Sparse Bayesian Dictionary Learning with a Gaussian Hierarchical Model

Linxiao Yang*, Jun Fang*, and Hongbin Li‡
*National Key Laboratory on Communications, University of Electronic Science and Technology of China
Chengdu 611731, China Emails: 201321190224@std.uestc.edu.cn, JunFang@uestc.edu.cn
‡Department of Electrical and Computer Engineering, Stevens Institute of Technology
Hoboken, NJ 07030 USA Email: Hongbin.Li@stevens.edu

***Abstract*— We consider a dictionary learning problem aimed at designing a dictionary such that the signals admits a sparse or an approximate sparse representation over the learned dictionary. The problem finds a variety of applications including image denoising, feature extraction, etc. In this paper, we propose a new hierarchical Bayesian model for dictionary learning, in which a Gaussian-inverse Gamma hierarchical prior is used to promote the sparsity of the representation. Suitable non-informative priors are also placed on the dictionary and the noise variance such that they can be reliably estimated from the data. Based on the hierarchical model, a Gibbs sampling method is developed for Bayesian inference. The proposed method have the advantage that it does not require the knowledge of the noise variance *a priori*. Numerical results show that the proposed method is able to learn the dictionary with an accuracy better than existing methods.**

***Index Terms*— Dictionary learning, Gaussian-inverse Gamma prior, Gibbs sampling.**

I. Introduction

Sparse representation has been of significant interest over past few years and found a variety of applications in practice [1]–[3]. In many applications such as image denoising and interpolation, signals often have a sparse representation over a pre-specified non-adaptive dictionary, e.g. discrete consine/wavelet transform (DCT/DWT) bases. Nevertheless, recent research [4], [5] has shown that the recovery, denoising and classification performance can be considerably improved by utilizing an adaptive dictionary that is learned from training signals [5], [6]. This has inspired studies on dictionary learning aimed to design overcompelete dictionaries that can better represent the signals. A number of algorithms, such as the K-singular value decomposition (K-SVD) [4], the method of optimal directions (MOD) [7], dictionary learning with the majorization method [8], and the simultaneous codeword optimization (SimCO) [9], were developed for overcomplete dictionary learning and sparse representation. Most algorithms formulate the dictionary learning as an optimization problem which is solved via a two-stage iterative process, namely, a sparse coding stage and a dictionary update stage. The main difference among these algorithms lies in the dictionary update stage. Specifically, the MOD method updates the dictionary via solving a least square problem. The K-SVD algorithm, instead, updates the atoms of the dictionary in a sequential manner and while updating each atom, the atom is updated along with the nonzero entries in the corresponding row vector of the sparse matrix. The idea of sequential atom update was later extended to provide sequential update of multiple atoms each time [9], and recently generalized to parallel atom-updating in order to further accelerate the convergence of the iterative process [10]. These methods [4], [7]–[10], although offering state-of-the-art performance, have several limitations. Specifically, they may require the knowledge of the sparsity level or the noise/residual variance for sparse coding (e.g. [4]), or this knowledge is needed for meticulously selecting some regularization parameters to properly control the tradeoff between the sparsity level and the data fitting error (e.g. [8], [10]). In practice, however, the prior information about the noise variance and sparsity level is usually unavailable and an inaccurate estimation may result in substantial performance degradation. To mitigate such limitation, a nonparametric Bayesian dictionary learning method called beta-Bernoulli process factor analysis (BPFA) was recently developed in [11]. The proposed method is able to automatically infer the required number of factors (dictionary elements) and the noise variance from the test image.

In this paper, we propose a new hierarchical Bayesian model for dictionary learning, in which a Gaussian-inverse Gamma hierarchical prior is used to promote the sparsity of the representation. Suitable non-informative priors are also placed on the dictionary and the noise variance such that they can be reliably inferred from the data. Based on the hierarchical model, a Gibbs sampling method is developed for Bayesian inference. Simulation results show that the proposed Gibbs sampling algorithm has notable advantages over other state-of-the-art dictionary learning methods in a number of interesting scenarios.

II. Hierarchical Model

Suppose we have L training signals $\{\boldsymbol{y}_l\}_{l=1}^L$, where $\boldsymbol{y}_l \in \mathbb{R}^M$. Dictionary learning aims at finding a common sparsifying dictionary $\boldsymbol{D} \in \mathbb{R}^{M\times N}$ such that these L training signals admit a sparse representation over the overcomplete dictionary $\boldsymbol{D}$, i.e.

$$\boldsymbol{y}_l = \boldsymbol{D}\boldsymbol{x}_l + \boldsymbol{w}_l \qquad \forall l \tag{1}$$

This work was supported in part by the National Science Foundation of China under Grant 61172114, and the National Science Foundation under Grant ECCS-1408182.

where $\boldsymbol{x}_l$ and $\boldsymbol{w}_l$ denote the sparse vector and the residual/noise vector, respectively. Define $\boldsymbol{Y} \triangleq [\boldsymbol{y}_1 \ \ldots \ \boldsymbol{y}_L]$, $\boldsymbol{X} \triangleq [\boldsymbol{x}_1 \ \ldots \ \boldsymbol{x}_L]$, and $\boldsymbol{W} \triangleq [\boldsymbol{w}_1 \ \ldots \ \boldsymbol{w}_L]$. The model (1) can be re-expressed as

$$\boldsymbol{Y} = \boldsymbol{D}\boldsymbol{X} + \boldsymbol{W} \tag{2}$$

Also, we write $\boldsymbol{D} \triangleq [\boldsymbol{d}_1 \ \ldots \ \boldsymbol{d}_N]$, where each column of the dictionary, $\boldsymbol{d}_n$, is called an atom.

In the following, we develop a Bayesian framework for learning the overcomplete dictionary and sparse vectors. To promote sparse representations, we assign a two-layer hierarchical Gaussian-inverse Gamma prior to $\boldsymbol{X}$. The Gaussian-inverse Gamma prior is one of the most popular sparse-promoting priors which has been widely used in compressed sensing [12]–[14]. Specifically, in the first layer, $\boldsymbol{X}$ is assigned a Gaussian prior distribution

$$\begin{aligned} p(\boldsymbol{X}|\boldsymbol{\alpha}) &= \prod_{n=1}^{N}\prod_{l=1}^{L} p(x_{nl}) \\ &= \prod_{n=1}^{N}\prod_{l=1}^{L} \mathcal{N}(x_{nl}|0, \alpha_{nl}^{-1}) \end{aligned} \tag{3}$$

where x_{nl} denotes the (n,l)th entry of $\boldsymbol{X}$, and $\boldsymbol{\alpha} \triangleq \{\alpha_{nl}\}$ are non-negative sparsity-controlling hyperparameters. The second layer specifies Gamma distributions as hyperpriors over the hyperparameters $\{\alpha_{nl}\}$, i.e.

$$\begin{aligned} p(\boldsymbol{\alpha}) &= \prod_{n=1}^{N}\prod_{l=1}^{L} \text{Gamma}(\alpha_{nl}|a,b) \\ &= \prod_{n=1}^{N}\prod_{l=1}^{L} \Gamma(a)^{-1} b^a \alpha_{nl}^{a-1} e^{-b\alpha_{nl}} \end{aligned} \tag{4}$$

where $\Gamma(a) = \int_0^\infty t^{a-1}e^{-t}dt$ is the Gamma function, and the parameters a and b used to characterize the Gamma distribution are chosen to be $a = 0.5$ and $b = 10^{-6}$.

In addition, in order to prevent the dictionary from becoming infinitely large, we assume the atoms of the dictionary $\{\boldsymbol{d}_n\}$ are mutually independent and each atom is placed a normal Gaussian prior, i.e.

$$p(\boldsymbol{D}) = \prod_{n=1}^{N} p(\boldsymbol{d}_n) = \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{d}_n|\boldsymbol{0}, \boldsymbol{I}) \tag{5}$$

The noise $\{\boldsymbol{w}_l\}$ are assumed independent multivariate Gaussian noise with zero mean and covariance matrix $(1/\gamma)\boldsymbol{I}$, where the noise variance $1/\gamma$ is assumed unknown *a priori*. To estimate the noise variance, we place a Gamma hyperprior over γ, i.e.

$$p(\gamma) = \text{Gamma}(\gamma|c,d) = \Gamma(c)^{-1} d^c \gamma^{c-1} e^{-d\gamma} \tag{6}$$

where we set $c = 0.5$ and $d = 10^{-6}$. The proposed hierarchical model provides a general framework for learning the overcomplete dictionary, the sparse codes, as well as the noise variance. In the following, we develop a Gibbs sampling method for Bayesian inference.

III. Proposed Sparse Bayesian Dictionary Learning

We now proceed to perform Gibbs sampler for the proposed hierarchical model. Let $\boldsymbol{\theta} \triangleq \{\boldsymbol{X}, \boldsymbol{\alpha}, \boldsymbol{D}, \gamma\}$ denote all hidden variables in our hierarchical model. We aim to find the posterior distribution of $\boldsymbol{\theta}$ given the observed data $\boldsymbol{Y}$

$$p(\boldsymbol{\theta}|\boldsymbol{Y}) \propto p(\boldsymbol{Y}|\boldsymbol{D}, \boldsymbol{X}, \gamma) p(\boldsymbol{D}) p(\boldsymbol{X}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha}) p(\gamma) \tag{7}$$

To provide an approximation to the posterior distribution of the hidden variables, the Gibbs sampler generates an instance from the distribution of each hidden variable in turn, conditional on the current values of the other hidden variables. It can be shown (see, for example, [15]) that the sequence of samples constitutes a Markov chain, and the stationary distribution of that Markov chain is just the sought-after joint distribution. Specifically, the sequential sampling procedure of the Gibbs sampler is given as follows.

- Sampling $\boldsymbol{X}$ according to its conditional marginal distribution $p(\boldsymbol{X}|\boldsymbol{Y}, \boldsymbol{D}^{(t)}, \boldsymbol{\alpha}^{(t)}, \gamma^{(t)})$;
- Sampling $\boldsymbol{D}$ according to its conditional marginal distribution $p(\boldsymbol{D}|\boldsymbol{Y}, \boldsymbol{X}^{(t+1)}, \boldsymbol{\alpha}^{(t)}, \gamma^{(t)})$;
- Sampling $\boldsymbol{\alpha}$ according to its conditional marginal distribution $p(\boldsymbol{\alpha}|\boldsymbol{Y}, \boldsymbol{D}^{(t+1)}, \boldsymbol{X}^{(t+1)}, \gamma^{(t)})$;
- Sampling γ according to its conditional marginal distribution $p(\gamma|\boldsymbol{Y}, \boldsymbol{D}^{(t+1)}, \boldsymbol{X}^{(t+1)}, \boldsymbol{\alpha}^{(t+1)})$.

Note that the above sampling scheme is also referred to as a blocked Gibbs sampler [16] because it groups two or more variables together and samples from their joint distribution conditioned on all other variables, rather than sampling from each one individually. Details of this sampling scheme are provided next. For simplicity, the notation $p(\boldsymbol{z}|-)$ is used in the following to denote the distribution of variable $\boldsymbol{z}$ conditioned on all other variables.

1). Sampling $\boldsymbol{X}$: Samples of $\boldsymbol{X}$ can be obtained by independently sampling each column of $\boldsymbol{X}$, i.e. $\boldsymbol{x}_l$. The conditional marginal distribution of $\boldsymbol{x}_l$ is given as

$$\begin{aligned} p(\boldsymbol{x}_l|-) &\propto p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{D}, \gamma) p(\boldsymbol{x}_l|\boldsymbol{\alpha}_l) \\ &\propto p(\boldsymbol{y}_l|\boldsymbol{D}, \boldsymbol{x}_l, \gamma) p(\boldsymbol{x}_l|\boldsymbol{\alpha}_l) \end{aligned} \tag{8}$$

where $\boldsymbol{\alpha}_l \triangleq \{\alpha_{nl}\}_{n=1}^N$ are the sparsity-controlling hyperparameters associated with $\boldsymbol{x}_l$, $p(\boldsymbol{y}_l|\boldsymbol{D}, \boldsymbol{x}_l, \gamma)$ and $p(\boldsymbol{x}_l|\boldsymbol{\alpha}_l)$ are respectively given by

$$\begin{aligned} p(\boldsymbol{y}_l|\boldsymbol{D}, \boldsymbol{x}_l, \gamma) &= \left(\frac{\gamma}{2\pi}\right)^{\frac{M}{2}} \exp\left(-\frac{\gamma\|\boldsymbol{y}_l - \boldsymbol{D}\boldsymbol{x}_l\|_2^2}{2}\right) \\ p(\boldsymbol{x}_l|\boldsymbol{\alpha}_l) &= \prod_{n=1}^{N} \mathcal{N}(x_{nl}|0, \alpha_{nl}^{-1}) \end{aligned} \tag{9}$$

Substituting (9) into (8) and after some simplifications, it can be readily verified that $p(\boldsymbol{x}_l|-)$ follows a Gaussian distribution

$$p(\boldsymbol{x}_l|-) = \mathcal{N}(\boldsymbol{\mu}_l^x, \boldsymbol{\Sigma}_l^x) \tag{10}$$

with its mean $\boldsymbol{\mu}_l^x$ and covariance matrix $\boldsymbol{\Sigma}_l^x$ given by

$$\boldsymbol{\mu}_l^x = \gamma \boldsymbol{\Sigma}_l^x \boldsymbol{D}^T \boldsymbol{y}_l \tag{11}$$

$$\boldsymbol{\Sigma}_l^x = (\gamma \boldsymbol{D}^T \boldsymbol{D} + \boldsymbol{\Lambda}_l)^{-1} \tag{12}$$

where $\boldsymbol{\Lambda}_l \triangleq \text{diag}(\alpha_{1l}, \ldots, \alpha_{Nl})$.

2). Sampling $\boldsymbol{D}$: There are two different ways to sample the dictionary: we can sample the whole set of atoms simultaneously, or sample the atoms in a successive way. Here, in order to expedite the convergence of the Gibbs sampler, we sample the atoms of the dictionary in a sequential manner. The log-conditional distribution of $\boldsymbol{d}_n$ can be written as

$$\begin{aligned}\ln p(\boldsymbol{d}_n|-) \propto & \ln p(\boldsymbol{Y}|\boldsymbol{X}, \{\boldsymbol{d}_k\}, \gamma) p(\boldsymbol{d}_n) \\ \overset{(a)}{\propto} & \ln p(\boldsymbol{Y}^{-n}|\boldsymbol{d}_n, \boldsymbol{x}_{n\cdot}, \gamma) p(\boldsymbol{d}_n) \\ \overset{(b)}{\propto} & \frac{1}{2}\gamma \text{tr}\{(\boldsymbol{Y}^{-n} - \boldsymbol{d}_n\boldsymbol{x}_{n\cdot})(\boldsymbol{Y}^{-n} - \boldsymbol{d}_n\boldsymbol{x}_{n\cdot})^T\} \\ & + \boldsymbol{d}_n^T\boldsymbol{d}_n \\ = & \frac{1}{2}\left[\boldsymbol{d}_n^T(\gamma\boldsymbol{x}_{n\cdot}\boldsymbol{x}_{n\cdot}^T + 1)^{-1}\boldsymbol{d}_n - 2\boldsymbol{d}_n\boldsymbol{Y}^{-n}\boldsymbol{x}_{n\cdot}^T\right]\end{aligned} \tag{13}$$

where in (a), we define

$$\boldsymbol{Y}^{-n} \triangleq \boldsymbol{Y} - \boldsymbol{D}^{-n}\boldsymbol{X} \tag{14}$$

in which $\boldsymbol{D}^{-n}$ is generated by $\boldsymbol{D}$ with the nth column of $\boldsymbol{D}$ replaced by a zero vector, and $\boldsymbol{x}_{n\cdot}$ denotes the nth row of $\boldsymbol{X}$, (b) comes from the fact that $\boldsymbol{Y}^{-n} - \boldsymbol{d}_n\boldsymbol{x}_{n\cdot} = \boldsymbol{W}$ and thus we have

$$p(\boldsymbol{Y}^{-n}|\boldsymbol{d}_n, \boldsymbol{x}_{n\cdot}, \gamma) = \frac{\gamma^{\frac{ML}{2}}}{2\pi}\exp\left(-\frac{1}{2}\gamma\|\boldsymbol{Y}^{-n} - \boldsymbol{d}_n\boldsymbol{x}_{n\cdot}\|_F^2\right) \tag{15}$$

Recalling (15), we can show that the conditional distribution of $\boldsymbol{d}_n$ follows a Gaussian distribution

$$p(\boldsymbol{d}_n|-) = \mathcal{N}(\boldsymbol{\mu}_n^d, \boldsymbol{\Sigma}_n^d) \tag{16}$$

with its mean and covariance matrix given by

$$\boldsymbol{\mu}_n^d = \gamma\boldsymbol{\Sigma}_n^d\boldsymbol{Y}^{-n}\boldsymbol{x}_{n\cdot}^T \tag{17}$$

$$\boldsymbol{\Sigma}_n^d = (\gamma\boldsymbol{x}_{n\cdot}\boldsymbol{x}_{n\cdot}^T + 1)^{-1}\boldsymbol{I} \tag{18}$$

3). Sampling $\boldsymbol{\alpha}$: The log-conditional distribution of α_{nl} can be computed as

$$\begin{aligned}\ln p(\alpha_{nl}|-) \propto & \ln p(\alpha_{nl}; a, b) p(x_{nl}|\alpha_{nl}) \\ \propto & \left(a - \frac{1}{2}\right)\ln\alpha_{nl} - \left(b + \frac{x_{nl}^2}{2}\right)\end{aligned} \tag{19}$$

It is easy to verify that α_{nl} still follows a Gamma distribution

$$p(\alpha_{nl}|-) = \text{Gamma}(\hat{a}, \hat{b}_{nl}) \tag{20}$$

with the parameters $\hat{a}$ and $\hat{b}_{nl}$ given as

$$\hat{a} = a + \frac{1}{2} \tag{21}$$

$$\hat{b}_{nl} = b + \frac{1}{2}x_{nl}^2 \tag{22}$$

4). Sampling γ: The log-conditional distribution of γ is given by

$$\begin{aligned}\ln p(\gamma|-) \propto & \ln p(\boldsymbol{Y}|\boldsymbol{D}, \boldsymbol{X}, \gamma) p(\gamma) \\ \propto & \ln\prod_{l=1}^{L} p(\boldsymbol{y}_l|\boldsymbol{D}, \boldsymbol{x}_l, \gamma) p(\gamma) \\ = & \left(\frac{ML}{2} + c - 1\right)\ln\gamma - \left(\frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 + d\right)\gamma\end{aligned} \tag{23}$$

from which we can arrive at

$$p(\gamma|-) = \text{Gamma}(\hat{c}, \hat{d}) \tag{24}$$

where

$$\hat{c} = a + \frac{ML}{2} \tag{25}$$

$$\hat{d} = d + \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 \tag{26}$$

So far we have derived the conditional marginal distributions for hidden variables $\{\boldsymbol{D}, \boldsymbol{X}, \boldsymbol{\alpha}, \gamma\}$. Gibbs sampler successively generates the samples of these variables according to their conditional distributions. After a burn-in period, the generated samples can be viewed as samples drawn from the posterior distribution $p(\boldsymbol{X}, \boldsymbol{D}, \boldsymbol{\alpha}, \gamma|\boldsymbol{Y})$. With those samples, the dictionary can be estimated by averaging the last few samples of the Gibbs sampler. For clarity, we now summarize the Gibbs sampling algorithm as follows.

Sparse Bayesian Dictionary Learning

1. Given the current samples $\boldsymbol{D}^{(t)}$, $\boldsymbol{\alpha}^{(t)}$ and $\gamma^{(t)}$. Generate a sample $\boldsymbol{X}^{(t+1)}$ according to (10).
2. Given the current samples $\boldsymbol{X}^{(t+1)}$, $\boldsymbol{\alpha}^{(t)}$ and $\gamma^{(t)}$. Generate a sample $\boldsymbol{D}^{(t+1)}$ according to (16).
3. Given the current samples $\boldsymbol{D}^{(t+1)}$, $\boldsymbol{X}^{(t+1)}$ and $\gamma^{(t)}$. Generate a sample $\boldsymbol{\alpha}^{(t+1)}$ according to (20).
4. Given the current samples $\boldsymbol{D}^{(t+1)}$, $\boldsymbol{X}^{(t+1)}$ and $\boldsymbol{\alpha}^{(t+1)}$. Generate a sample $\gamma^{(t+1)}$ according to (24).
5. Repeat the above steps and collect the samples after a burn-in period.

IV. Simulation Results

We now carry out experiments to illustrate the performance of our proposed sparse Bayesian dictionary learning (SBDL) method. Throughout our experiments, the parameters for our proposed method are set equal to $a = 0.5$, $b = 10^{-6}$, $c = 0.5$ and $d = 10^{-6}$. We compare our proposed methods with other several existing state-of-the-art dictionary learning methods, namely, the K-SVD algorithm [4], the atom parallel-updating (APrU-DL) method [10], and BPFA [11]. In our experiment, the parameters used in APrU-DL were tuned carefully and the best performances were reported.

We first consider to recover a known dictionary from samples. We generate a dictionary $\boldsymbol{D}$ of size 20×50, with each entry independently drawn from a normal distribution. Columns of $\boldsymbol{D}$ are then normalized to unit norm. The training

TABLE I

RECOVERY SUCCESS RATES

L	SNR	Algorithm	K = 3	K = 4	K = 5	Var. K
1000	10	K-SVD	80.52	36.36	2.52	0.80
		BPFA	76.76	57.12	22.56	43.32
		APrU-DL	85.64	**64.40**	**33.44**	**53.68**
		SBDL	**91.52**	62.48	6.32	41.80
	20	K-SVD	93.20	93.44	92.08	84.68
		BPFA	87.96	92.00	94.08	93.58
		APrU-DL	94.04	93.32	87.76	93.48
		SBDL	**99.64**	**99.16**	**97.52**	**99.12**
	30	K-SVD	94.24	94.32	93.92	86.64
		BPFA	87.04	91.20	94.56	92.60
		APrU-DL	94.24	94.92	88.16	93.96
		SBDL	**99.60**	**99.16**	**98.64**	**99.00**
2000	10	K-SVD	91.00	88.88	50.56	25.32
		BPFA	91.16	92.28	86.44	90.84
		APrU-DL	97.00	94.88	**86.24**	**95.44**
		SBDL	**98.56**	**95.72**	80.20	93.88
	20	K-SVD	95.64	96.68	95.16	94.00
		BPFA	89.68	91.76	94.72	93.04
		APrU-DL	95.40	96.48	95.80	96.56
		SBDL	**99.48**	**99.56**	**98.92**	**99.16**
	30	K-SVD	95.88	96.92	96.96	93.36
		BPFA	87.24	91.08	95.92	93.94
		APrU-DL	94.28	95.00	96.80	95.64
		SBDL	**99.40**	**99.16**	**99.52**	**99.32**

signals $\{\boldsymbol{y}_l\}_{l=1}^{L}$ are produced based on $\boldsymbol{D}$, where each signal $\boldsymbol{y}_l$ is a linear combination of K_l randomly selected atoms and the weighting coefficients are i.i.d. normal random variables. Two different cases are considered. First, all training samples are generated with the same number of atoms, i.e. $K_l = K, \forall l$, and K is assumed exactly known to the K-SVD method. The other case is that K_l varies from 3 to 6 for different l according to a uniform distribution. In this case, the K-SVD assumes that the sparsity level equals to 6 during the sparse coding stage. The observation noise is assumed multivariate Gaussian with zero mean and covariance matrix $\sigma^2\boldsymbol{I}$. The recovery success rate is used to evaluate the dictionary learning performance. Table I shows the average recovery success rates of respective algorithms, where we set $L = 1000$ and $L = 2000$, respectively, and the signal-to-noise ratio (SNR) varies from 10 to 30dB. Results are averaged over 50 independent trials. From Table I, we can see that the proposed method achieves the highest recovery success rates in most cases.

We now demonstrate the results by applying the above methods to image denoising. Suppose images are corrupted by white Gaussian noise with zero mean and variance σ^2. We partition a noise-corrupted image into a number of overlapping patches (of size 8×8 pixels) obtained with one pixel shifting. Half patches are selected for taring. The selected patches are then vectorized to generate the training signal $\{\boldsymbol{y}_l\}$. Also, in our experiments, we assume that the noise variance is perfectly known *a priori* by the K-SVD method. After the training by respective algorithms, the trained dictionary is then used for denoising. Table II shows the peak signal to noise ratio (PSNR) results obtained for different nature images by respective algorithms, where the noise standard deviation is set to $\sigma = \{15, 25, 50\}$, respectively, and the dictionary to be

TABLE II

PSNR RESULTS

σ	Algorithm	boat	cameraman	couple
15	K-SVD	29.2802	31.4638	31.4068
	BPFA	29.5446	31.1759	31.2875
	APrU-DL	29.5718	**31.7662**	**31.5304**
	SBDL	**29.5881**	31.6978	31.4473
25	K-SVD	26.9308	28.6211	28.6949
	BPFA	27.0726	28.4483	28.5825
	APrU-DL	26.8998	28.7069	28.5378
	SBDL	**27.1570**	**28.8380**	**28.8431**
50	K-SVD	22.9499	23.9898	24.3532
	BPFA	23.4165	23.7873	24.5719
	APrU-DL	22.7274	23.5888	24.1901
	SBDL	**23.4651**	**24.1899**	**24.7870**

inferred is assumed of size 64×256. From Table II, we see that the results of all methods are very close to each other in general. The proposed SBDL achieves a slightly higher PSNR than other methods in most cases, This result again demonstrates the superiority of the proposed method. In Fig. 1, we present the noise-corrupted images "cameraman" and "couple", and the denoised images using dictionaries trained by SBDL. The trained dictionaries are also shown on the right sides of Fig. 1.

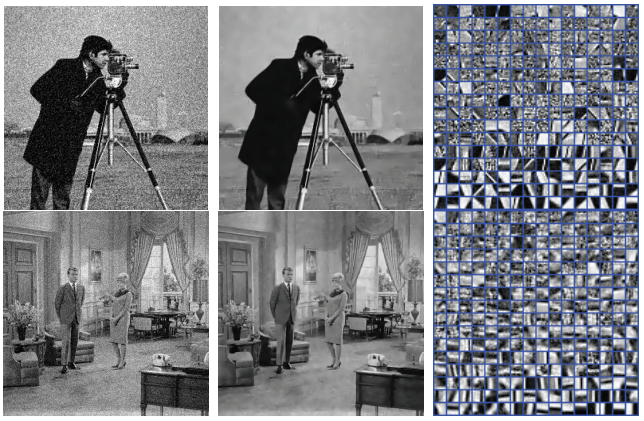

Fig. 1. Example of the denoising results for the image "Cameraman" ($\sigma = 25$. From left to right: the corrupted image, the denoised image by SBDL (28.8380dB), the dictionary trained by SBDL.

V. CONCLUSIONS

We developed a new Bayesian hierarchical model for learning overcomplete dictionaries based on a set of training data. This new framework extends the conventional sparse Bayesian learning framework to deal with the dictionary learning problem. Unlike some of previous methods, the proposed methods do not need to assume knowledge of the noise variance *a priori*, and can infer the noise variance automatically from the data. Numerical results show that the proposed methods are able to learn the dictionary with an accuracy notably better than existing methods.

REFERENCES

[1] E. Candés and T. Tao, "Decoding by linear programming," *IEEE Trans. Information Theory*, no. 12, pp. 4203–4215, Dec. 2005.

[2] J. M. Duarte-Carvajalino and G. Sapiro, "Learning to sense sparse signals: simultaneous sensing matrix and sparsifying dictionary optimization," *IEEE Trans. Image Processing*, vol. 18, no. 7, pp. 1395–1408, July 2009.

[3] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition vis sparse representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[4] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[5] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Processing*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[6] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791–804, Apr. 2012.

[7] K. Engan, S. O. Aase, and J. H. Hakon-Husoy, "Method of optimal directions for frame design," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Phoenix, AZ, March 15-19 1999.

[8] M. Yaghoobi, T. Blumensath, and M. E. Davies, "Dictionary learning for sparse approximations with the majorization method," *IEEE Trans. Signal Processing*, vol. 57, no. 6, pp. 2178–2191, June 2009.

[9] W. Dai, T. Xu, and W. Wang, "Simultaneous codeword optimization (simco) for dictionary update and learning," *IEEE Trans. Signal Processing*, vol. 60, no. 12, pp. 6340–6353, Dec. 2012.

[10] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "Learning overcomplete dictionaries based on atom-by-atom updating," *IEEE Trans. Signal Processing*, vol. 62, no. 4, pp. 883–891, Feb. 2014.

[11] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro, and L. Carin, "Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images," *IEEE Trans. Image Processing*, vol. 21, no. 1, pp. 130–144, Jan. 2012.

[12] D. P. Wipf and B. D. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Trans. Signal Processing*, vol. 55, no. 7, pp. 3704–3716, July 2007.

[13] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Processing*, vol. 56, no. 6, pp. 2346–2356, June 2008.

[14] J. Fang, Y. Shen, H. Li, and P. Wang, "Pattern-coupled sparse Bayesian learning for recovery of block-sparse signals," *IEEE Trans. Signal Processing*, vol. 63, no. 2, pp. 360–372, Jan. 2015.

[15] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*. Chapman and Hall/CRC, third edition, 2013.

[16] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2007.