Journal of Algorithms and Computation

journal homepage: http://jac.ut.ac.ir

Detecting and Reducing Error in Data by Tridiagonal Iterative Model

N. Bagherpour*[1] and A. Mohamadiyan†[1]

[1]Department of Engineering Sciences, University of Tehran, Tehran, 14155-6619, Iran.

ABSTRACT

Noise is a part of data whether the data is from measurement or experiment. There are a few techniques for fault detection and reduction to improve the data quality in recent years some of which are based on wavelet, orthogonalization and neural networks. In this paper, we suggest a low cost technique to improve the signal quality iteratively. In this method, we suggest a tridiagonal model which in fact describes the noise as a function of surrounding signal elements. To make the predicted noise more reliable, the algorithm is equipped with a learning/feedback approach. More precisely, in each iteration the most noisy elements are chosen and a tridiagonal matrix including some random parameters is suggested to model the larger noise values as a function of signal values around that entry. Our algorithm is used for both small and large noise values. We prove the linear convergence of the proposed algorithm. The numerical results confirm the efficiency of presented algorithm in most cases in comparison with orthogonalization based method introduced by Chen in 2015.

Keyword: Fault detection, Error Estimation, Noise Modeling, Machine Learning, Tridiagonal Linear Systems.

ARTICLE INFO

Article history:
Research paper
Received 12, January 2024
Accepted 19, February 2024
Available online 26, July 2024

AMS subject Classification: 15B04.

*Corresponding author: N. Bagherpour. Email: negin.bagherpour@ut.ac.ir
†mohamadiyan.777@ut.ac.ir

1 Introduction

Data analysis is a very common problem in machine learning and signal processing. Assume that a quantity X is measured in n different cases and we need to analyze the provided data. Since the data contains some error, whether it obtained by experiments or direct measurement tools, we need to detect and reduce the noise before starting any analysis. Different noise reduction algorithms are suggested with specific applications for audios or images; see for example [2, 1]. In recent years wavelet and least squares played an important role in suggested noise reduction algorithms. Chen [4] presented an algorithm based on noise orthogonalization. He also outlined a novel noise reduction technique by use of reverse least squares and shaping regularization [4, 8]. Moreover, he developed the first wavelet based algorithm for noise reduction in 2017 [3]. On the other hand, Huang [5] provided a singular spectrum analysis for 3D random noise. A few neural network based algorithms are also suggested for noise detection [6, 7]. There are some complications in solving noise detection problem such as difficult mathematical modeling, high computational cost and high sensitivity to noise quality and size. Learning approaches can solve these issues by following the error trend in consequtive iterations. In each iteration of a learning algorithm, the current noise estimation is evaluated to suggest a proper update. In this paper we provide a new algorithm for detecting and reducing the noise which has benefits over existing methods in some cases. Each iteration of this algorithm consists of three main steps:
1) It suggests a tridiagonal model for the signal entries with more noisy behavior.
2) The noise is approximated by solving the tridiagonal model.
3) It updates the input signal considering the assumed noise.
Our contributions are as follows: 1) We outline a two phase noise reduction algorithm which suggests a tridiagonal model to estimate noise, compute an approximated noise-free signal and check the improvement to verify the quality or revise the noise in each iteration, 2) The hybrid of regression phase besides the learning phase make the noise reduction process faster, 3) The complexity of the proposed algorithm is relatively low and 4) We can substitute the tridiagonal model by any proper matrix structure based on signal characteristics. We categorized some special cases in Section 3.

2 Our Algorithm

In this section the proposed algorithm is presented. The input is a data array consisting of noise in some entries and the output is the corrected data. There is no specific condition for the input structure; however, the best results obtained for scattered noise across most of the data entries. Now, we present the main steps of the algorithm and its convergence proof.

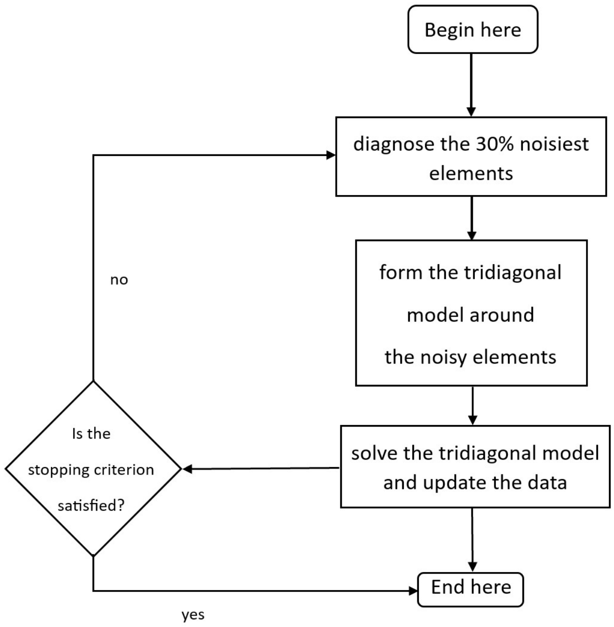

Figure 1: Flowchart of the Low-dimension Tridiagonal (LTD) Algorithm

2.1 Initialization

First, the moving average method is implied to provide a relatively smooth trend. As an initial value, the noise is assumed to have standard normal distribution. It then goes on with detecting the most noisy elements from ruined data and trying to reduce the error.

2.2 Approximation Loop

The algorithm enters a while loop that continues as long as the error E is greater than the specified tolerance and the iteration counter is within a preassumed range. Inside this loop, the second-order differences of data entries, GT, and its maximum value M is computed. The elements of GT satisfying $abs(DD) - 0.7 * M > 0$ are selected and stored in gt as proper canditates for noise reduction. These elements will be used in the approximation step. Actually, a tridiagonal system with some arbitrry elements is designed to decrease the noise. The linear system $Tf = N$ is solved to obtain the updated approximation vector f instead of gt. The error E is updated by calculating the norm of the difference between f and gt. The counter k is incremented by 1 to keep track of the number of iterations The Algorithm starts with guessing noise through the simple idea, moving average and assuming a normal distribution for the noise. These midpoints will be used in subsequent calculations. It then goes on with detecting the most noisy elements from ruined data and trying to reduce the error.

2.3 Approximation loop

The algorithm enters a while loop that continues as long as the error E is greater than the specified tolerance and the iteration counter is within a preassumed range. Inside this loop, the second-order differences of data entries, GT, and its maximum value M is computed. The elements of GT satisfying $abs(DD) - 0.7 * M > 0$ are selected and stored in gt as proper canditates for noise reduction. These elements will be used in the approximation step. Actually, a tridiagonal system with some arbitrry elements is designed to decrease the noise. The linear system $Tf = N$ is solved to obtain the updated approximation vector f instead of gt. The error E is updated by calculating the norm of the difference between f and gt. The counter k is incremented by 1 to keep track of the number of iterations.

2.4 Post-processing and analysis

Once the while loop is finished, the measured values GT are updated by replacing the selected elements in gt with the corresponding elements from f. This reflects the refined approximation. The measured values GT are plotted against the exact values G_{exact} to visualize the approximation and assess the quality of the results; see e.g. Figure 5. The mean squared errors (mse1 and mse2) between the exact values Gexact and the measured values $G_{measured}$ are also computed and stored in Table 2. These metrics provide a quantitative assessment of the approximation's accuracy. The algorithm iterates through the approximation loop, adjusting the approximation vector f based on the calculated tridiagonal system around the selected elements from gt. The goal is to refine the approximation and minimize the error between the measured values and the true values. The process continues until the error falls below the specified tolerance or the maximum number of iterations is reached.

2.5 Pseudocode

3 Convergence Analysis

In Theorem 1, we show the linear convergence rate of LTD algorithm, while our numerical results confirm the superlinear convergence rate.

Theorem 3.1. *The LTD algorithm reduce the scattered noise with a linear convergence rate.*

Proof Let assume there a considerable noise in ith enrty which cause a large second difference value between ith and $i + 1$th data entries. Substituting the noisy values with their tridiagonal approximation is in fact equivalent to assume a straight line instead of the peicewise linear form; see Figure 2. More precisely, in iteration $k + 1$ we have

$$e_{k+1} < \lambda e_k + (1 - \lambda)e_{k-1} \approx \alpha e_k.$$

Algorithm 1 Limited Tridiagonal Approach for Reducing Scattered Noise

```
procedure LTD(Nn, mydelta, kmax)
    Initialize variables and arrays
    Generate exact data G_exact
    Create a probability distribution PD with desired characteristics
    Generate noise data N_Data based on the distribution
    Add the noise data to the exact data to obtain GT
    Calculate midpoint values gm from GT
    Compute mean and standard deviation of differences between gm and GT
    Initialize error E and measured data G_measured
    while E > mydelta do
        Compute second-order differences DD of GT
        Find the maximum difference M in DD
        Select elements of GT based on a condition to obtain gt
        Determine the length n of gt
        if n > 0 then
            for k = 1 to kmax do
                Generate input values In
                Compute probability density values N based on PD1
                Initialize intermediate arrays and variables
                Perform random-based calculations to update f
                Construct a tridiagonal matrix T based on d, mu, and rho
                Solve the linear system T · f = N
                Update error E based on the difference between f and gt
            end for
        end if
        Update the measured data G_measured based on gt and f
    end while
    Calculate mean squared errors mse1 and mse2
    Return the final values of f, mse1, and mse2
end procedure
```

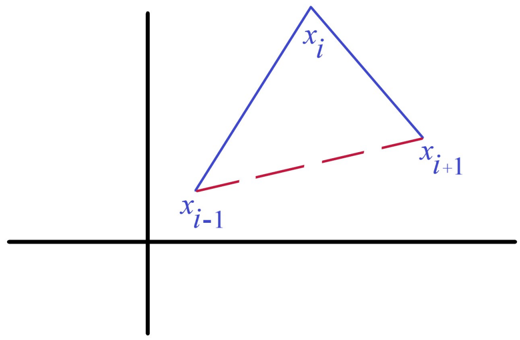

Figure 2: Tridiagonal modeling

Now, we consider three technically important points about the convergence of LTD algorithm:

1. Our numerical results confirm at leats superlinear convergence; see Figure 3.

2. As described in Section 2, in each iteration of LTD algorithm a low-dimension tridiagonal system is needed to improve the signal quality. To determine the most noisy elements, we suggest to select the entries with second difference greater than 70 percent of its maximum value as a rule of thumb. Based on our observtions it is a proper choice in most of the tests.

3. The values of $kmax$ and δ depends on data size. In Table 1 proper choices are represented.

Table 1: Suggested parametes

n	$kmax$	δ
100	10	E-06
500	10	E-05
1000	100	E-04
5000	100	E-04
10000	200	E-03

In Section 2 we assumed a tridiagonal model for the noise behavior. Although it works properly in most tests, there is no force to use the tridiagonal model. Actually, if the noisy elements are scattered wider, there is no reason to assume the noise concentrated around the diagonal entries. Hence, five-diagnal system might improve the efficiency.

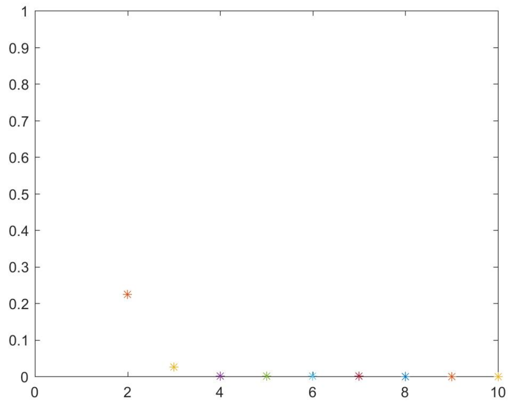

Figure 3: Superlinear decrease in error

4 Numerical Results

In this section, we present the numerical results. We implement LTD and MSSA algorithms in MATLAB 2023b on a computer with a 2.4 GHz core i5 CPU and 8Gb RAM. We then test the codes on real and randomly generated noisy data. To generte random tests, both rand and randn commands are used for the exact data and a normal noise is added. In each experiment, the goal is to capture the added noise as fast as possible. We compare both MSE and time to show the effectiveness of our proposed algorithm in approximating the noise term more precisely and in lower time. As shown in Figure 4, it is confirmed that LTD is faster than MSSA. This figure demonstrates the Dolan More profiles which is in fact a performance ratio and greater value means more test problems are solved in minimum time.

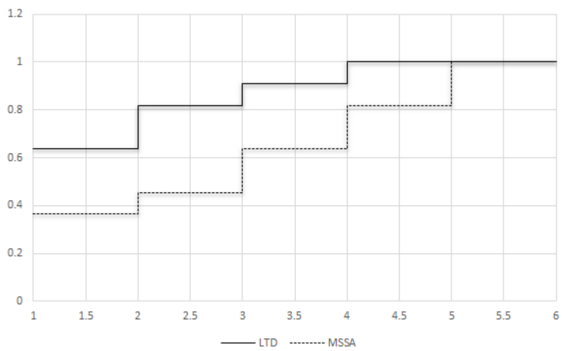

Figure 4: Dolan More time profiles for LTD and MSSA algorithms

Moreover, the average time and MSE are reported for random tests. To provide more accurate results we repeat each experiment 20 times and report the average results in Table 2. As large as the data size is, MSSA tends to outperform our algorithm; however, for data size not greater than 1000 LTD has two desirable features including lower MSE and lower computing time.

Table 2: Time and MSE comparison for LTD and MSSA [4]

n	LTD time	MSSA time	LTD MSE	MSSA MSE
100	0.2669	1.1526	0.0813	0.1149
500	0.5703	1.2832	0.0271	0.1198
1000	1.9634	1.4939	0.0164	0.1231
5000	4.7623	3.6403	0.0132	0.1238
10000	21.1465	6.1078	0.0125	0.1246

We also presented Figure 5 to confirm the decreasing behavior of noise. More precisely, in a test problem of size 60, the difference between measured data and our algorithm approximation is lower than the exact noise values generated randomly.

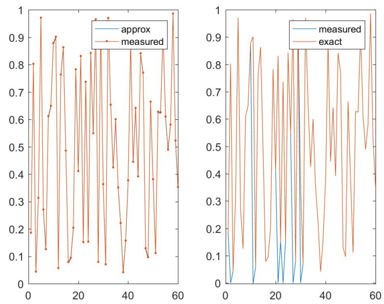

Figure 5: Noise reduction process in a test problem with size 60. The left is resulted by our noise reduction process while the right shows the initial noise which is clearly greater

Finally, our tests on real data extracted from noisy sounds and images shows the efficiency of our proposed algorithm in reducing scattered noise fast and accurately. In Figure 6 a noisy sound and a noisy image signals are plotted. We note that to provide a one dimentional data, we substitute the image matrix with its row average.
In Figure 7, the signals are denoised by use of LTD algorithm. It can be easily seen that our proposed approach successfully reduced these type of scattered noises.

5 Concluding Remarks

Noise reduction was the target of this paper. The most important contribution was to outline a low computational cost algorithm for detecting small data fluctuations. In our suggested algorithm two phase are introduced: first was to suggest a local tridiagonal model around the most noisy entries to detect the noise and the second was to design a learn/feedback process to decide whether the predicted noise satisfied the necessary quality conditions in next iterations. According to presented numerical results, the presented

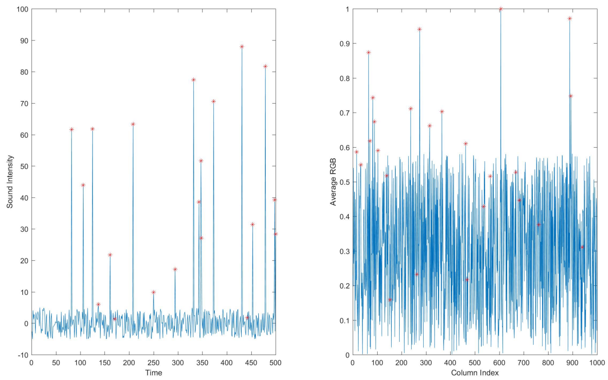

Figure 6: Two scattered noisy input signals: the left shows the intensity of a noisy sound and the right shows a row average of a noisy image

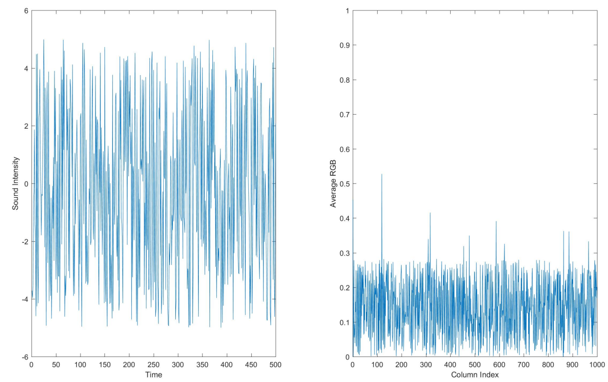

Figure 7: Denoised signals described in Figure 6

algorithm was able to detect the small fluctuations with lower men squared error in lower computational time. Working on optimal parallelization techniques can be suggested for future research to denoise large scale data sets.

5.1 Acknowledgments

The authors would like to thank the refrees for their valuable and supportive comments.

References

[1] R. S. Burwen. A dynamic noise filter. *J. Audio Eng. Soc.*, 19(1):1–7, 1971.

[2] R. C. Kompander verbessert magnettonkopie. *Radion Mentor (in German)*, 4:301–303, 1965.

[3] Y. Chen. Probing the subsurface karst features using time frequency decomposition. *Interpretation*, 4(4):533–542, 2017.

[4] Y. Chen and S. Fomel. Random noise attenuation using local signal-and-noise orthogonalization. *Geophysics*, 80(6):1–9, 2015.

[5] W. Huang, R. Wang, Y. Chen, H. Li, and S. Gan. Damped multi-channel singular spectrum analysis for 3d random noise attenuation. *Geophysics*, 81(4):261–270, 2016.

[6] M. Mafi, H. Martin, J. Andrian, A. Barreto, M. Cabrerizo, and M. Adjouadi. A comprehensive survey on impulse and gaussian denoising filters for digital images. *Signl Processing*, 157:236–260, 2019.

[7] A. Orazaev, P. Lyakhov, V. Baboshina, and D. Kalita. Neural network system for recognizing images affected by random-valued impulse noise. *Applied Sciences*, 13(3):1585–1594, 2023.

[8] Z. Xue, Y. Chen, S. Fomel, and J. Sun. Seismic imaging of incomplete data and simultaneous-source data using least squares reverse time migration with shaping regularization. *Geophysics*, 81(1):11–20, 2016.