A Differential Geometry Perspective about Multiple Data Streams Preprocessing

Li Wen-Ping[1, 2], Yang Jing[1] and Zhang Jian-Pei[1]
[1]College of Computer Science and Technology, Harbin Engineering University, China
[2]College of Mathematics Physics and Information Engineering, Jiaxing University, China

Abstract: *In the Multiple Data Streams (MDS) environment, data sources generate data with no end in sight. Because of the difference of data sources, transaction numbers of MDS are not always equal to each other during a same period. Preprocessing MDS to obtain same number of samples for each stream is an essential step for lots of mining tasks. All existing preprocessing methods assume that data arrive simultaneously. However, this assumption may not be true in many real environments due to multiple data sources and different ways of data generating. This asynchronous issue is explored in this paper, by introducing the differential geometry as a trick. First, we establish a novel stream model called POLAR. The POLAR is an intrinsic surface spanned by time, probability and value. And then, we propose a preprocessing approach, called COPOLAR, to obtain same number of samples for each stream of MDS. COPOLAR first projects original observations onto POLAR; and then merges points with shortest geodesic distances along a geodesic on surface into mid-point on the same geodesic iteratively and incrementally until the number of points which we hope to obtain is met. Experimental results on synthetic and real data show that COPOLAR is effective in terms of maintaining characteristics of both statistics and vector.*

Keywords: *Data mining, MDS, data preprocessing, data stream model, differential geometry, geodesic.*

Received May 18, 2013; accepted March 19, 2014; published online December 3, 2014

1. Introduction

The processing and mining of data streams have received considerable attention in various communities due to several important applications, for instance network analysis [22], moving object tracking [14], wireless sensor networks [15], financial data analysis [19], etc., In all of the applications cited above, data sources generate data with no end in sight. How to add the highly dynamic nature of data streams to previous data mining technologies is highly concerned.

In the fields of data stream mining, Multiple Data Streams (MDS) have attracted more and more researcher's interests recently. Lots of works have been done about it, such as classification [24], correlation analysis [21, 23], clustering [3, 6, 17], etc., it is undeniable that many mining tasks require that the length of each stream from MDS is equal to each other. Consequently, mostly all methods above are based on an assumption that sampling frequency is consistent for each stream. However, the scholars in MDS fields have long-standing neglected a fact that the transaction numbers of all streams from MDS are not always equal to each other, and they display much differences generally during a same period in many time-varying circumstances. If samples are extracted from such streams with a consistent sampling frequency as usual, the distribution of samples is bound to distort the distribution of collection in most cases.

We illustrate dominating motivation with a significant instance coming from fields of commercial data analysis. Imagining in a supermarket, lots of cash registers are deployed. Nevertheless, plenty of difficulties are emerging from this scenario. The problems that we are focusing on are as follows:

- Inconsistency of transaction numbers. It is obvious that the number of transactions for each cash register is mostly impossible to be equal during a same period.
- In many cases, the transaction numbers among some registers vary enormously in different periods.
- Time-varying characteristics. The difference of transaction numbers between two given registers is evolving persistently over time.

In many cases, these issues bring difficulties to mining tasks. A pregnant task is clustering on MDS, for instance SPE-cluster [6]. In SPE-cluster, many definitions, such as the spectral component based e-lag-correlated similarity and the distance between two streams, require same length for each stream of MDS.

In order to overcome these problems above, a preprocessing method is proposed to extract same number samples for each stream from MDS in this paper. To do so, we establish a novel data stream model. Even with a widespread adoption, existing data stream models, such as the sliding window model, landmark model and time decaying model and so on, share following two drawbacks:

- They fail to touch upon the probability distribution of stream.

- The time property is mentioned qualitatively only, but couldn't be represented quantificationally among them.

We introduce Probability wOrld modeL for datA stReams (POLAR), a novel representation model of data stream. The basic idea of POLAR is that stream is restrained on (or mapped onto) a surface spanned by time, probability and stream value. Because of the introduction of time and probability, POLAR could overcome aforementioned shortcomings of traditional stream models. The unique characteristics of POLAR are described as follows:

- Association of the probability distributions. The probabilistic consideration plays a significant role in time-varying environment. Yet, as an element of stream model, the probability distribution of stream has not been carried into living stream models. Thus, we draw probability into the novel model POLAR and regard it as an inherent component of the model. In POLAR, the probability shall be involved in the solution of mining.
- Calculability of the time. There is a common view that the importance of stream might vary from time to time. So the time, as a basic component of stream, has irreplaceable position in data mining. Unfortunately, time property has been researched by a qualitative fashion rather than by a quantitative way in existing models. Thus, we introduce time into POLAR. In POLAR, time shall also be involved in calculation.

Based on POLAR, we further propose a preprocessing approach, called COllapsed Probability wOrld modeL for datA stReams (COPOLAR), to extract same number samples for each stream from MDS. To extract same number samples, COPOLAR merges the nearest points along a geodesic on the surface of POLAR into a point on the same geodesic iteratively and incrementally. COPOLAR satisfies following requirements:

- During a same period, it is acceptable for COPOLAR to hold remarkable difference of numbers of arriving points from different streams.
- The data preprocessing is an incremental updating procedure with arriving of stream points. It can merge points on a geodesic in one pass with a small amount of memory and processing time per time tick.
- The probability distribution of stream is introduced. The preprocessing of points within one probability interval may differ from within another.
- The time is introduced and viewed as a basic component of stream. The preprocessing of points within one time interval may differ from within another.

The outline of this paper is as follows. In section 2, we discuss related works. After explaining fundamentals of POLAR in section 3, we propose COPOLAR in section 4. To evaluate the effectiveness of proposed preprocessing approach, section 5 illustrates experimental results and discusses important issues that we have realized from the experiments. Finally, we conclude this paper in section 6.

2. Related Works

Data preprocessing is required in almost all knowledge discovery tasks such as data stream mining which has been attracting great interests. Different preprocessing techniques have different influences on some mining tasks. In fact, early research results have shown that data preprocessing has a significant impact on predictive accuracy of many classifiers such as decision trees, neural networks and support vector machines, etc. [8]. Although, there are plenty of publications in these research topics, we have not seen any method for that could specially draw-out same number samples from MDS. We sketch out related literatures.

Although, standard data preprocessing includes lots of steps, i.e., dataset creation, data cleaning, integration, feature construction, feature selection, reduction, discretization, etc., [9] summarization is the principal concern in streaming settings because huge volume of original data is infeasible to be stored [7]. Summarization, instead of original raw data, has played a significant role in data stream mining. Summarization methods mainly include wavelet, sketch, histogram and sampling. In the field of data stream mining, sampling is significant, which is even regarded as the only appropriate methods to applications of MDS [22]. Therefore, we mainly overview sampling methods about data streams.

Sampling techniques about data streams are reflected in fields of data stream management and mining [2, 12]. Byung-Hoon *et al.* [5] proposed a fast sampling scheme for maintaining a sample with replacement from an ever-growing data stream. Braverman *et al.* [4] researched on optimal sampling with or without replacement from fixed or timestamp-based windows based on random sampling under the sliding windows model. Granmo and Oommen [13] considered the problem of allocating limited sampling resources in a "real-time" manner, with explicit purpose of estimating multiple binomial proportions. Palmer and Faloutsos [16] proposed a biased sampling technique which can maintain original data density. Although, this method is suitable for outlier analysis, the sampling quality may be reduced due to existence of hash conflict of hashing function which is used to map bins in space to a linear ordering. Demaine *et al.* [10] introduced a biased sampling algorithm to mining frequent elements of data streams. Although, the error

is controlled, it does not support deleting operation. Aggarwal [1] proposed a biased sampling solution with the use of temporal bias functions based on reservoir sampling. It could be applied to query evaluation and classification. Moreover, biased sampling technique has also been put into use in online correlation analysis for data streams [21]. Perhaps most interesting, however is the combination of uniform sampling and biased sampling. Recently, Zhang *et al*. [22] proposed a novel method, called Polynomially Biased Reservoir Sampling (PBRS), to summarize unordered traffic data streams. This method reduces relative speed bias in certain extent.

3. Data Stream Model: POLAR

The novel data stream model POLAR is described in this section. POLAR is an intrinsic surface spanned by time, probability and value. Before discussing its structure, it is necessary to describe the stream. A data stream is denoted as $x_1, x_2, ..., x_i, ...$, where the subscript i represents arrival order of point x_i. If point x_i is real type, it is also called value of the point. Without loss of generality, this paper discusses real type chiefly.

The timeliness is an elementary and significant characteristic of streaming data. That said, the importance of content within which data arrives varies among different times. A striking example is the slide window model whose valid data for mining tasks covers only the nearest time horizons. It is therefore significant to draw into the time $t \in R$ as a component of POLAR, where R represents real field.

In like manner, it is also important to introduce the probability as a component of POLAR. Common sense indicates that different values play different roles in probability space. Therefore, preprocessing of point x_i with probability p_i should be different from other point x_j with probability p_j.

We establish a surface S in $\mathbb{E}^3$ to structure POLAR, where $\mathbb{E}^3$ represents the three-dimensional Euclid space. Specially needed reminders are adoption of differential geometry denotations in description hereinafter. More details of differential geometry language can be referred from [20] or other text-books about it.

The function of parametric surface S corresponding to POLAR is defined by the form of vector function as follow:

$$S: r(v, t) = (v, f(v, t), t) \quad (1)$$

Where v and t are stream value and time respectively, $f(\bullet, t)$ is Probability Density Function (PDF) of stream at time t. As a note, we just focus on continuous distribution in this paper.

Generally speaking, the PDF of stream changes at some particular times only. Thereby, we can always find a time interval (a, b) such that $f(v, t)$ is constant at any time $t \in (a, b)$. One of its implications is that surface $r(v, t)$ can be divided by time t into many small surfaces $r(v, t_1), r(v, t_2), ..., r(v, t_k)$. In each small surface $r(v, t_k)$, $f(\bullet, t)$ is constant for any time. Without loss of generality, this research focuses only on the assumption that the PDF of stream is invariant.

Accounting for the hypothesis above, Equation 1 can be rewritten by the form of ruled surface as follow:

$$S: r(v, t) = a(v) + t\,l(v) \quad (2)$$

Where $a(v) = r(v, t_0) = (v, f(v, t_0), t_0)$ and $l(v)$ are directrix and generatrix of the ruled surface respectively.

Since, each generatrix of the ruled surface is parallel, generatrix can be normalized to $l(v) = l_0 \neq 0$. So, $l'(v) = l_0' = 0$. We thereby obtain the regular condition of surface S from Equation 2 as follow:

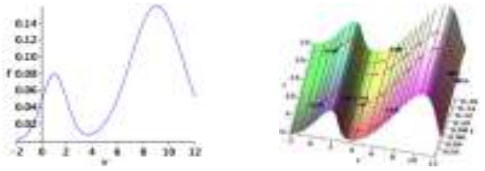
$$r_v \times r_t = [a'(v) + t\,l'(v)] \times l(v) = a'(v) \times l(v) = (1, \frac{\partial f}{\partial v}, 0) \times 1(v) \neq 0 \quad (3)$$

Where $r_v = \frac{\partial r}{\partial v}$ and $r_t = \frac{\partial r}{\partial t}$. Equation 3 demonstrates that surface s corresponding to POLAR described in Equation 2 is a regular parametric surface. Its regularity will be the foundation of follow-up to study about the preprocessing approach to streams, i.e., COPOLAR.

As an example, Figure 1 illustrates a surface, in that the PDF is Gaussian mixture distribution, i.e.

$$f(v, t) = f(v) = \frac{p}{\sqrt{2\pi}\sigma_1} exp(-\frac{(v-\mu_1)^2}{2\sigma_1^2}) + \frac{1-p}{\sqrt{2\pi}\sigma_2} exp(-\frac{(v-\mu_2)^2}{2\sigma_2^2}) \quad (4)$$

Where $p = 0.2, \mu_1 = 1, \sigma_1 = 1, \mu_2 = 9, \sigma_2 = 2.$



a) The PDF. b) Data stream model: POLAR.

Figure 1. Illustration of POLAR about gaussian mixture distribution.

When a new stream point x_i arrives, it will be mapped onto surface s described in Equation 2. To this end, we establish a mapping rule:

$$P: \mathbb{E}^2 \to \mathbb{E}^2$$
$$(x, i) \mapsto (v, t) = (M_v(x), M_t(i)) \quad (5)$$

Where x and i are value and arrival order of point x_i Functions $M_v(x)$ and $M_t(i)$ map original value x and arrival order i onto coordinate axis v and t of the parameter plane of surface s respectively. For instance, their straightforward forms are illustrated in Figure 1, where $v = M_v(x) = x$ and $t = M_t(i) = i$.

Undoubtedly, selection of concrete forms of functions $M_v(x)$ and $M_t(i)$ relies on special mining tasks. In many cases, stream value is generally required to be normalized into [0, 1]. Thereby function $M_v(x)$ could be selected from given normalized functions. Generally speaking, the gaussian normalization is a typical

representative thanks to its insensitiveness to the outlier, in which the form of function $M_v(x)$ is:

$$v = M_v(x) = (x - \mu)/3\sigma \tag{6}$$

Where μ and σ represent stream mean and variance respectively.

Nevertheless, unlike mapping function $M_v(x)$, selection of concrete forms of function $M_t(i)$ is usually difficult due to its dependence on concrete stream models. If it is expressed as follow:

$$M_t(i) = \begin{cases} n & if\ x_i\ is\ the\ nearest\ one \\ M_t(i-1) & otherwise \end{cases} \tag{7}$$

It can meet demands of the fixed-size sliding windows model [4] in which item arrives only one at a time and just only the most recent n items remain active, where n is the length of window. Analogously, there is another example which can meet demands of the timestamp-based sliding windows model [4] in which more than one item arrive at a single step and just only the last t steps remain active. The mapping values can be obtained as following procedure: Firstly set $M_t(i) \leftarrow M_t(i\text{-}1)+1$; secondly if x_i is the first point of the nearest step, then for each $k \le i$ set $M_t(k) \leftarrow M_t(k\text{-}1) - n_0$, where n_0 is the number of the oldest one step which expires just now.

Apparently, the point on surface s corresponding to point x_i of stream can be determined by Equation 2 in line with the map described in Equation 5.

So far, we establish a geometry in $\mathbb{E}^3$, which consists of a surface described in Equation 2 (even in Equation 1) and the points mapped from stream onto the surface in the light of map showed in Equation 5 We call the geometry POLAR. Stream preprocessing operations corresponding to COPOLAR, discussed immediately in next section, will be carried out on this geometry.

4. Preprocessing Data Streams

In this section, we will discuss the data stream preprocessing method COPOLAR. The primary mission of COPOLAR lies in obtaining same number samples from MDS to meet demands of some mining tasks. More precisely, we hope to seek a map $F: \{x_k^i\}_{k=1}^{n_i} \mapsto a\{\tilde{x}_k^i\}_{k=1}^{n}$ for each stream x^i of MDS x, i=1, 2, ..., p, where p is the number of streams in x, n_i is the number of points coming from x^i and n is the common number of points which we hope to obtain from all streams. Broadly speaking, n is a parameter given by users, but it is set to $n=\{n_1, n_2, ..., n_p\}$ below unless otherwise stated. Hereinafter, we denote p_k as the point on POLAR corresponding to x_k^i.

In the process of data stream preprocessing, two points p_k and p_l with shortest geodesic distance will be replaced by their mid-point along the geodesic passing through p_k and p_l on surface of POLAR. Such merger will be carried out iteratively until obtaining n points.

Three questions emerging in here are: What is the geodesic passing through any two points p_k and p_l; where is the position of their mid-point; how do we find out the shortest geodesic distance connecting given point p_k and other points? We will answer them in sections 4.1, 4.2 and 4.3 respectively and then we will propose an algorithm to portray the preprocessing steps in section 4.4.

4.1. Determination of Geodesic

As mentioned above, to merger two points p_k and p_l with shortest geodesic distance into their mid-point on the geodesic connecting p_k and p_l is the basic strategy of COPOLAR. Truly the top priority rests with the determination of the geodesic. In this subsection, we shall derive equation of the geodesic passing through any two points p_k and p_l.

As assumption that the PDF $f(v, t)= f(v, t_0)$ does not change with time, $f(v, t_0)$ is denoted as $f(v)$ and its derivative with respect to v as $f'(v)$ for simplicity. We further set $r(v, t)=a(v)+tl(v)=(v, f(v), 0)+t(0, 0, 1)$. By Equation 2, we get following results:

$$\begin{aligned} r_v &= (1, f'(v), 0) \\ r_t &= (0, 0, 1) \\ E &= r_v \bullet r_v = 1 + (f'(v))^2 \\ F &= r_v \bullet r_t = 0 \\ G &= r_t \bullet r_t = 1 \end{aligned} \tag{8}$$

Therefore, the first fundamental form of the surface described in Equation 2 is:

$$I = ds^2 = Edv^2 + 2Fdvdt + Gdt^2 = (1 + (f'(v))^2)dv^2 + dt^2 \tag{9}$$

Thanks to $F=r_v \bullet r_t=0$, the coordinate system (v, t) is orthogonal. So, differential equation of the geodesic could be expressed by the Liouville formula as follow:

$$\begin{cases} \dfrac{dv}{ds} = \dfrac{1}{2\sqrt{E}}\cos\theta = \dfrac{1}{2\sqrt{1 + (f'(v))^2}}\cos\theta \\ \dfrac{dt}{ds} = \dfrac{1}{2\sqrt{G}}\sin\theta = \dfrac{1}{2}\sin\theta \\ \dfrac{d\theta}{ds} = \dfrac{1}{2\sqrt{G}}\dfrac{\partial \ln E}{\partial t}\cos\theta - \dfrac{1}{2\sqrt{E}}\dfrac{\partial \ln G}{\partial v}\sin\theta = 0 \end{cases} \tag{10}$$

Where s is arc-length parameterization of the geodesic, and θ is angle between the geodesic and the v-curves. By Equation 10, we have:

$$\frac{dt}{dv} = \sqrt{1 + (f'(v))^2}\tan\theta,\ \theta = \theta_0 = const \tag{11}$$

Thus, equations of all geodesics can be obtained by solving the first equation above:

$$t = \int C\sqrt{1 + (f'(v))^2}dv + C_0 \tag{12}$$

Where C_0 and $C=\tan\theta_0$ are both constants.

For any two points $p_k=r(v_k, t_k)$ and $p_l=r(v_l, t_l)$, the question now is which curve within geodesics described in Equation 5 is the geodesic connecting given points. Apparently, we have:

$$\begin{cases} t = t_k + \int_{v_k}^{v} C\sqrt{1+(f'(h))^2}\,dh \\ t = t_l + \int_{v_l}^{v} C\sqrt{1+(f'(h))^2}\,dh \end{cases} \quad (13)$$

By solving it, we get:

$$C = (t_l - t_k) / \int_{v_k}^{v_l} \sqrt{1+(f'(h))^2}\,dh \quad (14)$$

Consequently, the geodesic equation passing through p_k and p_l could be written as:

$$t = t_k + C\int_{v_k}^{v} \sqrt{1+(f'(h))^2}\,dh \quad (15)$$

Its parametric form can be expressed as:

$$\bar{r}(v) = (v, f(v), t_k + C\int_{v_k}^{v} \sqrt{1+(f'(h))^2}\,dh) \quad (16)$$

Up to present moment, mission of this subsection has been fulfilled, i.e., deriving out equation of the geodesic passing through any two points. A follow-on effort to do is to locate their mid-point p_m.

4.2. Finding Mid-Point for Two Given Points

We are now in a position to locate the mid-point of two given points on POLAR, or rather, we will find out a point p_m on POLAR such that the geodesic distances between p_m and other two given points are equal. Admittedly, two questions emerging in here are: How do we calculate the geodesic distance between two given points; and where is the position of the mid-point?. By taking derivative of Equation 16, we have:

$$\bar{r}'(v) = (1, f'(v), C\sqrt{1+(f'(h))^2}) \neq 0 \quad (17)$$

Equation 17 tells us that v is a regularization parameter of the geodesic defined in Equation 16. In other words, the geodesic curve passing through points p_k and p_l is a regular curve. Thus, the oriented arc-length parameterization function starting from point $p_k = r(v_k, t_k) = \bar{r}(v_k)$ has following form:

$$s(v) - s(v_k) = \int_{v_k}^{v} |\bar{r}'(u)|\,du = C_1\int_{v_k}^{v} \sqrt{1+(f'(u))^2}\,du \quad (18)$$

Where $C_1 = \sqrt{1+C^2}$. In view of Equation 18, the length $s_{k,1}$ of the geodesic segment connecting points p_k and p_l can be written as follow:

$$s_{k,l} = |s(v_l) - s(v_k)| = \sqrt{1+1/C^2}\,|t_l - t_k| \quad (19)$$

If the mid-point between points p_k and p_l on the geodesic is $p_m = \bar{r}(v_m)$, the truth of following equation derives its guarantee:

$$|s(v_m) - s(v_k)| = s_{k,l}/2 \quad (20)$$

To simplify matters and without loss of generality, we could assume $t_k < t_m < t_l$ and $C > 0$. Under this premise, Equation 20 can be written equivalently as:

$$\int_{v_k}^{v_m} \sqrt{1+(f'(u))^2}\,du = \frac{t_l - t_k}{2C} \quad (21)$$

An unfortunate reality is that it will be difficult to obtain the explicit expression in terms of v_m because not only Equation 21 is nonlinear but also the PDF. $f(x)$ in its integrand may be varied in different application fields. But, we shouldn't be too despondent about it thanks to numerical technique as a candidate. A realistic approach is to employ the newton's method [11] to resolve it, which is also the solution in this paper in virtue of its fast convergence with fewer iteration steps. Of course, feasible alternatives don't rule out the use of any other numerical techniques such as bisection method, fixed point iteration, muller's method [11] and so on.

Following solving the basic solution about v_m from Equation 21, the value of t_m can be deduced by Equation 15. Consequently, the mid-point p_m is located.

Until now, mission of this subsection has been accomplished, i.e., locating the mid-point of two given points on POLAR. In next subsection, we will continue to explore other topics.

4.3. Calculating the Shortest Geodesic Distance

In this subsection, we will answer the third question mentioned in the beginning of current section, i.e., finding out the shortest geodesic distance connecting given point p_k and other points. And more to that point, we will find out the shortest geodesic distance s_{min} for p_k such that: $s_{min} = min\{s_{k,\,k+i}: j \in \mathbb{Z} - \{k\}\}$, where $\mathbb{Z}$ represents integer set (the same below).

To do so, the most straightforward way is to calculate all geodesic distances between p_k and other all points, and then to pick out the point with shortest geodesic distance from them. But, this strategy is obviously very time-consuming. As a matter of fact, we just need to get part of them rather than to calculate all.

To accomplish what we have outlined above, we need to make some hypotheses about $M_t(k)$ described in Equation 5. Because the design of pruning strategy may be various with different mapping $M_t(k)$ and it may take a far more complicated issue. To simplify matters, we assume:

$$0 \le M_t(k+1) - M_t(k) \le M_t(l+1) - M_t(l) \quad (22)$$

if and only if $k<1$. If the equality sign was founded, Equation 22 indicates that the importance of streams is equal along timeline; but when the inequality sign was set up, it indicates that the newcomers are important than before. In reality, this assumption is reasonable and not excessive.

For a given point p_k, Equation 22 implies also that the geodesic distances between p_k and other points will increase progressively while these points are far away from p_k along timeline. Therefore, an intuitive understanding is that we can reduce search spaces for finding the shortest geodesic distance connecting p_k. We present following theorem to state it.

- ***Theorem* 1**: Denote $B = \{s_{k,k+j}: |j| \le |z|, j \in \mathbb{Z} - \{0\}\}$, $A = \{s_{k,k+j}: j \in \mathbb{Z} - \{0\}\} - B$. Let $p_j = r(v_j, t_j)$, $s_{k,1}$ and s_{min} be any point in POLAR, the length of geodesic segment

connecting points p_k and p_l, and the shortest geodesic distance connecting given p_k respectively. There exists an integer $z\neq 0$ such that if $|t_{k+z}- t_k|\geq s_{min}$, then $s_{min}\in B$ and $s_{min}< s_{k,k+l}$ for any $s_{k,k+l}\in A$.

- ***Proof***: We need to prove two aspects: $s_{min}\in B$ and if $|j|>|z|$ then $s_{min} < s_{k,k+l}$. Denote p_{k+i} as the point with s_{min} connecting p_k and let $C_2 = 1/\sqrt{1+1/C^2_{k,k+i}}$. On the one hand, we have $s_{min}= s_{k,k+i}=|t_{k+i}- t_k|/C_2$. According to precondition $|t_{k+z}- t_k|\geq s_{min}$ we get $|t_{k+i}- t_k|\leq C_2|t_{k+z}- t_k|$. If $z>0$ and $i>0$, by Equation 5 and Equation 22, we have $t_{k+i}- t_{k+i-1}\leq t_{k+i}- t_k\leq t_{k+z}- t_{k+z-1}$-[$C_2(t_k- t_{k+z})+(t_{k+z}- t_{k+z-1})$]. Because $C_2(t_k- t_{k+z})+(t_{k+z}- t_{k+z-1})>0$, so $t_{k+i}- t_{k+i-1}< t_{k+z}- t_{k+z-1}$. By Equation 22, we have $|i|\leq |z|$ and when z and i take other values, we can get the same results. Thus, we conclude that $s_{min}\in B$. On the other hand, we assume that there is an element $s_{k,k+l}\in A$ such that $s_{min}\geq s_{k,k+l} =|t_{k+l}- t_k|$. Without loss of generality, let z, l, i >0, then we get $s_{min}\geq s_{k,k+l} =t_{k+l}- t_k$. Because $s_{k,k+l}\in A$, so $l>z$. By Equation 5 and Equation 22, we have $s_{min} \geq|t_{k+z}- t_k|$. Obviously, it contradicts with precondition $|t_{k+z}- t_k|\geq s_{min}$. Thereby, the assumption is invalid. Therefore, $s_{min}< s_{k,k+l}$.

Conclusions from Theorem 1 inspire us that we should stop searching if we find an integer $z\neq 0$ such that $|t_{k+z}- t_k|\geq s_{min}$, where s_{min} is the shortest geodesic distance connecting given p_k in current searching place.

In fact, we just need to search along a single direction to pick out the shortest geodesic distance connecting given p_k in series of points, ..., p_{k-1}, p_k, p_{k+1},... on POLAR. One optimal searching strategy may be to search towards lower indices because those points whose indices are greater than k have not arrived in real-time streams processing.

To this end, we need first to set up two variables or a data structure additionally for each point p_u. One is the shortest geodesic distance s^u_{min} connecting p_u and other points and the other one is index h, denoted as I_u, of point p_h which is the corresponding point in term of s^u_{min}. In other words, the geodesic distance $s_{u,h}$ between p_u and p_h equals to s^u_{min}. Initial values of s^u_{min} and I_u may be set to 0 and u respectively for each new arriving point p_u.

When a new point p_k arrives, we calculate the geodesic distances between p_k and other points whose indices are less than k until t_k-$t_{k-|z|}\geq s^k_{min}$. During calculation, two missions should be taken. On the one hand, if a new distance $s_{k,h} < s^k_{min}$ is found out, then let $s^k_{min} = s_{k,h}$ and $I_k=h$ for given point p_k. On the other hand, for another point p_h which has participated in calculation when searching for s^k_{min}, if $s^k_{min} < s^h_{min}$ is met, then update $s^h_{min}= s_{h,k}$ and $I_h=k$. We describe this procedure in Algorithm 1.

Algorithm 1: Get distance for point.

Input: *The given point* p_k.
Output: *The shortest geodesic distance connecting* p_k *and other points whose indices are less than k.*

// $s_{k,h}$ *is the geodesic distance between* p_k *and* p_h *described by Equation 19.*
// t_k *is the time value of point* p_k *on the POLAR.*
// I_k *is the shortest geodesic distance connecting* p_k *and other points.*
$h = k-2, s^k_{min} = s_{k,k-1}$;
While $t_k - t_h < s^k_{min}$ *and* $h>0$ *do*
  Calculate geodesic distance $s_{k,h}$ *by Equation 19;*
  If $s_{k,h} < s^k_{min}$ *then*
    $s^k_{min} = s_{k,h}, I_k = h$;
  End if
  If $s^k_{min} < s^h_{min}$ *then*
    $s^h_{min} = s^k_{min}, I_k = k$;
  End if
  $h = h-1$;
End while

By far, we discuss primarily the geodesic passing through any two points and the shortest geodesic distance connecting a given point. In next subsection, we shall further elaborate the stream preprocessing method COPOLAR in a form of algorithm.

4.4. COPOLAR Algorithm

We are now in a position to propose an algorithm to state stream preprocessing. Under the premise of not confusing, we also call this algorithm COPOLAR hereinafter.

The primary mission of COPOLAR is to obtain same number samples for each stream from MDS. To complete this task, the primary steps include: Projecting original observations onto the surface corresponding to POLAR; merging points with shortest geodesic distances and replacing two of them by their mid-point along the geodesic on POLAR.

Before stating more details of the algorithm, it is helpful to make a few assumptions. First, the algorithm is established based on the sliding window model and there are more than one observations arriving in each step. Secondly, if observation number of any stream of MDS is less than minimal threshold, we will randomly insert points into it to avert too little observations. Finally, we assume that the sliding widow has filled with observations coming from the last L steps before algorithm is called, where L represents the length of sliding window.

Algorithm 2: COPOLAR.

Input: f_i, x , n_{min}. *Where* f_i *represents the PDF of stream* X^i, $x=(x_1, x_2, ..., x_n)$ *is the newest observations,* $x_i = (x^i_1, x^i_2, \cdots, x^i_{n_i})^{\mathrm{T}}$ *is a vector containing the observations of* X^i *and* n_{min} *is the minimal threshold of the common number of points we hope to obtain.*
Output: The MDS X with common number for each stream X^i.

//Step 1: *Project the newest observations onto the POLAR.*
for i= 1, ..., *p do*
  Delete points from POLAR within the oldest window W_1 *for* X^i;
  Let $n^i = n^i - n^i_1 + n^i_L$; *//* $n^i_L = |x_i|$ *is the number of elements of* x_i ;
  If $n^i < n_{min}$ *Then*
    Insert $n_{min} - (n^i - n^i_1 + |x_i|)$ *points into* X^i *randomly;*

```
        Update n^i = n_min ;
    End if
    Project the newest observation x_i onto POLAR by Equation
    2 and Equation 5;
  End for
//Step 2: Calculate the shortest geodesic distances for the
 newest points.
 Let n = min{n^1, n^2, ..., n^p} ;
 for i = 1, ..., p do
    If n > n^i Then
      for w = 1, ..., n_L^i do
          GetDistanceForPoint (p_{n^i - n_L^i + w}); //Calls for
          Algorithm 1
      End for
    End if
 End for
//Step 3: Merge the points with shortest geodesic distances.
 For i = 1, ..., p do
    While n > n^i do
        Pick out n - n^i shortest geodesic distances and their
        corresponding points;
        Calculate the mid-point of these points by Equation
        21;
        Replace them by their mid-points respectively;
        Update n^i by the number of points of X^i.
    End while
 End for
```

The time complexity of Algorithm 2 is mainly determined by second loop statement. The loop body might be executed p-1 instances at the most. Thus, frequency of this statement which calls for Algorithm 1 is $T(\bar{c} * \bar{n}_L * (p-1))$, where $\bar{c}$ is average number to calculate the shortest geodesic distance for a given point and $\bar{n}_L = (\sum_{i=1}^{L} n_L^i)/n$ is average number of the newest observations. According to statistical laws, values of $\bar{c}$ and $\bar{n}_L$ can be considered as constants because they fluctuate within a constant range usually. Therefore, the time complexity of Algorithm 2 is $O(p)$.

5. Experiments and Analysis

We are now in a position to evaluate the effectiveness of our approaches by a series of simulation experiments. We first describe the methodology in section 5.1 and then analyze the results in section 5.2.

5.1. Methodology

We conduct three sets of simulation experiments. First set of experiments illustrate the effectiveness of COPOLAR by compared with optimal sampling [4] which is a newest sampling method and EM procedure [18] which is a classical algorithm for handling missing values on data streams. EM procedure derives from maximum likelihood estimation techniques and is available in many common statistical software packages. To this end, we primarily evaluate the cross covariance of MDS to illustrate the interrelationships within them.

The second set of experiments is presented to detect the influence of compression ratio on effectiveness of COPOLAR. Where, compression ratio equals to the ratio between sample numbers we obtained and original observation numbers. For this purpose, we typically compare the empirical cdf of preprocessing data with distribution of original data under different compression ratios.

But unlike the two formers, the last set of experiments is conducted to evaluate the differences in preprocessing among different distributions of streams. We principally compare two types of distributions, i.e., the symmetrical distributions and the skewed (or unsymmetrical) distributions.

For the sake of simplicity and without loss of generality, we set function for the time on POLAR to $M_t(i)= 1/(2* \sigma^2)+ M_t(i-1)$, where σ^2 is variance of original data.

We experiment on a PC with Inter Core Quad CPU @2.66GHz 2.67GHz and 2GB main memory using Matlab 7.12(R2011a) based on the OS Windows 7 Professional. The experiments are conducted on synthetic and real data sets. Description of data sets is as follows:

- Synthetic Data Sets: Stream values in each data set are generated according to different distributions. Three types of distributions are considered, i.e., the standard normal distribution $X \sim N(0,1)$ the lognormal distribution and the gaussian mixture two-peak distribution. Among them, the first one is symmetrical and others are skewed. The third distribution has presented in Equation 4. Unlike other two distributions which are familiar to us, the lognormal distribution may be unfamiliar for some readers. As a matter of fact, it is a common skewed distribution. For instance, the income of families in some regions follows this distribution. The lognormal PDF is:

$$y = f(x \mid \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} exp(-\frac{(\ln x - \mu)^2}{2\sigma^2}) \quad (23)$$

  Furthermore, values of parameters for each PDF shall be selected in accordance with different experiments. During the course of experiments, the synthetic data sets shall be adopted in almost all cases because of their representativeness and convenience to obtain.

- Real Data Set: The Diabetes data set is selected. It is a classical MDS and can be found from UCI machine learning repository (http: //archive.ics.uci. edu/ml/datasets/Diabetes). The Diabetes data set has 20 streams and the number of samples for each stream might be various in a same period (e.g., a month) or between two time intervals. During the course of experiments, the real data set is mainly employed for the first set of experiments. What's more, in order to avoid influence of different measurement units, data is normalized by the Gaussian normalization defined in Equation 6 before processing.

5.2. Results and Analysis

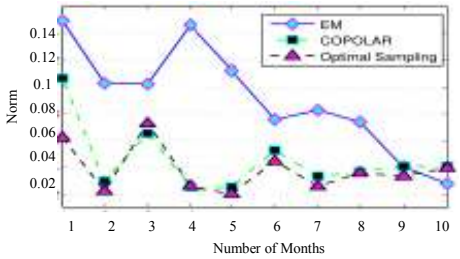
5.2.1. Experiment 1: Cross Covariance of MDS

In the first set of experiments, we primarily evaluate cross covariance of MDS to illustrate interrelationships within them. To this end, we compare the l_2 norm of Cross Covariance Matrix (CCM for short) of different data coming from three preprocessing approaches.

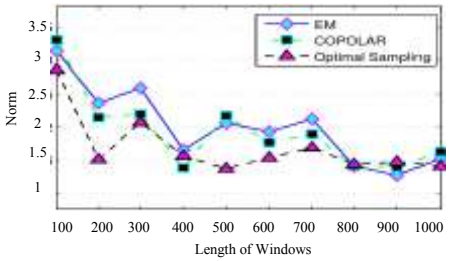
The experiments are carried out both on synthetic data set and Diabetes data set separately. The number of streams is set to 5. For synthetic data set, there is one stream whose window length for the i^{th} set is L_i=100×i and lengths of other streams are generated randomly within a closed interval [0.7×L_i, L_i], i= 1, ..., 10. But, for Diabetes data set, we first select randomly 5 distinct streams whose codes are from 33 to 72 and the length of each is no less than 1000.

A prerequisite to calculate the l_2 norm of CCM is that lengths are equal to each other for all streams. Therefore, we first employ three approaches separately to get same number of samples for all streams, and then calculate the l_2 norm. The first one adopted is EM Procedure which is used to fill values for those streams whose lengths are not maximum. Once processed by EM Procedure, lengths of all streams are equal to the maximum length of original data. In contrast, others, i.e., COPOLAR and optimal sampling are utilized to get fewer samples and lengths of all streams are equal to the minimum length of original data.

The l_2 norm of CCM processed by COPOLAR is compared with both the l_2 norms by other two approaches simultaneously and results are shown in Figure 2. The plots illustrate that the l_2 norm of CCM processed by COPOLAR interposes among others. Furthermore, the l_2 norm of CCM processed by each method gradually levels off with the increasing of window lengths, which inspires us that it is helpful to enlarge the length of windows appropriately.



a) Synthetic data set.



b) Diabetes data set.

Figure 2. l_2 norm of the CCM.

5.2.2. Experiment 2: Influence of Compression Ratio

In current experiment, we mainly investigate the influence of compression ratio on effectiveness of COPOLAR.

To evaluate the effectiveness, we utilize error to measure the difference between preprocessing data and original data. We define error for the empirical cdf as:

$$Error = \int_{x_0}^{x_1} (F_{pre} - F_{org}) dx \tag{24}$$

Where F_{pre} and F_{org} are the empirical cdf of preprocessing data D_{pre} and original data D_{org} respectively and x_0= $min\{D_{pre}, D_{org}\}$, x_1= $max\{D_{pre}, D_{org}\}$. Especially, we use cubic spine interpolation to obtain the empirical cdf precisely.

Experiments are repeated 30 times. And we evaluate their mean of errors. To simplify matters, we just only use synthetic data set. The data is generated in the same way as Experiment 1 did but window length is set to 1000 constantly for each repetition. The compression ratios are set from 0.3 to 0.9 with step size 0.05. The results are shown in Figure 3.

The plot illustrates that errors gradually converge to zero with the increasing of compression ratios, and that they fluctuate in a narrow range when compression ratio is more than 0.5. On these grounds, the availability of COPOLAR is validated because it is not common that difference of numbers for different streams is more than 50 percent.

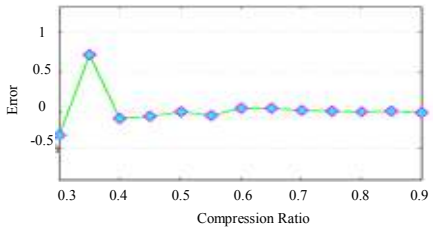


Figure 3. Influence of compression ratio on empirical cdf.

5.2.3. Experiment 3: Differences in Preprocessing Among Different Distributions

Inevitably, there is variability among different distributions for COPOLAR. The PDF is a significant parameter of POLAR which is the base of COPOLAR. The selection of the PDF varies with different data streams. Therefore, it shall be worth to cheer if the effectiveness of COPOLAR could be held for many streams with different distributions as far as possible. In the last set of experiments, we principally study the differences in preprocessing by COPOLAR among different distributions.

To evaluate differences, the error of empirical cdf defined in Equation 24 is utilized again. Experiments are conducted repeatedly for 100 times. The mean and variance of errors are employed to measure them. For each repetition, we generate three sets of data with length 1000 following distributions of standard normal

PDF, lognormal PDF defined in Equation 23 (Parameters are set to μ=1, σ=0.5) and gaussian mixture Pdf. defined in Equation 4 (Parameters are set to p=0.2, μ_1=1, σ_1=1, μ_2=9, σ_2=2) respectively. What's more, compression ratios are set to 0.8 consistently. The results are shown in Table 1. Statistics in the table show that errors under three distributions are all quite small and stable. Additionally, an interesting phenomenon which we cannot imagine that the error under gaussian mixture PDF is slightly less than errors under other two distributions, which indicates that COPOLAR can be utilized under complicated distributions.

Table 1. The mean and variance of errors about empirical cdf for different distributions

	Standard Normal	Lognormal	Gaussian Mixture
Mean of Errors	0.03964	0.04576	-0.01019
Variance of Errors	0.09027	0.04312	0.00012

In short, experiments above verify that COPOLAR is an effective preprocessing technique for MDS.

6. Conclusions and Future Issues

Preprocessing data is an essential step in the procedure of data stream mining. In this paper, we introduce the knowledge of differential geometry as a trick to preprocess MDS for the first time. The primary research contents of this study are as follows: A novel representation model of data stream called POLAR is established and a preprocessing approach called COPOLAR is proposed to extract same number samples from MDS.

The POLAR, as the basis of COPOLAR, is a geometry in $\mathbb{E}^3$, which consists of an intrinsic surface spanned by time, probability and stream value. To preprocess streams, COPOLAR merges points with shortest geodesic distances (also be regarded as the nearest one) along a geodesic on surface of POLAR into their mid-point on the same geodesic iteratively and incrementally until the number of points which we hope to obtain is met. By POLAR, time and probability, as two basic components of streams, could be involved in calculation in the procedure of COPOLAR. According to author's knowledge, the capability of being involved in calculation simultaneously about time, probability and value of streams, although as important as it is, has not found in existing literatures in fields of data streams mining.

To evaluate the effectiveness of our approaches, we conduct a series of simulation experiments on both synthetic and real data sets. Experimental results show that COPOLAR, as a preprocessing technique, is effective.

It's important to note that COPOLAR is different from sampling. They have different goals or missions. Properly speaking, COPOLAR is utilized to extract same number samples for some mining tasks purposes, but the goal of sampling is to obtain a part of samples for volume reducing purposes.

Although, there are some differences among them, COPOLAR might also be a sampling method inspired by the results of second experiment which indicates that the empirical cdf maintains original distribution effectively when compression ratio is no less than 0.5. Therefore, our future work is to study whether COPOLAR is feasible to be a sampling method.

Acknowledgements

This work is sponsored by the National Natural Science Foundation of China (61370083, 61073043, and 61073041), the National Research Foundation for the Doctoral Program of Higher Education of China (No.20112304110011 and No.20122304110012), the Natural Science Foundation of Heilongjiang Province (F200901) and the Harbin Outstanding Academic Leader Foundation of Heilongjiang Province of China (No.2011RFXXG015).

References

[1] Aggarwal C., "On Biased Reservoir Sampling in the Presence of Stream Evolution," *in Proceedings of the 32nd International Conference on Very Large Data Bases*, Seoul, Korea, pp. 607-618, 2006.

[2] Babcock B., Babu S., Datar M., Motwani R., and Widom J., "Models and Issues in Data Stream Systems," *in Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, Wisconsin, USA, pp. 1-16, 2002.

[3] Beringer J. and Hllermeier E., "Online Clustering of Parallel Data Streams," *Data and Knowledge Engineering*, vol. 58, no. 2, pp. 180-204, 2006.

[4] Braverman V., Ostrovsky R., and Zaniolo C., "Optimal Sampling from Sliding Windows," *in Proceedings of the 28th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, Rhode Island, USA, pp. 147-156, 2009.

[5] Byung-Hoon P., George O., and Nagiza F., "Sampling Streaming Data with Replacement," *Computational Statistics and Data Analysis*, vol. 52, no. 2, pp. 750-762, 2007.

[6] Chen L., Zou L., and Tu L., "A Clustering Algorithm for Multiple Data Streams based on Spectral Component Similarity," *Information Sciences*, vol. 183, no. 1, pp. 35-47, 2012.

[7] Ciampi A., Appice A., and Malerba D., "Summarization for Geographically Distributed Data Streams," *in Proceedings of the 14th International Conference Knowledge-Based and Intelligent Information and Engineering Systems*, Cardiff, UK, pp. 339-348, 2010.

[8] Crone S., Lessmann S., and Stahlbock R., "The Impact of Preprocessing on Data Mining :An Evaluation of Classifier Sensitivity in Direct Marketing," *European Journal of Operational Research*, vol. 173, no. 3, pp. 781-800, 2006.

[9] Davis J. and Clark A., "Data Preprocessing for Anomaly Based Network Intrusion Detection: A Review," *Computers and Security*, vol. 30, no. 6-7, pp. 353-375, 2011.

[10] Demaine E., López-Ortiz A., and Munro J., "Frequency Estimation of Internet Packet Streams with Limited Space," *in Proceedings of the 10th Annual European Symposium on Algorithms*, Rome, Italy, pp. 348-360, 2002.

[11] Éric W., *Numerical Methods and Optimization*, Springer, 2014.

[12] Gaber M., Zaslavsky A., and Krishnaswamy S., "Mining Data Streams a Review," *in ACM SIGMOD Record*, vol. 34, no. 2, pp. 18-26, 2005.

[13] Granmo O. and Oommen B., "Optimal Sampling for Estimation with Constrained Resources using a Learning Automaton-based Solution for the Nonlinear Fractional Knapsack Problem," *Applied Intelligence*, vol. 33, no. 1, pp. 3-20, 2010.

[14] Li J., Jia Q., Guan X., and Chen X., "Tracking a Moving Object via a Sensor Network with a Partial Information Broadcasting Scheme," *Information Sciences*, vol. 181, no. 20, pp. 4733-4753, 2011.

[15] Lim Y. and Kang S., "Intelligent Approach for Data Collection in Wireless Sensor Networks," *the International Arab Journal of Information Technology*, vol. 10, no. 1, pp. 36-42, 2013.

[16] Palmer C. and Faloutsos C., "Density Biased Sampling an Improved Method for Data Mining and Clustering," *in Proceedings of ACM SIGMOD International Conference on Management of Data*, Texas, USA, pp. 82-92, 2000.

[17] Serir L., Ramasso E., and Zerhouni N., "Evidential Evolving Gustafsonckessel Algorithm for Online Data Streams Partitioning using Belief Function Theory," *the International Journal of Approximate Reasoning*, vol. 53, no. 5, pp. 747-768, 2012.

[18] Smith J., Borckardt J., and Nash M., "Inferential Precision in Single-Case Time-Series Data Streams: How Well does the EM Procedure Perform when Missing Observations Occur in Autocorrelated Data," *Behavior Therapy*, vol. 43, no. 3, pp. 679-685, 2012.

[19] Sun J., He K., and Li H., "SFFS-PC-NN Optimized by Genetic Algorithm for Dynamic Prediction of Financial Distress with Longitudinal Data Streams," *Knowledge-Based Systems*, vol. 24, no. 7, pp. 1013-1023, 2011.

[20] Victor T., *Differential Geometry of Curves and Surfaces*, Springer-Verlag, 2006.

[21] Wang Y., Zhang G., and Qian J., "ApproxCCA: An Approximate Correlation Analysis Algorithm for Multidimensional Data Streams," *Knowledge-Based Systems*, vol. 24, no. 7, pp. 952-962, 2011.

[22] Zhang J., Xu J., and Liao S., "Sampling Methods for Summarizing Unordered Vehicle-to-Vehicle Data Streams," *Transportation Research: Part C Emerging Technologies*, vol. 23, pp. 56-67, 2012.

[23] Zhang T., Yue D., Gu Y., Wang Y., and Yu G., "Adaptive Correlation Analysis in Stream Time Series with Sliding Windows," *Computers and Mathematics with Applications*, vol. 57, no. 6, pp. 937-948, 2009.

[24] Zhang Z. and Zhou J., "Transfer Estimation of Evolving Class Priors in Data Stream Classification," *Pattern Recognition*, vol. 43, no. 9, pp. 3151-3161, 2010.

Li Wen-Ping received his DEng degree in College of Computer Science and Technology, Harbin Engineering University, China. Currently, he is working as Associate Professor in College of Mathematics Physics and Information Engineering, Jiaxing University, China. His research interests are in the areas of data stream, data mining, privacy preservation and membrane computing.

Yang Jing received her DEng degree in College of Computer Science and Technology, Harbin Engineering University, China. Currently, she is working as Professor in the College of Computer Science and Technology, Harbin Engineering University, China. Her research interests are in the areas of database, data mining and privacy preservation.

Zhang Jian-Pei received his DEng degree in College of Computer Science and Technology, Harbin Engineering University, China. Currently, he is working as Professor in the College of Computer Science and Technology, Harbin Engineering University, China. His research interests are in the areas of database, data mining and social network.