



Article A Machine Learning-Based Seismic Data Compression and Interpretation Using a Novel Shifted-Matrix Decomposition Algorithm

Milan Brankovic^{1,*}, Eduardo Gildin², Richard L. Gibson³ and Mark E. Everett¹

- ¹ Geology and Geophysics, Texas A&M University, 3115 TAMU, College Station, TX 77843-3115, USA; everett@geo.tamu.edu
- ² Harold Vance Department of Petroleum Engineering, Texas A&M University, 3116 TAMU, College Station, TX 77843-3116, USA; egildin@tamu.edu
- ³ NanoSeis, Department of Geology and Geophysics, Texas A&M University, 3115 TAMU, College Station, TX 77843-3115, USA; rick.gibson@nanoseis.com
- * Correspondence: mbrankov@tamu.edu or milanovstalni@gmail.com

Abstract: Seismic data provides integral information in geophysical exploration, for locating hydrocarbon rich areas as well as for fracture monitoring during well stimulation. Because of its high frequency acquisition rate and dense spatial sampling, distributed acoustic sensing (DAS) has seen increasing application in microseimic monitoring. Given large volumes of data to be analyzed in real-time and impractical memory and storage requirements, fast compression and accurate interpretation methods are necessary for real-time monitoring campaigns using DAS. In response to the developments in data acquisition, we have created shifted-matrix decomposition (SMD) to compress seismic data by storing it into pairs of singular vectors coupled with shift vectors. This is achieved by shifting the columns of a matrix of seismic data before applying singular value decomposition (SVD) to it to extract a pair of singular vectors. The purpose of SMD is data denoising as well as compression, as reconstructing seismic data from its compressed form creates a denoised version of the original data. By analyzing the data in its compressed form, we can also run signal detection and velocity estimation analysis. Therefore, the developed algorithm can simultaneously compress and denoise seismic data while also analyzing compressed data to estimate signal presence and wave velocities. To show its efficiency, we compare SMD to local SVD and structure-oriented SVD, which are similar SVD-based methods used only for denoising seismic data. While the development of SMD is motivated by the increasing use of DAS, SMD can be applied to any seismic data obtained from a large number of receivers. For example, here we present initial applications of SMD to readily available marine seismic data.

Keywords: data compression; noise reduction; machine learning; singular value decomposition

1. Introduction

The last decade has seen a great increase in hydrocarbon production from unconventional reservoirs. However, there are still many challenges in predicting their production potential. The quantity of hydrocarbons produced depends on the distribution and quantity of fractures present in the reservoir. Fracture identification and monitoring can be done by analyzing microseismic data [1]. With the goal of improving the ability to track the fracture distribution, the amount of seismic data acquired during reservoir monitoring has been increasing.

Recent developments in reservoir monitoring use fiber optic cables to record both low frequency strain and seismic waves in a technique called distributed acoustic sensing (DAS) [2]. DAS provides a new and unique view of the reservoir by recording strain rate data with a high sample rate and dense spatial sampling. Higher recording rates and more



Citation: Brankovic, M.; Gildin, E.; Gibson, R.L.; Everett, M.E. A Machine Learning-Based Seismic Data Compression and Interpretation Using a Novel Shifted-Matrix Decomposition Algorithm. *Appl. Sci.* 2021, *11*, 4874. https://doi.org/ 10.3390/app11114874

Academic Editors: Birendra Jha and José A. Peláez

Received: 31 March 2021 Accepted: 21 May 2021 Published: 26 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). receiver positions increase the amount of microseismic activity recorded while monitoring the reservoir. Recording microseismic events with DAS significantly increases the memory requirement of the monitoring data. Thus, data processing methods that can compress the data and reduce its noise would complement the new instrumental developments in reservoir monitoring.

Signal processing algorithms are often based on the transformation of signal into a new domain. Early examples of compression involve discrete cosine transform [3] and wavelet transforms [4–6] which use cosine functions or wavelets to represent the data in order to reduce the memory requirements. Reduced-rank methods, which approximate the noiseless seismic data using low rank matrices and tensors, have been used in noise reduction [7] while also reconstructing missing data [8,9]. In recent years, dictionary learning has seen wide application in seismic data. Because of its ability to provide a compact and informative representation of seismic signals, it has been used for both noise reduction [10–12] and data compression [13,14]. Unfortunately, the drawback of the dictionary learning applications is the computation time that comes with learning and updating the dictionary. There have been successful efforts to reduce the associated computational times [15], but the computational cost of applying dictionary learning to microseismic DAS recordings is still to great. Thus far, dictionary learning methods have been applied to data collected by geophones, which can be very large but are still much smaller than data obtained by a single fiber-optic cable which can sample strain thousands of times a second on hundreds, even thousands of receiver locations. Since DAS is used for microseismic monitoring, the great amounts of data would need to be processed in real time, which puts a constraint on the computation time of processing methods.

To achieve computationally efficient compression and denoising, we created a new data decomposition method by improving and further developing ideas developed as a part of the local SVD [16]. Similarly to the previously mentioned reduced-rank methods, local SVD applies SVD to a window in seismic data and represents the data using a small number of singular vectors. What makes local SVD unique is the process of shifting the columns of the window to maximize their correlation prior to applying SVD. This allows singular vectors to capture the signal in seismic data with high accuracy, while ignoring most of the noise. Once the data in the window is processed with SVD, the window is moved to the next location. This process is repeated until column shifting and SVD have been applied to every part of the data matrix. The path of the moving window as well as the number of singular vectors used at each location are predetermined. Local SVD can enhance a seismic data set even if it contains multiple wave arrivals. However, local SVD struggles to capture seismic signals if waves with different dips are interfering or present in the same window. The "dip" of the wave refers to the slope of the wave in the matrix, or how much the row position of a wave changes as we move from one column to its adjacent column.

Some of the problems encountered with local SVD were resolved with the development of structure-oriented singular value decomposition (SOSVD) [17]. By using plane wave destruction [18], SOSVD can identify several dominant slopes at each window location. While using plane wave destruction provides a noticeable improvement, numerical artifacts can still appear at the intersection of the waves with different slopes.

Our method is inspired by the SOSVD and local SVD, and it also shifts the columns of the matrix before applying SVD. However, in order to avoid numerical artifacts, we use different processes to determine how columns should be shifted. Specifically, we do not use a moving window with a predetermined path. Instead, we use a geometric mean filter that adaptively chooses which elements to use in the geometric mean. We apply the geometric mean filter to seismic data in order to highlight areas which contain wave arrivals. The windows from the matrix to which we apply SMD depend on the results of the geometric mean filter. This allows us to use more singular vectors to describe areas with multiple wave arrivals, and fewer singular vectors to describe areas dominated by noise. Additionally, local SVD and SOSVD use SVD results solely to denoise seismic data. They use them to reconstruct denoised version of seismic data as soon as they obtain them and do not discuss the compression achieved by storing SVD results. In our work, instead of reconstructing the data immediately, we store the SVD results. The final product of our algorithm is the collection of SVD results, that can later be reconstructed into the denoised version of original data. By doing this, our algorithm can be used for data compression as well as noise suppression. We call our new method the shifted-matrix decomposition, or SMD for short.

It should also be noted that DAS has seen increasing use outside geophysical exploration. Distributed acoustic sensing has been used to record signals from earthquakes and volcanic events [19]. Because of its unprecedented spatial and temporal resolutions, DAS is expected to see increasing use in earthquake monitoring, imaging of faults and many other geologic formations, and hazard assessment [20]. The growing potential of DAS application outside of geophysical exploration, adds importance to our method, which we believe will play an integral role in the processing of DAS data.

We organize the paper as follows: First we give an overview of singular value decomposition and present two simple examples that show advantages and drawbacks of its application to seismic data. Next, we demonstrate the improvements achieved by shifting the traces before extracting singular vectors. In the following subsection, we describe in detail each step of the SMD algorithm. The SMD algorithm depends on several parameters. The optimal values of said parameters are determined in a machine learning stage following the algorithm description subsection. In the training stage we use seismic field data obtained from marine seismic gathers, which has a large amount of interference between coherent waves, and noise which can be difficult to differentiate from signal. After training on marine seismic gathers, SMD provides accurate results on other seismic data as well as marine seismic gathers, which allows us to skip the training stage in future applications. To confirm the accuracy of SMD, we reproduce synthetic data from [17], and compare results of SMD to results of local SVD and SOSVD. The SMD is then tested on real seismic data. While SMD is primarily developed for application to microseismic data recorded by DAS, we currently do not have access to such data. Instead, we apply SMD to field data obtained from marine seismic gathers [21]. The results of applying SMD to field data are used to reconstruct a denoised version of the data as well as to estimate the elastic wave velocity. Finally, we discuss possible future applications of SMD and how its results could be used in signal detection during seismic monitoring.

2. Materials and Methods

2.1. Singular Value Decomposition

Consider seismic data stored in a matrix $\mathbf{M} \in \mathbb{R}^{n_t,n_r}$, where n_r is the number of the receivers recording the data and n_t is the number of time samples. An element of the matrix $M_{i,j}$ describes the ground motion at the j^{th} receiver, and at the i^{th} time step. Singular value decomposition method is a common matrix decomposition method that can be used to express the matrix \mathbf{M} of rank r as the product of matrices:

$$\mathbf{M} = \mathbf{U}_{\mathbf{r}} \mathbf{D}_{\mathbf{r}} \mathbf{V}_{\mathbf{r}}^{T}, \tag{1}$$

where $\mathbf{U_r} = [\mathbf{u^1}, \mathbf{u^2}, \dots, \mathbf{u^r}]$ contains the *r* left singular vectors as columns, the diagonal matrix $\mathbf{D_r} = diag(\lambda^1, \lambda^2, ..., \lambda^r)$ contains the *r* singular values, and $\mathbf{V_r} = [\mathbf{v^1}, \mathbf{v^2}, ..., \mathbf{v^r}]$ contains the *r* right singular vectors as columns. In traditional SVD, the left singular vectors and right singular vectors are normalized. However, multiplying the right singular vector by the singular value, allows us to store the singular value in the right singular vector, which slightly reduces the memory requirements of SMD results. For that reason, from this point on, the "right singular vector" refers to the normalized right singular vector multiplied by the singular value. Equivalent to Equation (1), SVD can also be used to express the the matrix **M** as a sum of outer products of left singular vectors and right singular vectors weighted by singular values [7]. However, due to the way we define the

right singular vector in this work, we can express is as just a sum of outer products of left singular vectors and right singular vectors:

$$\mathbf{M} = \sum_{k=1}^{r} \mathbf{u}^{k} \mathbf{v}^{k^{T}}.$$
 (2)

While Equation (2) may be less common, regarding SVD as sum of outer products is helpful for understanding the processes of SMD. In this work, we refer to any column from **U** as a column vector, and any column from **V** as a row vector. This is because in the outer product $\mathbf{u}^{\mathbf{k}}\mathbf{v}^{\mathbf{k}^{T}}$ all columns are multiples of the column vector $\mathbf{u}^{\mathbf{k}}$, and all rows are multiples of the row vector $\mathbf{v}^{\mathbf{k}^{T}}$.

Singular value decomposition can also be used as a data compression method. We assume singular values in D_r decrease from first to last row. When applying SVD to seismic data, the initial singular values are much greater than the average singular value in D_r , and describe most of the signal. By neglecting the small singular values, the singular value decomposition can be used to approximate matrix M as

$$\mathbf{M} \approx \sum_{k=1}^{r_s} \mathbf{u}^k \mathbf{v}^{k^T},\tag{3}$$

where $(r_s \ll r)$. By using only the first few singular vectors to describe the the matrix **M** we can significantly decrease the memory requirement of the seismic data. In some cases, a few pairs of singular vectors can very accurately describe the data stored in **M**. For example, if a wave packet arrives at all receivers at the same time, all columns in **M** will contain the same pattern, and in that case, we can represent **M** using a single outer product (see Figure 1).



Figure 1. (**a**) A matrix showing a single wave arriving to all receivers at the same time. (**b**) A pair of singular vectors obtained by applying SVD to the matrix. (**c**) The outer product of the two singular vectors which matches the original matrix.

However, if the wave packet arrives at the receivers at different times, as in Figure 2, we can no longer represent the matrix **M** using a single outer product. Figure 2 shows an example of the errors that will occur if reconstruction of the data from a single outer product is attempted. Though in practice this would never be done, the example does show that greater data compression can be achieved in this framework if signals in data are aligned prior to a decomposition.

If there is significant variation in signal time position for different receivers, a large number of singular vectors is needed to properly represent the signal. This negatively affects the compression as well as denoising, since using more singular vectors would also include representing more of the noise. Any methods utilizing SVD, such as PCA [22], will be subject to the same challenges. We note that PCA and SVD have been used interchangeably in terms of data compression. A mathematical relation between them is demonstrated in [23].



Figure 2. (a) A matrix showing a wave arriving to the receivers at different times (b) A pair of singular vectors obtained by applying SVD to the matrix. (c) The outer product of the pair of singular vectors which is very different from the original matrix.

The issues encountered with regular SVD (or PCA) were first addressed with the development of a local SVD by Bekara and Van der Baan [16], and later improvements led to the development of SOSVD by Gan et al. [17]. In both of these methods SVD is applied to a window M_w from matrix M with columns shifted in such a way to maximize the correlation between columns. The shape of the window doesn't change as the columns are shifted. This can be achieved in several ways, one of which is simply applying a wrap around condition so that values off the end of a matrix column, for example, are inserted at its beginning (and vice versa). An example of window shifting is described in Figure 3.



Figure 3. (a) The original data matrix M. (b) A selected window M_w from the original data matrix M. (c) The matrix M_{sw} , obtained by shifting the columns of M_w to maximize their correlation.

The resulting shifted window (M_{sw}) resembles the matrix from Figure 1, and can be described by the equation:

$$\mathbf{M}_{\mathbf{sw}} = \chi(\mathbf{M}_{\mathbf{w}}, \mathbf{s}) \tag{4}$$

where χ is the "shift" operator, $\mathbf{M}_{\mathbf{w}}$ is the window subset from \mathbf{M} and \mathbf{s} is the shift vector. The "shift" operator takes a matrix and a shift vector as its arguments and shifts the columns of the matrix based on the values prescribed by the shift vector. For example, if the value of \mathbf{s} for column j is n, then we would replace an element $M_{i,j}$ with the element $M_{i+n,j}$. The matrix decomposition used in local SVD and SOSVD can also be described with the Equation (5):

$$\mathbf{M}_{\mathbf{w}} \approx \sum_{k} \chi(\mathbf{u}^{\mathbf{k}} \mathbf{v}^{\mathbf{k}^{T}}, -\mathbf{s}^{\mathbf{k}})$$
(5)

where $\mathbf{s}^{\mathbf{k}}$ is the shift vector corresponding to the k^{th} pair of left and right singular vectors $\mathbf{u}^{\mathbf{k}}$ and $\mathbf{v}^{\mathbf{k}}$. By using Equation (5) and plugging in \mathbf{M} for $\mathbf{M}_{\mathbf{w}}$, the matrix \mathbf{M} from the previous example can be presented with a single outer product coupled with a shift vector as shown in Figure 4. A simple quantitative example that shows the effectiveness of column-shifting with a wave similar to the one from Figure 4 can be found in the Appendix A.



Figure 4. (a) A matrix showing a wave arriving to the receivers at different times. (b) The matrix being expressed as a pair of singular vectors and a shift vector. (c) The singular vectors and the shift vector being used to reconstruct a matrix that matches the original.

Unfortunately, both local SVD and SOSVD seem to fail to accurately denoise seismic data which contains interfering waves with different dips. In such scenarios, numerical artifacts appear around the area of intersection. We solve this problem by developing shifted-matrix decomposition (SMD) which uses a very different process for determining the shift vector. The following subsection will give a detailed description of the processes SMD uses to obtain the shift vectors and the pairs of singular vectors.

2.2. SMD Algorithm

The SMD algorithm can be described by the following steps:

- 1. Using a geometric mean filter, choose a specific point (row number and column number) in the data matrix **M** at which a displacement (or pressure) from a coherent wave was most likely recorded.
- 2. Using cross-correlation, find this wave in as many surrounding columns as possible. For each column, the relative row positions of the said wave are recorded in the shift vector.
- 3. Shift the columns of **M** using the shift vector **s** and record the row vector and the column vector. Subtract the shifted outer product of row vector and column vector from matrix **M** and shift the columns of **M** back to their original positions.
- 4. Repeat steps 1–3 until a certain performance criterion is satisfied.

A visual representation of the algorithm can also be found in Figure 5.

To provide a more in-depth understanding of the algorithm, in the following paragraphs we will give a detailed description of each of the four steps.



Figure 5. A flowchart describing the SMD algorithm.

2.2.1. Step 1

To identify a point in matrix \mathbf{M} at which a displacement (or pressure) from a coherent wave is likely recorded, we start by applying a geometric mean filter to the data matrix \mathbf{M} , to obtain the matrix \mathbf{E} (Equation (6)):

$$E_{i,j} = |\prod_{q=-n_E}^{n_E} M_{(i+\delta_{i,j}(q)),(j+q)}|^{\frac{1}{2n+1}}.$$
(6)

For a position (i, j), the parameter n_E indicates that n_E columns left of the position (i, j) and n_E columns right of the position (i, j) will be used when calculating $E_{i,j}$. If a position (i, j) is close to the first or the final column, fewer elements are used to calculate the geometric mean $E_{i,j}$ as to avoid stepping outside the matrix boundaries. The row positions of elements used in the geometric mean are also constrained in order to always remain within the matrix boundaries.

What makes this filter unique is the choice of elements which are used when calculating the geometric mean. From each of the $2n_E$ surrounding columns, only one element is used in the geometric mean. The variable δ in Equation (6) indicates which element is used in the geometric mean from each column. For example, when calculating the value of the the geometric mean in position (i, j), from column j + q we take the element at row position $i + \delta_{i,j}(q)$ to be used in the geometric mean. The process for determining the said set of elements is described with Figure 6. Consider calculating the geometric mean in order to determine the value of $E_{i,j}$. Before calculating the geometric mean, we must determine the set of elements used in the mean (determine the values of $\delta_{i,j}$). We start from the j^{th} column, from which we always use the element $M_{i,j}$ ($\delta_{i,j}(0) = 0$). Then, we proceed to find which elements to use from adjacent columns (Figure 6a). Assuming $M_{i,j}$ is positive, for column j + 1, we pick the element between positions (i - m, j + 1) and (i + m, j + 1) which has the highest value. If the element $M_{i,j}$ were negative, we would pick the lowest value. The parameter m represents the maximum dip of the wave rounded up to the nearest integer. The same process is used to determine which element to use from column j - 1. For any column $j + n_s$ where $1 < n_s \leq n_E$, for the geometric mean, we pick the element between positions $(i_s - 1, j + n_s)$ and $(i_s + 1, j + n_s)$ with the highest value (Figure 6b,c). The row number i_s is determined by following a linear trend based on the row position i in column j and the picked row position $i + \delta_{i,j}(n_s - 1)$ in the column $j + n_s - 1$. The same process is used to determine which element below j - 1.

The geometric mean filter is designed to imitate the process humans use for recognizing coherent waves in the data. When differentiating signal from noise, the distribution of displacement (or pressure) along a long range of receivers plays an important role. A weak wave might have an amplitude that is even smaller than that of noise in few noisy areas. However, if the displacement (or pressure) from the wave is present in a long range of receivers, and its arrival time follows a hyperbolic trend, a data reviewer would certainly notice the wave. The goal of the geometric mean filter is to assign higher values to such a weak wave than to a noisy area with high amplitudes. By testing the geometric mean filter on a large number of examples, we found few cases in which the weak wave was not assigned higher values than some areas of the noise. However, even in such cases the slight increase in filter output due to the weak wave, followed a hyperbolic trend in time on a large range of receivers, while the higher increases due to randomness of noise did not. By applying the geometric mean filter again, to the results of the first application of the filter, we can bring out all noticeable waves, no matter how weak, to have higher values than any area containing only noise. As we have achieved the desired results with the second application of the filter, a third application is unnecessary and would only increase computation time. Therefore, once all the elements of the matrix E have been calculated, the filter is used again, but this time on the matrix **E** to obtain the matrix **F**. When geometric mean filter is used to obtain F, the parameter which controls the number of column used in each geometric mean, n_F , is different than n_E which is used with the filter to obtain **E**. The optimal values of the parameters n_E and n_F are determined in a machine learning stage of the algorithm.

An example of the original data matrix **M**, and of the filter's result **F** is given in the Figure 7.

Matrix **F** is designed to produce large values at points at which a displacement (or pressure) from a coherent wave was recorded. We select the element in **F** with the highest value as the most likely element in the matrix **M** that describes a coherent wave. The row and column of the selected point designates the position of the coherent wave and are termed respectively the waveform row and waveform column, or w_r and w_c in equations and Figure 7.



Figure 6. The process of determining the set of elements $(\delta_{i,j})$ for calculating the geometric mean at the position (i, j). The curly braces indicate the set of elements from which the next element will be added to the geometric mean. (a) The first element, $M_{i,j}$ is automatically added to the set $\delta_{i,j}$ and the algorithm searches for the following elements in columns adjacent to column j. (b) The two elements from the adjacent columns are added and the search is now on columns j - 2 and j + 2. (c) Two more elements are added to $\delta_{i,j}$, from columns j - 2 and j + 2, and the search continues until $\delta_{i,j}$ is complete.



Figure 7. (a) The original data matrix **M**. (b) The filter's result **F** with the indicated waveform row and waveform column.

2.2.2. Step 2

In the first step we determine the position (w_r and w_c) of a specific point, or element, that describes a part of a coherent wave. Thus, we know the position of said wave in only one of the columns, namely the waveform column (w_c). In the second step, we find the row position of its waveform in the remaining columns of **M**, and in doing so we build the shift vector. In this context, the shift vector records the row position of the waveform in each of the columns, relative to the position of the waveform row (w_r). Therefore, the waveform position in any column *j* is equal to the sum of the waveform row w_r and the *j*th element of the shift vector *s*. The location of the waveform in the remaining columns is found using a cross-correlation procedure.

We select a sequence (ψ) from the waveform column that contains all the elements between rows $w_r - w$ and $w_r + w$ (Figure 8a,b). The parameter w represents the window size when determining the sequence (ψ), and its value is determined in the machine learning stage. The sequence of elements ψ is supposed to represent the waveform, or the greater part of it, and we refer to it as the waveform sequence. The position of the waveform in other columns is then determined by finding in each column a sequence of elements that has the highest correlation with the waveform sequence (Figure 8c).

We start from the columns adjacent to the waveform column w_c . The waveform position in columns adjacent to w_c must be between $w_r - m$ and $w_r + m$, where m is the maximum dip rounded up to the nearest integer. To reduce computation time, and to make sure to follow the same wave throughout all the columns, in adjacent columns we search for the ψ sequence from row $w_r - m - w$ to row $w_r + m + w$. Initially, the range of rows in which we search for the ψ sequence is entirely dependent on the waveform position in the closest column. For any column position $j > w_c$, we search for the ψ sequence from row $w_r + s_{j-1} - m - w$ to row $w_r + s_{j-1} + m + w$.

Once we have determined the waveform position in columns ranging from position $w_c - 2l$ to position $w_c + 2l$, we narrow the search window for the following columns. The value of the parameter l is determined in the machine learning stage. At this point, for a column $j > w_c + 2l$ we search for the waveform sequence from row $i_e - 1 - w$ to $i_e + 1 + w$.

The row position i_e in column j is determined by following the parabolic trend of waveform positions in columns j - 1, j - 1 - l, and j - 1 - 2l. Fitting parabolas to shift vectors will also be used in the Discussion section, for estimating the velocities of the recorded waves. While arrival times are usually modeled with a hyperbola, we use a parabola because it requires fewer points on the shift vector to find its coefficients. Since most parts of the hyperbola can locally be fairly accurately described with a parabola, our parabolic estimates are sufficiently accurate. Narrowing the search window allows us to estimate the waveform position in each column more quickly and to make estimating waveform position more resilient to perturbations from noise and other waves. The optimal values for parameters w and l are determined in the machine learning stage. Moreover,

11 of 27



if the maximum correlation with the waveform sequence in a certain column is below 0 the step terminates, as at that point we can be certain that the waveform is no longer present.

Figure 8. (a) Zoomed in area around position (w_r , w_c) from matrix **M**. (b) The selected waveform sequence ψ . (c) The location of ψ in the surrounding columns.

2.2.3. Step 3

Once the shift vector and all the waveform positions are determined, the shift vector and shift operator are used to shift the columns of the data matrix **M** so that the wave appears in the same set of rows in each column. At that point, the waveform is similar to the one shown in Figure 1, and it can be very accurately described as an outer product between two singular vectors. The number of rows storing the waveform pattern is much smaller than the total number of rows in **M**. Therefore, most of the elements in the column vector are not describing the waveform, hence are not relevant, and can be ignored. The length of the column vector need not be as long as columns in **M**, so it is reduced such that its first element specifies the number of zeros before the waveform and the last element specifies the number of zeros after the waveform. For example, the waveform row (w_r) would then be equal to the sum of the first element of the column vector (u_1) and half of the length of the recorded waveform (l_w). The optimal length of the waveform that is recorded in the column vector (l_w) (i.e., the number of elements in the column vector excluding the first and the last element), is determined in the machine learning stage. The row vectors and shift vectors are reduced in a similar manner to the column vector since the waveform is often contained in only a subset of columns of \mathbf{M} , i.e., the wave packet may not be recorded at a subset of receiver locations. The effect of these changes is to decrease the number of elements required to represent \mathbf{M} , thus decreasing the memory requirement of the SMD algorithm.

Once the two singular vectors (the column vector u and the row vector v) are extracted, their outer product is subtracted from the matrix **M**. The columns of **M** are also shifted back to their original locations. The new matrix **M**^{new} can be described with the equation:

$$\mathbf{M}^{\mathbf{new}} = \mathbf{M} - \chi(\mathbf{uv}^T, -\mathbf{s}).$$
(7)

As a result, the majority of the located wave disappears which allows the algorithm to focus on a potential second wave present in **M**. At the end of this step of the algorithm, a wave has been subtracted from matrix **M** and added to the compressed data as a pair of singular vectors and a corresponding shift vector.

2.2.4. Step 4

The three steps described above are repeated until a certain performance criterion is met. Herein we define that criterion in terms of memory. Once the memory usage of the compressed data reaches 20% of the original data (80% compression), the SMD algorithm terminates. Specifically, once the number of elements used to describe all the shift vectors and singular vectors is equal to 20% of the number of elements in the matrix describing the raw seismic data, the performance criterion is met and the SMD algorithm terminates.

2.3. Machine Learning

2.3.1. Training Data and Scoring the Model

In the algorithm description we have introduced five parameters that influence the results of SMD (shifted-matrix decomposition). The five parameters are n_E and n_F from Step 1, w and l from Step 2, and l_w from Step 3. However, we do not have an intuitive understanding of how the results of SMD are affected by the changes to the parameters, and we have no analytical solutions for optimal values for the parameters. For that reason, we use supervised machine learning to find a set of parameters that provide consistently good SMD results. Once the parameters are optimized in the training stage, SMD can be applied to new seismic data without the need to run the training stage again. Moreover, the parameter m, which also affects the results of SMD, is predetermined and therefore cannot be optimized.

The first step is to define what is meant by a good SMD output. We desire the matrix reconstructed from SMD to closely resemble the original seismogram matrix **M**. Specifically, with a predefined compression rate, we want to capture a big portion of the original data, and we want as little noise as possible to be recorded in the SMD results. We refer to the matrix reconstructed from decomposed data as $\mathbf{M}^{\mathbf{r}}$. When we measured the resemblance between **M** and $\mathbf{M}^{\mathbf{r}}$ as an L_1 or L_2 norm of the error $\mathbf{M} - \mathbf{M}^{\mathbf{r}}$, we obtained the smallest error norms when the SMD captured a lot of the noise with a few pairs of singular vectors and shift vectors. For this reason, rather than minimizing a norm of the error, we decided to maximize the dot product of matrices **M** and $\mathbf{M}^{\mathbf{r}}$:

$$\mathbf{M} \cdot \mathbf{M}^{\mathbf{r}} = \sum_{i,j} M_{i,j} M_{i,j}^r.$$
(8)

The dot product was more sensitive to SMD results representing a weak coherent wave or a small remaining part of a large wave than it was to SMD results representing primarily noise. We believe this is because the dot product of the two matrices is proportional to their correlation. Recording a large area of noise with a pair of singular vectors and a shift vector could decrease the error $M - M^r$ by a fair amount. However, because of the randomness of noise, the reconstructed data will still have weak correlation with the original data in the noisy area. Recording any area with coherent waves with a pair of singular vectors and a

shift vector usually provides high correlation between the original and the reconstructed data in said area.

Thus, we search for a set of parameters which maximize the dot product $\mathbf{M} \cdot \mathbf{M}^{r}$. For training data, we use recordings of reflected wave arrivals from 5 different shotgathers from marine seismic gathers. We use seismic field data obtained from marine seismic gathers because the data contain a large number of coherent waves with lots of interference, and noise with high coherence which can be difficult to differentiate from signal. To reduce the time spent in the training stage, from each shot-gather we only use 2 s recordings containing reflected waves and the preceding noise recorded by the first 200 receivers. More details on the field data will be provided in the following section. The five shot-gathers are labeled as $\mathbf{M}^{1} - \mathbf{M}^{5}$ and data reconstructed from decomposition of the five seismograms as $\mathbf{M}^{r,1} - \mathbf{M}^{r,5}$. Thus, the overall quality of SMD with a given set of parameters would be quantified as the SMD score (ζ):

$$\zeta = \sum_{k=1}^{5} \mathbf{M}^{\mathbf{k}} \cdot \mathbf{M}^{\mathbf{r},\mathbf{k}}.$$
(9)

The SMD score is only used in training stage to find an optimal set of parameters, and it will not be present in the following sections.

2.3.2. Derivative Free Optimization

There is no analytical formula directly relating the SMD score (Equation (9)) to the five input parameters. Therefore, we cannot use a gradient analysis to find optimal values for these parameters. Instead we use a derivative-free numerical optimization method. Specifically, we will use a pattern search. The five parameters to be optimized can only be natural numbers. Thus, the five parameters can be described by an element in \mathbb{N}^5 . In this optimization, we take a subset Ω of \mathbb{N}^5 , to be the set of all the reasonable values of the five parameters. Specifically, we seek an element in Ω for which SMD score has the highest value. This is done by performing a pattern search which takes the current position in Ω and checks adjacent points to see if any of them yield a higher SMD score. An adjacent position is defined as a position that can be reached by changing only one of the five parameters by the minimum amount. If the adjacent element with the highest SMD score has a higher score than that of the current position, the optimizing position is moved to the adjacent element. This is done until a local maximum for the SMD score is found. To increase the likelihood of finding the global maximum, the pattern search is repeated multiple times with a random starting location.

2.3.3. Application

Once the training stage is finished, SMD can be applied to a new seismic data set even if it is not from a marine gather. Without going through the training stage again, SMD was successfully applied to new marine gathers, synthetic data from [17], and to field data collected by linearly distributed geophones. However, the discussions in the following sections are focused on data from marine seismic gathers, rather than data gathered by geophones. Because marine seismic gathers were obtained from a much larger number of receivers and contain greater volumes of data, they provide a more realistic example of applications of SMD.

Even when skipping the training stage, the parameters still need to be adjusted before SMD is applied to new seismic data, based on the dominant frequency and the maximum slope of the waves in the new dataset. Specifically, parameters w from Step 2 and l_w from Step 3 need to scale proportionally to the period of the dominant frequency multiplied by the sampling rate. Moreover, parameters n_E and n_F from Step 1 and l from Step 2 should scale proportionally to the period of the dominant frequency multiplied by the sampling rate and divided by the maximum slope of arriving waves. Therefore, when applying SMD to a seismic data set, one should also include information about the dominant frequency

in the data set, the sampling rate of the receivers, and the maximum slope of the arriving waves. Knowing these values allows the user to apply SMD to new seismic data sets without going through the training stage. The flowchart in Figure 9 provides a visual description of how SMD is applied to a set of files containing seismic data.



Figure 9. Flowchart describing the process of applying SMD to seismic data. The fourth step from above is described by the flowchart presented in Figure 5.

3. Results

Once the SMD parameters have been optimized in the machine learning stage, its performance is tested on three datasets. The first one is a synthetic dataset similar to one from [17]. Structure-oriented SVD [17] was tested on several datasets and while it was successful at reducing noise in all examples, it produced numerical artifacts in one of them. In this work we reproduce that challenging dataset in order to demonstrate the improvements achieved with SMD. The other synthetic data test cases from [17] contain fewer coherent waves and don't show interference between waves of different slopes and are not as difficult for reliable compression and analysis. For that reason, the second and third data sets on which SMD is tested are from field data obtained during marine seismic gathers. The second dataset has many, interfering, strong arrivals which can together create a complicated signal. The third dataset has fewer arrivals, but the coherent waves are much weaker and difficult to differentiate from noise.

3.1. Synthetic Data

Figure 10 shows application of local SVD and SOSVD on the synthetic dataset from [17], while Figure 11 shows the application of SMD to a very similar data set. To create the synthetic data in Figure 11 we recreated the signal observed in Figure 10 and added random noise to it.



Figure 10. (a) Synthetic data showing several wave arrivals without noise. (b) The noisy data, created by adding noise to the synthetic data. (c) The noisy data filtered using local SVD. (d) The noisy data filtered with SOSVD. The blue rectangles highlight areas of interest in which local SVD and SOSVD produce numerical artifacts. This figure was modified from [17].

In Figure 10 we see that both local SVD and SOSVD are able to remove most of the noise. However, both local SVD and SOSVD struggle to properly reconstruct the signal in the area highlighted by the blue rectangle in Figure 10c,b. The highlighted area contains two waves of different slopes intersecting with each other.

Figure 11 shows results of applying SMD to the synthetic data set twice, with 95% compression (Figure 11c) and 80% compression (Figure 11d). In both SMD applications the coherent waves were reconstructed perfectly. We can see that there are no numerical artifacts in the area highlighted by the blue rectangle in Figure 11c,b, which contains intersecting waves with different slopes.

While we can see significant noise reduction, we could not find specific information in [17] describing the amount of noise present in synthetic data in Figure 10, However, we can estimate the signal-to-noise ratio in the synthetic data on which SMD is tested. By measuring the root mean square value from all receivers in time increment range from 300 to 320, where only noise is present, we can estimate the amplitude of the noise. To estimate the amplitude of the signal, we measure the root mean square value from all receivers in time increment range from 340 to 360, from the data presented in Figure 11a, which contains only pure signal. The signal-to-noise (ρ) ratio is therefore calculated with the Equation (10):

$$\rho = \frac{\sqrt{\sum_{i=340}^{i=360} \sum_{j} M_{i,j}^{p^{-2}}}}{\sqrt{\sum_{i=300}^{i=320} \sum_{j} M_{i,j}^{n^{-2}}}}$$
(10)

where $\mathbf{M}^{\mathbf{p}}$ represents the data containing only pure signal presented in Figure 11a, and $\mathbf{M}^{\mathbf{n}}$ represents the data from Figure 11b,c, or d, which contains some amount of noise.



Figure 11. (a) Synthetic data showing several wave arrivals without noise. (b) The noisy data, created by adding noise to the synthetic data. (c) Data obtained by applying SMD to the noisy data with 95% compression. (d) Data obtained by applying SMD to the noisy data with 80% compression. The blue rectangles highlight areas of interest in which local SVD and SOSVD produce numerical artifacts.

The signal-to-noise ratios in the original data, data reconstructed from SMD results after 80% compression, and after 95% compression were 1.9, 4.7, and 12.3, respectively. Notice that the signal-to-noise ratio is much larger when SMD is applied with 95% compression than when it is applied with 80% compression. This is because there is only a few coherent waves present in the data. These coherent waves are represented with first several singular vectors and shift vectors, while the following singular vectors and shift vectors are describing noise. However, SMD results represent noise a lot less efficiently than coherent waves, and the majority of the noise is still not recorded when SMD is applied with 80% compression. In conclusion, the SMD application with 80% reduction in memory requirements picked up some of the noise, while SMD application with 95% data compression ignored the noise almost completely.

3.2. Field Data

Finally, the SMD algorithm is tested on field data, collected between January and March 2016 during the CREST expedition, MGL1601, aboard the R/V Marcus G. Langseth. Pressure waves are generated by a tuned array of 36 air guns, towed at a depth of 6 m. The resulting acoustic waves are recorded using a 12,587.5 m hydrophone streamer, towed at the depth of 8 to 12 m, and carrying 1008 receivers. The receivers are spaced by 12.5 m, each of them recording pressure once every 4 ms. The maximum offset between two adjacent columns is a little greater than two rows, so we round it up to three rows. Further information regarding data acquisition can be found in [21]. Seismic data are also available at the NSF-sponsored Academic Seismic Portal hosted by the University of Texas Institute for Geophysics and can be accessed at https://www.marine-geo.org/tools/search/Files. php?data_set_uid=23597 (retrieved on 31 March 2021).

Unfortunately, when being applied to real seismic data such as in Figures 12 and 13 SMD with 95% compression cannot reconstruct the arriving waves properly. Due to multiple reflections and dispersion, the number of coherent waves is much larger in marine seismic gathers than in synthetic data. Furthermore, the noise has high coherence such that it can closely resemble wave arrivals. Using SMD with 80% compression ensures that the arriving waves will be properly reconstructed while also ensuring the noise is drastically reduced.



Figure 12. (a) The original data from ocean seismic gathers with strong reflections. (b) The data reconstructed after applying 80% compression with SMD to the original data. Areas highlighted in red are enlarged 3 times along *x*-axis and 6 times along *y*-axis.



Figure 13. (a) The original data from ocean seismic gathers with weak reflections. (b) The data reconstructed after applying 80% compression with SMD to the original data. Areas highlighted in red are enlarged (2 times along the x- and y-axes). The reflected waves are circled with purple lines, direct wave with green lines, and the preceding noise with black lines.

The first data set (Figure 12) contains strong reflections arriving far after the direct wave which can be seen at early times on the few receivers close to the source. In this test, SMD was able to identify prominent waveform-related features while almost completely ignoring noise-dominated sections of data. The Figure 12 highlights in red an enlarged portion of the data (3 times along *x*-axis and 6 times along *y*-axis) in which we can see the first arriving reflections. In the original data (Figure 12a), at the lower half of the enlarged window we can also see the preceding noise. However, once the data is processed with SMD (Figure 12b) the preceding noise is no longer present. A principal drawback of SMD is that in some scenarios it will also ignore weaker incoming waves, and fail to differentiate them from noise.

The second data set (Figure 13) is from a different shot-gather in which we can observe weak reflections arriving closely after the much stronger direct wave. In Figure 13 the reflections are circled with purple lines, direct wave with green lines, and the preceding

noise with black lines. Red lines are used to highlight and enlarge (2 times along the *x*and *y*-axes) an example of noise interfering with the reflected waves. The reflections are hard to differentiate from the more noticeable direct waves and because of their small amplitude, they are noticeably distorted by noise. In this test, SMD identified and properly reconstructed both direct waves and reflected waves, while reducing noise in all parts of the data. The preceding noise (circled in black) is noticeably weaker in the data reconstructed from SMD results (Figure 13b) than in the original data (Figure 13a). Furthermore, the enlarged window (circled in red) in the original data (Figure 13a) shows reflected waves distorted by noise. However, in the data reconstructed from SMD results (Figure 13b), the enlarged window (circled in red) shows a more clear, denoised version of the reflected waves. Because SMD reduces the noise everywhere in the data, the weak reflected waves can be seen more clearly in the data reconstructed from SMD results (Figure 13b).

The examples in Figures 12 and 13 show subsets of the entire data files which are too big to be presented in a single figure. However, the large data size provides a good opportunity for testing the efficiency of SMD. A single data file, which contains 12 s of recorded data, has 1008 traces, each trace containing 3000 pressure samples. Processing this amount of data on a laptop with an i5-6200U CPU and 8 GB of RAM, without parallelization, takes 4.2 s on average. Therefore, SMD is sufficiently fast when processing marine seismic gathers in real time. However, distributed acoustic sensing produces produces greater volumes of data than a long line of receivers during marine gathers.

We propose two solutions as we prepare SMD to future application on seismic data obtained by DAS. First, the current SMD algorithm is optimized to provide a best representation of the coherent waves, for a predefined memory requirement. To make SMD more applicable to new developments in the data acquisition (DAS), we could run a new training stage that considers computation time as well as the quality of the results. This could create a new version of SMD that is more applicable to data obtained by DAS. Second is the introduction of signal detection which we will discuss in the following section. The large data file that requires 4.2 s of computation time is heavily populated with signal. This is not the case during microseismic monitoring during which most of the files contain only noise. Running an initial test to check for the presence of coherent waves, prior to fully processing the data with SMD, may significantly reduce the time spent on processing data during seismic monitoring.

4. Discussion

In addition to seismic data compression and noise reduction, SMD also provides a new method of seismic data analysis. Rather than a list of displacements distributed in space and time to describe incoming waves, we have pairs of singular vectors paired with shift vectors. In an ideal scenario, for each arriving wave the column vector represents the waveform, the row vector represents the amplitudes at different receiver locations and the shift vector represents relative arrival times. Even though the ideal case is rarely achieved, we believe that the results of SMD provide an excellent advance in the realm of seismic analysis especially with its application of machine learning. With SMD, identifying features that can be used for building models is facilitated when noise-reduced data is represented in the SMD-compressed format.

The application of machine learning to data compression was explored in [24], which applied SVD to synthetic data and developed a model for estimating source location and orientation. However, SMD may provide greater opportunities for machine learning application.

It is instructive to provide an example of how the SMD algorithm can help estimate physical properties such as elastic wave velocities. To this end, herein we predict the average acoustic velocity (α) in our model (seawater) by analyzing the results of the SMD algorithm. Specifically, we use curvature of the shift vectors, and the zero offset time to estimate the average velocity of the reflected waves. Since we have a controlled source, a wave's zero offset time is simply it's row position in the first column of the data matrix multiplied by the time difference between consecutive recordings (t_d). Since the row

position of a wave can be determined from the shift vector and the first element of the column vector, our wave velocity estimation is obtained solely from the data stored in SMD results.

To derive the formula for wave velocity we must make several assumptions about our surroundings. First, we assume that the reflecting surface (the ocean floor) is horizontal, and that the depth of the ocean floor (z_f) is significantly larger than the horizontal distance from source to the receiver ($x \ll z_f$). This gives us the expression for the total distance travel by the reflected waves (d):

$$d \approx 2z_f + \frac{x^2}{4z_f}.\tag{11}$$

Since we are considering the average acoustic velocity (α) in our model, we can rewrite the expression (11) in terms of the reflected wave travel-time (*T*):

$$T \approx \frac{2z_f}{\alpha} + \frac{x^2}{4z_f \alpha}.$$
 (12)

The zero offset time T_0 is defined as travel-time T at zero horizontal distance (x = 0):

$$T_0 \approx \frac{2z_f}{\alpha}.\tag{13}$$

Taking the second derivative of the expression (12) with respect to horizontal position x gives:

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{1}{2z_f \alpha}.$$
 (14)

Using expression (13) to substitute $(2z_f)$ with $(T_0\alpha)$ in expression (14) gives:

$$\alpha \approx \left(\frac{\partial^2 T}{\partial x^2} T_0\right)^{\frac{-1}{2}} \tag{15}$$

Expression (15) can be used to calculate the acoustic velocity if we can express both $\frac{\partial^2 T}{\partial x^2}$ and T_0 in terms of SMD results.

In Section 2.2.2 it is explained that the row position (p_j) of a wave in any column j is equal to the sum of the j^{th} element of the shift vector (**s**) and the waveform row (w_r) :

$$p_j = s_j + w_r, \tag{16}$$

In Section 2.2.3 it is explained that the waveform row (w_r) can be obtained from the first element of the column vector (\mathbf{u}) and the predetermined length of the recorded waveform (l_w) :

$$w_r = u_1 + \frac{l_w - 1}{2}.$$
 (17)

Using expressions (16) and (17), the zero offset time (T_0) can be described with the row position of the wave in the first column (p_1) multiplied by the time difference between consecutive recordings (t_d):

$$T_0 \approx t_d(s_1 + u_1 + \frac{l_w - 1}{2}).$$
 (18)

The expression (18) will be used to obtain the zero offset time from SMD results.

If we assume that the relative arrival times were accurately recorded by the shift vector, we can make the following substitution:

$$\frac{\partial^2 T}{\partial x^2} = \frac{s'' t_d}{x_d^2} \tag{19}$$

where x_d is the distance between adjacent receivers. In the expression (19), s'' is the second derivative of an element in the shift vector s with respect to the position (j) of the element in the shift vector.

The value of s'' is determined by fitting a parabola to first n_r elements of the shift vector. The parameter n_r need not have an exact value. While we want to use enough elements from the shift vector to confidently fit a parabola, we also want to only use data from the receivers close to the source in order to follow the $(x \ll z_f)$ condition. Therefore, the parameter n_r is set to 50. Once we fit the parabola, the value of s'' is estimated to be the quadratic coefficient c_2 , multiplied by 2:

$$j \in [1, n_r] : s_j \approx c_0 + c_1 j + c_2 j^2,$$

$$s'' = 2c_2.$$
(20)

By plugging the expressions (18)–(20) into expression (15) we estimate the acoustic velocity. However, the values of a shift vector can be affected by interfering waves, or noise. To minimize the error from those sources, we estimate the velocity based on the first n_s pairs of shift vectors and column vectors. Similar to n_r , the n_s parameter does not need to be set to any specific value. In this example, the parameter n_s is set to 5. The average acoustic velocity α is estimated as the weighted average of the results from the first n_s extracted pairs of a of shift vector and a column vector. Each term is weighted by the quality of the parabolic fit, which is equal to the inverse of the error norm *e*:

$$s^{k} \rightarrow \{c_{0}^{k}, c_{1}^{k}, c_{2}^{k}\}$$

$$e^{k} = \sum_{j=1}^{n_{r}} \left(s_{j}^{k} - (c_{0}^{k} + c_{1}^{k}j + c_{2}^{k}j^{2})\right)^{2},$$

$$e_{sum} = \sum_{k=1}^{n_{s}} \frac{1}{e^{k}},$$

$$\alpha = \sum_{k=1}^{n_{s}} \left((2c_{2}^{k})(s_{1} + u_{1}^{k} + \frac{l_{w}}{2})\frac{t_{d}^{2}}{x_{d}^{2}}\right)^{\frac{-1}{2}} \left(\frac{1}{e^{k}e_{sum}}\right).$$
(21)

The formula for α , defined in expression (21), was applied to 20 marine field data files, each recording a seismic response from a unique and controlled seismic source. For each file, we used the formula from expression (21) to estimate the velocity. The average estimated velocity was $1601\frac{m}{s}$ and the standard deviation among the results was $236\frac{m}{s}$. Considering the ocean depth being about 3 km, the correct average acoustic velocity was likely about $1500\frac{m}{s}$. If we assume the correct average wave velocity experienced by the reflected waves was $1500\frac{m}{s}$, then the average and the median error from the 20 velocity estimates were 12.1% and 8.3%, respectively.

The experiment in this subsection proves that there is a strong correlation between the shift vector and relative arrival times of the wave at different receivers. We were able to use that correlation to estimate the velocity of the waves without relying on any information about the medium through which the waves were traveling. We believe that there is also a strong correlation between the column vector and the waveform, as well as between the row vector and the relative amplitudes of the waveform at different receivers. However, proving these correlations will require further testing.

In this example, SMD was applied to seismic data from a controlled source. In such cases we can always be certain that there are coherent waves present. However, we are planning a wide range of applications for SMD. During microseismic monitoring, often recorded by DAS, we might want to analyze coherent waves that are not coming from controlled sources. Therefore, when applying SMD to data from microseismic monitoring we do not know in advance whether the data contains signals of interest. For that reason, we developed a method for recognising coherent signal in noisy data, that also relies solely on the results from SMD.

We created a method that can differentiate between noisy data and data containing signal, by measuring the curvature in the first several shift vectors. Applying SMD to noisy data without coherent waves returns a set of completely unrelated shift vectors. On the other hand, applying SMD to data containing coherent waves, usually coming from the same source, returns a set of shift vectors most of which have similar curvatures. Therefore, the method differentiates between noisy data and data containing coherent waves, by calculating the standard deviation among the curvatures from the first 10 extracted shift vectors. To test this method, we applied SMD separately to two subsets of data from each of the 20 previously mentioned data files. The first subset contains only the recordings of noise, and the second subset contains recordings of both the coherent waves and the preceding noise (Figure 14).



Figure 14. Two subsets from a marine field data file. The first subset contains only noise, and the second subset contains both a part of the coherent waves and some of the preceding noise.

For each subset, we use expression (20) to estimate the curvatures of the first 10 extracted shift vectors and then we compute the standard deviation of the 10 curvature values. The results are presented in Figure 15, in which the standard deviation for each subset without coherent waves is presented in blue, and the standard deviation for each subset containing coherent waves is presented in red. Due to shift vector curvature having random values when SMD results represent noise, the standard deviation among shift vector curvatures should be greater for the data subsets which contain only noise. As can be seen in Figure 15, the standard deviation among shift vectors is consistently lower for subsets containing coherent waves, which are presented in red. This confirms that we can differentiate between data containing only noise and data containing signal, by using our described method.

It is important to emphasize that this signal detection method does not contain any information regarding the amplitude of the waves. It is entirely dependent on the curvature of the recorded shift vectors, which is a unique property. This means that this method, while accurate, can also be used to complement other signal detection methods, all of which rely on other properties in seismic data (such as STA/LTA for example, which relies on the amplitudes of the waves). Furthermore, this signal detection method only requires the first 10 shift vectors, which can be acquired very quickly compared to the to the processing time of the SMD. Therefore, when applying SMD to data obtained during seismic monitoring, we suggest first running a quick version of SMD that only extracts the first 10 shift vectors

in order to run signal detection. If the presence of coherent waves is confirmed, SMD may continue to fully process the data.



Figure 15. Standard deviation of shift vector curvature from each of the 20 data files. The results from subsets containing only noise are presented in blue and the results from subsets containing parts of coherent waves are presented in red.

The analysis of SMD results that was conducted in this subsection was not based on machine learning but instead on our understanding of the correlations between compressed data and attributes of the recorded waves. In the future, those strong correlations will be used with machine learning to train algorithms to accurately infer various properties of the source and the surrounding velocity model directly from compressed data. Additionally, application of data from marine surveys is difficult because of large amounts of interference between arriving waves. Thus, the SMD method may be more effective in different circumstances, such as in unconventional reservoirs monitored by distributed acoustic sensing.

Thus far, we only considered application of SMD to two-dimensional seismic data, obtained by a single line of receivers. However, receivers may often be distributed over an area, in a large number of lines. In such case, SMD may still be applied individually, to the data collected from each line. However, there may be a lot of redundancy between SMD results from each of the lines of receivers. Similar redundancies can also occur by applying SMD to multiple files from different times, obtained by a same set of receivers. In future work, we will take advantage of the fact that SMD results may be similar for batches of data collected from different receivers or during different times. Dictionary learning could be applied to SMD results to further compress the data, although we can not yet recommend a method for learning the dictionary. Future work will therefore likely include further compression by applying dictionary learning to SMD results from multiple data files or multiple receiver lines.

5. Conclusions

Shifted-matrix decomposition is a powerful tool that can simultaneously compress and improve seismic data. This is done by converting the seismic data from a matrix into a set of pairs of singular vectors coupled with shift vectors which require less memory to store the seismic data. Furthermore, reconstructing the seismic data from compression results, creates a denoised version of the original data. Shifted-matrix decomposition

24 of 27

provides an improvement to some of the existing denoising techniques such as SOSVD by avoiding numerical artifacts in areas with multiple intersecting waves of different slopes. This allows us to apply SMD to complicated field data with a large number of arriving waves and still achieve 80% compression.

When applied to synthetic data and ocean gathers it was able to boost coherent signal and erase the majority of the noise. In synthetic data with signal-to-noise ratio of 1.9, it was able to increase the ratio to 4.7 in the case of 80% compression and to 12.3 in the case of 95% compression. However, the excellent result achieved with 95% compression in the synthetic data is an artifact of the lack of coherent noise and the small number of coherent waves. Due to noise coherence and more complicated overlap of coherent waves, we recommend applying 80% compression to examples of field data in order to accurately represent the signal of interest. In field data, SMD was able to reduce the noise in areas preceding the signal as well as in areas containing coherent waves. As a results, weak waves that were difficult to notice in the original data, can be seen more clearly in the data reconstructed from SMD results. The only drawbacks are that in some scenarios SMD may fail to boost weaker signals and it is not meant to be applied to seismic data obtained from a small number of receivers.

There is a good correlation between the physical properties such as elastic wave velocity and the results of the SMD. As an example, the average wave velocity in the medium through which the waves propagate (seawater) was roughly estimated by analyzing the shift vector curvature. In the future, we will use machine learning to build models that infer with high accuracy the properties of the source and the velocity model, directly from the SMD results. Results of SMD can also be used for other analysis, such as signal detection. By analyzing only the first several shift vectors, we can check for the presence of coherent waves. Because it requires such a small amount of data, this technique can be executed very quickly by taking only the initial results of SMD. We recommend using it during microseismic monitoring before fully processing the data with SMD.

Author Contributions: M.B.: conceptualization, methodology, software; E.G., R.L.G., M.E.E.: review and guidance. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Crisman/Berg-Hughes, project number HF.12.19B.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All code used in this paper can be obtained by contacting the author at mbrankov@tamu.edu or milanovstalni@gmail.com.

Acknowledgments: We would like to thank Robert Reece and his student Jingxuan Wei for providing us with the field data which played an integral role in the development and testing of the SMD.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DAS	Distributed acoustic sensing
SMD	Shifted-matrix decomposition
SVD	Singular value decomposition
SOSVD	Structure-oriented singular value decomposition
SNR	Signal-to-noise ratio
STD	standard deviation
PCA	Principal component analysis
STA/LTA	short-term average over long-term average

Appendix A. The Quantative Example

Consider a small matrix $\mathbf{M^s} \in \mathbb{R}^{8,8}$ representing a simple wave arrival recorded by 8 receivers.

M ^s =		0 2	0 3	0 2	0 0	0 0	0 0	0 - 0
	1 _1	$-2 \\ 0$	$-3 \\ 0$	$-2 \\ 0$	1 _1	0 1	0	0
	0	0	0	0	0	-1	1	0
	0	0	0	0	0	0	-1	1
	0	0	0	0	0	0	0	$^{-1}$
	0	0	0	0	0	0	0	0

In the matrix \mathbf{M}^{s} , each receiver is represented by one of the columns, and the time increases from top row to the bottom row. The source is closest to the third receiver, and the perceived waveform is simply:

$$\mathbf{w^s} = \begin{bmatrix} 0\\1\\-1\\0\end{bmatrix}$$

.

Figure A1 loosely describes a toy model that can generate the matrix M^s.



Figure A1. A toy model corresponding to the data recorded in the matrix M^s.

By applying SMD to matrix $\mathbf{M}^{\mathbf{s}}$ we extract a pair of the singular vectors ($\mathbf{u}^{\mathbf{SMD}}, \mathbf{v}^{\mathbf{SMD}}$), and a shift vector (**s**). Keep in mind that when using SMD, the singular value λ^{SMD} is stored in right singular vector $\mathbf{v}^{\mathbf{SMD}}$ by multiplying the vector with it:

$$\mathbf{u}^{\mathbf{SMD}} = \begin{bmatrix} 0\\ 1/\sqrt{2}\\ -1/\sqrt{2}\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0 \end{bmatrix}, \quad \mathbf{v}^{\mathbf{SMD}} = \begin{bmatrix} \sqrt{2}\\ 2\sqrt{2}\\ 3\sqrt{2}\\ 2\sqrt{2}\\ \sqrt{2}\\ \sqrt$$

Using the singular vectors and the shift vector we can reconstruct a matrix M_{SMD} to be identical to M_s :

$$\mathbf{M}^{\mathbf{SMD}} = \chi(\mathbf{u}^{\mathbf{SMD}}\mathbf{v}^{\mathbf{SMD}^{T}}, \mathbf{s})$$
$$\mathbf{M}^{\mathbf{SMD}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 2 & 0 & 0 & 0 & 0 \\ 1 & -2 & -3 & -2 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We can also apply regular SVD to the matrix M^s to extract a singular value λ and singular vectors **u** and **v**:

$$\lambda \approx 5.92, \quad \mathbf{u} \approx \begin{bmatrix} 0\\ 0.68\\ -0.73\\ 0.05\\ 0\\ 0\\ 0\\ 0 \end{bmatrix}, \quad \mathbf{v} \approx \begin{bmatrix} -0.13\\ 0.48\\ 0.72\\ 0.48\\ -0.13\\ 0.01\\ 0\\ 0 \end{bmatrix},$$

We can reconstruct a matrix M^{SVD} from the extracted eigenpair with the following equation: $\mathbf{M}^{SVD} = \lambda(\mathbf{u}\mathbf{v}^T)$

				(
$M^{SVD}\approx$	ГО	0	0	0	0	0	0	[0
	-0.53	1.93	2.90	1.93	-0.53	0.03	0	0
	0.56	-2.06	-3.08	-2.06	0.56	-0.03	0	0
	-0.04	0.13	0.19	0.13	-0.04	0	0	0
	0	0	-0.01	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	LO	0	0	0	0	0	0	0

There is a significant difference between M^s and M^{SVD} . The data from matrix M^s cannot be described by a single pair of singular vectors and a singular value obtained by applying regular SVD. However, it can be described by a pair of singular vectors coupled with a shift vector obtained by applying SMD.

References

- 1. Wessels, S.A.; De La Peña, A.; Kratz, M.; Williams-Stroud, S.; Jbeili, T. Identifying faults and fractures in unconventional reservoirs through microseismic monitoring. *First Break* **2011**, *29*.
- Li, X.; Zhang, J.; Grubert, M.; Laing, C.; Chavarria, A.; Cole, S.; Oukaci, Y. Distributed Acoustic and Temperature Sensing Applications for Hydraulic Fracture Diagnostics. In Proceedings of the SPE Hydraulic Fracturing Technology Conference and Exhibition, The Woodlands, TX, USA, 4–6 February 2020.
- 3. Spanias, A.S.; Jonsson, S.B.; Stearns, S.D. Transform methods for seismic data compression. *IEEE Trans. Geosci. Remote Sens.* **1991**, 29, 407–416. [CrossRef]
- Villasenor, J.D.; Ergas, R.; Donoho, P. Seismic data compression using high-dimensional wavelet transforms. In Proceedings of the Data Compression Conference-DCC'96, Snowbird, UT, USA, 31 March–3 April 1996; IEEE: Piscataway, NJ, USA, 1996; pp. 396–405.
- Khene, M.; Abdul-Jauwad, S. Adaptive seismic compression by wavelet shrinkage. In Proceedings of the Tenth IEEE Workshop on Statistical Signal and Array Processing (Cat. No. 00TH8496), Pocono Manor, PA, USA, 16 August 2000; IEEE: Piscataway, NJ, USA, 2000; pp. 544–548.
- 6. Ma, J.; Plonka, G.; Chauris, H. A new sparse representation of seismic data using adaptive easy-path wavelet transform. *IEEE Geosci. Remote Sens. Lett.* **2010**, *7*, 540–544. [CrossRef]
- 7. Wang, C.; Wang, Y. Robust Svd Filtering For Low Signal-To-Noise Ratio Seismic Data. Geophysics 2021, 86, 1–51. [CrossRef]
- 8. Kreimer, N.; Sacchi, M.D. A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation. *Geophysics* 2012, 77, V113–V122. [CrossRef]
- 9. Cavalcante, Q.; Porsani, M.J. Prestack seismic data reconstruction and denoising by orientation-dependent tensor decomposition. *Geophysics* **2021**, *86*, V107–V117. [CrossRef]
- 10. Beckouche, S.; Ma, J. Simultaneous dictionary learning and denoising for seismic data. Geophysics 2014, 79, A27–A31. [CrossRef]
- 11. Zhu, L.; Liu, E.; McClellan, J.H. Seismic data denoising through multiscale and sparsity-promoting dictionary learning. *Geophysics* **2015**, *80*, WD45–WD57. [CrossRef]
- 12. Siahsar, M.A.N.; Gholtashi, S.; Abolghasemi, V.; Chen, Y. Simultaneous denoising and interpolation of 2D seismic data using data-driven non-negative dictionary learning. *Signal Process.* **2017**, *141*, 309–321. [CrossRef]
- 13. Payani, A.; Abdi, A.; Tian, X.; Fekri, F.; Mohandes, M. Advances in seismic data compression via learning from data: Compression for seismic data acquisition. *IEEE Signal Process. Mag.* **2018**, *35*, 51–61. [CrossRef]
- 14. Tian, X. Multiscale sparse dictionary learning with rate constraint for seismic data compression. *IEEE Access* **2019**, *7*, 86651–86663. [CrossRef]
- 15. Chen, Y. Fast dictionary learning for noise attenuation of multidimensional seismic data. *Geophys. J. Int.* **2017**, 209, 21–31. [CrossRef]
- 16. Bekara, M.; Van der Baan, M. Local singular value decomposition for signal enhancement of seismic data. *Geophysics* 2007, 72, V59–V65. [CrossRef]
- 17. Gan, S.; Chen, Y.; Zu, S.; Qu, S.; Zhong, W. Structure-oriented singular value decomposition for random noise attenuation of seismic data. *J. Geophys. Eng.* 2015, 12, 262–272. [CrossRef]
- 18. Fomel, S. Applications of plane-wave destruction filters. Geophysics 2002, 67, 1946–1960. [CrossRef]
- Krawczyk, C.; Jousset, P.; Currenti, G.; Weber, M.; Napoli, R.; Reinsch, T.; Riccobene, G.; Zuccarello, L.; Chalari, A.; Clarke, A. Monitoring volcanic and seismic activity with multiple fibre-optic Distributed Acoustic Sensing units at Etna volcano. In Proceedings of the EGU General Assembly Conference Abstracts, Vienna, Austria, 3–8 May 2020; p. 15252.
- 20. Zhan, Z. Distributed acoustic sensing turns fiber-optic cables into sensitive seismic antennas. *Seismol. Res. Lett.* **2020**, *91*, 1–15. [CrossRef]
- 21. Estep, J.; Reece, R.; Kardell, D.A.; Christeson, G.L.; Carlson, R.L. Seismic layer 2A: Evolution and thickness from 0-to 70-Ma crust in the slow-intermediate spreading South Atlantic. *J. Geophys. Res. Solid Earth* **2019**, *124*, 7633–7651. [CrossRef]
- 22. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. Chemom. Intell. Lab. Syst. 1987, 2, 37-52. [CrossRef]
- 23. Golub, G.H.; Van Loan, C.F. Matrix Computations; JHU Press: Baltimore, MD, USA, 2013; Volume 3.
- Yatsenko, M.; Brankovic, M.; Gildin, E.; Gibson, R.L. A Novel Approach to Discovery of Hidden Structures in Microseismic Data Using Machine Learning Techniques. In Proceedings of the SPE Europec featured at 81st EAGE Conference and Exhibition, London, UK, 3–6 June 2019.