

## Article

# Seismic Signal Compression Using Nonparametric Bayesian Dictionary Learning via Clustering

Xin Tian \* and Song Li

Electronic Information School, Wuhan University, Wuhan 430072, China; ls@whu.edu.cn

\* Correspondence: xin.tian@whu.edu.cn

Academic Editor: Girija Chetty

Received: 28 March 2017; Accepted: 31 May 2017; Published: 7 June 2017

**Abstract:** We introduce a seismic signal compression method based on nonparametric Bayesian dictionary learning method via clustering. The seismic data is compressed patch by patch, and the dictionary is learned online. Clustering is introduced for dictionary learning. A set of dictionaries could be generated, and each dictionary is used for one cluster's sparse coding. In this way, the signals in one cluster could be well represented by their corresponding dictionaries. A nonparametric Bayesian dictionary learning method is used to learn the dictionaries, which naturally infers an appropriate dictionary size for each cluster. A uniform quantizer and an adaptive arithmetic coding algorithm are adopted to code the sparse coefficients. With comparisons to other state-of-the-art approaches, the effectiveness of the proposed method could be validated in the experiments.

**Keywords:** seismic signal compression; nonparametric bayesian dictionary learning; clustering; sparse representation

## 1. Introduction

Oil companies are increasing their investment in seismic exploration, as oil is an indispensable resource for economic development. A large number of sensors (typically more than 10,000) are required to collect the seismic signals, which are generated by an active excitation source. Similar to other distributed vibration data collection methods [1,2], using cable for data transmission in seismic signal acquisitions is a typical approach. To improve the quality of depth images and simplify acquisition logistics, replacing cabling with wireless technology should be a new trend in seismic exploration. Benefitting from recent advances in wireless sensors, large areas could be measured with a dense arrangement of thousands of seismic sensors. This will create high quality of depth images with a large amount of data to be collected daily. However, the network throughput for single sensor is always limited (e.g., 150 kbps down to 50 kbps). Therefore, it is necessary to compress the seismic signals before transmission. How to represent the seismic signals efficiently with a transform or a basis set could be quite important for seismic signal compression. A lossy compression gain of approximately three has been achieved for the compression of seismic signals using the discrete cosine transform (DCT) [3]. To preserve the important features, a two-dimensional seismic-adaptive DCT is proposed [4]. However, complicated signals could not be well represented by the orthogonal basis used in the methods above. Multiscale geometric analysis methods (e.g., Rigelet, Contourlet and Curvelet [5,6]) have been favored in recent years. By neglecting the orthogonality and completeness, complicated signals could be well represented by inducing a lot of redundancy component. Thereby, the representation of the signals is sparse. For example, Curvelet is adopted in seismic interpretation by exploring the directional features of the seismic data [5]. Another application of Curvelet in seismic signal processing is seismic signal denoising [6]. From a small subset of randomly selected seismic traces, seismic signals are recovered, while noise could be efficiently reduced from the migration amplitudes with the help of sparsity.

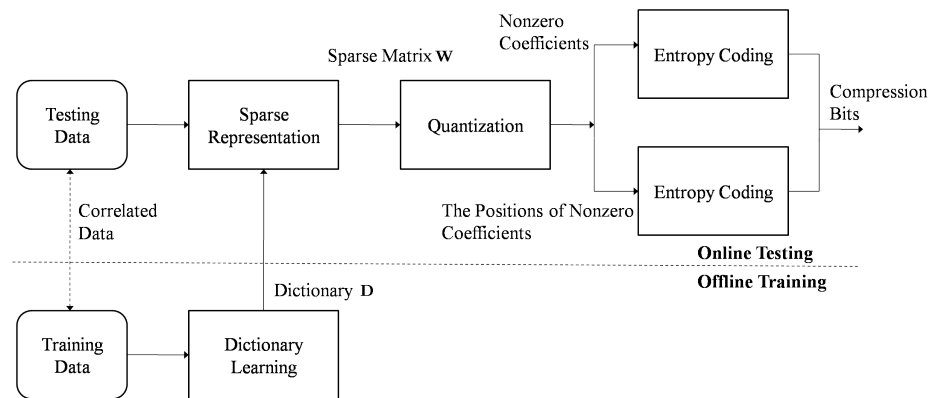
In the above methods, DCT or Curvelet is used to represent the signals, and could be deemed a set of fixed basis. Recent research efforts have been dedicated to learning a set of adaptive basis, called a dictionary. Hence, the signals could be sparsely represented. The dimensionality of the representation space could be higher than the input space. Moreover, the dictionary is inferred from signals themselves. These two properties lead to an improvement in the sparsity of the representation. Therefore, sparse dictionary learning could be widely used in the fields of compression—especially in image compression applications. An image compression scheme using a recursive least squares dictionary learning algorithm in the 9/7 wavelet domain is proposed [7]. This method is called as a compression method based on Offline Dictionary Learning (OffDL) in this paper. It achieves a better performance than using dictionaries learned by other methods. Reference [8] presents a boosted dictionary learning framework. In this work, an ensemble of complementary specialized dictionaries are constructed for sparse image compression. A compression scheme using dictionary learning and universal trellis coded quantization for complex synthetic aperture radar (SAR) images is proposed in [9], which achieves superior quality of decoded images compared with JPEG, JPEG2000, and CCSDS standards. In the above methods, offline training data is necessary for learning the dictionary for the sparse representation of the online testing data. Thus, its compression performance depends on the correlation between the offline training data and the online testing data. Reference [10] proposes an input-adaptive compression approach. Each input image is coded with a learned dictionary by itself. In this way, both the adaptivity and generality are achieved. An online learning-based intra-frame video coding approach is proposed to exploit the texture sparsity of natural images [11]. This is denoted as a compression method based on Dictionary Learning by Online Dictionary transmitting (DLOD). In this method, to synchronize the dictionary used in the coder and decoder, the residual between the current dictionary and the previous one is necessary for sending. This will increase the rates.

In this paper, we focus on how to compress seismic signals with the nonparametric Bayesian dictionary learning method via clustering. Seismic signals are generated from a seismic wave and recorded by different sensors, and are highly correlated. Clustering is introduced for seismic signal compression based on dictionary learning. A set of dictionaries can be generated, and each dictionary is used for one cluster's sparse coding. In this way, the signals in one cluster can be well-represented by their corresponding dictionaries. The dictionaries are learned by the nonparametric Bayesian dictionary learning method, which naturally infers an appropriate dictionary size for each cluster. Furthermore, the online seismic signals are utilized to train the dictionaries. Thus, the correlation of training data and testing data could be relatively high. A uniform quantizer and an adaptive arithmetic coding algorithm are used to code the sparse coefficients. Experimental results demonstrate better rate-distortion performance over other seismic signal compression schemes, which validates the effectiveness of the proposed method. The rest of this paper is organized as follows: in Section 2, we introduce a seismic signal compression method based on offline dictionary learning. A seismic signal compression method using nonparametric bayesian dictionary learning via clustering is introduced in Section 3. Experimental results are presented in Section 4, and conclusion is made in Section 5.

## 2. Seismic Signal Compression Based on Offline Dictionary Learning

In this section, we will introduce how to compress seismic signals in the offline dictionary learning approach (OffDL) [7]. Compared with a fixed basis set, learning an adaptive basis set to a specific set of signals could result in better performance. Supposing input signals  $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1,\dots,K} \in \mathbb{R}^{M \times K}$ , the dictionary  $\mathbf{D} \in \mathbb{R}^{M \times N}$ , and the sparse vector  $\mathbf{W} = \{\mathbf{w}_i\}_{i=1,\dots,K} \in \mathbb{R}^{N \times K}$ , then  $\mathbf{y}_i$  could be represented by a linear combination of the basis from the dictionary  $\mathbf{D}$ :  $\mathbf{y}_i = \mathbf{D}\mathbf{w}_i + \epsilon_i$ .  $\epsilon_i$  could be seen as the noise from the deviation of the linear model. To synchronize the dictionary in the coder and decoder, it is possible to use a pre-learned dictionary by an offline training data. Inspired by this idea, the diagram of the seismic signal compression method based on the offline dictionary learning is shown in Figure 1. It includes two steps: offline training and online testing. In the offline training step,

offline training data is used to train the dictionary  $\mathbf{D}$ . In the online testing step, the input testing data is sparsely represented by the trained dictionary  $\mathbf{D}$ , and the result is a sparse matrix  $\mathbf{W}$ . The sparse matrix is quantized and separated into the nonzero coefficients and their positions. The positions could be seen as a binary matrix, where 0 and 1 denote the coefficients of the current position's zero and nonzero value separately. Finally, entropy coding algorithms are used to code them.



**Figure 1.** Diagram of seismic signal compression based on offline dictionary learning.

To optimize the dictionary  $\mathbf{D}$  and sparse matrix  $\mathbf{W}$ , sparsity could be used as the regulation term, then the two variables  $\mathbf{D}$  and  $\mathbf{W}$  could be solved by two alternating stages: (1) Sparse representation—for a fixed dictionary  $\mathbf{D}$ ,  $\mathbf{w}_i$  can be solved by some sparse representation algorithms, such as order recursive matching pursuit (ORMP) [12] and partial search (PS) [13]; (2) Dictionary updating—when  $\mathbf{w}_i$  is fixed, the dictionary can be updated by methods such as the method of optimized directions (MOD) [14] and K-SVD [15]. The tree-structured iteration-tuned and aligned dictionary (TSITD) was proposed in [16]. TSITD can outperform other image compression algorithms in compression images belonging to specific classes. A classification and update step are repeated to train the dictionary in TSITD. Nevertheless, it is difficult to determine the number of classes and dictionary size in each iteration. Utilizing nonparametric Bayesian methods like the beta process [17], we can infer the number of dictionary elements needed to fit the data. This could reduce the size of the binary matrix generated from the quantized sparse matrix, which is beneficial for compression. To yield posterior distributions rather than point estimation for the dictionary and signals, the nonparametric Bayesian dictionary learning model based on Beta Process Factor Analysis (BPFA) [18] demonstrates its efficiency both in inferring a suitable dictionary size and sparse representation.

### 3. Seismic Signal Compression Using Nonparametric Bayesian Dictionary Learning via Clustering

The performance of the seismic signal compression method based on offline dictionary learning highly depends on the correlation between the offline training data and online testing data. However, the correlation is not always high. In this section, we will introduce a seismic signal compression method using nonparametric Bayesian dictionary learning via clustering. In a typical seismic survey, seismic waves are usually generated by special vibrators mounted on trucks. The seismic waves are reflected by subsurface formations, and return to the surface, where they are recorded by seismic sensors. The trucks are always moved to different locations, where different shots are generated by the vibrators. Obviously, seismic signals from the same seismic wave are highly correlated. If these correlated signals are clustered into the same group and each group learns its dictionary, the representation of these signals could be very sparse. This is useful for compression. Furthermore, we use the online seismic signals to train the dictionaries, which are updated synchronously in the coder and decoder. Meanwhile, the correlation of the online training data and testing data is high. Similar to other compression methods, it is typical to divide seismic signals into small patches for transmission.

The seismic signals recorded by one sensor corresponding to a single shot are always called a trace. In our method, traces of each patch are divided into small segments, which are placed as columns for dictionary learning. Suppose seismic signals of current patch are denoted as  $\mathbf{X}_P = [\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_N]$ , and seismic signals of previous  $L$  patches are denoted as  $\mathbf{X}_{P-L}$  to  $\mathbf{X}_{P-1}$ .  $\mathbf{x}_i$  is the  $i^{th}$  segment in  $\mathbf{X}_P$ , the dimension of which is  $M \times 1$ . As seismic signals of the previous  $L$  patches are transmitted to the decoder and the seismic signals from adjacent patches are highly correlated, we could use these signals (both existing in the coder and decoder) to learn the dictionaries for the sparse representation of seismic signals from the current patch. The diagram of the proposed method is shown in Figure 2. As mentioned above, clustering is introduced for dictionary learning. A set of dictionaries is generated. Seismic signals of the current patch are sparsely represented according to their cluster's dictionaries. This includes the following steps: (1) Online training—transmitted seismic signals are used to learn the parameters of clustering and the dictionary of each cluster; (2) Online testing—firstly, seismic signals are clustered with the parameters generated from the online training step; secondly, they are sparsely represented by the corresponding dictionary of their clusters. Furthermore, the sparse coefficients are quantized and coded for transmission.

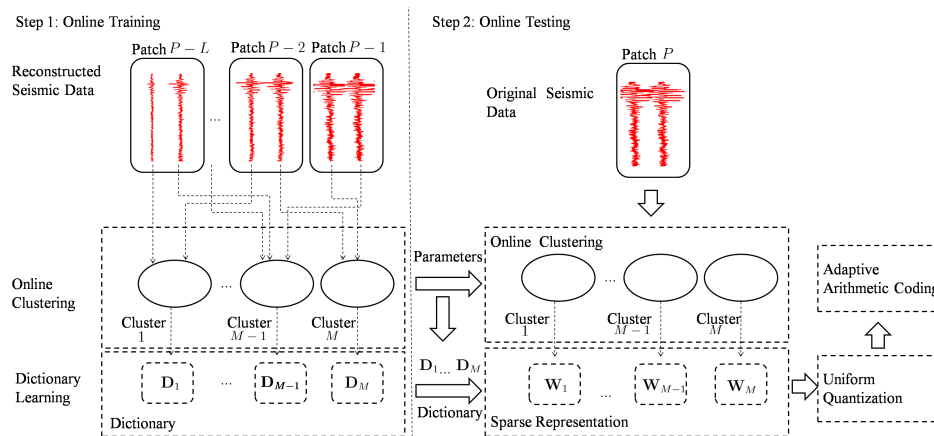


Figure 2. Diagram of proposed method.

### 3.1. Online Training

The mixed-membership naive Bayes model (MMNB) [19] and BPFA are utilized to learn a set of dictionaries based on clustering, the graphical representation of which is shown in Figure 3.

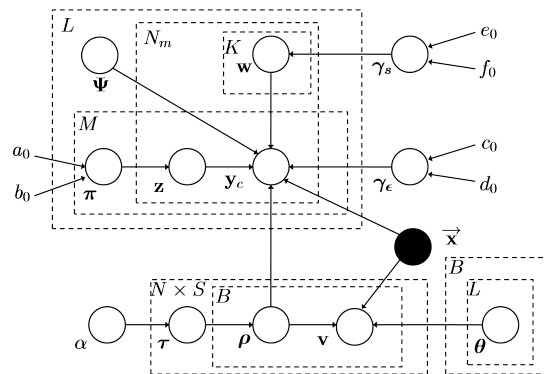


Figure 3. Graphical representation of proposed model.

Suppose that the transmitted seismic signals of the previous  $L$  patches are denoted as  $[\vec{\mathbf{X}}_{P-L}, \dots, \vec{\mathbf{X}}_{P-1}] = [\vec{\mathbf{x}}_1, \dots, \vec{\mathbf{x}}_i, \dots, \vec{\mathbf{x}}_{N \times L}]$ . The online training algorithm includes the following steps:

(1) Feature dimension reduction by principal component analysis (PCA)

To reduce the feature dimension and computation time, PCA is used to generate the feature  $\mathbf{v}_i$  from  $\vec{\mathbf{x}}_i$  as follows:

$$\mathbf{v}_i = R^T \vec{\mathbf{x}}_i \quad (1)$$

where the columns of matrix  $R \in \mathbb{R}^{M \times B}$  form an orthogonal basis. It maps  $\vec{\mathbf{x}}_i$  from an original space of  $M$  variables to a new space of  $B$  variables.

(2) Clustering via MMNB

The latent structure from the observed data could be well discovered by probabilistic mixture model—especially the mixture models. Therefore, MMNB is used for clustering with a Gaussian mixture model as

$$p(\mathbf{v}_i | \boldsymbol{\tau}, \boldsymbol{\theta}) = \sum_{c=1}^V p(\rho = c | \boldsymbol{\tau}) \prod_{j=1}^B p(\mathbf{v}_i | \boldsymbol{\theta}_{jc}) \quad (2)$$

$\boldsymbol{\tau}$  is a prior of the discrete distribution for clusters, and  $\boldsymbol{\theta}_{jc}$  is the parameters of the Gaussian model for features in cluster  $c$ . Firstly, we suppose the number of clusters could be a relatively large value as  $V$ . After clustering, small clusters will be merged for the requirement of dictionary learning.

The parameters  $\alpha$  (parameter of  $\boldsymbol{\tau}$ ) and  $\boldsymbol{\theta}$  are estimated in MMNB by using expectation maximization (EM) algorithms with the following two steps alternating:

(a) E step:

For each data point  $\mathbf{v}_i$ , find the optimal parameters

$$[\gamma_i^{(t)}, \boldsymbol{\phi}_i^{(t)}] = \arg \max_{\gamma_i, \boldsymbol{\phi}_i} L(\gamma_i, \boldsymbol{\phi}_i; \boldsymbol{\theta}_{jc}^{(t-1)}, \alpha^{(t-1)}, \mathbf{v}_i) \quad (3)$$

$\gamma_i$  is a Dirichlet distribution parameter, and  $\boldsymbol{\phi}_i$  is a parameter for discrete distributions of the latent components  $\rho$ .

(b) M step:

$\boldsymbol{\theta}_{jc}$  and  $\alpha$  can be updated as follows:

$$[\boldsymbol{\theta}_{jc}^{(t)}, \alpha^{(t)}] = \arg \max_{\boldsymbol{\theta}_{jc}, \alpha} L(\boldsymbol{\theta}_{jc}, \alpha; \gamma_i^{(t)}, \boldsymbol{\phi}_i^{(t)}, \mathbf{v}_i) \quad (4)$$

(3) Dictionary learning via BPFA

Small clusters are merged into other clusters to keep the number of training data large enough for dictionary learning. The following cluster merging algorithm is used (Algorithm 1).

$\mathbf{Y}_c = [\mathbf{y}_{c1}, \dots, \mathbf{y}_{cH}]$  represents the segments in cluster  $c$ , which are clustered from  $\vec{\mathbf{X}}$ . The number of clusters (denoted as  $N_m$ ) can be smaller than  $V$ . For each cluster, BPFA is an efficient method to learn a dictionary, which naturally infers a suitable dictionary size. It can be described as

$$\begin{aligned}
\mathbf{y}_i &= \mathbf{D}\mathbf{w}_i + \boldsymbol{\epsilon}_i, \quad \mathbf{w}_i = \mathbf{z}_i \odot \mathbf{s}_i \\
\mathbf{d}_k &\sim \mathcal{N}(0, P^{-1}\mathbf{I}_P), \quad \mathbf{s}_i \sim \mathcal{N}(0, \gamma_s^{-1}\mathbf{I}_K) \\
\boldsymbol{\epsilon}_i &\sim \mathcal{N}(0, \gamma_\epsilon^{-1}\mathbf{I}_P), \quad \pi_k \sim \text{Beta}\left(\frac{a_0}{K}, \frac{b_0(K-1)}{K}\right) \\
\gamma_s &\sim \text{Gamma}(c_0, d_0), \quad \gamma_\epsilon \sim \text{Gamma}(e_0, f_0) \\
\mathbf{z}_i &\sim \prod_{k=1}^K \text{Bernoulli}(\pi_k), \quad \boldsymbol{\pi} \sim \prod_{k=1}^K \text{Beta}(a_0/K, b_0(K-1)/K)
\end{aligned} \tag{5}$$

In Equation (5),  $\mathbf{w}_i$  is a sparse vector,  $\mathbf{a} \odot$  is the dot product, and  $\mathbf{I}_P(\mathbf{I}_K)$  is an identity matrix.  $a_0, b_0, c_0, d_0, e_0$ , and  $f_0$  are hyperparameters. As the variables in the above model are from the conjugate exponential function, a variational Bayesian or Markov chain Monte Carlo methods [20] like Gibbs sampling could be used for inference.

---

**Algorithm 1:** Cluster Merging

---

**Input:**  $\boldsymbol{\phi}_i$  (computed by Equation (3)),  $J$  (the required minimum number of segments for each cluster),  $V$  (the initial number of clusters),  $N \times L$  (the number of segments),  $N_k = 0, \Omega = \emptyset$

**for**  $i \leftarrow 1$  **to**  $N \times L$  **do**

cluster the  $i^{\text{th}}$  segment into cluster  $C_i$  by  $C_i = \max_j(\phi_{ij})$ ;  
 $\phi_{ij}$  represents the  $j^{\text{th}}$  element of the column vector  $\boldsymbol{\phi}_i$ ;

**end**

**while**  $N_k < J$  **do**

**for**  $c \leftarrow 1$  **to**  $V$  **do**

find the smallest cluster  $k$ , the number of segments in which is  $N_k (N_k \neq 0)$ ;  
 $\Omega \triangleq \Omega \cup k$ ;

**end**

**for**  $i \leftarrow 1$  **to**  $N \times L$  **do**

**if**  $C_i == k$  **then**

merge the segments of cluster  $k$  into new cluster by  $C_i = \max_j(\phi_{ij}), j \notin \Omega$ .

**end**

**end**

**end**

**Output:**  $C_i$

---

### 3.2. Online Testing

#### (1) Online clustering and sparse representation

Online clustering for seismic signals of the current patch  $\mathbf{X}_P$  could be seen as the E step Equation (3) with the parameters  $\alpha$  and  $\boldsymbol{\theta}$  generated by Algorithm 2. Sparse representation for segments of each cluster could also be solved from Equation (5) when the cluster's dictionary  $\mathbf{D}$  is given. In this way, Gibbs sampling can be used for sparse representation.

#### (2) Quantization and entropy coding

To transmit the sparse coefficients, a uniform quantizer is applied with a fixed quantization step  $\Delta$ . Moreover, an adaptive arithmetic coding algorithm [21] for mixture models is used to code the quantized coefficients. Let  $[\mathbf{W}_{P-L}, \mathbf{W}_{P-1}] = [\mathbf{W}^1, \dots, \mathbf{W}^c, \dots, \mathbf{W}^{N_m}]$  be the quantized coefficients of the previous  $L$  patches, which are separate from  $N_m$  clusters. We suppose that each nonzero coefficient belongs to an alphabet  $\mathcal{A} \in [-2^{Num-1}, \dots, -1, 1, \dots, 2^{Num-1}]$ . A mixture of  $N_m$  probability distributions  $\{f_{\mathbf{W}^c}\}_{c=1}^{N_m}$  could be seen as a combination of  $N_m$  probability density function  $f_{\mathbf{W}^c}$ . Therefore, the probability of the quantized coefficients with the value  $k$  can be written as follows:

$$f_{\mathbf{W}}(k) = \sum_{c=1}^{N_m} p(\rho = c) f_{\mathbf{W}^c}(k) \tag{6}$$

where  $f_{\mathbf{W}^c}(k) = \frac{\beta^c q^c(k)+1}{\beta^c h+2^{N_{um}}}$ .  $q^c(\cdot)$  is the frequency count table of the cluster  $c$ , and  $h$  is the number of nonzero coefficients.  $\beta^c$  is a positive integer, which can be optimized by an EM algorithm. The quantized coefficients  $\mathbf{W}_P$  of the current patch can be coded with the adaptive arithmetic coding algorithm based on the probability Equation (6) existing in the coder and decoder. We also send the nonzero coefficients' positions and their cluster number, which are coded by an arithmetic coding algorithm.

The online training algorithm based on mixed-membership naive Bayes model (MMNB) and Beta Process Factor Analysis (BPFA) is described as Algorithm 2.

---

**Algorithm 2:** Online Training Algorithm Based on MMNB and BPFA

---

**Input:**  $\vec{\mathbf{x}}_i$  (input seismic signals),  $N_m$  (the number of clusters),  $N \times L$  (the number of segments),  $I$  (the number of iterations),  $a_0, b_0, c_0, d_0, e_0, f_0, \alpha^{(0)}, \theta^{(0)}$

**Initialization:** : Choose  $\tau \sim \text{Dirichlet}(\alpha)$   
 Choose a component  $\rho \sim \text{Discrete}(\pi)$   
 Construct a set of dictionaries as  
 $\mathbf{D}_c^{(0)} = [\mathbf{d}_{c1}^{(0)}, \dots, \mathbf{d}_{ci}^{(0)}, \dots, \mathbf{d}_{ck}^{(0)}]: \mathbf{d}_{ci}^{(0)} \sim \mathcal{N}(0, P^{-1}\mathbf{I}_P), c \in [1, N_m]$   
 Draw the following values:  $\mathbf{s}_{ci}^{(0)}, \epsilon_{ci}^{(0)}, \pi_{ck}^{(0)}, \gamma_{cs}^{(0)}, \gamma_{ce}^{(0)}, \mathbf{z}_{ci}^{(0)}$  as Equation (5)

**for**  $i \leftarrow 1$  **to**  $N \times L$  **do**  
 | Compute  $\mathbf{v}_i$  from  $\vec{\mathbf{x}}_i$ ;  
**end**

**for**  $t \leftarrow 1$  **to**  $I$  **do**  
 | **for**  $i \leftarrow 1$  **to**  $N \times L$  **do**  
 | | Compute  $\gamma_i^{(t)}$  and  $\phi_i^{(t)}$  based on Equation (3);  
 | | Compute  $\theta_{ic}^{(t)}$  and  $\alpha^{(t)}$  based on Equation (4);  
 | **end**  
**end**

Generate  $\mathbf{Y}_c$  based on Algorithm 1;

**for**  $c \leftarrow 1$  **to**  $N_m$  **do**  
 | **for**  $t \leftarrow 1$  **to**  $I$  **do**  
 | | Generate the dictionary  $\mathbf{D}_c^{(t)}$  using BPFA with Gibbs sampling based on Equation (5);  
 | **end**  
**end**

**Output:**  $\mathbf{D}_c = \mathbf{D}_c^{(I)} (c \in [1, N_m]), \alpha = \alpha^{(I)}, \theta = \theta^{(I)}$

---

The online testing algorithm can be summarized as Algorithm 3.

---

**Algorithm 3:** Online Testing Algorithm Based on MMNB and BPFA

---

**Input:**  $\mathbf{X}_P = [\mathbf{x}_1, \dots, \mathbf{x}_N]$ ,  $I$  (iteration number),  $\mathbf{D}_c (c \in [1, N_m]), \alpha, \theta$

**Initialization:** : Draw the following values:  $\mathbf{s}_{ci}^{(0)}, \epsilon_{ci}^{(0)}, \pi_{ck}^{(0)}, \gamma_{cs}^{(0)}, \gamma_{ce}^{(0)}, \mathbf{z}_{ci}^{(0)}$  as Equation (5)

**for**  $i \leftarrow 1$  **to**  $N$  **do**  
 | Compute  $\mathbf{v}_i$  from  $\mathbf{x}_i$ ;  
 | Compute  $\gamma_i$  and  $\phi_i$  using Equation (3);  
**end**

Generate  $\mathbf{Y}_c$  based on Algorithm 1;

**for**  $c \leftarrow 1$  **to**  $N_m$  **do**  
 | **for**  $t \leftarrow 1$  **to**  $I$  **do**  
 | | Compute  $\mathbf{z}_{ci}^{(t)}$  and  $\mathbf{s}_{ci}^{(t)}$  with the given dictionary  $\mathbf{D}_c$  based on Equation (5);  
 | | Compute  $\mathbf{w}_{ci}^{(t)} = \mathbf{z}_{ci}^{(t)} \odot \mathbf{s}_{ci}^{(t)}$ ;  
 | **end**  
**end**

Quantize  $\mathbf{w}_{ci}^{(I)}$  and use the probability model (Equation (6)) to code it, generate the code  $\mathcal{A}$ .

**Output:**  $\mathcal{A}$

---



### 3.3. Performance Analysis

In our algorithm, the coded information includes the value of nonzero coefficients, their position information, and their cluster information. The position information is a binary sequence where 0 indicates a zero and 1 denotes a nonzero value. This binary sequence can be encoded using an adaptive arithmetic coder. We suppose the input data is  $\mathbf{X}_P \in \mathbb{R}^{M \times N}$  from patch  $P$ , including  $N_m$  clusters. The corresponding dictionary of cluster  $c$  is denoted as  $\mathbf{D}_c$ . By using BPFA, the dictionary size of different clusters can be distinct, denoted as  $\mathbf{D}_c \in \mathbb{R}^{M \times I^c}$ . The number of nonzero elements in cluster  $c$  is expressed as  $E_c$ . The total rate  $R$  can be approximately computed as:

$$R = \frac{N \times \log_2(N_m) + \sum_{c=1}^{N_m} g_c + Sum \times \rho}{M \times N} \quad (7)$$

where  $Sum = \sum_{c=1}^{N_m} E_c$ , and  $g_c = N \times I^c \times f(\frac{E_c}{N \times I^c})$  is the entropy of coefficients' positions.  $\rho$  is the entropy of nonzero coefficients. Multiple segments can be combined together, sharing the same cluster information for the reduction of rate.

The reconstruction quality is evaluated by SNR as follows:

$$SNR = 10 \log_{10} \frac{\varepsilon_{\text{signal}}^2}{\varepsilon_{\text{noise}}^2} \quad (8)$$

where  $\varepsilon_{\text{signal}}$  and  $\varepsilon_{\text{noise}}$  are the variances of the signal and the noise, respectively.

## 4. Experimental Results

The seismic data [22] is adopted to validate the performance of proposed method. The Bison 120-Channels sensor is used for seismic data collection. The length of the test area was around 300 m, and the receiver interval was 1 m. It included 72 sensors, and each sensor included 135 traces. For each trace, 1600 time samples were used in our experiments. The seismic signals were divided into small segments for dictionary learning and sparse representation. The dimension of each segment was  $16 \times 1$ . Some test samples (50 traces from one sensor) are shown in Figure 4.

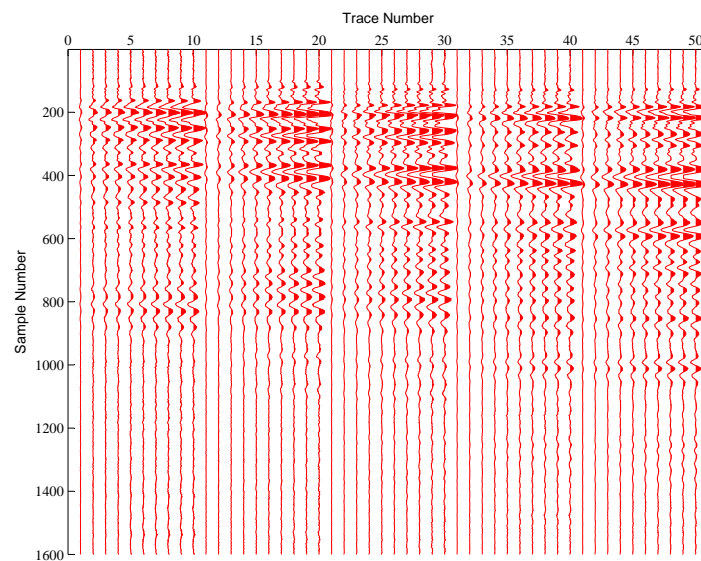


Figure 4. Some test samples.



#### 4.1. Clustering Experiment

Firstly, we carried out the experiment of clustering based on MMNB for seismic signals. A clustering algorithm based on the naive Bayes (NB) model [23] was adopted for comparison. We used *perplexity* [24] as the measurement, which can be given by

$$\text{perplexity} = \exp\left\{-\frac{\sum_{i=1}^n \log p(\mathbf{x}_i)}{\sum_{i=1}^n m_i}\right\}, \quad (9)$$

A log-likelihood  $\log p(\mathbf{x}_i)$  is assigned to each segment  $\mathbf{x}_i$ .  $m_i$  is the number of features extracted from  $\mathbf{x}_i$ , and  $n$  is the number of segments. We used 6 patches as the testing data and 6 previous patches as the training data. The experimental results are shown in Tables 1 and 2. MMNB had a lower (better) perplexity on most of the training and testing data when compared with NB.

**Table 1.** Perplexity of mixed-membership naive Bayes model (MMNB) and naive Bayes (NB) on the training data.

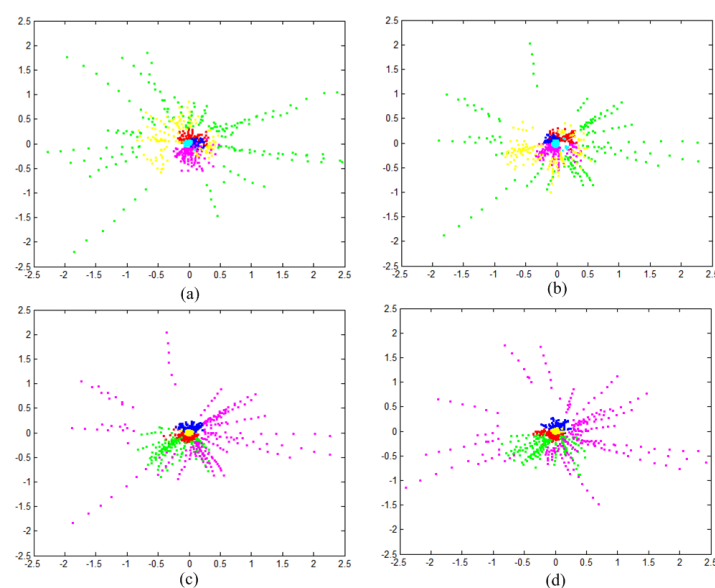
Training Data	1	2	3	4	5	6
NB	$0.198 \pm 0.022$	$0.175 \pm 0.023$	$0.144 \pm 0.015$	$0.182 \pm 0.016$	$0.196 \pm 0.020$	$0.217 \pm 0.023$
MMNB	$0.184 \pm 0.019$	$0.167 \pm 0.018$	$0.143 \pm 0.014$	$0.185 \pm 0.019$	$0.178 \pm 0.018$	$0.195 \pm 0.021$

**Table 2.** Perplexity of MMNB and NB on the testing data.

Training Data	1	2	3	4	5	6
NB	$0.257 \pm 0.035$	$0.242 \pm 0.032$	$0.220 \pm 0.023$	$0.251 \pm 0.027$	$0.268 \pm 0.042$	$0.302 \pm 0.061$
MMNB	$0.232 \pm 0.031$	$0.230 \pm 0.029$	$0.219 \pm 0.021$	$0.251 \pm 0.027$	$0.263 \pm 0.041$	$0.292 \pm 0.052$

#### 4.2. Dictionary Learning Experiment

Figure 5 shows the clustering results after cluster merging based on Algorithm 1. The dimension of data points (feature of segments) is reduced for demonstration, where the data points with the same color belong to the same cluster. Two training and testing data are shown here for demonstration. The initial number of clustering is 8. After merging of the cluster, the actual number of clusters for data 1 (both testing and training) and data 2 are separately 6 and 5.



**Figure 5.** Clustering Results (a) Training Data 1; (b) Testing Data 1; (c) Training Data 2; (d) Testing Data 2.

Secondly, we wanted to validate the efficiency of non-parametric Bayesian dictionary learning based on clustering (NBDLC). Different dictionary learning and sparse representation methods, including K-SVD+OMP, K-SVD+ORMP, K-SVD+PS and TSITD, are compared. The initial dictionary size was  $16 \times 128$  and the size of the test seismic signals in this experiment was  $16 \times 1000$ . In K-SVD+OMP, K-SVD+ORMP, K-SVD+PS, TSITD, and NBDLC, the number of nonzero coefficients is separately controlled by the sparsity and the sparsity prior parameters ( $a_0$  and  $b_0$ ). The experimental result is shown in Figure 6. From Figure 6, NBDLC could have the best reconstruction quality with a similar number of nonzero coefficients compared with other dictionary learning methods. Furthermore, the dictionary sizes are inferred from the seismic signals of each cluster in NBDLC. For example, in our experiments, the minimum dictionary size of NBDLC was  $16 \times 83$ . This is beneficial in reducing the rate of coding for nonzero coefficients' position.

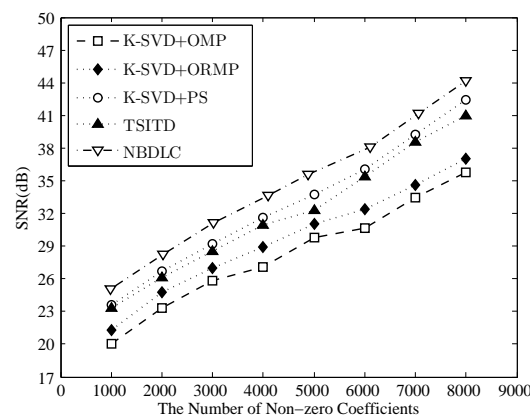


Figure 6. Experimental result of dictionary learning.

#### 4.3. Comparison of Compression Performance

Finally, we carried out a qualitative comparison of seismic signal compression methods based on DCT, Curvelet, OffDL, DLOD and the proposed method on four test data from different sensors. In OffDL, offline data (different from the above four data) is adopted to train the dictionary. K-SVD and PS were used as the dictionary learning and sparse representation methods. A pseudo-random algorithm was used to generate the same random variables in the coder and decoder for the Bayesian dictionary learning process. In DCT and Curvelet, a desired sparsity can be obtained by only maintaining some significant coefficients. For compression, the nonzero coefficients are quantized and coded with an adaptive arithmetic coding algorithm. The quantization step in this experiment is 1024. The experimental results are shown in Figure 7. Curvelet performs better than DCT in most situations—especially in higher rates. OffDL, DLOD and the proposed method outperformed DCT and Curvelet. The compression performance of OffDL highly depends on the correlation between the training data and the testing data. For example, for sensor 3, its performance could be close to DLOD, yet its performance deteriorated for sensor 1. The performance of the proposed method is made better than OffDL and OLOD by the use of clustering. Although the rate will increase by transmission for the information of clustering, the distortion could be efficiently reduced—especially when the rate is high. For example, the rate gain of the proposed method could be approximately 22.4% and 77.8% when SNR is about 28 dB for the seismic signals of sensor 1. Then, we could conclude that the proposed method is an efficient method for seismic signal compression.

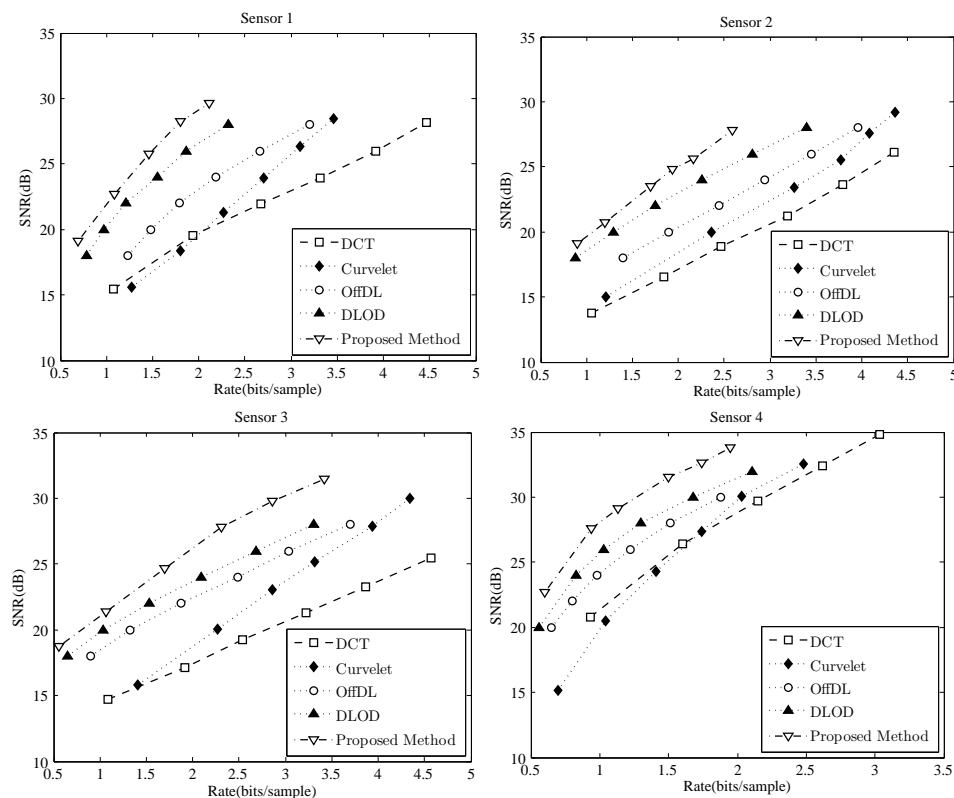


Figure 7. Compression performance comparison of seismic signals from different sensors.

## 5. Conclusions

In this paper, we have shown how to compress seismic signals efficiently by using a clustering-based nonparametric Bayesian dictionary learning method. The previous transmitted data is used to train the dictionary for sparse representation. After clustering by their structural similarities, each cluster could have its own dictionary. Then, the seismic signals of each cluster could be well represented. A nonparametric Bayesian dictionary learning method is used to train the dictionary, which infers an adaptive dictionary size. Experimental results demonstrate better rate-distortion performance over other seismic signal compression schemes, validating the effectiveness of the proposed method.

**Acknowledgments:** This work is supported by the Basic Research (Free Exploration) Project from Commission on Innovation and Technology of Shenzhen, the Nature Science Foundation of China, under grants 61102064, and Key Laboratory of Satellite Mapping Technology and Application, National Administration of Surveying, Mapping and Geoinformation.

**Author Contributions:** All of the authors contributed to the content of this paper. Xin Tian wrote the paper and conceived the experiments. Song Li performed the theoretical analysis.

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

## References

1. Yunusa-Kaltungo, A.; Sinha, J.K.; Nembhard, A.D. Use of composite higher order spectra for faults diagnosis of rotating machines with different foundation flexibilities. *Measurement* **2015**, *70*, 47–61.
2. Yunusa-Kaltungo, A.; Sinha, J.K. Sensitivity analysis of higher order coherent spectra in machine faults diagnosis. *Struct. Health Monit.* **2016**, *15*, 555–567, doi:10.1177/1475921716651394.
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.

4. Aparna, P.; David, S. Adaptive Local Cosine transform for Seismic Image Compression. In Proceedings of the 2006 International Conference on Advanced Computing and Communications, Surathkal, India, 20–23 December 2006.
5. Douma, H.; de Hoop, M. Leading-order Seismic Imaging Using Curvelets. *Geophysics* **2007**, *72*, S231–S248.
6. Herrmann, F.J.; Wang, D.; Hennenfent, G.; Moghaddam, P.P. Curvelet-based seismic data processing: A multiscale and nonlinear approach. *Geophysics* **2008**, *73*, A1–A5, doi:10.1190/1.2799517.
7. Skretting, K.; Engan, K. Image compression using learned dictionaries by RLS-DLA and compared with K-SVD. In Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 22–27 May 2011; pp. 1517–1520.
8. Nejati, M.; Samavi, S.; Karimi, N.; Soroushmehr, S.M.R.; Najaran, K. Boosted multi-scale dictionaries for image compression. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 1130–1134.
9. Zhan, X.; Zhang, R. Complex SAR Image Compression Using Entropy-Constrained Dictionary Learning and Universal Trellis Coded Quantization. *Chin. J. Electron.* **2016**, *25*, 686–691.
10. Horev, I.; Bryt, O.; Rubinstein, R. Adaptive image compression using sparse dictionaries. In Proceedings of the 2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP), Vienna, Austria, 11–13 April 2012; pp. 592–595.
11. Sun, Y.; Xu, M.; Tao, X.; Lu, J. Online Dictionary Learning Based Intra-frame Video Coding. *Wirel. Pers. Commun.* **2014**, *74*, 1281–1295.
12. Adler, J.; Rao, B.D.; Kreutz-Delgado, K. Comparison of basis selection methods. In Proceedings of the Conference Record of the Thirtieth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 3–6 November 1996; pp. 252–257.
13. Skretting, K.; Håkon Husøy, J. *Partial Search Vector Selection for Sparse Signal Representation*; NORSIG-03; Stavanger University College: Stavanger, Norway, 2003.
14. Engan, K.; Aase, S.O.; Håkon Husøy, J. Method of optimal directions for frame design. In Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP99 (Cat. No. 99CH36258), Phoenix, AZ, USA, 15–19 March 1999; pp. 2443–2446.
15. Aharon, M.; Elad, M.; Bruckstein, A. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Trans. Signal Process.* **2006**, *54*, 4311–4322.
16. Zepeda, J.; Guillemot, C.; Kijak, E. Image Compression Using Sparse Representations and the Iteration-Tuned and Aligned Dictionary. *IEEE J. Sel. Top. Signal Process.* **2011**, *5*, 1061–1073.
17. Paisley, J.; Zhou, M.; Sapiro, G.; Carin, L. Nonparametric image interpolation and dictionary learning using spatially-dependent Dirichlet and beta process priors. In Proceedings of the 2010 IEEE International Conference on Image Processing, Hong Kong, China, 26–29 September 2010; pp. 1869–1872.
18. Zhou, M.; Chen, H.; Paisley, J.; Ren, L.; Li, L.; Xing, Z.; Dunson, D.; Sapiro, G.; Carin, L. Nonparametric Bayesian Dictionary Learning for Analysis of Noisy and Incomplete Images. *IEEE Trans. Image Process.* **2012**, *21*, 130–144.
19. Shan, H.; Banerjee, A. Mixed-membership naive Bayes models. *Data Min. Knowl. Discov.* **2011**, *23*, 1–62.
20. Stramer, O. Monte Carlo Statistical Methods. *J. Am. Stat. Assoc.* **2001**, *96*, 339–355, doi:10.1198/jasa.2001.s372.
21. Masmoudi, A.; Masmoudi, A.; Puech, W. An efficient adaptive arithmetic coding for block-based lossless image compression using mixture models. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 5646–5650.
22. UTAM Seismic Data Library. Available online: <http://utam.gg.utah.edu/SeismicData/SeismicData.html> (accessed on 21 May 2009).
23. Santafe, G.; Lozano, J.A.; Larranaga, P. Bayesian Model Averaging of Naive Bayes for Clustering. *IEEE Trans. Syst. Man Cybern. Part B* **2006**, *36*, 1149–1161.
24. Dhillon, I.S.; Mallela, S.; Modha, D.S. Information-theoretic Co-clustering. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03), New York, NY, USA, 24–27 August 2003; pp. 89–98.

