CIT. Journal of Computing and Information Technology, Vol. 26, No. 3, September 2018, 191–207
doi: 10.20532/cit.2018.1004248

Spatial Index for Uncertain Time Series

Diwei Zheng, Li Yan and Yu Wang

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing,China

A search for patterns in uncertain time series is time-expensive in today's large databases using the currently available methods. To accelerate the search process for uncertain time series data, in this paper, we explore a spatial index structure, which uses uncertain information stored in minimum bounding rectangle and ameliorates the general prune/search process along the path from the root to leaves. To get a better performance, we normalize the uncertain time series using the weighted variance before the prune/hit process. Meanwhile, we add two goodness measures with respect to the variance to improve the robustness. The extensive experiments show that, compared with the primitive probabilistic similarity search algorithm, the prune/hit process of the spatial index can be more efficient and robust using the specific preprocess and variant index operations with just a little loss of accuracy.

ACM CCS (2012) Classification: Mathematics of computing → Probability and statistics → Statistical paradigms → Time series analysis
Information systems → Information retrieval → Retrieval models and ranking → Similarity measures

Keywords: time series, spatial index, uncertainty, varying distance threshold

1. Introduction

Time series widely exists in various application fields such as GIS [6], stock market [16], astronomy [33], [37], medical application [36], etc. With the development of modern technology and applications, the requirements of dealing with time series dramatically increases. There are several examples of processing time series data. In medical application [36], real-time health timestamp data are used to detect patients' health condition. In the stock market [16], time series data are used to predict the values of the index in upcoming days. In GIS [6], a recognition system is proposed for time series data through acoustic emission. More recently, time series are used for modeling coronal mass ejection in [33], [37], which are analyzed by clustering and visualization [34], [35].

With massive time series data available, an efficient process for searching a specific pattern from the database is clearly becoming more and more essential. A lot of effort has been devoted to working with time series and some essential issues have been investigated, such as probabilistic range queries [13], [19], similarity match for uncertain time series [2], [7], [9], [11], pattern detection for uncertain data [18], and so on. Among these issues, one of the common requirements is to efficiently find probabilistically approximate matches from a collection of data items for a given query item.

To speed up the match process of a time series, several variant indexes are developed to handle diverse data objects. With an index structure, the target metric such as similarity measurement can be calculated by using feature extraction or retrieval values. *R*-tree* [24], for example, is a kind of spatial index and it treats each timestamp as one dimension in a spatial space. The dimension of this spatial space is of equal length as the time series. Here it is assumed that all series in the database must be isometric and the prune/hit function can be

implicitly or explicitly represented by geometry metrics including distance, margin and area [24].

Uncertainty extensively happens in the real world and has been studied in [1]–[4], [7], [9], [11], [12], [14], [15], [17], [18]. Instead of storing a single value at each timestamp in the classical time series, each timestamp can be modeled as a range of possible bucket or a variable with noise that is linked with a probability density function (*pdf*). In contrast to the deterministic time series, similarity queries for the uncertain one is more uncertain due to the underlying noise of data objects. As a result, the returned answers are always probably approximately correct, with probability $1 - \delta$ indicating degrees in which they meet the query.

Traditional time series with a large size is facing an overhead in efficiency, let alone the uncertain time series. For the massive data set consisting of the uncertain time series, matching or searching is not simple, since we need to consider a huge number of noisy items taking along some probability information. Taking the probability information into consideration significantly increases the time cost in similarity metric calculation [3]. In addition, the existing classical time series models cannot properly cooperate with uncertain information. Hence, a lot of effort has been carried out for the uncertain time series in several indirect ways, such as transforming the original question into classical deterministic models [19], [31], modifying traditional measurements or proposing new measurements [2], [3], [9], [17] and optimization operations [15].

Uncertain data management has been studied in the context of databases and now it resurges anew with the development of modern technology and applications. Due to the efficient algorithm for the ideal data object without noise, it is indispensable to carefully reconstruct the structure to embed uncertain information.

We strive to develop faster-searching methods to search a database consisting of a plenty of time series. Although the spatial index (e.g., *R*-trees*) can be used to search approximation queries, this approach exploits two assumptions: the first one is that data sequences and query sequences all have the same length; the second one is that the sequences are all definite. The probabilistic approach to processing similarity queries over uncertain data streams, namely (*PROUD*) [2] and the novel distance measure *DUST* [3] are both time-expensive methods since the prune/hit process involves integral calculation over the pdf. The traditional spatial index methods simply ignore the noise behind the item and do not take advantage of variance in each timestamp at all. This causes a heavy accuracy loss in final results.

In this paper, we explore a spatial index structure in connection with the uncertainty entries. Based on the *PROUD*, we plug and exploit the variance in minimum bounding rectangle (*MBR*) which is a directory for speeding up search process in the spatial index and refine the general prune/search process along the path from the root to leaves. To keep a better approximation in metric measures defined in a Euclidean distance, we propose a new preprocess method with weighted variance for uncertain times series. At the same time, we improve the robustness of the index using the variance in each *MBR*. Our contributions in this paper are summarized as follows:

1. We accommodate uncertain information in the classical spatial index R*-tree and show that the key to the combination is the uncertain monotonic direction of the distance threshold.
2. We investigate how to use the variance of uncertainty information to make less visits to deeper nodes, which will evidently improve the index robustness.
3. We propose a heuristic method with the variance taken into consideration to prune the candidates of the time series in which each time stamp has different random variance.

The rest of this paper is organized as follows. In Section 2, we give a brief description of the related work for the uncertain time series. Section 3 presents the model and the algorithm *PROUD* proposed in [2] for the uncertain time series. We present how to combine uncertainties with a classical index to efficiently search and construct a variant spatial index in Section 4. The experiments are presented in Section 5. We finally conclude the paper in Section 6.

2. Related Work

2.1. Uncertain Time Series and Querying

There has been plenty of work on representing and querying uncertain data. However, only a few parts of them address querying and indexing uncertain time series data. So far, there are two kinds of the proposed models for uncertain time series. The first one views the timestamp as a bucket used to record the historical values and the second one, called a pdf-based model, regards each timestamp as a variable with a random error noise. On the basis of the set, the notion of uncertain time series was formalized and two novel and essential types of range queries over uncertain time series were proposed in [1]. However, the number of combination choices for the series is exponential and must be refined by the boundaries proposed in [1]. In [2], *PROUD*, which is based on the Central Limit Theorem, was presented in the pdf-based model and offered a flexible control through distance or probability thresholds defined by users. The experiments in [2] showed exactly a trade-off between false alarms and false drops controlled by the user-defined distance and probability. In [3], the notion of the measurement for the uncertain time series was generalized, in which, based on several properties, more probability statistical information (e.g., totally various *pdfs*) were accommodated. In each timestamp, the measurement named *DUST* was quantified by approximately comparing the probabilities using the inequation

$$P\left(DIST\left(X_1, Y\right) \le \varepsilon\right) > P\left(DIST\left(X_2, Y\right) \le \varepsilon\right).$$

In [9], [17], the relationship between two series was explored through the correlation statistics. And with a preprocess, the relationship was used to convert the correlation threshold into the distance threshold. Finally, they used the measurement on both pdf-based model and multiset-based model in experiments and showed a flexible result of the measurement.

2.2. Index Method for Classical Time Series

With a large number of time series data available, there have been several efforts to model and process time-sequenced data. Efficient queries contain two vital steps, as shown in Figure 1.

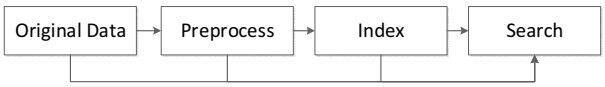


Figure 1. Flow chart.

1. Preprocess. In the real world, we are always stuck in a dilemma of balancing accuracy and efficiency. This question is a complex overhead when the length of the series is too long to efficiently calculate a metric. Thus, several methods (e.g., Wavelet decomposition [26] and Discrete Fourier transform (*DFT*)) were proposed to extract the features of series and reduce the original dimension to a new space, which has the least loss in retrieving values. In essence, the operation of dimension reduction is to map the points in a space with higher dimension into a lower dimensional space which is spanned by only a few new orthogonal vectors.
2. Index. After the preprocess in which every value of the timestamp is mapped into a new space, each time series can be correspondingly regarded as a point in this new space. It is clear that the spatial index can be constructed with those points. Based on some classical index structures such as *R*-tree*, *X-tree*, *S-tree* and so on, the target measurement can be directly calculated by coefficients in the index or implicitly obtained by retrieval values. Note that most of the existing work is focused on exact data without noise. The assumption with the noise absent is hardly adaptable to the physical environment which always contains uncertainties.

2.3. Index for Managing Uncertain Time Series

To rapidly deal with the large sequences with limited computation source, summarization methods for data streams have been consid-

ered. However, for uncertain time series, few proposed indexes can be used for uncertain statistics.

The first effort was regarding uncertain time series as cloaked time series based on a synopsis model in which each time stamp only knows the mean and deviation of the variance. The pattern match query was redesigned to work together with this model. Due to the dimensionality curse, a more efficient algorithm was proposed to construct the index with extra statistics such as mean and variance. Based on the framework proposed in [20], a more flexible measurement was offered in [2] by using the cumulative distribution function (*cdf*), which controls the false/true alarms or false/true drops. Aiming to efficiently prune unqualified series, a *Haar* decomposition index was used in [2]. The measurement can be directly calculated from the decomposed values in the index. However, it only focuses on the update operation for infinite stream data in a limited memory source. It can get more flexibility but without great efficiency improvement in the prune process.

Note that using the methods like feature extraction or decomposition based on different models or different assumptions, index structure should be explicitly or implicitly modified to adapt to the uncertainty information. In this paper, we propose a spatial index which can be generalized to expand the content of *MBR* and uses the uncertain parameter to update the structure.

It is clear that the preprocess such as dimension reduction, normalization, etc. can enhance the performance of index. However, as we can see, there is no great progress in the index and the classical index still keeps an original state even when we face a greatly different data object. So far, the study about the preprocess for uncertain time series receives a lot of attention while the optimization is ignored. In this paper, we present an optimized spatial index *R*-tree* for the uncertain time series, as well as a modified preprocess which cooperate with uncertainty information.

3. Preliminaries

In this section, we introduce the model used for uncertain time series as well as the algorithm *PROUD* based on the model in [2]. Although the multi-based model proposed in [1] can be also indexed abruptly, the exponential number of the combination is time-costly and the optimization method in [1] can only be taken at the level of the leaves. So, we use a continuous model for uncertain time series and develop a variant index based on this model.

3.1. Continuous Model

An uncertain time series $\hat{S}_u$ is a time series that may contain uncertainty at each time point. Given a time point j, the value of the uncertain time series at j is denoted by $\hat{S}_u[j]$ and is represented as follows [20]

$$\hat{S}_u[j] = d_{uj} + e_{uj}.$$

Here d_{uj} is the true data value and e_{uj} is the arbitrary error. An uncertain time series is illustrated in Figure 2.

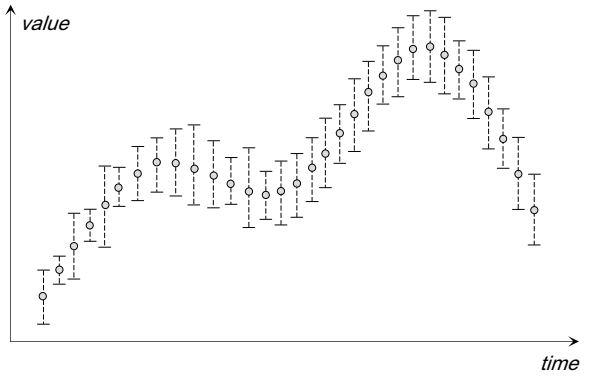


Figure 2. Continuous uncertain time series.

The above model of uncertain time series regards a timestamp as a variance and the pruned framework deals with the probability statistics. A measurement based on the probability function is explored in [2], which transforms the question into a cumulative distribution function (*cdf*). With a generation notion about the measurement for uncertain time series, a novel measurement quantifying the uncertainty is proposed in [3], which will degenerate to the Euclidean distance when the distance is large enough relative to the error.

3.2. *PROUD*

We define $\hat{S}_{ref}$ as a reference series with uncertainty and $\hat{S}_u$ as one of the items with noise stored in the database. Both kinds of series consist of the random variable in each time series. Given the definition of

$$Dst\left(\hat{S}_{ref},\hat{S}_u\right)=\sum_i\left(\hat{S}_{ref}[i]-\hat{S}_u[i]\right)^2,$$

searching solves the probabilistic problem with user defined distance threshold r and probability threshold $\tau \in (0, 1]$ [2]

$$Pr\left(Dst\left(\hat{S}_{ref},\hat{S}_u\right)\le r\right)^2\ge\tau. \tag{1}$$

PROUD [2] addresses an efficient judge inequation which is transformed from the original question for selecting candidates utilizing the property of cumulative distribution.

According to the *Central Limit Theorem*, the distance between $\hat{S}_{ref}$ and $\hat{S}_u$ namely $Dst\left(\hat{S}_{ref},\hat{S}_u\right)$ is treated as a joint variable with a corresponding mean and variance

$$Dst\left(\hat{S}_{ref},\hat{S}_u\right)\sim N\left(E\left(\hat{S}_{ref},\hat{S}_u\right),Var\left(\hat{S}_{ref},\hat{S}_u\right)\right). \tag{2}$$

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 a normal distribution, its *cdf* can be formalized as follows:

$$P(X\le x)=\Ö_{\mu,\sigma}(x)=\frac{1}{2}\left(1+erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right). \tag{3}$$

Finally, the cumulative distribution of the variable modeled by a pair of time series X and Y can be defined as:

$$Pr\left(Dst\left(\hat{S}_{ref},\hat{S}_u\right)\le r^2\right)=\frac{1}{2}\left(1+erf\left(\frac{Dst\left(\hat{S}_{ref},\hat{S}_u\right)-E\left(\hat{S}_{ref},\hat{S}_u\right)}{\sqrt{2}Var\left(Dst\left(\hat{S}_{ref},\hat{S}_u\right)\right)}\right)\right). \tag{4}$$

Since the monotonicity of ϕ increases, the candidates for the uncertain time series can be eventually transformed into the following inequation.

$$r_{norm}\left(\hat{S}_{ref},\hat{S}_u\right)=\frac{r^2-E\left(Dst\left(\hat{S}_{ref},\hat{S}_u\right)\right)}{\sqrt{2}Var\left(Dst\left(\hat{S}_{ref},\hat{S}_u\right)\right)}\ge\sqrt{2}erf^{-1}(2\tau-1)=r-limit. \tag{5}$$

Here, *r-limit* is a normalized threshold for the matching process. The error ratio is defined as the number of incorrect candidates divided by all candidates and the miss ratio is defined as the correct candidates divided by all correct candidates. Experiments in [2] show a flexible trade-off between the error ratio and the miss ratio through the user-defined distance threshold and probability threshold.

4. Index Construction with Uncertainty

In this section, we propose a novel approach to index the uncertain time series based on *PROUD*. Resulting from the uncountable measurement in *PROUD*, the first step is to give a measurable distance such as *Euclidean* and keep the same solution space compared to the primitive question. Then we find the correlation between the index structure and the measurement, with respect to the two user-defined thresholds. In the experiment, we find the limitation of the index for uncertain time series, especially for the one with the large noise. Hence, it is necessary to give a preprocess and optimization operation for the matching process.

4.1. Index Construction for Uncertain Time Series

We use a synopsis model with the *PROUD* scheme. Although there is a Haar decomposition on the *PROUD*, the index concentrates on the single stream data prune process and does not make use of the any relationship between two time series. We define $\hat{S}_{ref}$ as a reference series with uncertainty and $\hat{S}_u$ as a series with

noise stored in the database. Both kinds of series consist of the random variable in each timestamp. In [2], a pruned algorithm is explored by using the inequation $r_{norm}\left(\hat{S}_{ref},\hat{S}_u\right)\geq r-limit$ for the candidates satisfying the following filter inequation:

$$Pr\left(Dst\left(\hat{S}_{ref},\hat{S}_u\right)\leq r^2\right) = \frac{1}{2}\left(1+erf\left(\frac{Dst\left(\hat{S}_{ref},\hat{S}_u\right)-E\left(\hat{S}_{ref},\hat{S}_u\right)}{\sqrt{2}Var\left(Dst\left(\hat{S}_{ref},\hat{S}_u\right)\right)}\right)\right). \tag{6}$$

To construct the spatial index with *Euclidean* distance, we delve further into the algorithm *PROUD* to transform this uncountable measurement into a monotonic and consecutive form. In particular, under the assumption of the same distribution of timestamp in a time series for the sake of simplicity, the prune function can be changed as the following equation, which is a vital transformation for equal justice of candidates.

$$\begin{aligned}
&Pr\left(Dst\left(\hat{S}_{ref},\hat{S}_u\right)\leq r-limit\right)\geq\tau\\
&\Rightarrow\frac{r^2-E\left(Dst\left(\hat{S}_{ref},\hat{S}_u\right)\right)}{\sqrt{Var\left(Dst\left(\hat{S}_{ref},\hat{S}_u\right)\right)}}\geq r-limit\\
&\Rightarrow\sum_{i=1}^{len}\left(\mu_i-\mu_i^r\right)^2+r\\
&\quad -limit\sqrt{4\left(\sigma^2+\left(\sigma^r\right)^2\right)\sum_{i=1}^{len}\left(\mu_i-\mu_i^r\right)^2}\\
&\quad -\left(r^2-len\left(\sigma^2+\left(\sigma^r\right)^2\right)\right)\leq 0\\
&\Rightarrow 0\leq\sqrt{\sum_{i=1}^{len}\left(\mu_i-\mu_i^r\right)^2}\leq\frac{-b+\sqrt{b^2-4ac}}{2a}\\
&s.t.\ b^2-4ac\geq 0,\ \ a=1,\\
&\quad b=r-limit\sqrt{4\left(\sigma^2+\left(\sigma^r\right)^2\right)},\\
&\quad c=-r^2+len\left(\sigma^2+\left(\sigma^r\right)^2\right)
\end{aligned} \tag{7}$$

where $E\left(Dst\left(\hat{S}_{ref},\hat{S}_u\right)\right)$ and $Var\left(Dst\left(\hat{S}_{ref},\hat{S}_u\right)\right)$ are expanded and calculated in [2]. As we can see, the derivation explores the possibility of transforming an uncertain uncountable measurement into an exact monotonic form for dynamic progress. The last inequation turns out that *PROUD* can be changed as a *Euclidean* distance for the variance. Also, *Wavelet Summarization* is proposed in [2] for online stream data decomposition in a dynamic environment. It is focused on how to retain the vital coefficient in a limited memory source and on efficient summarization by keeping from extracting the coefficients from the index structure. However, they do not take into consideration the relationship of uncertain information between two random time series. On the contrary, we make full use of the uncertain information in each series by the inequation (7). As we can see, the required operation for the uncertain time series changes as a deterministic requirement for the series that consists of the means. However, the distance thresholds for these series are different from each other. The monotonic property must be shown as follows:

$$\begin{aligned}
VarThd\left(\sigma^2\right)=&\sqrt{\left(r\text{-}limit^2-len\right)\left(\sigma^2+\left(\sigma^r\right)^2\right)+r^2}\\
&-r\text{-}limit\sqrt{\sigma^2+\left(\sigma^r\right)^2}
\end{aligned} \tag{8}$$

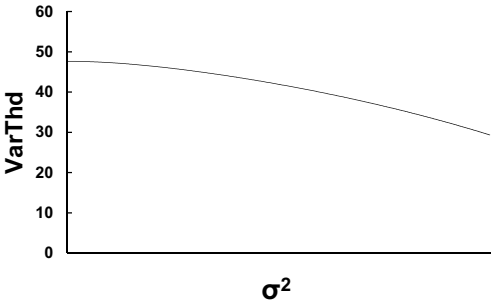
$$\begin{aligned}
&\frac{dVarThd\left(\sigma^2\right)}{d\sigma^2}\\
&=\frac{\left(r\text{-}limit^2-len\right)\sqrt{\sigma^2+\left(\sigma^r\right)^2}}{2\sqrt{\left(\left(r\text{-}limit\right)^2-len\right)\left(\sigma^2+\left(\sigma^r\right)^2\right)+r^2}\sqrt{\sigma^2+\left(\sigma^r\right)^2}}\\
&-\frac{r\text{-}limit\sqrt{\left(r\text{-}limit^2-len\right)\left(\sigma^2+\left(\sigma^r\right)^2\right)+r^2}}{2\sqrt{\left(\left(r\text{-}limit\right)^2-len\right)\left(\sigma^2+\left(\sigma^r\right)^2\right)+r^2}\sqrt{\sigma^2+\left(\sigma^r\right)^2}}
\end{aligned} \tag{9}$$

with a valid user-defined probability threshold τ in [0, 1], as well as the value of *r-limit* in $\left[-4\sqrt{2},4\sqrt{2}\right]$. Since the length is always greatly larger than $4\sqrt{2}$, which is a very trivial condition, the varying distance threshold is monotonically decreasing along σ when $\tau\geq 0.5$.

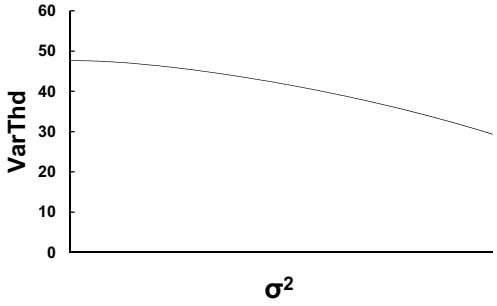
Most of the time, the length of the sequence is larger than the limited *r-limit*. Then we can assume that the coefficient *r-limit* – *len* is always a negative. With those sign assumptions, we can judge the monotonic property for σ, which is described in the following theorem.

Theorem1. Thefunction $f(x) = a\sqrt{x} - b\sqrt{ax+c}$ with $a < 0$, $-4\sqrt{2} \le b \le 4\sqrt{2}$, $c \ge 0$, $x \ge 0$ is a negative if $b > 0$ or $x > b^2c/(a(a-b^2))$ and a positive if $0 \le x \le b^2c/(a(a-b^2))$.

The theorem is simple and here we ignore the concrete detail of its proof, which is present in the appendix. We present a corresponding example of the varying threshold in Figure 3.



r-limit = 2, *len* = 40, σ^r = 1, *r* = 50



r-limit = –2, *len* = 40, σ^r = 1, *r* = 50

Figure 3. Varying threshold.

In our actual application, we define a = *r-limit*^{2} – *len*, b = *r-limit* and $c = r^2$, $x = \sigma^2 + (\sigma^r)^2$. Therefore, according to Theorem 1, with the case $\tau > 0.5$, we have *r-limit* > 0 and the *VarThreshold* is monotonically decreasing with $\sigma^2 + (\sigma^r)^2$. However, if *r-limit* ≤ 0 namely $\tau \le 0.5$, there are two monotonical directions which are divided by $b^2c/(a(a-b^2))$ named critical point. When the critical point is located in the range $[\sigma_{\min}, \sigma_{\max}]$ in an *MBR*, we must check both $\sigma_{\min}$ and $\sigma_{\max}$ for the minimum threshold and calculate the maximum threshold by critical point using $b^2c/(a(a-b^2))$.

Finally, an MBR in Figure 4 can be judged directly if it satisfies one of the following inequations:

$$\begin{aligned}&\max\sum_{i=1}^{d}\left(\mu_i-\mu_i^r\right)^2\\&\le\min\left(\sqrt{\left(r\text{-}limit^2-len\right)\left(\sigma^2+\left(\sigma^r\right)^2+r^2\right)}-r\text{-}limit\sqrt{\sigma^2+\left(\sigma^r\right)^2}\right)\\&=\sqrt{\left(r\text{-}limit^2-len\right)\left(\sigma_{\max}^2+\left(\sigma^r\right)^2+r^2\right)}-r\text{-}limit\sqrt{\sigma_{\max}^2+\left(\sigma^r\right)^2}\end{aligned} \tag{10}$$

$$\begin{aligned}&\min\sum_{i=1}^{d}\left(\mu_i-\mu_i^r\right)^2\\&\ge\max\left(\sqrt{\left(r\text{-}limit^2-len\right)\left(\sigma^2+\left(\sigma^r\right)^2+r^2\right)}-r\text{-}limit\sqrt{\sigma^2+\left(\sigma^r\right)^2}\right)\\&=\sqrt{\left(r\text{-}limit^2-len\right)\left(\sigma_{\min}^2+\left(\sigma^r\right)^2+r^2\right)}-r\text{-}limit\sqrt{\sigma_{\min}^2+\left(\sigma^r\right)^2}\end{aligned} \tag{11}$$

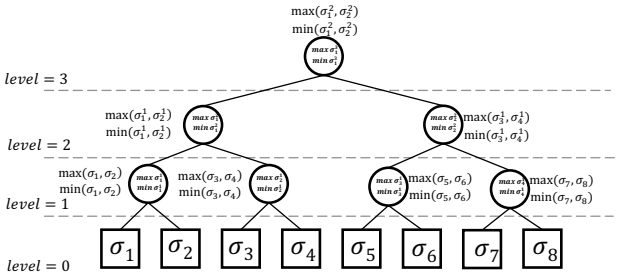


Figure 4. MBR with σ.

All leaf entries included in an index node satisfying the first inequation must be the candidates because no *VarThd* in all entries included in the *MBR* could be larger than their parent nodes. Therefore, the distance of leaf nodes to μ_{ref} covered by children entries must satisfy the first inequation, whereas those satisfying the second inequation must be pruned. Also, both extreme distances must be in the corners of the *MBR*, which means the complexity of finding max and min distances is $O(N)$, where N is the dimension of the point.

4.2. Search Strategy

In a recurrence processing that starts from the root of the index, we check all its children and determine if they satisfy one of the above inequations. It is shown that the time complexity of visiting all nodes in the index is $O(1)$ under the best condition and nearly $O(a^{\log_b^n})$ under the worst condition. If the height of the index is 1, the searching is degenerated into an ordered searching with the worst time complexity of $O(n)$. Compared with the time for calculating the measurement for two uncertain time series (especially in high dimensions), the visiting time takes up a few parts, meaning that in a global view, the index performance taking advantages outweighs taking disadvantages most of the time.

We summarize the searching in Algorithm 2. Given a target series $\hat{S}_{ref}$ and a root R of the *R*-tree* index, the time series which are preprocessed in advance, a distance threshold r, and a probability threshold τ, Algorithm 2 outputs a set of all candidates in which the probabilities that the distances of these candidates to the target reference series exceed r, is less than τ. First, we start with searching for a path from the root of the index and then compare each pair $\sigma_{\min}$, $\sigma_{\max}$ in *MBR*s included in the node. We check whether the inequation (10) or (11) is satisfied. If the inequation (10) is satisfied, we can directly get the leaves covered by this node. If the inequation (11) is satisfied, we just stop the deeper search along the path from this node. Otherwise, we have a deeper visit in the lower level.

4.3. Random Variance

Given the uncertain time series with the same variance, we construct the leaf entries including the same minimum and maximum σ valued variance at each timestamp. We can see that the forms of σ boundaries in each MBR are similar to each other. It means that we can heuristically treat the leaf entry as a point that includes the min and max σ in a time series. The following inequation (12) shows the correctness of this heuristic method but it may not be efficient because sometimes the approximation is sensitive to the large amount of noise.

Algorithm 1. GetCoveredLeaves.

Input: root N of sub tree
Output: *LEqueue*
1. **initialize:** Create *queue*; $N \overset{\text{push}}{\Rightarrow}$ *queue*
2. **while** *queue is not empty* **do**
3. $cN \overset{\text{pop}}{\Leftarrow}$ *queue*
4. **for** $i = 1, ..., cN.childNum$ **do**
5. **if** $cN.level = 0$ **then**
6. $cN.Child[i] \overset{\text{push}}{\Rightarrow}$ *LEqueue*
7. **else**
8. $cN.Child[i] \overset{\text{push}}{\Rightarrow}$ *queue*
9. **end if**
10. **end for**
11. **end while**

Algorithm 2. Search.

Input: root R of *R*-tree*, $\hat{S}_{ref}$ target uncertain time series, r distance treshold, τ probability treshold
Output: *candidates*
1. **initialize:**
2. Create *queue*
3. $R \overset{\text{push}}{\Rightarrow}$ *queue*
4. $r\text{-}limit \leftarrow \sqrt{2}\, erf^{-1}(2\tau - 1)$
5. judge the monotonical property
6. **while** *queue is not empty* **do**
7. $cN \overset{\text{pop}}{\Leftarrow}$ *queue*
8. **for** $i = 1, ..., cN.childNum$ **do**
9. calculate minimum/maximum tresholds
10. **if** minimum distance > maximum treshold **then**
11. //pruned
12. **else if** maximum distance < minimum treshold **then**
13. *GetCoveredLeaves* $\overset{\text{push}}{\Rightarrow}$ *candidates*
14. **else**
15. $cN.child[i] \overset{\text{push}}{\Rightarrow}$ *queue*
16. **end if**
17. **end for**
18. **end while**

With the max σ, this normalized distance will be minimum. If this minimum distance is larger than the maximum threshold, it satisfies (10), which means that the judge function in the leaf entry is the same as in the internal node of the index. Therefore, we can treat the leaf entry with random σ as internal node hit/pruned by (10)/(11).

$$\frac{r^2-\left(\left(\sigma_{\max}^2+\left(\sigma_{\max}^r\right)^2\right)\times len+\sum_{i=1}^{len}\left(\mu_i-\mu_i^r\right)^2\right)}{4\left(\sigma_{\max}^2+\left(\sigma_{\max}^r\right)^2\right)\sum_{i=1}^{len}\left(\mu_i-\mu_i^r\right)^2}$$
$$\le\frac{r^2-\left(\sum_{i=1}^{len}\left(\sigma_i^2+\left(\sigma_i^r\right)^2\right)+\sum_{i=1}^{len}\left(\mu_i-\mu_i^r\right)^2\right)}{4\sum_{i=1}^{len}\left(\sigma_i^2+\left(\sigma_i^r\right)^2\right)\left(\mu_i-\mu_i^r\right)^2}$$
$$\le\frac{r^2-\left(\left(\sigma_{\min}^2+\left(\sigma_{\min}^r\right)^2\right)\times len+\sum_{i=1}^{len}\left(\mu_i-\mu_i^r\right)^2\right)}{4\left(\sigma_{\min}^2+\left(\sigma_{\min}^r\right)^2\right)\sum_{i=1}^{len}\left(\mu_i-\mu_i^r\right)^2} \quad (12)$$

4.4. Optimization for Random Variance

The result of experiments with random variance time series has shown that the time cost of processing is close to the one shown by *PROUD* for the approximation of the variance. Since σ has a large variance, the index will hardly meet the filter equations and always search in leaf entries. For a closer approximation, several preprocesses can be considered. The preprocess such as Discrete Fourier transform (*DFT*) or Moving Average (*MA*) dimension reduction operation is carried out. Different preprocesses focus on different targets. *DFT* focuses on selecting high frequencies and is inefficient for white noise. It merely carries out the dimension reduction by centralizing the energy on several dimensions with high frequency. The *MA* makes use of the relation of adjacent timestamps for a better approximation for noisy series with probabilistic information.

In this paper, we apply a soft preprocess based on *MA* like [15]. In [15], an Uncertain Moving Average (*UMA*) is presented to accommodate σ in a traditional preprocess. There are two kinds of *MA* for uncertain time series in [15]. The first one is to simply use mathematic average and the second one is based on the exponential function. Both of them are weighted the values by σ at each timestamp. This means that it is useful for uncertain time series with random σ and the values with larger σ offer less contribution to the final average. For the sake of simplicity, it is assumed that the whole series has the same σ for presenting the Euclidean distance between *UMA* values by classical *MA* values.

Since

$$x_i^{UMA}=\frac{1}{2w+1}\sum_{k=i-w}^{i+w}\frac{x_k}{\sigma_k},\quad 1\le i\le m \quad (13)$$

with a window width w and variance σ_k corresponding to k-th timestamp influences the values sensitively by σ_k and may change excessively the primitive x_k, here we present a new preprocess named weighted variance uncertain moving average, based on [15].

$$x_i^{WUMA}=\sum_{k=i-w}^{i+w}\frac{\frac{1}{\sigma_k^2}}{\sum_{j=i-w}^{i+w}\frac{1}{\sigma_j}}x_k,\quad 1\le i\le m \quad (14)$$

It shows that with a larger σ, the values contribute less reliable distances to the real original distance. A larger distance with a smaller σ tells us that this stamp is greatly possible to have large real distance and hence greatly contribute to the real distance which will be essential when comparing it with an original threshold. Meanwhile, it keeps a light scale on the primitive value.

Suppose that the time series has the same variance in each timestamp, then we have

$$x_i^{WUMA}=\frac{1}{(2w+1)\sigma}\sum_{k=i-w}^{i+w}x_k=x_i^{UMA} \quad (15)$$

It explores the relationship between x_i^{WUMA} and x_i^{UMA}. Compared with *UMA*, there will be no difference in performance with a series consisting of the same variance in each timestamp. Moreover, we normalized the x_i with the weighted σ which will not dramatically affect measurement and hence keep a closer approximation about the sum of all primitive measurements. The mean and variance for the normalized timestamp are shown in Figure 5.

$$E\left(x_i^{WUMA}\right)=\frac{1}{\sum_{j=i-w}^{i+w}\frac{1}{\sigma_j}}\sum_{k=i-w}^{i+w}\frac{w_k}{\sigma_k},\quad 1\le i\le m$$

$$Var\left(x_i^{WUMA}\right)=\frac{1}{\left(\sum_{j=i-w}^{i+w}\frac{1}{\sigma_j}\right)^2}\sum_{k=i-w}^{i+w}\frac{1}{\sigma_k^3},\quad 1\le i\le m \quad (16)$$

After normalizing the time series using (16), the distribution of σ will be more uniform and critical point has less impact on the efficiency. Then we construct the *MBR* entries and build the index based on this preprocessed time series.

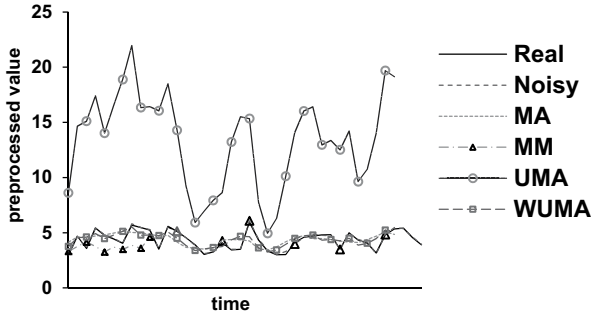


Figure 5. Preprocess methods.

The algorithm *Adjust* and other operations in the traditional index must be altered for uncertain time series. We just consider a vital operation called *Insert* for this step, including *Split* and *Reinsert*, which both determine the path length and the search efficiency. The traditional split algorithm does not consider the use of uncertainty information while grouping the entries in the index node. In addition, the goodness measure for the *Reinsert* does not involve the variance. Heuristic optimization for the split and re-insert operation cooperates with the minimum and maximum variances which are plugged in every *MBR*. The larger interval of the σ is, the more possible is the value corresponding to the peak point (see Figure 3) located in the range minimum and maximum of σ. Hence, the queries profit from the smaller variance just like the margin described in [24]. Central σ requires less visit to prune/hit the entry in the leaf node by using judge inequations.

Before the variant *Split* and *Reinsert* are introduced, we review the general *Split* algorithm as well as *Reinsert* algorithm. The *R*-tree* in [24] uses the following method to find good splits: along each axis, the entries are first sorted by the lower value and then by the upper value of their rectangles. For each sort of M – 2*m* + 2 distributions of the *M* + 1 entries, two groups are determined by three goodness measure values. The first group contains the first (m – 1) + *k* entries and the second group contains the remaining entries. For each distribution, goodness measure values are determined by

1. area-value,
2. margin-value, and
3. overlap-value.

To achieve dynamic recognition, the *R*-tree* forces entries to be reinserted along the insertion routine by

1. choosing subtree,
2. judging whether it overflows and
3. doing *Reinsert* or *Split* according to the inserted time.

As we can see, all of them optimize the index from the view of geometry, without considering uncertainty gradients. This results in a deep visit like shown in Figure 6.

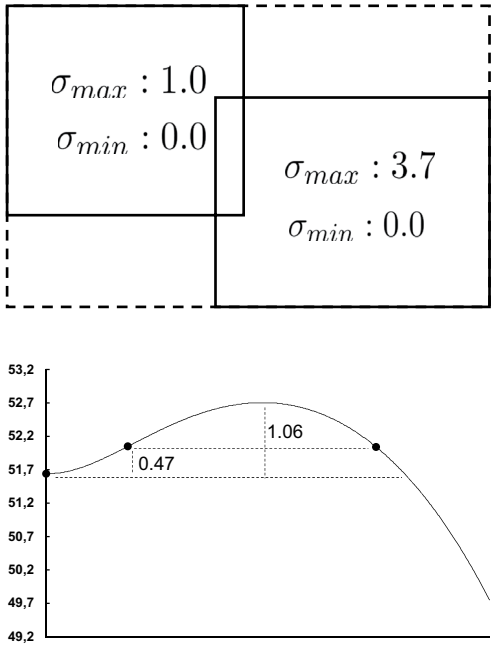


Figure 6. MBR threshold interval.

Given *r-limit* = –2, *len* = 40, σ^r = 1, *r* = 50, the *MBR* with min and max σ below the border 2.3 can give a monotonical direction of the threshold. Obviously, in Figure 6, the left-top *MBR* has a varying threshold in a monotonical and tighter varying threshold range $[VarThd(\sigma_{\min}), VarThd(\sigma_{\max})]$ and the right-bottom *MBR* presents a wider range with a deter-

ministic peak

$[\min(VarThd(\sigma_{\min}), VarThd(\sigma_{\max})),$

$VarThd((r\text{-}limit^2 r^2)/((len - r\text{-}limit^2)len) - (\sigma^r)^2)]$.

Finally, the range of the left-top *MBR* is *VarThd* (1.0) – *VarThd* (0.0), which equals to 0.47. The range of the right-bottom *MBR* is *VarThd* (2.3) – *VarThd* (0.0), which equals to 1.06. Therefore, more distance is in the wider range in the right-bottom *MBR*, which causes the specific distance value to hardly meet one of the prune/candidate inequations (10)/(11).

We present two goodness measures against variance in the following formulations.

1. *Split*. With a new goodness measure, *VarMargin*:

$$\sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} \left(\delta_{\max}^{i} - \delta_{\min}^{i}\right)$$

the *split* is inclined to generate two groups with minimum variance interval. We set this goodness measure with the top level when the split operation happens.

2. *Reinsert*. With a new goodness measure, *VarOverlap*

$$\sum_{i=1}^{N} \sum_{j=1, j\neq 1}^{N} \left(\min\left(\delta_{\max}^{i}, \delta_{\max}^{i}\right) - \max\left(\delta_{\min}^{i}, \delta_{\min}^{i}\right)\right)$$

Reinsert trends to minimum interval overlap of the variance in each *MBR*. We set the priority of this goodness measure value higher than the one measured against the relative distance to the *MBR* center.

5. Experimental Performances

Under the assumption that each timestamp in uncertain time series has the same variance, performance of the index approach is the same as the one of *PROUD*. Hence, in this section, we focus only on the random variance at each timestamp for all series.

To evaluate the performance of *R*-tree*, we conducted experiments with synthetic data and compared the efficiency of *Search* operation and qualified the performance by

$$miss\ ratio = \frac{true\ candidates\ not\ in\ selection}{all\ true\ candidates\ in\ database}$$

$$error\ ratio = \frac{false\ candidates\ in\ selection}{number\ of\ candidates\ selected}$$

First, we compared the *Search* operation of our index approach with the *PROUD* prune algorithm in an ordered sequence. Then we evaluated the improvement of the central variance *Split* against the general *Split*. The general *Split* does not make use of the variance stored in the *MBR*. Our experiments were conducted in Visual Studio and run on the PC with 2.4 GHz CPU and 4 GB RAM.

5.1. Experiments with Synthetic Data

The parameters which are used in our experiments are listed in Table 1. The distance threshold is generated dynamically during the processing of the beginning phase. We generated the data set including *num* series that have *len* timestamps. Each experiment would run *group* times and the average results were finally presented. The *dratio* was used to multiply he original uniform variance to change the variance of each timestamp. We picked τ in [0.1, 0.2, …, 0.9] when we compared the *PROUD* with our index approach using miss/error ratio as our evaluation standard. We constructed our index after the preprocess for less overlap between the interval of the variance in each *MBR* and make the Euclidean distance of σ the same as raw measurement. The ground truth is based on real data without uncertain noise. As pointed out in [15], the impact of filtering uncertain time series works only in a specified range. We

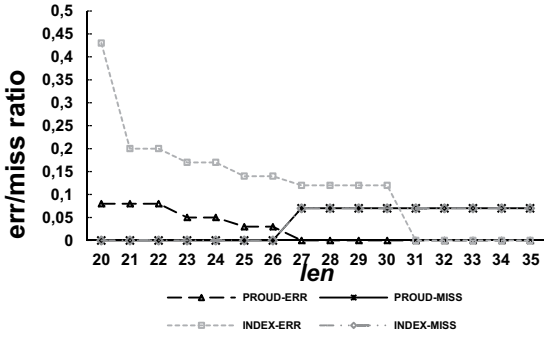
Table 1. Parameters we use in experiments.

Parameter	Value	Desc.
len	100	Sequence length
τ	[0.1, 0.9]	Probability threshold
σ	1.0	Variance of sequence
dratio	[0.1, 2.0]	Variance ratio
num	≤ 5000	Sequence number per group
group	10	Test groups
x_i	[0, 3]	The value in each timestamp

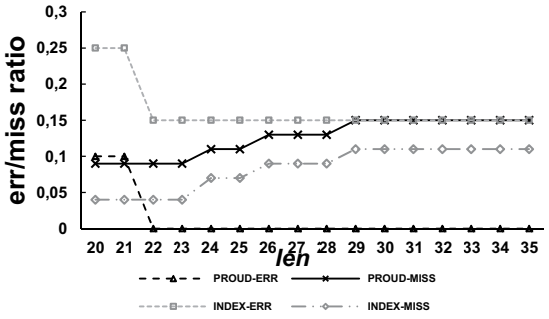
calculated the average distance from the uncertain series to the reference series as the distance threshold to enlarge the performance gap between the effect of the index and the *PROUD*.

Figure 7 shows the results of the index compared with the *PROUD*. It is evident that the index has a transaction between efficiency and error ratio. As we can see, both algorithms have a violent reaction to the distance threshold instead of the probability. Moreover, when both have the same error ratio, the distance threshold of the index is less than the one of the *PROUD*, which means that the index needs more rigid threshold and performs poorly in selecting true candidates. On the contrary, the miss ratio of the index is much looser than the one of the *PROUD*. However, there is no difference between *PROUD* and index in sensitivity for the distance or the probability since the slopes of error ratio are parallel.

In Figure 8 and Figure 9, the error ratio of the index is always higher than the one of the *PROUD*. It is obviously true with larger noise, since the interval $\delta_{\max} - \delta_{\min}$ in leaf entries is too big to keep a closer approximation by for-



(a) $\tau = 0.7$



(b) $\tau = 0.2$

Figure 8. Missing and error ratio with synthetic data.

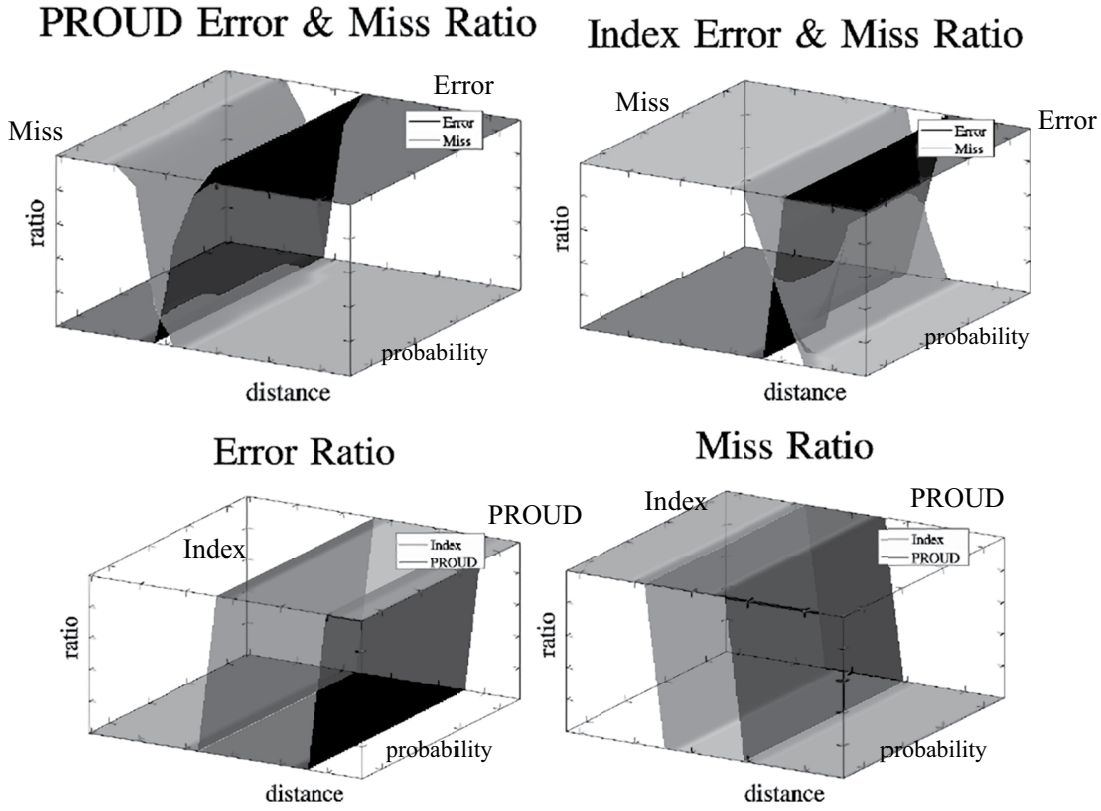


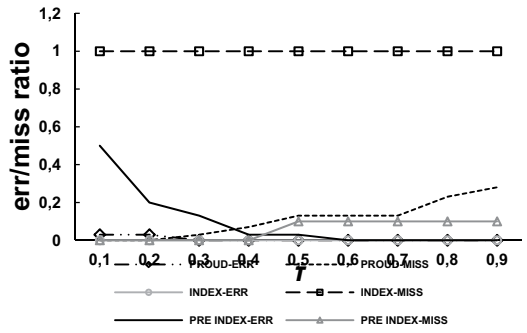
Figure 7. Missing and error ratio of *PROUD* and Index.

mulations (10) and (11). Hence, the preprocess for uncertain time series before constructing the index gives a great improvement in Figure 9, as we can see, the index for the preprocessed series has a great balance between error ratio and miss ratio like the *PROUD*. Even with a lot of noise, the index is still performing well.

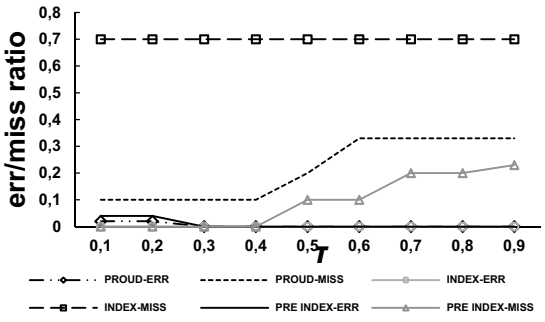
Here, we give the performance comparison for *PROUD* and *R*-tree* under different dimensions of time series. There is no doubt that the slopes of curves are the same, since the key judgement for candidates is determined by user-defined distance threshold and probability thresholds. Specifically, with the probabilistic thresholds of 0.7 and 0.2, both performances have a dramatic change. Afterward, we compare the influence of probability threshold on the pruning process under different noise. It turns out that with a large variance of the σ, the index without preprocess didn't work well, since the *MBR* is likely to cross the critical point, which makes threshold non-monotonic in relation to the variance. But then, after using the preprocess, the index performs better, both for the miss ratio and for the error ratio. In summary, index structure gives a flexible tradeoff between performance and efficiency with a little loss in accuracy.

5.2. Experiments with Data Originated from Real Data

We use the archive files containing average daily temperatures for 157 U.S. and 167 international cities. Source data for these files are from the Global Summary of the Day (*GSOD*) database archived by the National Climatic Data Center (*NCDC*). The average daily temperatures posted on this site are computed from 24 hourly temperature readings in the Global Summary of the Day (*GSOD*) data. The data fields in each file posted on this site are month, day, year, average daily temperature (*F*). Data containing "–99" no-data flag is not available. Since some cities contain heavily missed data and the spatial index requires equal-length series, we exclude cities *azflagst*, *azyuma*, *behmlton*, *bibjmbra*, *bwdhaka*, *cnmontrl*, *cynicosi*, *dewilmin*,



(a) dratio $\in [0.1, 1.0]$



(b) dratio $\in [0.1, 2.0]$

Figure 9. Preprocess with synthetic data.

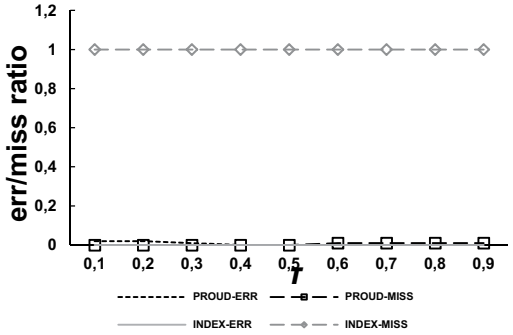


Figure 10. PROUD vs. *INDEX* using data originated from real data.

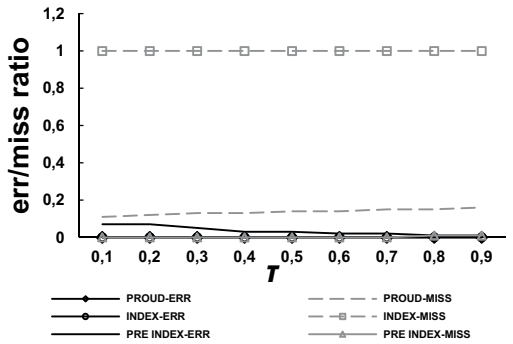


Figure 11. Preprocess with data originated from real data.

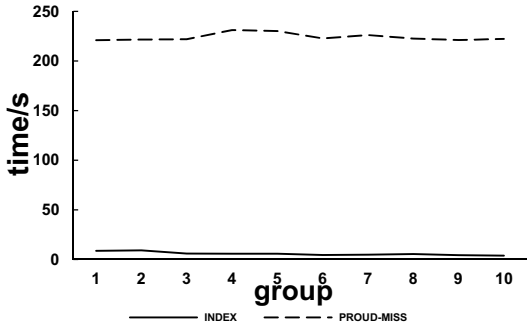
dlbonn, *dlfrnkft*, *fldaytna*, *gygrgtwn*, *istelavi*, *istelaviv*, *labatonr*, *mwlilngw*, *paharris*, *rstblisi*, *rsyervan*, *slfretwn*, *ygpristn*. The way of generating noise as well as the synthetic data through *dratio* and σ is done in the same manner.

5.3. Evaluations of Running Time

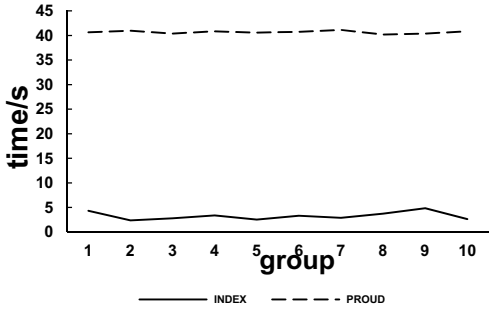
Clearly, the running time for querying target item with index should be more efficient than with the *PROUD*, because we condense the linear comparing operation in several directories of the index. Here we perform several experiments over databases with different sizes. The *dratio* is always generated between 0.1 and 2.0. The results shown in Figure 12 are as what we expected. Performance of the index with size 5000 is as well as good the size 500.

5.4. Variant Insertion

To evaluate performance of the variant index, we conduct extensive experiments and compare the running time of *Insert* with the variant *Insert* under both synthetic data and real data. It is shown in Figure 13 that, with a larger variance, which means greater uncertainty, general *Insert* produces more overlap of the variance, which requires deeper visits in the subtree and eventually performs unsteadily in time cost. Meanwhile, the *Variant Index* keeps a smooth fluctuation under different uncertainty levels and with lower time cost than *Index* does under most conditions.

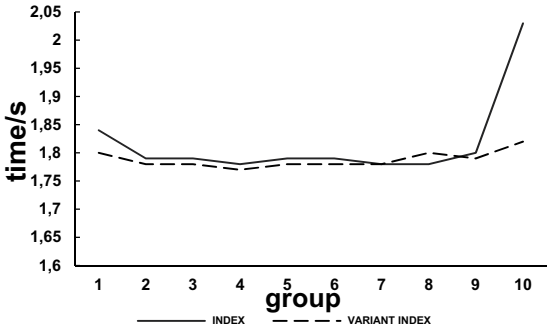


(a) num = 5000

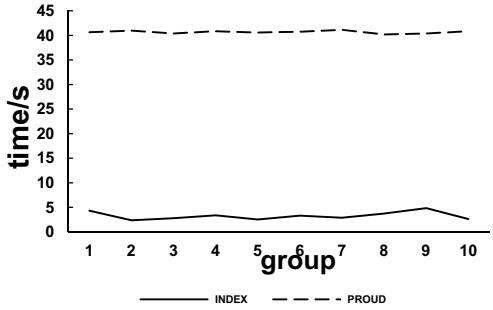


(a) num = 500

Figure 12. Running time with synthetic data.



(a) dratio $\in [0.1, 1.0]$



(a) dratio $\in [0.1, 2.0]$

Figure 13. Running time using synthetic data.

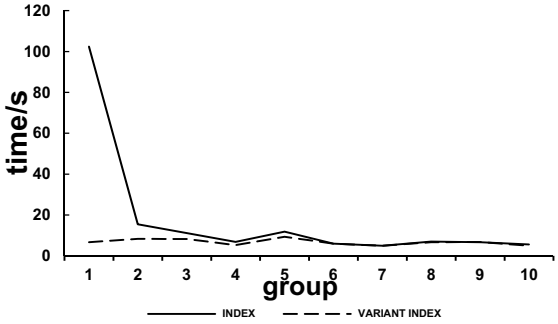


Figure 14. Running time using data originated from real data.

6. Conclusions

For a given time series with random variance, Figure 7 shows the sensitive range by *PROUD* prune process as well as the index search process. Although the index search process threshold delays, the change gradient is almost parallel. It means that the sensitivity to distance or the probability are similar. However, if the variance interval in one series is too large to meet the index candidate inequation, the result will be inclined to higher miss and lower error ratio since the critical point is likely to be positioned in the variance interval. By decreasing the effect of the variance and protecting *Euclidean* distance from a dramatic change using preprocess, we can see that the index performance is good in Figure 9.

It is believed that the index can accelerate search in static databases. In this paper, we deal with how to construct a spatial index for uncertain time series. In our future work, we will work on indexing dynamic databases or data streams in a quickly changing environment. Moreover, given different targets, the index performance will dramatically be up and down. From the research here presented, it is clear that the index with variant *Insert* is more robust and smoother compared to general *Insert*.

Acknowledgement

This work was in part supported by National Natural Science Foundation of China (61772269).

References

[1] J. Assfalg *et al*., "Probabilistic Similarity Search for Uncertain Time Series", in *Proceedings of the 2009 International Conference on Scientific and Statistical Database Management*, 2009, pp. 435–443. https://doi.org/10.1007/978-3-642-02279-1_31

[2] M.-Y. Yeh *et al*., "*PROUD*: a Probabilistic Approach to Processing Similarity Queries Over Uncertain Data Streams", in *Proceedings of the 12th International Conference on Extending Database Technology*, 2009, pp. 684–695. http://dx.doi.org/10.1145/1516360.1516439

[3] S. R. Sarangi and K. Murthy, "*DUST*: a Generalized Notion of Similarity Between Uncertain Time Series", *Proceedings of the 2010 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 383–392. http://dx.doi.org/10.1145/1835804.1835854

[4] Y. Zhao *et al*., "On Wavelet Decomposition of Uncertain Time Series Data Sets", in *Proceedings of the 2010 ACM International Conference on Information and Knowledge Management*, 2010, pp. 129–138. http://dx.doi.org/10.1145/1871437.1871458

[5] G. Batista *et al*., "A Complexity-invariant Distance Measure for Time Series", in *Proceedings of the 2011 SIAM International Conference on Data Mining*, 2011, pp. 699–710. http://dx.doi.org/10.1137/1.9781611972818.60

[6] C. C. Kuo *et al*., "Time Series Index for GIS Partial Discharge Detection", in *Proceedings of the 2011 Asia-Pacific International Conference on Lightning*, 2011, pp. 364–367. http://dx.doi.org/10.1109/APL.2011.6110143

[7] K. L. Liao *et al*., "Wavelet Decomposition Algorithm for Uncertain Data Streams", in *Proceedings of the 2011 International Conference on Computer Science and Education*, 2011, pp. 965–970. http://dx.doi.org/10.1109/ICCSE.2011.6028796

[8] D. Oliver *et al*., "Geo-referenced Time-series Summarization Using k-full Trees: a Summary of Results", in *Proceedings of the 2012 IEEE International Conference on Data Mining Workshops*, 2012, pp. 797–804. http://dx.doi.org/10.1109/ICDMW.2012.64

[9] M. Orang and N. Shiri, "A Probabilistic Approach to Correlation Queries in Uncertain Time Series Data", in *Proceedings of the 2012 ACM International Conference on Information and Knowledge Management*, 2012, pp. 2229–2233. http://dx.doi.org/10.1145/2396761.2398607

[10] J.-W. Roh *et al*., "Efficient Bitmap-based Indexing of Time-based Interval Sequences", *Information Sciences*, vol. 194, no. C, pp. 38–56, 2012. http://dx.doi.org/10.1016/j.ins.2011.08.013

[11] Y. Zuo *et al*., "Similarity Matching Over Uncertain Time Series", in *Proceedings of the 2012 International Conference on Computational Intelligence and Security*, 2012, pp. 1357–1361. http://dx.doi.org/10.1109/CIS.2011.302

[12] M. Orang and N. Shiri, "An Experimental Evaluation of Similarity Measures for Uncertain Time Series", in *Proceedings of the 18th International Conference on Database Engineering and Applications Symposium*, 2014, pp. 261–264. http://dx.doi.org/10.1145/2628194.2628207

[13] M. S. Gil *et al*., "Fast Index Construction for Distortion-free Subsequence Matching in Time-Se-

ries Databases", in *Proceedings of the 2015 International Conference on Big Data and Smart Computing*, 2015, pp. 130–135. http://dx.doi.org/10.1109/35021BIGCOMP.2015.7072822

[14] J. Hwang *et al*., "GPU Acceleration of Similarity Search for Uncertain Time Series", in *Proceedings of the 2015 International Conference on Network-Based Information Systems*, 2015, pp. 627–632. http://dx.doi.org/10.1109/NBiS.2014.89

[15] M. Orang and N. Shiri, "Improving Performance of Similarity Measures for Uncertain Time Series Using Preprocess Techniques", in *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*, 2015, pp. 1–12. http://dx.doi.org/10.1145/2791347.2791385

[16] U. Agarwal and A. S. Sabitha, "Time Series Forecasting of Stock Market Index", in *Proceedings of the 2016 India International Conference on Information Processing*, 2016, pp. 1–6. http://dx.doi.org/10.1109/IICIP.2016.7975381

[17] M. Orang and N. Shiri, "Correlation Analysis Techniques for Uncertain Time Series", *Knowledge and Information Systems*, 2017, vol. 50, no. 1, pp. 79–116. http://dx.doi.org/10.1007/s10115-016-0939-7

[18] B. Goswami *et al*., "Abrupt Transitions in Time Series with Uncertainties", *Nature Communications*, vol. 9, no. 1, pp. 48, 2018. http://dx.doi.org/10.1038/s41467-017-02456-6

[19] R. Cheng *et al*., "Efficient Indexing Methods for Probabilistic Threshold Queries over Uncertain Data", *Proceedings of the 2004 International Conference on Very Large Data Bases*, 2004, pp. 876–887. http://dx.doi.org/10.1016/B978-012088469-8.50077-2

[20] X. Lian *et al*., "Pattern Matching Over Cloaked Time Series", in *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, 2008, pp. 1462–1464. http://dx.doi.org/10.1109/ICDE.2008.4497590

[21] A. Guttman, "*R-trees*: a Dynamic Index Structure for Spatial Searching", in *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, 1984, pp. 47–57. http://dx.doi.org/10.1145/602259.602266

[22] N. Roussopoulos and D. Leifke, "Direct Spatial Search on Pictorial Databases Using Packed *R-trees*", *SIGMOD Record*, vol. 14, no. 4, pp. 17–31, 1985. http://dx.doi.org/10.1145/971699.318900

[23] U. Deppisch, "*S-Tree*: a Dynamic Balanced Signature Index for Office Retrieval", in *Proceedings of the 9th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1986, pp. 77–87. http://dx.doi.org/10.1145/253168.253189

[24] N. Beckmann *et al*., "The *R*-tree*: an Efficient and Robust Access Method for Points and Rectangles", in *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, 1990, pp. 322–331. http://dx.doi.org/10.1145/93597.98741

[25] S. Berchtold *et al*., "The *x-tree*: an Index Structure for High-dimensional Data" in *Proceedings of the 22nd International Conference on Very Large Data Bases*, 1996, pp. 28–39.

[26] G. Kaiser, "The Fast *Haar* Transform", *IEEE Potentials*, vol. 17, no. 2, pp. 34–37, 1998. http://dx.doi.org/10.1109/45.666645

[27] Y. Matias *et al*., "Wavelet-based Histograms for Selectivity Estimation", *SIGMOD Record*, vol. 27, no. 2, pp. 448–459, 1998. http://dx.doi.org/10.1145/276305.276344

[28] B. K. Yi *et al*., "Efficient Retrieval of Similar Time Sequences Under Time Warping", in *Proceedings of the 1998 International Conference on Data Engineering*, 1998, pp. 201–208. http://dx.doi.org/10.1109/ICDE.1998.655778

[29] K. P. Chan and A. W. C. Fu, "Efficient Time Series Matching by Wavelets", in *Proceedings of the 1999 International Conference on Data Engineering*, 1999. pp. 126–133. http://dx.doi.org/10.1109/ICDE.1999.754915

[30] S. Y. Park *et al*., "Efficient Searches for Similar Subsequences of Different Lengths in Sequence Databases", in *Proceedings of the 2000 International Conference on Data Engineering*, 2000, pp. 23–32. http://dx.doi.org/10.1109/ICDE.2000.839384

[31] B. K. Yi and C. Faloutsos, "Fast Time Sequence Indexing for Arbitrary Lp Norms", in *Proceedings of the 26th International Conference on Very Large Data Bases*, 2000, pp. 385–394. http://dx.doi.org/10.1184/r1/6605618

[32] M.-Y. Yeh *et al*., "LeeWave: Level-wise Distribution of Wavelet Coefficients for Processing KNN Queries Over Distributed Streams", in *Proceedings of the VLDB Endowment*, 2008, pp. 586–597. http://dx.doi.org/10.14778/1453856.1453921

[33] R. Ma *et al*., "Solar Flare Prediction Using Multivariate Time Series Decision Trees", in *Proceedings of the 2017 IEEE International Conference on Big Data*, 2017, pp. 2569–2578. http://dx.doi.org/10.1109/BigData.2017.8258216

[34] R. Ma and R. A. Angryk, "Distance and Density Clustering for Time Series Data", in *Proceedings of the 2017 IEEE International Conference on Data Mining Workshops*, 2017, pp. 25–32. http://dx.doi.org/10.1109/ICDMW.2017.11

[35] R. Ma *et al*., "A Data-driven Analysis of Interplanetary Coronal Mass Ejecta and Magnetic Flux Ropes", in *Proceedings of the 2016 IEEE International Conference on Big Data*, 2016, pp. 3177–3186. http://dx.doi.org/10.1109/BigData.2016.7840973

[36] D. M. Woodbridge *et al*., "Time Series Discord Detection in Medical Data Using a Parallel Relational Database", in *Proceedings of the 2015 IEEE International Conference on Bioinformatics and Biomedicine*, 2015, pp. 1420–1426. http://dx.doi.org/10.1109/BIBM.2015.7359885

[37] R. Ma *et al*., "Coronal Mass Ejection Data Clustering and Visualization of Decision Trees", *The Astrophysical Journal Supplement Series*, vol. 236, no. 1, pp. 4, 2018. http://dx.doi.org/10.3847/1538-4365/aab76f

Received: December 2017
Revised: June 2018
Accepted: July 2018

Contact addresses:
Diwei Zheng
College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Nanjing 211102
China
e-mail: 1165696276@qq.com

Li Yan*
College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Nanjing 211102
China
e-mail: yanli@nuaa.edu.cn
*Corresponding author

Yu Wang
College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Nanjing 211102
China
e-mail: 826531768@qq.com

Appendix

Proof of Theorem 1

We separate the function into two conditions according to the sign of b.

1) $b \leq 0$

Obviously $f(x)$ will be monotonically increased since x is non-negative.

2) $b > 0$

$$\begin{aligned} f(x) > 0 &\Rightarrow a\sqrt{x} - b\sqrt{ax+c} > 0 \\ &\Rightarrow a\sqrt{x} > b\sqrt{ax+c} \\ &\Rightarrow \left(a^2 - b^2 a\right)x > b^2 c \\ &\Rightarrow x > b^2 c / \left(a^2 - b^2 a\right) \end{aligned}$$

In a similar way, we have $f(x) \leq 0 \Rightarrow x \leq (b^2 c)/(a^2 - b^2 a)$.

We can make sure that the sign of b^2c is non-negative since $c \geq 0$. But there is no constraint on $a^2 - b^2$ a. Hence the sign about the target $f(x)$ cannot be sure when $b > 0$.

Diwei Zheng is currently a master candidate at the College of Computer Science and Technology at the Nanjing University of Aeronautics and Astronautics, China. His research interests include time series analysis and uncertain data management.

Li Yan is a full professor at the College of Computer Science and Technology at the Nanjing University of Aeronautics and Astronautics, China. Her current research interests include uncertain data and knowledge engineering.

Yu Wang is currently a master candidate at the College of Computer Science and Technology at the Nanjing University of Aeronautics and Astronautics, China. His research interests include time series analysis and uncertain data management.