Periodicity Testing with Sublinear Samples and Space

Funda Ergun [*] S. Muthukrishnan [†] S. Cenk Sahinalp [‡]

Abstract

In this work, we are interested in finding representative trends in long large data streams in the presence of computational constraints; to this end we present algorithms for discovering periodic trends in a data stream S of length n in the combinatorial property testing model, using $o(n)$ samples and space.

In accordance with the property testing model, we first explore the notion of being close to periodic by we relaxing different notions of exact periodicity and obtaining three different notions of self-distance. An input S is then approximately periodic if it exhibits a small self-distance (with respect to any one self-distance defined). We show that even though the different definitions of exact periodicity are equivalent, the resulting definitions of self-distance, approximate periodicity, are not. A close investigation of the self-distances show that they are constant approximations of each other. We then present algorithms which distinguish between the two cases of S being exactly periodic and S being far from periodic with only a constant probability of error. Our algorithms need to sample only $O(\sqrt{n}$ polylog $n)$ positions and use as much space. They can also be used to find, using $o(n)$ samples and space, the largest/smallest period, andn/or all of the approximate periods of S.

Our algorithms may be viewed as working on streaming inputs where each data item is seen once and in order, storing only a sublinear – $O(\sqrt{n}$ polylog $n)$ – size sample from which periodicities are identified.

[*]School of Computing Science, Simon Fraser University. `fundacs.sfu.ca`.

[†]Department of Computer Science, Rutgers University. `muthu@cs.rutgers.edu`.

[‡]School of Computing Science, Simon Fraser Univeristy. `cenk@cs.sfu.ca`.

1 Introduction

There is an abundance of time series data collected by a varying and ever-increasing set of current applications. For instance, telecommunications companies collect traffic information in the form of number of calls, dropped calls, bytes sent, connections established etc. at each of their network links at small intervals. Such collections of data tend to be very large and usually exhibit very distinct trends. The information obtained by analyzing these trends is used for purposes such as making business decisions, forecasting and sizing. Similarly, time series data is crucially used in decision support systems in many arenas including finance, weather prediction, and network management. Time series data management involves processing these massive data sets to discover various events and patterns.

In this paper, we are motivated by applications where the data is analyzed for discovering self-similar trends; in particular, we study the *periodicity* of data. In real life, no time series is exactly periodic; i.e., repetition of a single pattern over and over again does not occur. For example, the number of bytes sent over an IP link in a network is almost surely not a perfect repeat of a daily, weekly or a monthly trend. However, many time series data are likely to be "approximately" periodic; i.e., a data pattern may repeat over and over every time with small changes. The main objective of this paper is to determine if a time series data stream is *approximately* periodic.

Given the large size of most time series data, it is not always realistic to assume that one will have the necessary computational resources to store, and process the entire data to obtain precise results. As a result, we would like to explore the discovery of these trends under computational limitations. Most significantly, since the type of data that we consider tends to be a rather long stream, we assume that it does not fit into the main memory in its entirety and focus on discovering periodic trends by looking at a small portion of the input. Our approach is based on combinatorial pattern matching and random sampling – we define notions of approximate periodicity and explore methods for discovering approximate periodic behavior of time series data streams using a small number of random bits from the input; the behavior of our algorithm is specified within the combinatorial property testing context [5, 3]. [1]

Our main contributions in this paper are as follows.

1. By generalizing equivalent definitions of the *exact* periodicity of a data stream, we obtain three notions of *self distance*, each representing the self-similarity of the stream with respect to a particular notion of exact periodicity. Our distances are either in terms of the distance between the stream and an appropriately shifted version of itself, or in terms of the distance between different portions of the stream. We use these distances to relax the notion of periodicity: data streams with small self-distance are deemed to be approximately periodic (with respect to the particular distance used).

2. We present efficient algorithms for determining if the input data stream is approximately periodic with respect to any one particular self-distance. Our algorithms rely only on sampling a sublinear — $O(\sqrt{n}$ polylog $n)$— number of positions in the input.

[1] The limitations on space and the number of characters that we can sample from the input turn out to be crucial for our algorithms, linear space and samples would make our efforts for re-using samples redundant and our algorithms would be much simpler.

3. We show that our notions of approximate self-distance, as well as the notions of approximate periodicity that they induce, are closely related.

In searching for periodic trends, we make no initial assumptions as to what the period might be if the stream is to be found approximately periodic. In fact, we assume that the period could be as short as a single character or as long as half the size of the stream. Our algorithms obtain a small pool of random samples from the input initially, and use this pool to discover the periodicities. We show that there is always a subset of this pool sufficiently large to compute the self-distance under any potential period. In this scheme, a sample is typically used in determining whether the input is approximately p-periodic for multiple values p. For two notions of approximate periodicity we define here, our methods are quite simple; for the third notion, the sampling (in Section 3.1) is more involved with two stages where the second stage depends on the first.

Related Work Algorithmic literature on time series data analysis mostly focuses on indexing and searching problems based on various distance measures amongst multiple time series data. Common distance measures are L_p norms, hierarchical distances motivated by wavelets, etc. [2]

Although most available papers do not consider the combinatorial periodicity notions we explore here, one relevant paper [6] aims to find "average period" of a given time series data in a combinatorial fashion. This paper describes $O(n \log n)$ space algorithms to estimate average periods by using *sketches*.

Our work here deviates from that in [6] in a number of ways. First, we present the first known $o(n)$, in fact, $O(\sqrt{n} \cdot \text{polylog } n)$ space algorithm for periodic trend analysis in contrast to the $\omega(n)$ space methods in [6]. We do not know of a way which partitions the input and uses sketches of *all* the partitions as in [6] which yields the guarantees that we obtain. Rather, sampling seems to be ideal for us here; with a small number of samples we are able to perform computations for multiple period lengths–this can be viewed as sketching a small portion of the input. Second, we consider more general periodic trends than those in [6].

Sampling algorithms are known for computing Fourier coefficients with sublinear space [2], which can be used in the determining of approximate periods. However the algorithm in [2] is quite complex and expensive, using $(B \log n)^{O(1)}$ samples for finding B significant periodic components – the $O(1)$ factor is also rather large. In general, there is a rich theory of sampling in time series data analysis [10, 9]; our work is interesting in the way that it recycles random samples among multiple computations, and adds to this growing knowledge. Our methods are more akin to sublinear methods for property testing. In particular, in parallel with this work, authors in [1] present sublinear sampling methods for testing whether the edit distance between two strings is at least linear or at most n^α for $\alpha < 1$ by obtaining a directed sample set where the queries are at times evenly spaced within the strings.

[2] A survey is in the tutorial offered at KDD 2000 [7]; see also [8].

2 Notions of Approximate Periodicity

Our definitions of approximate periodicity are based on the notion of *exact periodicity* from combinatorial pattern matching. We will first review that notion before presenting our main results.

Let $S \in \Sigma^n$ denote a time series data stream where Σ denotes a constant size alphabet. $S[i]$ refers to the ith character of S and $S[i:j]$ to the segment $S[i]S[i+1]\ldots S[j]$. The size p prefix (resp. suffix) of S is the segment $S[1:p]$ (resp. $S[n-p+1, n]$). It is folklore that the exact periodicity of a data stream S with respect to a period of size p can be described in two alternative but equivalent ways as follows.

Definition 1 *A data stream $S \in \Sigma^n$ is said to be exactly p-periodic if for some $p \in \{1, .., \lfloor n/2 \rfloor\}$ either*

a. The size $n-p$ suffix and size $n-p$ prefix of S are identical; i.e., $S[1:n-p] = S[p+1:n]$, or, alternatively,

b. S consists of repetitions of the same block B of size p; i.e. $S = B^k B'$ where $B \in \Sigma^p$, B' is a prefix of B and $k = \lfloor n/p \rfloor$.

When examining the p-periodicity of a data stream S, we denote by b_i^p the ith block of S of size p, that is, $S[(i-1)p+1 : ip]$. Notice that $S = b_1^p, b_2^p, \ldots b_k^p, b'$ where $k = \lfloor n/p \rfloor$ and b' is the length $n - kp$ suffix of S. When the choice of p is clear from the context, we drop it; i.e. we write $S = b_1, b_2, \ldots b_k, b'$. For simplicity, unless otherwise noted, we assume that the stream consists of a whole number of blocks, i.e., $n = kp$ for some $k > 0$, for any p under consideration. If an unfinished block at the end of the stream is padded with *don't care* symbols until the desired format is obtained; our proofs still remain valid.

2.1 Self Distances and Approximate Periodicity

The above definitions of exact periodicity can be relaxed into a notion of approximate periodicity. Intuitively, a data stream S can be considered approximately periodic if it can be made exactly periodic by changing a small number of its entries. To formally define approximate periodicity, we present the notion of a "self-distance" for a data stream. We will call a stream S approximately periodic if its self-distance is small. We will also have that a stream S is exactly periodic if its self distance is zero. In what follows we introduce three self-distance measures, (*shiftwise, blockwise* and *pairwise* distances, denoted respectively as σ^p, β^p and π^p) each of which is defined with respect to a "base" distance between two streams. We will first focus on the Hamming distance $h()$ as the base distance for all three measures and subsequently discuss how to generalize our methods to other base distances.

2.1.1 Shiftwise Self Distance

We will relax Definition (a) of exact periodicity to obtain what we call the *shiftwise self-distance* of a data stream. As a preliminary step we define a simple notion of self-distance that we call the *single-shift self-distance* as follows.

Definition 2 *The single-shift self-distance of a data stream S with respect to period size p is $\rho^p(S) = h(S[p+1:n], S[1:n-p])$.*

If one assumes for the sake of simplicity that $n = kp$, then it is possible to write $S = b_1^p b_2^p \dots b_k^p$, and alternatively define the single-shift self-distance of S as $\rho^p(S) = \sum_{i=1}^{k-1} h(b_i^p, b_{i+1}^p)$. Note that S is exactly p-periodic if and only if $\rho^p(S) = 0$.

Unfortunately the single-shift self-distance of S fails to provide a satisfactory basis for approximate periodicity. A small $\rho^p(S)$ does not necessarily imply that S can be made exactly p-periodic by changing a small number of its entries: Let $p = 1$ and $S = 00000000001111111111$. It is easy to see that $\rho^1(S) = 1$. However, to make S periodic with $p = 1$ (in fact with *any* p) one needs to change a linear number of entries of S.

Even though S is "self similar" under $\rho^1()$, it is far from being exactly periodic as stipulated in Definition 1 (b). Thus while Definition 1 (a) and (b) are equivalent in the context of exact periodicity, that one stream is "close" (with respect to $\rho()$) to satisfying (a) does not necessarily mean that it is close to satisfying (b). Our goal here is to define a self-distance from each definition of periodicity which will give us a notion of approximate periodicity. If we use $\rho()$ as such a distance, we run the risk that the two notions of approximate periodicity obtained from (a) and (b) will be too different.

To make our notion of single-shift self-distance of S a more robust measure of self-similarity, we now incorporate more information into our understanding of self-similarity. Observe that if a data stream S is exactly p-periodic, it is also exactly $2p$-, $3p$-, ... periodic; i.e., if $\rho^p(S) = 0$, then $\rho^{2p}(S) = \rho^{3p}(S) = \dots = 0$. However, when $\rho^p(S) = \ell > 0$ one cannot say much about $\rho^{2p}(S), \rho^{3p}(S), \dots$ in relation to ℓ. In fact, given S and p, $\rho^{ip}(S)$ can grow linearly with i: observe in the example above that $\rho^1(S) = 1,\ \rho^2(S) = 2, \dots\ \rho^i(S) = i \dots\ \rho^{n/2}(S) = n/2$. A more robust notion of shiftwise self-distance can thus consider the self-distance of S with respect to all multiples of p as follows.

Definition 3 *The shiftwise self-distance of a given data stream S of length n with respect to p is defined as*

$$\sigma^p(S) = \max_{j=1,\dots n/(2p)} h(S[jp+1:n], S[1:n-jp]).$$

In the subsequent sections we show that the shiftwise self-distance can be used to relax both definitions of exact periodicity up to a constant factor.

2.1.2 Blockwise Self Distance

Shiftwise self-distance is based on Definition [a] of exact periodicity. We now define a self-distance based on the alternative definition, which relates to the "average trend" of a data stream $S \in \Sigma^n$ ([6]) defined in terms of a "representative" block b_j^p of S. More specifically, given block b_j^p of S, we consider the distance of the given streamg from one which consists only of repetitions of b_j^p. Define $\beta_j^p(S) = \sum_{\forall \ell} h(b_\ell^p, b_j^p)$. Based on this notion of average trend, our alternative measure of self-distance for S (also used in [6]) is obtained as follows.

Definition 4 *The blockwise self-distance of a data stream S of length n with respect to p is defined as $\beta^p(S) = \min_i \beta_i^p(S)$.*

Blockwise self-distance is closely related to the shiftwise self-distance as will be shown in the following sections.

2.1.3 Pairwise Self-Distance

We finally present our third definition of self-distance, which, for a given p, is based on comparing all pairs of size p blocks. We call this distance the *pairwise self-distance* and define it as follows.

Definition 5 *Let S consist of k blocks $b_1^p, \ldots, b_k^p$, each of size p. The pairwise self-distance of S with respect to p and discrepancy δ is defined as*

$$\pi^p(S, \delta) = \frac{1}{k^2} |\{(b_i, b_j) \ : \ h(b_i, b_j) > \delta p\}|.$$

That is, $\pi^p(S, \delta)$ is the ratio of dissimilar block pairs (i.e., pairs which differ by more than a given factor δ) to the number of all block pairs. A pairwise self-distance of ϵ reflects an accurate measure of the number of entries that need to be changed to make S exactly p-periodic up to an additive factor of $O((\epsilon + \delta)n)$ and thus is closely related to the other two self-distances.

2.1.4 Approximate Periodicity

Given the three notions of self-distance that we defined in the previous sections, we now describe an approximately periodic stream as one having a small self distance.

Definition 6 *A data stream $S \in \Sigma^n$ is ϵ-approximately p-periodic with respect to σ^p (resp. β^p and π^p) if $\sigma^p(S) \leq \epsilon n$ (resp. $\beta^p(S) \leq \epsilon n$ and $\pi^p(S, \delta) \leq \epsilon n$) for some $p \leq n/2$.*

2.2 Relationships Between Three Notions of Self-Similarity

In this section we explore the relationship between the three notions of self similarity that we have defined in terms of our three distances $\sigma(), \beta()$, and $\pi()$. We do this by relating approximate periodicity with respect to the three distances to each other as well as to the distance to an exactly periodic stream. We first show that, even though the blockwise self-distance $\beta^p(S)$ seems to be quite different from shiftwise self-distance $\sigma^p(S)$, the two measures are within a factor two of each other.

Theorem 1 *Given $S \in \Sigma^n$ and $p \leq n/2$, $\beta^p(S)/2 \leq \sigma^p(S) \leq 2\beta^p(S)$.*

Proof. We first show the upper bound.
Let $b_i = b_i^p$ be the representative trend of S (of size p), that is, $i = argmin_{1 \leq j \leq k} \sum_{\ell=1}^{k} h(b_\ell, b_j)$. By definition, $\sigma^p(S) = \max_{1 \leq j \leq k} \rho^{jp}(S) = \max_j \sum_{\ell=1}^{k-j} h(b_\ell, b_{\ell+j})$.

By the triangular inequality, $\sigma^p(S) \leq \max_j [\sum_{\ell=1}^{k-j} h(b_\ell, b_i) + \sum_{\ell=1}^{k-j} h(b_i, b_{\ell+j})] \leq \max_j [\sum_{\ell=1}^{k} h(b_\ell, b_i) + \sum_{\ell=1}^{k} h(b_i, b_\ell)]$. Since h is symmetric, this is at most $2\sum_{\ell=1}^{k} h(b_i, b_\ell)$, which is exactly $2 \cdot \beta^p(S)$.

For the lower bound, note that $\beta^p(S) \leq \frac{1}{k}\sum_{j=1}^{k}\sum_{\ell=1}^{k} h(b_\ell, b_j)$.
But $\sigma^p(S) \geq \frac{1}{k}\sum_{j=1}^{k}\sum_{\ell=1}^{k-j} h(b_j, b_{j+\ell}) \geq \frac{1}{2k}\sum_{j=1}^{k}\sum_{\ell=1}^{k} h(b_\ell, b_j) \geq \beta^p(S)/2$. ∎

Our notions of approximate periodicity seem to be different from each other, however, they relate to whether changing a small number of characters of S would make S periodic. The following lemmas explore this relation.

Lemma 2 *If a data stream S is ϵ-approximately p-periodic with respect to β^p then S can be made exactly p-peridoic by changing γn of its entries for $\epsilon/2 \leq \gamma \leq \epsilon$.*

Proof. Let $B = argmin_{b \in \Sigma^p} \sum_{\ell=1}^{k} h(b_\ell, b)$. Clearly S is almost p-periodic if $\sum_{\ell=1}^{k} h(b_\ell, B) \leq \gamma n$. Similarly let $b_i = b_i^p$ be the representative trend of S; i.e. $i = argmin_{1 \leq j \leq k} \sum_{\ell=1}^{k} h(b_\ell, b_j)$. However:

$$k\epsilon = k\sum_{\ell=1}^{k} h(b_\ell, b_i) \leq \sum_{\forall \ell}\sum_{\forall j} h(b_\ell, b_j) \leq \sum_{\forall \ell}\sum_{\forall j} h(b_\ell, B) + h(B, b_j) \leq 2k\sum_{\ell=1}^{k} h(b_\ell, B) = 2k\gamma.$$

The second part of the inequality is trivial. ∎

Lemma 3 *If a given data stream S is ϵ-approximately p-periodic with respect to π^p then S can be made exactly periodic by changing $O((\delta + \epsilon)n)$ of its entries.*

Non-Hamming Measures. We showed above how to test whether a data stream S of size n is ϵ-approximately p-periodic using self-distances $\sigma^p()$ and $\beta^p()$. We assumed that the comparison of blocks was done in terms of the Hamming distance. We now show how to use other distances of interest.

First, consider the L_1 distance. Note that, since our alphabet Σ is of constant size, the L_1 distance between two data streams is within a constant factor of their Hamming distance. More specifically, let $q = |\Sigma|$. Then, for any $R, S \in \Sigma^n$, $q \cdot h(R, S) \geq L_1(R, S)$. Thus, the method of estimating the Hamming distance will satisfy the requirements of our test for L_1 albeit with different constant factors. Let σ' and β' be the self-distance measures which modify the Hamming distance based measures of σ and β by the use of L_1 distance. Then, for any given p our estimate $\sigma'^p(S)$ will still be within at most a constant factor of $\beta'^p(S)$.

Now consider the L_2 distance. Again, assuming that our alphabet Σ is of size q one can observe that, if $h(R, S) = p$, then $\sqrt{p} \leq L_2(R, S) \leq q\sqrt{p}$. Thus, by making the necessary adjustments to the allowed distance, one can obtain a test with different constant factors as with the L_1 distance. In fact, a similar argument holds for any L_i distance.

Similar discussions apply for π^p as well and are hence omitted.

3 Sublinear Algorithms for Measuring Self-Distances and Approximate Periodicity

In this section, we present sublinear algorithms for testing whether a given data stream S is approximately periodic under each of the three self-distance measures. By the results in the previous section, these algorithms can be used to estimate the minimum number of entries that must be changed to make S exactly periodic.

3.1 Checking Approximate Periodicity Under σ^p

We now show how to check whether a given stream $S \in \Sigma^n$ is ϵ-approximately p-periodic for a fixed $p \leq n/2$ under σ^p. We generalize this to the case where p is not known in advance (such as finding the smallest p for which S is ϵ-approximately p-periodic) following the discussion on the other similarity measures.

Note that the definition of approximate periodicity with respect to σ^p leads to the following property analogous to that of exact periodicity.

Property 1 *If S is ϵ-approximately p-periodic under σ^p then it is ϵ-approximately ip-periodic under σ^p for all $i \leq n/2p$.*

We remind the reader that as typical of probabilistic tests, our method distinguishes self-distances of over ϵn from those below $\epsilon' n$. In our case, $\epsilon' = c\epsilon$ for some small constant $0 < c < 1$ which results from using probabilistic bounds. [3] The output of our algorithm is not constrained when the self-distance is between ϵn and $\epsilon' n$; that is, both answers of the form "S is p-approximately periodic" and "S is not p-approximately periodic" will be considered equally acceptable in that case.

We first observe that to estimate $\rho^p(S)$ within a constant factor, it suffices to use a constant number of samples from S. More precisely, Given $S \in \Sigma^n$ and $p \leq n/2$, one can determine whether $\rho^p(S) \leq \epsilon n$ or $\rho^p(S) \geq \epsilon' n$ with constant probability using $O(1)$ random samples from S – all one needs to do is to estimate whether $h(S[p+1:n], S[1:n-p])$ is below $\epsilon' n$ or above ϵn. A simple application of Chernoff bounds shows us that comparing a constant number of sample pairs of the form $(S[i], S[i+p])$ is sufficient to obtain a correct answer with constant probability.

Recall that to test whether S is ϵ-approximately p-periodic with respect to σ^p, we need to estimate each $\rho^{ip}(S)$ for $ip \leq n/2$. When p is small, there are a linear number of such distances that we need to compute. If we choose to compute each $\rho^{ip}(S)$ separately, with different random samples, together with the addition of a logarithmic factor for guaranteeing correctness for each period tested this translates into a superlinear number of samples. In that case one might more easily choose to read the entire stream. Instead, we choose to economize on the number of samples from S by

[3] Depending on ϵ, one has an amount of freedom in choosing c; for instance, $c = 1/2$ can be achieved using Chernoff bounds and the confidence can be boosted by increasing the number of samples logarithmically in the confidence parameter. Thus for the rest of this paper we will use ϵ and ϵ' without mentioning their exact relationship with this implicit understanding.

"recycling" a sublinear pool of samples. This is viable as our analysis does not require the samples to be determined independently.

With the ultimate goal (among others) of finding the smallest p for which S is ϵ-approximately p-periodic, we now explore how many samples are needed to estimate $\rho^p(S)$ in the above sense for *all* $p = 1, 2, \cdots n/2$.

Above we observed that in order to estimate $\rho^p(S)$ for a specific p we need to compare $O(1)$ sample pairs of the form $(S[i], S[i+p])$. We now would like to determine the number of samples required to guarantee that a sufficient number of sample pairs $(S[i], S[i+p])$ will be available for each $n/2 \geq p \geq 1$. The following lemma states that a pool of $O(\sqrt{n} \cdot \text{polylog } n)$ samples suffices.

Lemma 4 *A uniformly random sample pool of size $O(\sqrt{n} \cdot \textit{polylog } n)$ from S has $\Omega(1)$ sample pairs of the form $(S[i], S[i+p])$ for every $1 \leq p \leq n/2$ with constant probability.*

Proof. For any given p, one can use the birthday paradox to show that the presence of $O(\sqrt{n})$ uniform samples from S will guarantee, with constant probability (say, at least $1-\rho$), the availability of $\Omega(1)$ sample pairs of the form $(S[i], S[i+p])$.

For all possible values of p, the probability that at least one of them will *not* have enough samples is at most $1 - (1-\rho)^{n/2}$. Repeating the sampling $O(\text{polylog } n)$ times, this failure probability can be reduced to any desired $1/\text{poly } (n)$. ∎

The lemma above demonstrates that by using $O(\sqrt{n} \cdot \text{polylog } n)$ independent random samples from S one can test whether S is ϵ-approximately p-periodic for any p. The theorem below then follows immediately from the definition of approximate periodicity.

Theorem 5 *It is possible to test whether a given $S \in \Sigma^n$ is ϵ-approximately p-periodic or is not ϵ'-approximately p-periodic under σ^p by using $O(\sqrt{n} \cdot \textit{polylog } n)$ samples and space with high probability.*

3.2 Checking Approximate Periodicity Under β^p

As Theorem 1 implies, the two notions of self-distance (under Hamming measure) are equivalent up to a factor of 2. We have shown how to test whether the shiftwise self-distance of S, $\sigma^p(S)$ is no more than some ϵn for any given p by using only a sublinear $(O(\sqrt{n} \cdot \text{polylog } n))$ number of samples from S and similar space. The above lemma implies that this is also doable for $\beta^p(S)$; i.e. one can test whether the blockwise self-distance of S is no more than some ϵn for any given p by using $O(\sqrt{n} \cdot \text{polylog } n)$ samples from S and similar space.

The method presented in [6] can also perform this test by first constructing from S a superlinear $(O(kn \log n))$ size pool of "sketches"; here k is the size of an individual sketch which depends on their confidence bound. Since this pool can be too large to fit in main memory, a scheme is developed to retrieve the pool from secondary memory in smaller chunks. In contrast, our overall memory requirement (and sample size) is sublinear; this comes at a price of some small loss of accuracy.

Due to the fact that $\sigma^p()$ and $\beta^p()$ are within a factor 2 of each other, they can be estimated in the same manner. Thus, the theorem below follows from its counterpart for σ^p, (Theorem 3), which states that approximate p-periodicity can be efficiently checked.

Theorem 6 *It is possible to test whether a given $S \in \Sigma^n$ is ϵ-approximately p-periodic or is not ϵ'-approximately p-periodic under β^p by using $O(\sqrt{n} \cdot \text{polylog } n)$ samples and space with high probability.*

Here the "gap" between ϵ and ϵ' is within factor 4 of the gap for $\sigma^p()$.

3.3 Checking Approximate Periodicity Under π^p

Recall that π^p is a measure of the frequency of dissimilar blocks of size p in S. In this section, we show how to efficiently test whether S is ϵ-approximately p-periodic under π^p for any p where p is not known a priori. In order to be able to estimate $\pi^p(S, \delta)$ for all p, we would like to explicitly compare pairs of blocks of size p. This requires comparing as many as polylogarithmic sample pairs for each pair of blocks (b_i^p, b_j^p). Unfortunately, our pool of samples from the previous section turns out to be too small to yield enough sample pairs of the above kind for all p – in fact, it can be seen easily that a sublinear uniform random sample pool will never achieve the desired sample distribution and the desired confidence bounds in this case. The problem stems from the fact that we need several identically distributed samples within the two blocks every time we compare a pair. To fix this problem, we will force our sampling to obtain identically distributed points from our block choices. Our technique will still use a sublinear size sample pool, but with some dependence between where the locations of the samples are picked.

A Two-Phase Scheme to Obtain The Sample Pool for Estimating π^p. To achieve a sublinear sample pool from S which will have enough per block samples, we obtain our samples in two phases.

In the first phase we obtain a uniform sample pool from S, as in the previous section, of size $O(\sqrt{n} \cdot \text{polylog } n)$; these samples are called *primary samples*.

In the second phase, we obtain, for each primary sample $S[i]$, a polylogarithmic set of *secondary samples* distributed identically around i except when a secondary sample falls outside the boundaries of S, in which case we do not take that sample. To do this, we pick $O(\text{polylog } n)$ offsets relative to a generic location i as follows. We pick $O(\log n)$ neighborhoods of size 1, 2, 4, 8, ... n around i.[4] Neighborhood k refers to the interval $S[i - 2^{k-1} : i + 2^{k-1} - 1]$; e.g., neighborhood 3 (of size 8) of $S[i]$ is $S[i - 4 : i + 3]$. From each neighborhood we pick $O(\text{polylog } n)$ uniform random locations and note their positions relative to i. Note that the choosing of offsets is performed only once for a generic i; the same set of offsets will later be used for all primary samples.

To obtain the secondary samples for any primary sample $S[i]$, we sample the locations indicated by the offset set with respect to location i (as long as the sample location is within S). [5] As a result,

[4]Since we are only choosing offsets, we allow neighborhoods to go past the boundaries of S. We handle invalid locations during the actual sampling. Also, for simplicity, we assume n to be a power of 2.

[5]For bookkeeping, for each secondary sample the size of its relevant neighborhood is noted.

the secondary samples for any two primary samples $S[i]$ and $S[j]$ will be distributed identically around respective locations i and j.

Estimating π^p. We can now use standard techniques to decide whether $\pi^p(S, \delta)$ is large or small. We start by uniformly picking primary sample pairs ($S[i]$, $S[j]$) such that $i - j$ is a multiple of p. [6] Let b_k and b_l of size p denote the blocks containing $S[i]$ and $S[j]$ respectively. We can now proceed to check whether $h(b_k, b_l)$ is large by comparing these two blocks at random locations. To obtain the necessary samples for this comparison, we use our sample pool and the neighborhoods used in creating it as follows. We consider the smallest neighborhood around $S[i]$ which contains b_k and use the secondary samples of $S[i]$ from this neighborhood that fall within b_k. We then pick samples from b_l in a similar way and compare the samples from b_k and b_l to check $h(b_k, b_l)$. We repeat the entire procedure for the next block pair until sufficient block pairs have been tested.

To show that this scheme works, we first show that we have sufficient primary samples for any given p to compare enough pairs of blocks. To do this, for any p, we need to pick $O(\text{polylog } n)$ pairs of size p blocks uniformly, which is possible given our sample set as the following simple lemma demonstrates.

Lemma 7 *Consider all sample pairs $(S[i], S[j])$ from a set of $O(\sqrt{n} \cdot \text{polylog } n)$ primary samples uniformly picked from a data stream S of length n. Given any $0 < p \leq n/2$, the following hold with high probability:*

(a) There are $\Omega(\text{polylog } n)$ pairs $(S[i], S[j])$ such that $i - j$ is a multiple of p.

(b) Consider block pair (b_i, b_j) containing a sample pair $(S[i], S[j])$ as described in (a). (b_i, b_j) is uniformly distributed in the space of all block pairs of of S of size p. [7]

Proof. (a) follows easily from Lemma 4.

To see (b), consider two block pairs (b_i, b_j) and (b_k, b_l). There are p sample pairs which will induce the picking of the former pair, and the same holds for the latter pair. Thus, any block pair will be picked with equal probability. ∎

Thus, our technique allows us to have, for any p, a polylogarithmic size uniform sample of block pairs of size p. Now, consider the secondary samples within a block that we pick for comparing two blcoks as explained before. It is easy to see that these particular samples are uniformly distributed within their respective blocks, since secondary samples within any one neighborhood are uniformly distributed. Additionally, they are located at identical locations within their blocks. All we need is there to be a sufficient number of such samples, which we argue below.

Lemma 8 *Let $S[i]$ and $S[i + rp]$ be two primary samples. Let b_l and b_m be the blocks of size p that contain $S[i]$ and $S[i + rp]$ respectively. Then, with the sampling scheme described above we will have picked sufficient secondary samples to tell whether $h(b_l, b_m) \geq \delta p$ high probability.*

[6] There are several simple ways of doing this without violating our space bounds which involve time/space tradeoffs that are not immediately relevant to this paper. Additionally, picking the pairs without replacement makes the final analysis more obvious but makes the selection process slightly more complicated.

[7] For simplicity we assume that p divides n; otherwise one needs to be a little careful during the sampling to take care of the boundaries.

Proof. Consider t such that $2^{t-1} < p \leq 2^t$. The $(t+1)$-neighborhood of $S[i]$ is of size at most $4p$, and contains b_l. Since b_l occupies at least $1/4$ of this neighborhood, it is expected to contain at least a quarter of the secondary samples of $S[i]$ from this neighborhood, which will be uniformly distributed in b_l. The case is the same for b_m and the samples it contains. As a result, we have $\Omega(\text{polylog } n)$ uniform random samples from both b_l and b_m, which, as we argued before, can be viewed as pairs of points located identically within their respective blocks. Then, one can test whether $h(b_l, b_m) \geq \delta p$ with high probability by comparing the corresponding sample pairs from each block. ∎

Combining the choice of blocks and the comparison of block pairs, we obtain the following theorem.

Theorem 9 *It is possible to test whether a given $S \in \Sigma^n$ is ϵ-approximately p-periodic or is not ϵ'-approximately p-periodic under π^p by using $O(\sqrt{n} \cdot \text{polylog } n)$ samples and space with high probability.*

Since our algorithm does not require advance knowledge of p, to find all periods, the smallest period, etc. under this measure, it suffices to try the test with different values of p without increasing the sample size, as we argue in the next section.

3.4 Checking Periodicity for All Periods

In general we do not have access to a hypothetical period and may want to know whether a data stream S is ϵ-approximately periodic with any period, and/or what its smallest period p is. These can easily be determined once the particular similarity measure is evaluated for all possible p. Since σ^p and β^p involve computing similarities for all p, for these two measures it is easy to extend the computation to all p. As for π^p, checking for approximate periodicity for a fixed p is easy, but the trivial technique of picking blocks and sampling will not extend to efficiently checking for all p. However, our technique as described in the previous section is specially designed so that its sample set will work with high probability for any and every valid p. Thus, checking periodicity for varying periods is now possible by using sublinear samples.

Theorem 10 *Given $S \in \Sigma^n$, it is possible to perform any of the following tasks under σ^p, β^p, and π^p by using $O(\sqrt{n} \cdot \text{polylog } n)$ independent random samples from S and similar space:*
a) to find out if S is ϵ-approximately p-periodic,
b) to find all periods p (and thus the smallest period) for which S is ϵ-approximately p-periodic.

Note that if the smallest approximate period of S is determined to be p then we guarantee that $\sigma^p(S) \leq \epsilon' n$ and there exists no $j < p$ such that $\sigma^j(S) < \epsilon n$. The same holds for β^p and π^p as well.

4 Concluding Remarks

We introduced new notions of time series data streams being approximately periodic based on significance of combinatorial scores in terms of self-distances. We presented the first known sublinear– $O(\sqrt{n}$ polylog $n)$ space– algorithms for detecting such approximate periodicities in time series data streams based on sampling, and reusing these random samples for multiple potential period lengths. Besides such periodicities, there may be other representative trends in a data stream; it could be interesting to develop efficient, sublinear sampling algorithms for detecting such trends.

5 Acknowledgements

The authors would like to thank the anonymous referees for significantly improving the presentation of this paper.

References

[1] T. Batu, F. Ergun, J. Kilian, A. Magen, S. Raskhodnikova, R. Rubinfeld and R. Sami. *A sublinear algorithm for weakly approximating edit distance. STOC 2003, 316–324.*

[2] A. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan and M. Strauss. *Near-optimal sparse fourier representations via sampling. Proc. STOC 2002. 152–161.*

[3] O Goldreich, S. Goldwasser and D. Ron. *Property testing and its connection to learning and approximation*, *Journal of the ACM 45(4):653–750, 1998.*

[4] R. Rubinfeld. Talk on sublinear algorithms. http://external.nj.nec.com/homepages/ronitt/

[5] R. Rubinfeld and M. Sudan, *Robust Characterization of Polynomials with Applications to Program Testing*, *SIAM Journal of Computing 25(2):252–271, 1996.*

[6] P. Indyk and N. Koudas and S. Muthukrishnan *Identifying Representative Trends in Massive Time Series Data Sets Using Sketches. Proc. VLDB 2000. 363–372.*

[7] G. Das and D. Gunopoulos. *Time Series Similarity Measures. http://www.acm.org/sigs/sigkdd/kdd2000/Tutorial-Das.htm*

[8] G. Kollios. *Timeseries Indexing. http://www.cs.bu.edu/faculty/gkollios/ada01/LectNotes/tsindexing.ppt*

[9] F. Olken and D. Rotem. *Random sampling from databases: A Survey. Bibliography at http://pueblo.lbl.gov/ olken/mendel/sampling/bibliography.html*

[10] S. Chaudhuri, G. Das, M. Datar, R. Motwani and V. Narasayya. *Overcoming Limitations of Sampling for Aggregation Queries. Proc. ICDE 2001.*