Neighbor Profile: Bagging Nearest Neighbors for Unsupervised Time Series Mining

Conference Paper · April 2020

DOI: 10.1109/ICDE48307.2020.00039

Yuanduo He

Peking University

Yasha Wang

Peking University

Neighbor Profile: Bagging Nearest Neighbors for Unsupervised Time Series Mining

Yuanduo He*†, Xu Chu*‡, Yasha Wang*†§
* Key Laboratory of High Confidence Software Technologies, Ministry of Education, Beijing 100871, China
† National Engineering Research Center of Software Engineering, Peking University, Beijing 100871, China
‡ School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
§ The corresponding author, email: wangyasha@pku.edu.cn

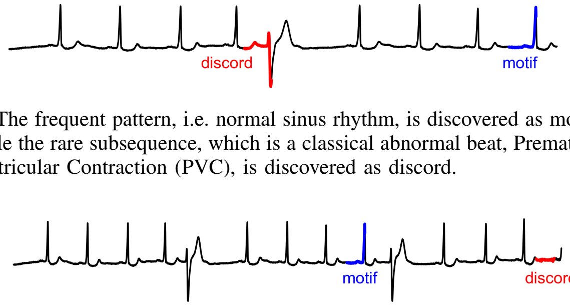
***Abstract*—Unsupervised time series mining has been attracting great interest from both academic and industrial communities. As the two most basic data mining tasks, the discoveries of frequent/rare subsequences have been extensively studied in the literature. Specifically, frequent/rare subsequences are defined as the ones with the smallest/largest 1-nearest neighbor distance, which are also known as motif/discord. However, discord fails to identify rare subsequences when it occurs more than once in the time series, which is widely known as the *twin freak problem*. This problem is just the "tip of the iceberg" due to the 1-nearest neighbor distance based definitions. In this work, we for the first time provide a clear theoretical analysis of motif/discord as the 1-nearest neighbor based nonparametric density estimation of subsequence. Particularly, we focus on *matrix profile*, a recently proposed mining framework, which unifies the discovery of motif and discord under the same computing model. Thereafter, we point out the inherent three issues: *low-quality density estimation*, *gravity defiant behavior*, and *lack of reusable model*, which deteriorate the performance of matrix profile in both efficiency and subsequence quality.**

To overcome these issues, we propose *Neighbor Profile* to robustly model the subsequence density by bagging nearest neighbors for the discovery of frequent/rare subsequences. Specifically, we leverage multiple subsamples and average the density estimations from subsamples using adjusted nearest neighbor distances, which not only enhances the estimation robustness but also realizes a reusable model for efficient learning. We check the sanity of neighbor profile on synthetic data and further evaluate it on real-world datasets. The experimental results demonstrate that neighbor profile can correctly model the subsequences of different densities and shows superior performance significantly over matrix profile on the real-world arrhythmia dataset. Also, it is shown that neighbor profile is efficient for massive datasets.

***Index Terms*—Time series, motif, discord, nearest neighbor, unsupervised mining**

I. Introduction

Mining subsequences of interest unsupervisedly from time series has been receiving widespread attention from both academic and industrial communities [1], [2]. Particularly, a lot of effort has been devoted to the discovery of frequent patterns and rare subsequences, which have been successfully applied in a variety of fields, such as medicine, astronomy, etc [3]–[5]. The frequently occurring patterns describe the regularity in the time series, which may indicate some common underlying events, while the rare (or unusual) subsequences describe the irregularity, which often suggests an anomaly [3], [6]. To illustrate, Fig. 1a shows a frequent subsequence (the blue segment) and a rare one (the red segment) in an Electrocardiograph (ECG) signal. The frequent pattern depicts the normal sinus rhythm, while the rare subsequence depicts a classical abnormal beat, Premature Ventricular Contraction.



(a) The frequent pattern, i.e. normal sinus rhythm, is discovered as motif, while the rare subsequence, which is a classical abnormal beat, Premature Ventricular Contraction (PVC), is discovered as discord.

(b) The twin freak problem: *when two PVCs exist, they cannot be discovered as rare subsequence by discord.* Note that this signal is the same with the above one but with only extra 5 seconds.

Fig. 1. Examples of frequent and rare subsequences in ECG signals [7].

Intuitively, frequent and rare subsequences should be implemented according to the *subsequence density*. However, the widely accepted implementations, motif and discord, are developed using the 1-nearest neighbor (1NN): motif/discord is defined as the subsequence with the smallest/largest 1NN distance, which have been shown to identify frequent and rare subsequences successfully in many cases [3], [8]. For example, the frequent and the rare subsequences in Fig. 1a are actually discovered as the motif and the discord respectively. Also, Keogh et al. provided an explanation that existing density based algorithms fail because rare subsequences "do not necessarily live in sparse areas" and 1NN distance can identify the anomaly (similar reason applies to motif) [3].

In this work, we argue that 1NN distance based implementations for frequent and rare subsequences, i.e. motif and discord, are *still* a kind of density estimation, namely *1NN based nonparametric density estimation* [9], [10]. In spite of the many achieved success of motif and discord, they are inherently with issues deteriorating the estimated density, which further results in the failure of mining expected subsequence. A prominent phenomenon is the *twin freak problem*: when a particular kind of rare subsequence is not unique in time series, it may be discovered as common or even frequent subsequences [11], [12]. In Fig. 1b, the abnormal beat occurring only twice fails to be recognized as rare subsequences by discord. This problem results from the low-quality density of 1NN (issue 1) [10] and the gravity-defiant behavior of learning curve (issue 2) [13]. Although using k-nearest neighbor (kNN) distance is considered to be able to avoid such cases [11], this modification is not *trivial*, which will cause a lot of storage and computing overhead. Also, 1NN based methods (e.g., matrix profile) do not explicitly[1] build models, making it inefficient in both running time and storage space for mining long time series (issue 3).

To address these issues, we propose *Neighbor Profile*, a novel time series mining framework, to discover frequent and rare subsequences by robustly modeling the density of subsequences with bagged nearest neighbors. Considering nearest neighbors in multiple subsamples, neighbor profile solves the three issues simultaneously: (issue 1) 1-nearest neighbor in a subsample can be alternatively viewed as k-nearest neighbor, which enhances the robustness of estimation [10]; (issue 2) subsampling helps alleviate the behavior of gravity defiant [14]; (issue 3) it constructs an ensemble model using only a small subset of samples, in which way the computational cost is greatly reduced and a reusable model is also obtained. It is also worth mentioning that matrix profile can be viewed as an extreme case of neighbor profile with only a single subsample as the whole datasets. We perform extensive experiments on both synthetic and real-world dataset to evaluate the performance of neighbor profile.

In summary, the contributions of this work are three folds:

- To our best knowledge, this is the first work theoretically demonstrating the essence of motif and discord as the 1-nearest neighbor based nonparametric density of subsequences. Based on that, we further point out three issues inherently along with the two implementations.
- We develop new implementations of frequent and rare subsequences by proposing an ensemble framework, neighbor profile, to robustly estimate the subsequence density with bagged nearest neighbors.
- We evaluate neighbor profile on both synthetic and real world datasets. The experimental results show that neighbor profile can describe the density of subsequences efficiently and robustly compared with motif and discord.

The rest of this paper is organized as follows. In section II, we introduce necessary background knowledge and present a detailed theoretical analysis of motif and discord. Particularly, we focus on matrix profile to illustrate the three issues. To overcome these issues, we propose neighbor profile in section III. Section IV presents experimental results on both synthetic and real-world datasets. Related works are reviewed in section V. In the end, we conclude this work in section VI.

II. Background

In this section, we introduce background knowledge about motif and discord. Particularly, we focus on a recently proposed time series mining framework, matrix profile [15], which unifies the definitions of motif and discord and draws great attentions in the community[2]. We aim to answer the following questions:

What is the essence of matrix profile? And what issues does it bring to the mining of frequent and rare subsequences?

A. Concepts and Notations

Key concepts and notations are introduced as below.

Definition 1 (Time Series)**.** *A time series $T \in \mathbb{R}^l$ is a sequence of scalars as $(t_1, t_2, \cdots, t_l)$, $t_i \in \mathbb{R}$, where $l \in \mathbb{N}$ is the length.*

Definition 2 (Subsequence)**.** *A subsequence of time series T is a sequence of scalars as $T_{i,m} = (t_i, t_{i+1}, \cdots, t_{i+m-1}) \in \mathbb{R}^m$,where $m \in \mathbb{N} \leq l$ is the length of subsequence and $i \in \{1, \cdots, l-m+1\}$ is the starting position.*

Given a time series T, its all possible subsequences of length m is denoted as the set $S_T^m = \{T_{i,m}|_{i=0,1,\cdots,l-m+1}\}$. The length m of interest usually depends on the dataset.

The subsequence mining task of interest aims to find $S \in S_T^m$ such that S is frequent or rare in S_T^m. Thus, the core concepts are the definitions of frequent and rare subsequences. The existing widely accepted definitions, i.e. motif and discord, are introduced in the following.

Definition 3 (Distance)**.** *The distance between two equal-length subsequences*[3] *$T_{i,m}$ and $T_{j,m}$ are defined as below*:

$$d(T_{i,m}, T_{j,m}) = \sqrt{\Sigma_{k=0}^{m-1}(t_{i+k} - t_{j+k})^2}. \quad (1)$$

In the real applications, subsequences are often normalized before calculating distance. Two basic normalization methods are *demean* and *zscore*: demean normalizes the mean as 0, and zscore further normalizes the standard deviation as 1. The corresponding distances are then named as demean-distance and zscore-distance. For instance, matrix profile uses zscore-distance to measure the similarity between subsequences.

Definition 4 (Nearest Neighbor)**.** *The nearest neighbor of a subsequence $\mathbf{y}$ w.r.t. a subsequence set S is the subsequence $\mathbf{x} \in S$ that has the smallest distance between them*:

$$nn(\mathbf{y}; S) = \underset{\mathbf{x} \in S}{\operatorname{argmin}}\ d(\mathbf{y}, \mathbf{x}). \quad (2)$$

Definition 5 (Nearest Neighbor Distance)**.** *The nearest neighbor distance of a subsequence $\mathbf{y}$ w.r.t. a subsequence set S is*:

$$nnd(\mathbf{y}; S) = d(\mathbf{y}, nn(\mathbf{y}; S)). \quad (3)$$

In fact, definitions 4 and 5 describe the 1-nearest neighbor and its distance, both of which can be easily generalized to the case of k-th nearest neighbor. Existing implementations of frequent and rare subsequences, motif and discord, are based on 1NN distance as below.

[1] Or equivalently, using the whole dataset itself as the model.

[2] Note that though matrix profile provides a variety of mining algorithms, the term "matrix profile" in this section is mainly referred to the 1NN based way of modeling the frequency of subsequences, including motif and discord.

[3] The two subsequences can also be from different time series.

Definition 6 (Motif)**.** *A motif is the subsequence* $\mathbf{y}$ *in* S_T^m *with the smallest 1-nearest neighbor distance*:

$$motif(T) = \underset{\mathbf{y}\in S_T^m}{\operatorname{argmin}}\; nnd(\mathbf{y}; S_T^m \setminus \{\mathbf{y}\}). \tag{4}$$

Definition 7 (Discord)**.** *A discord is a subsequence* $\mathbf{y}$ *in* S_T^m *with the largest 1-nearest neighbor distance*:

$$discord(T) = \underset{\mathbf{y}\in S_T^m}{\operatorname{argmax}}\; nnd(\mathbf{y}; S_T^m \setminus \{\mathbf{y}\}). \tag{5}$$

The above definitions are actually the top-1 motif and top-1 discord, which can be extended to top-k motif and top-k discord by introducing the k-th smallest and k-th largest 1-nearest neighbor distance. To unify motif and discord, matrix profile is proposed as follows.

Definition 8 (Matrix Profile)**.** *A matrix profile of a time series* T *is an annotation series, where each element* mp_i *is the nearest neighbor distance of the subsequence* $T_{i,m}$:

$$mp(T) = (mp_1, mp_2, ..., mp_{l-m+1}), \tag{6}$$

where $mp_i = nnd(T_{i,m}; S_T^m \setminus \{T_{i,m}\})$.

According to the definitions of motif and discord, the minimum value in $mp(T)$ corresponds to the motif; and the maximum one is the discord. Note that in definitions 6, 7 and 8, the handling of trivial match is omitted for brevity. For details, please refer to [15].

B. The Essence of Matrix Profile

We argue for the first time that, *matrix profile is essentially the 1-nearest neighbor based nonparametric density estimation of subsequences*. To illustrate, consider $S_T^m \in \mathbb{R}^m$ as the samples of a multivariate random variable drawn from some underlying distribution, and estimate the density $p(\mathbf{x})$ of a subsequence $\mathbf{x} \in \mathbb{R}^m$ based on S_T^m in a nonparametric way. According to [16], given a small region $\mathcal{R}$ containing $\mathbf{x}$, we have

$$k/n \approx Pr(\mathbf{x} \in \mathcal{R}) = \int_{\mathcal{R}} p(\mathbf{x}\prime)d\mathbf{x}\prime \approx p(\mathbf{x})V, \tag{7}$$

where n is the number of total elements in S_T^m, k is the number of subsequences lying in region $\mathcal{R}$, V is the volume of region $\mathcal{R}$, and $Pr(\cdot)$ is the probability.

The left approximation in (7) is the maximum likelihood estimate of the probability of $\mathbf{x}$ in $\mathcal{R}$; and the right one is the approximation of the integral. Then, $p(\mathbf{x})$ is estimated as

$$p(\mathbf{x}) \approx \frac{k/n}{V}. \tag{8}$$

Given (8), there are two ways to complete the final estimation: (1) fixing V, count k; (2) fixing k, calculate V. The former one is widely known as Parzen-Rosenblatt window method [17] or kernel density estimation; the latter one is the k-nearest neighbor based method [9], [10]. Particularly, we consider using 1-nearest neighbor method to estimate $p(\mathbf{x})$ as

$$\hat{p}(\mathbf{x}; S_T^m) = \frac{1/n}{\frac{\pi^{m/2}}{\Gamma(\frac{m}{2}+1)} nnd(\mathbf{x}; S_T^m)^m}, \tag{9}$$

where $k = 1$ and the denominator is the volume formula of an m-dimensional hypersphere as the region $\mathcal{R}$. Thus, we have the following relationship

$$mp_i \propto -\log \hat{p}(T_{i,m}; S_T^m). \tag{10}$$

According to (10), motif and discord can be interpreted from the perspective of density of subsequences: *motif corresponds to the subsequence* $\mathbf{x}$ *with the maximum estimated density; and discord the minimum.* It can been seen that the definitions 6 and 7 are consistent with the original concepts, i.e. frequently occurring patterns and most unusual subsequences. Specifically, discord is actually the classical distance-based outlier [18]–[20].

In conclusion, matrix profile can be equivalently viewed as the 1-nearest neighbor based nonparametric density estimation of subsequences in a time series. Based on the estimated density, motif and discord can be defined and identified under a same data mining framework.

C. Three Issues about Matrix Profile

In spite of the achieved encouraging success, there are still *three* major issues inherently with matrix profile as the 1-nearest neighbor based nonparametric density estimator.

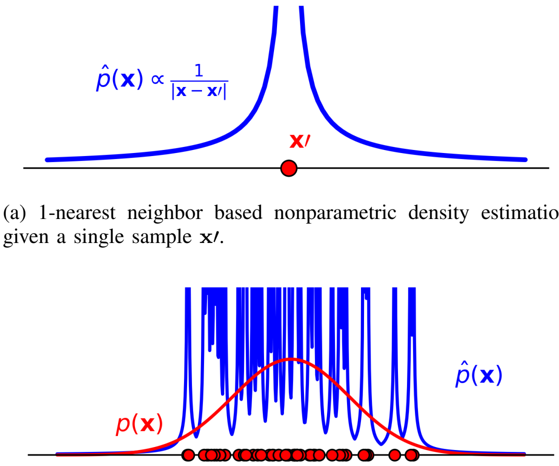
1. A low-quality estimator due to the 1-nearest neighbor based density estimation. It has already been pointed out that "*the overall estimates obtained by the nearest neighbor method are not very satisfactory*" [10]. Particularly, when x is too close to its nearest neighbor, the estimator $\hat{p}(x)$ suffers from *spike problem*, i.e. not smooth and discontinuous at all the same points in S_T^m, which is illustrated in Fig. 2. The spikes are caused by the estimator $\hat{p}(x)$ as a negative exponent, which seriously affects the estimated density and may result in the discovery of spurious frequent or rare subsequences. Subsequently, the discovered subsequence may not be satisfying either. For details about density estimation, please refer to [10].

It is worth mentioning that using k-nearest neighbor ($k > 1$) may help alleviate this issue theoretically [10]. However, it is nontrivial in the real implementation, which incurs a lot of overhead for matrix profile to consider k nearest neighbors. Specifically, it at least needs extra $\mathcal{O}(kl)$ storage space to save k distances and $\mathcal{O}(kl)$ to maintain the order of k nearest neighbor. As a result, the time and space efficient of matrix profile can be seriously affected. Moreover, how to choose an appropriate k is not a trivial problem either, since the quantity of rare subsequences is usually unknown to data miners. The incurred overhead for matrix profile also makes an intractable search for an appropriate k. Last but not least, considering the k-nearest neighbor can not address the following issue.

2. The gravity defiant behavior due to the k-nearest neighbor based outlier detection. In the community of outlier detection, it has been recently studied by independent researchers that *using more samples results in lower accuracy of k-nearest neighbor based anomaly detection*, which is known as gravity-defiant behavior[4] [13], [14], [21], [22]. This behavior can be intuitively explained that as data size grows, more anomalies contaminate the dataset, and the average distances between anomaly and normal samples are decreased, making it difficult to detect distance-based outliers. An illustrative example is shown in Fig. 3. For interested readers, a detailed theoretical analysis is in [13]. As for matrix profile, discord is actually the k-nearest neighbor based anomaly, and hence suffers from the gravity defiant behavior. In the context of time series data mining, this issue turns out to be much more serious, since the data size can be considerably large given a long time series [23].

In this position, we can clearly see that it is the failure of robust density modeling and the gravity defiant behavior that result in the twin freak problem. Specifically, the spike may cause a high estimated density for infrequent or even rare subsequences; when the size of subsequences is large, more anomalies are mixed in and the average nearest neighbor distances among them become smaller, which reflects the gravity defiant behavior.

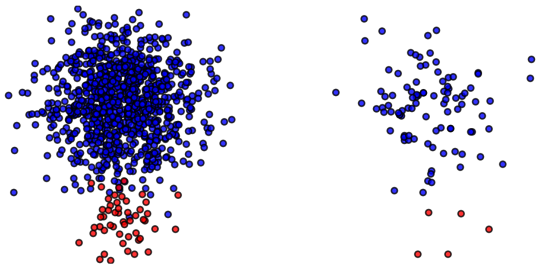
3. The lack of trained model due to the lazy learning of nearest neighbor method. It is widely known that nearest neighbor has an extremely high runtime cost when the training dataset grows very large, which is quite common for time series data; a lot of effort has been devoted to alleviating this problem by leveraging highly optimized Fast Fourier Transform and GPU devices [15], [23], [24]. However, the theoretical time complexity remains unchanged as $\mathcal{O}(l^2 \log l)$ given a time series of length l [15]. In addition, since it does not learn a concise model from the original dataset, matrix profile has to leverage the whole dataset itself as the model when mining subsequences from other time series, making it inefficient in storage. Therefore, the lack of trained model puts a limit upon the reusability and applicability of matrix profile.



(a) 1-nearest neighbor based nonparametric density estimation given a single sample $\mathbf{x}\prime$.

(b) 1-nearest neighbor based nonparametric density estimation given 50 samples drawn from a normal distribution $p(\mathbf{x})$. Both $p(\mathbf{x})$ and $\hat{p}(\mathbf{x})$ are scaled in height; and $\hat{p}(\mathbf{x})$ is also cut off in height for better visualization.

Fig. 2. Illustration of the spike phenomenon in the one-dimensional space.



(a) Total sample size is 1000: 950 normal data, and 50 anomalies, which are mixed.

(b) Total sample size is 100: 95 normal data, and 5 anomalies, which are well separated.

Fig. 3. An example of gravity defiant behavior: normal (blue) and anomaly (red) data form two Gaussian distributions with different sample sizes.

In conclusion, the above three issues deeply affect the performance of frequent and rare subsequences mining, which results in the discovery of spurious subsequences and limits the further success of unsupervised subsequence mining.

III. Neighbor Profile

In this section, we present *neighbor profile*, a novel ensemble framework, to address the issues of matrix profile.

A. Insight

The basic idea is still to perform the nonparametric density estimation of subsequences *but* in a bagging way [25]. Bagging learns models from multiple subsamples and aggregates them by averaging, which reduces variance and improves the stability of estimated density. Bagging is a simple yet “silver bullet” to the three issues of matrix profile:

- Issue1: 1NN based density estimation in a subsample can be approximately viewed as kNN based, which can help improve the quality of estimation. Moreover, bagging itself can enhance the robustness of the estimated density.
- Issue2: learning from subsamples reduces the training size, which effectively avoids the gravity defiant behavior and improves the performance of detecting anomaly.
- Issue3: using a small subset of training data not only leads to a storage-efficient reusable model but also achieves a low computational cost.

In addition to bagging, we also propose an adjusted nearest neighbor distance to robustly estimate the subsequence density for each subsample, which is detailed in the followings.

B. Density Estimation with Subsamples

In this subsection, we present how to perform subsequence density estimation in a bagging way: (1) density estimation in subsamples and (2) aggregation.

1. Density Estimation in a Single Subsample. We begin with the definition of subsample. Particularly, we consider sampling without replacement, because the duplication of subsequences results in a nearest neighbor distance of 0, which will cause the numerical instability.

[4] Literally, the “gravity defiant” describes the phenomenon that the error rate curve rises as the data size grows *too* large.

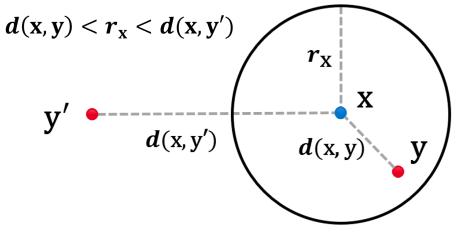

$d(\mathbf{x},\mathbf{y}) < r_{\mathbf{x}} < d(\mathbf{x},\mathbf{y}')$
$r_{\mathbf{x}}$
$\mathbf{x}$
$\mathbf{y}'$
$d(\mathbf{x},\mathbf{y}')$
$d(\mathbf{x},\mathbf{y})$
$\mathbf{y}$

Fig. 4. An illustrative example of adjusted distance by nearest neighbor balls. $\mathbf{x}$ is a sampled subsequence, forming a nearest neighbor ball with a radius $r_{\mathbf{x}}$. $\mathbf{y}$ and $\mathbf{y}\prime$ are two subsequences for density estimation, and $\mathbf{x}$ is the nearest neighbor for both of them. When the subsequence y falls inside the ball, its nearest neighbor distance is adjusted as r_x for a robust estimation.

Definition 9 (Subsample)**.** *A subsample Q is a subset randomly drawn from S_T^m without replacement.*

A straightforward way is to use 1-nearest neighbor method directly on a subsample for density estimation. However, when a subsequence $\mathbf{y}$ is too close to a subsequence $\mathbf{x} \in Q$, the estimated density would become extremely large and unrobust according to (9). Thus, we consider an adjusted nearest neighbor distance by leveraging nearest neighbor balls, as is shown in Fig. 4.

Definition 10 (Nearest Neighbor Ball)**.** *Given a subsample Q, the nearest neighbor ball of $\mathbf{x} \in Q$ is centered at $\mathbf{x}$ with the radius $r_{\mathbf{x}} = nnd(\mathbf{x}; Q \setminus \{\mathbf{x}\})$:*

$$B(\mathbf{x}; Q) = \{\mathbf{y} | d(\mathbf{x}, \mathbf{y}) \leq nnd(\mathbf{x}; Q \setminus \{\mathbf{x}\})\}. \tag{11}$$

According to (9), the estimated density of the subsequence $\mathbf{y}$ given a subsample Q is

$$\tilde{p}(\mathbf{y}; Q) = \frac{1/n}{\frac{\pi^{m/2}}{\Gamma(\frac{m}{2}+1)} r(\mathbf{y}; Q)^m}, \tag{12}$$

where $r(\mathbf{y}; Q)$ is the adjusted nearest neighbor distance defined as

$$r(\mathbf{y}; Q) = \begin{cases} nnd(\mathbf{y}; Q), & \text{if } \mathbf{y} \notin \bigcup_{\mathbf{x} \in Q} B(\mathbf{x}; Q), \\ nnd(nn(\mathbf{y}; Q); Q \setminus \{nn(\mathbf{y}; Q)\}), & \text{otherwise.} \end{cases} \tag{13}$$

In (13), when $\mathbf{y}$ is too close to its nearest neighbor, i.e. falling inside the nearest neighbor ball, its nearest neighbor distance will be automatically adjusted as the radius of the nearest neighbor ball for a robust estimation. A direct benefit is that when the subsequence $\mathbf{y}$ happens to be in a subsample, its nearest neighbor distance will be adjusted from 0 to a positive number for a numerically stable estimation[5].

2. Aggregation. With $\tilde{p}(\mathbf{y}; Q_1), \cdots, \tilde{p}(\mathbf{y}; Q_n)$ estimated from n subsamples, we show how to aggregate them into the final result, i.e. neighbor profile, in this part.

A simple method is to directly calculate (12) and use the arithmetic mean. However, a usual subsequence length could be greater than 100, which will result in the numerical instability of (12) due to the power of m in the denominator. Therefore, we consider using geometric mean as an alternative for the final estimation as

$$\tilde{p}(\mathbf{y}; S_T^m) = \sqrt[n]{\prod_{i=1}^{n} \tilde{p}(\mathbf{y}; Q_i)}. \tag{14}$$

Since the relative sizes of estimated subsequence densities are enough for unsupervised mining, we further simplify (14) by taking logarithm and removing constant coefficient as

$$\tilde{p}(\mathbf{y}; S_T^m) \propto -\frac{1}{n} \sum_{i=1}^{n} \log r(\mathbf{y}; Q_i). \tag{15}$$

Finally, neighbor profile is presented as follows.

Definition 11 (Neighbor Profile)**.** *A neighbor profile of a time series T is an annotation series. Each element np_i is the estimated relative density for $T_{i,m}$:*

$$np(T) = (np_1, np_2, ..., np_{l-m+1}), \tag{16}$$

where $np_i = \frac{1}{n} \sum_{j=1}^{n} \log r(T_{i,m}; Q_j)$ and Q_j's are subsamples drawn from S_T^m.

To make it consistent with matrix profile, np_i is defined as the opposite number of (15): the larger the np_i is, the lower the estimated density of subsequence $T_{i,m}$ is. **Thus, the one with the smallest np_i is identified as the frequent subsequence; the largest as the rare subsequence.** It is also worth mentioning that neighbor profile allows a variety of distance measures in (13), making neighbor profile flexible for different tasks.

Note that neighbor profile can be viewed as an extension of matrix profile. Particularly, when there is only a single subsample, i.e. $n = 1$, and also Q_1 equals S_T^m, neighbor profile will be reduced to matrix profile equivalently.

C. Algorithm

The algorithm for neighbor profile is presented in alg. 1. It consists of two stages: the model construction from line 1 to 7, and the inference from line 8 to 21.

The first stage constructs subsamples $Q_1, \cdots, Q_n$ from all subsequences S_T^m in the first two lines. In the following loop, for each subsample Q_i, it performs pairwise distance calculation to learn the nearest neighbor distance for $B(\mathbf{x}; Q_i)$. The learned nearest neighbor balls model the density of subsequences from the perspective of the subsample. In the second stage, we calculate neighbor profile np_i for each subsequence $T_{i,m}$. In the loop starting in line 11, we leverage learned nearest neighbor balls to estimate the density according to (13). The final estimation is made in line 20.

The last line finds the frequent and the rare subsequences according to the learned neighbor profile: subsequences with lower np_i are considered as frequent subsequences, while the ones with higher np_i are as rare ones.

[5] It is assumed that sampling without replacement is enough to avoid a nearest neighbor ball of 0 radius, which can be usually satisfied due to the ubiquitous noise in the data.

Algorithm 1 Calculation of Neighbor Profile

```
Input: A time series $T$ of length $l$, the length of subsequence $m$,
    subsample number $n$ and subsample size $s$
Output: The neighbor profile $np$ of time series $T$
 1: construct $n$ subsamples $Q_i$ of size $s$ from $S_T^m$
 2: for $i = 1$ to $n$ do
 3:     for all $\mathbf{x} \in Q_i$ do
 4:         calculate $nnd(\mathbf{x}; Q_i \setminus \{\mathbf{x}\})$ for $B(\mathbf{x}; Q_i)$
 5:     end for
 6: end for
 7: $np \leftarrow$ empty array
 8: for $i = 1$ to $l - m + 1$ do
 9:     $np_i \leftarrow 0$
10:     for $j = 1$ to $n$ do
11:         $r_j \leftarrow 0$
12:         if $T_{i,m} \in \bigcup_{\mathbf{x} \in Q_j} B(\mathbf{x}; Q_j)$ then
13:             $r_j \leftarrow nnd(nn(T_{i,m}; Q_j); Q_j \setminus \{nn(T_{i,m}; Q_j)\})$
14:         else
15:             $r_j \leftarrow nnd(T_{i,m}; Q_j)$
16:         end if
17:         $np_i \leftarrow np_i + \log r_j$
18:     end for
19:     $np_i \leftarrow \frac{1}{n} np_i$
20: end for
21: find the frequent and the rare subsequences w.r.t. $np$
```

D. Time and Space Complexity

The time complexity of training is $\mathcal{O}(nms^2)$, i.e. the pairwise distance calculation for nearest neighbor balls in line 3 to 7. As for testing, the time complexity is $\mathcal{O}(nmsl)$, which usually dominates the whole time complexity since $l \gg s$. The experiment evaluation is further presented in section IV-C. As for the space complexity, both the model and the testing phase require $\mathcal{O}(nms)$ for all subsamples and estimation respectively.

E. Hyperparameter Settings

Compared to matrix profile, there are two extra hyperparameters in neighbor profile: the number of subsample n and the size of subsample s. The subsample number controls the convergence of neighbor profile. A slightly large n (e.g., 100) is enough for a converged result. Hence, unless otherwise stated we shall use $n = 100$ by default in the experiment.

As for the subsample size, it usually depends on the specific data mining task. However, existing anomaly detection works indirectly show that a small subsample size can often lead to a satisfying performance [14], [26]. Theoretically, a too large s not only incurs a lot of computing overhead but also makes neighbor profile behave like matrix profile. On the other hand, when s is too small, the subsamples may only contain the most frequent subsequences such that rare and normal subsequences cannot be distinguished from each other. A detailed analysis will be further presented in section IV-A.

IV. Experiment

In this section, we conduct experiments to evaluate the proposed data mining method. We aim to answer the following three research questions with various experiments:

- **RQ1:** *How effective does neighbor profile model the density of subsequences for unsupervised mining?*
- **RQ2:** *How does neighbor profile perform when compared with existing 1NN based implementations?*
- **RQ3:** *How efficient is neighbor profile?*

The baselines are the 1NN based density estimations, which have been implemented by dozens of works [3], [4], [6], [27], [28]. Particularly, we choose the algorithms from matrix profile to mine motif and discord as the baseline results throughout these experiments, since they are both exact and efficient algorithms. We use the open source implementation of matrix profile[6]. To ensure the reproducibility of experiment, we upload all the experiment code and dataset on this website[7].

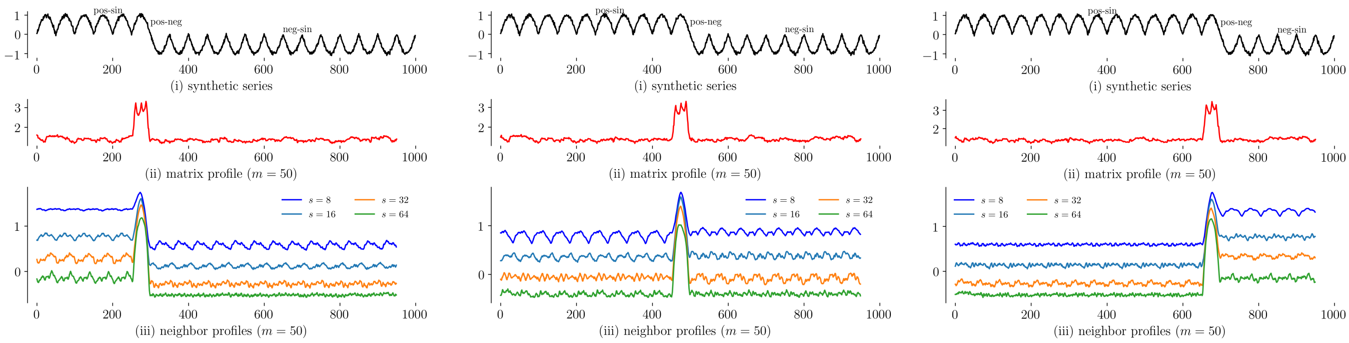
A. Density Modeling Evaluation (RQ1)

In this experiment, we qualitatively examine the characteristic of neighbor profile with a series of case studies on both synthetic data and real-world datasets.

1) On the Synthetic Data: It is supposed that for subsequences of different densities, neighbor profile can distinguish them effectively and correctly. To this end, we particularly design an experimental synthetic series, each of which consists of subsequences of three kinds with different quantities:

[6] https://github.com/TDAmeritrade/stumpy

[7] https://sites.google.com/view/neighbor-profile



(a) pos-sin (30%), pos-neg, neg-sin (70%).

(b) pos-sin (50%), pos-neg, neg-sin (50%).

(c) pos-sin (70%), pos-neg, neg-sin (30%).

Fig. 5. Density modeling on synthetic time series.

- pos-sin: the positive part of a single-period sine wave.
- neg-sin: the negative part of a single-period sine wave.
- pos-neg (or neg-pos): the transition from pos-sin (or neg-sin) to neg-sin (or pos-sin).

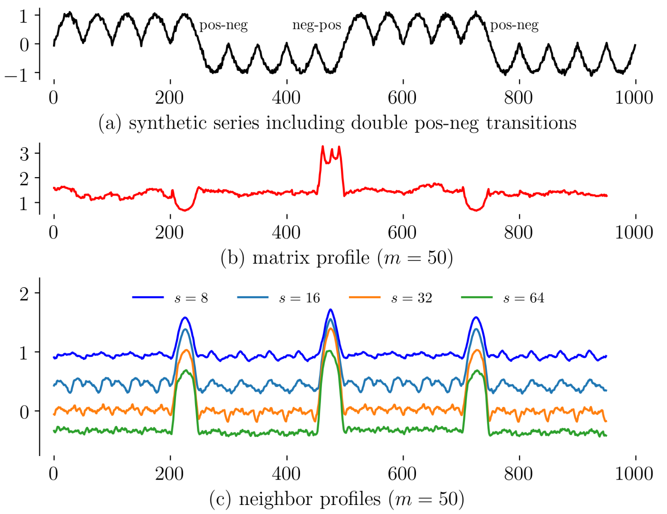
We further include white noise in the final synthetic series.

Varying subsequence densities. We experiment on the series composed of pos-sin, pos-neg and neg-sin with the results shown in Fig. 5a, 5b, 5c. We vary the quantities of pos-sin/neg-sin in the artificial series from $30\%/70\%$, $50\%/50\%$ to $70\%/30\%$ as the black series in the top. For all three series, the rarest subsequence is the pos-neg, while the most frequent ones are neg-sin, neg-sin/pos-sin, and pos-sin respectively. The series length is $1,000$ and the subsequence length of interest is set as 50.

Matrix profiles are depicted as the red series in the middle, which all correctly mark the most unusual subsequences as there only exists a single trans-sine in all the series. However, matrix profiles remain indistinguishable for pos-sin and neg-sin of different quantities, because a nearest neighbor with similarly small distance can always be discovered for each pos-sin or neg-sin regardless of their quantities. Thus, merely considering the nearest neighbor distance results in the failure of subsequence density modeling.

Neighbor profiles are shown at the bottom. The subsample size s varies from $8, 16, 32$ to 64. Similarly, they successfully capture the rare subsequence, pos-neg, as we can see the peaks in the corresponding positions. However, compared to matrix profile, it is able to clearly distinguish subsequences of different densities: the more pos-sin subsequences (or neg-sin) there are in the series, the lower the neighbor profile is. When the quantities are the same in Fig. 5b, the neighbor profiles of pos-sin and neg-sin become close with each other.

Hyperparameter settings. We further analyze how subsample size influences the modeling of subsequence density. As we can see, a wide range of subsample sizes from 8 to 64 allow a satisfying density modeling for subsequences. When s grows larger, the rare subsequence becomes more prominent while the others become similar, i.e. a smaller difference between neighbor profiles of pos-sin and neg-sin. This is consistent with the previous analysis that neighbor profile will be reduced to matrix profile as s approaches l. As s becomes smaller, it turns out that the most frequent subsequence becomes more prominent compared with the others, since a small s will lead to subsamples of only frequent subsequences. The above observations intuitively reveal how to choose subsample size for particular tasks: (1) when targeting on mining rare subsequences, consider a slightly larger s; (2) otherwise, consider a slightly smaller one.



(a) synthetic series including double pos-neg transitions

(b) matrix profile ($m = 50$)

(c) neighbor profiles ($m = 50$)

Fig. 6. Evaluation on twin freak problem.

Twin freak problem. We consider another synthetic series with double pos-neg (the rarer subsequence) and single neg-pos (the rarest subsequence) to reproduce a twin freak case in Fig. 6. Note that both pos-sin and neg-sin are the most frequent subsequences in this time series.

Matrix profile correctly identifies the neg-pos by the discord, which is the rarest subsequence. However, it mistakes the two pos-neg for even most frequent subsequences due to the freak twin problem, since the two pos-neg are so close with each other that the (nearest neighbor) distance between them is the smallest. On the contrary, all the neighbor profiles not only identify most rare subsequences but also correctly discover pos-sin and neg-sin as frequent subsequences and pos-neg as rarer subsequences. Particularly, when subsample size $s \geq 16$, the neighbor profile is effective to even distinguish the tiny density difference between neg-pos and pos-neg, i.e. a slightly higher neighbor profile of neg-pos than that of pos-neg. Therefore, by leveraging multiple subsamples, neighbor profile robustly models the density of subsequences and mitigates the twin freak problem effectively.

2) *On the Insect Electrical Penetration Graph (EPG):* The EPG describes the behavior of sucking sap from plants by Beet leafhopper. We consider mining frequent patterns to learn the regularity of this data, which has been studied in [28].

The whole series length is $33,021$ and the subsequence length of interest is 480. The subsample size is set as 16 and the distance is the normal Euclidean distance. The discovered frequent pattern by neighbor profile is shown in Fig. 7. According to [28], this subsequence is recognized to appear frequently, indicating that the plant sap ingestion by the insect.

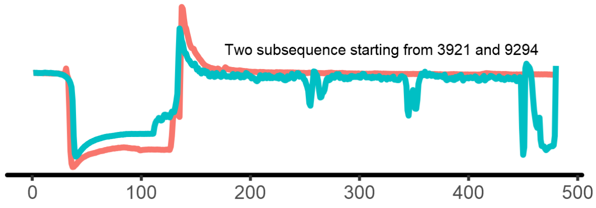


Fig. 7. The frequents subsequence discovered in the insect EPG.

3) *On the Electroencephalogram (EEG):* EEG detects electrical brain activity, which can be used for the diagnosis of

epilepsy and sleep disorders. We consider mining frequent patterns from EEG data, which has also been studied in [28].

The whole series length is $180,214$ and the subsequence length is set as 800. The subsample size is 8 and the distance is the normal Euclidean distance. Fig. 8 depicts the discovered the frequent subsequences, which is consistent with the results in [28]. Specifically, this subsequence contains a particular shape, i.e. K-complex, which is a frequent pattern during the sleep and can be used for identifying the sleep stage [29].

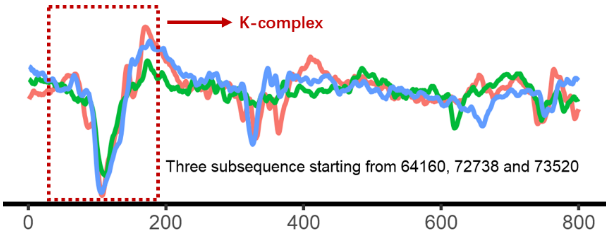


Fig. 8. The frequent subsequences discovered in the EEG.

4) On the Power Consumption Series: We consider mining unusual subsequences in the power consumption data, which describes the power consumption of a Dutch research facility of the year of 1997 [30]. The length of this series is $35,040$ and the subsequence length is 750, which captures a week of power consumption. The subsample size is 32 and the distance is the zscore Euclidean distance. The top 3 rarest subsequences are shown in Fig. 9. Although the found rare subsequences by neighbor profile are not exactly the same with the ones by discord [30], they all capture the irregular weeks, which contains special holidays.

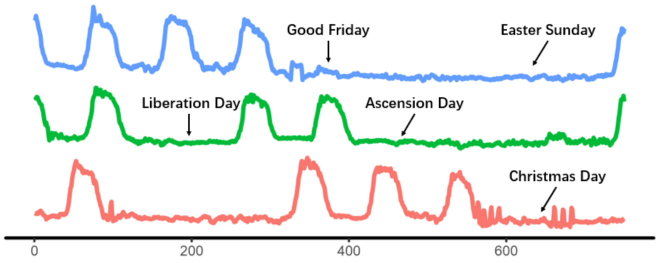


Fig. 9. The rare subsequences discovered in the power series.

B. Quantitative Evaluation on Real-World Dataset (RQ2)

Existing works mainly adopt a case study to show the successfully mined frequent and rare subsequences [15], [28], [31]. In this experiment, however, we perform a quantitative study to compare neighbor profile with existing definitions: we consider binary classifications by examining whether or not the discovered frequent and rare subsequences are the truly frequent and rare subsequences in the series.

1) Dataset and settings: We consider the widely studied real-world dataset, MIT-BIH Arrhythmia Database [7], for this evaluation. Though a single dataset, it contains as many as 96 (48 2-channel) half-hour ECG excerpts from subjects with various conditions. Furthermore, it has been fully annotated, from which we can learn the truly frequent and rare subsequences in the time series for a quantitative evaluation. The sampling rate is 360Hz and the average series length is about 648,000. The subsequence length of interest is set as 180 (0.5s), which is appropriate for capturing a single beat. The hyperparameters of neighbor profile are similar with previous experiments: the subsample size $s \in \{8, 16, 32, 64\}$. Also, we introduce three distance measures, i.e. normal/demean/zscore Euclidean distances, to evaluate their performances in real-world datasets of different kinds.

We consider three experiments: (1) rare subsequence mining, (2) rare subsequence mining based on a model learned from a healthy subject, and (3) frequent subsequence mining, respectively. For each series in each experiment, every subsequence $\mathbf{y}$ is assigned two values $np(\mathbf{y})$ and $mp(\mathbf{y})$, indicating the estimated densities. With the groundtruth labels $\in \{0, 1\}$ of $\mathbf{y}$ from the annotations, we can use AUC to evaluate the estimated results and compare the density estimations by neighbor profile and matrix profile, which is similar with the evaluation of binary classification. A higher AUC indicates a better performance of frequent subsequence mining. However, since this dataset contains as many as 96 series, we omit the AUC results here due to the space limitation. Instead, we directly show the comparison results, i.e. the number of win/tie/lose on the total series, for each experiment. The Wilcoxon signed-rank test is also performed to evaluate the significance of multiple comparisons. For detailed implementations and AUC results, please see the website.

2) Mining rare subsequence: We formulate a binary classification problem: one class is focused as the beat of *premature ventricular contraction* (PVC) while the other class is the rest. PVC is a medically abnormal beat, which is also rare or infrequent in the arrhythmia database. Only 74 recordings contain PVC, resulting in 74 valid problems.

The experimental results are summarized in Table I. It can be seen that neighbor profile outperforms matrix profile *significantly* under 10 out of 12 settings. Demean distance performs the best since it can remove the vertical shifting in the signals. On the contrary, zscore distance is worse than the other distances. Particularly, when $s = 8$ or 16, zscore distance based neighbor profile is worse than matrix profile, which suggests that zscore distance is not suitable for rare sub-

TABLE I
COMPARISON OF AUCS BETWEEN NEIGHBOR PROFILE AND MATRIX PROFILE ON 74 VALID PROBLEMS FOR RARE SUBSEQUENCE MINING.

Distance	Subsample Size s			
	$s = 8$	$s = 16$	$s = 32$	$s = 64$
normal	**47**/1/26*	**48**/2/24*	**53**/2/19*	**55**/2/17*
demean	**48**/2/24*	**50**/2/22*	**58**/2/14*	**62**/2/10*
zscore	21/1/**52***	34/1/**39**	**44**/2/28*	**53**/1/20*
Best	**68**/2/4*			

$*$ indicates a Wilcoxon signed-rank test with p-value < 0.05.

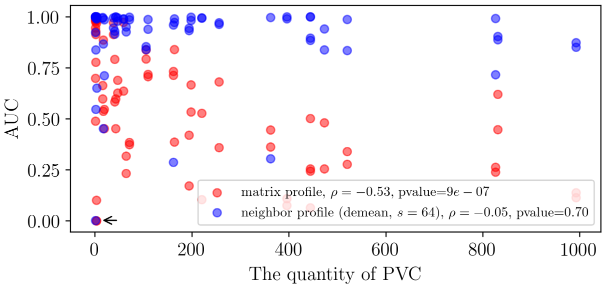


Fig. 10. AUC v.s. quantity of rare subsequences.

TABLE II
COMPARISON OF AUCS BETWEEN NEIGHBOR PROFILE AND MATRIX PROFILE ON 35 VALID PROBLEMS FOR RARE SUBSEQUENCE MINING WITH THE MODEL TRAINED ON RECORD 122 CHANNEL MLII.

Distance	Subsample Size s			
	$s = 8$	$s = 16$	$s = 32$	$s = 64$
normal	**25**/0/10*	**25**/0/10*	**25**/0/10*	**25**/0/10*
demean	**28**/0/7*	**27**/0/8*	**29**/0/6*	**31**/1/3*
zscore	7/0/**28***	19/0/**16**	**25**/0/15	**19**/0/16
Best	**34**/0/1*			

$*$ indicates a Wilcoxon signed-rank test with p-value < 0.05.

sequences mining in this dataset. It can be easily understood that zscore distance changes the subsequence significantly by normalizing its variance as 1, which usually decreases the difference between infrequent and normal subsequences.

We further study the twin freak problem when mining rare subsequences by investigating the relationship between the mining performance and the quantity of abnormal subsequences. Particularly, we compare matrix profile and neighbor profile (demean, $s = 64$) in Fig. 10. It can be clearly seen that as the quantity of PVC rises, the performance of matrix profile deteriorates while neighbor profile remains about the same. The Pearson correlation coefficient of matrix profile suggests a *significant* (pvalue $\ll 0.05$) negative relationship ($\rho = -0.53$), which indicates that the performance of matrix profile is seriously affected by the twin freak problems. On the contrary, neighbor profile generally performs well regardless of the quantity of rare subsequences with an insignificant $\rho = -0.04$, which is in accordance with the experimental results on the synthetic sine series.

It is worth mentioning an intriguing result of neighbor profile pointed by an arrow in the lower-left corner of Fig. 10. This AUC is close to 0, indicating a very bad mining performance (even worse than random guess). With a close look at the corresponding time series, i.e. subject 121 channel *V1*, we notice that it contains a single abnormal subsequence yet with a quite close distance to normal noisy subsequences, making neighbor profile fail to recognize it as unusual subsequence. Fortunately, it can be simply avoided using zscore distance by retaining the main shape and reducing the influence of noise. The AUC of neighbor profile (zscore, $s = 64$) is as high as 0.9264. This suggests the importance of choosing an appropriate similarity measurement for particular tasks.

3) Mining rare subsequence with a pre-trained model: In this experiment, we evaluate the effectiveness of neighbor profile for anomaly detection with an existing model. This is quite useful in the real-world data mining tasks. When subsequences of a time series form a distribution where real anomalies occur more often than normal subsequences (e.g. an ECG signal from a patient with serious heart problems), mining rare or frequent subsequence from this single time series will result in a discovery contradicting the common sense: abnormal subsequence is discovered as frequent subsequence. To avoid this, we may train a model from the dataset with a standard distribution and use this model for subsequence mining. Particularly, we choose record 122 channel MLII to build the model, since this signal only contains normal heart beat, which is appropriate for detecting abnormal beats. Similarly, we still focus on mining PVC abnormal beats and consider 12 settings for neighbor profile. As for matrix profile, it has to use the whole signal (length: $649,800$) itself as the model to perform two time series join for detecting PVC. The number of valid problem is 35, i.e. channel MILL with PVC.

The AUC of PVC detection for both methods are listed in Table II. Apart from the settings with zscore distance, neighbor profile has a significantly better performance than matrix profile. This indicates that neighbor profile with normal or demean distance successfully captures the distribution of normal beats for an effective anomaly detection. However, zscore distance has a rather bad performance for PVC detection. Specifically, when subsample size is quite small ($s = 8$), it is significantly worse than matrix profile. As s grows, neighbor profile becomes better. This phenomenon is consistent with the previous experiment for rare subsequence mining.

It is worth mentioning that neighbor profile builds a competitive model with only a small subset of the original time series, which is efficient in both storage and computation. Furthermore, due to the gravity defiant behavior, neighbor profile with only 8×100 subsequences and normal distance is able to outperform matrix profile significantly.

4) Mining frequent subsequence: Mining frequent subsequence from ECG can be used for a quick judgement of health condition. Similar with the first experiment, we formulate another binary classification problem: the most frequent annotation is considered one class and the rest are the other class. The most frequent subsequence varies with ECG recordings, including *normal beat*, *left/right bundle branch block beat*, etc. Four excerpts contain only a single kind of heart beat. Thus, the valid number of ECG recordings is 92.

We compare the AUCs of neighbor profile and matrix profile on total 92 problems, which is summarized in Table III. From the table, we can clearly see that under a wide range of settings, neighbor profile has an overall superior performance over matrix profile for identifying frequent subsequences. In 7 out of 12 settings, neighbor profile is significantly better than matrix profile. In addition, demean and zscore distances show relatively better results than normal distance, since they can

TABLE III

COMPARISON OF AUCS BETWEEN NEIGHBOR PROFILE AND MATRIX PROFILE ON 92 VALID PROBLEMS FOR FREQUENT SUBSEQUENCE MINING.

Distance	Subsample Size s			
	$s = 8$	$s = 16$	$s = 32$	$s = 64$
normal	**50**/0/42	**52**/0/40	**51**/0/41	**52**/0/40*
demean	**57**/0/35	**58**/0/34*	**60**/0/32*	**58**/0/34*
zscore	46/0/46	**57**/0/35*	**54**/0/38*	**64**/0/28*
Best	**80**/0/12*			

$*$ indicates a Wilcoxon signed-rank test with p-value < 0.05.

remove the shifting or scaling noise in the time series.

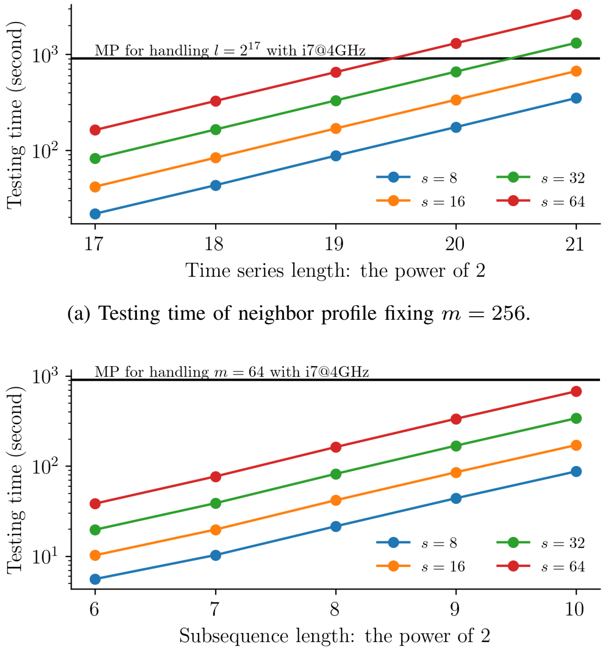
Compared to matrix profile, which is parameter free, different parameter settings (i.e. distance function and subsample size) render the flexibility to neighbor profile for different data mining tasks. For example, when the settings are chosen as the best one for the 92 problems given current parameter grid, neighbor profile can win as many as 80 problems. This may be done through a heuristic way by trying and tuning, when the label information is available. A potential issue is that searching for an appropriate setting may cost a lot of time. However, as we will see later in section IV-C, neighbor profile is extremely efficient for a tractable hyperparameter tuning.

C. Time Efficiency Evaluation (RQ3)

In this experiment, we focus on the study of time efficiency of neighbor profile. The time complexity of neighbor profile is $\mathcal{O}(nmsl)$ while matrix profile is $\mathcal{O}(l^2 \log l)$. Theoretically, when $l \gg s$, neighbor profile is usually more efficient than matrix profile, since neighbor profile only uses a subset of all subsequences for data mining. However, it has been shown that in the real implementation, matrix profile is able to handle millions of data efficiently with the help of existing highly optimized FFT procedures and GPU device [23]. Thus, the experimental philosophy here is to *qualitatively* compare neighbor profile with matrix profile and show that neighbor profile can *also* efficiently handle massive data.

We implement alg. 1 in Python. Particularly, the most time consuming pairwise distance calculation is realized in batch to leverage the existing fast procedure *scipy.spatial.distance*. We perform two experiments with varying random series length $l \in \{2^{17}, 2^{18}, 2^{19}, 2^{20}, 2^{21}\}$ and subsequence length $m \in \{64, 128, 256, 512, 1024\}$. For both experiments, subsample sizes $s \in \{8, 16, 32, 64\}$ are also considered. The wall clock time is measured on an Intel i9@3.6GHz. The training time is omitted since they are all less than 0.2 second.

The results in Fig. 11 clearly show the linear growth w.r.t l, m, s, which confirms the time complexity analysis. We also directly use the experiment results from [15] as the black lines for a qualitative comparison. Though implemented and experimented in different environments, it shows that neighbor profile shows a comparable or even better performance than matrix profile. For example, the time required by matrix profile to handle $l = 2^{17}$ is already enough for neighbor profile to process $l = 2^{19}, s = 64$. The time efficiency of neighbor profile can be further improved through simple parallelism. A potential issue may be a too large s. However, as shown before, a smaller s can usually lead to a satisfying performance.



(a) Testing time of neighbor profile fixing $m = 256$.

(b) Testing time of neighbor profile fixing $l = 2^{17}$.

Fig. 11. Testing time of neighbor profile.

We compare the time consumption of neighbor profile and matrix profile for real-world data mining tasks in section[8] IV-B, which is summarized in TABLE IV. These experiments are also conducted on the same CPU. The average time series length is $650,000$ and the subsequence length of interest is 180. We can see that there are no significant differences among neighbor profiles with different distance measurements. Also, neighbor profile under current settings is much more efficient than matrix profile, since it can robustly estimate the density of each subsequence with only a small subset ($\sim 0.1\%$) of all subsequences.

TABLE IV

AVERAGE REQUIRED TIME OF NEIGHBOR PROFILE AND MATRIX PROFILE

Distance	Subsample Size s			
	$s = 8$	$s = 16$	$s = 32$	$s = 64$
normal	78.46s	142.74s	279.30s	555.09s
demean	79.40s	143.55s	280.83s	555.37s
zscore	80.58s	144.45s	281.18s	556.46s
Matrix Profile	1140.11s			

Last, it is worth discussing the computing issue for long subsequences: the time complexity of alg. 1 grows linearly with subsequence length m, which may result in an intractable computing when s is too large. Existing works on motif

[8] The time consumption of experiments in section **??** are omitted since the lengths are far less than those of the ECG excerpts.

and discord shows two ways to enhance the computing of neighbor profile [4], [23]: (1) symbolization techniques for the reduction of the data quantity for calculation and (2) GPU device for the boosting of matrix calculation. However, the further enhancement of neighbor profile is out of the scope of this paper, which will be addressed in the future works.

V. Related Work

In this section, we review related work for frequent and rare subsequence mining.

A. Motif and Discord

Frequent pattern mining and anomaly detection are two major data mining tasks [32]. In the literature of time series data mining, motif [31] and discord [3], as the widely accepted implementations of frequently occurring patterns and rare subsequences, have been attracting great attention and extensively studied in the community.

Motifs are considered *frequently occurring patterns* in a time series, which may represent meaningful knowledge in the time series. There are two kinds of definitions for motif, i.e. similarity-based and support-based[9] [8]. In this work, we have considered the former one (definition 6) since it is widely adopted due to its simplicity for implementation [15], [28], [33]. Since its introduction in 2002 [31], a large number of algorithms have been developed to speed up the discovery of motifs [6], [28], [33], [34]. It has also been applied in the applications of sensor, trajectory, etc [35], [36]. For more details, please refer to the two surveys [8], [37].

In contrast to motifs, discords are considered *most unusual subsequences* in a time series, which have been used to detecting anomalies in ECG, astronomy, building sensor data, etc [3]–[5], [38]. Most of existing works focus on increasing the mining efficiency of discord by leveraging symbolization, clustering, wavelet transformation, etc [3], [4], [27], [39]. Huang et al. firstly make an effort to explicitly handle the twin freak problem by considering j distance discord [12]. However, as we point out, choosing an appropriate k is not trivial, which may incur a lot of overhead. Besides, it still suffers from the gravity defiant behavior as a nearest neighbor based outlier detector, which may result in a degraded performance [13].

B. Matrix Profile

Matrix profile unifies the discovery of motif and discord in the same computing framework, which has drawn great attention in the community[10] [15]. It mainly provides a bundle of algorithms to efficiently learn matrix profile under different situations. At the heart of these algorithms is the fast calculation of pairwise distances for S_T^m based on Fast Fourier Transformation (FFT) [24], since motif and discord are both defined by 1-nearest neighbor distance.

In addition to these algorithms, it has also been explored by researchers to extend matrix profile to mine subsequences under different requirements. For example, it has been shown to incorporate domain knowledge into matrix profile for knowledge-guided subsequence mining [40].

Although matrix profile has achieved success in several cases, it still adopts the very early definitions, i.e. motif and discord, which deeply undermines the practical performance of matrix profile. Also, the extensions based on matrix profile are inevitably degraded due to the three inherent issues. Neighbor profile, however, aims at the more fundamental problem about the definitions of frequent and rare subsequences, and addresses current issues by leveraging bagged nearest neighbors. The experiment shows that the novel definitions lead to robust estimations for a better mining performance, and the algorithm naively developed from neighbor profile is already comparably efficient. Besides, unlike matrix profile, which integrates with zscore Euclidean distance, neighbor profile is flexible to allow a variety of distance measures to improve the performance for different kinds of mining tasks.

VI. Conclusion and Future Work

In this work, we present a study on the unsupervised time series mining for frequent and rare subsequences. We summarize the extensively studied implementations, i.e. motif and discord, and provide a theoretical view for matrix profile as the 1-nearest neighbor based nonparametric density estimation. Given that, we demonstrate the inherent three issues with matrix profile. To address these issues, we develop neighbor profile to robustly perform subsequence density estimation by leveraging bagging and nearest neighbors. The experiments on both synthetic and real-world datasets show the promising performance of neighbor profile.

However, neighbor profile is not the final solution for unsupervised time series mining. We point out the future research directions as follows. First, though robust, neighbor profile only models the global density. Thus, frequent and rare subsequences are identified by the absolute order of estimated densities among all the subsequences. However, as many outlier detectors are designed to discover local anomaly [41]–[43], it is also a need to extend neighbor profile to mining local frequent and rare subsequences. Second, it is also promising to adapt neighbor profile to different data mining tasks, such as mining from multi-dimensional time series, incorporating domain knowledge into mining, interactive mining with data miners, etc. A better performance than those based on matrix profile could be expected as the result of the robust subsequence density estimation by neighbor profile. Last, as is discussed before, it is worth further improving the computing efficiency of neighbor profile for mining long subsequences.

Acknowledgment

This work is supported by National Natural Science Foundation of China (61772045). We are grateful for all the data providers. Particularly, we would like to thank Chaohe Zhang as well as all the reviewers for their valuable help and suggestions to this work.

[9] For interested readers, support-based definition actually corresponds to the kernel density estimation.

[10] For details, see https://www.cs.ucr.edu/∼eamonn/MatrixProfile.html.

References

[1] T.-c. Fu, “A review on time series data mining,” *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164–181, 2011.

[2] P. Esling and C. Agon, “Time-series data mining,” *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, p. 12, 2012.

[3] E. Keogh, J. Lin, and A. Fu, “Hot sax: efficiently finding the most unusual time series subsequence,” in *Fifth IEEE International Conference on Data Mining (ICDM’05)*, Nov 2005, pp. 8 pp.–.

[4] Y. Bu, T.-W. Leung, A. W.-C. Fu, E. Keogh, J. Pei, and S. Meshkin, “Wat: Finding top-k discords in time series database,” in *Proceedings of the 2007 SIAM International Conference on Data Mining*. SIAM, 2007, pp. 449–454.

[5] D. Yankov, E. Keogh, and U. Rebbapragada, “Disk aware discord discovery: Finding unusual time series in terabyte sized datasets,” *Knowledge and Information Systems*, vol. 17, no. 2, pp. 241–262, 2008.

[6] P. Patel, E. Keogh, J. Lin, and S. Lonardi, “Mining motifs in massive time series databases,” in *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* IEEE, 2002, pp. 370–377.

[7] G. B. Moody and R. G. Mark, “The impact of the mit-bih arrhythmia database,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[8] A. Mueen, “Time series motif discovery: dimensions and applications,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 2, pp. 152–159, 2014.

[9] D. O. Loftsgaarden, C. P. Quesenberry *et al.*, “A nonparametric estimate of a multivariate density function,” *The Annals of Mathematical Statistics*, vol. 36, no. 3, pp. 1049–1051, 1965.

[10] B. W. Silverman, *Density estimation for statistics and data analysis*. Routledge, 2018.

[11] L. Wei, E. Keogh, and X. Xi, “Saxually explicit images: Finding unusual shapes,” in *Sixth International Conference on Data Mining (ICDM’06)*. IEEE, 2006, pp. 711–720.

[12] T. Huang, Y. Zhu, Y. Wu, and W. Shi, “J-distance discord: an improved time series discord definition and discovery method,” in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE, 2015, pp. 303–310.

[13] K. M. Ting, T. Washio, J. R. Wells, and S. Aryal, “Defying the gravity of learning curve: a characteristic of nearest neighbour anomaly detectors,” *Machine learning*, vol. 106, no. 1, pp. 55–91, 2017.

[14] A. Zimek, M. Gaudet, R. J. Campello, and J. Sander, “Subsampling for efficient and effective unsupervised outlier detection ensembles,” in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 428–436.

[15] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh, “Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets,” in *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 2016, pp. 1317–1322.

[16] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[17] E. Parzen, “On estimation of a probability density function and mode,” *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[18] E. M. Knorr and R. T. Ng, “A unified notion of outliers: Properties and computation.” in *KDD*, vol. 97, 1997, pp. 219–222.

[19] S. Ramaswamy, R. Rastogi, and K. Shim, “Efficient algorithms for mining outliers from large data sets,” in *ACM Sigmod Record*, vol. 29, no. 2. ACM, 2000, pp. 427–438.

[20] F. Angiulli and C. Pizzuti, “Fast outlier detection in high dimensional spaces,” in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2002, pp. 15–27.

[21] M. Sugiyama and K. Borgwardt, “Rapid distance-based outlier detection via sampling,” in *Advances in Neural Information Processing Systems*, 2013, pp. 467–475.

[22] C. C. Aggarwal and S. Sathe, “Theoretical foundations and algorithms for outlier ensembles,” *Acm Sigkdd Explorations Newsletter*, vol. 17, no. 1, pp. 24–47, 2015.

[23] Y. Zhu, Z. Zimmerman, N. S. Senobari, C.-C. M. Yeh, G. Funning, A. Mueen, P. Brisk, and E. Keogh, “Matrix profile ii: Exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins,” in *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 2016, pp. 739–748.

[24] A. Mueen, Y. Zhu, M. Yeh, K. Kamgar, K. Viswanathan, C. Gupta, and E. Keogh, “The fastest similarity search algorithm for time series subsequences under euclidean distance,” August 2017, http://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html.

[25] L. Breiman, “Bagging predictors,” *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[26] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[27] A. W.-c. Fu, O. T.-W. Leung, E. Keogh, and J. Lin, “Finding time series discords based on haar transform,” in *Advanced Data Mining and Applications*, X. Li, O. R. Zaïane, and Z. Li, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 31–41.

[28] A. Mueen, E. Keogh, Q. Zhu, S. Cash, and B. Westover, “Exact discovery of time series motifs,” in *Proceedings of the 2009 SIAM international conference on data mining*. SIAM, 2009, pp. 473–484.

[29] R. B. Berry, R. Brooks, C. E. Gamaldo, S. M. Harding, C. Marcus, B. V. Vaughn *et al.*, “The aasm manual for the scoring of sleep and associated events,” *Rules, Terminology and Technical Specifications, Darien, Illinois, American Academy of Sleep Medicine*, vol. 176, 2012.

[30] E. Keogh, J. Lin, S.-H. Lee, and H. Van Herle, “Finding the most unusual time series subsequence: algorithms and applications,” *Knowledge and Information Systems*, vol. 11, no. 1, pp. 1–27, 2007.

[31] J. Lonardi and P. Patel, “Finding motifs in time series,” in *Proc. of the 2nd Workshop on Temporal Data Mining*, 2002, pp. 53–68.

[32] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[33] A. Mueen and N. Chavoshi, “Enumeration of time series motifs of all lengths,” *Knowledge and Information Systems*, vol. 45, no. 1, pp. 105–132, 2015.

[34] B. Chiu, E. Keogh, and S. Lonardi, “Probabilistic discovery of time series motifs,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 493–498.

[35] A. Vahdatpour, N. Amini, and M. Sarrafzadeh, “Toward unsupervised activity discovery using multi dimensional motif detection in time series,” in *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[36] T. Oates, A. P. Boedihardjo, J. Lin, C. Chen, S. Frankenstein, and S. Gandhi, “Motif discovery in spatial trajectories using grammar inference,” in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2013, pp. 1465–1468.

[37] S. Torkamani and V. Lohweg, “Survey on time series motif discovery,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 7, no. 2, p. e1199, 2017.

[38] C. Miller, Z. Nagy, and A. Schlueter, “Automated daily pattern filtering of measured building performance data,” *Automation in Construction*, vol. 49, pp. 1–17, 01 2015.

[39] G. Li, O. Brysy, L. Jiang, Z. Wu, and Y. Wang, “Finding time series discord based on bit representation clustering,” *Knowledge-Based Systems*, vol. 54, pp. 243 – 254, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705113002931

[40] H. A. Dau and E. Keogh, “Matrix profile v: A generic technique to incorporate domain knowledge into motif discovery,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 125–134.

[41] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: identifying density-based local outliers,” in *ACM sigmod record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.

[42] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, “Enhancing effectiveness of outlier detections for low density patterns,” in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2002, pp. 535–548.

[43] T. R. Bandaragoda, K. M. Ting, D. Albrecht, F. T. Liu, Y. Zhu, and J. R. Wells, “Isolation-based anomaly detection using nearest-neighbor ensembles,” *Computational Intelligence*, vol. 34, no. 4, pp. 968–998, 2018.