SLIDE: STREAMING AND LOAD-ADAPTIVE PERIODICITY ESTIMATION

Deepak S. Turaga, Michail Vlachos, Spiros Papadimitriou, and Philip S. Yu

IBM T.J. Watson Research Center
19 Skyline Dr, Hawthorne, NY

ABSTRACT

We introduce a *CPU-adaptive* algorithm for spectrum estimation on streaming data. Our approach combines a fast and intelligent load-shedding scheme with a closed form incremental spectrum computation, which adapts gracefully to the available computational resources.

1. INTRODUCTION

Several emerging applications, including network traffic monitoring, financial data feeds, telemetry applications, medical data (e.g., ECGs), etc., contain streaming data arriving at high rates. Stream processing systems employ a collection of data analytic units which typically compete for available computational and memory resources. In such systems real-time response is crucial, therefore light-weight and efficient algorithms for processing and analyzing such data are highly desirable. Ultimately, however, the dominating constraint is the available system resources. Therefore, there is need for methods that can also gracefully degrade result accuracy based on CPU or memory availability; a good-enough answer is better than no answer at all.

Our goal is to efficiently estimate and update the periodogram of a signal, within a sliding time window. For fixed-length signals (with N samples), the periodogram can be estimated in $O(N \log N)$ time using the FFT. For dynamically updated sequences, the Momentary Fourier Transform (MFT) [1] can be employed to update the estimate over a sliding window. Recently, [2] proposes methods for periodicity estimation on streams, based on retaining the k most significant Fourier coefficients. However, none of the streaming approaches address the issue of resource adaptation.

A simple approach for load adaptation is to subsample the signal at regular intervals. However, this can lead to data aliasing and deteriorate the quality of the estimated periodogram. In contrast, our approach (see Fig. 2) uses a linear predictor which retains a sample only if its value cannot be predicted from its neighbors (leading to unevenly spaced samples). This scheme allows us to efficiently make on-the-fly decisions whether to discard a sample. We incorporate an estimator unit that adjusts the error tolerance of the predictor, based on available CPU time. Furthermore, we also introduce a closed-form Fourier approximation using uneven samples and we show how to update it incrementally. We call our load-adaptive methodology SLIDE (**S**treaming and **L**oad-adapt**i**ve Perio**d**icity **E**stimation). A schematic of our approach is provided on Fig. 1.

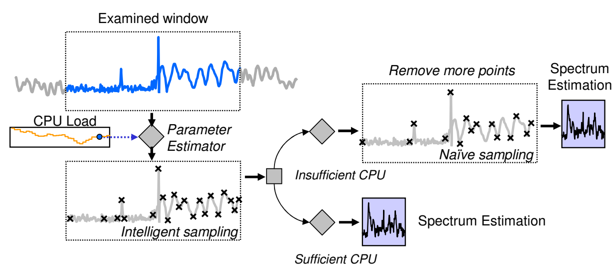

Fig. 1. Visual depiction of SLIDE.

Notation In the following, $x[k]$ is the k-th element ($k \in \mathbb{Z}^+$) of a discrete signal and $X[m]$ are its DFT coefficients. In this paper we use the periodogram of the signal as an estimator of the spectrum (and use the terms interchangeably). The notation $x[k_i]$ is used for the unevenly sampled signal, $k_i, i \in \mathbb{Z}^+$. Finally, we measure the complexity of our algorithms in terms of the number of additions (subtractions), multiplications and divisions (making the analysis independent of the underlying processor architecture). We label the complexity of a single multiplication as ξ_{Mul}, of a division as ξ_{Div} and of an addition/subtraction as ξ_{Sub}.

2. LOAD-SHEDDING SCHEME

We consider the typical problem of running spectral analysis where we slide a window across the temporal signal and incrementally update the signal's DFT (and the respective periodogram). As the data window slides by a fixed amount, we discard n_1 points from the beginning of the signal and add n_2 points to the end ($n_1 = n_2$ for evenly sampled signals). However, if the available CPU cycles do not allow us to update the DFT using all the points, we can adaptively prune the set of added points to $\hat{n}_2$ using uneven sub-sampling to meet the CPU constraint, while minimizing the impact on the accuracy of the updated DFT.

2.1. Intelligent sampling via a linear predictor

We determine if a sample can be discarded based on whether it can be linearly predicted from its neighbors. In particular, for sample k_i we compare the actual value $x[k_i]$ with the interpolated value $x^{int}[k_i]$:

$$x^{int}[k_i] = \frac{x[k_{i-1}](k_{i+1} - k_i) + x[k_{i+1}](k_i - k_{i-1})}{k_{i+1} - k_{i-1}} \quad (1)$$

where sample k_{i-1} is the last retained sample before sample k_i and sample k_{i+1} is the immediately following sample. If the instantaneous error $\delta_{k_i} := |x^{int}[k_i] - x[k_i]| \leq \Delta \times |x[k_i]|$ we can discard the sample k_i, otherwise we retain it. The parameter Δ is an adaptive threshold that determines the quality of the approximation. The complexity of this algorithm for M samples is :

$$\xi^{samp}(M) = (2\xi_{Mul} + 4\xi_{Sub} + \xi_{Div})(M - 2) \quad (2)$$

In Section 2.2 we discuss how to tune the threshold Δ in order to obtain the desired number of N samples, out of the original M samples.

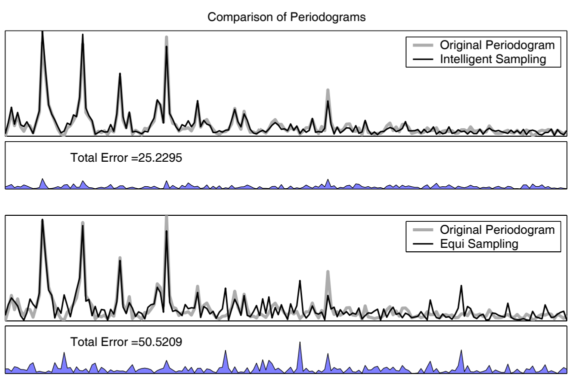

Fig. 2. Comparison of spectrum estimation errors for intelligent sampling and equi-sampling techniques.

Fig. 2 compares the spectrum estimates for a snapshot of a data stream, using the intelligent sampling method against a naïve equi-sampling technique. We execute our algorithm for a specific threshold and reduce the data points within a window from M down to N. We estimate the resulting periodogram (see section 3) as well the periodogram derived by equi-sampling every N/M points. Through the intelligent sampling we can provide higher quality reconstruction of the periodogram, because the important data stream features are retained.

It is possible to derive a worst case bound of the cumulative error for each discarded sample. Consider an interval $0 \leq k \leq K + 1$, for evenly sampled signal $x[k]$, where we retain samples $x[0]$ and $x[K + 1]$ and discard the rest. The absolute cumulative error $\epsilon_k := |x[k] - (x[K + 1] - x[0])/k|$ is the difference between the actual value and the interpolated value based on the retained samples $x[0]$ and $x[K+1]$, whereas the instantaneous error is the difference $\delta_k := |x[k] - (x[k+1] - x[0])/k|$ where the interpolation is based on the next sample

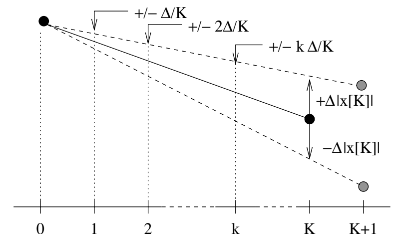

Fig. 3. Contribution of instantaneous error at sample K to cumulative error of previously discarded samples.

$x[k + 1]$, instead of $x[K + 1]$. For $x[K]$ the absolute cumulative error is the instantaneous error, i.e. $\epsilon_K = \delta_K$. By triangle similarity, it is easy to see (Fig. 3) that the worst case contribution of ϵ_K to the cumulative error for sample $x[k]$, $1 \leq k \leq K - 1$, is

$$(k/K)\epsilon_K \leq (k/K)\Delta|x[K]| \leq (k/K)\Delta \max_{j=1}^{K}|x[j]|.$$

Summing these up we get

$$\begin{aligned}\epsilon_k &\leq \Big(\Delta + k\Delta\big(1/(k+1) + \cdots + 1/K\big)\Big) \max\nolimits_k|x[k]| \\ &\approx \Big(\Delta + k\Delta\big(\ln K - \ln k\big)\Big) \max\nolimits_k|x[k]| \\ &= \Delta(1 + k\ln(K/k)) \max\nolimits_k|x[k]|,\end{aligned}$$

where we approximate the harmonic series by $\sum_{k=1}^{K} \frac{1}{k} \approx \ln K$. This is maximized for $k = K \max_k |x[k]|/e$ and the maximum (over all discarded samples) of the worst case cumulative error is $\Delta(1 + K/e)$. Since the sequence is variance scaled, $\max_k |x[k]|$ is typically small and can be ignored.

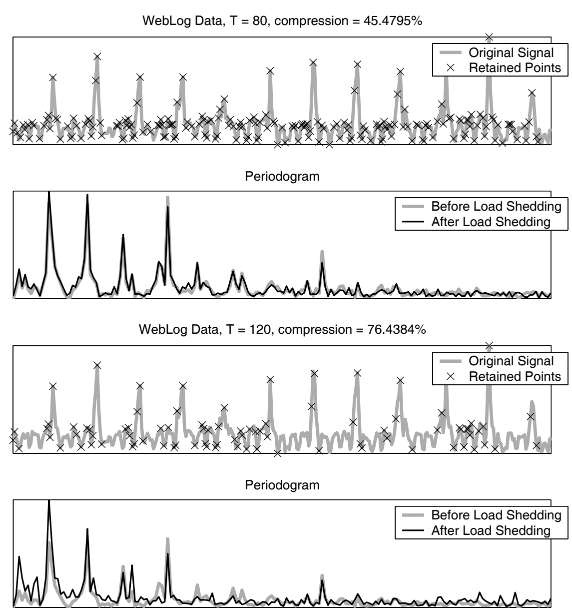

Fig. 4. Spectrum approximation for different threshold values

2.2. Threshold Estimator

Our goal is to predict the threshold Δ that will produce a desired number of uneven samples N during the next time window of length M in the future. The estimation of Δ is based on the behaviour of the signal during the past window of same size. Formally, let k_c be the current sample at the time we wish to readjust for available resources and let $x[k : k+M-1] := \big(x[k], x[k+1], \ldots, x[k+M-1]\big) \in \mathbb{R}^M$ be a window of M even samples. Then, we want a mapping $p : \mathbb{R}^M \times \mathbb{N} \mapsto \mathbb{R}$ giving us the threshold $\Delta = p\big(x[k_c - M + 1 : k_c], N\big)$.

However, the domain $\mathbb{R}^M \times \mathbb{N}$ of p has excessively high dimensionality. Our practical solution is to summarize $x[k : k+M-1]$ by a small set of *features* which capture the "irregularity" of the signal within that window. Formally, let $f : \mathbb{R}^M \mapsto \mathcal{F}$ be a mapping from the actual window to a sufficiently small feature set $\mathcal{F} \subseteq \mathbb{R}^d$, where $d \ll M$. The threshold estimator we use in practice is a mapping $\hat{p} : \mathcal{F} \times \mathbb{N} \mapsto \mathbb{R}$. The interpolation threshold we choose is $\hat{\Delta} = \hat{p}\big(f(x[k_c - M + 1 : k_c], N)\big)$.

The feature we use is a per-band variance f_{freq}. If $X[m]$, $0 \le m \le M-1$ are the DFT coefficients of $x[k : k+M-1]$, then we divide the frequencies into B bands of equal width to obtain

$$f_{\text{freq}}\big(x[k : k+M-1]\big) := (v_0, v_1, \ldots, v_{B-1}) \in \mathbb{R}^B,$$

where $v_j := \sum_{m=j(M-1)/B+1}^{(j+1)(M-1)/B} X^2[m]$, $0 \le j \le B-1$. Note that the DC coefficient $X[0]$ is omitted from f_{freq}. When $B = 1$, then $v_0 = \text{Var}(x[k : k+M-1])$. The per-band variance provides a finer characterization of the irregularities than just the variance. Small number of bands B (e.g., $B = 2$ or $B = 4$) provide good estimators without increasing the space complexity. These features can be incrementally maintained over a sliding window of size M.

The next step is how we compute the estimate $\hat{p}$. To that end, we use a training set $\mathcal{W} := \{W_j \mid W_j = x[i_j : i_j + M - 1], 1 \le j \le w\}$, consisting of w windows. We run our algorithm on each window for several different thresholds and get value of N for each of them. This produces a training set $\mathcal{T}$ of examples $S_l \in \mathcal{F} \times \mathbb{N} \times \mathbb{R}$, i.e., $\mathcal{T} := \big\{S_l \mid S_l = (f(W_{j_l}), N_l, \Delta_l),\ W_{j_l} \in \mathcal{W}\big\}$. We use the subscript l to identify elements of $\mathcal{T}$ ($1 \le l \le |\mathcal{T}|$).

We employ a k-NN (k nearest-neighbor) interpolation scheme to estimate $\hat{p}$, where the neighbor distance is computed only with respect to the features $f(W_{j_l})$ and the number of retained samples N_l. More specifically, if $W := x[k_c - M + 1 : k_c]$, then $\text{kNN}(f(W), N)$ is the set of k elements $S_l \in \mathcal{T}$ with the smallest distances $\|(f(W_{j_l}, N_l) - (f(W), N)\|$ from $(f(W), N)$, among all elements of $\mathcal{T}$. Then

$$\hat{\Delta} = \hat{p}_{\text{kNN}}(f(W), N) := \tfrac{1}{k} \sum_{S_l \in \text{kNN}(f(W),N)} \Delta_l.$$

Additionally, $\mathcal{T}$ can also be incrementally refined over time, by incorporating examples that haven't been encountered during the training phase. This can minimize the potential errors of the threshold estimator, even under significant changes in the stream pattern.

3. SPECTRUM ESTIMATION FOR UNEVENLY SAMPLED SIGNALS

Given N uneven samples $x[k_n]$, $0 \le n \le N-1$, we estimate the periodogram as follows. Conceptually, we first use linear interpolation (as for the sub-sampling) to reconstruct the evenly sampled signal $x[k]$ and then estimate the DFT $X[m]$ from it. However, we do not actually need to perform the interpolation and instead, we can directly derive closed form expressions, as in [3], for the DFT of $x[k_i]$ as:

$$X[m] = \textstyle\sum_{n=1}^{N-1} X_n[m] \tag{3}$$

where, for $m = 1, \ldots, M-1$,

$$\begin{aligned} X_n[m] = \tfrac{1}{(k_n - k_{n-1})(\frac{2\pi m}{M})^2} \Big[&\big(x[k_{n-1}] - x[k_n]\big) \cdot \\ &\cdot \big(e^{-j\frac{2\pi m k_{n-1}}{M}} - e^{-j\frac{2\pi m k_n}{M}}\big) + \\ &+ j\tfrac{2\pi m}{M}\big(x[k_n] e^{-j\frac{2\pi m k_n}{M}} - x[k_{n-1}] e^{-j\frac{2\pi m k_{n-1}}{M}}\big)\Big], \end{aligned} \tag{4}$$

and for $m = 0$,

$$X_n[0] = \tfrac{1}{2}(x[k_{n-1}] + x[t_n])(k_n - k_{n-1}). \tag{5}$$

Note that, while $x[k_i]$ has N samples, the DFT has at least $M = k_{N-1} - k_0$ samples to avoid time domain aliasing.

3.1. Incremental Spectrum Estimation for Streaming Data

A significant benefit of equation (3) is that the DFT for unevenly sampled signals can be evaluated incrementally. Hence, if we shift the window (of size M) such that n_1 points are discarded, and n_2 new points are added (i.e. we have $N + n_2 - n_1$ points), then the DFT of the signal may be updated as:

$$X^{new}[m] = X^{old}[m] - \textstyle\sum_{n=1}^{n_1} X_n[m] + \sum_{n=N}^{N+n_2-1} X_n[m] \tag{6}$$

We now examine the complexity of this update. Similar to prior analyses of FFT complexity, we do not consider the complexity of computing $e^{\frac{j2\pi m k_n}{M}}$ (and the intermediate value $\frac{2\pi m k_n}{M}$). The complexity of computing $X_n[m]$ is

$$\hat{\xi}_{nz} = 6\xi_{Mul} + 5\xi_{Sub} + \xi_{Div}, \qquad m = 1, \ldots, M-1, \tag{7}$$

$$\hat{\xi}_{z} = 2\xi_{Mul} + 2\xi_{Sub}, \qquad m = 0. \tag{8}$$

If we define $\hat{\xi}_{all} = (M-1)\hat{\xi}_{nz} + \hat{\xi}_z$, the total update complexity is

$$\xi^{up}(M, n_1, n_2) = (n_1 + n_2)[\hat{\xi}_{all} + M\xi_{Sub}] + 2M\xi_{Sub} \tag{9}$$

3.2. Complexity Reduction with Sub-sampling

When the window shifts, we cannot adapt the number of points discarded (n_1), however we can reduce the number of new points added (n_2) through intelligent sub-sampling. Consider

Dataset	Δ (%)	Window Compression (%)	Error Equi-Sampling	Error Intelligent	Improvement (%)
ECG	20	80.96	1627.55	450.79	72.30
	60	91.40	2434.59	1326.23	45.52
	100	95.79	2934.84	2171.04	26.02
EEG	20	6.73	79.79	2.76	96.53
	60	18.45	202.03	33.10	83.61
	100	32.81	221.16	105.99	52.07
RTT	20	35.90	147.76	26.68	81.94
	60	60.69	174.24	81.21	53.38
	100	75.55	210.69	123.98	41.15
WebTrace	20	13.97	22.08	4.04	81.70
	60	37.26	46.29	18.98	58.99
	100	61.36	52.31	47.52	9.15

Table 1. Accuracy of periodogram using Intelligent and Equi-Sampling

that the sub-sampling results in $\hat{n}_2$ samples ($\hat{n}_2 \leq n_2$). Comparing equations (9) and (2) we realize that the overall complexity of updating the spectrum estimate is reduced when:

$$\xi^{up}(M, n_1, n_2) \geq \xi^{up}(M, n_1, \hat{n}_2) + \xi^{samp}(n_2) \qquad (10)$$

Consider a simple case when $\hat{n}_2 = n_2 - 1$, i.e. sub-sampling discards one sample. The sub-sampling complexity is $(2\xi_{Mul} + 4\xi_{Sub} + \xi_{Div})(n_2 - 2)$ while the decrease in the update complexity is $(M-1)(6\xi_{Mul}+5\xi_{Sub}+\xi_{Div})+(2\xi_{Mul}+2\xi_{Sub})+M\xi_{Sub}$. Clearly, since $\hat{n}_2 < n_2 \leq M$, we can easily realize that the reduction in update complexity far outweighs the sub-sampling complexity. In general, equation (10) is always true when the sub-sampling reduces the number of samples (i.e when $\hat{n}_2 < n_2$). If, at a certain time, the CPU imposes a computation constraint of ξ^{limit}, and $\xi^{up}(M, n_1, n_2) > \xi^{limit}$ we can determine the optimal number of samples to retain $\hat{n}_2$, as:

$$\hat{n}_2 \leq \frac{\xi^{limit} - \xi^{samp}(n_2) - 2M\xi_{Sub}}{(M-1)(6\xi_{Mul}+5\xi_{Sub}+\xi_{Div})+(2\xi_{Mul}+2\xi_{Sub})+M\xi_{Sub}} - n_1 \qquad (11)$$

We can achieve this by tuning the threshold Δ based on the algorithm described in Section 2.2.

4. EXPERIMENTS

We examine two parameters of the resource-adaptive spectrum estimation: (1) The accuracy of the approximated periodogram, (2) The CPU adaptiveness of our technique, which depends on the quality of the threshold estimator. We measure the periodogram error on various datasets, and for different threshold values of the linear predictor. For a given threshold, a data window of length M will be reduced to N samples. We compare the quality of the approximated periodogram against a rudimentary approach that performs equi-sampling every N/M points. The results are given in Table 1 and clearly indicate that the proposed load-shedding scheme leads to high quality spectrum estimates. The reduction in the estimation error compared to equi-sampling, ranges from 10% to more than 90%.

Next, we measure the accuracy of the threshold estimator on a streaming automotive measurement dataset with a window $M = 1024$. We consider a synthetic CPU load, requiring sample reductions in the range $1/20 \leq N/M \leq 1$. We use $B = 2$ per-band variance features. The accuracy of the estimator is measured for each sliding data window as $N - \hat{N}$, where N is the desired number of samples to retain and $\hat{N}$ is the actual number. We evaluate both 1-NN interpolation and 5-NN interpolation. Histograms of the approximation error are provided in Fig. 5. The empirical error distribution indicates that for the majority of cases the estimation error is small. Furthermore, the instances of overestimated threshold (fewer remaining points than expected) are higher than the underestimated.

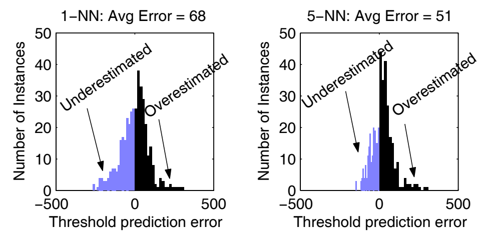

Fig. 5. Histogram of the threshold estimator error, indicating the cases of *overestimated* and *underestimated* threshold

5. CONCLUSION

We presented a spectrum estimation method that can adapt its quality based on the CPU load. Compared to equi-sampling, our intelligent load-shedding scheme can introduce improvements on the spectrum estimation ranging from 10% to 90%.

6. REFERENCES

[1] A. Papoulis. *Signal Analysis*. McGraw-Hill, 1977.

[2] A.C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss. Near-optimal Sparse Fourier Representations via Sampling. In *STOC*, pages 152–161, 2002.

[3] P. Castiglioni, M.D. Rienzo, and H. Yosh. A Computationally Efficient Algorithm for Online Spectral Analysis of Beat-to-Beat Signals. In *Computers in Cardiology:29, 417:420*, 2002.