GEOPHYSICS, VOL. 82, NO. 6 (NOVEMBER-DECEMBER 2017); P. O91–O104, 14 FIGS., 4 TABLES.
10.1190/GEO2016-0373.1

Check for updates

Downloaded 07/19/18 to 131.111.184.3. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

Bayesian feature learning for seismic compressive sensing and denoising

Georgios Pilikos[1] and A. C. Faul[1]

ABSTRACT

Extracting the maximum possible information from the available measurements is a challenging task but is required when sensing seismic signals in inaccessible locations. Compressive sensing (CS) is a framework that allows reconstruction of sparse signals from fewer measurements than conventional sampling rates. In seismic CS, the use of sparse transforms has some success; however, defining fixed basis functions is not trivial given the plethora of possibilities. Furthermore, the assumption that every instance of a seismic signal is sparse in any acquisition domain under the same transformation is limiting. We use beta process factor analysis (BPFA) to learn sparse transforms for seismic signals in the time slice and shot record domains from available data, and we use them as dictionaries for CS and denoising. Algorithms that use predefined basis functions are compared against BPFA, with BPFA obtaining state-of-the-art reconstructions, illustrating the importance of decomposing seismic signals into learned features.

INTRODUCTION

Seismic surveying is an indispensable tool for the geophysics community. It is the process by which we are able to visualize the interior structure of our planet and guide our understanding of its physical properties. An artificial source of body waves is used at the surface, which creates reflections from deep impedance changes at rock layer boundaries, which are recorded by grids of receivers. In land and marine seismic surveys, we frequently have traces or groups of traces missing either because receivers malfunctioned, they could not be placed in some locations, or because it was impossible to collect all the planned shots. It could also be the fact that some local source of noise renders a receiver's output as unusable. There are many reasons why patches of missing data could occur with a great interest in their reconstruction using temporal, spectral, or spatial information (Shen et al., 2015).

Compressive sensing (CS) is a framework that reconstructs the signal of interest using spatial information and allows perfect reconstruction of a particular class of signals using a lower sampling rate than the Nyquist rate (Nyquist, 2002). These signals are either sparse in the acquisition domain, or they are sparse in other domains defined by dictionaries of basis functions. Seismic interpolation is treated as an inverse problem where seismic events are assumed to be sparse in some transform such as the Fourier (Sacchi et al., 1998), the Radon (Trad et al., 2002), the curvelet (Herrmann and Hennenfent, 2008), or the focal transform (Kutscha and Verschuur, 2016). These basis functions are used in conjunction with a sparse solver to obtain a solution given the data.

Projection onto convex sets (POCS) (Abma and Kabir, 2006) transforms the available data to the Fourier domain and uses hard or soft thresholding (Stanton et al., 2015). Iteratively reweighted least-squares were also proposed (Zwartjes and Sacchi, 2007) to suppress the artifacts in the Fourier domain. The iterative soft thresholding (IST) and the curvelet transform were used successfully for seismic interpolation (Herrmann and Hennenfent, 2008). A faster version of IST was proposed (Beck and Teboulle, 2009), namely, the fast iterative soft thresholding algorithm (FISTA) and then applied to seismic data (Pérez et al., 2013). The relevance vector machine (RVM) has also shown some success (Pilikos and Faul, 2016) using the discrete cosine transform (DCT) in the time slice domain, with the additional benefit of providing a probabilistic interpretation of the estimated values with its uncertainty measure. Spectral projected gradient for L1 (SPGL1) (van den Berg and Friedlander, 2009) was proposed to solve the l_1-norm minimization problem. A faster gradient projection method based on the curvelet transform was proposed (Cao et al., 2015) with comparative reconstruction accuracy but faster computational time. Tensor completion (Kreimer and Sacchi, 2011) algorithms were also proposed to solve this issue and to scale to larger dimensions. In a recent comparison of 5D solvers, POCS was found to preserve the amplitudes better (Stanton et al., 2012).

Manuscript received by the Editor 15 July 2016; revised manuscript received 27 January 2017; published ahead of production 15 August 2017; published online 19 October 2017.

[1]University of Cambridge, Laboratory for Scientific Computing, Maxwell Centre, Department of Physics, Cambridge, UK. E-mail: ggp29@cam.ac.uk; acf22@cam.ac.uk.

© 2017 Society of Exploration Geophysicists. All rights reserved.

Downloaded 07/19/18 to 131.111.184.3. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

All the above algorithms use dictionaries of predefined basis functions for sparse representation. This limits the reconstruction to the assumption that every seismic signal, with any structure at any instance of operation, is sparse in the same transform as every other instance of that signal. This assumption does not allow for large signal variations, and potential loss of reconstruction could occur. An alternative to the predefined dictionaries would be to learn the basis functions from the seismic measurements. This approach is used by Zhu et al. (2015) for the purpose of denoising seismic data with great success. The main algorithm is a modification of K-singular-value decomposition (K-SVD) (Elad and Aharon, 2006) which alternates between the coefficients and the dictionary, optimizing for the given data. Furthermore, simultaneous denoising and feature learning of seismic signals is performed by Beckouche and Ma (2014), and further dictionary learning for denoising is undertaken by Turquais et al. (2015).

In this paper, we apply beta process factor analysis (BPFA) (Zhou et al., 2012) to seismic signals for the purpose of denoising and for CS. We use BPFA to learn sparse representations of seismic signals in the shot record and time slice domains. We perform various experiments in both domains and comparisons with POCS, SPGL1, and K-SVD are undertaken.

The structure of the paper is as follows: First, the CS theory is introduced with connections to feature learning. Then, BPFA is described — its theoretical and practical aspects. A typical seismic survey setup is provided next along with how efficient data acquisition applies to different domains. CS results are then presented in the time slice and shot record domains with example reconstructions. Accompanying the results, basis functions learned from the data are presented. Afterward, denoising results are shown along with typical illustrations of noisy and cleaned signals in both domains. Computational times are also recorded for each algorithm for CS and denoising.

CS AND FEATURE LEARNING

CS allows perfect reconstruction of sparse signals from fewer measurements than the number determined by the Nyquist rate. In mathematical terms, let $\mathbf{w} \in \mathbb{R}^N$ be the sparse signal and $\mathbf{x} \in \mathbb{R}^N$ be the original signal defined by

$$\mathbf{x} = \mathbf{D}\mathbf{w}, \tag{1}$$

where $\mathbf{D} \in \mathbb{R}^{N\times N}$ maps the sparse domain to the acquisition domain and its lth column is the lth basis element, $\mathbf{d}_l \in \mathbb{R}^N$, evaluated at all N possible measuring points. CS aims to reconstruct this signal using M measurements with $M \ll N$. These measurements are described by $\mathbf{y} = \mathbf{\Omega}\mathbf{D}\mathbf{w}$, where $\mathbf{y} \in \mathbb{R}^M$ is known as the collapsed signal and $\mathbf{\Omega} \in \mathbb{R}^{M\times N}$ is the sensing matrix. Matrices with random numbers are often used for $\mathbf{\Omega}$ that correspond to the linear combination of the measurements with random coefficients. Nevertheless, such a choice limits the location of the sampling points and is restrictive for the real world. Therefore, $\mathbf{\Omega}$ is set as the zero matrix, apart from one nonzero entry equal to 1 per row. $\mathbf{\Phi} = \mathbf{\Omega}\mathbf{D}$ is used for simplicity, and therefore

$$\mathbf{y} = \mathbf{\Phi}\mathbf{w}, \tag{2}$$

where $\mathbf{\Phi} \in \mathbb{R}^{M\times N}$. The lth column of $\mathbf{\Phi}$ is the lth basis element evaluated at only M points, denoted by $\phi_l \in \mathbb{R}^M$. Variations to the formulation of equation 2 exist, such as POCS, which inserts zeros at the location of missing data points and operates in $\mathbb{R}^N$, but for the moment, the discussion is continued with $\mathbf{\Omega}$ as defined above.

One approach to solve this under-determined system is to set a sparsity constraint, by minimizing the l_0 norm of $\mathbf{w}$, $\|\mathbf{w}\|_0$. However, this problem cannot be solved in polynomial time in general (Natarajan, 1995). The breakthrough in CS was made by a series of papers (Candes and Tao, 2006; Donoho, 2006) that enabled linear programming methods to find an approximate solution to the minimization of the l_0 norm by minimizing the l_1 norm using the following formulation:

$$\hat{\mathbf{w}} = \min_{\mathbf{w}} \|\mathbf{w}\|_1 \quad \text{subject to } \mathbf{\Phi}\mathbf{w} = \mathbf{y}. \tag{3}$$

Feature learning

The choice of appropriate dictionary of basis functions $\mathbf{D}$ is fundamental for the solution (Bengio et al., 2013). Researchers have been using their domain expertise to design suitable basis functions for their specific application and careful engineering is necessary to identify those that model the data well. Feature learning is a set of techniques that allow machines to learn features/basis functions from raw data with the algorithm deciding which are suitable. In the context of CS, the task is to find a sparse representation for the training data. These can be basis functions at one common scale, or at multiple scales acquired through deep learning using many layers. In this paper, we will focus on learning basis functions at only one scale.

There are different routes to the solution: direct or indirect ones. Indirectly solving this problem involves methods that use available training data offline, learn $\mathbf{D}$, and then use the learned dictionary of basis functions for a desired task. Such models are the denoising autoencoders (Vincent et al., 2008), contractive autoencoders (Rifai et al., 2011), and restricted Boltzmann machines (Hinton, 2002) to name a few. On the other hand, a direct way learns the basis functions online, at the same time as interpolating/denoising the data, using only the available measurements.

To achieve this, the signal $\mathbf{x} = \mathbf{D}\mathbf{w} \in \mathbb{R}^N$ is divided into T subsets $\mathbf{x}^{(i)}$, $i = 1, \ldots, T$ of size $K = N/T$. For example, if we want to learn basis functions for a 2D signal of size 128×128, that is, $N = 16,384$, we can split the signal into $T = 256$ patches of size 8×8, that is, $N/T = 64$. Patches are usually extracted with overlaps to increase the number of training subsets. It is assumed that each training subset arises from a vector of coefficients, $\mathbf{w}^{(i)}$ in the sparse domain under the same transform, $\mathbf{D} \in \mathbb{R}^{K\times K}$ with additive noise $\boldsymbol{\epsilon}^{(i)}$. In other words,

$$\mathbf{x}^{(i)} = \mathbf{D}\mathbf{w}^{(i)} + \boldsymbol{\epsilon}^{(i)}. \tag{4}$$

Let $X \in \mathbb{R}^{K\times T}$ be the matrix with columns $\mathbf{x}^{(i)}$, $i = 1, \ldots, T$ and let $\mathbf{W} \in \mathbb{R}^{K\times T}$ be the matrix with columns $\mathbf{w}^{(i)}$, $i = 1, \ldots, T$. The goal is to infer simultaneously $\mathbf{D}$ and $\{\mathbf{w}^{(i)}\}_{i=1}^{T}$ from the signal subsets $\{\mathbf{x}^{(i)}\}_{i=1}^{T}$ via the optimization problem

$$\min_{\mathbf{D},\mathbf{W}} \|\mathbf{X} - \mathbf{D}\mathbf{W}\|_2^2 \quad \text{subject to } \|\mathbf{w}^{(i)}\|_0 \le T_0, \quad \text{for } i = 1, \ldots, T, \tag{5}$$

where $T_0 \ll K$ is the sparsity (number of nonzero elements) of the signal. This is done in K-SVD, which alternates between $\mathbf{D}$ and $\mathbf{W}$

Downloaded 07/19/18 to 131.111.184.3. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

and uses a pursuit algorithm to compute the coefficients $\mathbf{w}^{(i)}$ for each training subset $\mathbf{x}^{(i)}$.

Note that in equation 5, no sensing matrix $\boldsymbol{\Omega}$ is used, which would possibly be different for each subset. Instead of using $\mathbf{y}^{(i)} = \boldsymbol{\Omega}^{(i)}\mathbf{x}^{(i)}$, the components of $\mathbf{x}^{(i)}$ where data are missing are set to zero (Aharon et al., 2006). In the case of noisy signals, the original values with noise are used (Elad and Aharon, 2006). An operator is used to alter the objective function of the optimization problem to indicate the locations of available data. Inserting zeros in the place of missing data points is also done in POCS, which helps to preserve the location and structure inside each $\mathbf{x}^{(i)}$. However, SPGL1, which will also be used in the experiments, uses a sensing matrix $\boldsymbol{\Omega}$ and collapses the data as in equation 2 which is the traditional formulation (Candes and Wakin, 2008).

Learning the basis functions at the same time as performing denoising and/or interpolation uses training data that are corrupted. One might expect that the learned dictionary is only useful for sparsely representing the corrupted signals. However, this is not the case as examined in various models for feature learning (Vincent et al., 2008; Srivastava et al., 2014) and in seismic applications (Beckouche and Ma, 2014) where denoising and feature learning were performed simultaneously. In fact, adding noise or dropping out measurements from the training data is recommended as a regularization to avoid over-fitting (Bengio et al., 2013). Furthermore, many training subsets mitigate the risk of learning corruption.

BETA PROCESS FACTOR ANALYSIS

Most of the methods in the seismic literature use predefined basis functions $\mathbf{D}$ with a fixed size that is very limiting. BPFA (Zhou et al., 2012) is a method that overcomes these limitations. In the real world, there are infinitely many possibilities for $\mathbf{D}$. In fact, $\mathbf{D}$ itself could be infinite with $\mathbf{D} \in \mathbb{R}^{K\times L}$ where $L \to \infty$. We assume the data matrix X is generated by an underlying process with its columns $\mathbf{x}^{(i)}$, $i = 1, \ldots, T$ generated by the graphical model in Figure 1. To introduce this, we explicitly separate the value of a coefficient in $\mathbf{w}^{(i)}$ from the fact whether it is nonzero or zero. This means that if the coefficient is nonzero, the corresponding basis function is used when generating $\mathbf{x}^{(i)}$. In particular, we introduce probabilistic variables $\mathbf{z}^{(i)}$ and $\mathbf{s}^{(i)}$ such that

$$\mathbf{w}^{(i)} = \mathbf{z}^{(i)} \odot \mathbf{s}^{(i)}, \tag{6}$$

where $\odot$ represents the elementwise vector product, $\mathbf{z}^{(i)} \in \mathbb{R}^{L}$ signifies whether a basis function is used, and $\mathbf{s}^{(i)} \in \mathbb{R}^{L}$ are the values of the coefficients. We assume that $\mathbf{z}^{(i)}$ are generated by a Bernoulli process parametrized by a beta process,

$$\mathbf{z}^{(i)} \sim \prod_{l=1}^{L} \mathrm{Bernoulli}(\pi_l), \tag{7}$$

where π_l is the probability that the lth basis function is used when $\mathbf{x}^{(i)}$ is generated. The probabilities $\pi = (\pi_1, \ldots, \pi_L)$ themselves are generated by a beta process

$$\pi \sim \prod_{l=1}^{L} \mathrm{Beta}(a/L, b(L-1)/L), \tag{8}$$

where a, b are parameters characterizing the process. The variable $\mathbf{s}^{(i)}$ on the other hand is a priori normally distributed with zero mean and variance $\gamma_s^{-1}\mathbf{I}_L$, where γ_s is modeled by a hyper prior with gamma distribution, gamma(c, d), with c, d characterizing it and $\mathbf{I}_L$ is the $L \times L$ identity matrix. To summarize,

$$\mathbf{x}^{(i)} = \mathbf{D}\mathbf{w}^{(i)} + \boldsymbol{\epsilon}^{(i)}, \tag{9}$$

where the columns $\{\mathbf{d}_l\}_{l=1}^{L}$ of $\mathbf{D}$ are modeled by

$$\mathbf{d}_l \sim \mathcal{N}(0, K^{-1}\mathbf{I}_K) \tag{10}$$

and the noise is modeled by

$$\boldsymbol{\epsilon}^{(i)} \sim \mathcal{N}(0, \gamma_\epsilon^{-1}\mathbf{I}_K), \tag{11}$$

where $\mathbf{I}_K$ is the $K \times K$ identity matrix and γ_ϵ is modeled by a gamma distribution, gamma(e, f). There are many parameters that govern the model with a summary given in Figure 1, and a discussion on their settings is given later on.

The algorithm then estimates posterior conditional probabilities for $\mathbf{z}^{(i)}$ by adjusting the Bernoulli distributed $\{\pi_l\}_{l=1}^{L}$. In addition, it estimates similar posterior probabilities for $\mathbf{s}^{(i)}$ and $\{\mathbf{d}_l^{(i)}\}_{l=1}^{L}$ by adjusting the mean and variance of their respective Gaussian distributions. For $\mathbf{s}^{(i)}$ and $\boldsymbol{\epsilon}^{(i)}$, their respective precisions γ_s and $\gamma_{\boldsymbol{\epsilon}}$ need to be updated. The strategy for the reestimations maximizes the likelihood that this choice of variables generates $\{\mathbf{x}^{(i)}\}_{i=1}^{T}$. After a predefined number of iterations, the final values of the variables are set from their respective posterior distributions.

This process formulation is known more generally as Bayesian nonparametrics. These approaches use prior distributions within the Bayesian framework that could represent objects on an array of infinite size, in the particular case of feature learning, an infinite array of features (Griffiths and Ghahramani, 2011). Bayes' rule uses a prior distribution (before observing any data) and a likelihood to estimate the posterior distribution (after observing any data) of

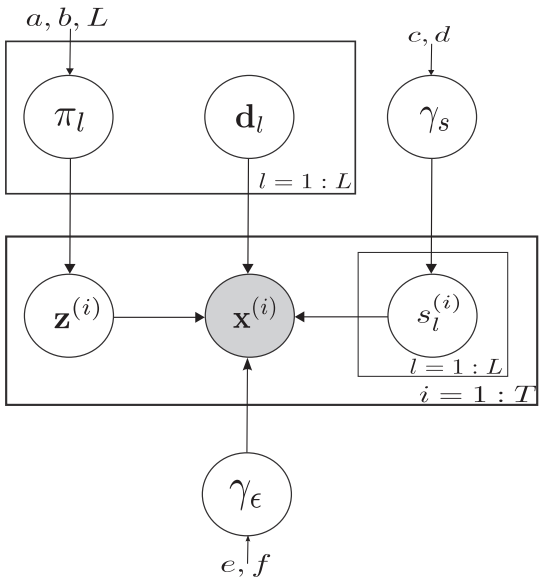

Figure 1. Graphical model for the BPFA. The circles represent the random variables of the model that are described by probability distributions. The others are the parameters that govern each probability distribution.

Downloaded 07/19/18 to 131.111.184.3. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

the model's variables. For example, equations 10 and 11 are prior distributions that reflect assumptions about their variables. The likelihood model is the probability of producing the observations (data samples) given the current configuration of the variables (i.e., equation 9) and ideally should be maximized. The posterior distribution of a variable is the distribution after using the data samples for the maximization of the likelihood. It is proportional to the product of the prior and the likelihood distributions.

Practical importance of variables and parameter settings

Figure 1 organizes all the model's variables and shows which parameters are necessary to be set. First, $\{c, d, e, f\}$ are parameters that describe the gamma distributions. These are all set to 10^{-6} as is done usually to make them noninformative (Tipping, 2001). The parameters $\{a, b\}$ describe the beta distribution that controls the probabilities whether a particular basis function generates a particular training subset. As discussed in Paisley and Carin (2009) as $L \to \infty$, the sparsity of $\mathbf{z}^{(i)}$ is a random variable drawn from a Poisson's distribution, Poisson (a/b). In practice, however, we fix L to a specific number, and as is shown by Zhou et al. (2009), the parameters $\{a, b\}$ are in general noninformative with the sparsity inferred from the data. Thus, we set $a = 1$ and $b = T/8$ as specified by Zhou (2012).

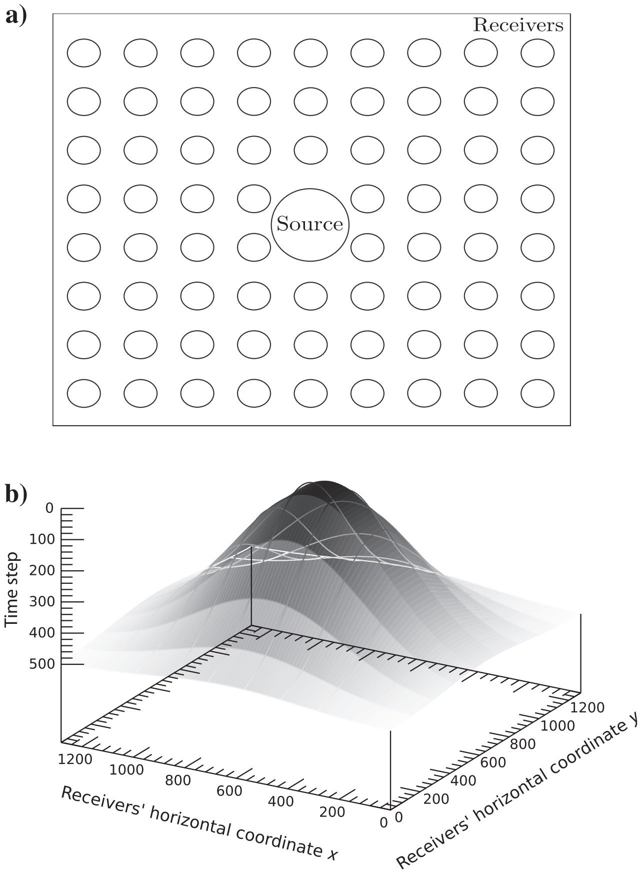

Figure 2. An illustration of a typical seismic survey setup with a regularly spaced grid of receivers (a). A multivariate Gaussian distribution plotted for illustration purposes only, to aid in the explanation of the different seismic domains (b).

In our experiments, the upper limit of the dictionary's size L is set to $L = 256$. Similar results are obtained with $L = 512$ (Zhou et al., 2012), and therefore to reduce the computational cost, the former was set. However, to learn more basis functions for future investigations, we fixed this number and did not allow shrinking. As discussed before, in equation 4, the training data can be extracted from many signals or from just one provided that there is enough data to prevent under-fitting. We performed the experiments on the reconstruction of 128×128 signals. Each signal was reconstructed individually; that is, for each 128×128 signal, only training data from that signal was used. Each $\mathbf{x}^{(i)}$ was of size 8×8 extracted from the 128×128 signal with overlaps. Further information on the patch processing can be found in Zhou et al. (2009).

Initialization, inference, and analogy with POCS

All the unknown variables $\{\mathbf{z}^{(i)}\}_{i=1}^{T}$, $\{\mathbf{s}^{(i)}\}_{i=1}^{T}$, $\{\mathbf{d}_l\}_{l=1}^{L}$, $\{\pi_l\}_{l=1}^{L}$, $\{\boldsymbol{\epsilon}^{(i)}\}_{i=1}^{T}$ need to be inferred using the observed training data. Analytic equations for each variable have been derived in Zhou et al. (2012), where the conditional probability distribution of each, conditioned on all others, is obtained. Thus, it is possible to find an approximate solution by alternating between the variables, keeping the ones that have already been estimated fixed and estimating the one that is not fixed. To start, all variables have to be initialized. The term $\mathbf{D}$ is initialized based on a singular-value decomposition (SVD) of X which was found to converge faster as opposed to random initialization. Furthermore, the noise precision γ_ϵ is initialized and scaled by the inverse variance of the available training data in a similar fashion as in Tipping and Faul (2003). This way, we ensure that the noise variance is not overestimated. All other variables are initialized randomly from their respective prior distributions.

An analogy can be drawn between BPFA and POCS. POCS transforms X to a predefined sparse domain (e.g., Fourier) and estimates the coefficients of the sparse transform of the data. The same idea of decomposing the data as the linear combination, $\mathbf{X} = \mathbf{DW}$, is used. The terms $\mathbf{W}$ are the Fourier coefficients, and $\mathbf{D}$ is the Fourier base, where in the case of POCS (Abma and Kabir, 2006), the fast Fourier transform (FFT) operator is used for efficiency. One iteration of POCS is analogous to one iteration of BPFA for obtaining the coefficients $\mathbf{W}$ but only partly. BPFA then considers the coefficients (or rather the variables that compose the coefficients $\{\mathbf{s}^{(i)}\}_{i=1}^{T}$ and $\{\mathbf{z}^{(i)}\}_{i=1}^{T}$) as fixed and obtains the dictionary $\mathbf{D}$.

SEISMIC EXPERIMENTS

Seismic surveys usually consist of arrays of sources and receivers in a generally regular pattern at or near the earth's surface. Body waves created by a surface source are reflected back by impedance changes caused by changes in rock properties. The reflected waves are recorded by receivers on a continual basis and then discretized. A schematic of this setup is illustrated in Figure 2a. Figure 2b shows, for illustration purposes only, a plot of a multivariate Gaussian distribution to help the discussion about the different domains used. The x- and y-axes correspond to the spatial coordinates of the receivers, and the z-axis corresponds to time. We make use of the shot record domain keeping constant only one of the x- or y-coordinates, giving a 2D projection with the time on one axis and the respective coordinate on the other. Furthermore, in the time slice, x and y coordinates are used and the z-axis, time, is kept constant. Missing receiver data are different in the shot record and in the time

Downloaded 07/19/18 to 131.111.184.3. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

slice domain. For the former, if a receiver's data are missing, then the corresponding column of data points is missing. On the other hand, for a time slice, this means that only one data point is missing on the location of the receiver. To perform comparisons, test data were extracted from a synthetic data set that was generated numerically using the SEAM-II (Oristaglio, 2012) model as input. The modeling was carried out by BP in Houston. Results of the experiments in CS and denoising follow.

CS results

The aim of the experiments is to show the level of reconstruction accuracy with varying number of receivers both removing them randomly or regularly. Furthermore, the computational cost is examined, and illustrations of learned basis functions from BPFA are provided. In all experiments, the SPGL1 package (van den Berg and Friedlander, 2007) with the DCT is used. Similarly, for POCS, the MATLAB code (Abma, 2011) was used, and for BPFA the package in Zhou (2012) was applied to the problem. One figure of merit for evaluating the reconstruction accuracy is feature similarity (FSIM) (Zhang et al., 2011), which creates phase congruency and edge detection feature maps for the reconstruction and for the original signals. More specifically, FSIM is calculated by

$$\mathrm{FSIM} = \frac{\sum_{i\in\Lambda} S_L(i) PC_m(i)}{\sum\limits_{i\in\Lambda} PC_m(i)}, \tag{12}$$

where Λ is the space of the signal, $PC_m()$ is the maximum phase congruency between the two signals at a specific location, and $S_L(i) = S_{PC}(i)S_G(i)$, where

$$S_{PC}(i) = \frac{2PC_1(i)PC_2(i)}{PC_1^2(i) + PC_2^2(i)} \tag{13}$$

and

$$S_G(i) = \frac{2G_1(i)G_2(i)}{G_1^2(i) + G_2^2(i)}. \tag{14}$$

The terms $PC_1()$ and $G_1()$ are the phase congruency and edge detection maps of the original, respectively, and $PC_2()$ and $G_2()$ are the respective maps of the reconstruction. Using these two feature maps for the original and reconstruction, FSIM compares their similarity. Higher values (closest to 1.00) indicate greater similarity.

In addition, the quality of reconstruction Q is also calculated to provide a better understanding about the differences in accuracy. The quality Q is defined as in Kazemi et al. (2016) by

$$Q = 10 \log \frac{\|\mathbf{x}\|_2^2}{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2}, \tag{15}$$

where $\mathbf{x}$ is the original signal and $\hat{\mathbf{x}}$ is the reconstruction.

Variability of parameters for algorithms

Different initialization of parameters, patch sizes that they operate on, the choice of the basis functions for SPGL1, stopping criteria for all algorithms, different thresholding operators for POCS to name a few, all can affect the results. To address this variability in full, experiments with all possible setups are necessary. However, in this paper, we decided to explore two key elements: the patch size and the stopping criteria. We performed experiments in the time slice domain for POCS and SPGL1 to determine a suitable set of parameters. The choice of the dictionary for SPGL1 was fixed to the DCT. The parameters of BPFA were fixed to those discussed in the previous section. To ensure that the results are consistent over different instances of signals with different structures and variance, we have extracted 250 sections of size 128 × 128 from time slices and 200 sections of size 128 × 128 from shot records from the SEAM-II data set described earlier.

POCS configurations for time slices

We experimented with different numbers of iterations for the algorithm to terminate, and the patch size was varied between $\{8\times8, 16\times16, 32\times32, 64\times64, 128\times128\}$ and nonoverlapping. A plot of mean Q against the measurements over all sections is given in Figure 3a. It can be seen that the larger the patch size, the better the

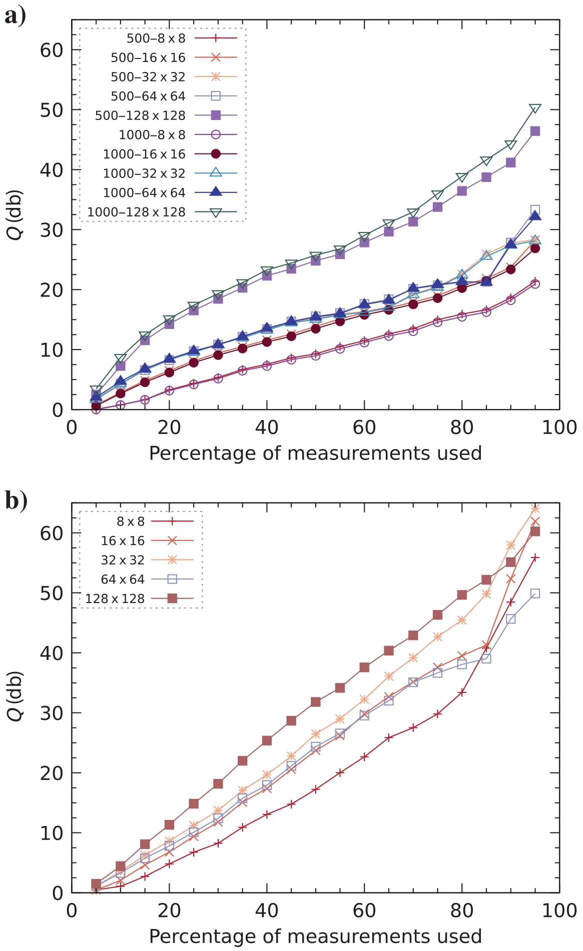

Figure 3. Mean Q plots over 250 sections of time slices of size 128 × 128. Different POCS configurations (a) are explored by varying the number of iterations to termination (first number in the legend) and by varying the patch size (second number in the legend). Different SPGL1 configurations (b) are explored by varying the patch size (number in the legend).

Downloaded 07/19/18 to 131.111.184.3. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

reconstruction using the Fourier transform. The configuration with the best performance in our experiments operates on 128 × 128 patches; however, the number of iterations is not obvious whether using 500 or 1000 iterations — the results are very similar with 1000 iterations being slightly better. Therefore, in the following comparisons, POCS with 500 iterations and a patch size of 128 × 128 will be used as the best compromise between time and accuracy.

SPGL1 configurations for time slices

SPGL1 can be modified greatly with regard to its stopping criteria, and an exhaustive parameter search would be required. One stopping criterion checks the residual between the available measurements and the estimation, another checks convergence of intermediate solvers, and another sets the maximum number of iterations. We experimented with the value of the residual and suggest using a difference that is much smaller than the l_2 norm of the available data, e.g., between $10^{-6}\|\mathbf{y}\|_2$ and $10^{-9}\|\mathbf{y}\|_2$. Figure 3b shows the mean Q with patch sizes from 8 × 8 to 128 × 128. The 128 × 128 patch size gives the best performance when < 85% of the measurements are used, slightly better than 32 × 32 and much better than the rest. Larger patch sizes could perform better if the stopping criteria were tuned, i.e., by changing the number of iterations. From these experiments, we chose the 128 × 128 SPGL1 configuration to compare against the BPFA because it gives the best accuracy.

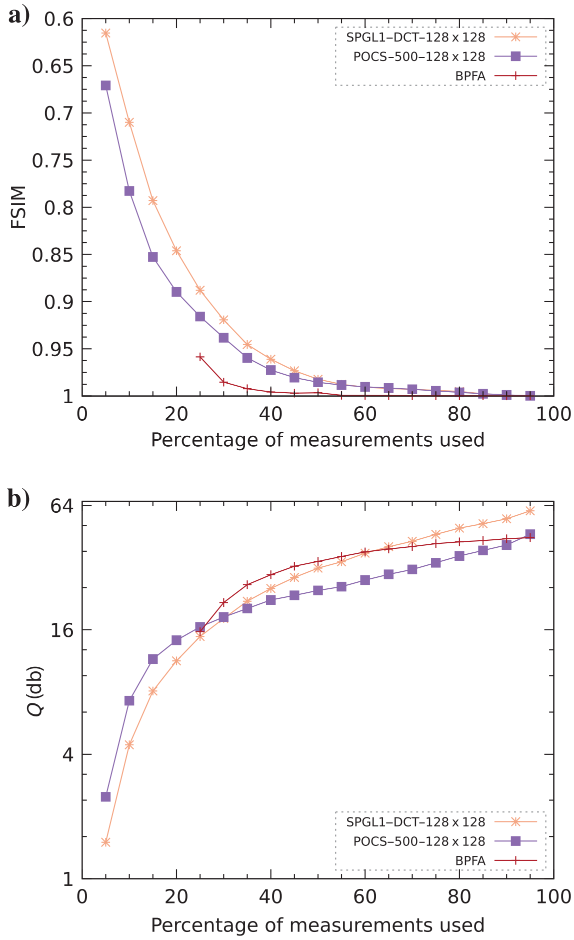

Figure 4. Mean reconstruction accuracy over 250 sections of time slices of size 128 × 128 for different measurements using FSIM (a) and Q (b) as quality measures. BPFA results are shown with 25% being the least possible percentage.

Time slice comparisons of different algorithms

Using the selected configurations, we use the same 250 sections of time slices to compare against BPFA. BPFA parameters were fixed to the ones discussed in the previous section. Figure 4a shows the mean FSIM plotted for BPFA, SPGL1, and POCS for varying percentages. BPFA outperforms both algorithms when 25% and greater receivers are used. With fewer, BPFA does not have sufficient training data to learn the basis functions (discussed in more detail later). POCS performs better than SPGL1 when 55% of receivers and fewer are used. Figure 4b shows the mean Q for different measurements.

There are some differences with BPFA performing better than both other algorithms when measurements are between 30% and 60%. POCS performs better than SPGL1 with 30% and fewer and SPGL1 is better than both algorithms with 70% and greater measurements used. An example using all algorithms can be seen in Figure 5 with the error difference maps in Figure 6.

BPFA performs significantly better than the other algorithms by learning the appropriate dictionary of basis functions. A collection of learned basis functions can be seen in Figure 7. We will discuss later in the Discussion section some technical insights on what makes the performance of one algorithm better than another and potential reasons for the metrics' differences.

Lower limit for BPFA

Learning a dictionary of basis functions is not always possible. When the percentage of receivers used is less than 25%, the model underfits with not enough training data and it already starts to perform badly with <30% (Figure 4). Figure 8a shows a section from a time slice with only 30% of the data. The BPFA reconstruction in Figure 8b is successful by learning the basis functions in Figure 8e that capture the signature of the time slice and fit the data well. Nevertheless, when only 20% are used in Figure 8c, the basis functions learned in Figure 8f do not capture the variations in the data, which results in poor reconstruction as seen in Figure 8d.

Computational complexity

Depending on the convergence criteria, the algorithms could terminate earlier than expected; nevertheless, the worst-case scenario is mentioned here. The SPGL1 is composed of three potentially heavy computational steps: two matrix–vector products and a step that computes the projection of data. The worst-case complexity for the projection is $\mathcal{O}(n \log n)$, where n is the dimensionality of the signal, but on average it performs much better (van den Berg and Friedlander, 2009). POCS main computations are the FFT and inverse FFT (IFFT), which are $\mathcal{O}(n \log n)$ (Abma and Kabir, 2006), and are also dependent on the number of iterations until termination. BPFA scales linearly as a function of the patch size K, the dictionary size L, the sparsity level T_0 of the signals, and the number of available training data T (Zhou et al., 2009).

Downloaded 07/19/18 to 131.111.184.3. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

The orders of computational complexity are generally informative; however, because there are many algorithmic variations, we recorded the computational time to get a better understanding of their cost. All experiments were performed as single-core jobs on machines with Intel Xeon CPU E5-2650 with 2.00GHz. The mean computational time for three different percentages for all three algorithms is shown in Table 1 averaged over all 250 sections along with the mean FSIM and Q. Experiments were performed using the respective MATLAB packages. The configurations used were the same as the ones used in the reconstruction accuracy experiments. POCS is the fastest solver with the use of the FFT operator. BPFA is the slowest due to the extra requirement of learning the basis functions.

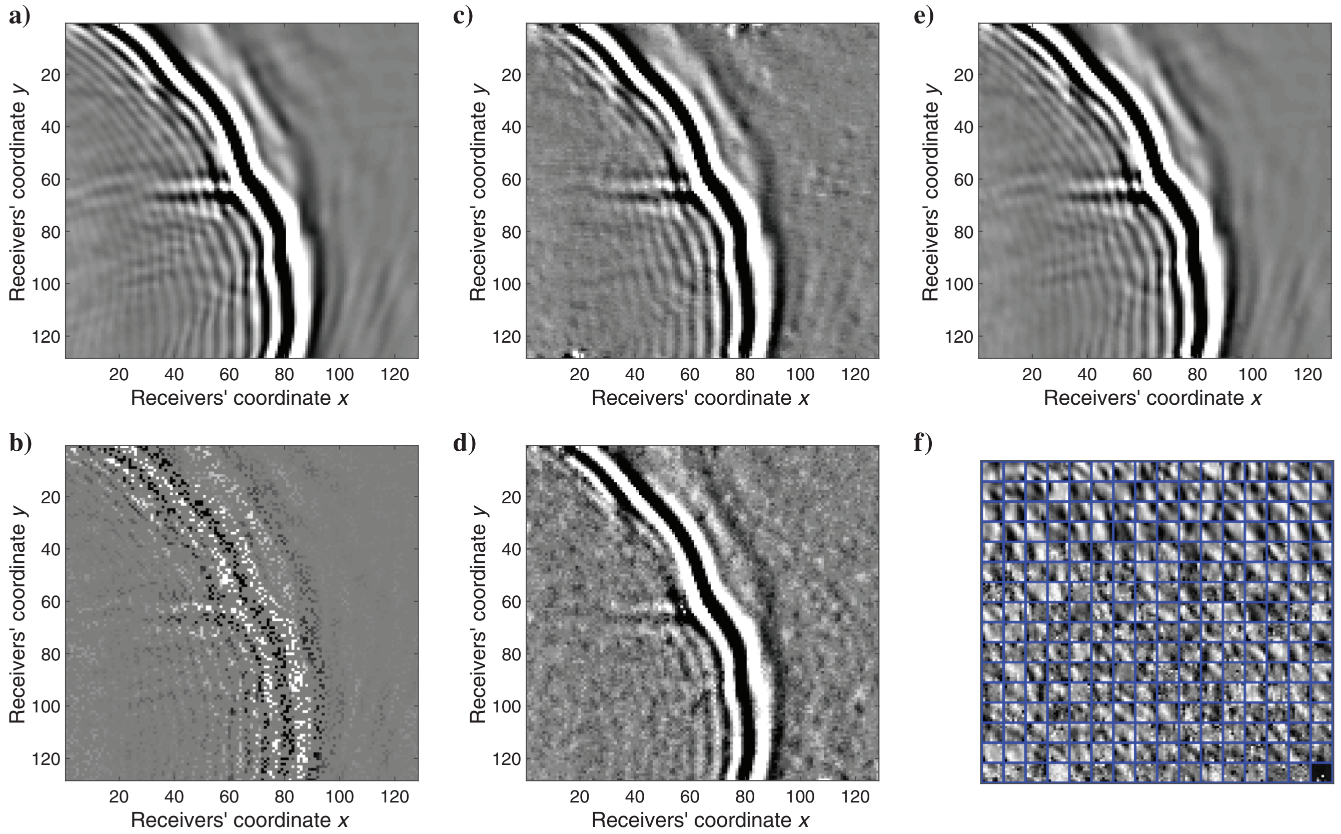

Figure 5. An example of a section from a time slice from the SEAM-II data set (a) using only 30% of the receivers (b). Reconstructions with different algorithms are illustrated showing POCS (Q = 30.75 db) (c) doing better than SPGL1 (Q = 25.51 db) (d). BPFA (Q = 35.45 db) (e) is better than both by learning the appropriate basis functions (f) for the given section.

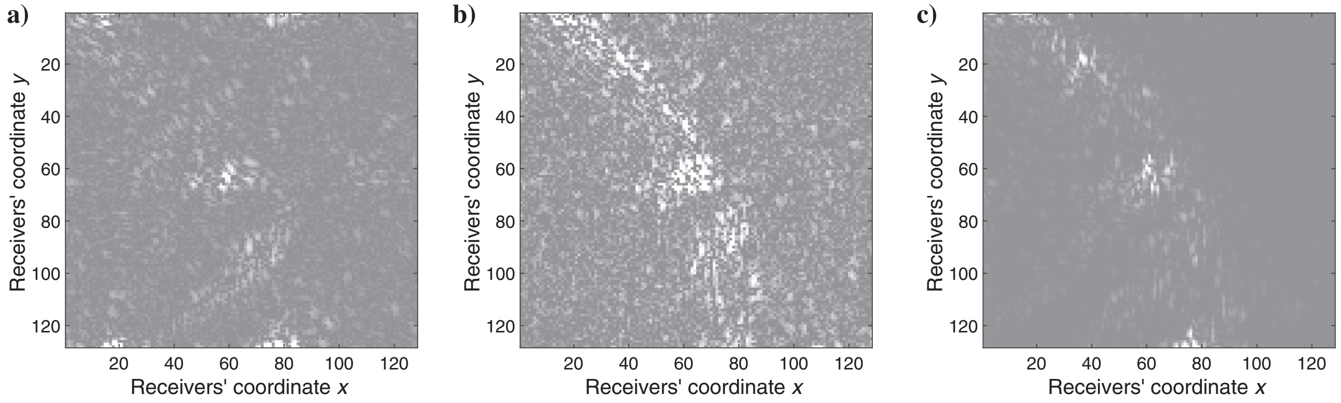

Figure 6. Maps of the absolute reconstruction error for each algorithm for the signal in Figure 5. (a) The error of POCS that is visible in the regions of large changes. (b) The error of SPGL1 which is larger. (c) The minimal error of the BPFA.

Downloaded 07/19/18 to 131.111.184.3. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

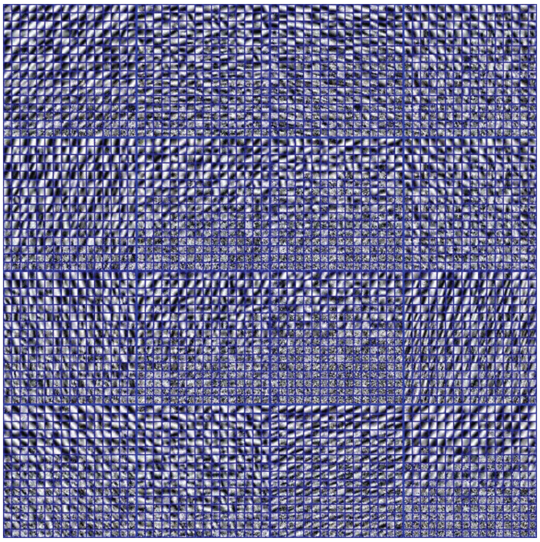

Figure 7. Each dictionary of 256 basis functions is learned from an individual section of a time slice, resulting in as many dictionaries as reconstructions. An ensemble of dictionaries is available that captures different signal variations (depending on the time slice used for training) with different orientations of large changes.

Shot record comparisons of different algorithms

Missing data in the shot record domain is equivalent to columns of missing data points (often called traces). Thus, the pattern of data removal differs from that in the time slice domain and investigating its effect on performance is required. In addition, the signal structure is different and therefore it is worth investigating whether the basis functions learned are different. Two hundred sections of size 128×128 of shot records were extracted from the SEAM-II data set. Different sized blocks of traces of width $\{2, 4, 8\}$ were removed every $\{4,8,16,32,64,128\}$ traces from the starting point of removal. The configurations of the algorithms were set based on the time slice experiments. POCS was used with 500 iterations and on 128×128 patches. The SPGL1 was set with the DCT with the same stopping criteria as before and operating on 128×128 patches. BPFA had the settings mentioned in the previous section. An example of shot record reconstruction can be seen in Figure 9.

Figure 10 shows an ensemble of dictionaries learned from different sections of shot records. The signal variations captured by the basis functions are similar to those learned for the time slice domain, with the difference that there are two main orientations of signal changes approximately on 45° and 135° as opposed to approximately four orientations in the time slice domain (Figure 7). This illustrates that

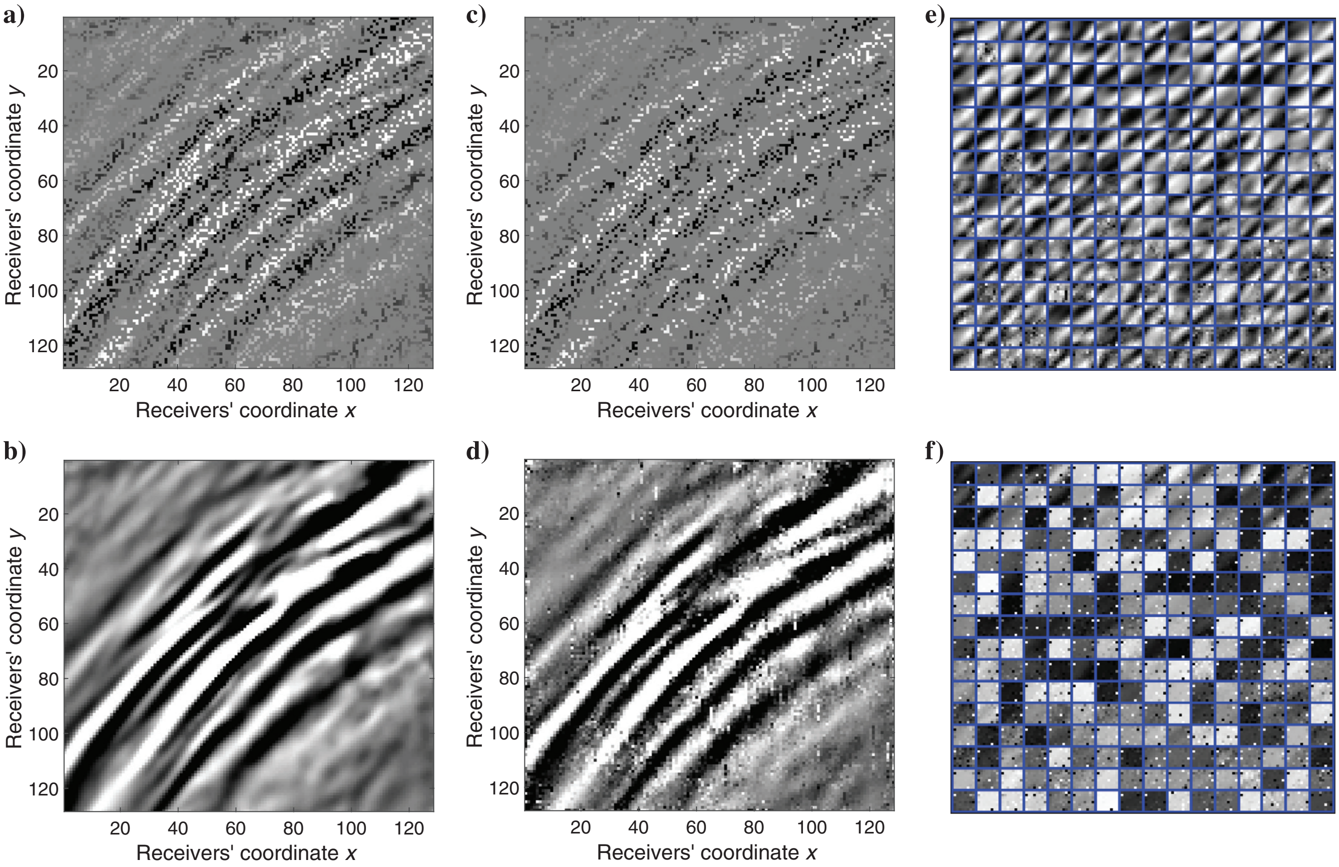

Figure 8. Example of a section from a time slice from the SEAM-II data set using only 30% of the receivers (a). BPFA reconstruction (b) by learning appropriate basis functions (e). If the training data are 20% (c) the model underfits (d) with the learned basis functions not capturing all the variations in the data (f).

Downloaded 07/19/18 to 131.111.184.3. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

if the dictionaries learned in the time slice domain were used in the shot record domain, almost half of the basis functions would have been redundant with different features being important in different domains.

The mean FSIM and mean Q for 200 sections of size 128×128 from shot records for BPFA, SPGL1 with DCT and POCS are illustrated. When blocks of two, in Table 2, or four traces, in Table 3, are missing, BPFA is the best on average. However, in the case of blocks of eight missing in Table 4, BPFA performs badly. This is due to the fact that BPFA splits each 128×128 section to 8×8 patches and thus BPFA does not have sufficient training data. Further analysis is given in the "Discussion" section.

Denoising results

The task of denoising is estimating the level of noise and choosing the appropriate basis functions with the correct coefficients that correspond to the noise-free signal. Usually, the dictionary is prefixed and chosen to provide a sparse representation. In this paper, we propose using BPFA to learn the basis functions from the available data as was done in recent studies (Beckouche and Ma, 2014; Zhu et al., 2015). To evaluate the performance of BPFA, we used the SEAM-II data set and added Gaussian noise with increasing levels of distortion, controlled by the noise variance. We extracted 200 sections of size 128×128 from time slices of varying structures and 200 sections of size 128×128 from shot record signals. We compared BPFA against the K-SVD, which has shown success in seismic denoising (Turquais et al., 2015; Zhu et al., 2015). The K-SVD results were produced using the MATLAB package from one of the authors' website (Elad, 2006) and the BPFA results from the same source as before (Zhou, 2012). We have used the predefined settings of the packages with no tuning.

Different levels of noise in the seismic signals translate to a varying signal-to-noise ratio (S/N). There are numerous definitions of the S/N and it is difficult to compare between studies. However, in this paper, the important value is not the S/N but rather the quality

Table 1. Accuracy and time trade-off analysis for different algorithms averaged over 250 sections of 128×128 of time slices (bold are best values).

Trade-off between accuracy and computation						
Percentage used	30%		60%		90%	
	Time (s)	FSIM/Q (db)	Time (s)	FSIM/Q (db)	Time (s)	FSIM/Q (db)
BPFA	218.73	**0.985/21.7**	396.49	**0.999/38.1**	579.24	0.999/44.1
POCS	**13.94**	0.938/18.5	**12.55**	0.990/27.8	**14.23**	0.998/41.2
SPGL1	33.85	0.919/18.2	33.15	0.991/37.6	21.53	**0.999/55.1**

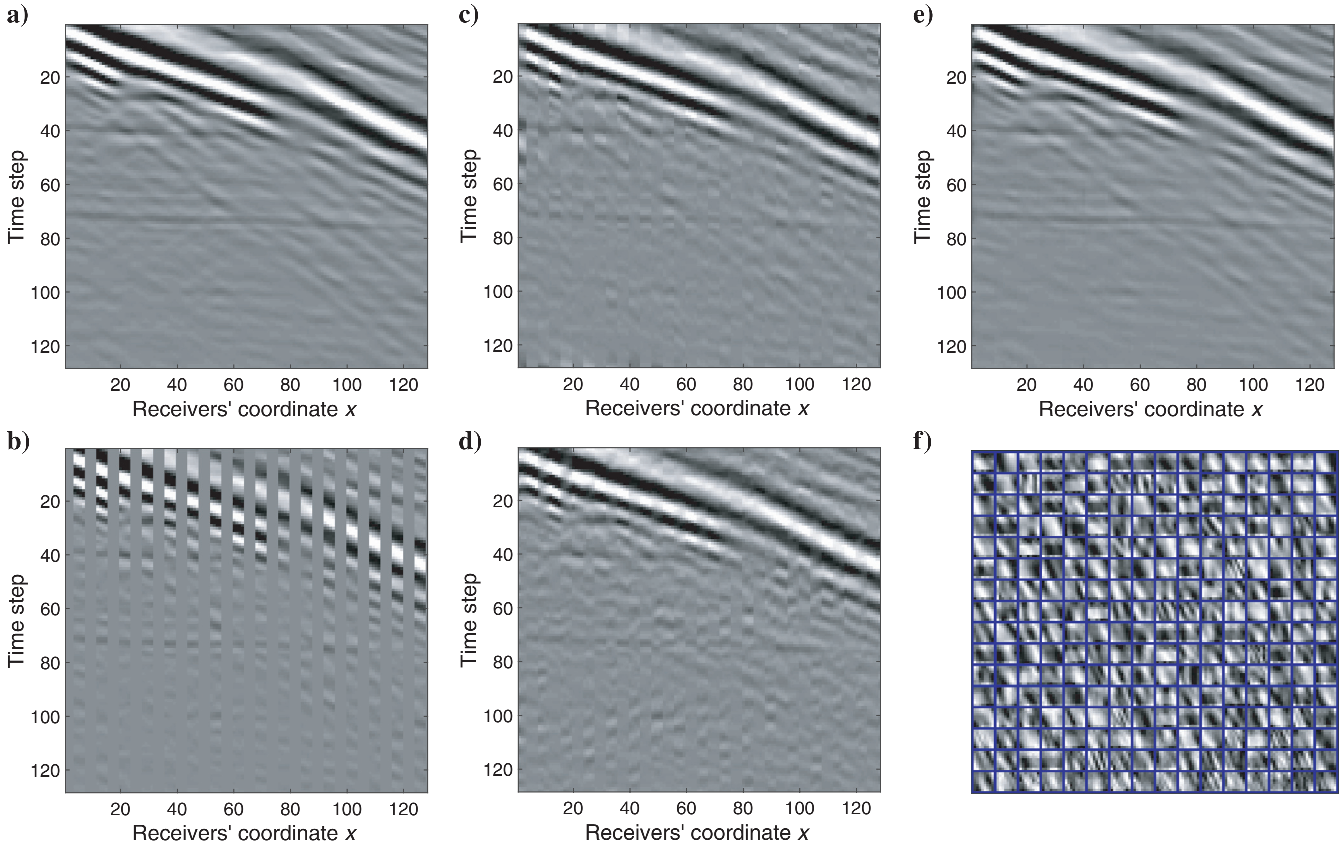

Figure 9. Shot (a) reconstruction from pattern of removal of four receivers missing every eight receivers (b). POCS ($Q = 22.97$ db) (c) performs better than SPGL1 ($Q = 21.50$ db) (d) but BPFA ($Q = 29.54$ db) gives the best accuracy (e) with learned basis functions (f).

Downloaded 07/19/18 to 131.111.184.3. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

of reconstruction Q of each algorithm and how it compares with the other. In our experiments, we varied the noise variance to control this ratio and we define it as it was done in the denoising study (Kazemi et al., 2016) with

$$\mathrm{S/N} = \frac{\alpha_{\mathrm{rms}}^2}{\sigma_n^2}, \tag{16}$$

where α_{rms} is the root mean square amplitude of the noise-free signal and σ_n^2 is the noise variance. Our experiments were undertaken over multiple seismic signals, and thus the mean S/N was calculated over all signals. Six different values of the noise variance were used, resulting in six different mean S/N values for time slices and six for shot records. To evaluate the reconstruction, we define the quality as was defined in equation 15.

The mean Q for all sections is plotted against varying mean S/N values. It can be seen in Figure 11a and 11b that the BPFA attains higher levels of Q than the K-SVD for all S/N, illustrating its superiority. Because accuracy is not always enough, the computational time is shown in Figure 11c with the K-SVD being faster than the BPFA. Nevertheless, by learning basis functions and finding the most popular dictionaries, it could be possible to reuse them without needing to learn every time. An example of time slice denoising by both algorithms can be seen in Figure 12 and for shot record denoising in Figure 13. The BPFA learns a dictionary of basis functions with high-frequency characteristics as opposed to the K-SVD. Using these basis functions, it can reconstruct more details.

CS and denoising

To illustrate the potential of BPFA, we experimented also with both denoising and at the same time interpolating the data. An example of this can be seen in Figure 14, where 50% of the receivers are used and, on those, Gaussian noise is added with $\mathrm{S/N} = 20.84$. BPFA was able to reconstruct the signal preserving the most important features even with the presence of noise.

DISCUSSION

Different configurations of the algorithms' parameters can yield different results in performance, in reconstruction accuracy and in

Table 2. Reconstruction accuracy with patterns of removal of blocks of two traces (bold are best values).

Shot record accuracy with missing traces of blocks of two						
Pattern	4	8	16	32	64	128
	FSIM/Q (db)	FSIM/Q (db)	FSIM/Q (db)	FSIM/Q (db)	FSIM/Q (db)	FSIM/Q (db)
BPFA	**0.981/37.6**	**0.993/46.4**	**0.999/50.8**	**0.999/52.9**	**0.999/53.9**	**0.999/54.4**
POCS	0.905/26.8	0.968/33.1	0.978/36.1	0.985/38.2	0.992/39.7	0.995/40.7
SPGL1	0.943/32.7	0.992/42.5	0.997/46.8	0.999/50.3	0.999/52.0	0.999/52.8

Table 3. Reconstruction accuracy with patterns of removal of blocks of four traces (bold are best values).

Shot record accuracy with missing traces of blocks of four					
Pattern	8	16	32	64	128
	FSIM/Q (db)	FSIM/Q (db)	FSIM/Q (db)	FSIM/Q (db)]	FSIM/Q (db)
BPFA	**0.976/35.4**	**0.979/38.9**	**0.996/46.0**	**0.999/48.2**	**0.999/49.3**
POCS	0.899/27.1	0.957/31.9	0.972/34.5	0.983/36.3	0.990/37.6
SPGL1	0.936/30.1	0.976/36.6	0.988/40.1	0.993/42.3	0.995/43.5

Table 4. Reconstruction accuracy with patterns of removal of blocks of eight traces (bold are best values).

Shot record accuracy with missing traces of blocks of eight				
Pattern	16	32	64	128
	FSIM/Q (db)	FSIM/Q (db)	FSIM/Q (db)	FSIM/Q (db)
BPFA	0.775/15.7	0.857/21.4	0.928/26.6	0.986/36.4
POCS	0.881/26.2	0.938/29.8	0.963/32.2	0.977/33.8
SPGL1	**0.895/26.8**	**0.949/30.9**	**0.971/33.4**	**0.982/35.0**

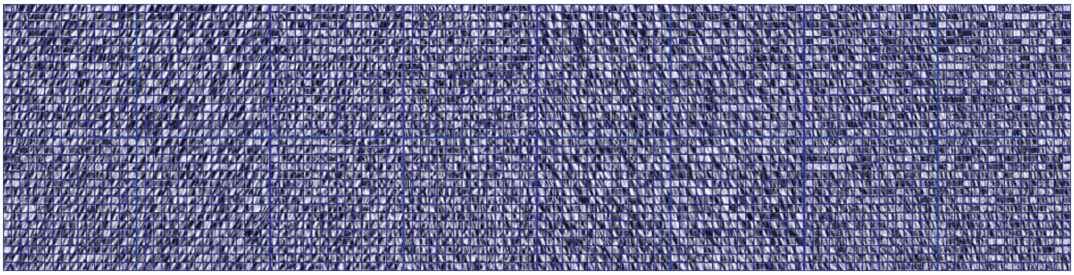

Figure 10. Each dictionary is trained on a section of a shot record with a certain percentage of receivers.

Downloaded 07/19/18 to 131.111.184.3. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

computational time. In addition, different data sets and different domains of operation (time slice and shot record) can give different results. To obtain the best possible combination, an exhaustive parameter search would be necessary. This would require many experiments, and this is not the purpose of the paper; nevertheless, we chose to experiment in the time slice domain with certain parameter settings that we think are essential for an algorithm's success. These are the convergence criteria and the patch sizes. We varied the patch size for POCS and SPGL1 and obtained different accuracy and computational times with fixed convergence criteria over 250 sections to allow for variability in the data set. For POCS, we varied the number of iterations as well. The initialization of BPFA with regards to the noise variance was set appropriately to avoid overestimation. Overestimating the noise variance leads to under-fitting, the algorithm would assume that early termination is justified because variations are explained by noise.

From the POCS experiments, the performance with the best reconstruction accuracy was obtained when the patch sizes and the number of iterations were the largest. Larger patch sizes contain more signal structure, and this could allow the algorithm to use

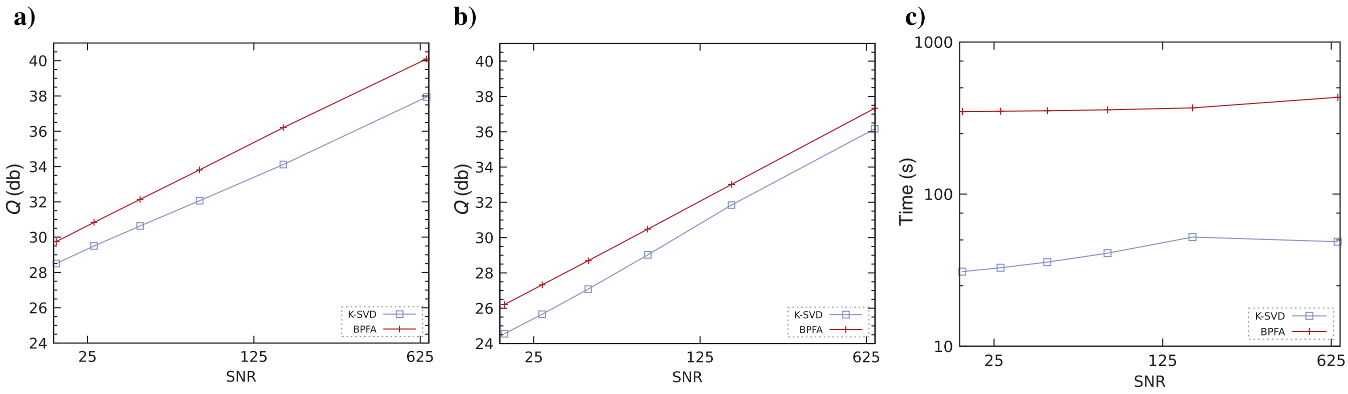

Figure 11. Mean reconstruction accuracy for time slices (a), for shot records (b) and mean computational time (c) for time slices for 200 sections of 128 × 128 per domain with varying S/N.

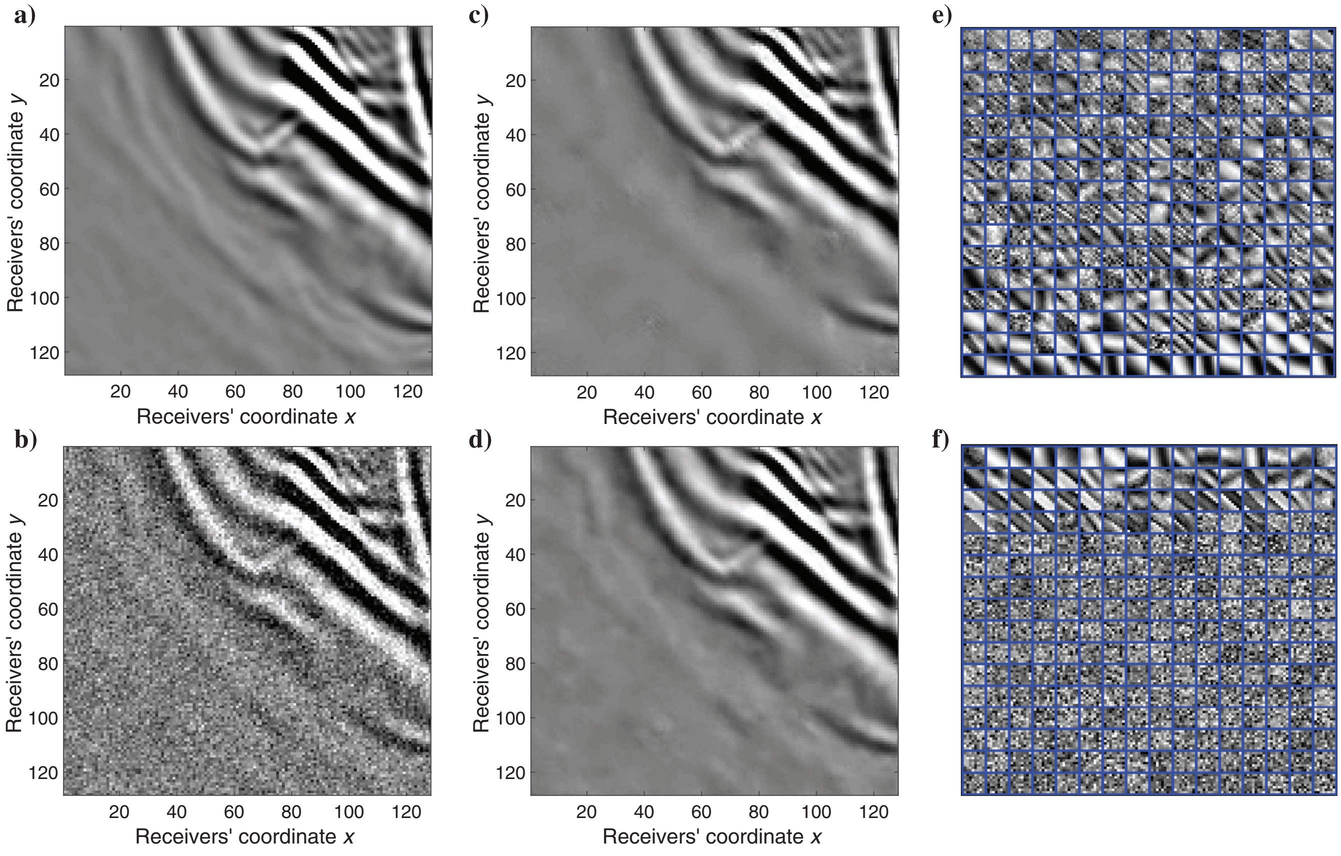

Figure 12. A section of a time slice from the SEAM-II (a) is corrupted (S/N = 20.67) (b). BPFA (Q = 24.82 db) (d) obtains better quality than the K-SVD (Q = 24.25 db) (c). The learned basis functions of K-SVD (e) and BPFA (f) are shown. BPFA puts greater emphasis on higher frequencies.

Downloaded 07/19/18 to 131.111.184.3. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

more information for reconstruction. Running for a longer time allows POCS to reconstruct more signal details, high- and low-frequency components. However, from our experiments, the difference in reconstruction is not large enough to deem the extra computational time necessary. This could be the case when using more dimensions. In the SPGL1 experiments, the convergence criteria were set fixed but we performed some preliminary experiments on the residual tolerance and suggest that the residual should be orders of magnitude smaller than the l_2 norm of the available data. Another technical insight that might improve the SPGL1 is the formulation of the problem in equation 2. Collapsing the signal leads to the loss of some location information, and changing this could be useful in the reconstruction.

To evaluate the performance of BPFA, we used two different reconstruction accuracy measures: the FSIM as defined in equation 12 and the quality Q as defined in equation 15. With FSIM, the emphasis is on evaluating the accuracy with respect to the features present in the original and in the reconstruction. Features are sudden changes in the signal with different orientations and magnitude. On the other hand, Q evaluates the reconstruction over the entire signal. Using both measures, BPFA outperforms the rest of the algorithms in general. In the FSIM evaluation, BPFA is by far better due to the fact that it learns basis functions that resemble the features contained in the signals. Because FSIM focuses on obtaining and reconstructing the features, the performance of BPFA in this metric is better.

The data set contains signals with varying structures with different variance, containing high- and low-frequency characteristics. When there are not enough measurements available (i.e., < 60%), the fixed basis functions used by POCS and SPGL1 do not capture all these variations, especially the high frequencies. BPFA is able to adapt the basis functions and is able to capture the high frequencies, the details of the signals, resulting in a higher quality of reconstruction. Nevertheless, it is worth mentioning that the dictionary of basis functions learned by the BPFA is not optimum for the signal at hand. The optimization problem solved is nonconvex with only a local solution obtained. Different initialization provides different basis functions and is thus very sensitive to the starting point. However, the set of basis functions in practice yield a significant increase in reconstruction accuracy.

Due to the fact that the learning of basis functions is done simultaneously with reconstruction/denoising, BPFA uses a distorted version of the seismic signals. However, dropping out or adding noise in the training data is in fact recommended as a regularizer. The percentage of measurements dropped out in the training data is important. In our experiments, we found that dropping out more than 75% of the measurements does not allow enough training data for BPFA to learn basis functions and under-fits. With more measurements available there is a higher quality of reconstruction, but, the basis functions learned are not necessarily the most informative. In Figure 4b, BPFA performs better between 30% and 60%; any basis functions learned between that range could be suitable.

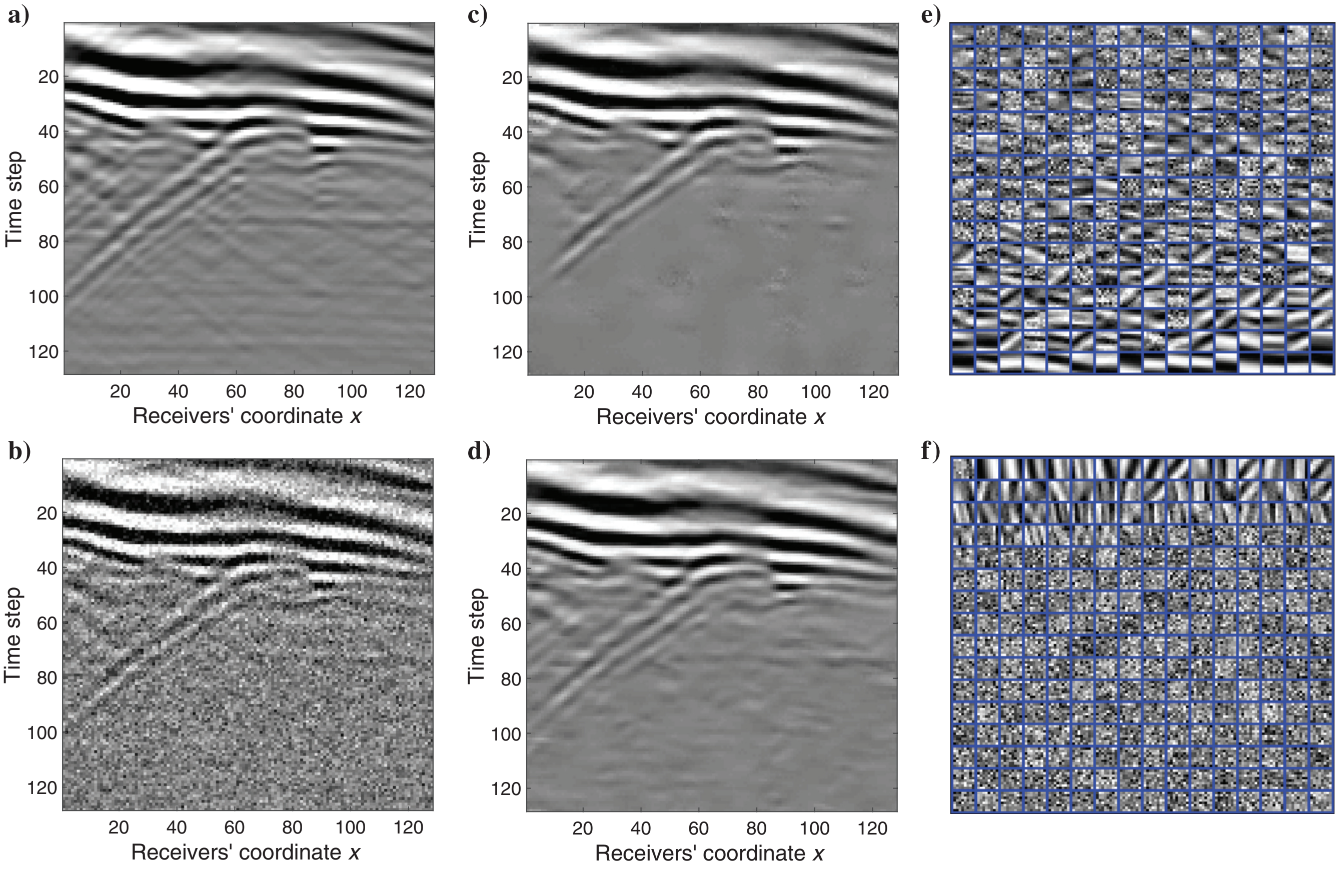

Figure 13. A section of a shot record from the SEAM-II (a) corrupted (S/N = 21.28) (b). BPFA (Q = 24.21 db) (d) obtains higher reconstruction quality than K-SVD (Q = 23.20 db) (c). The learned basis functions of K-SVD (e) and BPFA are also illustrated (f).

Downloaded 07/19/18 to 131.111.184.3. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

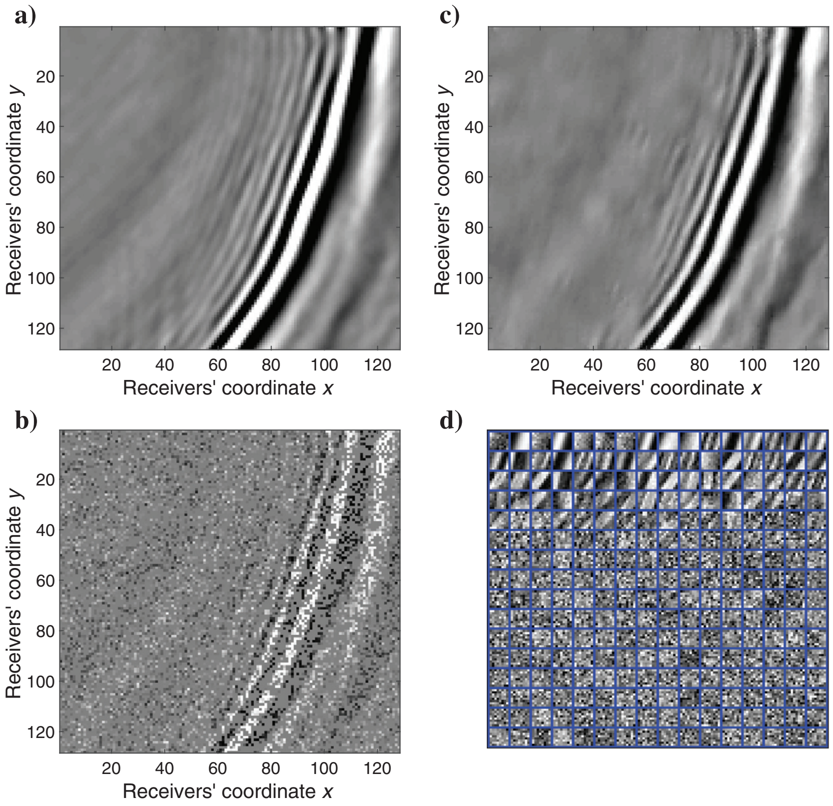

Figure 14. Reconstruction of a section of 128 × 128 of a time slice (a) with noise and missing receivers (b) using the BPFA (c). The BPFA is able to learn basis functions with the presence of noise and missing receiver data (d).

Varying the percentage of receivers used and evaluating the basis functions obtained is an interesting research question.

Experiments in the time slice and in the shot record domain provided similar reconstruction accuracy, with BPFA being better in both domains given enough training data. However, one significant difference is the dictionary of basis functions learned. By using the data in different domains, the signal structure is different, and thus it is necessary to have different sets of basis functions that correspond to this and do not use the assumption that the basis functions suitable in one domain would be suitable in the other. Basis functions learned in the time slice and in the shot record differ in the orientations of the signals' largest variations.

The computational time of the algorithms was also recorded with BPFA being the slowest. Building on this work, a universal dictionary of basis functions could be learned where relearning every time would not be necessary, thus reducing the computational time. Furthermore, informed initialization of BPFA with previously learned basis functions or even initializing BPFA with other analytic sparse transforms (Fourier, Radon, and curvelets) could allow for faster convergence. Finally, faster algorithms have been introduced recently that could help speed up the inference (Sertoglu and Paisley, 2015).

CONCLUSION

BPFA is introduced for seismic CS and denoising. CS experiments were undertaken with regular and irregular sampling, using synthetic data provided by BP. Different percentages of receivers were used, and different sampling schemes were carried out to investigate the performance of BPFA for seismic data acquisition. Comparisons with other algorithms in the literature have shown that BPFA provides state-of-the-art reconstruction accuracy. Features via basis functions were learned using the available measurements. Denoising is also possible using BPFA providing much cleaner and more detailed signals than the K-SVD. A combination of CS and denoising was also illustrated.

The importance of learning basis functions from seismic data is growing. BPFA is an excellent example of how the reconstruction accuracy can be improved greatly with learned basis functions rather than by using predefined dictionaries.

ACKNOWLEDGMENTS

We would like to thank BP for allowing the publication of this research. In addition, we would like to thank R. Abma, N. Philip, C. Brookes, M. Foster, and J. Keggin from BP for very useful discussions on seismic data acquisition. Also, we would like to thank M. Tipping, C. Schoenlieb, N. Nikiforakis, and A. O'Neill for several aspects of this paper. We also thank SEAM-II for the model and C. Regone of BP for modeling the seismic data. We acknowledge funding from the Engineering and Physical Sciences Research Council (EPSRC) through an Industrial CASE studentship (1502944).

REFERENCES

Abma, R., 2011, Projection Onto Convex Sets (POCS) software, http://www.freeusp.org/synthetics/POCS_example/, accessed 4 May 2016.

Abma, R., and N. Kabir, 2006, 3D interpolation of irregular data with a POCS algorithm: Geophysics, **71**, no. 6, E91–E97, doi: 10.1190/1.2356088.

Aharon, M., M. Elad, and A. Bruckstein, 2006, K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation: IEEE Transactions of Signal Processing, **54**, 4311–4322, doi: 10.1109/TSP.2006.881199.

Beck, A., and M. Teboulle, 2009, A fast iterative shrinkage-thresholding algorithm for linear inverse problems: SIAM Journal Imaging Sciences, **2**, 183–202, doi: 10.1137/080716542.

Beckouche, S., and J. Ma, 2014, Simultaneous dictionary learning and denoising for seismic data: Geophysics, **79**, no. 3, A27–A31, doi: 10.1190/geo2013-0382.1.

Bengio, Y., A. Courville, and P. Vincent, 2013, Representation learning: A review and new perspectives: IEEE Transactions on Pattern Analysis and Machine Intelligence, **35**, 1798–1828, doi: 10.1109/TPAMI.2013.50.

Candes, E., and T. Tao, 2006, Near-optimal signal recovery from random projections: Universal encoding strategies: IEEE Transactions on Information Theory, **52**, 5406–5425, doi: 10.1109/TIT.2006.885507.

Candes, E. J., and M. B. Wakin, 2008, An introduction to compressive sampling: IEEE Signal Processing Magazine, **25**, 21–30, doi: 10.1109/MSP.2007.914731.

Cao, J., Y. Wang, and B. Wang, 2015, Accelerating seismic interpolation with a gradient projection method based on tight frame property of curvelet: Exploration Geophysics, **46**, 253–260, doi: 10.1071/EG14016.

Donoho, D., 2006, Compressed sensing: IEEE Transactions on Information Theory, **52**, 1289–1306, doi: 10.1109/TIT.2006.871582.

Elad, M., 2006, K-singular value decomposition (SVD) software, http://www.cs.technion.ac.il/elad/software/, accessed 4 May 2016.

Elad, M., and M. Aharon, 2006, Image denoising via sparse and redundant representations over learned dictionaries: IEEE Transactions on Image Processing, **15**, 3736–3745, doi: 10.1109/TIP.2006.881969.

Downloaded 07/19/18 to 131.111.184.3. Redistribution subject to SEG license or copyright; see Terms of Use at http://library.seg.org/

Griffiths, T. L., and Z. Ghahramani, 2011, The Indian buffet process: An introduction and review: Journal of Machine Learning Research, **12**, 1185–1224.

Herrmann, F. J., and G. Hennenfent, 2008, Non-parametric seismic data recovery with curvelet frames: Geophysical Journal International, **173**, 233–248, doi: 10.1111/j.1365-246X.2007.03698.x.

Hinton, G. E., 2002, Training products of experts by minimizing contrastive divergence: Neural Computation, **14**, 1771–1800.

Kazemi, N., E. Bongajum, and M. D. Sacchi, 2016, Surface-consistent sparse multichannel blind deconvolution of seismic signals: IEEE Transactions on Geoscience and Remote Sensing, **54**, 3200–3207, doi: 10.1109/TGRS.2015.2513417.

Kreimer, N., and M. D. Sacchi, 2011, A tensor higher order singular value decomposition (HOSVD) for prestack simultaneous noise reduction and interpolation: 81st Annual International Meeting, SEG, Expanded Abstracts, 3069–3074, doi: 10.1190/1.3627833.

Kutscha, H., and D. J. Verschuur, 2016, The utilization of the double focal transformation for sparse data representation and data reconstruction: Geophysical Prospecting, **64**, 1498–1515, doi: 10.1111/1365-2478.12362.

Natarajan, B. K., 1995, Sparse approximate solutions to linear systems: SIAM Journal on Computing, **24**, 227–234, doi: 10.1137/S0097539792240406.

Nyquist, H., 2002, Certain topics in telegraph transmission theory: Proceedings of the IEEE, **90**, 280–305.

Oristaglio, M., 2012, Seam phase II land seismic challenges: The Leading Edge, **31**, 264–266, doi: 10.1190/1.3694893.

Paisley, J., and L. Carin, 2009, Nonparametric factor analysis with beta process priors: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 777–784, doi: 10.1145/1553374.1553474.

Pérez, D. O., D. R. Velis, and M. D. Sacchi, 2013, High-resolution prestack seismic inversion using a hybrid FISTA least-squares strategy: Geophysics, **78**, no. 5, R185–R195, doi: 10.1190/geo2013-0077.1.

Pilikos, G., and A. C. Faul, 2016, Relevance vector machines with uncertainty measure for seismic Bayesian compressive sensing and survey design: Presented at the 15th IEEE International Conference on Machine Learning and Applications, doi: 10.1109/ICMLA.2016.0166.

Rifai, S., P. Vincent, X. Muller, X. Glorot, and Y. Bengio, 2011, Contractive auto-encoders: Explicit invariance during feature extraction: Proceedings of the 28 International Conference on Machine Learning.

Sacchi, M., T. Ulrych, and C. Walker, 1998, Interpolation and extrapolation using a high-resolution discrete Fourier transform: IEEE Transactions on Signal Processing, **46**, 31–38, doi: 10.1109/78.651165.

Sertoglu, S., and J. Paisley, 2015, Scalable Bayesian nonparametric dictionary learning: 23rd European Signal Processing Conference (EUSIPCO), 2771–2775, doi: 10.1109/EUSIPCO.2015.7362889.

Shen, H., X. Li, Q. Cheng, C. Zeng, G. Yang, H. Li, and L. Zhang, 2015, Missing information reconstruction of remote sensing data: A technical review: IEEE Geoscience and Remote Sensing Magazine, **3**, 61–85, doi: 10.1109/MGRS.2015.2441912.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 2014, Dropout: A simple way to prevent neural networks from overfitting: Journal of Machine Learning Research, **15**, 1929–1958.

Stanton, A., N. Kreimer, D. Bonar, M. Naghizadeh, and M. Sacchi, 2012, A comparison of 5D reconstruction methods: 82nd Annual International Meeting, SEG, Expanded Abstracts, doi: 10.1190/segam2012-0269.1.

Stanton, A., M. D. Sacchi, R. Abma, and J. A. Stein, 2015, Mitigating artifacts in Projection Onto Convex Sets interpolation: 85th Annual International Meeting, SEG, Expanded Abstracts, 3779–3783, doi: 10.1190/segam2015-5754691.1.

Tipping, M. E., 2001, Sparse Bayesian learning and the relevance vector machine: Journal of Machine Learning Research, **1**, 211–244.

Tipping, M. E., and A. Faul, 2003, Fast marginal likelihood maximisation for sparse Bayesian models: Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics, 3–6.

Trad, D. O., T. J. Ulrych, and M. D. Sacchi, 2002, Accurate interpolation with high-resolution time-variant Radon transforms: Geophysics, **67**, 644–656, doi: 10.1190/1.1468626.

Turquais, P., E. G. Asgedom, W. Sllner, and E. Otnes, 2015, Dictionary learning for signal-to-noise ratio enhancement: 85th Annual International Meeting, SEG, Expanded Abstracts, 4698–4702, doi: 10.1190/segam2015-5846080.1.

van den Berg, E., and M. P. Friedlander, 2007, SPGL1: A solver for large-scale sparse reconstruction, http://www.cs.ubc.ca/labs/scl/spgl1, accessed 4 May 2016.

van den Berg, E., and M. P. Friedlander, 2009, Probing the pareto frontier for basis pursuit solutions: SIAM Journal on Scientific Computing, **31**, 890–912, doi: 10.1137/080714488.

Vincent, P., H. Larochelle, Y. Bengio, and P.-A. Manzagol, 2008, Extracting and composing robust features with denoising autoencoders: Proceedings of the 25th International Conference on Machine Learning, ACM, 1096–1103.

Zhang, L., D. Zhang, X. Mou, and D. Zhang, 2011, FSIM: A feature similarity index for image quality assessment: IEEE Transactions on Image Processing, **20**, 2378–2386, doi: 10.1109/TIP.2011.2109730.

Zhou, M., 2012, Beta process factor analysis software, http://mingyuanzhou.github.io/Code.html, accessed 4 May 2016.

Zhou, M., H. Chen, J. W. Paisley, L. Ren, L. Li, Z. Xing, D. B. Dunson, G. Sapiro, and L. Carin, 2012, Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images: IEEE Transactions Image Processing, **21**, 130–144, doi: 10.1109/TIP.2011.2160072.

Zhou, M., H. Chen, J. W. Paisley, L. Ren, G. Sapiro, and L. Carin, 2009, Nonparametric Bayesian dictionary learning for sparse image representations: Advances in Neural Information Processing Systems, **22**, 2295–2303.

Zhu, L., E. Liu, and J. H. McClellan, 2015, Seismic data denoising through multiscale and sparsity-promoting dictionary learning: Geophysics, **80**, no. 6, WD45–WD57, doi: 10.1190/geo2015-0047.1.

Zwartjes, P. M., and M. D. Sacchi, 2007, Fourier reconstruction of nonuniformly sampled, aliased seismic data: Geophysics, **72**, no. 1, V21–V32, doi: 10.1190/1.2399442.