http://ansinet.com/itj

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (23): 7280-7288, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.7280.7288


# Trend Based Sketching for Massive Uncertain Time Series Clustering

[1]Jingyu Chen, [1]Ping Chen and [2]Xian'gang Sheng
[1]School of Computer Science and Technology, Xidian University, No. 2 South Taibai Road, Xi'an, 710071, Shaanxi, China
[2]College of Information Engineering, Qingdao University, Qingdao, 266071, Shandong, China

**Abstract:** Due to the inaccuracy and noisy, uncertainty is inherent in time series data and increases the complexity of clustering. For the massive data size, efficient data storage is a crucial task. Based on the Hilbert SFC, a trend sketches is constructed to store trends of the uncertain time series. And based on divergence and sketch metric, a sketch based similarity is given. Then a clustering algorithm is proposed to improve the quality of clustering. The experimental results are shown in Final.

**Key words:** Uncertainty, Hilbert SFC, sketch, divergence, clustering

## INTRODUCTION

With the development of sensor and internet technique, the capabilities of collecting data are enhanced greatly. Many applications, such as: Web log analysis, network traffic monitoring, real-time traffic monitoring, produce massive time series data from various sensors. Analysis and mining these massive time series data can be extremely challenging. Due to the massive data size, it is necessarily to adopt the compressed and compression and approximation techniques to improve the efficiency and consumption of the mining algorithm.

In many applications, uncertainty often exists because of network failure, noise and sampling error, etc. For time series data, there may be several different values at a time. The uncertainty of values in time series should be handler in the similarity calculation of clustering to improve the accuracy of mining.

To address the problem of clustering uncertain data time series, we suggest a Hilbert SFC (Lawder, 2000) based sketch approach to reduce storage spaces and computational complexity. We use Hilbert SFC to compress sequences fragment to gain trend sequence information and store the trends in trend sketches. Then, based on the KL divergence (Jiang *et al*., 2013) and sketch *-metric (Anceaume and Busnel, 2012), we present our sketch based similarity for time series objects distances measure. And then according the average distances in trend sketches, we map objects to different core-sets. Based on core-sets, we measure the similarity of objects and select represent object for each core-set. By using the max-min cluster distance measure, the initial cluster centers selection algorithm is proposed to improve the quality of clustering. Based on outlier handling and UK-means (Wang and Yuan, 2006), we design an algorithm to cluster uncertain data streams. Final, we verify the accuracy and efficiency of the proposed scheme via experiments.

The rest of the paper is organized as follows. Section II discusses related work. Section III presents the Hilbert SFC based trend sketches model for uncertain time series data. Section IV outlines divergence based similarity and clustering methods. Simulation methodology and performance evaluation result and analysis are presented in section V and we conclude the work in section VI.

## RELATED WORKS

Mining massive time series data has been attracting much attention in research and practice. Uncertainty brings new challenges to clustering, since it increases the complexity of the measurement of similarity between uncertain time series objects.

To handle the uncertain time series data, it is a common way to expand data mining algorithms through handling the uncertainty of data. In recent years there have been a plenty of methods for managing and mining uncertain time series and data and streams. Ngai *et al*. (2006) proposed the UK-means method that enhances the K-means algorithm to handle data uncertainty. UK-means measures the distance between an uncertain object and the cluster center with the expected

**Corresponding Author:** Jingyu Chen, School of Computer Science and Technology, Xidian University, No. 2 South Taibai Road, Xi'an, 710071, Shaanxi, China

distance. Kriegel and Pfeifle (2005) proposed the FDBSCAN algorithm based DBSCAN algorithm (Ester *et al*., 1996) for clustering probabilistic data. Kriegel and Pfeifle (2005) developed a probabilistic extension of OPTICS (Ankerst *et al*., 1999) called FOPTICS for clustering uncertain data objects. Ackermann *et al*. (2012) present a new corsets trees based clustering algorithm to improve quality of stream clustering. Tran *et al*. (2010) present the PODS model for processing uncertain data using continuous random variables. Nie *et al*. (2012) employ a time-varying graph model to represent imprecise object relationships with compression and present a probabilistic algorithm to estimate the most likely location.

For handling the massive data size quickly and efficiently, sketching is a popular method for processing massive time series data. Sketch techniques use a sketch vector as a data structure to store the streaming data compactly in a small-memory footprint. The main advantage of using these sketch techniques (Cormode and Muthukrishnan, 2005) is that they require a storage which is significantly smaller than the input stream length. Sketch techniques are used in stream data frequent items mining (Manerikar and Palpanas, 2008), clustering (Aggarwal, 2009) and anomaly detection (Liu *et al*., 2010) recently. Papapetrou *et al*. (2010) present a novel sketching technique ECM-sketch that allows effective summarization of distributed data streams over sliding windows with probabilistic accuracy guarantees. For measuring the distance between updatable sketches, Anceaume *et al*. present a novel metric Sketch *-metric that reflects the relationships between any two discrete probability distributions in the massive data streams (Anceaume and Busnel, 2012).

Nowadays, the study about clustering time series data is mainly about similarity metrics. Similarity between two probability distributions can be measured by the Kullback-Leibler divergence (KL divergence). Ackermann *et al*. (2010) develope an approximation algorithm for the k-medoids problem with respect to an arbitrary similarity measure, such as squared Euclidean distance, KL divergence, Mahalanobis distance, etc. Banerjee *et al*. theoretically analyze the k-means based on Bregman divergences which is a general case of KL divergence (Banerjee *et al*., 2005). Jiang *et al*. adopt the KL divergence to measure similarity between uncertain objects in both the continuous and discrete cases for clustering uncertain objects (Jiang *et al*., 2013).

Our work is closely related to cluster massive uncertain time series data based on the sketches. In this paper, we construct hash-compressed representations for storing and analysis the trends of uncertain time series. And based on the KL divergence and sketch *-metric, we present a similarity measurement method to calculate the distances between uncertain time series objects. In order to reduce the computation, we construct core-sets and present an initial cluster centers select algorithm to optimize clustering quality.

## HILBERT SFC BASED TREND SKETCHES

In many applications, such as sensors, traffic, stock, etc., there may be many observation values for a item in an uncertain time series. The uncertainty of the observation values increases the complexity for analyzing and clustering these massive time series objects. Based on the trend extracting, we construct compressed sketch to improve the computation and efficiency for clustering massive uncertain time series data.

**Uncertain time series:** Let us=$<r_1, r_2, \ldots, r_t>$ be an uncertain time series, represented by a sequence of time stamped values, where $r_i$ represent the items at time i. The $r_i$ can be represented by a vector $< x^1_i, x^2_i, \ldots, x^k_i >$, where $x^j_i$ is an element value at time i and k is the number (k-dimension) of possible elements of $r_i$.

For an uncertain time series, the variety probabilities of values at a time lead to produce the diversity of possible sequences. For a one-dimensional uncertain time series $us_a$ of length $n_a$, let k be the sample size (k-dimension) of each items and $PS_a$ be the set of all possible k-length sequences that can be derived from the combination of items according the order of the time dimension, $us_a = \{r_1, r_2, \ldots, r_n\} = \{< x^1_1, x^2_1, \ldots, x^k_1 >, \ldots, < x^1_n, x^2_n, \ldots, x^k_n >\}$ and the size of $PS_a$ is $| PS_a | = n_a^k$.

For massive uncertain time series, the one of the main aims focuses on splitting an uncertain time series into meaningful segmentation to obtain the frequency trend of uncertain time series.

**Hilbert SFC and Hilbert keys:** For an uncertain time series US = $\{r_1, r_2, \ldots\}$, we choose n min wise sequence permutations, represented as $\{\pi_1, \pi_2, \ldots, \pi_n\}$. The problem with this approach is that it is not feasible to feasible to permute large sequence set. Drawing random sequence permutations is time consuming and practically inefficient. Fortunately, it is possible to simulate the effect of a random sequence permutation by using universal hashing functions. The standard universal hash function is defined as:

$$H_i(x) = ( (a_i * x + b_i) \bmod p ) \bmod n, \; i=1,2,\ldots,n \quad (1)$$

where p>n is a prime number and n is the number of hash functions. The parameters $a_i$ and $b_i$ are chosen uniformly

from {0,1,...,p-1}. Instead of picking all possible sequence permutations, we pick n universal hashing functions $\{H_1, H_2, \ldots, H_i, \ldots, H_n\}$ to approximate the random sequence permutations.

For an uncertain time series $us_a = \{r_1, r_2, \ldots, r_n\} = \{< x^1_1, x^2_1, \ldots, x^k_1 >, \ldots, < x^1_n, x^2_n \ldots, x^k_n >\}$, a random sequence permutations set $PS_n(US)$ can be chosen and constructed from possible values for each item $r_i$, where n is the number of sequence permutations:

$$\pi_i(US) = < x_1^{Hi(1)}, x_2^{Hi(2)}, \ldots, x_j^{Hi(j)}, \ldots, x_n^{Hi(n)} > \quad (2)$$

For a 3-length and 3-dimensional uncertain time series $us_a$ (3) = $\{r_1, r_2, r_3\} = \{<x_1^1, x_1^2, x_1^3>, <x_2^1, x_2^2, x_2^3>, <x_3^1, x_3^2, x_3^3>\}$, let the number of permutations n = 4 and based on the standard universal hash function $\{H_1, H_2, H_3, H_4, H_5\}$, we can gain a random permutations set $PS_4(us_a(3)) = \{\pi_1, \pi_2, \pi_3, \pi_4\} = \{(x_1^1 x_2^2 x_3^1), (x_1^2 x_2^3 x_3^1), (x_1^1 x_2^3 x_3^2), (x_1^3 x_2^1 x_3^3)\}$.

For each permutation in the permutations set $\{\pi_1, \pi_2, \ldots, \pi_n\}$ of an uncertain time series, we use Hilbert Space-Filling Curve (SFC) (Lawder, 2000) to map a n-dimensional permutation to a one- dimensional value (Hilbert key).

**Definition 1:** (Hilbert SFC) Hilbert Space-Filling Curve (SFC) maps a d-dimensional attribute space to a one-dimensional space, regard as Hb: $R^d \rightarrow I$. If a point $p \in R^d$, then $Hb(p) \in I$.

Hilbert Space-Filling curve (SFC) maps a d-dimensional values space to a one-dimensional identifier space. Figure 1 shows a permutation $\pi_j$ with k values $\{x_1, x_2, \ldots, x_d\}$ being mapping onto an m-bit key where each value is represented as an (m/d)-bit value. In uncertain time series, each value $x_i$ represents a possible value and each (m/d)-bit dimension value represents a possible value. A Hilbert SFC Hb ($\pi_j$) maps each value $x_i$ of a sequence $\pi_j$ according its position i in the sequence to the i-th (m/d)-bit of the m-bit key in a one-dimensional identifier space.

A Hilbert SFC maps the coordinate point to an m-bit key in a one-dimensional identifier space. Figure 2 illustrates the mapping of $(00_2, 00_2)$ and $(11_2, 11_2)$ to $0000_2$ and $1010_2$, respectively. One criterion to evaluate the performance of SFC is the clustering property: Two one-dimensional values that are close in the one-dimensional space represent two d-dimensional points that are close in the d-dimensional space. Jagadish (1990) has shown that for multi-dimensional indexing and range query, Hilbert SFC minimizes the number of clusters, compared to other types of SFC.1.

Based on the Hilbert SFC, we represent the trends information and permutations of an uncertain time series

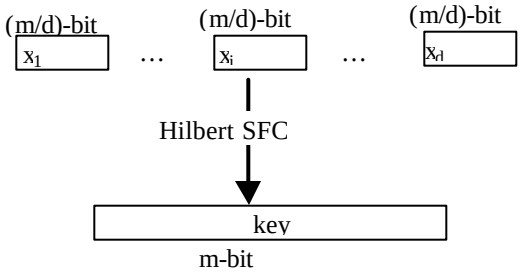


Fig. 1: Sequence values and key

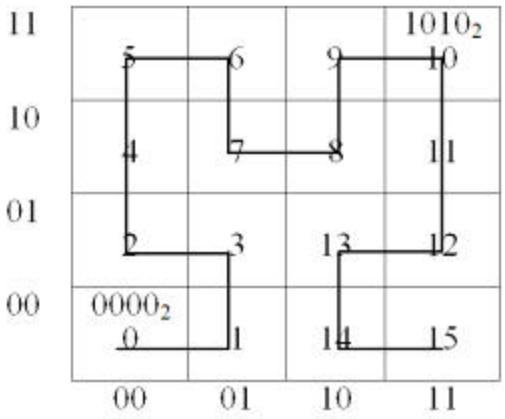


Fig. 2: Hilbert SFC on Two-Dimensional Space (m =4-bit)

with the Hilbert m-bit keys. For example, from the random permutations set $PS_4(us_a(3)) = \{\pi_1, \pi_2, \pi_3, \pi_4\} = \{(x_1^1 x_2^2 x_3^1), (x_1^2 x_2^3 x_3^1), (x_1^1 x_2^3 x_3^2), (x_1^3 x_2^1 x_3^3)\}$, we can get the Hilbert m-bit keys $HK(PS(us_a(3))) = \{Hb(|Đ_1), Hb(|Đ_2), Hb(|Đ_3), Hb(|Đ_4)\} = \{(Hb(x_1^1 x_2^2 x_3^1)), (Hb(x_1^2 x_2^3 x_3^1), (Hb(x_1^1 x_2^3 x_3^2)), (Hb(x_1^3 x_2^1 x_3^3)\} = \{(\text{"310"}), (\text{"432"}), (\text{"563"}), (\text{"795"})\}$.

**Hilbert key based trend sketches:** For the massive time series, the length of time series may be very large. For mining massive time series data efficiently, it is necessarily to segment the time series data and extract represent sequences. Based on segmentation, we extract the different length of Hilbert keys to gain the most common sequence fragments of uncertain time series.

**Definition 2:** (Trend Extension Degree) A trend extension degree of a sequence fragment te (sf) is the ratio of the frequency of all its subsequences with the frequency of the sequence sf in an uncertain time series us:

$$te(sf) = f(Sub(sf)) / f(sf) \quad (3)$$

where f (sf) is the occurrence frequency of sf in the uncertain time series us and Sub (sf) is the all possible subsequences of sequence fragment sf.

**Definition 3:** (Represent Fragment) A represent fragment rf is the sequence fragment that the trend extension degree te(rf)>θ in an uncertain time series us. Let θ = 0.3 in this paper and θ can be defined by user.

For massive time series data, the number of sequences and fragments may be huge. For counting the frequency efficiently, we adopt sketches to store the frequency of various sequence fragments. Since the massive data size will increase the computational and storage cost, Sketch is an efficient technique for compressed storing data size.

Sketch based approaches are designed for enumeration of different kinds of frequency statistics of data sets. A commonly-used sketch is the count-min method. The count-min sketch use w = [ln(1/δ)] pairwise independent hash functions, each hash function maps data into uniformly random integer in the range h = [0, e/ε], where e is the base of the natural logarithm. The data structure itself consists of a two dimensional array with w•h cells with a length of h and width of w. Each hash function corresponds to one of w 1-dimensional arrays with h cells each. In standard applications of the count-min sketch, the hash functions are used in order to update the counts of the different cells in this 2-dimensional data structure.

Based on the count min sketch, we construct the extended trends based sketch T-sketch to store massive uncertain time series data in compression way.

**Definition 4:** (Trend Sketch)A Trend sketch $TS_i$ includes a two-dimensional matrix $TS_i[w, c]$ ($w \ll N$, $c$), $c = e/\varepsilon$ is the maximum value of hash value range and $w$ is the number of hash functions hr[w]. The range of the k-length Hilbert keys are denoted by RK, such as {"432", "563", …}. Let $hr_i$ be the i-th hash function in hr[w]: RK→{0,…,c} be a hash function that hash a Hilbert key to the i-th row number and store the frequency at the $hr_i$ column. The initial value of each element in the sequential sketch is 0. For each Hilbert key hk in the length Hilbert keys set, we count the frequency value of hk in the time series data to SK[i, hr[i](sp)]:

$$TS[i, hr_i(hk)] = TS[i, hr_i(hk)] + 1 \quad (4)$$

Figure 3 shows the update process of the sequential sketch.

We use a set of trend sketches TSS= {$TS_1$, $TS_2$, …, $TS_m$}. We show an example of m trend sketches in Fig. 4.

For granules in an uncertain time series, if any sequence has j position, the sequence and its possibility will be store in the j-th sequential sketch $SK_j$.

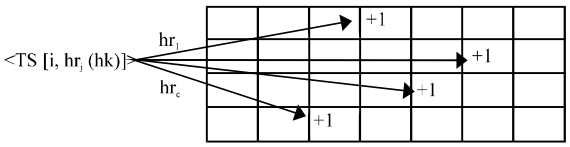


Fig. 3: Update process of the sequential sketch

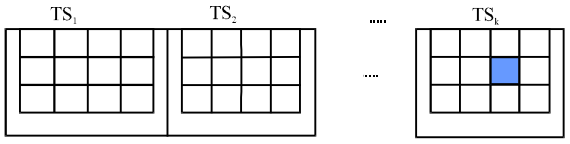


Fig. 4: k trends sketches

For massive time series data, the entire series data are divided into some fixed-length sequences. The fixed length is wl and $wl = 2^n$. In each fixed-length sequences of an uncertain time series, we intercept $\{2^1, 2^2, 2^3, \ldots, 2^k\}$ length subsequences as sequence fragments to gain different length Hilbert keys. For counting the trend extension degree of each fragment, we count each length sequences through different trend sketches.

For a wl-length sequence, we first gain the $2^1$ and $2^2$ length subsequences and calculate the corresponding two Hilbert keys sets. Then, we store the $2^1$ length Hilbert keys sets into the trend sketch $TS_1$. And for a $2^2$ length subsequences, before we add its frequency to the trend sketch $TS_1$, we calculate its trend extension degree according the frequency of its $2^1$ length subsequences in the trend sketch $TS_1$. If the $2^2$ length subsequence is a represent fragment, we add its frequency to the trend sketch $TS_2$; otherwise, we reduce the frequency of all its $2^1$ length subsequences in the trend sketch $TS_1$. Then, as for $2^i$ length subsequences, wither the $2^i$ length subsequences can be store in trend sketch $TS_i$ is based on the frequency of its $2^{i-1}$ length subsequences in the trend sketch $TS_{i-1}$. We handle all the $\{2^1, 2^2, 2^3, \ldots, 2^k\}$ length subsequences according the trend extension degree.

For an uncertain time series, we use k trend sketches to store its different length represent fragments. For any two uncertain time series, we calculate the similarity according the trend sketches.

## TREND SKETCHES BASED CLUSTERING

**Similarity:** The similarity between two uncertain time series can be defined as the Jaccard similarity between two corresponding sets of represent fragments. Since the represent fragments of uncertain time series are stored in the trend sketches, the computation of similarity should

be designed based on the trend sketches. We use the Sketch *-metric (Anceaume and Busnel, 2012) and the KL divergence (Jiang *et al*., 2013) to quantify the similarity between two uncertain time series $us_a$ and $us_b$.

**Definition 5:** (Kullback-Leibler divergence) The KL divergence is a robust metric for measuring the statistical difference between two data time series. Given p and q two distributions in discrete domain Ω with a finite number of values, the Kullback-Leibler divergence between p and q is then defined as:

$$KD(p \| q) = \sum_{i \in \Omega} p_i \log \frac{p_i}{q_i} \quad (5)$$

**Definition 6:** (Sketch-metric) Let p and q be any two Ω- point distributions. Given a precision parameter k and any generalized metric $\phi$ on the set of all Ù-point distributions, there exists a Sketch-metric $\phi'_k$ defined as follows:

$$\varphi'_k = \max_{\rho \in P_k(\Omega)} (\varphi(\hat{p}_\rho \| \hat{q}_\rho)), \quad \forall a \in \rho, \hat{p}_\rho(a) = \sum_{i \in a} p(i) \quad (6)$$

where, $P_k(\Omega)$ is the set of all partitions of an Ù-element set into exactly k nonempty and mutually exclusive cells.

As the sketch *-metric algorithm defined in (Anceaume and Busnel, 2012), each line of the trend sketch corresponds to a $\rho_i$ partition, $1 \le i \le c$. Thus the similarity between two j-length trend sketches is the maximal value over all the c partitions $\rho_i$ of the distance metric. We apply the KL divergence as the distance metric to the i-th lines of the two j-length trend sketches $TS^a_1$ and $TS^b_2$, $1 \le i \le c$. We treat each line TS[i] in a trend sketch TS as distributions of a partition $\tilde{n}_i$ and the count frequency of each column in this line as a distribution value of each column.

The similarity of two j-length trend sketches can be defined as:

$$SD_j(p \| q) = \max_{\rho \in TS_j} (KD(\hat{p}_\rho \| \hat{q}_\rho)) \quad (7)$$

where, $\hat{p}_\rho = TS_j[i]$, $1 \le i \le c$ and KD is the KL divergence defined in Eq. 1.

The similarity of two uncertain time series $us_a$ and $us_b$ can be defined as the minimal distance of corresponding length trend sketches:

$$D(us_a \| us_b) = \min_{0<j<k} (SD_j(\hat{p}_\rho \| \hat{q}_\rho)) \quad (8)$$

Based on the KL divergence and the sketch *-metric, the similarity of two uncertain time series is measured according the trend sketches.

**Construction of core sets:** UK-means is a known cluster algorithm based on K-means and is used on uncertain time series. According our analysis, the time complexity of UK-means is mainly due to the large number calculations of expected distance. Therefore, we use a partitions policy to reduce the cost of calculations by divide time series into different core sets.

**Definition 7:** (Core set) A core set *cs* includes seven parts {id, num, o_set, min_sk, max_sk, avg_sk, avg_dist, rep_o}, where the id is the identity of the core set; the num is the number of uncertain time series objects in the core set; the o_set represents all uncertain time series objects in the core set; the min_sk is a set of k trend sketches that each trend sketches store the minimal frequency of each element among all the trend sketches of the core set objects; the max_sk is a set of k trend sketches that each trend sketches store the maximal frequency of each element among all the trend sketches of the core set objects; the avg_sk is a set of k trend sketches that each trend sketches store the average frequency of each element among all the trend sketches of the core set objects; the avg_dist is the average similarity among the all core set objects; the rep_o is the represent object in core set that its distance to the avg_sk is minimal.

To construct core-sets, we should partition all N uncertain time series objects into different m core-sets and make the objects in same core-set as similar as possible. The m is the number of core-set and $m=N/2^k$, where N is the total number of the uncertain time series objects.

Based on the trend sketches, we also partition uncertain time series objects into different core sets. First, we random choose m objects in entire time series data and set each object as represent object rep_o of a new core set. We also set the trend sketches of the first object as the {min_sk, max_sk, avg_sk} of the core set. Then, we randomy select an object $o_i$ and calculate the average distance according the similarity between the $o_i$ with all the other m-1 objects and set the average distance as the avg_dist of each core set. Then, for other objects $o_j$, we calculate the distance between the trend sketches $TSS(o_j)$ of the object and the {min_sk, max_sk, avg_sk} of a core set. Among the three distances $D(TSS(o_j)$, min_sk), $D(TSS(o_j)$, max_sk), $D(TSS(o_j)$, avg_sk), any of them are smaller than σ*avg_dist we add the object $o_j$ to the core set and update the core set information. The σ = 0.9

represents the acceptable range of the distances and it can be defined by user.

The method of core-sets construction is shown as following:

```
Method of core-sets construction
Input:
 N trend sketches TSS of N uncertain time series objects
m: the number of core-sets
| Ô: the acceptable range of the distances
Output:
 m core-sets
Randomly select m uncertain time series objects ¡ úOm ;
Randomly select an object o in Om and its TSS_o ;
For an object p in Om and its trend sketches TSS_p
 Avg_Distance=Average(D(TSS_o,TSS_p))
End For
Set each object in Om as the rep_o of each m core set';
Set the Avg_Distance as the avg_dist of all core sets
For each other object q and its Avg_q
    For each core-set cs_i
 Get the min_sk , max_sk, avg_sk, avg_dist of cs_i;
 Td_min_q=D(TSS(q), min_sk) ;
      Td_max_q=D(TSS(q), max_sk);
      Td_avg_q=D(TSS(q), avg_sk);
      Avg_q =min(Td_min_q, Td_max_q, Td_avg_q);
If Avg_q <σ* avg_dist
        Add the object q to the core-set cs_i //partition
If Td_avg_q< D(TSS(rep_o), avg_sk)
rep_o=q; //set new rep_o
    End For
 if no core-set to add q
 create a new core-set cs_m+1 ;
 m++;
 Add q to cs_m ;
  End if
End For
```

We construct m core-sets to improve the efficient of clustering. For massive objects, the average number of objects in a core-set $|Cs_i|$ = 200 k will get much better clustering results according (Xu and Li, 2008).

**Initial cluster centers selection:** The clustering results of UK-means largely depend on the selection of initial cluster centers. If the initial cluster centers are selected unreasonably, the clustering result is likely to be local minima.

To choose the initial cluster centers, we design a selection policy based on a max-min cluster distance algorithm to calculate distances among the objects in the core-set. The main idea of max-min cluster distance algorithm is to select maximum distance among the other represent objects of core-sets and minimal distance among other cluster centers.

The max-min cluster distance based initial cluster centers selection algorithm is shown as following:

```
Algorithm of initial cluster centers selection
Input:
 m core-sets CS
 α: The threshold value
Output:
 a set of initial clusters centers IC
Randomly select a represent object ro_1 of a core set in CS;
Set ro_1 to the first initial clusters center ic_1 of IC
For each object ro_i in the set RO of represent objects
 Max_object((SD(ro_i, ic_1)) ¡ ú ic_2
End For
Let SD(ic_1 ,ic_2) ¡ ú an expected initial distance Eid_1
j=2
while(true)
  For each other objects o_i in RO
Min_object(SD(ic_1, o_i),.., SD(ic_j, o_i)) ¡ úan object set MS
    End For
     mk=|MS|
     For each other objects mo_i in MS
       if(Max(SD(ic_1, mo_i)…SD(ic_j, mo_i))>á*avg(Eid_1... Eid_j-1))
set mo_i ¡ ú ic_1
j++
Eid_j= SD(ic_j-1 ,ic_j)
Else
mk--
end if
End For
  if(mk==0) return //no object meet the test conditions
End While
```

First, the initial cluster centers selection algorithm randomly selected an object $o_1$ from m represent objects and set $o_1$ as the first initial cluster center $ic_1$. Then select a represent object $o_2$ which has the maximum distance with $ic_1$ as the second initial cluster center $ic_2$. Set the distance between $ic_1$ and $ic_2$ as an expected initial distance $Eid_1$. For other represents object $o_i$, if max (min (SD ($o_i$, $ic_1$), SD ($o_i$, $ic_2$), …, SD($o_i$, $ic_j$))> $\alpha$*average($Eid_1$, $Eid_2$,…,$Eid_{j-1}$), select $o_i$ as $ic_j$; j++; $Eid_j$= SD($ic_{j-1}$, $ic_j$). If no object meets the test condition, then finish the algorithm.

The initial cluster centers selection algorithm selects centers based on the test conditions $\alpha$*avg ($Eid_1$...$Eid_j$). The expected initial distance $Eid_j$ is the distance SD ($ic_{j-1}$, $ic_j$) between the last two initial cluster centers $ic_{j-1}$ and $ic_j$. The threshold value á is very important for the number of initial cluster centers. The smaller value is the test parameter $\alpha$, the more initial cluster centers will generate. The threshold value á should be choose properly.

**UTSClu clustering algorithm:** Based on the selection of initial cluster centers and UK-means algorithm, we build our UTSClu algorithm for clustering uncertain time series data.

The UTSClu algorithm includes four main phases:

- Construct trend sketches $TSS_p$ for an uncertain time series object p. Each sketch $TS_i$ in the trend sketches stores Hilbert key of i-length sequence fragments

- Construct core-sets. Based on the core-set construction policy, partition objects to different core-sets and select represent object for each core-set
- Select initial cluster centers. Based on max-min distances, UTSClu selects initial cluster centers from represent objects
- Cluster the uncertain time series. Based on initial cluster centers, UTSClu use each initial cluster center to represent a cluster. For other uncertain time series objects, UTSClu calculate the distance between an object and a cluster center. Then, add the object to the nearest cluster and recalculate the cluster center of the cluster

The (1) to (2) phases are to construct Hilbert key based trend sketches for compressing storage size. The (3) phase is to select initial cluster centers for improving clustering quality. In the (4) phase, the UTSClu algorithm clusters the uncertain time series objects based on UK-means.

## EXPERIMENT

This section shows the results from our experiment to validate the accuracy and efficiency of our proposed clustering algorithm.

**Evaluation standard:** We will show the accuracy of our algorithm based on the evaluation standard of RandIndex (Somasundaram and Nedunchezhian, 2011). For uncertain data set D (includes N objects), let $T=\{T_1, T_2, \ldots, T_k\}$ represent the original clusters and $C=\{C_1, C_2, \ldots, C_m\}$ be the clusters produced by a clustering algorithm. Let a represent an object that is in a cluster of C and in a cluster of T either. Let b represent an object that is in a cluster of C but NOT in any cluster of T:

$$SRAND = \frac{a+b}{n(n-1)/2} \quad (9)$$

The SRAND represent the degree of matching between T and C. The greater of the SRAND value means the better of the clustering.

**Experimental results:** We compare our UTSClu algorithm to UK-means by using the Census 1990, Tower and Covertype data sets. For each record x of an obiect $o_i$ in a data set, we add 2 possible data $\{x^1, x^2\}$ and probability value $\{p^0, p^1, p^2\}$, $\{p^0, p^1, p^2\}$ means the probability value of $\{x, x^1, x^2\}$. According the following formulas, we use two parameters $\alpha = 0.0.1$ and $\beta = 0.05$ to calculate $x^1$ and $x^2$:

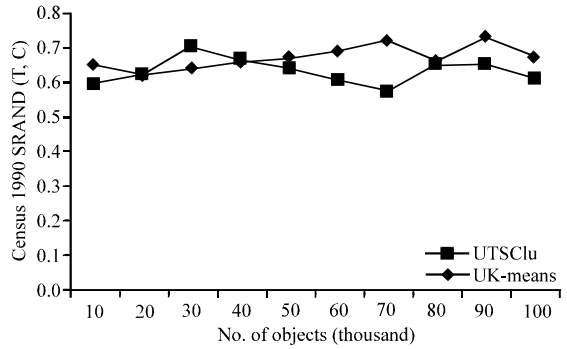


Fig. 5: Clustering quality comparison on Census 1990

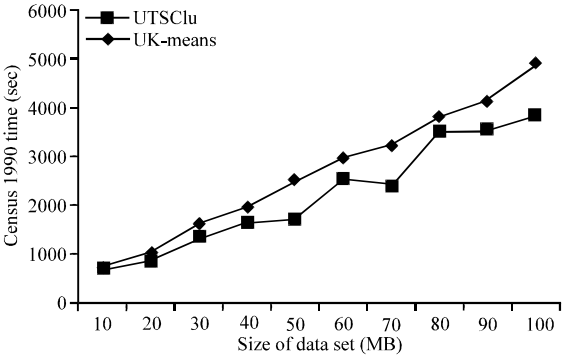


Fig. 6: Time consuming comparison

$$x^1 = x*(1+y*\alpha),$$
$$x^2 = x*(1+y*\alpha+\beta) \quad (10)$$

where, y is a random value in [-1, 1].

The probability value $\{p^0, p^1, p^2\}$ can be assigned following the normal distribution from the range (0, 1] to each item in the data set, according the difference among $\{x, x^1, x^2\}$.

Figure 5 shows the comparison of clustering quality between UTSClu and UK-means with the increase in the size of the Census 1990 data set. In Fig. 5, the SRAND value of UTSClu is lower than that of UK-means. Based on the Hilbert keys and trend sketches, the UTSClu can only calculate the approximate similarity, so the quality of clustering is not high. But for massive time series data, this quality is still acceptable and the UTSClu also increase the calculation of initial cluster centers selection to improve the clustering quality.

**Clustering quality comparison on Census 1990:** Figure 6 shows the comparison of time consuming between UTSClu and UK-means with the increase in the size of the Census 1990 data set. In figure 6, the time consuming of UTSClu is lower than that of UK-means.

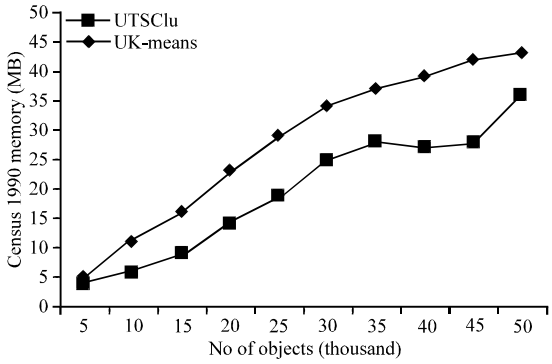


Fig. 7: Memory space comparison

Based on the Hilbert keys and trend sketches, the UTSClu store the time series compactly and reduce the calculation cost significantly. And the construction of the core set also reduces the similarity calculation among the massive uncertain time series objects.

**Time consuming comparison:** Figure 7 shows the comparison of memory space consuming between UTSClu and UK-means with the increase in the size of the Census 1990 set. In Fig. 7, the memory space consuming of UTSClu is smaller than that of UK-means. Because the UTSClu adopt Hilbert keys and trend sketches based method to compress storage space and reduce the memory consumption.

**Memory space comparison:** Experiment results show that the UTSClu algorithm can effectively reduce the time and memory consumption in clustering uncertain data time series with acceptable clustering quality. The UTSClu benefits from using compressed storage space and initial cluster centers.

## CONCLUSION

This study has addressed a trend sketches and sketch metric method for clustering uncertain massive uncertain time series. First, for reducing memory consumption, based on the Hilbert SFC, we construct trends sketches to compress massive uncertain time series data. And based on divergence and sketch metric, a sketch metric is given to measure the similarity between two uncertain time series objects. Then, based on the sketch metric, a core-set construction strategy is given to select represent objects to reduce the objects size of calculation. And then, based on the Max-min cluster distance algorithm, an algorithm for selecting the initial cluster centers is proposed. And based on the initial cluster centers, our UTSClu algorithm for clustering uncertain data is given. Experiment results show that the UTSClu algorithm can cluster uncertain data time series with lower time and memory consumption. In future work, we plan to design distributed sketches and clustering algorithm for distributed uncertain data time series.

## ACKNOWLEDGMENTS

The study is supported by the Fundamental Research Funds for the Central Universities.

## REFERENCES

Ackermann, M.R., J. Blomer and C. Sohler, 2010. Clustering for metric and nonmetric distance measures. ACM Trans. Algorithms, Vol. 6, No. 4. 10.1145/1824777.1824779

Ackermann, M.R., M. Martens, C. Raupach, K. Swierkot, C. Lammersen and C. Sohler, 2012. StreamKM++: A clustering algorithm for data streams. J. Exp. Algorithmics, Vol. 17. 10.1145/2133803.2184450

Aggarwal, C., 2009. A framework for clustering massive-domain data streams. Proceedings of the 25th International Conference on Data Engineering, March 29-April 2, 2009, Shanghai, China, pp: 102-113.

Anceaume, E. and Y. Busnel, 2012. Sketch ?-metric: Comparing data streams via sketching. Technical Report, CIDER-IRISA. http://hal.inria.fr/docs/00/72/12/11/PDF/AB13-INFOCOM-RR.pdf

Ankerst, M., M.M. Breunig, H.P. Kriegel and J. Sander, 1999. Optics: Ordering points to identify the clustering structure. ACM SIGMOD Rec., 28: 49-60.

Banerjee, A., S. Merugu, I.S. Dhillon and I. Ghosh, 2005. Clustering with Bregman divergences. J. Mach. Learn. Res., 6: 1705-1749.

Cormode, G. and S. Muthukrishnan, 2005. An improved data-stream summary: The count-min sketch and its applications. J. Algorithms, 55: 58-75.

Ester, M., H.P. Kriegel, J. Sander and X. Xu, 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, August 2-4, 1996, Portland, pp: 226-231.

Jagadish, H.V., 1990. Linear clustering of objects with multiple attributes. Proceedings of the ACM SIGMOD International Conference on Management of Data, May 23-26, 1990, Atlantic City, NJ., USA., pp: 332-342.

Jiang, B., J. Pei, Y.F. Tao and X.M. Lin, 2013. Clustering uncertain data based on probability distribution similarity. IEEE Trans. Knowledge Data Eng., 25: 751-763.

Kriegel, H.P. and M. Pfeifle, 2005. Density-based clustering of uncertain data. Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, August 21-24, 2005, Chicago, IL., USA., pp: 672-677.

Lawder, J.K., 2000. Calculation of mappings between one and *n*-dimensional values using the Hilbert space-filling curve. Technical Report No. JL1/00, August 15, 2000, University of London, UK. http://www. dcs.bbk.ac.uk/TriStarp/pubs/JL1_00.pdf

Liu, Y., L.F. Zhang and Y. Guan, 2010. Sketch-based streaming PCA algorithm for network-wide traffic anomaly detection. Processing of the IEEE 30th International Conference on Distributed Computing Systems, June 21-25, 2010, Genova, Italy, pp: 807-816.

Manerikar, N. and T. Palpanas, 2008. Frequent items in streaming data: An experimental evaluation of the state-of-the-art. Technical Report DISI-08-017, University of Trento, Trento, Italy, March 2008. http://disi.unitn.it/~themis/frequentitems/dke09.pdf

Ngai, W.K., B. Kao, C.K. Chui, R. Cheng, M. Chau and K.Y. Yip, 2006. Efficient clustering of uncertain data. Proceedings of the 6th International Conference on Data Mining, December 18-22, 2006, Hong Kong, pp: 436-445.

Nie, Y., R. Cocci, Z. Cao, Y.L. Diao and P. Shenoy, 2012. SPIRE: Efficient data inference and compression over RFID streams. IEEE Trans. Knowledge Data Eng., 24: 141-155.

Papapetrou, O., M. Garofalakis and A. Deligiannakis, 2010. Sketch-based querying of distributed sliding-window data streams. Proc. VLDB Endowment, 5: 992-1003.

Somasundaram, R.S. and R. Nedunchezhian, 2011. Evaluation of three simple imputation methods for enhancing preprocessing of data with missing values. Int. J. Comput. Appl., 21: 14-19.

Tran, T.T.L., L.P. Peng, B.D. Li, Y.L. Diao and A.N. Liu, 2010. PODS: A new model and processing algorithms for uncertain data streams. Proceedings of the International Conference on Management of Data, June 6-11, 2010, Indianapolis, IN., USA., pp: 159-170.

Wang, X.M. and D.B. Yuan, 2012. A query verification scheme for dynamic outsourced databases. J. Comput., 7: 156-160.

Xu, H.J. and G.H. Li, 2008. Density-based probabilistic clustering of uncertain data. Proceedings of the International Conference on Computer Science and Software Engineering, Volume 4, December 12-14, 2008, Wuhan, Hubei, China, pp: 474-477.