Article

Seismic Signal Compression Using Nonparametric Bayesian Dictionary Learning via Clustering

Xin Tian [1]* and Song Li [1]

[1] Electronic Information School, Wuhan University
* Correspondence: xin.tian@whu.edu.cn

Academic Editor: name
Version May 26, 2017 submitted to Algorithms

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

~~The oil~~ Oil companies are increasing their investment in seismic exploration~~due to the strong fluctuations in crude oil prices~~, as oil is an indispensable resource for economic development. A large number of sensors(~~always more than 10000~~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 ~~sensor~~sensors, large areas could be measured with a dense arrangement of thousands of seismic ~~sensor, which~~ sensors. This will create high quality of depth images with ~~sufficient textures. It will produce~~ a large amount of data to be collected daily~~, but~~ . However, the network throughput for single sensor is always limited ~~, for example~~ (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 ~~set of basis~~ basis set could be quite important for seismic signal compression. A lossy compression gain of ~~approximate~~ approximately three has been achieved for the compression of seismic signals using the ~~Discrete Cosine Transform~~discrete cosine transform (DCT) [3]. To preserve the important features, a ~~two dimensional~~ two-dimensional seismic-adaptive DCT is proposed [4]. However, complicated signals could not be well represented by the orthogonal basis used in ~~above methods~~ the methods above. Multiscale geometric analysis methods (~~such as~~ e.g., Rigielet, Contourlet and Curvelet~~[5,6] are~~ [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 ~~denosing~~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 ~~could be seemed as~~ is used to represent the signals, and could be deemed a set of fixed basis~~, used to represent the signal~~. Recent research ~~effort has~~ efforts have been dedicated to learning a set of adaptive basis, called ~~as a dictionary, for the purpose of sparse representation~~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 ~~signal itself~~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~~ 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) ~~[7]~~in this paper. It achieves a better performance than using dictionaries learned by other methods. [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 ~~codied~~ coded quantization for complex synthetic aperture radar(SAR) images is proposed in [9], which achieves superior quality of ~~decode~~ decoded images compared with JPEG, JPEG2000, and CCSDS standards. In the above methods, ~~an~~ 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. [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~~ 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)~~[11]~~. 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 ~~in an online way~~ with the nonparametric ~~bayesian~~ Bayesian dictionary learning method via clustering. Seismic signals ~~of multiple sensors are highly correlated, especially in a propagation form of a seismic wave~~ 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. ~~We optimize for a~~ A set of dictionaries ~~, one for each cluster, for which the signals from the same seismic wave could be well reconstructed in a sparse representation way. Nonparametric bayesian~~ 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~~is used to learn the dictionaries~~, which naturally infers an appropriate dictionary size for each cluster. Furthermore, the ~~transmitted~~ online seismic signals are utilized to train ~~these dictionaries, which could exists both in the coder and decoder~~the dictionaries. Thus, the ~~dictionary transmission and synchronization problem is well solved~~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, ~~validating~~ which validates the effectiveness of the proposed method. The rest of this paper is organized as follows: In Section ~~2~~2, we introduce a seismic signal compression method based on offline dictionay learning. A seismic signal compression method using nonparametric bayesian dictionary learning via clustering is introduced in Section ~~3.~~ 3. Experimental results are presented in Section ~~4~~4, and conclusion is made in Section ~~5.~~ 5.

2. ~~Seimsic~~ Seismic Signal Compression ~~based~~ Based on Offline Dictionary Learning

In this section, we will introduce how to compress seismic signals in the offline dictionary learning ~~way~~approach (OffDL)~~[7]. As stated above~~ [7]. Compared with a fixed basis set, learning an adaptive ~~set of basis~~ basis set to a specific set of signals could result in better performance~~than a fixed set of basis. Suppose~~. Supposing input signals $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1,...,K} \in \mathbb{R}^{M\times K}$, the dictionary $\mathbf{D} \in \mathbb{R}^{M\times N}$, and the sparse vector $\mathbf{W} = \{\mathbf{w}_i\}_{i=1,...,K} \in \mathbb{R}^{N\times K}$, then $\mathbf{y}_i$ could be ~~well~~ represented by a linear ~~combinations~~ combination of the basis from the dictionary $\mathbf{D}$: $\mathbf{y}_i = \mathbf{D}\mathbf{w}_i + \epsilon_i$. ϵ_i could be ~~seemed~~ seen as the noise from the deviation of the linear model. To synchronize the dictionary ~~used~~ in the coder and decoder, ~~one choice is to send the learned dictionary D. However, this will increase the rates by the transmission of the dictionary. Therefore, it is typical~~ it is possible to use a pre-learned dictionary by an offline training data~~, existing in the coder and decoder. Inspired from~~. Inspired by this idea, the diagram of the seismic signal compression method based on the offline dictionary learning is shown ~~Figure 1.~~ in Figure 1. It includes two steps: offline training and online testing. In the offline training step, ~~an~~ offline training data is ~~adopted~~ used to train the dictionary $\mathbf{D}$. In the online testing step, the input testing data is sparsely represented by the trained dictionary $\mathbf{D}$, ~~the result of which~~ and the result is a sparse matrix $\mathbf{W}$. ~~Furthermore, the sparse matrix is quantized. The quantized~~ The sparse matrix is ~~separately~~ quantized and separated into the nonzero coefficients and their positions. The positions could be ~~seemed~~ seen as a binary matrix, where 0 and 1 denote the coefficients of the current position~~are~~'s zero and nonzero value separately. Finally, ~~they are separately coded by~~ entropy coding algorithms ~~.~~ are used to code them.

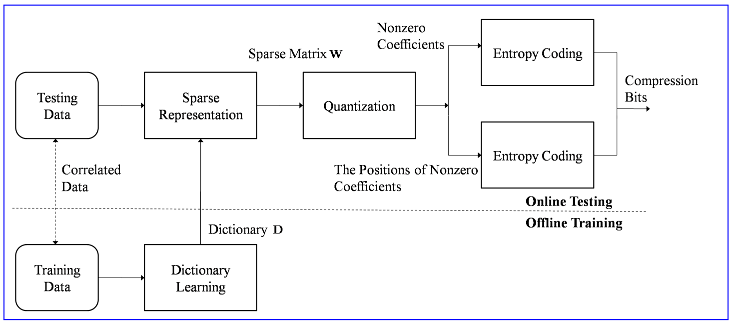


Figure 1. Diagram of seismic signal compression based on offline dictionary learning.

~~To solve above optimization problems in dictionary learning and sparse representation~~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~~ representation—for a fixed dictionary $\mathbf{D}$, $\mathbf{w}_i$ ~~could~~ can be solved by some sparse representation algorithms~~such as: Order Recursive Matching Pursuit~~, such as order recursive matching pursuit (ORMP)~~[12] and Partial Search~~ [12] and partial search (PS) [13]. 2) Dictionary ~~updating. When~~ updating—when $\mathbf{w}_i$ is fixed, the dictionary ~~could~~ can be updated by ~~some methods like Method of Optimized Directions~~methods such as the method of optimized directions (MOD) [14] and K-SVD[15]. The ~~Tree-Structured Iteration-Tuned and Aligned Dictionary~~tree-structured iteration-tuned and aligned dictionary (TSITD) ~~has been proposed [16]. It shows using TSITD for compressing images belonging to specific classes could~~ 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 ~~hard~~ difficult to determine the number of ~~class~~ classes and dictionary size in each iteration~~of TSITD. These methods have demonstrates their efficiency in the applications of denosing, interpolation and so on. However, the size of the dictionary is always set a prior or fixed.~~ . Utilizing nonparametric Bayesian methods like the beta process [17], we ~~could~~ 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, ~~nonparametric bayesian~~ the nonparametric Bayesian dictionary learning model based on ~~beta process~~Beta Process Factor Analysis(BPFA)[18] demonstrates its efficiency both in inferring a suitable dictionary size and sparse representation.

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

~~The drawback of above~~ The performance of the seismic signal compression method ~~is its efficiency highly depending~~ based on offline dictionary learning highly depends on the correlation ~~of the online testing data and~~ between the offline training data and online testing data. However, ~~it is difficult to keep the correlation always high as the seismic wave is changed~~the correlation is not always high. In this section, we will introduce a seismic signal compression method using nonparametric ~~bayesian~~ 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 ~~similar~~highly correlated. If these ~~similar signals could be~~ 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 ~~transmitted~~ online seismic signals to train the dictionaries, ~~thus the dictionaries could be updated synchronously both~~ which are updated synchronously in the coder and decoder. ~~Therefore, we do not need to send these dictionaries, and high~~ Meanwhile, the correlation of the online training data and testing data ~~could be always guaranteed~~is high. Similar to other compression methods, it is ~~common~~ typical to divide seismic signals into small patches for transmission. The seismic signals recorded by one sensor corresponding to a single shot ~~is always called as~~ 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 $\mathbf{X}_P = [\mathbf{x}_1, ..., \mathbf{x}_i, ..., \mathbf{x}_N]$, 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 $M \times 1$. ~~The number of segments for each patch is the same.~~ 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 ~~exists~~ 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.~~ 2. As mentioned above, clustering is introduced for dictionary learning. A set of dictionaries ~~, one for each cluster,~~ is generated. ~~Then seismic signals of current patch could be well reconstructed with these dictionariesin a sparse representation way. It~~ Seismic signals of the current patch are sparsely represented according to their cluster's dictionaries. This includes the following steps: 1) Online ~~training. Transmitted~~ training—transmitted seismic signals are used to learn the parameters of clustering and the dictionary of each cluster. 2) Online ~~testing. Firstly~~testing—firstly, seismic signals are clustered with the parameters ~~learned in the above step. Secondly~~generated from the online training step; secondly, they are ~~sparse~~ sparsely represented by the corresponding dictionary of their ~~cluster. Moreover~~clusters. Furthermore, the sparse coefficients are quantized and coded for transmission. The details are as follows:

3.1. Online ~~training~~Training

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

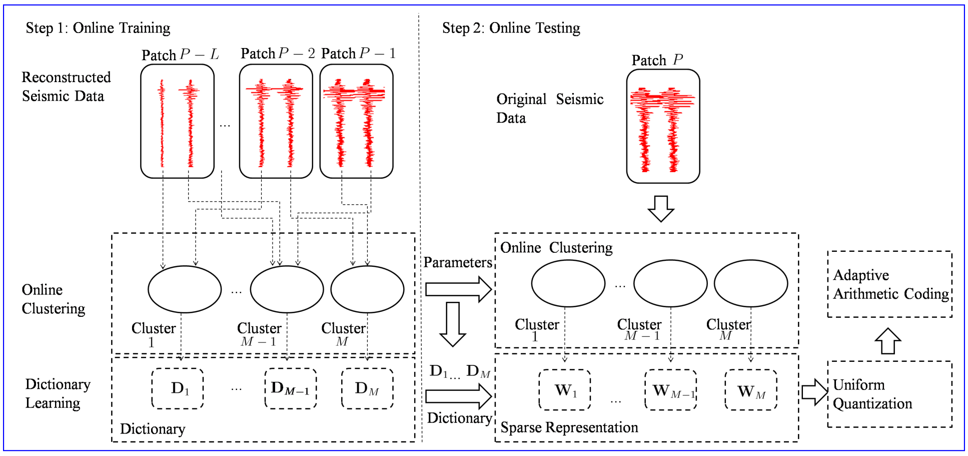
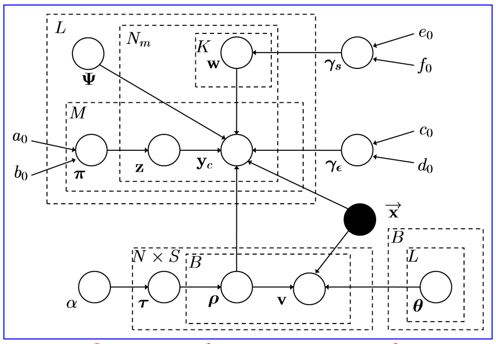


Figure 2. Diagram of proposed method.

3. Suppose that the transmitted seismic signals of the previous L patches are denoted as



~~Suppose the reconstructed~~

Figure 3. Graphical representation of proposed model.

$[\vec{\mathbf{X}}_{P-L}, \ldots, \vec{\mathbf{X}}_{P-1}] = [\vec{\mathbf{x}}_1, ..., \vec{\mathbf{x}}_i, ..., \vec{\mathbf{x}}_{N\times L}]$. The online training algorithm includes the following steps:

1) ~~Reduction of feature dimension by Principal Component Analysis~~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:

$$\mathbf{v}_i = R^T \vec{\mathbf{x}}_i \tag{1}$$

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

~~Probabilistic mixture model is a popular research to latent cluster structure discovery from observed data, especially~~ 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~~ Gaussian mixture model as

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

$\boldsymbol{\tau}$ ~~denotes a discrete distribution as a prior over the clusters. The density of~~ 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~~is a Gaussian with the parameters~~ $\cancel{\boldsymbol{\theta}_{jc} = (\boldsymbol{\mu}_{jc}, \boldsymbol{\delta}_{jc})}$. Firstly, we suppose the number of clusters could be a

~~relative~~ relatively large value as V. After clustering, small clusters will be merged for the requirement of dictionary learning.

~~The learning task in MMNB is to estimate the~~ The parameters α (parameter of $\boldsymbol{\tau}$) and $\boldsymbol{\theta}$ ~~such that the likelihood of observing the whole data set is maximized. A general method for this task is to use Expectation Maximization~~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

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

$\gamma_{\boldsymbol{i}}$ is a Dirichlet distribution parameter~~over $\boldsymbol{\tau}$~~, and $\boldsymbol{\phi}_{\boldsymbol{i}}$ is ~~the~~ a parameter for discrete distributions ~~over~~ of the latent components $\boldsymbol{\rho}$.

b) M step:

~~The model parameters could be estimated~~ $\boldsymbol{\theta}_{jc}$ and α can be updated as follows:

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

~~Especially, $\theta_{jc} = (\mu_{jc}, \delta_{jc})$~~

$$\begin{cases} \mu_{jc} = \dfrac{\sum_{i=1}^{N\times L} \phi_{ijc} v_{ij}}{\sum_{i=1}^{N\times L} \phi_{ijc}} \\ \delta_{jc}^2 = \dfrac{\sum_{i=1}^{N\times L} \phi_{ijc} (v_{ij} - u_{jc})^2}{\sum_{i=1}^{N\times L} \phi_{ijc}} \end{cases}$$

3) Dictionary learning via BPFA

Small clusters are merged into other clusters to keep the number of training data ~~not too small~~ large enough for dictionary learning. The following cluster merging algorithm is used (Algorithm 1).

Algorithm 1: Cluster ~~Merging~~merging

Input: $\boldsymbol{\phi}_{\boldsymbol{i}}$ (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), $N_k = 0, \Omega = \varnothing$

for $i \leftarrow 1$ **to** $N \times L$ **do**
  cluster the i^{th} segment into cluster C_i by ~~$C_i = max_j(\phi_{ij})$;~~ $C_i = max_j(\phi_{ij})$;
  ϕ_{ij} represents the j^{th} element of the column vector $\boldsymbol{\phi}_{\boldsymbol{i}}$~~.~~;
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

$\mathbf{Y}_c = [\mathbf{y}_{c1}, \ldots, \mathbf{y}_{cH}]$ represents the segments in cluster c, which are clustered from $\overrightarrow{\mathbf{X}}$. The number of ~~cluster~~clusters (denoted as N_m) ~~could~~ 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 ~~could~~ can be described as

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

~~The vector $\mathbf{w}_i$ is always sparse, which is enforced by placing a **Beta-Bernoulli** prior on $\mathbf{w}_i$.~~ In Equation (5), ~~variables $\mathbf{z}_i \sim \prod_{k=1}^K \mathbf{Bernoulli}(\pi_k)$ and $\boldsymbol{\pi} \sim \prod_{k=1}^K \mathbf{Beta}(a_0/K, b_0(K-1)/K)$, where π_k is the k^{th} component of $\boldsymbol{\pi}$. $\mathbf{d}_k$ represents the k^{th} column(atom) of $\mathbf{D}$, and~~ $\mathbf{w}_i$ is a sparse vector, a $\odot$ is the ~~elementwise or Hadamard vector product. $\mathbf{I}_P$ and~~ dot product, and $\mathbf{I}_P$ ($\mathbf{I}_K$~~represents a $P \times P$ or $K \times K$~~) is an identity matrix. ~~The constants~~ $a_0, b_0, c_0, d_0, e_0,$ and f_0 are ~~called hyperparameters. Consecutive elements~~ hyperparameters. As the variables in the above ~~hierarchical model are in~~ model are from the conjugate exponential ~~family, and the inference could be implemented by~~ function, a variational Bayesian or Markov chain Monte Carlo methods [20] like Gibbs sampling could be used for inference.

~~Online~~ The online training algorithm based on MMNB and BPFA is described as Algorithm 2.

Algorithm 2: Online ~~Training Algorithm~~ training algorithm based on MMNB and BPFA

Input: $\overrightarrow{\mathbf{x}}_i$ (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, $\alpha^{(0)}$, $\boldsymbol{\theta}^{(0)}$

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

for $i \leftarrow 1$ **to** $N \times L$ **do**
  Compute $\mathbf{v}_i$ from $\overrightarrow{\mathbf{x}}_i$;
end
for $t \leftarrow 1$ **to** I **do**
  for $i \leftarrow 1$ **to** $N \times L$ **do**
    Compute $\gamma_i^{(t)}$ and $\boldsymbol{\phi}_i^{(t)}$ 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 ~~the~~ BPFA with Gibbs sampling based on Equation (5);
  end
end

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

3.2. Online ~~testing~~Testing

1) Online clustering and sparse representation

Online clustering for ~~the~~ seismic signals of the current patch $\mathbf{X}_P$ could be ~~seemed~~ seen as the E step (Equation 3) with the parameters α and $\boldsymbol{\theta}$ generated by Algorithm 2. Sparse representation for

~~the~~ 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 ~~could~~ 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 Δ. Moreover, an adaptive arithmetic coding algorithm [21] for mixture models is used to code the quantized coefficients. Let $[\mathbf{W}_{P-L},, \mathbf{W}_{P-1}] = [\mathbf{W}^1, \ldots, \mathbf{W}^c, \ldots, \mathbf{W}^{N_m}]$ be the quantized coefficients of the previous L patches, which are ~~separately~~ separate from N_m clusters. We suppose that each nonzero coefficient ~~take a value from~~ belongs to an alphabet $\mathcal{A} \in [-2^{Num-1}, \ldots, -1, 1, \ldots, 2^{Num-1}]$~~composed of 2^{Num} symbols~~. A mixture of N_m probability distributions $\{f_{\mathbf{W}^c}\}_{c=1}^{N_m}$ could be ~~seemed~~ 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 ~~could~~ can be written as follows:

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

where $f_{\mathbf{W}^c}(k) = \frac{\beta^c q^c(k)+1}{\beta^c h + 2^{Num}}$. $q^c(\cdot)$ is the frequency count table of the cluster c, and h is the number of nonzero coefficients. β^c is a positive integer, which ~~could~~ can be optimized by ~~a~~ an EM algorithm. ~~Hence, the~~ The quantized coefficients $\mathbf{W}_P$ of ~~current patch could~~ 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.

~~Online testing algorithm could~~ The online testing algorithm can be summarized as Algorithm 3.

Algorithm 3: Online ~~Testing Algorithm~~ testing algorithm based on MMNB and BPFA

Input: $\mathbf{X}_P = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, I (iteration number), $\mathbf{D}_c (c \in [1, N_m])$, α, $\boldsymbol{\theta}$
Initialization: : Draw the following values: $\mathbf{s}_{ci}^{(0)}$, $\boldsymbol{\epsilon}_{ci}^{(0)}$, $\pi_{ck}^{(0)}$, $\gamma_{cs}^{(0)}$, $\gamma_{c\epsilon}^{(0)}$, $\mathbf{z}_{ci}^{(0)}$ as Equation (5)
for $i \leftarrow 1$ **to** N **do**
  Compute $\mathbf{v}_i$ from $\mathbf{x}_i$;
  Compute γ_i and $\boldsymbol{\phi}_i$ 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 ~~give~~ 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 ~~could~~ 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 ~~cluster could be distrinct~~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 ~~could~~ can be approximately computed as:

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

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. ρ is the entropy of nonzero coefficients. Multiple segments ~~could~~ can be combined together, sharing the same cluster information for the reduction of rate.

The reconstruction quality is evaluated by SNR as follows:

$$\mathrm{SNR} = 10\log_{10}\frac{\varepsilon^2_{\mathrm{signal}}}{\varepsilon^2_{\mathrm{noise}}} \tag{8}$$

where $\varepsilon_{\mathrm{signal}}$ and $\varepsilon_{\mathrm{noise}}$ 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~~[22]. It includes~~ . 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 ~~includes~~ included 135 traces. For each trace, 1600 time samples ~~are~~ were used in our experiments. The seismic signals ~~are~~ were divided into small segments for dictionary learning and sparse representation. The dimension of each segment ~~is~~ was 16×1. Some test samples (50 traces from one sensor) are shown in Figure ~~4~~4.

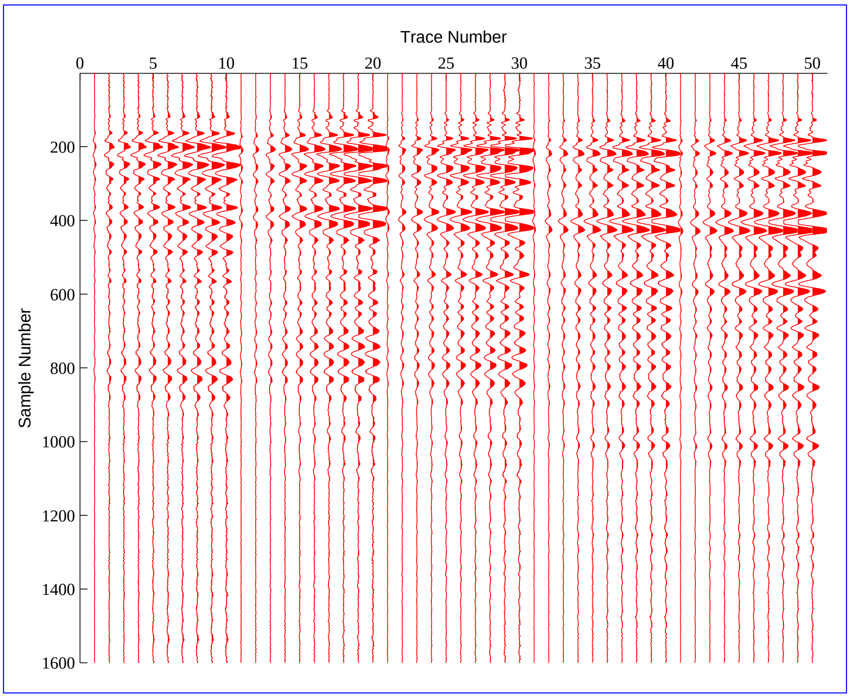


Figure 4. Some test samples.

4.1. ~~Experiment of~~ Clustering Experiment

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

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

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 ~~use~~ used 6 patches as the testing data and 6 previous patches as the training data. The experimental results are shown in Table ~~1 and Table 2. MMNB has~~ 1 and Table 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 NB on the ~~Training Data~~training data.

Training ~~dataset~~data	**1**	**2**	**3**	
NB	~~0.1982 ± 0.022~~ 0.198 ± 0.022	~~0.1754 ± 0.023~~ 0.175 ± 0.023	~~0.1435 ± 0.015~~ 0.144 ± 0.015	~~0.1820 ± 0.0~~
MMNB	~~0.1836 ± 0.019~~ 0.184 ± 0.019	~~0.1667 ± 0.018~~ 0.167 ± 0.018	~~0.1425 ± 0.014~~ 0.143 ± 0.014	~~0.1850 ± 0.0~~

Table 2. Perplexity of MMNB and NB on the ~~Testing Data~~testing data.

Training ~~dataset~~data	**1**	**2**	**3**	
NB	~~0.2574 ± 0.035~~ 0.257 ± 0.035	~~0.2415 ± 0.032~~ 0.242 ± 0.032	~~0.2201 ± 0.023~~ 0.220 ± 0.023	~~0.2514 ± 0.0~~
MMNB	~~0.2316 ± 0.031~~ 0.232 ± 0.031	~~0.2301 ± 0.029~~ 0.230 ± 0.029	~~0.2190 ± 0.021~~ 0.219 ± 0.021	~~0.2512 ± 0.0~~

~~Figure 5~~ 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.

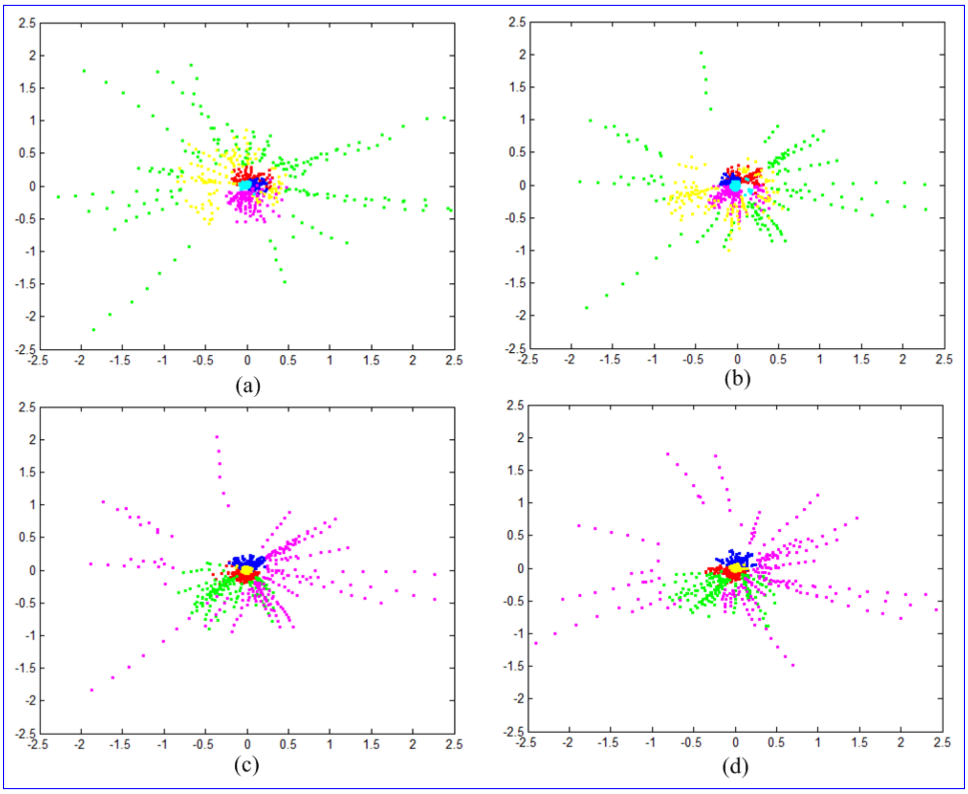


Figure 5. Clustering Results (**a**)Training Data 1 (**b**)Testing Data 1 (**c**)Training Data 2 (**d**)Testing Data 2.

4.2. *Dictionary Learning Experiment*

4.3. ~~*Experiment of Dictionary Learning*~~

Secondly, we ~~want~~ wanted to validate the efficiency of ~~Non-parametric Bayesian Dictionary Learning based on Clustering~~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 ~~is~~ was 16×128 and the size of the test seismic signals in this experiment ~~is~~ was 16×1000. 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~~6. From Figure ~~6~~6, NBDLC could have the best reconstruction quality with ~~the~~ a similar number of nonzero coefficients ~~while~~ 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 ~~is~~ was 16×83. This is beneficial in reducing the rate of coding for nonzero coefficients' position.

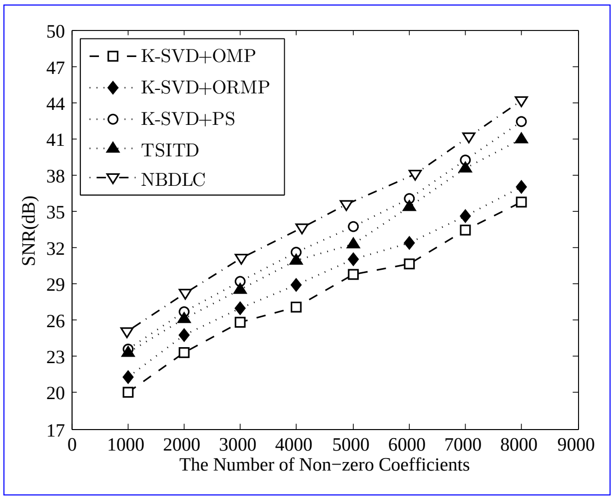


Figure 6. Experimental result of dictionary learning.

4.3. *Comparison of Compression Performance*

Finally, we ~~carry~~ 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, ~~an~~ offline data (different from the above four data) is adopted to train the dictionary. K-SVD and PS ~~are~~ were used as the dictionary learning and sparse representation ~~method. Pseudo random algorithm is~~ methods. A pseudo-random algorithm was used to generate the same random variables in the coder and decoder for the ~~bayesian~~ 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~~ situations—especially in higher rates. OffDL, ~~OLOD and proposed method outperform~~ 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 ~~descends~~ deteriorated for sensor 1. The performance of the proposed method is made better than OffDL and OLOD by ~~using~~ the use of clustering. Although the rate will increase by transmission for the information of clustering, the distortion could be efficiently ~~reduced, especially~~ 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.

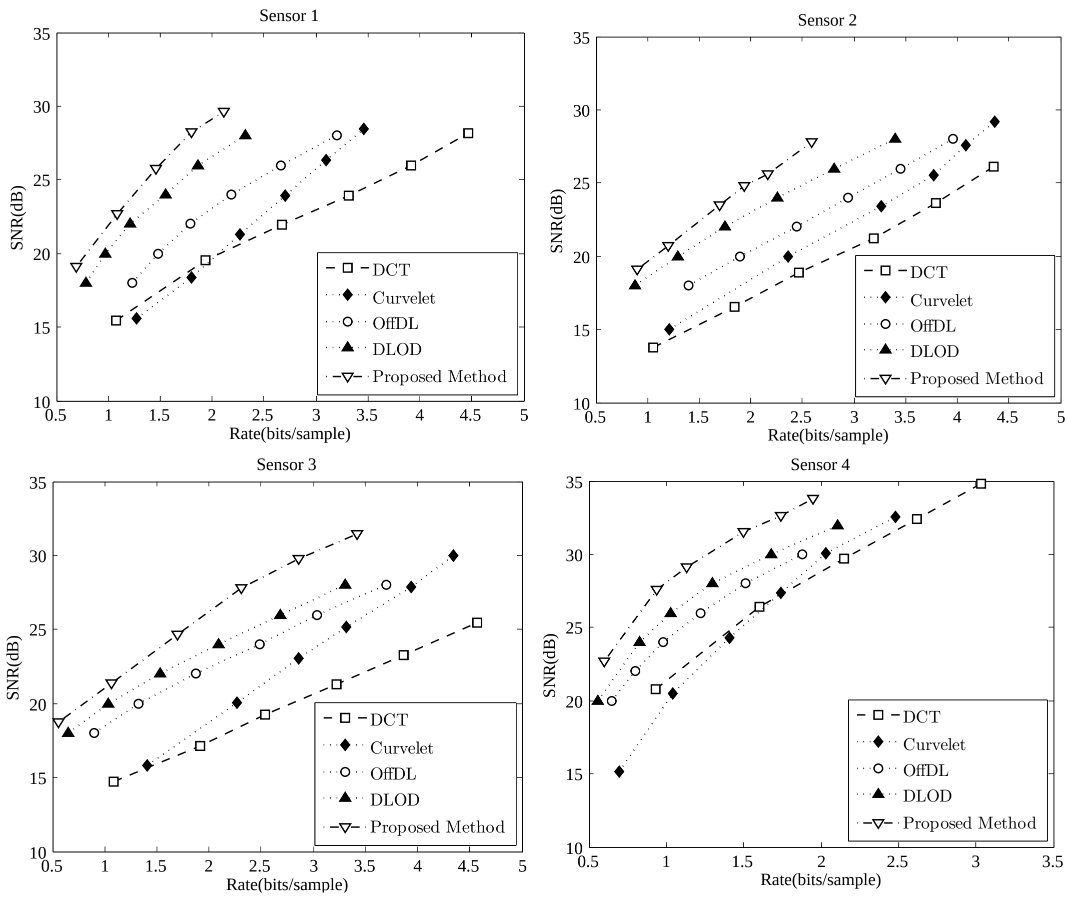


Figure 7. Compression performance comparison of seismic signals from different sensors.

5. Conclusion

In this paper, we have shown how to compress seismic ~~signal~~ signals efficiently by using a ~~clustered based nonparametric bayesian~~ clustering-based nonparametric Bayesian dictionary learning method. The previous transmitted data ~~both existing in the coder and decoder,~~ 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. ~~Nonparametric bayesian~~ 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.

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. *Structural Health Monitoring* **2016**, *15*, 555–567, [http://dx.doi.org/10.1177/1475921716651394].

3. Spanias, A.S.; Jonsson, S.B.; Stearns, S.D. Transform Methods for Seismic Data Compression. *IEEE trans. on Geoscience and Remote sensing* **1991**, *29*, 407–416.
4. Aparna, P.; David, S. Adaptive Local Cosine transform for Seismic Image Compression. 2006 International Conference on Advanced Computing and Communications, 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, [http://dx.doi.org/10.1190/1.2799517].
7. Skretting, K.; Engan, K. Image compression using learned dictionaries by RLS-DLA and compared with K-SVD. 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, pp. 1517–1520.
8. Nejati, M.; Samavi, S.; Karimi, N.; Soroushmehr, S.M.R.; Najaran, K. Boosted multi-scale dictionaries for image compression. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 1130–1134.
9. ZHAN Xin, Z.R. Complex SAR Image Compression Using Entropy-Constrained Dictionary Learning and Universal Trellis Coded Quantization. *Chinese Journal of Electronics* **2016**, *25*, 686.
10. Horev, I.; Bryt, O.; Rubinstein, R. Adaptive image compression using sparse dictionaries. 2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP), 2012, pp. 592–595.
11. Sun, Y.; Xu, M.; Tao, X.; Lu, J. Online Dictionary Learning Based Intra-frame Video Coding. *Wireless Personal Communications* **2014**, *74*, 1281–1295.
12. Adler, J.; Rao, B.D.; Kreutz-Delgado, K. Comparison of basis selection methods. Signals, Systems and Computers, 1996. Conference Record of the Thirtieth Asilomar Conference on, 1996, Vol. 1, pp. 252–257 vol.1.
13. Skretting, K.; Håkon Husøy, J. Partial search vector selection for sparse signal representation. in NORSIG-03, 2003.
14. Engan, K.; Aase, S.O.; Håkon Husøy, J. Method of optimal directions for frame design. Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on, 1999, Vol. 5, pp. 2443–2446 vol.5.
15. Aharon, M.; Elad, M.; Bruckstein, A. K -SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing* **2006**, *54*, 4311–4322.
16. Zepeda, J.; Guillemot, C.; Kijak, E. Image Compression Using Sparse Representations and the Iteration-Tuned and Aligned Dictionary. *IEEE Journal of Selected Topics in Signal Processing* **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. 2010 IEEE International Conference on Image Processing, 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 Transactions on Image Processing* **2012**, *21*, 130–144.
19. Shan, H.; Banerjee, A. Mixed-membership naive Bayes models. *Data Mining and Knowledge Discovery* **2011**, *23*, 1–62.
20. Stramer, O. Monte Carlo Statistical Methods. *Journal of the American Statistical Association* **2001**, *96*, 339–355, [http://dx.doi.org/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. 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 5646–5650.
22. UTAM seismic data library. http://utam.gg.utah.edu/SeismicData/SeismicData.html. [Online].
23. Santafe, G.; Lozano, J.A.; Larranaga, P. Bayesian Model Averaging of Naive Bayes for Clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **2006**, *36*, 1149–1161.
24. Dhillon, I.S.; Mallela, S.; Modha, D.S. Information-theoretic Co-clustering. Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; ACM: New York, NY, USA, 2003; KDD '03, pp. 89–98.