Active Dictionary Learning for Image Representation

Tong Wu, Anand D. Sarwate, and Waheed U. Bajwa

Department of Electrical and Computer Engineering
Rutgers, The State University of New Jersey, Piscataway, NJ 08854

ABSTRACT

Sparse representations of images in overcomplete bases (i.e., redundant dictionaries) have many applications in computer vision and image processing. Recent works have demonstrated improvements in image representations by learning a dictionary from training data instead of using a predefined one. But learning a sparsifying dictionary can be computationally expensive in the case of a massive training set. This paper proposes a new approach, termed *active screening*, to overcome this challenge. Active screening sequentially selects subsets of training samples using a simple heuristic and adds the selected samples to a "learning pool," which is then used to learn a newer dictionary for improved representation performance. The performance of the proposed *active dictionary learning* approach is evaluated through numerical experiments on real-world image data; the results of these experiments demonstrate the effectiveness of the proposed method.

Keywords: Active screening, dictionary learning, sparse representation

1. INTRODUCTION

With the development of new, affordable smartphone and "prosumer" cameras, massive amounts of high-resolution images are being captured, stored, and subsequently transmitted daily through communication and online social networks. These massive collections of high-resolution images have the ability to transform the landscape of computer vision and image processing research and practice. In particular, massive image datasets are expected to enable researchers achieve the holy grail of machine vision whose performance is at least on par with that of human vision. But almost all processing tasks in image processing and computer vision are preceded by identification of intrinsic (geometric) features of images. It is therefore important to develop efficient approaches to learning the structure/features of images in this so-called "big data" regime, which can be challenging due to memory and storage constraints. In particular, in a setting of ever-accumulating data sets, it may not be feasible to use all available images to build representations or learn features. Instead, we need to develop methods that can efficiently identify the most informative images for learning the structure/features of images; doing so will reduce the computation and storage complexity in numerous applications.

1.1 Our Contribution and Relation to Prior Work

Dictionary learning—a nonlinear, data-driven feature learning framework—has emerged in recent years as one of the most popular approaches to learning the geometric structure of images.[1] The basic assumption is that (patches of) images can be well-approximated using (sparse) linear combinations of a few columns (also called *atoms*) in an appropriate overcomplete basis, referred to as a *dictionary*. Put formally, let $y \in \mathbb{R}^m$ be a (vectorized) image (patch). We say y admits a sparse representation in a dictionary $D \in \mathbb{R}^{m \times K}$ if $y \approx D\theta$ and the number of nonzero entries in $\theta \in \mathbb{R}^K$ is small compared to K—the number of atoms in D. Note that the *sparsifying* dictionary D in general can either be constructed analytically using a predefined model, such as wavelets[2] or curvelets,[3] or it can be learned from the images themselves.[1,4,5] Since predefined dictionaries do not adapt to the underlying images, they do not always result in best image representations.[4] Dictionary learning, on the other hand, refers to the problem of obtaining a dictionary from the available images in a data-adaptive manner. In particular, the representation power of learned dictionaries has been known to exceed that of predefined ones.[4] In this paper, we therefore focus on the problem of (data-adaptive) dictionary learning. We can

E-mails: {tong.wu.ee, anand.sarwate, waheed.bajwa}@rutgers.edu. This work is supported in part by the Army Research Office under grant W911NF-14-1-0295 and by an Army Research Lab Robotics CTA subaward.

then use the resulting dictionary for other computer vision and image processing tasks. Our main contribution in this regard is a novel method for efficient dictionary learning from massive image datasets. The proposed method, which we call *active dictionary learning*, relies on a judiciously-chosen subset of training images for efficient construction of data-adaptive dictionaries.

In terms of relation to prior work, there exist several algorithms in the literature that are remarkably efficient in learning dictionaries from small- or medium-scale datasets.[6] However, since these algorithms use every training sample to learn a dictionary, they become less desirable in terms of computational and storage costs when the number of training images becomes very large (or effectively infinite). Active dictionary learning, on the other hand, addresses the challenge of finite memory and computation resources in dictionary learning by relying on a novel heuristic—which we term *active screening*—that selects only a few samples from the large-scale image repository, adds them to the "learning pool," and then incrementally refines the learned dictionary using this learning pool. This particular approach to active dictionary learning is motivated by studies of variable screening and selective sampling in the statistics and machine learning literature, respectively.[7,8] Our method also bears some resemblance to stochastic optimization methods such as stochastic gradient descent.[9] The main advantage of active screening, which can be specialized to other feature learning methods besides dictionary learning, is that it "screens" the data for meaningful images and discards uninformative "raw" images; this leads to dictionary learning with much lower storage and computational costs. In order to demonstrate the effectiveness of active screening and active dictionary learning, we carry out several numerical experiments on real-world image datasets. The results of these experiments validate our heuristic that discarding of uninformative data can lead to significant storage and computational savings in the context of feature learning.

1.2 Notational Convention

Throughout the paper, we use lower-case and upper-case letters for vectors and matrices, respectively. The i-th element of a vector v is denoted by $v(i)$ and the (i,j)-th element of a matrix A is denoted by $A(i,j)$. The m-dimensional zero vector is denoted by $\mathbf{0}_m$. Given a set Ω, $[A]_{:,\Omega}$ (resp., $[v]_\Omega$) denotes the submatrix of A (resp., subvector of v) corresponding to the columns of A (resp., entries of v) indexed by Ω. Given a matrix A, a_j and $a_{j,T}$ denote the j-th column and the j-th row of A, respectively. Superscript $(\cdot)^T$ denotes the transpose operation and $\|\cdot\|_0$ counts the number of nonzero entries in a vector. Finally, the Frobenius norm of a matrix A is denoted by $\|A\|_F$ and $\|v\|_2$ denotes the ℓ_2 norm of a vector v.

1.3 Organization

The rest of this paper is organized as follows. In Section 2, we provide a brief overview of the problem of dictionary learning. In Section 3, we describe our approach to active dictionary learning from massive image data. In Section 4, we describe the results of numerical experiments that validate the effectiveness of our proposed method for active dictionary learning. We finally conclude in Section 5 with some remarks.

2. AN OVERVIEW OF DICTIONARY LEARNING

Suppose we are given a set of $\mathcal{N}$ (vectorized) images $Y = [y_1, \ldots, y_\mathcal{N}] \in \mathbb{R}^{m \times \mathcal{N}}$, where m denotes the dimensionality of each image. The problem of learning a reconstructive sparsifying dictionary of K atoms using the training data Y can be expressed as the following matrix factorization program:[1]

$$(D, \Theta) = \arg\min_{D,\Theta} \|Y - D\Theta\|_F^2 \quad \text{subject to} \quad \|\theta_i\|_0 \leq s, \forall i = 1, \ldots, \mathcal{N}. \tag{1}$$

Here, $D \in \mathbb{R}^{m \times K}$ is an overcomplete dictionary (i.e, $K > m$) with unit ℓ_2-norm columns and $\Theta = [\theta_1, \ldots, \theta_\mathcal{N}] \in \mathbb{R}^{K \times \mathcal{N}}$ denotes the coefficient matrix in which each column $\theta_i, i = 1, \ldots, \mathcal{N}$, has no more than s nonzero entries. In words, the goal of dictionary learning is to find an overcomplete basis such that each example in the training set can be well represented using a linear combination of no more than s atoms of the basis. This problem, as expressed in (1), is non-convex in (D, Θ). Instead of minimizing the objective in (1) over these two variables simultaneously—which is computationally challenging, a tractable approach to solving this problem is to iteratively alternate between minimization over the two variables. Two of the most well-known approaches to dictionary learning in this regard include the *method of optimal directions*[10] and K-SVD.[1] In this paper, our

active screening approach to dictionary learning from massive image data is based on the K-SVD algorithm. We therefore provide a brief description of it in here, while a pseudocode of the algorithm is provided in Appendix A. Note however that active screening is trivially applicable to other dictionary learning methods.

K-SVD involves alternate minimization of (1) between: (i) solving (1) for Θ using a fixed D, which is termed as the *sparse coding* step; and (ii) solving (1) for D using a fixed Θ, which is termed as the *dictionary update* step. Typically, K-SVD is initialized with a random dictionary D. Next, given a fixed D, the sparse coding step amounts to solving Θ as follows:

$$\forall i, \quad \theta_i = \arg\min_{\theta} \|y_i - D\theta\|_2^2 \quad \text{subject to} \quad \|\theta\|_0 \leq s. \tag{2}$$

While this step can be implemented in a number of ways, one popular method for sparse coding is the *orthogonal matching pursuit* (OMP) algorithm,[11–13] which is also listed in Appendix A. Afterward, given a fixed Θ, the dictionary update step in K-SVD involves sequentially updating one atom $d_k, k = 1, \ldots, K$, at a time, while keeping other atoms $\{d_j : j \neq k\}$ in the dictionary fixed. Specifically, to update d_k, define $E_k = Y - \sum_{j \neq k} d_j \theta_{j,T}$ and let ω_k denote the indices of training examples in Y that use the k-th atom of D in their representations. Then the problem of updating d_k can be expressed as the following rank-1 optimization problem:

$$(d_k, [\theta_{k,T}]_{\omega_k}^T) = \arg\min_{d,g} \|[E_k]_{:,\omega_k} - dg^T\|_F^2 \quad \text{subject to} \quad \|d\|_2^2 = 1. \tag{3}$$

This rank-1 optimization problem can be easily solved through a singular value decomposition (SVD) of the "reduced" error matrix $[E_k]_{:,\omega_k}$. We refer the reader to Algorithm 2 in the appendix for further details.

While K-SVD and its variants have been demonstrated to have fast computational times for small- and medium-scale training datasets,[6] their computational complexity in the case of massive training data becomes large due to the SVD step in (3). The fundamental reason for this is simple: *K-SVD incorporates every training sample into its learning framework*. In contrast, we introduce in the next section our framework for massive datasets in which the training samples arrive over time in batches. We then describe our approach to dictionary learning in this *batch streaming* model, which involves refining the dictionary estimate using only a subset of the incoming training batch and that in turn makes dictionary learning computationally scalable.

3. ACTIVE SCREENING OF DATA FOR EFFICIENT DICTIONARY LEARNING

Our focus in this paper is on the case of massive (potentially infinite) training data. In order to formulate the problem of dictionary learning in this setting, we first describe a batch streaming model for training data. Note however that our discussion in the following is easily applicable to a non-streaming, finite-but-massive training data setting; the only difference in that case would be that one would have to manually partition the training data into a number of smaller training batches.

3.1 System Model

Our streaming model assumes that we are given an initial training dataset $Y_{\text{init}} \in \mathbb{R}^{m \times N_0}$ at time $t = 0$ that consists of N_0 training samples. Next, at each discrete time $t = 1, 2, \ldots$, we assume a new set of N training samples $Y_t = [y_{1,t}, \ldots, y_{N,t}] \in \mathbb{R}^{m \times N}$ is made available to the dictionary learning algorithm. The goal in this batch streaming setting is to produce a new dictionary $D_t \in \mathbb{R}^{m \times K}$ at time t after the arrival of the t-th training batch such that all training samples up to time t can be well represented through no more than s atoms of D_t.

Clearly, if we use all of the training samples up to time t to learn D_t, then at each time t, the problem simply reduces to solving the dictionary learning problem (1) with $N_0 + tN$ samples. But the storage cost of such an approach will be $O(N_0 + tN)$. Also, it has been shown that the computational cost of K-SVD scales roughly linearly with the number of training samples.[6] In order to control the storage and computational cost of dictionary learning in this streaming setting, our goal is to limit the number of training samples that get added to the "learning pool" for refining D_t at each time t.

Algorithm 1 Active Dictionary Learning

Inputs: Initial training set Y_{init}, number of training examples to be selected in each batch $\{B_t\}$, number of atoms K, and sparsity level s.

Initialization: Learn an initial dictionary $D_0 \in \mathbb{R}^{m \times K}$ from Y_{init} using K-SVD, and set $P_0 \leftarrow Y_{\text{init}}$.

for all t **do**
  Receive training batch $Y_t = [y_{1,t}, \ldots, y_{N,t}]$.
  for all $i = 1$ to N **do**
    Compute $\hat{y}_{i,t} \leftarrow D_{t-1}\theta_{i,t}$ with $\|\theta_{i,t}\|_0 \leq s$ by using OMP to generate an s-sparse representation of $y_{i,t}$.
    Calculate the representation error $\epsilon(y_{i,t}) \leftarrow \|y_{i,t} - \hat{y}_{i,t}\|_2^2$.
  end for
  Set $S_t \leftarrow \big\{y_{i,t} \in Y_t : y_{i,t}$ is one of the B_t elements in Y_t with the largest error $\epsilon(y_{i,t})\big\}$.
  Set the training pool $P_t \leftarrow P_{t-1} \cup S_t$.
  Retrain a sparsifying dictionary $D_t \in \mathbb{R}^{m \times K}$ with training set P_t using K-SVD.
end for

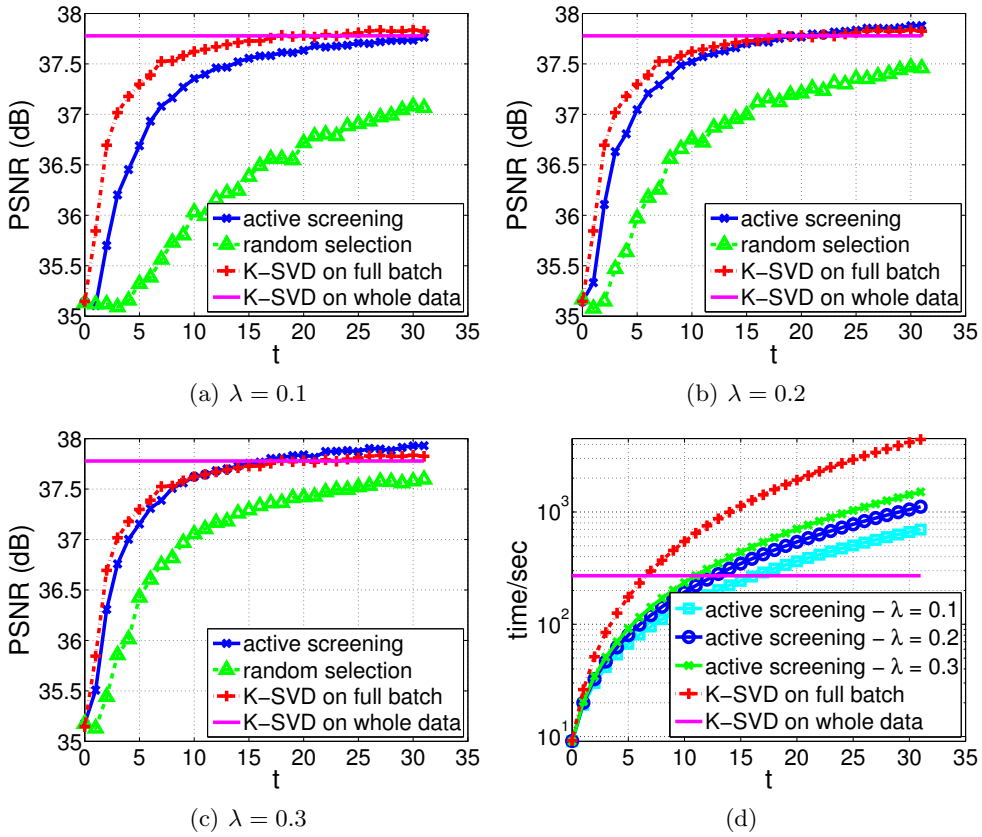
3.2 Active Data Screening

At each time t, we are interested in choosing the most informative and useful samples from the newest batch of training data. Our sample selection in this regard is guided by the following heuristic: *many of the samples will be redundant in a large training set.* In particular, training samples that are well represented by the current dictionary are less likely to add much information in refining the dictionary. Instead, we should primarily focus on those samples that have large representation errors since they are much more likely to provide information that the current dictionary does not capture; indeed, one expects such samples to correspond to parts of the data space that are not well represented by the sparsifying dictionary. In this paper, we use the squared-error metric $\epsilon(y) = \min_{\theta:\|\theta\|_0 \leq s} \|y - D\theta\|_2^2$ for any new training sample y as a guide for detecting samples that are not adequately represented by the current dictionary D and are thus more informative. More precisely, at time t, we use the dictionary D_{t-1} to represent each sample in Y_t using no more than s atoms of D_{t-1}. Next, we calculate the representation error $\epsilon(y_{i,t}) = \|y_{i,t} - D_{t-1}\theta_{i,t}\|_2^2, i = 1, \ldots, N$, where $\theta_{i,t}$ denotes the best s-sparse representation of $y_{i,t}$ in the dictionary D_{t-1}. Finally, we only retain a fraction of samples from the latest training batch whose errors $\epsilon(y_{i,t})$'s are the largest. The samples selected using this greedy heuristic, which we term *active screening*, are then added to the learning pool that is finally used to learn a newer dictionary D_t. This entire procedure of *active dictionary learning*, which relies on active data screening to find informative samples at each time t, is given in Algorithm 1. Note that by reducing the number of samples added to the learning pool, active dictionary learning controls both the storage cost and the computational complexity of learning a new dictionary after each training batch. We formally validate this claim using numerical experiments in the next section.

4. NUMERICAL EXPERIMENTS

In this section, we present the results of some numerical experiments on two facial image datasets to validate the effectiveness of Algorithm 1 for dictionary learning from massive datasets. Since our datasets have finite numbers of images, we emulate our streaming model as follows. Suppose we are given a training set with $\mathcal{N}$ samples. We randomly divide this set into $T + 1$ batches such that $\mathcal{N} = N_0 + NT$, with the batch having N_0 samples in it corresponding to the initial training batch Y_{init} and each of the remaining T batches with N samples corresponding to one of the training batches $Y_t, t = 1, \ldots, T$. All the results reported in the following for Algorithm 1 correspond to constant $B_t = B = \lambda N$ for all T iterations, where the parameter $\lambda \in (0, 1]$. In terms of comparison with other approaches, we compare the results of Algorithm 1 with that of: (i) random selection of B elements in each of the T iterations, (ii) adding the full batch Y_t to the learning pool for every t, which corresponds to $\lambda = 1$ in our setting, and (iii) K-SVD on the entire training set of $\mathcal{N} = N_0 + NT$ samples.* In

*During the course of the preparation of this final manuscript, we became familiar with a recent preprint[14] that also studies dictionary learning using active selection of training samples. While the selection criteria used there is different from ours, we found that the results generated by the two approaches are very similar to each other. We defer a detailed comparison between these selection criteria and ours to a sequel of this work.



(a) $\lambda = 0.1$ (b) $\lambda = 0.2$

(c) $\lambda = 0.3$ (d)

Figure 1. (a)–(c) show PSNR results for different dictionary learning methods on the Extended Yale B dataset, while (d) shows the semi-log plot of the computation time for these methods as a function of t.

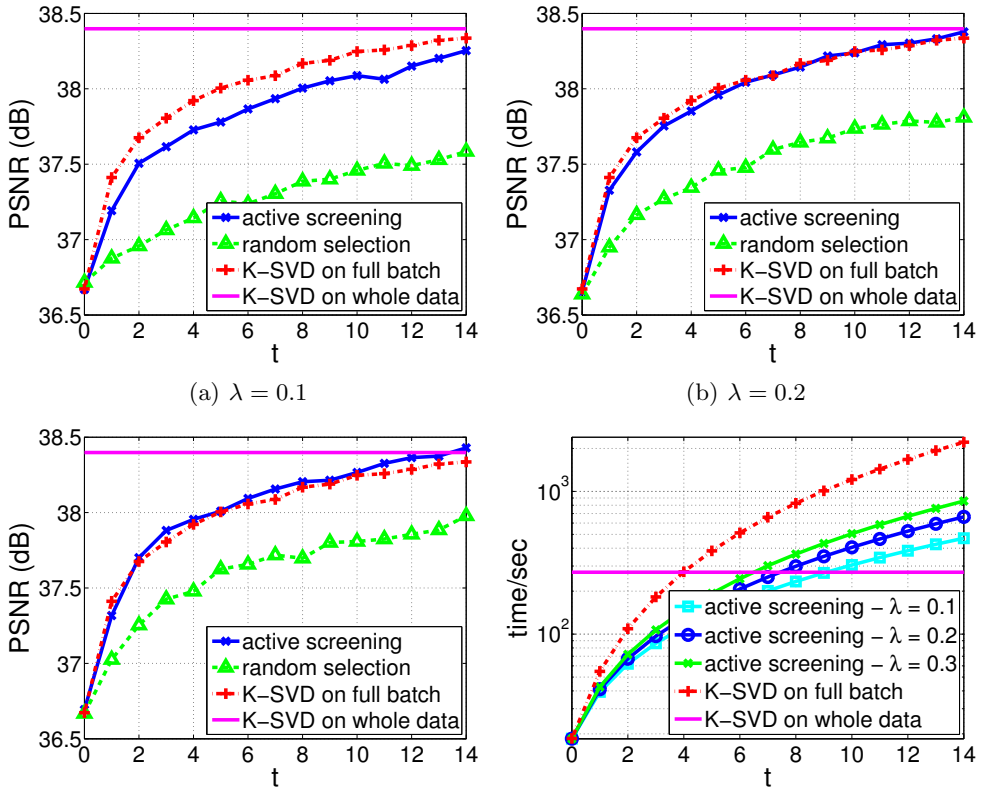
the case of active dictionary learning, we focused on three values of $\lambda = 0.1$, 0.2 and 0.3. All experiments are performed with Matlab R2013a on an Intel i7-2600 3.4GHz CPU with 16 GB RAM.

We first perform experiments on the Extended Yale B Database.[15] This dataset contains a collection of 192×168 pixel images from 38 subjects. For every subject, there are 64 frontal images taken under different illumination conditions. In our experiments, we choose subjects 1–10 and 21–30. For each of the 20 subjects, we randomly choose 48 of their images for training and the remaining 16 images of each subject are reserved for testing purposes (this results in 960 training images in total). This random selection is repeated 10 times in our experiments for cross-validation purposes. In the streaming setup, we split the 960 training images into 30 initial training images and $T = 31$ additional batches with 30 images in each batch. This step is also repeated 10 times, so the results reported in here correspond to an average of 100 trials. Further, each image is divided into 224 non-overlapping patches of size 12×12 for dictionary learning, so that $m = 144$, $N_0 = N = 6720$ and $\mathcal{N} = 215040$. Finally, we learn the dictionary using parameters $K = 500$ and $s = 10$.

Table 1. Comparison between active screening and K-SVD on full batch/entire data for Extended Yale B dataset

	$\lambda = 0.1$		$\lambda = 0.2$			$\lambda = 0.3$			$\lambda = 1$	K-SVD on
	$t = 10$	$t = 16$	$t = 8$	$t = 12$	$t = 21$	$t = 7$	$t = 10$	$t = 17$	$t = 6$	entire data
PSNR (dB)	37.35	37.58	37.38	37.6	37.79	37.39	37.62	37.8	37.39	37.78
time (sec)	144.5	266	143.1	250	585	141.7	234.9	537.7	232.6	270.6

The results of these experiments are reported in Figure 1 and Table 1. Figures 1(a)–1(c) plot the representation errors on the test data in terms of the peak signal-to-noise ratio (PSNR) for different λ's as a function of t. Specifically, given an M-dimensional vectorized image I and its approximation $\hat{I}$ ($M = 32256$ in this



(a) $\lambda = 0.1$ (b) $\lambda = 0.2$

(c) $\lambda = 0.3$ (d)

Figure 2. (a)–(c) show PSNR results for different dictionary learning methods on the LFW dataset, while (d) shows the semi-log plot of the computation time for these methods as a function of t.

experiment), the PSNR (in dB) is defined as $PSNR = 10\log_{10}\frac{MAX_I^2}{\|I-\hat{I}\|_2^2/M}$ where MAX_I denotes the maximum pixel value of I. Comparing active screening with random selection, we see that for almost every fixed t and λ, the performance of active screening is much better than that of random selection. We can also see that the plots for active screening and K-SVD on full batch are very close to each other in the case of $\lambda = 0.2$. Further, active screening outperforms K-SVD on the entire training data for large t in the case of $\lambda = 0.2$ and 0.3. Finally, we compare the computational cost of active dictionary learning with that of K-SVD on the full batch and entire training data. In the case of Algorithm 1, the computation time calculations include time for sparse coding of the data and sample selection in each batch as well as the time needed for refining the dictionary. The final computation times are plotted in Figure 1(d) and some numerical values are also listed in Table 1. These results clearly demonstrate the efficiency of active dictionary learning in terms of the tradeoff between storage, computation, and performance. To be specific, these results show that active screening takes roughly half of the time that K-SVD takes on the whole training data, but it still achieves a PSNR that is only 0.4 dB less than canonical K-SVD. Furthermore, active screening can use less than 1/9 of the entire training data to achieve a PSNR that is within 0.18 dB of the PSNR of K-SVD on the entire training data. In order to better understand the reason active screening performs so well, we also visualize some image patches in Figure 3 that are selected by active screening and random selection. It can be seen from this figure that while there exist many dark patches in the learning pool obtained using random selection, active screening always selects informative patches for all t. Indeed, these dark patches returned by random selection correspond to edges of the images and are not very useful for representing the more interesting regions of the images.

Next, we repeat these experiments on face images from the Labeled Faces in the Wild (LFW) dataset.[16] In these experiments, we use a set of 530 images of George W. Bush of size 250×250 pixels each. We randomly split these 530 images into 450 training and 80 test images. The training images are then further divided into

Table 2. Comparison between active screening and K-SVD on full batch/entire data for LFW dataset

	$\lambda = 0.1$		$\lambda = 0.2$			$\lambda = 0.3$			$\lambda = 1$	K-SVD on
	$t = 5$	$t = 9$	$t = 4$	$t = 7$	$t = 14$	$t = 4$	$t = 6$	$t = 14$	$t = 3$	entire data
PSNR (dB)	37.78	38.05	37.85	38.09	38.38	37.95	38.09	38.43	37.8	38.40
time (sec)	140.1	268.7	130	251.1	663.3	147.2	244.5	856.2	182.3	271.6

30 initial training images and $T = 14$ additional batches with 30 images each. Each image is finally divided into 625 non-overlapping patches of size 10×10, thereby resulting in $m = 100$, $N_0 = N = 18750$ and $\mathcal{N} = 281250$. In these experiments, we learn the dictionary using parameters $K = 400$ and $s = 10$. The results of these experiments on test data are reported in Figure 2 and Table 2. Specifically, Figures 2(a)–2(c) report the PSNR results for active screening and K-SVD on the full batch/entire training data, while Figure 2(d) and Table 2 help us understand the storage–computation–performance tradeoffs of active dictionary learning. In addition, Figure 4 provides visualization of some selected patches obtained through active screening and random selection. All these results once again suggest the superiority of active screening in terms of its low computational/storage cost with minimal performance loss compared to K-SVD using the whole training data. Further, active screening always selects patches that tend to be different from each other; this coincides with our main intuition for active screening that it selects training samples that are more informative.

5. CONCLUSION

In this paper, we introduced a novel algorithm, termed active dictionary learning, for learning sparsifying dictionaries from massive datasets. At the heart of active dictionary learning is a method, referred to as active screening, for judiciously selecting most informative training examples from a given dataset. Compared with a random selection method, active screening sequentially selects samples from the training data that have the largest representation errors using the current learned dictionary. These samples are then added to the learning pool for refining the dictionary estimate, which greatly improves the representation capability of the dictionary. We also carried out numerical experiments on two image datasets to demonstrate the effectiveness of active dictionary learning.

APPENDIX A. PSEUDOCODE FOR K-SVD AND OMP

Algorithm 2 K-SVD Algorithm

Inputs: Training data $Y \in \mathbb{R}^{m \times \mathcal{N}}$, number of atoms K, and target sparsity s.
Task: Find an $m \times K$ dictionary D by solving

$$\min_{(D,\Theta)} \|Y - D\Theta\|_F^2 \quad \text{subject to} \quad \|\theta_i\|_0 \leq s, \forall i. \tag{4}$$

Initialize: Set a nonzero matrix $D^{(0)} \in \mathbb{R}^{m \times K}$ with unit ℓ_2-norm columns; Set $J \leftarrow 1$.
Repeat until convergence (stopping rule):

1: *Sparse Coding:* For each sample $y_i, i = 1, \ldots, \mathcal{N}$, use OMP (see Algorithm 3) to compute the representation coefficients θ_i. Set $\Theta^{(J)} \leftarrow [\theta_1, \ldots, \theta_{\mathcal{N}}]$.
2: *Dictionary Update:* For each column $d_k, k = 1, \ldots, K$, in $D^{(J-1)}$, update it by:

- Set $\omega_k \leftarrow \{i | 1 \leq i \leq \mathcal{N}, \theta_{k,T}(i) \neq 0\}$.
- Set $E_k \leftarrow Y - \sum_{j \neq k} d_j \theta_{j,T}$.
- Given the SVD decomposition $[E_k]_{:,\omega_k} = U \Sigma V^T$, set $d_k \leftarrow u_1$, and $[\theta_{k,T}]_{\omega_k} \leftarrow \Sigma(1,1) \cdot v_1^T$.

3: Set $J \leftarrow J + 1$.

Algorithm 3 Orthogonal Matching Pursuit

Inputs: Dictionary $D = [d_1, \ldots, d_K] \in \mathbb{R}^{m \times K}$, data vector $y \in \mathbb{R}^m$, and target sparsity s.
Initialize: $r_0 \leftarrow y$, $\Lambda_0 \leftarrow \emptyset$, and $\Phi_0 \leftarrow \emptyset$.

1: **for** $t_m = 1$ to s **do**
2: $\quad \lambda_{t_m} \leftarrow \arg\max_{k=1,\ldots,K} |d_k^T r_{t_m - 1}|$.
3: $\quad \Lambda_{t_m} \leftarrow \Lambda_{t_m - 1} \cup \{\lambda_{t_m}\}$, and $\Phi_{t_m} \leftarrow [\Phi_{t_m - 1}\ d_{\lambda_{t_m}}]$.
4: $\quad e_{t_m} \leftarrow \arg\min_e \|y - \Phi_{t_m} e\|_2^2$.
5: $\quad a_{t_m} \leftarrow \Phi_{t_m} e_{t_m}$, and $r_{t_m} \leftarrow y - a_{t_m}$.
6: **end for**
7: $\theta \leftarrow \mathbf{0}_K$, $[\theta]_{\Lambda_s} \leftarrow e_s$.

Output: Sparse representation coefficient vector θ such that $y \approx D\theta$ and $\|\theta\|_0 \leq s$.

REFERENCES

[1] Aharon, M., Elad, M., and Bruckstein, A., "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.* **54**(11), 4311–4322 (2006).

[2] Mallat, S., [*A wavelet tour of signal processing*], Academic, 3rd ed. (2009).

[3] Candès, E. J. and Donoho, D. L., "New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities," *Comm. Pure Appl. Math.* **57**(2), 219–266 (2004).

[4] Elad, M. and Aharon, M., "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.* **15**(12), 3736–3745 (2006).

[5] Mairal, J., Bach, F., Ponce, J., and Sapiro, G., "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.* **11**, 19–60 (2010).

[6] Rubinstein, R., Zibulevsky, M., and Elad, M., "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," Tech. Rep. CS-2008-08, Technion Computer Science Department (2008).

[7] Guyon, I. and Elisseeff, A., "An introduction to variable and feature selection," *J. Mach. Learn. Res.* **3**, 1157–1182 (2003).

[8] Cesa-Bianchi, N., Gentile, C., and Orabona, F., "Robust bounds for classification via selective sampling," in [*Proc. Intl. Conf. Machine Learning (ICML)*], 121–128 (2009).

[9] Bottou, L., "Large-scale machine learning with stochastic gradient descent," in [*Proc. Intl. Conf. Computational Statistics (COMPSTAT)*], 177–187 (2010).

[10] Engan, K., Aase, S. O., and Husøy, J. H., "Method of optimal directions for frame design," in [*Proc. IEEE Intl. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*], **5**, 2443–2446 (1999).

[11] Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S., "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in [*Proc. 27th Asilomar Conf. Signals, Systems, and Computers*], 40–44 (1993).

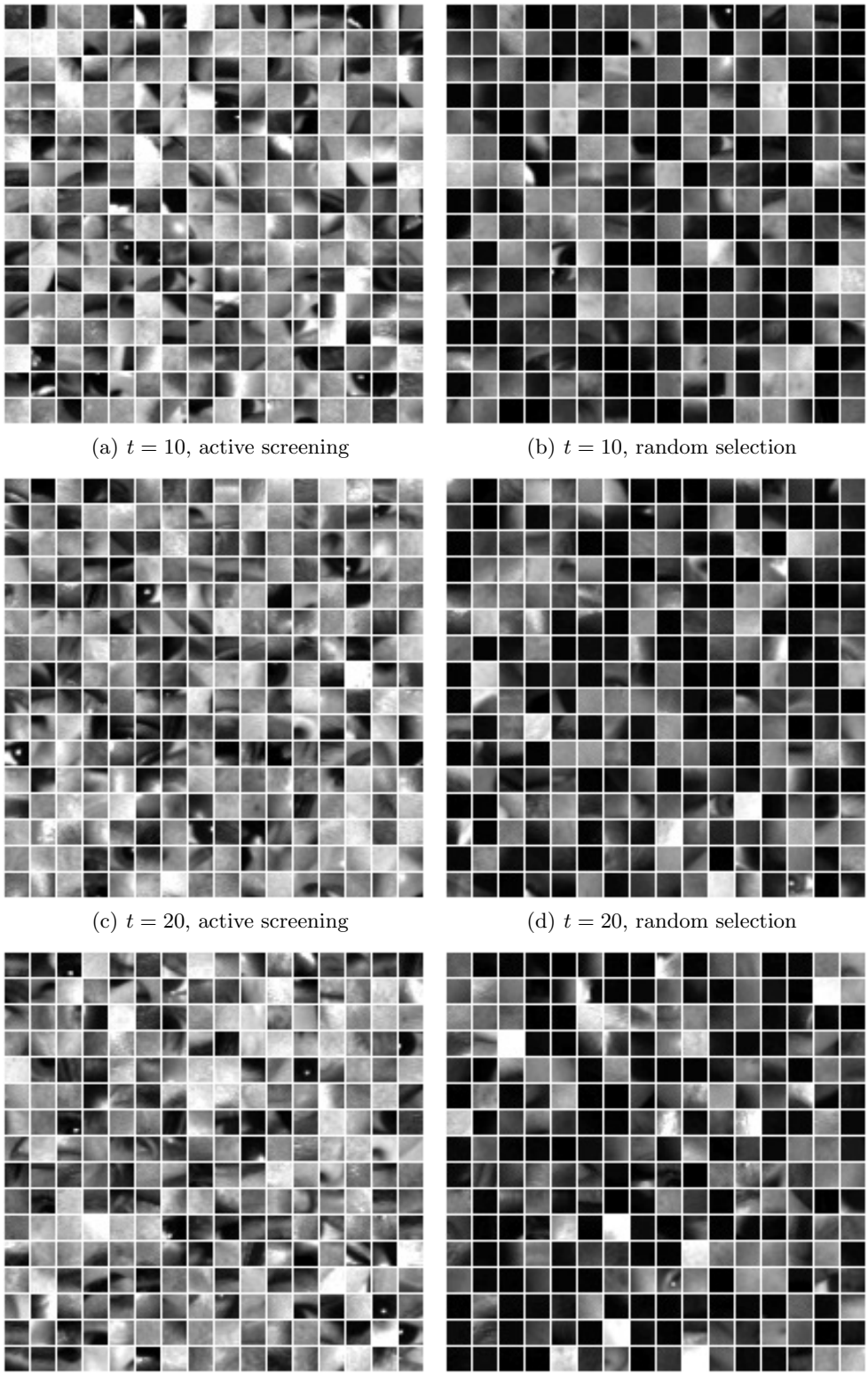
[12] Mallat, S. G. and Zhang, Z., "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.* **41**(12), 3397–3415 (1993).

[13] Tropp, J. A. and Gilbert, A. C., "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory* **53**(12), 4655–4666 (2007).

[14] Tsuchida, T. and Cottrell, G. W., "Example selection for dictionary learning," Tech. Rep. arXiv:1412.6177, ArXiV (2014).

[15] Lee, K.-C., Ho, J., and Kriegman, D. J., "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(5), 684–698 (2005).

[16] Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E., "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Amherst (2007).



(a) $t = 10$, active screening

(b) $t = 10$, random selection

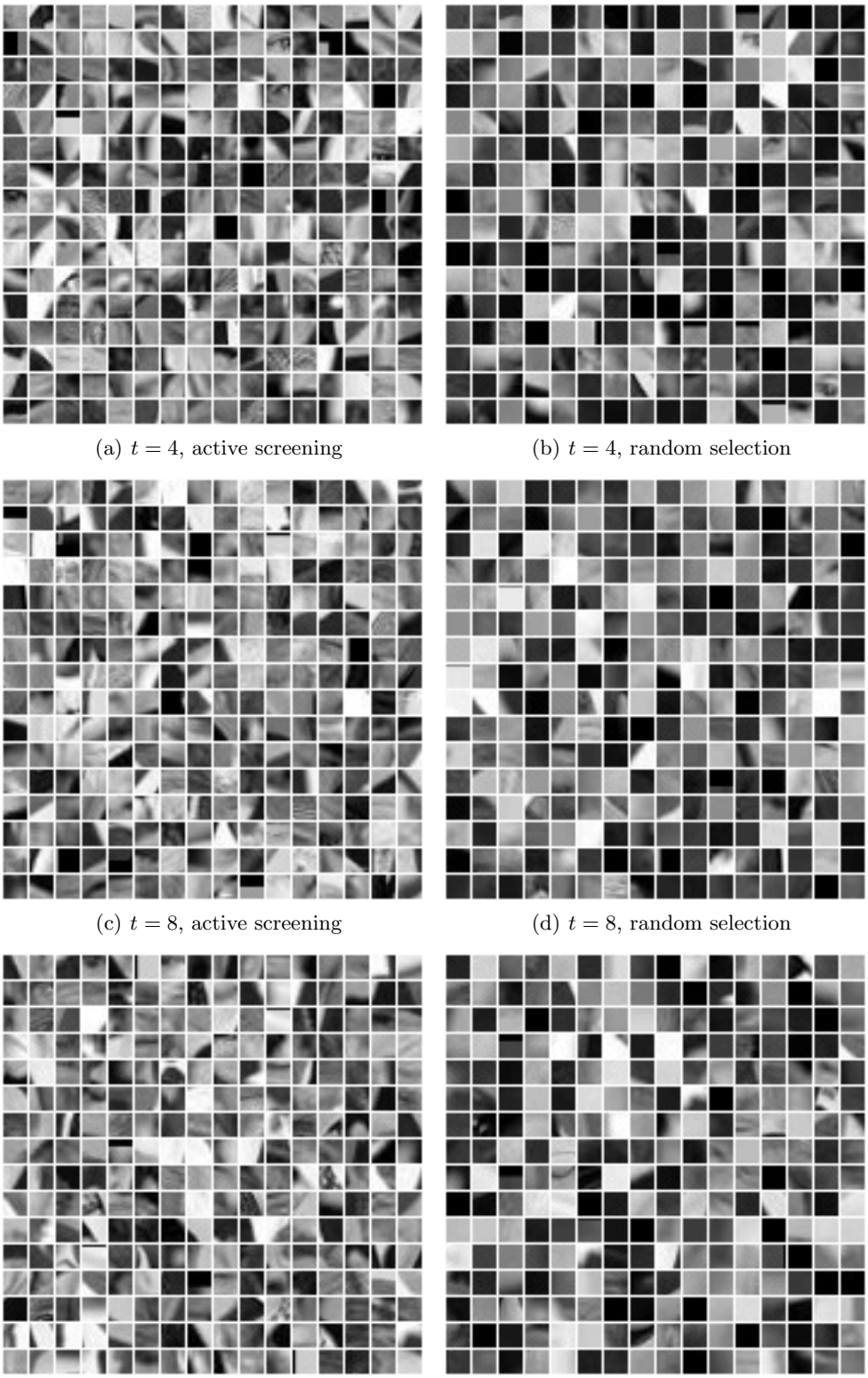
(c) $t = 20$, active screening

(d) $t = 20$, random selection

(e) $t = 30$, active screening

(f) $t = 30$, random selection

Figure 3. Some selected patches using active screening ($\lambda = 0.2$) and random selection for Extended Yale B dataset.



(a) $t = 4$, active screening

(b) $t = 4$, random selection

(c) $t = 8$, active screening

(d) $t = 8$, random selection

(e) $t = 12$, active screening

(f) $t = 12$, random selection

Figure 4. Some selected patches using active screening ($\lambda = 0.2$) and random selection for LFW dataset.