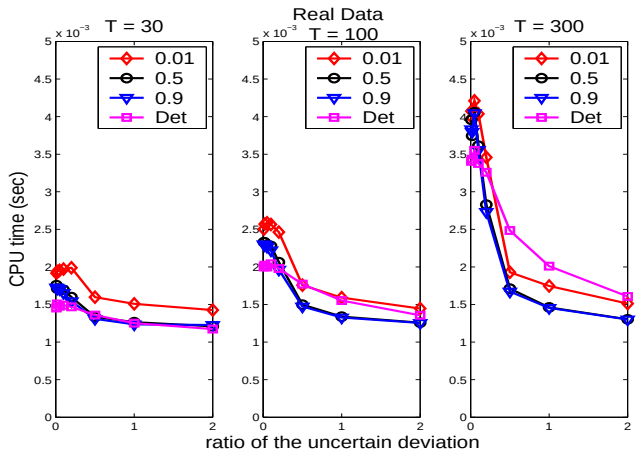

Figure 9: The computation time cost of PROUD and Det (real data.)

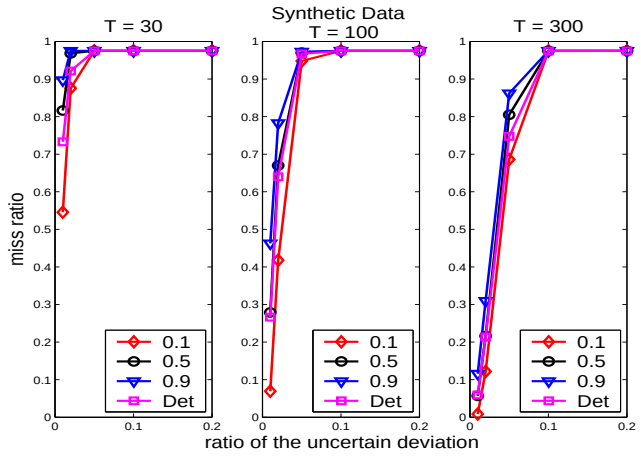

Figure 10: The miss ratio of PROUD and Det (synthetic data.)

$T = 100$, and about 3/4 when $T = 300$. This is because when the time range is large, more random variables are involved, and the *law of large number* dominates. Almost all uncertain distance values fall very close to the expected value. Therefore, we cannot tell much difference under different τ's. The size of uncertainty decides the quality.

Fig. 9 shows the corresponding computation time cost of Fig. 7. The computation cost decreases when the deviation of the uncertainty increases for both Det and PROUD. This is because when the uncertainty is really high, the expected value of the uncertain distance becomes very high as well. As a result, unqualified or even qualified candidates are easily pruned away. After pruning, only few candidate streams are left. In addition, with a smaller T, the computation time for Det is smaller than that for PROUD. However, with a larger T, and the uncertain deviation ratio is higher, PROUD spent less time than Det. This shows that, under similar query result quality, the prune efficiency of PROUD is even higher.

In summary, PROUD and Det have similar computation costs. However, PROUD offers a flexible trade-off between miss ratios and false alarms by controlling τ's. Det does not have such flexibility. This trade-off is important as in some applications false negatives are more costly, while in others, it is more critical to keep the false positives low.

5.2 Experiments with Synthetic Data

The synthetic data were generated by a random walk data model proposed in [35]. For a stream S_i, it was generated as follows:

$$S_i = 100 + \sum_{j=1}^{i} (u_j - 0.5),$$

where u_j was randomly picked from [0,1]. We generated 1,000 streams in total, where each stream has 20,000 data points. Here we consider the following ratios of the uncertain deviation [0.01, 0.02, 0.05, 0.1, 0.2].

The behavior of false alarm ratio in synthetic data is similar to that in real data. Therefore, we omit it here. Fig. 10 shows the miss ratios for both Det and PROUD under $T = 30$ to $T = 300$. Basically, under all T values, the miss ratio increases as the ratio of the uncertain deviation increases. At $T = 30$, the miss ratio is very high even when the deviation ratio is only 0.01. This is because, σ_{S_u}, the deviation of the entire 20,000-point-long random walk data stream, is quite high compared to the data of small range of 30. As T gets larger, the miss ratio at a small deviation ratio reduces. This phenomenon is not observed in the experiments with real dataset. It is because that the deviation of a 3,416-long temperature stream of a city is relatively low.

For the computation time cost, the results are also similar as those for real data. The corresponding charts are shown in Fig. 11. The computation time costs for both PROUD and Det are similar. As T gets larger, the computation time cost increases as well. Also, when the deviation ratio is higher, a larger expected value of the uncertain distance results in earlier pruning, hence the computation cost is smaller. When T gets larger, the pruning efficiency of PROUD is better than Det.

6. CONCLUSION

In this paper, we presented PROUD - a probabilistic approach to processing similarity queries over multiple uncertain data streams. We demonstrated how various probabilistic theories can help us deal with similarity queries over uncertain data streams. We showed how we can progressively prune candidates. Furthermore, we showed how to apply PROUD using only wavelet synopses instead of raw data. We conducted extensive experiments with both real and synthetic data. The results show that, compared with Det, PROUD provides a flexible trade-off between false alarms and miss ratios by controlling a threshold, while maintaining a similar computation cost. In contrast, Det does not have such flexibility.

As future work, we will extend our work to other kinds of queries on uncertain streams, like probabilistic nearest neighbor queries for example. We will also explore probabilistic similarity queries over uncertain streams using other similarity measurements than the Euclidean distance.

7. REFERENCES

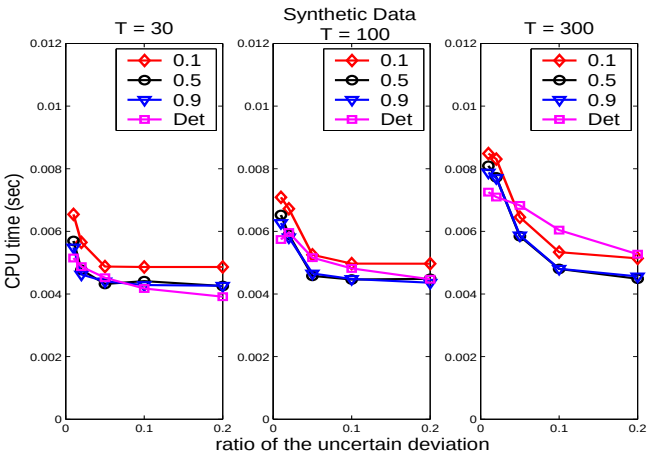

Figure 11: The computation time cost of PROUD and Det (synthetic data.)

[1] C. C. Aggarwal. On unifying privacy and uncertain data models. In *Proc. of IEEE ICDE*, 2008.

[2] C. C. Aggarwal and P. S. Yu. On high dimensional indexing of uncertain data. In *Proc. of IEEE ICDE*, 2008.

[3] L. Antova, T. Jansen, C. Koch, and D. Olteanu. Fast and simple relational processing of uncertain data. In *Proc. of IEEE ICDE*, 2008.

[4] L. Antova, C. Koch, and D. Olteanu. 10106 worlds and beyond: Efficient representation and processing of incomplete information. In *Proc. of IEEE ICDE*, 2007.

[5] O. Benjelloun, A. Sarma, A. Halevy, and J.Widom. Uldbs: Databases with uncertainty and lineage. In *Proc. of VLDB*, 2006.

[6] C. Bohm, A. Pryakhin, and M. Schubert. The gauss-tree: Efficient object identification in databases of probabilistic feature vectors. In *Proc. of IEEE ICDE*, 2006.

[7] R. Cheng, J. Chen, M. Mokbel, and C.-Y. Chow. Probabilistic verifiers: Evaluating constrained nearest-neighbor queries over uncertain data. In *Proc. of IEEE ICDE*, 2008.

[8] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In *Proc. of ACM SIGMOD*, 2003.

[9] R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J. Vitter. Efficient indexing methods for probabilistic threshold queries over uncertain data. In *Proc. of VLDB*, 2004.

[10] G. Cormode and M. Garofalakis. Sketching probabilistic data streams. In *Proc. of ACM SIGMOD*, pages 281–292, 2007.

[11] G. Cormode, M. Garofalakis, and D. Sacharidis. Fast approximate wavelet tracking on streams. In *Proc. of EDBT*, pages 4–22, 2006.

[12] N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In *Proc. of VLDB*, 2004.

[13] A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss. One-pass wavelet decompositions of data streams. *IEEE TKDE*, 15(3):541–554, 2003.

[14] S. Guha and B. Harb. Wavelet synopsis for data streams: minimizing non-euclidean error. In *Proc. of ACM SIGKDD*, pages 88–97, 2005.

[15] M. Hua, J. Pei, X. Lin, and W. Zhang. Efficiently answering probabilistic threshold top-k queries on uncertain data. In *Proc. of IEEE ICDE*, 2008.

[16] H.-P. Hung and M.-S. Chen. Efficient range-constrained similarity search on wavelet synopses over multiple streams. In *Proc. of ACM CIKM*, pages 327–336, 2006.

[17] T. S. Jayram, A. McGregor, S. Muthukrishnan, and E. Vee. Estimating statistical aggregates on probabilistic data streams. In *Proc. of ACM PODS*, pages 243–252, 2007.

[18] P. Karras and N. Mamoulis. One-pass wavelet synopses for maximum-error metrics. In *Proc. of VLDB*, pages 421–432, 2005.

[19] X. Lian, L. Chen, and J. X. Yu. Pattern matching over cloaked time series. In *Proc. of IEEE ICDE*, 2008.

[20] V. Ljosa and A. Singh. Apla: Indexing arbitrary probability distributions. In *Proc. of IEEE ICDE*, 2007.

[21] Y. Matias, J. S. Vitter, and M. Wang. Wavelet-based histograms for selectivity estimation. In *Proc. of ACM SIGMOD*, pages 448–459, 1991.

[22] G. W. Oehlert. A note on the delta method. *American Statistician*, 46(1):27–29, 1992.

[23] J. Pei, B. Jiang, X. Lin, and Y. Yuan. Probabilistic skylines on uncertain data. In *Proc. of VLDB*, 2007.

[24] I. Popivanov and R. J. Miller. Similarity search over time-series data using wavelets. In *Proc. of IEEE ICDE*, 2002.

[25] C. Re, N. Dalvi, and D. Suciu. Efficient top-k query evaluation on probabilistic data. In *Proc. of IEEE ICDE*, 2007.

[26] A. D. Sarma, O. Benjelloun, A. Y. Halevy, and J. Widom. Working models for uncertain data. In *Proc. of IEEE ICDE*, 2006.

[27] A. D. Sarma, M. Theobald, and J. Widom. Exploiting lineage for confidence computation in uncertain and probabilistic databases. In *Proc. of IEEE ICDE*, 2008.

[28] P. Sen and A. Deshpande. Representing and querying correlated tuples in probabilistic databases. In *Proc. of IEEE ICDE*, 2007.

[29] S. Singh, C. Mayfield, S. Prabhakar, R. Shah, and S. Hambrusch. Indexing uncertain categorical data. In *Proc. of IEEE ICDE*, 2007.

[30] M. A. Soliman, I. F. Ilyas, and K. C. Chang. Top-k query processing in uncertain databases. In *Proc. of IEEE ICDE*, 2007.

[31] 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 *Proc. of VLDB*, 2005.

[32] E. W. Weisstein. Central limit theorem. From MathWorld - A Wolfram Web Resource.

[33] M.-Y. Yeh, K.-L. Wu, P. S. Yu, and M.-S. Chen. Leewave: Level-wise distribution of wavelet coefficients for processing knn queries over distributed streams. In *Proc. of VLDB*, 2008.

[34] K. Yi, F. Li, D. Srivastava, and G. Kollios. Efficient processing of top-k queries in uncertain databases. In *Proc. of IEEE ICDE*, 2008.

[35] Y. Zhu and D. Shasha. Statstream: statistical monitoring of thousands of data streams in real time. In *Proc. of VLDB*, 2002.

Appendix

Here, we provide the proof that $r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)$ is monotonically non-increasing during the updating process. First, we prove that it is non-increasing in raw uncertain series cases, and then we prove that it is also non-increasing in the case when we use the wavelet coefficients directly.

A1 Proof of non-increasing r_{norm} in raw uncertain series case

For ease of exposition, we restate Eq. (26) using some substitutes. According to Eq. (22) and Eq. (23), we can substitute the terms as follows. Let

$$\begin{aligned} \alpha &= \sum_{j=t_s}^{t_k-1} (\mu_{refj} - \mu_{uj})^2, \\ \beta &= (\mu_{reft_k} - \mu_{ut_k})^2, \end{aligned}$$

and

$$\Delta = (\sigma_{ref}^2 + \sigma_u^2) \cdot (t_e - t_s + 1).$$

With the above substitution, we have

$$\begin{aligned} & E(Dst(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_k}) \\ = & \sum_{j=t_s}^{t_k-1} (\mu_{refj} - \mu_{uj})^2 + (\mu_{reft_k} - \mu_{ut_k})^2 \\ = & \alpha + \beta, \end{aligned}$$

and

$$\begin{aligned} & Var(Dst(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_k}) \\ = & 4(\sigma_{ref}^2 + \sigma_u^2) \sum_{j=t_s}^{t_k-1} (\mu_{refj} - \mu_{uj})^2 \\ & +4(\sigma_{ref}^2 + \sigma_u^2)(\mu_{reft_k} - \mu_{ut_k})^2 \\ = & 4(\sigma_{ref}^2 + \sigma_u^2)(\alpha + \beta). \end{aligned}$$

Therefore, Eq. (26) is restated as follows:

$$\begin{aligned} & r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_k-1} - r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_{t_s}^{t_k} \\ = & \frac{r^2 - \alpha - \Delta}{\sqrt{4(\sigma_{ref}^2 + \sigma_u^2) \cdot \alpha}} - \frac{r^2 - \alpha - \beta - \Delta}{\sqrt{4(\sigma_{ref}^2 + \sigma_u^2) \cdot (\alpha + \beta)}} \\ = & \frac{(r^2 - \Delta + \sqrt{\alpha(\alpha+\beta)}) \cdot (\sqrt{\alpha+\beta} - \sqrt{\alpha})}{\sqrt{4(\sigma_{ref}^2 + \sigma_u^2) \cdot \alpha \cdot (\alpha + \beta)}}. \end{aligned} \quad (36)$$

It is clear that $r^2 - \Delta$ is a constant when t_s and t_e are given. Also, it is obvious that $(\sqrt{\alpha+\beta} - \sqrt{\alpha})$ is always non-negative. As the timestamp t_k approaches to the end of query time range t_e, the term $\sqrt{\alpha(\alpha+\beta)}$ is definitely growing or non-decreasing. Therefore, once

$$(r^2 - \Delta + \sqrt{\alpha(\alpha+\beta)}) \geq 0,$$

,which means that

$$r^2 - (\sigma_{ref}^2 + \sigma_u^2) \cdot (t_e - t_s + 1) + \sqrt{\sum_{j=t_s}^{t_k-1} (\mu_{refj} - \mu_{uj})^2 \cdot \sum_{j=t_s}^{t_k} (\mu_{refj} - \mu_{uj})^2} \geq 0,$$

the Eq. (36) will always be positive. This gives us the proof that $r_{norm}|_{t_s}^{t_k}$ is then non-increasing as t_k approaches to t_e.

A2 Proof of non-increasing r_{norm} in wavelet synopses case

To prove that Eq. (33) is non-increasing, we prove the following. When updating the statistics from level $\rho + 1$ to ρ, the difference of r_{norm} values at two levels is:

$$r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_{\rho+1}^{L} - r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_{\rho}^{L}. \quad (37)$$

For ease of exposition, we do the following substitutions which are similar to the previous section. Let

$$\begin{aligned} \alpha &= \sum_{l=\rho+1}^{L} \sum_{p} [n_{(l,p)}^{(ref)} - n_{(l,p)}^{(u)}]^2 \times 2^l, \\ \beta &= \sum_{p} [n_{(\rho,p)}^{(ref)} - n_{(\rho,p)}^{(u)}]^2 \times 2^\rho, \end{aligned}$$

and

$$\Delta = (\sigma_{ref}^2 + \sigma_u^2) \cdot (t_e - t_s + 1).$$

According to Eq. (31) and Eq. (32), Eq. (37) then becomes:

$$\begin{aligned} & r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_{\rho+1}^{L} - r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_{\rho}^{L} \\ = & \frac{r^2 - E(Dst(\widetilde{S}_{ref}, \widetilde{S}_u)|_{\rho}^{L})}{\sqrt{Var(\widetilde{S}_{ref}, \widetilde{S}_u)|_{\rho+1}^{L})}} - \frac{r^2 - E(Dst(\widetilde{S}_{ref}, \widetilde{S}_u)|_{\rho}^{L})}{\sqrt{Var(\widetilde{S}_{ref}, \widetilde{S}_u)|_{\rho}^{L})}} \\ = & \frac{(r^2 - \Delta + \sqrt{\alpha(\alpha+\beta)}) \cdot (\sqrt{\alpha+\beta} - \sqrt{\alpha})}{\sqrt{4(\sigma_{ref}^2 + \sigma_u^2) \cdot \alpha \cdot (\alpha + \beta)}}. \end{aligned}$$

It is clear that $r^2 - \Delta$ is a constant when t_s and t_e are given. Also, it is obvious that $(\sqrt{\alpha+\beta} - \sqrt{\alpha})$ is always non-negative. As the we progress from level $\rho + 1$ to level ρ, the term $\sqrt{\alpha(\alpha+\beta)}$ is definitely growing or non-decreasing. Therefore, once

$$(r^2 - \Delta + \sqrt{\alpha(\alpha+\beta)}) \geq 0,$$

,which means that

$$r^2 - (\sigma_{ref}^2 + \sigma_u^2) \cdot (t_e - t_s + 1) + \sqrt{\sum_{l=\rho+1}^{L} \sum_{p} [n_{(l,p)}^{(ref)} - n_{(l,p)}^{(u)}]^2 \times 2^l} \cdot \sqrt{\sum_{l=\rho}^{L} \sum_{p} [n_{(l,p)}^{(ref)} - n_{(l,p)}^{(u)}]^2 \times 2^l} \geq 0,$$

the value $r_{norm}(\widetilde{S}_{ref}, \widetilde{S}_u)|_{\rho}^{L}$ is non-increasing as ρ approaches to the lowest level.