Periodicity and Cyclic Shifts via Linear Sketches

Michael S. Crouch[⋆] and Andrew McGregor[⋆⋆]

Department of Computer Science, University of Massachusetts, Amherst, MA 01003

Abstract. We consider the problem of identifying periodic trends in data streams. We say a signal $\mathbf{a} \in \mathbb{R}^n$ is p-periodic if $a_i = a_{i+p}$ for all $i \in [n-p]$. Recently, Ergün et al. [4] presented a one-pass, $O(\operatorname{polylog} n)$-space algorithm for identifying the smallest period of a signal. Their algorithm required $\mathbf{a}$ to be presented in the *time-series* model, i.e., a_i is the ith element in the stream. We present a more general linear sketch algorithm that has the advantages of being applicable to a) the *turnstile stream model*, where coordinates can be incremented/decremented in an arbitrary fashion and b) the *parallel* or *distributed* setting where the signal is distributed over multiple locations/machines. We also present sketches for $(1+\epsilon)$ approximating the ℓ_2 distance between $\mathbf{a}$ and the nearest p-periodic signal for a given p. Our algorithm uses $O(\epsilon^{-2} \operatorname{polylog} n)$ space, comparing favorably to an earlier time-series result that used $O(\epsilon^{-5.5}\sqrt{p} \operatorname{polylog} n)$ space for estimating the Hamming distance to the nearest p-periodic signal. Our last periodicity result is an algorithm for estimating the periodicity of a sequence in the presence of noise. We conclude with a small-space algorithm for identifying when two signals are exact (or nearly) cyclic shifts of one another. Our algorithms are based on bilinear sketches [10] and combining Fourier transforms with stream processing techniques such as ℓ_p sampling and sketching [11, 13].

1 Introduction

We consider the problem of identifying periodic trends in data streams. Motivated by applications in computational biology and data mining, there has recently been a series of papers related to finding such trends in large data sets [3–5, 9, 16]. We say a signal $\mathbf{a} \in \mathbb{R}^n$ is *p-periodic* if it can be expressed as a concatenation $\mathbf{a} = \mathbf{x} \circ \ldots \circ \mathbf{x} \circ \mathbf{x}'$ for some $\mathbf{x} \in \mathbb{R}^p$ and some $\mathbf{x}' \in \mathbb{R}^{n-p\lfloor n/p \rfloor}$ that is a prefix of $\mathbf{x}$. We say $\mathbf{a}$ is *perfectly p-periodic* if $\mathbf{a}$ is p-periodic and $p \mid n$. Given a signal $\mathbf{a} \in \mathbb{R}^n$, we define the distance to p-periodicity as

$$\mathrm{D}_p(\mathbf{a}) \equiv \min_{\mathbf{y} \in P_{p,n}} \|\mathbf{a} - \mathbf{y}\|_2 \quad \text{where } P_{p,n} = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{y} \text{ is } p\text{-periodic}\}$$

where $\|\mathbf{v}\|_2 = \sqrt{v_1^2 + \ldots + v_n^2}$ denotes the ℓ_2 norm of the vector $\mathbf{v} \in \mathbb{R}^n$. (We will later discuss our choice of distance measure and observe that many of our results still hold if an alternative measure is chosen.) We denote the minimum period of a signal $\mathbf{a} \in \mathbb{R}^n$ by

$$\operatorname{period}(\mathbf{a}) = \min\{p : \mathbf{a} \text{ is } p\text{-periodic}\} \ .$$

[⋆] Email: `mcc@cs.umass.edu`.

[⋆⋆] Email: `mcgregor@cs.umass.edu`. Research supported by NSF Award CCF-0953754.

In this paper, we consider signals defined by a stream of data. Previous periodicity work assumes that the stream *is* the signal, e.g., the stream $\langle 1,2,3,4\rangle$ defines the signal $\mathbf{a} = [1,2,3,4]$. However, we wish to consider a more general setting. For example, consider a sensor network in which each node is tasked with recording the times when certain local events occur. These records are forwarded through the network to some central node for processing. In this situation, there is no guarantee that the records are received in the order they were generated. Hence, we would need an algorithm that could identify patterns even if the records arive out of order. A yet more challenging example would be if each sensor monitors the local temperature at each time step and we are interested in identifying periodic trends in the average temperature. In this case, not only can records arrive out of order but the signal will be determined by the value of multiple records.

Following the terminology of Muthukrishnan [14, pg. 12–13], we consider three different stream models in which the signal $\mathbf{a}$ of interest can be defined. In the *time-series* model the stream $S = \langle a_0 \ldots a_{n-1}\rangle$ defines the signal directly. More general is the *permutation* model where coordinates of $\mathbf{a}$ may arrive out of order, i.e., $S = \langle (\pi(0), a_{\pi(0)}) \ldots (\pi(n-1), a_{\pi(n-1)})\rangle$, for some permutation π of $\{0,\ldots,n-1\}$. Finally, in the *turnstile* model, $\mathbf{a}$ is defined by a sequence of increments and decrements, i.e., for a stream

$$S = \langle (u_1, \varDelta_1), \ldots, (u_m, \varDelta_m)\rangle \quad \text{where } u_i \in \{0,\ldots,n-1\},\ \varDelta_i \in \mathbb{R}$$

we define $\mathbf{a}$ by $a_j = \sum_{i:u_i=j} \varDelta_i$. All of our algorithms work in the turnstile model and are *sketch-based*. We will discuss sketches in more detail in Sect. 2 but note here that one of their main advantages is that they work in a distributed setting where parts of the streams are monitored at different locations: after the stream concludes, it is sufficient to communicate only the sketches, as these can then be merged in order to estimate the global property of interest. This would enable data aggregation in the sensor network example outlined above.

1.1 Our Results and Related Work

Our first result is an $O(\epsilon^{-2} \operatorname{polylog} n)$ space algorithm that $(1+\epsilon)$-approximates $\mathrm{D}_p(\mathbf{a})$ for any given p (where p need not divide the length of the sequence). In contrast, an earlier paper by Ergün et al. [4] presented and an algorithm using $O(\epsilon^{-5.5}\sqrt{p} \operatorname{polylog} n)$ space for estimating the Hamming distance to the nearest p-periodic signal. They also present a single-pass, $O(\operatorname{polylog} n)$-space algorithm for computing period($\mathbf{a}$) in the time-series model. Our second result generalizes this result to the turnstile model although our algorithm in this case requires that $\mathbf{a}$ is perfectly periodic.

Next we examine estimating the periodicity of a sequence in the presence of noise. While a seemingly natural problem, defining the precise problem is subtle. For example, should we deem the noisy signal

$$\mathbf{a} = [1,2,3,1,2,3.5,1,2,3.1,1,2,3.4] \tag{1}$$

to be 3-periodic, 6-periodic, or aperiodic? Our algorithm achieves a natural "gap promise" guarantee: given φ, ϵ with $0 < \varphi < \epsilon < 1$, it returns a period $p \mid n$ with

$$\mathrm{D}_p(\mathbf{a}) \leq \epsilon \|\mathbf{a}\|_2 \qquad \text{and} \qquad p \leq \min\{q \mid n : \mathrm{D}_q(\mathbf{a}) \leq (\epsilon - \varphi)\|a\|_2\} \ .$$

(Note that there is always such a p, since any length-n signal trivially has $\mathrm{D}_n(\mathbf{a}) = 0$.) In other words, we ensure that $\mathbf{a}$ is close to being perfectly p-periodic and that there is no $q \leq p$ such that $\mathbf{a}$ is "significantly closer" to being perfectly q-periodic. This algorithm operates in the general turnstile model and uses $\mathrm{poly}(\log n, \varphi^{-1})$ space. The algorithm is based on sampling in the Fourier domain and was actually inspired by Shor's algorithm for quantum factorization [17]. There is no analog in the recent Ergün et al. [4] paper but an earlier result [5] in the combinatorial property-testing model can be applied in the streaming setting if we may use $O(\sqrt{n}\,\mathrm{polylog}\, n)$ space.

We conclude with a simple sketch algorithm for the related problem of identifying when two sequences are cyclic shifts of one another. This algorithm uses $O(\epsilon^{-2}\sqrt{n}\,\mathrm{polylog}\, n)$ space and has the additional feature that it actually approximates how close the strings are to being cyclic shifts.

Notation. We write $[n] = \{0, 1, 2, \ldots, n-1\}$. We denote signals in lower-case bold and their corresponding Fourier transforms in upper-case bold. For a complex number $z \in \mathbb{C}$ we denote the real and imaginary parts by $\mathrm{Re}(z)$ and $\mathrm{Im}(z)$ respectively. For functions $f(n), g(n)$, we write $f(n) = \tilde{O}(g(n))$ when there is a constant k such that $f(n) = O(g(n) \log^k n)$. $I[\varphi]$ is the 0-1 indicator function which is 1 whenever φ is true.

Precision. Throughout, we will assume that the values of the signals can be exactly stored with $1/\mathrm{poly}(n)$ precision. For example, this would be guaranteed in the turnstile model with a number of updates $m = \mathrm{poly}(n)$ and with each $\Delta_j \in \{-M, -M+1, \ldots, M-1, M\}$ for some $M = \mathrm{poly}(n)$. We also assume that the approximation parameters $\epsilon, \varphi, \delta$ satisfy $1/\epsilon, 1/\delta, 1/\varphi \in O(\mathrm{poly}\, n)$.

2 Fourier Preliminaries and Choice of Distance Function

In this section, we review the basic definition and properties of the discrete Fourier transform. We then discuss the utility of the transform in the context of sketch-based data stream algorithms.

2.1 Discrete Fourier Transform and Sketches

Given a signal $\mathbf{a} \in \mathbb{R}^n$, the discrete Fourier transform of $\mathbf{a}$, denoted $\mathbf{A} = \mathcal{F}(\mathbf{a})$, is defined as

$$\mathbf{A} = (A_0, A_1, \ldots, A_{n-1}) \qquad \text{where} \qquad A_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} a_j e^{\frac{2\pi i}{n} jk} \ .$$

The following proposition states some standard properties that will be of use.

Proposition 1. *For any signal* $\mathbf{a} \in \mathbb{R}^n$*,*

1. $\mathbf{a}$ *is perfectly* p*-periodic iff* $(A_k \neq 0 \Rightarrow n/p \mid k)$*.*
2. $\|\mathbf{a}\|_2 = \|\mathbf{A}\|_2$ *(Parseval's identity).*

Of particular importance in the context of data streams is the fact that the transformation from $\mathbf{a}$ to $\mathbf{A}$ is a *linear* transformation, i.e.,

$$\mathbf{A}^T = V\mathbf{a}^T \text{ where } V \in \mathbb{C}^{n \times n} \text{ and } V_{kj} = \frac{1}{\sqrt{n}} e^{\frac{2\pi i}{n} kj} \text{ for } k, j \in [n] \ . \qquad (2)$$

This is significant because many data stream algorithms are based on randomized linear projections called *sketches*. Suppose we are interested in a function f of $\mathbf{x} \in \mathbb{R}^n$ where each coordinate x_j is determined by the turnstile stream $S = \langle (u_1, \Delta_1), \ldots, (u_m, \Delta_m) \rangle$ according to $x_j = \sum_{i: u_i = j} \Delta_i$. A sketching algorithm chooses a random linear map $W \in \mathbb{R}^{k \times n}$ such that $W\mathbf{x}^T$ can be post-processed to yield an estimate of $f(\mathbf{x})$ (with certain error and probability guarantees). The algorithm computes $W\mathbf{x}^T$ incrementally using space proportional to k rather than n:

$$W\mathbf{x}^T = (\ldots(((W\mathbf{e}^{u_1}) + W\mathbf{e}^{u_2}) + W\mathbf{e}^{u_3}) + \ldots) + W\mathbf{e}^{u_m}$$

where $\mathbf{e}^{u_i} = (0, \ldots, 0, \Delta_i, 0, \ldots, 0)^T$ has the non-zero entry in the u_i-th position. For many functions, such as quantiles and heavy hitters [2], distinct items [12], and ℓ_1 and ℓ_2 norms [8], such sketches exist where k is only polylogarithmic in n. Of course, it would still defeat the object of small-space computation if the algorithm needed to explicitly store a random $k \times n$ matrix. Instead the random matrices of interest are constructed either using limited independence or via a pseudo-random generator, e.g., Nisan [15]. Either way, the relevant entries can be reconstructed from some small seed as required.

We will make use of the simple, but very useful, observation that rather than estimating functions in the time domain, we may estimate these functions in the frequency domain by combining the change of basis matrix V with the sketch matrix W. For example, if the random sketch matrix $W \in \mathbb{R}^{k \times n}$ can be used to estimate the number of non-zero entries in $\mathbf{a}$ then the sketch matrix $WV \in \mathbb{C}^{k \times n}$ can be used to estimate the number of non-zero entries[1] in $\mathbf{A}$.

2.2 Choice of Distance Function

In the context of the Fourier transform and many signal processing applications, the natural measure of dissimilarity between two signals is the ℓ_2 norm of their

[1] To be precise, it is often necessary to separate real and imaginary parts of V. That is, we consider $W \in \mathbb{R}^{k \times 2n}$ and let $V \in \mathbb{R}^{2n \times n}$ have entries $V_{kj} = \cos(2\pi jk/n)$ for $k \in \{0, \ldots, n-1\}$ and $V_{kj} = \sin(2\pi jk/n)$ for $k \in \{n, \ldots, 2n-1\}$. In calculating the ℓ_2 norm this causes no difficulties, but in other cases we may need to be careful. If we counted the number of nonzero entries of V, for example, we would find the total number of non-zero real parts and non-zero imaginary parts.

difference. In contrast, Ergün and coauthors [4,5] considered a measure based on the Hamming distance, $\mathrm{D}_p^0(\mathbf{a}) \equiv \min_{\mathbf{y}\in P_{p,n}} \Delta(\mathbf{a},\mathbf{y})$ where $\Delta(\mathbf{a},\mathbf{y}) = |\{i \in [n] : a_i \neq y_i\}|$. While different measures are suited to different applications, many of our algorithms can also be applied to approximate the Hamming distance, at least in the permutation model.

Suppose $\Sigma = \{\sigma_1, \ldots, \sigma_r\}$ and consider the mapping from $\Sigma \to \{0,1\}^r$:

$$h(\sigma) = x_1 \ldots x_r \qquad \text{where } x_j = \begin{cases} 1 & \text{if } \sigma = \sigma_j \\ 0 & \text{otherwise} \end{cases} .$$

The following lemma demonstrates that $\mathrm{D}_p^0(\mathbf{a})$ and $(\mathrm{D}_p(h(\mathbf{a})))^2/2$ are closely related. Hence, if each element of the sequence is first transformed using h (as is possible in the permutation model) then the Hamming distance to periodicity can be approximated via the ℓ_2 distance to periodicity. The approximation is by a factor close to 1 if the sequence is close to being p-periodic. Note that we would expect this to be the more relevant case in the sense that we would be measuring the distance from periodicity of a nearly-periodic sequence.

Lemma 1. *For any* $\mathbf{a} \in \Sigma^n$*, with* $\Sigma = \{\sigma_1, \ldots, \sigma_r\}$*, let* $T(\mathbf{a}) = (\mathrm{D}_p(h(\mathbf{a})))^2/2$*. Then we have,*

$$\tfrac{1}{2}\,\mathrm{D}_p^0(\mathbf{a}) \leq T(\mathbf{a}) \leq \mathrm{D}_p^0(\mathbf{a}) \ . \tag{3}$$

Furthermore, if $\mathbf{a}$ *is almost periodic in the sense that at least a* $1-\epsilon$ *fraction of the elements* $\{a_j, a_{j+p}, \ldots, a_{j+n-p}\}$ *are identical for each* $j \in [p]$*, then* $(1-\epsilon)\,\mathrm{D}_p^0(\mathbf{a}) \leq T(\mathbf{a}) \leq \mathrm{D}_p^0(\mathbf{a})$*.*

We can also relate $\mathrm{D}_p(\mathbf{a})$ to the ℓ_1 distance to the nearest p-periodic signal. For this, consider the alphabet $\Sigma = \{1, \ldots, t\}$, and use the mapping $h(\sigma) = x_1 \ldots x_r$ where $x_j = I[\sigma \geq j]$.

3 Periodicity

3.1 Distance from Fixed Periodicity

We first present a fast algorithm for measuring the distance between the signal and the closest (under the ℓ_2 norm) p-periodic sequence, for fixed p. In this section, we emphasize that we do not require that the length of sequence is a perfect multiple of the periods considered. For $p < n$, we write $n = dp + r$ where

$$d = \lfloor n/p \rfloor \qquad \text{and} \qquad r = n \bmod p \ .$$

Basic properties of the ℓ_2 norm imply that the p-periodic pattern that is ℓ_2-closest to a vector $\mathbf{a}$ is the arithmetic mean of length-p segments of the vector:

Lemma 2. *For any sequence* $\mathbf{a} \in \mathbb{R}^n$*, let* $\mathbf{c} = \operatorname{argmin}_{\mathbf{y}\in P_{n,p}} \|\mathbf{a}-\mathbf{y}\|_2$ *be the* p*-periodic vector which is* ℓ_2*-closest to* $\mathbf{a}$*. Then* $\mathbf{c} = \mathbf{b} \circ \ldots \mathbf{b} \circ (b_0 b_1 \ldots b_{r-1})$ *where*

$$b_i = \begin{cases} \sum_{j=0}^{d} a_{i+jp}/(d+1) & \text{for } 0 \leq i < r \\ \sum_{j=0}^{d-1} a_{i+jp}/d & \text{for } r \leq i \leq p-1 \end{cases} .$$

With this explicit form for $\mathbf{c}$, there is a natural algorithm using Tug-of-War sketches [1] to approximate $\mathrm{D}_p(\mathbf{a}) = \|\mathbf{a}-\mathbf{c}\|_2$. Alon et al. showed that if the entries of a random vector $\mathbf{z} = z_0 \ldots z_{n-1} \in \{-1,1\}^n$ are chosen with 4-wise independence then the random variable $T = \sum_{i=0}^{n-1} z_i(a_i - c_i)$ satisfies $E\left[T^2\right] = \|\mathbf{a}-\mathbf{c}\|_2^2$. They show that the estimator has sufficiently low variance that, by averaging $O(\epsilon^{-2}\log\delta^{-1})$ independent estimators, we can find a $(1+\epsilon)$ approximation for $\|\mathbf{a}-\mathbf{c}\|_2^2$. The value of T can easily be constructed in a streaming fashion: when the ith element of $\mathbf{a}$ is incremented by Δ we increment

$$T \mathrel{+}= \left(z_i - \sum_{j: i = j \bmod p} \frac{z_j}{|\{j : 0 \leq j \leq n-1, i = j \bmod p\}|} \right) \Delta$$

A naive implementation of this update method takes $\Omega(n/p)$ time per update. To avoid this we adapt the bilinear sketch method of Indyk and McGregor [10]. This technique was originally designed to detect correlations in data streams but we can exploit the structure of this sketch to reduce the update time. Rather than view $\mathbf{a}$ as a length n vector, we encode it as a $(d+1)\times p$ matrix A where $A_{ij} = a_{ip+j}$ if $ip + j \leq n-1$ and $A_{ij} = b_j$ otherwise. Similarly let C be the $(d+1)\times p$ matrix where $C_{ij} = b_j$. E.g., for $n = 10$ and $p = 4$ we have the matrices

$$A = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 & a_7 \\ a_8 & a_9 & b_2 & b_3 \end{pmatrix} \qquad \text{and} \qquad B = \begin{pmatrix} b_0 & b_1 & b_2 & b_3 \\ b_0 & b_1 & b_2 & b_3 \\ b_0 & b_1 & b_2 & b_3 \end{pmatrix} .$$

Let $\mathbf{x} \in \{-1,1\}^p$ and $\mathbf{y} \in \{-1,1\}^{d+1}$ be random vectors whose entries are 4-wise independent. Indyk and McGregor extended the Alon et al. result to show that the outer product of $\mathbf{x}$ and $\mathbf{y}$ had sufficient randomness for a result similar to the Tug-of-War sketch. In our context, the result implies that if $T = \sum_{0\leq i\leq d, 0\leq j\leq p-1} x_j y_i (A_{ij} - C_{ij})$, then by appealing to Lemma 2, we have that

$$E\left[T^2\right] = \sum_{0\leq i\leq d, 0\leq j<p} (A_{ij} - C_{ij})^2 = \mathrm{D}_p^2(\mathbf{a})$$

and there is still sufficiently low variance for $O(\epsilon^{-2}\log\delta^{-1})$ parallel repetitions to be sufficient for constructing a $(1+\epsilon)$ approximation with probability $1-\delta$. We next show that each T can be constructed in only $O(1)$ update time. To do this, decompose T as

$$T = \sum_{\substack{0\leq i\leq d \\ 0\leq j<p}} x_j y_i A_{ij} - \sum_{\substack{0\leq i\leq d \\ 0\leq j<p}} x_j y_i C_{ij} = \sum_{\substack{0\leq i\leq d \\ 0\leq j<p}} x_j y_i A_{ij} - \left(\sum_{0\leq i\leq d} y_i\right)\left(\sum_{0\leq j<p} x_j b_j\right)$$

and define $T_1 = \sum_{0\leq i\leq d, 0\leq j<p} x_j y_i A_{ij}$ and $T_2 = \sum_{0\leq j<p} x_j b_j$. Since $\sum_{0\leq i\leq d} y_i$ can be computed in pre-processing, it suffices to compute T_1 and T_2. We initialize

$T_1 = T_2 = 0$. As the stream is read T_1 and T_2 are updated in $O(1)$ time using the following rule: when the $(ip+j)$th entry of $\mathbf{a}$ is incremented by Δ,

$$T_1 \mathrel{+}= \left(x_j y_i + I[j \geq r] \frac{x_j y_d}{d} \right) \Delta \quad \text{and} \quad T_2 \mathrel{+}= \left(I[j < r] \frac{x_j}{d+1} + I[j \geq r] \frac{x_j}{d} \right) \Delta$$

where $r = n \bmod p$ and I is the indicator function.

Theorem 1. $\mathrm{D}_p(\mathbf{a})$ *can be approximated up to a factor* $(1+\epsilon)$ *with probability* $1-\delta$ *using* $\tilde{O}(\epsilon^{-2})$ *space and* $\tilde{O}(\epsilon^{-2})$ *update time. The algorithm operates in the turnstile model using one pass.*

3.2 Determining Perfect Periodicity: Noiseless Case

In this and the next section we consider *finding* the period of a sequence that is perfectly periodic, i.e., we now assume that period divides the length. In this case, a possible approach to detecting periodicity with unknown period would be to use the above algorithm to test all factors $p \mid n$ and return the minimum p such that $\mathrm{D}_p(\mathbf{a}) = 0$ (it suffices to set $\epsilon = 1$ for this purpose). Unfortunately, in the worst case n may have $d(n) = O(\exp(\log n / \log\log n))$ factors [7, pp. 260–264] and therefore this approach would take too much time and space. However, a simple modification suffices: we check for periodicity at each prime or power-of-a-prime factor k of n. Define the set

$$K(n) = \{k : k \text{ divides } n \text{ and is the power of a prime}\} .$$

We first observe that $|K(n)| \leq O(\log n)$ (since each prime factor of n is at least 2, we have from the prime factorization $n = p_1^{r_1} p_2^{r_2} \ldots p_t^{r_t}$ that $|K(n)| = \sum r_i \leq \log_2 n$). The following lemma (see the appendix for the proof) demonstrates that testing periodicity for $p \in K(n)$ is sufficient to determine period$(\mathbf{a})$.

Lemma 3. *For any* $\mathbf{a} \in \mathbb{R}^n$ *which is perfectly periodic,*

$$\mathrm{period}(\mathbf{a}) = \mathrm{GCD}(n/k : k \in K(n) \textit{ and } \mathbf{a} \textit{ is } n/k\textit{-periodic}) .$$

We can thus detect the minimum p for which $\mathbf{a}$ is perfectly p-periodic by running $|K| = O(\log n)$ parallel copies of the algorithm from Section 3.1. With $O(\log n)$ points of failure, we must ensure that each algorithm fails with probability at most $\delta / \log n$; this increases the space by a $\log\log n$ factor which is dominated by other factors in the analysis.

Theorem 2. *There is a single-pass, turnstile algorithm for computing* period$(\mathbf{a})$ *of perfectly periodic strings that uses* $O(\mathrm{polylog}\, n)$ *space and update time.*

3.3 Determining Perfect Periodicity: Noisy Case

In this section, we present an algorithm for estimating the periodicity of a noisy signal. As a stepping stone to this result, we discuss an alternative approach for the noiseless case based on sampling. An advantage of the alternative approach is that it does not require the factorization of n to be computed thereby avoiding any (admittedly sublinear time) preprocessing. However, the guarantee achieved is weaker.

Fourier Sampling. If $\mathbf{a}$ is perfectly periodic with period p, then the Fourier transform $\mathbf{A} = \mathcal{F}(\mathbf{a})$ has at most p nonzero components. Letting $d = n/p$, we know by Prop. 1 that the only non-zero coordinates of $\mathbf{A}$ are A_{kd} for $k \in \{0, \ldots, p-1\}$. For the case of general $\mathbf{a}$, let $\mathbf{X}_p$ denote the restriction of $\mathbf{A}$ to the coordinates corresponding to a perfectly p-periodic signal, i.e.,

$$\mathbf{X}_p = (A_0, 0, \ldots, 0, A_d, 0, \ldots, 0, \ldots, A_{(p-1)d}, 0, \ldots, 0) \ .$$

In the frequency domain, $\mathbf{X}_p$ is the closest Fourier transform of a period-p vector to $\mathbf{A}$. By Plancherel's theorem, $\mathcal{F}$ and $\mathcal{F}^{-1}$ preserve inner products and ℓ_2 distances. Therefore, $\mathcal{F}^{-1}(\mathbf{X}_p)$ is the p-periodic vector that is closest to $\mathbf{a}$ in the ℓ_2 distance. This implies that

$$\mathrm{D}_p(\mathbf{a}) = \left\|\mathbf{a} - \mathcal{F}^{-1}(\mathbf{X}_p)\right\|_2 = \left\|\mathbf{A} - \mathbf{X}_p\right\|_2 = \left\|\mathbf{Y}_p\right\|_2 = \sqrt{\sum_{d \nmid k} |A_k|^2} \ . \tag{4}$$

Our algorithms in this section are based on combining the above relationship with a technique for sampling in the Fourier domain.

Recently, Monemizadeh and Woodruff [13] presented a general approach for *ℓ_p-sampling in the time-domain*: for a signal $\mathbf{a} \in \mathbb{R}^n$ defined in the turnstile model, the goal here is to output k with probability in the interval

$$\left[(1-\alpha)\frac{|a_k|^p}{\ell_p^p(\mathbf{a})}, (1+\alpha)\frac{|a_k|^p}{\ell_p^p(\mathbf{a})}\right]$$

for some small user-defined parameter $\alpha > 0$. They show that this can be performed efficiently in space $\mathrm{poly}(\alpha^{-1} \log n)$.[2]

For our purposes, rather than considering the time-series vector $\mathbf{a}$, we consider the vector

$$\mathbf{A}' = (\mathrm{Re}(A_1), \ldots, \mathrm{Re}(A_n), \mathrm{Im}(A_1), \ldots, \mathrm{Im}(A_n)) \in \mathbb{R}^{2n} \ .$$

defined by applying the appropriate Fourier transform matrix to the signal. If ℓ_2-sampling is performed on $\mathbf{A}'$ and we return the value modulo n, then the probability that k is returned is in the interval:

$$\left[(1-\alpha)\frac{|A_k|^2}{\|\mathbf{A}\|_2^2}, (1+\alpha)\frac{|A_k|^2}{\|\mathbf{A}\|_2^2}\right] \ , \tag{5}$$

because $\frac{\mathrm{Re}(A_k)^2+\mathrm{Im}(A_k)^2}{\sum_{i\in[n]} \mathrm{Re}(A_j)^2+\mathrm{Im}(A_j)^2} = \frac{|A_k|^2}{\|\mathbf{A}\|_2^2}$.

To perform this sampling we follow the approach suggested in Sect. 2. Specifically we use the fact that Monemizadeh and Woodruff's ℓ_p-sampling algorithm can be performed using a sketch matrix W and that there exists a matrix transformation $V \in \mathbb{R}^{2n \times n}$ that transforms any signal $\mathbf{a} \in \mathbb{R}^n$ into the corresponding $\mathbf{A}'$ vector. Hence, applying the sketch matrix WV allows us to sample from $\mathbf{A}'$ as required. We will show how to use this sampling to the next two sections.[3]

[2] There is an additive error probability of n^{-C} for arbitrarily large constant C but this can be ignored in our subsequent analysis.

[3] The reader might also observe that the technique of sketching in the Fourier domain gives an alternative approach to estimating the distance to perfect periodicity using

Application to the Noiseless Case. Suppose there is no noise and that $p = \text{period}(\mathbf{a})$. Let the samples collected be $k_1, \ldots, k_w \in [n]$. We know from Prop. 1 that each sample $k_i = cd$ for some $c \in [p]$. Let $q = n/\operatorname{GCD}(k_1, \ldots, k_w, n)$. We have $q = p/c'$ for some $c' \mid p$. Next we will show that for sufficiently large w, with high probability, either $q = p$ or the sequence was nearly perfectly q-periodic. (For example, in the case of the sequence in Eq. (1), perhaps we return $q = 6$.)

Choose an approximation parameter $\varphi > 0$. Assume for contradiction that $q = p/c'$ for some $c' > 1$, but that $\mathrm{D}_q(\mathbf{a}) \geq \varphi\sqrt{1+\alpha}\|\mathbf{a}\|_2$. Summing over bins j, by appealing to Eq. (4), we have that

$$\sum_{n/q \dagger j} \frac{|A_j|^2}{\|\mathbf{A}\|_2^2} \;=\; \frac{1}{\|\mathbf{a}\|_2^2} \sum_{n/q \dagger j} |A_j|^2 \;=\; \frac{(\mathrm{D}_q(\mathbf{a}))^2}{\|\mathbf{a}\|_2^2} \;\geq\; \varphi^2(1+\alpha)\;.$$

Therefore, using the $(1+\alpha)$ approximation to ℓ_2-sampling, the probability that we return a sample that is not a multiple of n/q is at least φ^2. Taking $w = O(\varphi^{-2}\log(\delta^{-1}\log p))$ samples ensures that we find some sample that is not a multiple of n/q for all $O(\log p)$ prime factors q of p. Consequently, if the algorithm does not return the exact value of $\text{period}(\mathbf{a})$, it returns a value $h \mid \text{period}(\mathbf{a})$ such that the sequence was very close to being h-periodic with high probability.

Application to the Noisy Case. For noisy signals, a natural question is to find the smallest period p such that $\mathrm{D}_p(\mathbf{a}) \leq \epsilon\|\mathbf{a}\|_2$. Unfortunately, since $\mathrm{D}_p(\mathbf{a})$ could be just under $\epsilon\|\mathbf{a}\|_2$ while another value $q < p$ may have $\mathrm{D}_q(\mathbf{a})$ just larger than $\epsilon\|\mathbf{a}\|_2$, this is too much to hope for. Instead we consider two parameters ϵ, φ with $\epsilon > \varphi > 0$, and use a slight modification of the above approaches to accept some $p \mid n$ such that $\mathrm{D}_p(\mathbf{a}) \leq \epsilon\|\mathbf{a}\|_2$, and for no smaller q do we have $\mathrm{D}_q(\mathbf{a}) \leq (\epsilon - \varphi)\|\mathbf{a}\|_2$.

Our algorithm proceeds by taking samples of the Fourier coefficients as before. It then returns the smallest value $p \mid n$ such that at least $1 - (\epsilon - \varphi/2)$ fraction of the samples are of Fourier coefficients $k = cn/p$. With probability at least $1 - \delta$, we can guarantee that this condition is satisfied for all p with $\mathrm{D}_p(\mathbf{a}) \leq (\epsilon - \varphi)\|\mathbf{a}\|_2$, and by no p with $\mathrm{D}_p(\mathbf{a}) > \epsilon\|\mathbf{a}\|_2$; this requires $O(\varphi^{-2}\log\delta^{-1})$ samples by an application of the Chernoff bounds.

any sketch-based algorithm that returns a $(1+\epsilon)$ approximation for ℓ_2, e.g., [1,8,11]. For example, consider the Tug-of-War sketch matrix $W \in \{-1,1\}^{t \times 2n}$ used by Alon et al. [1] for ℓ_2 estimation, and the matrix

$$U \in \mathbb{R}^{2n \times 2n} \text{where } U_{kj} = \begin{cases} 1 & \text{for } j = k \text{ and } d \mid j \\ 0 & \text{otherwise} \end{cases}\;.$$

By appealing to (4), $\|U V\mathbf{a}\|_2^2 = (D_p(\mathbf{a}))^2$. Then, following the analysis of [1] for W, we find $E\left[(WUV\mathbf{a})^2\right] = \mathrm{D}_p(\mathbf{a})^2$ if W is chosen according to the appropriate distribution. Furthermore, the variance is sufficiently low such that a $(1+\epsilon)$ approximation can be constructed with probability $1 - \delta$, it suffices to set $t = O(\epsilon^{-2}\log\delta^{-1})$. This leads to a one-pass algorithm using $O(\epsilon^{-2}\log\delta^{-1}\operatorname{polylog} n)$ space.

Theorem 3. *For any ϵ, φ, δ, there exists a single-pass, $O(\text{poly}(\log n, \varphi^{-1}))$-space turnstile algorithm which returns $p \mid n$ such that both of the following conditions are satisfied with high probability:*

1. $\mathrm{D}_p(\mathbf{a}) < \epsilon \|\mathbf{a}\|_2$
2. *There does not exist $q < p$ such that $q \mid n$ and $\mathrm{D}_q(\mathbf{a}) < (\epsilon - \varphi)\|\mathbf{a}\|_2$.*

4 Cyclic Shifts

In this section, we consider the problem of identifying whether two sequences $\mathbf{a}, \mathbf{b} \in \Sigma^n$ are close to being cyclic shifts of each other. We will assume for convenience that $\Sigma \subset \mathbb{R}$. Let $\mathrm{CS}_s : \mathbb{R}^n \to \mathbb{R}^n$ be the function that "rotates" the input sequence by s positions, i.e.,

$$\mathrm{CS}_s(a_1 a_2 \dots a_n) = a_{s+1} a_{s+2} \dots a_n a_1 \dots a_s \ .$$

Then $\mathbf{a}$ and $\mathbf{b}$ are cyclic shifts iff there exists s such that $\mathbf{b} = \mathrm{CS}_s(\mathbf{a})$.

Our goal is to recognize cyclic shifts using linear sketches. We first note that the analogous problem in the simultaneous communication model is rather straightforward. Supose Alice knows $\mathbf{a} \in \Sigma^n$ and Bob knows $\mathbf{b} \in \Sigma^n$. They can easily determine whether $\mathrm{CS}_s(\mathbf{a}) = \mathbf{b}$ for some s by each transforming $\mathbf{a}$ and $\mathbf{b}$ into some canonical form and then using an equality test. Specifically, consider an arbitrary ordering of the sequences in Σ^n. Alice generates the cyclic shift $\hat{\mathbf{a}}$ of $\mathbf{a}$ that is minimal under this ordering. Similarly, Bob generates the minimal cyclic shift $\hat{\mathbf{b}}$ of $\mathbf{b}$. Clearly $\hat{\mathbf{a}} = \hat{\mathbf{b}}$ iff $\mathbf{a}$ is a cyclic shift of $\mathbf{b}$. This can be verified with $O(\log n)$ communication using standard fingerprinting techniques.

Obviously such an approach is not possible in the data stream model. In the time-series model, existing work combined with simple observations leads to an efficient algorithm for determining if two sequences are cyclic shifts. We first review this before presenting a new streaming algorithm that is sketch-based and thus applies in the turnstile steaming model. Furthermore, it can estimate the distance of two sequences from being cyclic shifts.

Time-Series Model. In the time-series model, a one-pass $O(\text{polylog}\, n)$-space algorithm follows from Ergün et al.'s extensions [4] of the pattern matching algorithm of Porat and Porat [16]. The algorithm works when one of the strings precedes the other, i.e., $S = \langle a_0, a_1, \dots, a_{n-1}, b_0, b_1, \dots, b_{n-1} \rangle$, or when the strings are interleaved, i.e., $S = \langle a_0, b_0, a_1, b_1, \dots, a_{n-1}, b_{n-1} \rangle$. (It is actually sufficient for the elements of one sequence to always precede the corresponding elements of the other; e.g., the stream $S = \langle a_0, b_0, a_1, a_2, b_1, a_3, b_2, b_3 \rangle$ is acceptable.)

The pattern-matching algorithm of [4] uses a fingerprinting function Φ to maintain a series of exponentially-lengthening fingerprints $\varphi_j = \Phi(a_0 \dots a_{2^j - 1})$; by cleverly updating appropriate fingerprints of $\mathbf{b}$, they keep track of each match for φ_j which occurred within the last 2^j characters. When we reach the final character of $\mathbf{b}$, for each m such that $\Phi(b_m \dots b_{m+2^j-1}) = \Phi(a_0 \dots a_{2^j-1})$, we have access to the fingerprints $\Phi(b_0 \dots b_{m-1})$, $\Phi(b_m \dots b_{m+2^j-1})$, and $\Phi(b_{m+2^j} \dots b_{n-1})$.

By adjusting the fingerprints appropriately, we can determine whether there exists $m \in [n]$ such that

$$\Phi(a_0 \dots a_{n-1}) = \Phi(b_m \dots b_{m+2^j-1} b_{m+2^j} \dots b_{n-1} b_0 \dots b_{m-1}) \ .$$

4.1 Cyclic Shift Distance

In this section, we present a simple turnstile algorithm for estimating how close two sequences are to being cyclic shifts. We define the cyclic shift distance, CSD, between two strings as

$$\mathrm{CSD}(\mathbf{a}, \mathbf{b}) = \min_s \|\mathbf{a} - \mathrm{CS}_s(\mathbf{b})\|_2 \ .$$

Clearly, if $\mathbf{b}$ is a cyclic shift of $\mathbf{a}$ then $\mathrm{CSD}(\mathbf{a}, \mathbf{b}) = 0$.

The algorithm proceeds as follows: assume for simplicity that n is a perfect square. We will use two sets of candidate shifts, $S = \{0, 1, 2, \dots, \sqrt{n} - 1\}$ and $T = \{\sqrt{n}, 2\sqrt{n}, 3\sqrt{n}, \dots, n\}$. As we process the turnstile stream, we construct Tug-of-War sketches [1] of $\mathrm{CS}_s(\mathbf{a})$ and $\mathrm{CS}_t(\mathbf{b})$ for each $s \in S$, $t \in T$. Using $O(\epsilon^{-2} \log \frac{1}{\delta} \log n)$-sized sketches, this allows us to $(1+\epsilon)$-approximate $\| \mathrm{CS}_s(\mathbf{a}) - \mathrm{CS}_t(\mathbf{b})\|_2$ for each $s \in S$ and $t \in T$ with probability at least $1 - \delta'$. Since for all r, s we have that $\mathbf{a} - \mathrm{CS}_s(\mathbf{b}) = \mathrm{CS}_r(\mathbf{a}) - \mathrm{CS}_{r+s}(\mathbf{b})$, these shifts suffice to $(1+\epsilon)$-approximate $\|\mathbf{a} - \mathrm{CS}_u(\mathbf{b})\|_2$ for each $u \in \{1, \dots, n\}$.

Choosing $\delta' = \frac{\delta}{n}$, we have that each pair r, s is simultaneously a $(1+\epsilon)$-approximation with probability $\geq 1 - \delta$. We then find:

$$\Pr\left[\left| \min_{s \in S, t \in T} \|\mathrm{CS}_s(\mathbf{a}) - \mathrm{CS}_t(\mathbf{b})\|_2 - \mathrm{CSD}(\mathbf{a}, \mathbf{b})\right| \geq \epsilon \, \mathrm{CSD}(\mathbf{a}, \mathbf{b})\right] \leq \delta \ . \quad (6)$$

Theorem 4. *There exists a single pass algorithm using space $\tilde{O}(\epsilon^{-2}\sqrt{n})$ that returns a $(1+\epsilon)$ approximation for $\mathrm{CSD}(\mathbf{a}, \mathbf{b})$ with probability at least $1 - \delta$.*

5 Conclusion

We presented one-pass data stream algorithms for detecting periodic sequences and cyclic shifts, and for measuring the distance to the closest periodic sequence or cyclic shift. Our principle goal was to minimize the space used, and all of our periodicity algorithms used $O(\mathrm{polylog}\, n)$ space. Our algorithms used a range of techniques including bilinear sketches and combining a Fourier change of basis transform with a range of sketching techniques. This second technique is particularly powerful and we would be surprised if it didn't have applications that were still to be discovered (either via the Fourier basis or other bases). An important future direction is analyzing the structure of the sketches formed by combining the transform and sketch matrices: among other things, this could lead to more time-efficient algorithms. Another question is to generalize our results in Sects. 3.2 and 3.3 to estimate the period of signals that conclude with a partial repetition. This was not an issue with time-series data since there would always be a point near the end of the stream where there had been an exact number of repetitions. In the turnstile model the issue is more complicated, but we are hopeful that a more involved analysis of the Fourier approach may yield results.

Acknowledgements. We thank an anonymous reviewer for mentioning that ideas from the pattern matching result in Ergün et al. can be applied to cyclic shifts in the time-series model without requiring a second pass. We also thank Graham Cormode for suggesting a simplification to our cyclic shift algorithm.

References

1. Alon, N., Matias, Y., Szegedy, M.: The space complexity of approximating the frequency moments. J. Comput. Syst. Sci. 58(1), 137–147 (1999)
2. Cormode, G., Muthukrishnan, S.: An improved data stream summary: The count-min sketch and its applications. J. Algorithms 55, 58–75 (2005)
3. Czumaj, A., Gasieniec, L.: On the complexity of determining the period of a string. In: CPM. pp. 412–422 (2000)
4. Ergün, F., Jowhari, H., Saglam, M.: Periodicity in streams. In: APPROX-RANDOM. pp. 545–559 (2010)
5. Ergün, F., Muthukrishnan, S., Sahinalp, S.C.: Periodicity testing with sublinear samples and space. ACM Transactions on Algorithms 6(2) (2010)
6. Gilbert, A.C., Guha, S., Indyk, P., Muthukrishnan, S., Strauss, M.: Near-optimal sparse fourier representations via sampling. In: STOC. pp. 152–161 (2002)
7. Hardy, G.H., Wright, E.M.: An Introduction to The Theory of Numbers (Fourth Edition). Oxford University Press (1960)
8. Indyk, P.: Stable distributions, pseudorandom generators, embeddings, and data stream computation. J. ACM 53(3), 307–323 (2006)
9. Indyk, P., Koudas, N., Muthukrishnan, S.: Identifying representative trends in massive time series data sets using sketches. In: VLDB. pp. 363–372 (2000)
10. Indyk, P., McGregor, A.: Declaring independence via the sketching of sketches. In: SODA. pp. 737–745 (2008)
11. Kane, D.M., Nelson, J., Woodruff, D.P.: On the exact space complexity of sketching and streaming small norms. In: SODA. pp. 1161–1178 (2010)
12. Kane, D.M., Nelson, J., Woodruff, D.P.: An optimal algorithm for the distinct elements problem. In: PODS. pp. 41–52 (2010)
13. Monemizadeh, M., Woodruff, D.P.: 1-pass relative-error L_p-sampling with applications. In: SODA (2010)
14. Muthukrishnan, S.: Data streams: Algorithms and applications. Foundations and Trends in Theoretical Computer Science 1(2) (2005)
15. Nisan, N.: Pseudorandom generators for space-bounded computation. Combinatorica 12, 449–461 (1992)
16. Porat, B., Porat, E.: Exact and approximate pattern matching in the streaming model. In: FOCS. pp. 315–323 (2009)
17. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. 26(5), 1484–1509 (1997)