ONLINE DICTIONARY LEARNING OVER DISTRIBUTED MODELS

Jianshu Chen, Zaid J. Towfic, and Ali H. Sayed

Department of Electrical Engineering
University of California, Los Angeles

ABSTRACT

In this paper, we consider learning dictionary models over a network of agents, where each agent is only in charge of a portion of the dictionary elements. This formulation is relevant in big data scenarios where multiple large dictionary models may be spread over different spatial locations and it is not feasible to aggregate all dictionaries in one location due to communication and privacy considerations. We first show that the dual function of the inference problem is an aggregation of individual cost functions associated with different agents, which can then be minimized efficiently by means of diffusion strategies. The collaborative inference step generates local error measures that are used by the agents to update their dictionaries without the need to share these dictionaries or even the coefficient models for the training data. This is a useful property that leads to an efficient distributed procedure for learning dictionaries over large networks.

Index Terms— Dictionary learning, distributed model, diffusion strategies, dual decomposition.

1. INTRODUCTION AND RELATED WORK

Dictionary learning is a useful procedure by which dependencies among input features can be represented in terms of suitable bases. It has found applications in many machine learning and inference tasks including image denoising [1, 2], dimensionality-reduction [3, 4], biclustering [5], feature-extraction and classification [6], and novel document detection [7]. Dictionary learning usually alternates between two steps: (i) an inference (sparse coding) step and (ii) a dictionary update step. The first step finds a sparse representation for the input data using the existing dictionary by solving an ℓ_1-regularized regression problem, and the second step usually employs gradient descent to update the dictionary entries.

With the increasing complexity of various learning tasks, it is natural that the size of the learning dictionaries is becoming increasingly demanding in terms of memory and computing requirements. It is therefore important to study scenarios where the dictionary need not be available in a single location but is instead spread out over multiple locations. This is particularly true in big data scenarios where multiple large dictionary models may be already available at separate locations and it is not feasible to aggregate all dictionaries in one location due to communication and privacy considerations. This observation motivates us to examine how to learn a dictionary model that is stored over a network of agents, where each agent is in charge of only *a portion* of the dictionary elements. Compared with other works, the problem we solve in this article is how to learn a *distributed* dictionary model, which is, for example, different from the useful work in [8] where it is assumed instead that each agent maintains the *entire* dictionary model.

This work was supported in part by NSF grant CCF-1011918. Emails: {cjs09, ztowfic, sayed}@ucla.edu

In this paper, we will first formulate a modified version of the sparse coding problem, where we add an additional ℓ_2 regularization term besides the ℓ_1 term (also known as elastic net regularization [3]). This modified problem is not in a form that is directly amenable to a distributed implementation. However, we will show that the modified problem has a dual function that can be solved in a distributed manner using diffusion strategies [9–13]. Useful consensus strategies [14, 15] can also be used. However, since it has been noted that diffusion strategies have enhanced stability and learning abilities over consensus strategies [16], we continue our presentation by focusing on diffusion strategies.

The inference algorithm that we develop is fully distributed in the sense that each agent only needs to apply a local gradient descent step followed by an information exchange step of the dual variable within its neighborhood. We will show that this dual variable has a useful interpretation, namely, it corresponds to the representation error for the input data sample relative to all dictionary elements. Therefore, the agents do not need to share their (private) dictionary elements but only this representation error, which is computed in a distributed manner through local interactions. We test our algorithm on a typical image denoising task. The dictionary is learned from a collection of patches arising from natural scenes and the learned dictionary is used to reconstruct a noisy image not included in the training set. The denoised image's peak-signal-to-noise-ratio (PSNR) is found to rival that of a *centralized* dictionary learning algorithm [2]. In other words, our results show that the distributed solution does not limit performance. On the contrary, it can perform as well as a fully centralized solution. This observation has useful ramifications for dealing with large dictionaries and large data sets.

2. PROBLEM FORMULATION

We seek to solve the following *global* dictionary learning problem over a network of N agents connected by a topology:

$$\min_{W} \quad \mathbb{E}\left[\frac{1}{2}\|\boldsymbol{x}_t - W\boldsymbol{y}_t^o\|_2^2 + \gamma\|\boldsymbol{y}_t^o\|_1 + \frac{\delta}{2}\|\boldsymbol{y}_t^o\|_2^2\right] \tag{1}$$

$$\text{s.t.} \quad \|w_k\|_2^2 \le 1, \quad k = 1, \ldots, N \tag{2}$$

where $\mathbb{E}\boldsymbol{x}$ denotes the expectation operator, $\boldsymbol{x}_t$ is the $M \times 1$ input data vector at time t (we use boldface letters to represent random quantities), W is an $M \times N$ dictionary matrix, w_k is the k-th column of W (also known as the k-th dictionary element, or *atom*), γ and δ are positive regularization factors for the ℓ_1 and ℓ_2 terms, respectively, and $\boldsymbol{y}_t^o$ is the solution to the following sparse coding problem for each input data sample x_t at time t (the regular font x_t

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

denotes a realization for $\boldsymbol{x}_t$):

$$y_t^o = \arg\min_y \underbrace{\left[\frac{1}{2}\|x_t - Wy\|_2^2 + \gamma\|y\|_1 + \frac{\delta}{2}\|y\|_2^2\right]}_{\triangleq Q(W,y;x_t)} \quad (3)$$

Note that dictionary learning consists of two steps: the sparse coding step (inference) for the realization x_t at each time t in (3), and the dictionary update step (learning) in (1)–(2). Let y_k denote the k-th entry of the $N \times 1$ vector y. Then, the objective function of the inference step (3) can be written as

$$Q(W, y; x_t) \triangleq \frac{1}{2}\left\|x_t - \sum_{k=1}^N w_k y_k\right\|_2^2 + \sum_{k=1}^N \left(\gamma \cdot |y_k| + \frac{\delta}{2} \cdot y_k^2\right) \quad (4)$$

The dictionary elements $\{w_k\}$ are linearly combined to represent each input data sample, and the first term in the cost function (4) requires the representation error to be small. In this paper, we focus on using quadratic costs to measure the representation error. In [17], we generalize the results to any differentiable strictly convex costs. The second and third terms in (4), which correspond to the ℓ_1 and ℓ_2 regularizations in (3), are meant to ensure that the resulting combination coefficients $\{y_k\}$ are sparse and small. The ℓ_2 term makes the regularization strongly convex, which will allow us to develop a fully decentralized strategy that enables the dictionary elements $\{w_k\}$ and the corresponding coefficients $\{y_k\}$ to be stored and learned in a distributed manner over the network. That is, each agent k will infer its own y_k and update its own dictionary element, w_k, by relying solely on limited interactions with its neighboring agents. Furthermore, as explained in [17], such strongly convex regularization terms help transform the non-differentiable primal cost (4) into a better-conditioned smooth optimization problem — see (16) further ahead. Figure 1 shows the configuration of the knowledge and data distribution over the network. The dictionary elements $\{w_k\}$ can be interpreted as the "wisdom" that is distributed over the network, and which we wish to combine in a distributed manner to form a greater "intelligence" for interpreting the data sample $\boldsymbol{x}_t$. By being distributed, we would like the networked agents to find the *global* solutions to both the inference problem (3) and the learning problem (1)–(2) with interactions that are limited to their neighborhoods.

Note that the problem we are solving in this paper is different from [8] and the traditional distributed learning setting [9, 10, 12, 18, 19], where the entire set of model parameters (the dictionary elements $\{w_k\}$ in this case) are maintained at each agent in the network, whereas the data samples are collected and processed over the network, i.e., these previous scenarios correspond to *data distributed* formulations. What we are studying in this paper is to find a distributed solution where each agent is only in charge of a portion of the model (e.g., w_k for each agent k). This scenario corresponds to a *model distributed* formulation. This case is important because each agent may be limited in its memory and computing power and may not be able to store large dictionaries. By having many agents cooperate with each other, a larger model that is beyond the ability of any single agent can be stored and analyzed in a distributed manner.

3. LEARNING OVER DISTRIBUTED MODELS

3.1. Inference over distributed models

Observe that solving the cost function (4) directly requires knowledge of all dictionary elements $\{w_k\}$ and coefficients $\{y_k\}$ from the other agents due to the sum inside the $\|\cdot\|_2^2$ that runs from $k = 1$ up

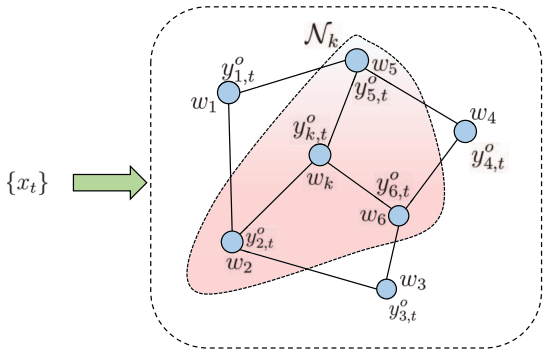

Fig. 1. Each agent is in charge of one dictionary element, w_k, and the corresponding coefficient, y_k, and the data sample x_t at each time t is available to all agents in the network. The results in this paper are generalized to the case where the data sample x_t is only available to a subset of the agents, and where each agent is responsible for a submatrix of W consisting of multiple columns and not only a single atom, w_k — see the extended work [17].

to N. Therefore, this formulation is not directly amenable to a distributed solution. However, we can arrive at an efficient distributed strategy by transforming the original optimization problem into a dual problem. To begin with, we first transform the minimization of (4) into the following equivalent constrained optimization problem:

$$\min_{\{y_k\}, z} \quad \frac{1}{2}\|x_t - z\|_2^2 + \sum_{k=1}^N \left(\gamma \cdot |y_k| + \frac{\delta}{2} \cdot y_k^2\right) \quad (5)$$

$$\text{s.t.} \quad z = \sum_{k=1}^N w_k y_k \quad (6)$$

Note that the above problem is convex over both $\{y_k\}$ and z since the objective is convex and the equality constraint is linear. By strong duality [20, p.514], it follows that the optimal solution to (5)–(6) can be found by solving its corresponding dual problem and then recovering the optimal $\{y_k\}$ and z. To arrive at the dual problem, we introduce the Lagrangian of (5)–(6) for each input realization x_t as

$$L(\{y_k\}, z, \nu; x_t) = \frac{1}{2}\|x_t - z\|_2^2 + \sum_{k=1}^N \left(\gamma|y_k| + \frac{\delta}{2} \cdot y_k^2\right) + \nu^T\left(z - \sum_{k=1}^N w_k y_k\right) \quad (7)$$

where $\{y_k\}$ and z are the primal variables and ν is the Lagrange multiplier (also known as the dual variable). The dual function $g(\nu; x_t)$ is defined as the minimization of $L(\{y_k\}, z, \nu; x_t)$ over the primal variables $\{y_k\}$ and z for each given ν:

$$g(\nu; x_t) \triangleq \min_{\{y_k\}, z} L(\{y_k\}, z, \nu; x_t) \quad (8)$$

Given that strong duality holds, it is known that the optimal solution of (5)–(6) can be found by solving the following dual problem:

$$\nu_t^o = \arg\max_\nu g(\nu; x_t) \quad (9)$$

and then recovering the optimal primal variables $y_{k,t}^o$ and z_t^o via

$$(\{y_{k,t}^o\}, z_t^o) = \arg\min_{\{y_k\}, z} L(\{y_k\}, z, \nu_t^o; x_t) \quad (10)$$

Notice from (7) that the minimization in (10) over the variables $\{y_k\}$ and z for a given ν is decoupled, and the minimization over each y_k is also decoupled for different k. Therefore, the minimization over the primal variables can be done independently. Computing the derivative of $L(\{y_k\}, z, \nu; x_t)$ with respect to z and setting it to zero, we obtain, for each given ν, the optimal solution of z satisfies

$$-(x_t - z) + \nu = 0 \quad \Leftrightarrow \quad z = x_t - \nu \tag{11}$$

Furthermore, since $L(\{y_k\}, z, \nu; x_t)$ is not differentiable in y_k, the condition for minimizing $L(\{y_k\}, z, \nu; x_t)$ with respect to y_k is given by [21, p.133]:

$$0 \in \partial_{y_k} L(\{y_k\}, z, \nu; x_t) = \delta \cdot y_k + \gamma \cdot \partial_{y_k} |y_k| - \nu^T w_k \tag{12}$$

where ∂_{y_k} denotes the sub-differential (the set of all subgradients) with respect to y_k, and the sub-differential for $|y_k|$ is

$$\partial_k |y_k| = \begin{cases} \text{sign}(y_k), & y_k \neq 0 \\ [-1, 1], & y_k = 0 \end{cases} \tag{13}$$

Applying an argument similar to the one used in [22] to Eq. (12), we can express the optimal y_k as

$$y_k = \mathcal{T}_{\frac{\gamma}{\delta}} \left(\frac{\nu^T w_k}{\delta} \right) \tag{14}$$

where $\mathcal{T}_\lambda(\cdot)$ denotes the following soft-thresholding scalar-valued operator of $x \in \mathbb{R}$:

$$\mathcal{T}_\lambda(x) \triangleq (|x| - \lambda)_+ \text{sgn}(x) \tag{15}$$

where $(x)_+ = \max\{0, x\}$. Observe that the solutions obtained in (11) and (14) are optimal for a given ν. Only when we have the optimal ν_t^o to the dual problem (9), the corresponding z and y_k acquired from (11) and (14) become the optimal solution to the original problem (5)–(6); the notation z_t^o and $y_{k,t}^o$ will be used to represent the z and y_k solutions corresponding to ν_t^o. Substituting (11) and (14) into (7), we obtain the dual function as

$$\begin{aligned} g(\nu; x_t) &= -\frac{1}{2}\|\nu\|^2 + \nu^T x_t - \sum_{k=1}^N \mathcal{S}_{\frac{\gamma}{\delta}} \left(\frac{\nu^T w_k}{\delta} \right) \\ &= -\sum_{k=1}^N \underbrace{\left\{ \frac{1}{2N}\|\nu\|^2 - \frac{1}{N}\nu^T x_t + \mathcal{S}_{\frac{\gamma}{\delta}} \left(\frac{\nu^T w_k}{\delta} \right) \right\}}_{\triangleq J_k(\nu;\, x_t)} \end{aligned} \tag{16}$$

where we introduced the following scalar-valued function of $x \in \mathbb{R}$, which is a differentiable convex function:

$$\mathcal{S}_{\frac{\gamma}{\delta}}(x) \triangleq -\frac{\delta}{2} \cdot \mathcal{T}^2_{\frac{\gamma}{\delta}}(x) - \gamma \cdot \left| \mathcal{T}_{\frac{\gamma}{\delta}}(x) \right| + \delta \cdot x \cdot \mathcal{T}_{\frac{\gamma}{\delta}}(x) \tag{17}$$

The functions $\mathcal{T}_\lambda(x)$ and $\mathcal{S}_\lambda(x)$ are illustrated in Fig. 2. Therefore, the maximization of the dual problem (9) is equivalent to the following minimization problem

$$\min_\nu \sum_{k=1}^N J_k(\nu; x_t) \tag{18}$$

Note that the new equivalent form (18) is an aggregation of individual costs associated with different agents; each agent k is associated with cost $J_k(\nu; x_t)$, which only requires knowledge of w_k and x_t.

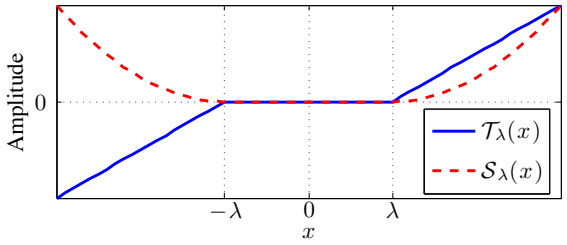

Fig. 2. Illustration of the functions $\mathcal{T}_\lambda(x)$ and $\mathcal{S}_\lambda(x)$.

Therefore, we can now directly apply the diffusion strategies developed in [11, 12] to solve the above problem in a fully distributed manner over the network:

$$\psi_{k,i} = \nu_{k,i-1} - \mu_\nu \cdot \nabla_\nu J_k(\nu_{k,i-1}; x_t) \tag{19}$$

$$\nu_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \cdot \psi_{\ell,i} \tag{20}$$

where $\nu_{k,i}$ denotes the estimate of the optimal ν_t^o at each agent k at iteration i (we will use i to denote the i-th iteration of the inference, and use t to denote the t-th data sample), $\psi_{k,i}$ is an intermediate variable, μ_ν is the step-size parameter chosen to be a small positive number, and $a_{\ell k}$ is the combination coefficient that agent k assigns to the information shared from agent ℓ and it satisfies

$$\sum_{\ell \in \mathcal{N}_k} a_{\ell k} = 1, \quad a_{\ell k} > 0 \text{ if } \ell \in \mathcal{N}_k, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{21}$$

Let A denote the matrix that collects $a_{\ell k}$ as its (ℓ, k)-th entry. Then, it is shown in [11, 12] that as long as the matrix A is primitive, doubly-stochastic and the step-size is sufficiently small, then the algorithm (19)–(20) converges to the optimal solution of (18) with a small bias on the order of $O(\mu_\nu^2)$ in squared Euclidean norm. Finally, after ν_t^o is estimated at each agent k, the optimal z and y_k can be recovered from ν by substituting ν_t^o into (11) and (14), respectively:

$$z_t^o = x_t - \nu_t^o \tag{22}$$

$$y_{k,t}^o = \mathcal{T}_{\frac{\gamma}{\delta}} \left(\frac{w_k^T \nu_t^o}{\delta} \right) \tag{23}$$

Note that (23) only requires local knowledge of w_k. An important remark we have is a physical interpretation for the optimal dual variable ν_t^o. Since z_t^o and $y_{k,t}^o$ are the optimal solutions to problem (5)–(6), then z_t^o and $y_{k,t}^o$ also need to satisfy constraint (6) so that

$$z_t^o = \sum_{k=1}^N w_k y_{k,t}^o \tag{24}$$

Expressions (22) and (24) imply that

$$\nu_t^o = x_t - \sum_{k=1}^N w_k y_{k,t}^o \tag{25}$$

In other words, ν_t^o admits the interpretation of corresponding to the optimal prediction error of the input data sample x_t using all the dictionary $\{w_k\}$. In this way, the diffusion algorithm (19)–(20) is able to estimate the prediction error in a distributed manner for all agents.

3.2. Distributed dictionary updates

We now derive the strategy that updates the local dictionary element w_k at each agent k. Specifically, we need to solve the constrained stochastic optimization problem (1)–(2), which can be rewritten as

$$\min_{W} \quad \mathbb{E}Q(W, \boldsymbol{y}_t^o; \boldsymbol{x}_t) \tag{26}$$

$$\text{s.t.} \quad \|w_k\|_2 \leq 1, \quad k = 1, \ldots, N \tag{27}$$

where $\boldsymbol{y}_t^o \triangleq \text{col}\{\boldsymbol{y}_{1,t}^o, \ldots, \boldsymbol{y}_{N,t}^o\}$ and $Q(W, \boldsymbol{y}_t^o; \boldsymbol{x}_t)$ is defined in (4). Our strategy is to apply stochastic gradient descent to the cost function (26) with respect to each w_k followed by a projection onto the constraint set $\{w_k : \|w_k\| \leq 1\}$. The stochastic gradient of the cost function (26) with respect to w_k is the gradient of $Q(W, \boldsymbol{y}_t^o; \boldsymbol{x}_t)$ with respect to w_k. Therefore, the algorithm can be described as

$$w_{k,t} = \Pi_B\Big(w_{k,t-1} - \mu_w \cdot \nabla_{w_k} Q(W, y_t^o; x_t)\Big) \tag{28}$$

where $\Pi_B(x)$ is the projection operator onto $\{w_k : \|w_k\| \leq 1\}$. From (4), the stochastic gradient can be computed as

$$\nabla_{w_k} Q(W, y_t^o; x_t) = -\left(x_t - \sum_{k=1}^{N} w_k y_{k,t}^o\right) y_{k,t}^o \tag{29}$$

On the face of it, expression (29) requires global knowledge of all dictionary elements $\{w_k\}$ across the network, which would prevent the distributed implementation. However, recalling (25), the expression inside the parenthesis on the right-hand side of (29) is nothing but ν_t^o, which is estimated locally by each agent by means of the distributed inference algorithm (19)–(20). Therefore, the dictionary learning update (28) can be expressed as

$$w_{k,t} = \Pi_B\Big(w_{k,t-1} + \mu_w \cdot \nu_t^o y_{k,t}^o\Big) \tag{30}$$

where each agent k replaces the above ν_t^o by the estimate $\nu_{k,i}$ after a sufficient number of inference iterations (large enough i). The rightmost update term in (30) for dictionary element k is effectively the correlation between the global prediction error, ν_t^o, and the coefficient $y_{k,t}^o$ (the activation).

4. EXPERIMENT

We consider learning a 100×196 dictionary W over a network of $N = 196$ agents. The network is generated according to a random graph, where the probability that any agent is connected to another agent is 0.2. The network connectivity is checked by inspecting the algebraic connectivity of the graph Laplacian matrix, and we will repeat this random graph generation until we find a connected topology. Each agent in the network is in charge of one dictionary element. We extract a total of 1 million 10×10 patches from images 101-200 of the the non-calibrated natural image dataset [23]. Each image is originally 1536×1024 pixels in size, but the border two pixels were discarded around each image and the top-left 1019×1019 pixels were then used for patch extraction. With each data sample being a 10×10 patch from a certain image, the dimension of the input data sample is $M = 100$ (vertically stacked columns). In each experiment, we randomly initialize each entry of the dictionary matrix W with a zero mean unit variance Gaussian random variable. The columns are then scaled to guarantee that the sub-unit-norm constraint (2) is satisfied. Furthermore, in the combination step (20) of the distributed inference, we use the Metropolis rule [9, 24, 25],

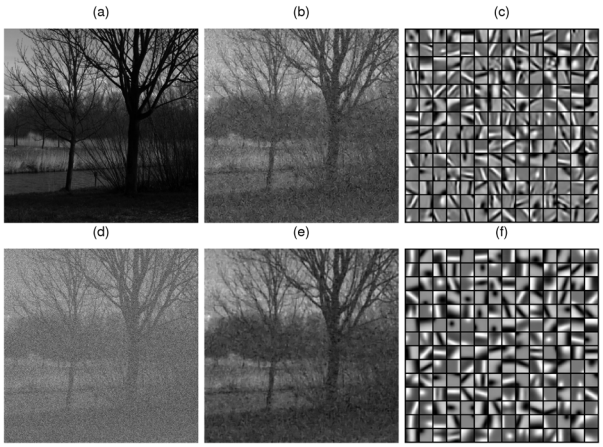

Fig. 3. Application of dictionary learning to image denoising. (a) Original image; (b) denoised image by using the centralized method from [2]; (c) dictionary obtained by the centralized method from [2]; (d) image corrupted by additive white Gaussian noise; (e) denoised image by our proposed distributed method at agent 1; (f) dictionary obtained by our proposed distributed method.

which is known to be doubly-stochastic. The patch extraction, preprocessing, and image reconstruction code utilized (excluding dictionary learning and patch inference steps) is borrowed from [26].

For the dictionary learning, we utilize $\gamma = 45$, $\delta = 0.1$, and $\mu_\nu = 0.7$. Computer code from the SPAMS toolbox was used to compare the algorithm from [2] using its default parameters except where otherwise stated. A step-size of $\mu_w = 5 \times 10^{-5}$ was utilized for adapting the dictionary atoms. The number of iterations for the diffusion algorithm to optimize (3) was chosen to be 300 iterations. The data were presented in minibatches [27] of size four samples/minibatch and the dictionary update gradients $\nu_t^o y_{k,t}^o$ were averaged over the four samples at each step[1].

In the far right of Fig. 3, we show the dictionary learned over the 196 agents in the network (bottom) as well as the one learned by using the centralized method in [2] as a benchmark (top). The learned dictionary can be used to denoise an image corrupted by noise as shown in the left four images of Fig. 3. Observe that since the dictionaries were trained on patches arising from natural scenes, these dictionaries are capable of denoising other natural scenes since they are expected to share the same statistics. In denoising Fig. 3, the step-size for our algorithm's inference was increased to be $\mu_\nu = 1$ to increase the quality of the inference result (ν^o). The number of iterations of the inference step increased to 500 iterations to ensure convergence and $\gamma = 45$ and $\delta = 0.1$ remained constant for all algorithms. The corrupted image's PSNR[2] is 14.056dB, while the PSNR for the recovered images using the centralized solution of [2] and our proposed distributed solution were found to be 21.771dB and 21.976dB (at agent 1), respectively. Furthermore, the average denoising PSNR performance across the distributed network was found to be 21.979dB with a standard deviation of 0.00340dB. We observe that the performance is relatively uniform across the network.

[1] We perform the inference for four samples $(x_1, \ldots, x_4)$ at a time to obtain $\{\nu_{k,1}^o, \ldots, \nu_{k,4}^o\}$ (all using the same dictionary W). Then, we update W by averaging the gradient listed in (30) for those four samples.

[2] PSNR is the peak-signal-to-noise ratio defined as $\text{PSNR} \triangleq 10\log_{10}(I_{\max}^2/\text{MSE})$, where $I_{\max}$ is the maximum pixel intensity in the image and MSE is the mean-square-error over all image pixels.

5. REFERENCES

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

[2] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, Mar. 2010.

[3] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 265–286, Jan. 2006.

[4] H. Shen and J. Z. Huang, "Sparse principal component analysis via regularized low rank matrix approximation," *Journal of Multivariate Analysis*, vol. 99, no. 6, pp. 1015–1034, Jul. 2008.

[5] M. Lee, H. Shen, J. Z. Huang, and J. S. Marron, "Biclustering via sparse singular value decomposition," *Biometrics*, vol. 66, no. 4, pp. 1087–1095, Dec. 2010.

[6] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," in *Proc. NIPS*, Lake Tahoe, Nevada, Dec. 2008, pp. 1033–1040.

[7] S. P. Kasiviswanathan, H. Wangy, A. Banerjeey, and P. Melville, "Online ℓ_1-dictionary learning with application to novel document detection," in *Proc. NIPS*, Lake Tahoe, Nevada, Dec. 2012, pp. 2267–2275.

[8] P. Chainais and C. Richard, "Learning a common dictionary over a sensor network.," in *Proc. IEEE CAMSAP*, St Martin, French West Indies, Dec. 2013.

[9] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[10] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.

[11] J. Chen and A. H. Sayed, "On the limiting behavior of distributed optimization strategies," in *Proc. Allerton Conf.*, Monticello, IL, Oct. 2012, pp. 1535–1542.

[12] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion adaptation," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 205–220, Apr. 2013.

[13] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks — Part I: Transient analysis," *submitted for publication* [also available as arXiv:1312.7581], 2013.

[14] S. Kar, J. M. F. Moura, and K. Ramanan, "Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication," *IEEE Trans. Inf. Theory*, vol. 58, no. 6, pp. 3575–3605, Jun. 2012.

[15] S. Lee and A. Nedic, "Distributed random projection algorithm for convex optimization," *IEEE Journal Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 221–229, Apr. 2013.

[16] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[17] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," *submitted for publication,* [also available as arXiv: 1402.1515], Feb. 2014.

[18] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4692–4707, Oct. 2011.

[19] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1419–1433, Mar. 2013.

[20] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 2nd edition, 1999.

[21] B. Polyak, *Introduction to Optimization*, Optimization Software, NY, 1987.

[22] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[23] J. H. van Hateren and A. van der Schaaf, "Independent component filters of natural images compared with simple cells in primary visual cortex," *Proc. Biological Sciences*, vol. 265, no. 1394, pp. 359–366, Mar. 1998.

[24] A. H. Sayed, "Diffusion adaptation over networks," *in* Academic Press Library in Signal Processing, vol. 3, R. Chellapa and S. Theodoridis, *editors,* pp. 323–454, Elsevier, 2014 [also available online as arXiv:1205.4220v2, May 2012].

[25] X. Zhao and A. H. Sayed, "Performance limits for distributed estimation over LMS adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5107–5124, Oct. 2012.

[26] G. Peyré, "The numerical tours of signal processing - advanced computational signal and image processing," *IEEE Computing in Science and Engineering*, vol. 13, no. 4, pp. 94–97, Jul. 2011.

[27] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction using mini-batches," *The Journal of Machine Learning Research*, vol. 13, pp. 165–202, Jan. 2012.