HIGH-DIMENSIONAL LEARNING FROM RANDOM PROJECTIONS OF DATA THROUGH REGULARIZATION AND DIVERSIFICATION

By

Mohammad Aghagolzadeh

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering—Doctor of Philosophy

2015

ABSTRACT

HIGH-DIMENSIONAL LEARNING FROM RANDOM PROJECTIONS OF DATA THROUGH REGULARIZATION AND DIVERSIFICATION

By

Mohammad Aghagolzadeh

Random signal measurement, in the form of random projections of signal vectors, extends the traditional point-wise and periodic schemes for signal sampling. In particular, the well-known problem of sensing sparse signals from linear measurements, also known as Compressed Sensing (CS), has promoted the utility of random projections. Meanwhile, many signal processing and learning problems that involve parametric estimation do not consist of sparsity constraints in their original forms. With the increasing popularity of random measurements, it is crucial to study the generic estimation performance under the random measurement model. In this thesis, we consider two specific learning problems (named below) and present the following two generic approaches for improving the estimation accuracy: 1) by adding relevant constraints to the parameter vectors and 2) by diversification of the random measurements to achieve fast decaying tail bounds for the empirical risk function.

The first problem we consider is Dictionary Learning (DL). Dictionaries are extensions of vector bases that are specifically tailored for sparse signal representation. DL has become increasingly popular for sparse modeling of natural images as well as sound and biological signals, just to name a few. Empirical studies have shown that typical DL algorithms for imaging applications are relatively robust with respect to missing pixels in the training data. However, DL from random projections of data corresponds to an ill-posed problem and is not well-studied. Existing efforts are limited to learning structured dictionaries or dictionaries for structured sparse

representations to make the problem tractable. The main motivation for considering this problem is to generate an adaptive framework for CS of signals that are not sparse in the signal domain. In fact, this problem has been referred to as 'blind CS' since the optimal basis is subject to estimation during CS recovery. Our initial approach, similar to some of the existing efforts, involves adding structural constraints on the dictionary to incorporate sparse and autoregressive models. More importantly, our results and analysis reveal that DL from random projections of data, in its unconstrained form, can still be accurate given that measurements satisfy the diversity constraints defined later.

The second problem that we consider is high-dimensional signal classification. Prior efforts have shown that projecting high-dimensional and redundant signal vectors onto random low-dimensional subspaces presents an efficient alternative to traditional feature extraction tools such as the principle component analysis. Hence, aside from the CS application, random measurements present an efficient sampling method for learning classifiers, eliminating the need for recording and processing high-dimensional signals while most of the recorded data is discarded during feature extraction. We work with the Support Vector Machine (SVM) classifiers that are learned in the high-dimensional ambient signal space using random projections of the training data. Our results indicate that the classifier accuracy can be significantly improved by diversification of the random measurements.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

KEY TO ABBREVIATIONS

AR:	Autoregressive
BCS:	Blind Compressed Sensing
CS:	Compressed Sensing
CM:	Concentration of Measure
DL:	Dictionary Learning
HSI:	Hyperspectral Imaging
HSC:	Hyperspectral Cube
LASSO:	Least Absolute Shrinkage and Selection Operator
MAP:	Maximum A Posteriori
MMV:	Multiple Measurement Vector
MRI:	Magnetic Resonance Imaging
OAO:	One Against One
OAA:	One Against All
PSNR:	Peak Signal to Noise Ratio
PRESS:	Projected Residual Error Sum of Squares
RIP:	Restricted Isometry Property
SVM:	Support Vector Machine
TRESS:	True Residual Error Sum of Squares

CHAPTER 1

INTRODUCTION

Cameras are being integrated into smartphones, tablet devices and the new trend of wearable consumer devices. This calls for low-cost low-rate image sampling methods as opposed to traditional full-pixel sampling. Some of the other scenarios in which the sampling efficiency becomes critical are in medical imaging where radiation dosage must be kept minimal for patient safety, in hyperspectral imaging where it is not feasible to sample every electromagnetic frequency at every pixel due to the slow scanning process and in wireless sensor networks where the sampling rate and the signal transmission rate are limited by the sensor power and complexity.

Efficient sensing not only uses fewer samples, it also exploits the inherent structure of natural signals throughout the recovery process. For example, a property of natural images that distinguishes them from random vectors is that they can be closely approximated by a sparse vector through linear transformation. The Compressed Sensing (CS) theory [1, 2] and its extensions provide the bounds for the minimum number of linear measurements as well as tractable recovery algorithms for the perfect or near perfect recovery of sparse signals.

Other than signal recovery, many applications involve learning or estimating model parameters that are used for, for example, event detection or classification. In particular, in some of these applications, the parameters are constantly being adapted to the new incoming signal. In such tasks, efficient sampling becomes crucial since the system is usually bounded in terms of energy, size or the processing power. Specifically, in very low sampling rates, signal recovery may be infeasible or even unnecessary. In fact, with a careful design of the learning system, it is possible to bypass the signal recovery and directly exploit the (incomplete) measurements for estimation, resulting in a more efficient learner. However, it remains to analyze the performance and accuracy of general learning from such incomplete measurements.

In this thesis, we consider the class of linear measurements. Specifically, we work with *random* linear measurements which have been promoted by the CS theory because they have smaller *restricted isometry constants* as defined in the CS theory [1] compared to periodic point-wise sampling. Our goal, in addition to high quality signal recovery, is to employ these random measurements in later stages of learning that normally take fully sampled signals as input.

In many cases, learning from incomplete data corresponds to an inverse problem. Unfortunately, learning from incomplete data poses the risk of obtaining an ill-posed inverse problem which would result in an infinite number of solutions for the optimal parameters. A conventional technique to avoid such ill-posed scenarios is through regularization, i.e. constraining the solution space of the problem to make it less ill-posed. These constraints are problem-specific and usually reflect the natural constraints of the parameter space. For example, sparsity is a constraint that is added to the problem of natural image recovery as in CS. Regularization is one of the techniques considered in this thesis for making high-dimensional learning from random projections of data less ill-posed. We provide analytical reasons for why

regularization works in a general learning framework and how to quantify the improvements that are due to regularization.

In addition to regularization, which is a well-known technique, we propose a novel technique that we believe has yet to receive attention from the signal processing and machine learning communities; namely, measurement diversification. In contrast to regularization which modifies the learning problem in a specific fashion, diversification works with the original problem but requires the measurements to have diversity; that is random measurements from different signals must have different supports. We show that diversification reduces the ill-posedness of the inverse problem without the need to introduce new constraints over the parameters. Our results are confirmed both empirically and analytically through the theories of the concentration of measure [3].

To make our analysis and presentation more concrete, we work with two well-known learning problems: 1) dictionary learning and 2) signal classification. These two problems are briefly discussed in the following paragraphs.

Dictionaries extend the notion of orthogonal signal bases that allow for sparser signal representation using data-driven redundant frames. For example, the method of K-SVD [4] extends the traditional Singular Value Decomposition (SVD) for extracting basis vectors from empirical data. A typical Dictionary Learning (DL) algorithm takes a set of training signal vectors as input and generates a dictionary that can be generalized to the testing data as well. However, there are other scenarios where the testing and training data are the same except that the training data is noisy and the goal is to approximate the testing data using sparse representation with respect to the learned dictionary [5]. This approach is motivated by the

assumption that the complex noise structure does not enter the learned dictionary and gets attenuated during DL.

In this thesis, we consider a novel application of DL. Specifically, the training data for our problem consists of random low-dimensional projections of signal vectors and the testing data is the set of original signal vectors (in the ambient signal space). In other words, the purpose is to learn a dictionary for a set of signal vectors while only random projections of those vectors are available. It is not hard to show that this learning corresponds to an ill-posed inverse problem. However, there is a strong motivation for this problem which is to generate an adaptive framework for CS-based signal recovery (from random projections of signals). Contrary to conventional (non-adaptive) CS, which uses a fixed basis or frame for signal representation, our framework employs a flexible representation that adapts to the specific signal structure.

The other problem that we consider is high-dimensional signal classification. Similar to the described DL problem, we consider a novel scenario for learning a linear classifier from random projections of the data. Conventional classifier learning takes a set of (high-dimensional) training signals with labels as input and generates a classifier that generalizes to the testing data. It must be noted that high-dimensional training data is usually mapped to the feature space before training. Meanwhile, in the problem that we consider, only random projections of the training data (with labels) are available. The testing data is also provided in the random projection domain. A particular feature of our framework is that the classifier is trained in the high-dimensional ambient space, rather than in the low-dimensional measurement space. We study the application of linear signal classification for hyperspectral pixel classification where each pixel is composed of hundreds of spectral components.

RELATED WORK

Similar problems to the described DL problem have been proposed before under the name of Blind compressed sensing (BCS) [6]. BCS is defined as an extension of CS where the optimal sparse representation basis is assumed to be unknown and subject to estimation during signal recovery. However, albeit the importance of such problem, existing works in BCS are limited. The original BCS framework focused on making the ill-posed problem tractable by adding constraints to the dictionary. Some of the proposed schemes were sparse dictionaries, block-diagonal dictionaries and a dictionaries that are variable sets of fixed bases. Their following work [7] utilized some of the theories from the area of low-rank matrix completion [8] to show that BCS with unconstrained dictionaries can still be tractable if the sparse coefficient matrix is group-sparse, which can be regarded as low-rank sparse matrix. Interestingly, the low-rank matrix completion theory requires that random projections of data blocks be distinct which is strongly related to the diverse sampling requirement proposed in our work here. A BCS-related work was presented in [9] which is the closest work to our proposed unconstrained DL. However, the mentioned work is purely empirical and does not give any sort of analysis for why such method works. Yet another related work is the method of best basis compressed sensing [10] where the representation basis is selected according to a tree structure from a highly overcomplete structured dictionary during the signal recovery. Best basis is defined as the basis that minimizes the distance between the compressive measurements and the sparse signal representation (similar to DL). An advantage of the best basis representation over naive signal representation using overcomplete frames is that the selected bases have low-coherency. However, best basis CS is constrained to a finite set of basis vectors for signal representation.

Reviewing the relevant works for the signal classification problem is clearly a more difficult task due to the vast amount of existing work in this area. Probably the most relevant work would be the framework of compressed learning [11] where the accuracy of the linear SVM classification has been analyzed under the scheme of having only random projections of data. Compressed learning states that linear classification in the low-dimensional projection domain, with a high probability, has an accuracy close to the accuracy of the linear classification in the original (high-dimensional) signal domain. However, as we demonstrate later, linear classification in the projection domain can be unreliable when the random projection is very low-dimensional. Indeed the reason for the proposed diverse measurement scheme is to make linear classification reliable even in the presence of a single measurement per data point.

THESIS ORGANIZATION

This section provides a brief summary of the upcoming chapters and the main contributions in each chapter. However, before that, we need to specify two types of random measurements that we frequently refer to: repeated-block sampling and random-block sampling. In our data model, the data or the signal is composed of N blocks. For example, for the image data, which is the main type of data considered in his work, blocks correspond to small rectangular image patches. In the repeated-block sampling scheme, each block is sampled according to the same pattern. For example, the traditional periodic sampling can be considered a special case of a repeated-block scheme. Random-block sampling uses a different sampling pattern for each block.

To give an example, assume an image were divided into patches of size 2 by 2 pixels, resulting in blocks of size 4. A well-known example of repeated-block sampling for color images would be the Bayer color filter array which is shown in Figure 1 (left). Figure 1 (right) shows a pseudorandom arrangement of six color filters corresponding to a random-block measurement.

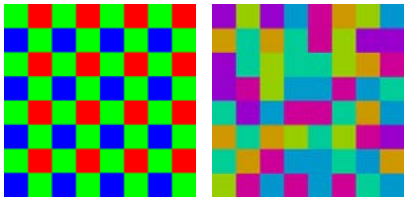


Figure 1. Repeated-block (left) vs. random-block (right) sampling for color imaging. These patterns correspond to color filter arrays that are used in consumer-level cameras for capturing a single color component per pixel. The color image is later reconstructed from these (incomplete) measurements. Each image patch is assumed to be 2 by 2 pixels in this example.

Chapter III: Sparse Regularization for DL from Repeated-Block Measurements

The first dictionary constraint that we employed for addressing the mentioned ill-posed DL problem was, what we called, the two-layered dictionary model which was adopted from the double-sparse representation model [12]. In this model, a dictionary can be factorized into the product of a fixed frame with a sparse matrix (hence the double-sparse name). Double-sparse dictionaries were originally proposed to reduce the amount of required memory for storing large dictionaries by representing each atom using only a few non-zero coefficients with respect to the overcomplete cosine frame. Learning two-layer dictionaries from random projections of data becomes a tractable inverse problem which makes it suitable for our task.

Chapter IV: Autoregressive Regularization for DL from Repeated-Block Measurements

In our second approach, we utilized non-parametric regularized regression within the DL framework to learn smooth dictionaries. In this approach, dictionary atoms are modeled as autoregressive processes with known covariance matrices that are trained over natural images. Specifically, different atoms are modeled as independent processes and there is correlation only within each atom. In this approach, we employ minimum-mean-square-error estimation to update the dictionary using random measurements and the computed sparse coefficients. An advantage of this model compared to the parametric sparse model of our first approach is its lower computational complexity.

Chapter V: Unconstrained DL from Random-Block Measurements

While repeated-block sampling necessitates the use of regularization or additional structure in the dictionary, in random-block sampling, the typical least squares learning algorithm works without much trouble. This observation suggests that random-block measurements carry more information compared to repeated-block projections when the data is correlated. This observation was our first clue to the importance of measurement diversification for general learning which is further developed below.

Chapter VI: Mathematical Analysis

Recently, it was shown that random-block projections carry nearly an equal amount of information about the underlying image as if the image was sensed using a dense projection matrix[1] [13]. Following a similar procedure for computing the recovery accuracy in the theory of compressed sensing, we characterize the (stochastic) accuracy of the dictionary learning under both repeated-block and random-block measurements. The main factors to be considered other than the number of measurements per image block are the number of blocks in the image and the strength of cross-correlation between different image blocks. We will employ some of the tools from the area of concentration of measure [3] and its extensions in [13] to compute how accurate and stable is learning in the projected domain compared to learning in the original image domain. These mathematical derivations address both repeated-block and random-block measurements.

Furthermore, we extend the analysis to generic learning where the empirical risk is used to approximate the true risk (which is unavailable to the algorithm). It desired that the learned parameters using the empirical risk closely approximate the parameters that minimize the true risk. We compute error bounds for parametric estimation given tail bounds of the distribution of the empirical risk and make strong connections with the well-known empirical risk minimization principle in learning theory.

[1] Dense random projection matrices are not very practical but present the benchmark measurement matrices for compressed sensing.

Chapter VII: Hyperspectral Classification

Numerous recent studies have promoted the utility of random hyperspectral measurement because it enables hyperspectral recovery using the low-rank or sparse recovery algorithms [33, 36]. However, many learning algorithms are not designed to work with random measurements directly. Specifically, we study the problem of hyperspectral classification using random (incomplete) measurements without the need for missing value substitution, that is without signal recovery. Learning directly from the observed data is more efficient and superior to learning from recovered data because the recovery process introduces computational overhead and possible data oversimplification due to the employed recovery model.

Diversification may initially pose as an obstacle to classification. Specifically, signal or feature vectors would have varying supports, making it difficult to approximate the pair-wise similarities of data points. We show that not only such diversity does not harm the classification performance, the learned classifier is more reliable and closer to the ground-truth classifier compared to a classifier that is learned from random-block measurements. For hyperspectral classification, we selected to work with the linear support vector machine.

CHAPTER 2

BACKGROUND AND PROBLEM FORMULATION

In this chapter we present the required mathematical background for the following chapters. We also give formal definitions for the problems of compressed sensing, dictionary learning, dictionary adaptation for compressed sensing, linear SVM classification and their associated challenges. We start by introducing our notation in this dissertation.

NOTATION

Upper-case letters are used for matrices and lower-case letters are used for vectors and scalars. For a matrix A, a_j denotes its jth column, $\bar{a}_i$ denotes its ith row and $a_{ij} = [A]_{ij}$. For the dictionary learning problem, we reserve the letters listed in the following table (which are properly defined later in this chapter).

Table 1. List of reserved letters and notation for the DL problem.

Letter	Reserved for
N	Total number of signal blocks
n	Dimension of each block
m	Number of measurements per block
d	Number of columns in the dictionary
t	Iteration number
$D \in \mathbb{R}^{n \times d}$	Dictionary
$X = [x_1 \dots x_N]$ where $x_j \in \mathbb{R}^n$	Signal/data matrix (each column is a block)
$A = [a_1 \dots a_N]$ where $a_j \in \mathbb{R}^d$	The coefficient matrix

The vector ℓ_p norm is defined as:

$$\|x\|_p = \left(\sum_i |x_i|^p \right)^{\frac{1}{p}}$$

The matrix operator $A \otimes B$ represents the Kronecker product (also known as outer product, or tensor product) which is defined as (for $A \in \mathbb{R}^{m \times n}$):

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}$$

The operator $A \odot B$ represents the Hadamard product (also known as the element-wise product) which is defined as (for $A, B \in \mathbb{R}^{m \times n}$):

$$A \odot B = \begin{bmatrix} a_{11}b_{11} & \cdots & a_{1n}b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1}b_{m1} & \cdots & a_{mn}b_{mn} \end{bmatrix}$$

The operator $vec(A)$ reshapes a matrix to its column-major vectorized format. That is, if $A = [a_1 \dots a_n] \in \mathbb{R}^{m \times n}$, $vec(A) = [a_1^T \dots a_n^T]^T$.

The vector inner product is extended to matrices as $\langle A, B \rangle = Tr(A^T B)$ where $Tr(A)$ denotes the matrix trace, i.e. the sum of A's diagonal entries.

The Frobenius norm of $A \in \mathbb{R}^{m \times n}$ is defined as:

$$\|A\|_F = \left(\sum_{ij} |a_{ij}|^2 \right)^{\frac{1}{2}}$$

LINEAR MEASUREMENT MODEL

In the linear class of signal measurement operators, each measurement y_i is a linear function of the signal values. The collective set of measurements can be expressed as a system of linear equations:

$$\begin{cases} \phi_{11}x_1 + \phi_{12}x_2 + \cdots + \phi_{1n}x_n = y_1 \\ \phi_{21}x_1 + \phi_{22}x_2 + \cdots + \phi_{2n}x_n = y_2 \\ \vdots \\ \phi_{m1}x_1 + \phi_{m2}x_2 + \cdots + \phi_{mn}x_n = y_m \end{cases}$$

or more compactly:

$$\Phi x = y$$

where Φ represents an $m \times n$ sampling matrix. In under-sampling scenarios, where we have a smaller number of linear measurements m compared to the signal ambient dimension n, resulting

in ‘short and fat’ sampling matrices. As a consequence, the solution space for $\hat{x}$ such that $\Phi\hat{x} = y$, which is basically a translated copy of the null space of Φ, has an infinite cardinality. This makes the inverse problem ill-posed without additional information about the underlying signal x.

For example, periodic point-wise sampling or point-sampling[2] is a special case of linear measurement where there is only a single non-zero entry in each row of the sampling matrix. Another scheme is when the signal is convolved with a low-pass filter (anti-aliasing filter) before the point-wise sampling. Generally speaking, such periodic measurement operations result to structured circulant sampling matrices. Meanwhile, recent advances in signal sensing and recovery suggest that randomly generated sampling matrices are very efficient for sensing sparse signals. Perhaps a more accurate term for these randomly generated sampling matrices is pseudorandom matrices. However, for simplicity and similar to most other works, we use the term ‘random sampling matrices’. Random sampling matrices are proven to be superior (in terms of the sensing efficiency) to conventional point-sampling matrices when no prior knowledge is assumed about the underlying sparse signal. Random sampling matrices are usually assumed to be generated using a random (independent and identically distributed) Gaussian distribution while many other distributions, including Rademacher and centered and bounded distributions, have proven to perform similarly as well [14].

[2] In point-wise sampling the signal value is captured once in every few samples, using periodically.

COMPRESSED SENSING

Compressed Sensing (CS) in its simplest form is an inverse problem in which we are given a set of underdetermined linear measurements of x in the form $\Phi x = y$ and we are asked to find the sparsest solution $\hat{x}^*$ that adheres to the measurements. However, instead of searching the solution space of $\Phi\hat{x} = y$ for the solution $\hat{x}$ with the minimum number of non-zeros which turns out to be an NP-hard problem, CS suggests minimizing a *relaxed* function to replace the non-convex sparsity objective [1]. The convex objective is simply the sum of absolute values of the signal or the ℓ_1-norm of the signal:

$$\|\hat{x}\|_1 = \sum_{i=1}^{n} |\hat{x}_i|$$

It is conventional to say that the ℓ_1-norm relaxes the ℓ_0-(pseudo)norm which is defined as the number of non-zero signal values. The proposed relaxation, however, is only valid when the sampling matrix Φ satisfies a condition known as the Restricted Isometry Property or RIP [1, 15] described below.

Definition 2.1 *[15] For each integer $s = 1,2, \ldots$, define the restricted isometry constant δ_s of a matrix Φ to be the smallest number such that*

$$(1 - \delta_s)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta_s)\|x\|_2^2 \qquad (2.1)$$

holds for all s-sparse vectors x. A vector is said to be s-sparse if it has at most s non-zero entries. The matrix Φ is said to satisfy the s-restricted isometry property with restricted isometry constant δ_s.

Note that $\delta_1 \leq \delta_2 \leq \delta_3 \leq \cdots$ for any Φ, meaning that the isometry constant becomes worst as the signal becomes denser. For example, $\delta_{2s} < 1$ guarantees that the ℓ_0-minimization problem has a unique solution for s-sparse vectors but does not give acceptable recovery guarantees for the ℓ_1-minimization problem. A tighter bound $\delta_{2s} < \sqrt{2} - 1$ guarantees that the solution to the ℓ_1-minimization problem is exactly the same as the solution to the ℓ_0-minimization problem [15]. Moreover, the ℓ_1-based recovery can be proven to be very stable with respect to noise[3].

Define the noisy CS problem as:

$$\hat{x}^* = \arg\min_{\hat{x}} \|\hat{x}\|_1 \quad \text{subject to} \quad \|y - \Phi\hat{x}\|_2 < \epsilon \tag{2.2}$$

which can also be written as the LASSO optimization problem [16]:

$$\hat{x}^* = \arg\min_{\hat{x}} \frac{1}{2}\|y - \Phi\hat{x}\|_2^2 + \lambda\|\hat{x}\|_1 \tag{2.3}$$

for a proper choice of λ which is described in [17]. Candès, in his short and elegant notes [15], has reformulated the following theorem about the stability of the CS recovery under additive noisy measurements $y = \Phi x + z$:

Theorem 2.1 *[15] Assume that $\delta_{2s} < \sqrt{2} - 1$ and $\|z\|_2 < \epsilon$. Then the noisy CS solution (2.2) obeys:*

$$\|\hat{x}^* - x\|_2 \leq C_0 s^{-\frac{1}{2}}\|x - x_s\|_1 + C_1\epsilon \tag{2.4}$$

with small constants C_0 and C_1 that are explicitly given in [15] and x_s denoting the best s-sparse approximation of x by keeping the largest s entries of x.

[3] In this context, usually, noise refers to the small non-zero entries in the signal. A noisy sparse signal x can be written as the sum of a s-sparse noiseless signal x_s and a low-energy noise with $\|x - x_s\|_2 < \epsilon$.

The original theories of CS have been extended significantly over the past decade for different types of measurement noise, non-sparsity noise, structured sparsity models which make CS more robust in real-world scenarios. However, reviewing these works is beyond the scope of this dissertation or any single work.

DICTIONARY LEARNING

Standard formulation

Let $x = Da$ denote the representation of signal $x \in \mathbb{R}^n$ with respect to the dictionary $D \in \mathbb{R}^{n \times d}$ (here d represents the number of columns or *atoms* in the dictionary) with the coefficient vector $a \in \mathbb{R}^d$. Specifically when $d > n$, the dictionary is called *overcomplete* (or sometimes called redundant), as opposed to orthogonal bases which are called complete. This naming convention is due to the fact that overcomplete dictionaries, not only span the whole signal space, allow for various representations of the same signal and eventually sparser representations of signals can be obtained using convex optimization algorithms.

As an alternative to model-based and mathematically induced dictionaries such as wavelets, Dictionary Learning (DL) [18] is a data-driven and algorithmic approach to build sparse representations for natural signals. Let $X = [x_1, x_2, \dots, x_N] \in \mathbb{R}^{n \times N}$ denote the data matrix which consists of N n-dimensional (training) signals and let $x_j = Da_j$ denote the representation of block j. Expressed in a matrix form, $X = DA$ with $A = [a_1, a_2, \dots a_N] \in \mathbb{R}^{d \times N}$. In a DL

problem, we are given the training data matrix X and are asked to find a dictionary that minimizes the sum of squared errors for the sparse representation:

$$(D^*, a_1^*, a_2^*, \dots, a_N^*) = \arg \min_{D, a_1, a_2, \dots, a_N} \sum_{j=1}^{N} \frac{1}{2} \left\| x_j - D a_j \right\|_2^2 \text{ subject to } \forall j: \left\| a_j \right\|_0 \leq k \quad (2.5)$$

To make the problem more tractable, the ℓ_0 norm may be replaced by the ℓ_1 norm and combined with the objective function using the Lagrangian method:

$$(D^*, a_1^*, a_2^*, \dots, a_N^*) = \arg \min_{D, a_1, a_2, \dots, a_N} \sum_{j=1}^{N} \frac{1}{2} \left\| x_j - D a_j \right\|_2^2 + \lambda \sum_{j=1}^{N} \left\| a_j \right\|_1 \quad (2.6)$$

The matrix D is typically assumed to have unit-norm columns as in orthonormal bases and preventing unbounded solutions for D. This formulation of DL, along with the matrix factorization formulation presented below, correspond to very high-dimensional non-convex optimization problems. Therefore, it is extremely difficult to solve the DL problem in this form. The following sections describe bilevel formulation of DL which can be efficiently solved using convex optimization.

Matrix factorization formulation

Essentially, DL for exact signal representation is a matrix factorization problem where the data matrix X is represented as the product of the matrix D and a sparse matrix A. In mathematical terms:

$$X = DA \text{ subject to } \|vec(A)\|_0 < K \quad (2.7)$$

where $vec(A) = [a_1^T, a_2^T, \dots, a_N^T]^T$ represents the column-major vectorized format of the matrix A. In read-world applications, A consists of a few large entries (in terms of their magnitudes) and many (relatively) small entries. To address this *noisy* scenario and the issue of the non-convexity of the ℓ_0 norm, the above DL formulation is typically *relaxed* by replacing the ℓ_0 norm with the convex ℓ_1 norm and using the Lagrangian method:

$$(D^*, A^*) = \arg\min_{D,A} \frac{1}{2}\|X - DA\|_F^2 + \lambda\|vec(A)\|_1 \text{ subject to } \|D\|_F \leq \sqrt{n} \qquad (2.8)$$

The constraint $\|D\|_F \leq \sqrt{n}$ (bound on the dictionary norm) prevents the trivial solution of $D \to \infty^{d\times n}$ and $A \to 0^{n\times N}$. Similar to the previous formulation, the above problem represents a non-convex optimization with respect to the (D, A) tuple [19].

Although equivalent to the standard DL formulation presented in the previous section, the matrix factorization formulation presents a more concise formulation and is sometimes preferred. For the remaining of this dissertation we use them interchangeably depending on the context.

Bilevel formulation and optimization

A typical remedy to the non-convexity of the DL problem (for example see [20]) is to write the DL problem as a bilevel optimization problem where both its *lower-level* and *upper-level* problems are convex:

$$\begin{cases} \text{lower level:} & A^* = \arg\min_{A} \frac{1}{2}\|X - D^*A\|_F^2 + \lambda\|vec(A)\|_1 \\ \text{upper level:} & D^* = \arg\min_{D} \frac{1}{2}\|X - DA^*\|_F^2 \;\; subject\ to\ \|D\|_F \leq \sqrt{n} \end{cases} \qquad (2.9)$$

Numerically, the bilevel DL problem is solved by alternating between the lower-level and upper-level problems:

$$\begin{cases} A^{(t+1)} = \arg\min_{A} \frac{1}{2}\left\|X - D^{(t)}A\right\|_F^2 + \lambda\|vec(A)\|_1 \\ D^{(t+1)} = \arg\min_{D} \frac{1}{2}\left\|X - DA^{(t+1)}\right\|_F^2 \ subject\ to\ \|D\|_F \le \sqrt{n} \end{cases} \tag{2.10}$$

This can be accomplished by alternating between the two steps:

1) *Sparse coding* (lower-level): by fixing $D^{(t)}$ and optimizting with respect to $A^{(n+1)}$. This represents a Lasso optimization problem which can be solved efficiently, for example, using the least angle regression algorithm [16].

2) *Dictionary update* (upper-level): by fixing $A^{(n+1)}$ and optimizing with respect to $D^{(t+1)}$. This represents a constrained quadratic optimization problem which can be solved efficiently, for example, using gradient descent algorithms.

Note that, although individual optimization problems for individual variables have convex objective functions, the combined DL objective is not convex and the global optimum cannot be reached using greedy algorithms. Still, in most cases, even a local optimum represents a favorable solution compared to the starting point.

DL FROM RANDOM PROJECTIONS OF DATA

Recall that, in repeated-block measurement, all columns (or blocks) of the data matrix X are measured using the same measurement matrix. Given these measurements, expressed in the matrix form $Y = \Phi X$, the DL problem must be modified as follows:

$$\left(\widehat{D}^*, \hat{A}^*\right) = \arg\min_{D,A} \frac{1}{2}\|\mathrm{Y} - \Phi DA\|_F^2 + \lambda\|vec(A)\|_1 \text{ subject to } \|D\|_F \leq \sqrt{n} \quad (2.11)$$

The bilevel form of the above problem follows:

$$\begin{cases} \text{lower level:} \quad \hat{A}^* = \arg\min_{A} \frac{1}{2}\left\|Y - \Phi\widehat{D}^* A\right\|_F^2 + \lambda\|vec(A)\|_1 \\ \text{upper level:} \quad \widehat{D}^* = \arg\min_{D} \frac{1}{2}\left\|Y - \Phi D\hat{A}^*\right\|_F^2 \;\; subject\ to\ \|D\|_F \leq \sqrt{n} \end{cases} \quad (2.12)$$

Note that the lower-level problem correspond to a noisy CS problem in the form of LASSO [16]. Therefore, in this thesis, we focus on the upper-level (dictionary update) problem since the lower-level problem is a relatively well-studied problem.

When considering the random-block measurements, it is more convenient to write the DL problem in a block-wise format:

$$(D^*, A^*) = \arg\min_{D,A} \sum_{j=1}^{N} \frac{1}{2}\left\|x_j - Da_j\right\|_F^2 + \lambda\left\|a_j\right\|_1 \quad (2.13)$$

This way, DL with general (random or repeated) block measurements becomes:

$$\left(\widehat{D}^*, \hat{A}^*\right) = \arg\min_{D,A} \sum_{j=1}^{N} \frac{1}{2}\left\|y_j - \Phi_j Da_j\right\|_F^2 + \lambda\left\|a_j\right\|_1 \quad (2.14)$$

subject to $\|D\|_F \leq \sqrt{n}$.

Finally, note that block measurements $\forall j$: $y_j = \Phi_j x_j$ could be expressed as:

$$vec\ (Y) = \begin{bmatrix} \Phi_1 & & \\ & \ddots & \\ & & \Phi_N \end{bmatrix} vec(X)$$

where the overall measurement matrix is block-diagonal.

SUPPORT VECTOR MACHINE CLASSIFICATION

Classification is task of assigning categorical labels to the input signals based on some decision rule. Most classifiers are inherently composed of binary decision rules. Specifically, in multi-categorical classification, multiple binary classifiers are trained according to either One-Against-All (OAA) or One-Against-One (OAO) schemes and voting techniques are employed to combine the results [21]. For example, in a OAA linear Support Vector Machine (SVM) classification problem, an affine decision hyperplane is computed between each class and the rest of the training data, while in a OAO scheme, a hyperplane is learned between each pair of classes. As a consequence, most studies focus on the canonical binary classification. Similarly in here, our analysis is presented for the binary classification problem which can be extended to multi-categorical classification.

In the linear SVM classification problem [22, 23], we are given a set of training data points (each corresponding to a hyperspectral pixel) $x_j \in \mathbb{R}^d$ for $j \in \{1,2,\dots,N\}$ and the associated labels $z_j \in \{-1,+1\}$. The inferred class label for x_j is $sign\left(x_j^T \omega - b\right)$ that depends on the classifier $\omega \in \mathbb{R}^d$ and the bias term $b \in \mathbb{R}$. The classifier ω is the normal vector to the affine hyperplane that divides the training data in accordance with their labels. The mapping $x_j \to sign\left(x_j^T \omega - b\right)$ can be regarded as dividing the feature space into two partitions using the hyperplane $x^T \omega = b$ and making a binary decision about x_j by observing which side of the learned hyperplane it lies on. The maximum-margin SVM classifier can be expressed as the following optimization problem:

$$(\omega^*, b^*) = \arg\min_{\omega,b} \|\omega\|_2$$

Subject to

$$\forall j \in \{1,2,\dots,N\}: \qquad z_j\left(x_j^T\omega - b\right) \geq 1 \tag{2.15}$$

Note that minimizing $\|\omega\|_2$ is effectively equivalent to maximizing the margin which is defined as the distance between the learned hyperplane and the closest x_j.

Unfortunately, it is not always possible to find an affine hyperplane that perfectly divides the training data in accordance with their labels which makes the above hard-margin SVM problem infeasible for some data. When the training classes are inseparable by an affine hyperplane, maximum-margin soft-margin SVM is used which relies on a loss function to penalize the amount of misfit. For example, a widely used loss function is $\ell(r) = (\max\{0,1 - r\})^p$ with $r = z_j\left(x_j^T\omega - b\right)$. For $p = 1$, this loss function is known as the hinge loss, and for $p = 2$, it is called the squared hinge loss or simply the quadratic loss. The optimization problem for soft-margin SVM becomes

$$(\omega^*, b^*) = \arg\min_{\omega,b} \frac{1}{n}\sum_{j=1}^{n} \ell\left(z_j\left(x_j^T\omega - b\right)\right) + \frac{\lambda}{2}\|\omega\|_2^2 \tag{2.16}$$

Similar results can be obtained using the dual form [24]. Recent works have shown that advantages of the dual form can be obtained in the primal as well [24] where it is noted that the primal form convergences faster to the optimal parameters (ω^*, b^*) than the dual form. For the purposes of our work, it is more convenient to work with the primal form of SVM although the analysis can be properly extended to the dual form.

A well-known constrained formulation for the soft-margin SVM problem, which is closer to the its original formulation, is obtained by adding a set of N slack variables ξ_j and solving the following constrained optimization problem

$$(\omega^*, b^*) = \arg\min_{\omega,b,\xi} \|\omega\|_2^2 + C \sum_{j=1}^{N} \xi_j^p$$

Subject to

$$\forall j \in \{1,2,\dots,N\}: \quad z_j\left(x_j^T \omega - b\right) \geq 1 - \xi_j, \quad \xi_j \geq 0 \tag{2.17}$$

The advantage of this formulation is that it represents a quadratic program that can be solved efficiently using off-the-shelf software packages.

CHAPTER 3

SPARSE REGULARIZATION FOR DL FROM REPEATED-BLOCK MEASUREMENTS

The ultimate goal of this chapter is to improve the sparse image reconstruction from repeated-block measurements through real-time optimization of the dictionary. We first introduce the two-layer dictionary structure as it appeared in [12] for the first time. Next, we describe an efficient dictionary learning approach customized to the specific structure of the dictionary. Finally, we evaluate the performance of the proposed dictionary learning algorithm as a function of the number of acquired measurements. This chapter is based on our work [25].

THE TWO-LAYER DICTIONARY MODEL

A two-layer dictionary is the product of a fixed frame Ψ, called the base dictionary, with a sparse parameter matrix Θ:

$$D = \Psi\Theta$$

Examples of the base dictionary include the overcomplete cosine frame, the real Fourier frame and undecimated wavelet frames all of which can be efficiently used for signal representation. A layer of adaptivity is added to the signal representation by multiplying the base dictionary with a tunable matrix Θ. This matrix, which makes up the outer layer of the dictionary, is constrained to be sparse to reduce the amount of memory that is required for storing the dictionary and also reduce the computational burden of subsequent matrix multiplications [12]. It is usually assumed that Θ is a $d \times d$ square matrix and Ψ is a $n \times d$ frame. For complete frames $n = d$.

OPTIMIZATION OF TWO-LAYER DICTIONARIES

Optimization of the two-layer dictionaries is achieved by minimizing the total representation error for a specific level of sparsity in the parameter matrix Θ. Formally, the original problem of two-layer dictionary learning problem [12] was described as:

$$\min_{\Theta,A} \frac{1}{2}\sum_{j=1}^{N}\left\|x_j - \Psi\Theta a_j\right\|_2^2 \quad \text{subject to} \quad \begin{cases}\forall i \in \{1,2,\dots,d\}: \|\theta_i\|_0 \le t \\ \forall j \in \{1,2\dots,N\}: \left\|a_j\right\|_0 \le k\end{cases} \tag{3.1}$$

However, we are interested in a relaxed and more tractable form of this problem by replacing the non-convex sparsity constraints with convex ℓ_1-norm constraints. Using the Lagrangian method for constrained optimization with fixed γ and λ:

$$\min_{\Theta,A} \frac{1}{2}\sum_{j=1}^{N}\left\|x_j - \Psi\Theta a_j\right\|_2^2 + \gamma\sum_{i=1}^{d}\|\theta_i\|_1 + \lambda\sum_{j=1}^{N}\left\|a_j\right\|_1 \tag{3.2}$$

The bilevel formulation for this problem becomes:

$$\begin{cases} \text{lower level:} & A^* = \arg\min_{A} \frac{1}{2}\sum_{j=1}^{N}\left\|x_j - \Psi\Theta^* a_j\right\|_2^2 + \lambda\left\|a_j\right\|_1 \\ \text{upper level:} & \Theta^* = \arg\min_{\Theta} \frac{1}{2}\sum_{j=1}^{N}\left\|x_j - \Psi\Theta a_j^*\right\|_2^2 + \gamma\sum_{i=1}^{d}\|\theta_i\|_1 \end{cases} \tag{3.3}$$

Hence, after computing the coefficient matrix in iteration t of the DL algorithm (using LASSO), we can again use the LASSO optimization to update the dictionary parameters. However, one can see with a close inspection that the dictionary update problem is still not in the typical LASSO form and needs to be rearranged. The rearrangement is explained in the following section.

OBTAINING THE LASSO FORM FOR THE UPPER LEVEL

Assume that the columns of the parameter matrix Θ are sequentially updated. To update θ_i (column i of the parameter matrix), we must solve the following optimization problem:

$$\min_{\theta_i} \frac{1}{2}\sum_{j}^{N}\left\|e_j^i - \Psi\theta_i a_{ij}\right\|_2^2 + \gamma\|\theta_i\|_1 \tag{3.4}$$

where e_j^i is defined as the error associated with the atom number i of the dictionary for signal number j and is computed as: $e_j^i = x_j - \sum_{\ell\neq i}^{d}\Psi\theta_\ell a_{\ell j}$. Also define E^i as the matrix of error vectors for atom number i with columns e_j^i. Using [12, Lemma 1] we can show that the optimization problem of (3.2) can be reduced to:

$$\min_{\theta_i} \frac{1}{2}\left\|E^i\bar{a}_i^T - \Psi\theta_i\right\|_2^2 + \gamma\|\theta_i\|_1 \tag{3.5}$$

which has the typical LASSO form.

We should note that we update each θ_i independently of the rest of columns of Θ which means the coefficient matrix A is assumed fixed during the dictionary update. Unlike the direct

least squares approach of dictionary learning reviewed in Section 2.4, the optimization problem (3.5) represents an inverse problem of finding the parameters that characterize the dictionary.

DL FROM REPEATED-BLOCK MEASUREMENTS

The bilevel formulation for DL from repeated-block measurements becomes:

$$\begin{cases} \text{lower level:} & \hat{A}^* = \arg\min_{A} \frac{1}{2}\sum_{j=1}^{N} \left\| x_j - \Phi\Psi\widehat{\Theta}^* a_j \right\|_2^2 + \lambda \left\| a_j \right\|_1 \\ \text{upper level:} & \widehat{\Theta}^* = \arg\min_{\Theta} \frac{1}{2}\sum_{i=1}^{d} \left\| \hat{E}^{*\,i} \bar{\hat{a}}_i^{*\,T} - \Phi\Psi\theta_i \right\|_2^2 + \gamma \|\theta_i\|_1 \end{cases} \tag{3.6}$$

where $\hat{E}^{*\,i} = \left[\hat{e}_1^{*\,i}, \hat{e}_2^{*\,i}, \dots, \hat{e}_N^{*\,i}\right]$ with $\hat{e}_j^{*\,i} = y_j - \sum_{\ell \neq i}^{d} \Phi\Psi\hat{\theta}_\ell^* \hat{a}_{\ell j}^*$.

As we mentioned before, unconstrained DL from repeated-block measurements corresponds to an ill-posed problem. Our motivation for regularization of DL through a parametric dictionary model was to reduce the amount of information required for characterizing the dictionary. Given that the parameter matrix Θ is sparse (or decays fast if sorted), we are imposing some prior information upon the dictionary which resembles a Bayesian framework. In fact, it is not difficult to show that the optimization problem (20) corresponds to a MAP estimator with a double exponential distribution for Θ. In a Bayesian framework, measurements represent observations and are deployed in a MAP estimation to infer the system parameters. Clearly, acquiring more observations results in a more accurate estimation. With no observations, $\Theta = I_d$ and the dictionary would be equal to the base dictionary Ψ.

ALGORITHM

We present the algorithm for adapting two-layer dictionaries in Algorithm 1.

Algorithm 1. The algorithm for adapting the two-layer dictionary from measurements.

Input: Base frame $\Psi_{n\times d}$, measurements $Y_{m\times N}$, the sampling matrix $\Phi_{\mathrm{m\times n}}$, LASSO regularization parameters λ, γ, number of iterations T

Outputs: learned dictionary $\widehat{D}^*$, Estimated patches $\hat{X}^*$

Initialization: $\Theta^{(0)} = I_{d\times d}$

Do for t from 0 to $T-1$:

Compute $a_j^{(t)}$ for $j = 1,2,\dots,N$ using LASSO:

$$a_j^{(t)} = \arg\min_{a_j} \frac{1}{2}\left\|y_j - \Phi D^{(t)} a_j\right\|_2^2 + \lambda\left\|a_j\right\|_1$$

Update $\theta_i^{(t+1)}$ for $i = 1,2,..,d$ using LASSO:

$$\theta_i^{(t+1)} = \arg\min_{\theta_i} \frac{1}{2}\left\|\left(E^i \bar{a}_i^T\right)^{(t)} - \Psi\theta_i\right\|_2^2 + \gamma\|\theta_i\|_1$$

End

Return the dictionary $\widehat{D}^* = D^{(T)}$ and the estimated patches: $\hat{X}^* = D^{(T)}A^{(T)}$

SIMULATION RESULTS

Performance analysis using the learning curve

To better understand the effectiveness and the convergence of the learning algorithm, we analyze a performance curve that we refer to by the *learning curve*. Learning curve shows the

quality of the image recovery as a function of the iteration number t or equivalently as a function of time[4]. We use the well-known Peak-Signal-to-Noise-Ratio (PSNR) to measure the recovery performance:

$$\text{PSNR} = 20 \times \log_{10} \frac{\max_{ji} x_{ji}}{\sqrt{\frac{1}{N}\Sigma_j^N \|x_j - \tilde{x}_j\|_2^2}} \tag{3.7}$$

where x_j is the underlying image block j and $\tilde{x}_j = \widetilde{D}\tilde{\alpha}_j$ is its recovery.

Normally, the learning curve must converge to a local or global optimum of the objective function which is, in our case, directly related to the PSNR. However, when only partial information is known about the underlying image, which is captured in a set of linear measurements, the algorithm does not always converge to an optimal PSNR and when it does, the convergence is slower for smaller number of measurements. An example of the learning curve is shown in Figure 2 for illustration. The red dashed line in this figure shows the PSNR for nonadaptive recovery using the fixed dictionary which is also the starting point for the learning algorithm.

[4] Assuming a fixed number of operations is performed in each iteration of the learning algorithm.

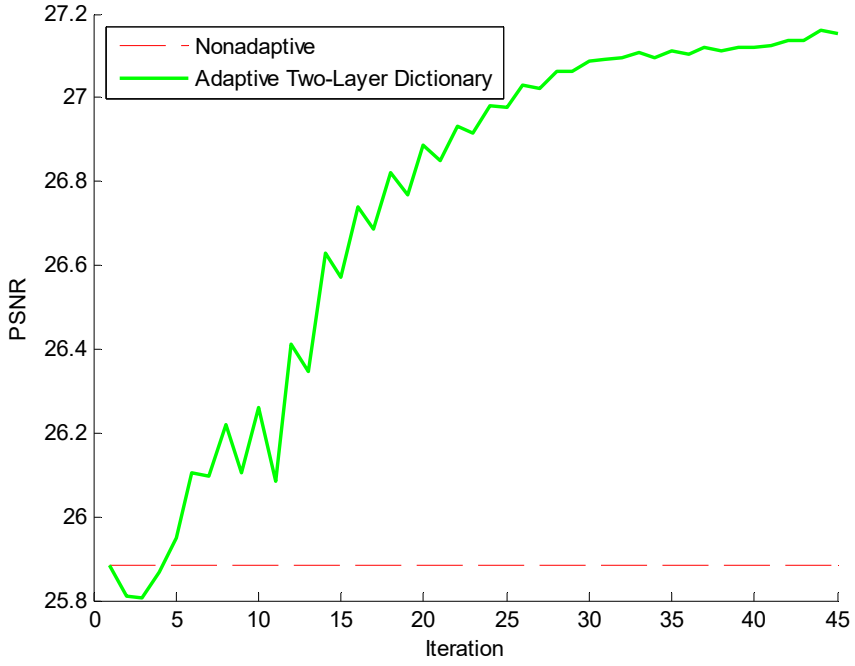


Figure 2. An example of a learning curve for the image 'Barbara' with sampling rate $\frac{m}{n} = \frac{1}{2}$. PSNR is in dBs.

Selection of the base dictionary

Instead of using a discrete cosine basis for the base dictionary as in [13], we use a real Fourier basis that consists of real basis vectors for 2D signals. We derived the real Fourier by exploiting the complex conjugate property of real signals in the Fourier domain. In Figure 3, the real Fourier basis is plotted against the discrete cosine basis. Intuitively, and as can be seen in the figure, the Fourier basis vectors have directional constructions that are more suitable for representing image structures like edges and texture. On the contrary, the cosine basis vectors are designed to uncorrelated the signal and capture as most energy in the first few coefficients that capture the low-frequency end of the signal spectrum. Unlike traditional recovery tasks where it was preferable to recover the lower end of the spectrum, in the problem of compressed sensing, it

is preferable to use both high-frequency and low-frequency constructions to sparsify and recover the signal. Also, through a series of empirical tests, we found that the Fourier basis is more suitable choice as the base frame for the construction of adaptive two-layer dictionaries.

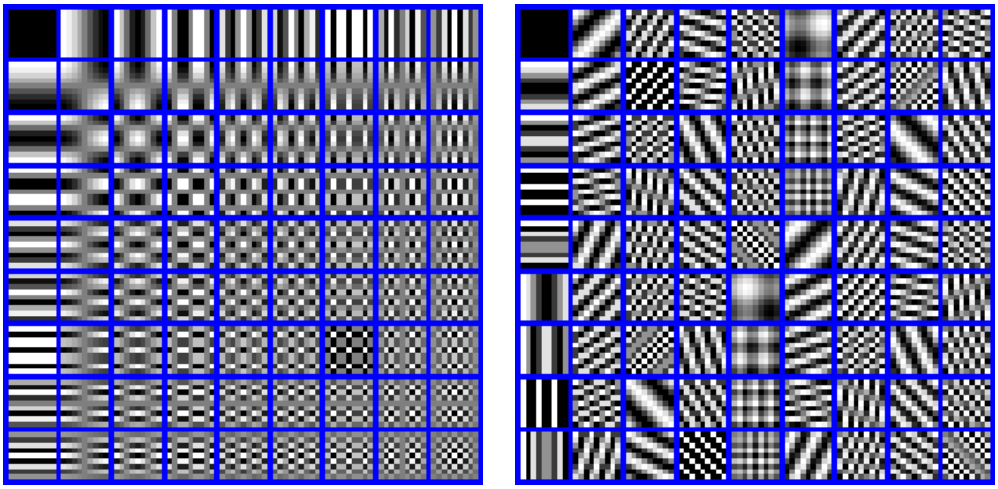


Figure 3. Left: discrete cosine basis. Right: real discrete Fourier basis.

Results

The set of image that were used for testing is displayed in Figure 4[5]. These images that are 512×512 are down-scaled for illustration.

[5] There is a fine texture in the images house and matches that may not be visible in the downscaled versions of these images.

Figure 4. The set of images used for performance evaluations (down-scaled). From left to right and top to bottom: Barbara, Lena, house, rocket, boat, fingerprint, matches and the man.

For demonstration purposes, we test with two sampling rates corresponding to very low rate at 20% (1 in every 5 samples) and an average rate at 50% (1 in every 2 samples). For 1-in-5 sampling we use $\lambda = 0.05$ and for 1-in-2 sampling we use $\lambda = 0.01$ while $\gamma = 0.05$ for both cases. Generally speaking, with more measurements, we can relax the ℓ_1 penalty which is why we use smaller λ for higher sampling rates. The sampling matrix Φ is sampled from a random independent identically distributed (i.i.d.) Gaussian distribution with normalized (unit norm) rows. In our experiments, we divide each image into 9×9 blocks for sampling and recovery.

For the average performance, we must recover each image several times, each time with a different (randomly generated) sampling matrix. However, before that, we study a few runs of the same experiment with different sampling matrices to check the variance of the learning process. Figure 5 shows the results for the 1-in-5 sampling for the image Lena under 10 different constructions of the sampling matrix. As can be seen, the learning curve can behave differently

with each sampling matrix. Consequently, there is not much point in studying the average of the learning curve.

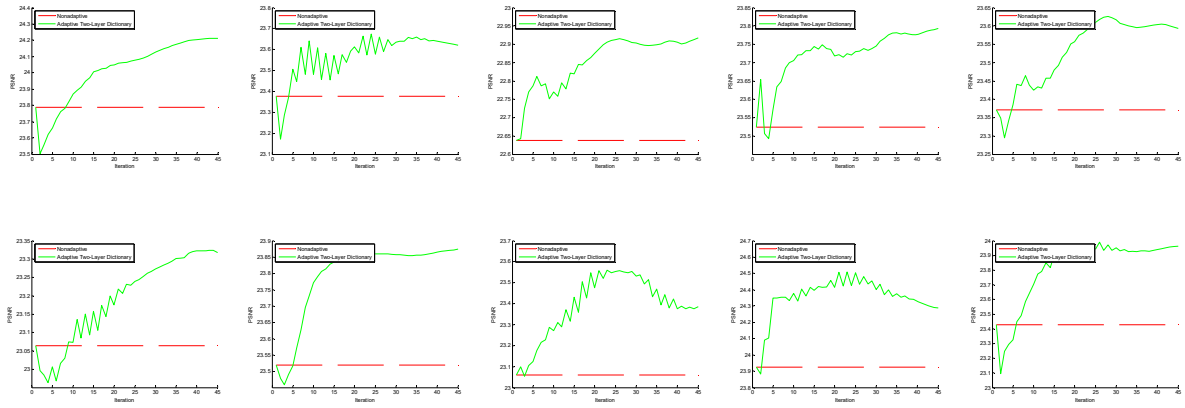


Figure 5. Multiple runs of the same experiment with different sampling matrices. (Experiment: Recovery from 1-in-5 sampling for the test image Lena.)

The average results for 1-in-5 sampling are presented in Table 2. The PSNR results are averaged over 10 trials, each with 50 iterations of the algorithm.

Table 2. Average PSNR results for the 1-in-5 sampling (adaptive two-layer dictionary). PSNRs are in dBs.

Image	Nonadaptive PSNR	Adaptive PSNR	Improvement
Barbara	21.62	21.90	0.28
Lena	23.56	23.86	0.30
house	22.70	23.04	0.34
rocket	25.37	25.58	0.21
boat	22.36	22.69	0.33
fingerprint	16.91	17.33	0.42
matches	20.52	20.86	0.33
the man	23.00	23.39	0.40

Table 3 contains the results for the case of 1-in-2 sampling. Similarly, each result is the average of 10 trials with 50 iterations each.

Table 3. Average PSNR results for the 1-in-2 sampling (adaptive two-layer dictionary). PSNRs are in dBs.

Image	Nonadaptive PSNR	Adaptive PSNR	Improvement
Barbara	26.59	27.13	0.54
Lena	29.16	29.88	0.72
house	28.02	28.25	0.23
rocket	32.88	34.59	1.71
boat	27.35	27.64	0.29
fingerprint	22.45	22.50	0.05
matches	25.76	25.84	0.08
the man	27.54	27.93	0.39

For instance, consider a single run for the image Lena. The learning curve is plotted in Figure 6 below.

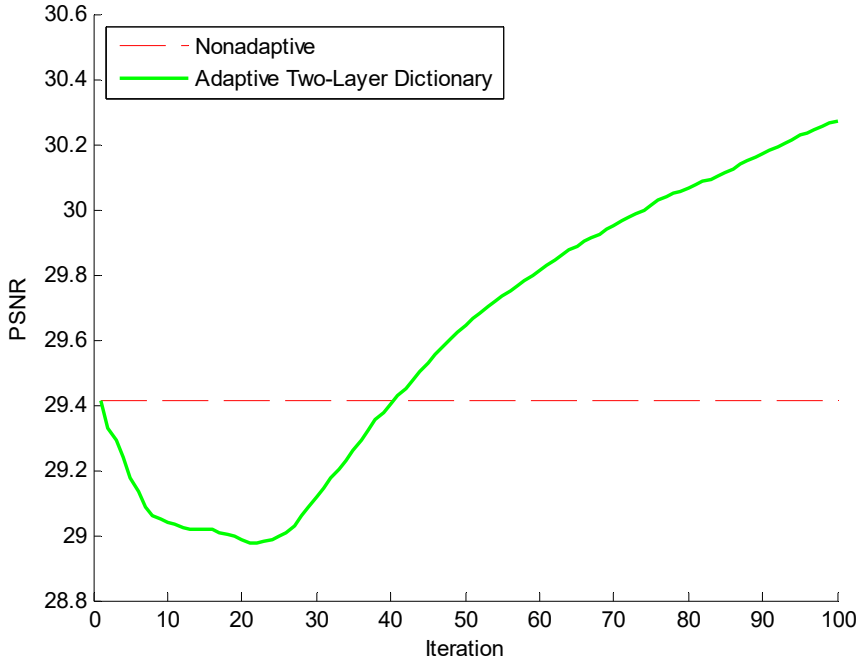


Figure 6. Learning curve for 1-in-2 sampling for Lena (two-layer dictionary).

The algorithm initially seems to be degrading the recovery throughout the optimization process[6] while after about 30 iterations it reaches a point that lies in the way of a potential optimum. Because of the inverse nature of the problem and the random construction of sampling matrices, it is hard to predict the algorithm behavior even after hundreds of iterations. Note that even in the traditional dictionary learning problem, which is based on complete knowledge of the signal, the optimization cost is a function with lots of local minima (even though each optimization layer has a convex objective function).

[6] The measurement-based cost function does not reflect the true recovery cost and only provides an estimate. This is the reason why decreasing the cost can sometimes result in loss of PSNR.

Concluding remarks

Concluding from the empirical results, the slow convergence rate of the algorithm is its main drawback even though the improvements after several iterations are consistent. On top of that, the complexity of solving for two-layer dictionaries can be overwhelming for image recovery tasks. The two-layer constrained dictionary model represents our first successful attempt to tackle the problem of dictionary learning when only partial information is accessible about the underlying data. Continuing the same line of work, we have been developing an efficient dictionary model (the autoregressive dictionary model) that is a more viable option for image recovery. Meanwhile, later in this work, we study random-block sampling that is superior to repeated-block sampling and results in a more reliable DL and signal recovery.

CHAPTER 4

AUTOREGRESSIVE REGULARIZATION FOR DL FROM REPEATED-BLOCK MEASUREMENTS

Similar to the previous chapter, our goal in this chapter is to improve the sparse image recovery from repeated-block measurements through real-time adaptation of the dictionary. This time, we utilize a non-parametric Bayesian approach to tackle the ill-posed nature of the problem. In this approach, each dictionary atom is modeled (a-priori) as a correlated process while different atoms are assumed to be independent. We employ a Maximum-a-Posteriori (MAP) estimation to update the dictionary using an empirical covariance matrix that is derived offline from a dataset of real-world images.

THE AR DICTIONARY MODEL

In the proposed autoregressive (AR) dictionary model, each atom is modeled as a 2D stationary process with a known covariance matrix C_d while different atoms are assumed to be independent. In this model, the same covariance matrix is used for all atoms. An instance of the

empirical covariance matrix is shown in Figure 7 where each block shows the pair-wise covariance of each pixel with every other pixel. As expected for natural signals, the covariance function monotonically decreases with distance.

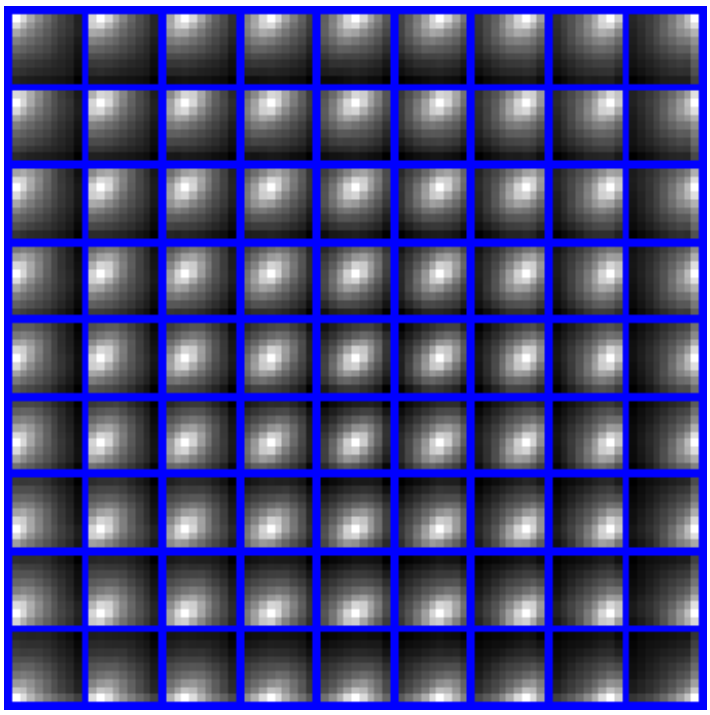


Figure 7. The empirical covariance matrix for AR dictionary.

Employing an empirical covariance matrix for the estimation process imposes a degree of smoothness onto the learned dictionary atoms.[7] In fact, Bayesian estimation with autoregressive priors can be viewed as non-parametric Gaussian regression [26] where the objective is to interpolate the sampled data using a smooth curve. In such interpolation applications, the smoothness prior prevents overfitting when samples are not sufficiently dense or the signal to noise ratio is small. However, for our task, the smoothness prior prevents the learned dictionary to get 'trapped' in the low-dimensional space of measurements. More details are provided below.

[7] The degree of smoothness is tuned to natural images. Care must be taken in that the dataset of images used for computing the empirical covariance matrix must not include the image under recovery.

LEARNING AR DICTIONARIES FROM REPEATED-BLOCK MEASUREMENTS

The original dictionary update objective from repeated-block measurements is:

$$\frac{1}{2}\sum_{j=1}^{N}\left\|y_j - \Phi D a_j\right\|_2^2$$

where $y_j = \Phi x_j$. As discussed before, decreasing this objective function (by updating the dictionary D) does not necessarily result in a decrease in the true objective function $\frac{1}{2}\sum_{j=1}^{N}\left\|x_j - Da_j\right\|_2^2$. Therefore, without additional information, overfitting is inevitable.

A typical solution to the overfitting issue is the assumption of smoothing priors that are usually derived from an empirical analysis of a training dataset. Specifically, the regularized objective function can be written as the sum of the original objective with an additional term:

$$\frac{1}{2}\sum_{j=1}^{N}\left\|y_j - \Phi D a_j\right\|_2^2 + \frac{1}{2}\sum_{i=1}^{d}\|\Gamma d_i\|_2^2 \qquad (4.1)$$

where the matrix Γ is usually called the Tikhonov regularization matrix [27]. If vectors d_i were modeled as zero-mean multivariate Gaussian variables, $\Gamma^T\Gamma = C_d^{-1}$ which is a direct result of writing the posterior probability in a Bayesian framework. To simplify the rest of equations, we can write the dictionary objective as:

$$\frac{1}{2}\mathrm{Tr}\big((Y - \Phi DA)^T(Y - \Phi DA)\big) + \frac{1}{2}\mathrm{Tr}(D^T C_d^{-1} D) \qquad (4.2)$$

The next step is to derive the gradient of the objective with respect to the dictionary D at the current iteration of the algorithm $D^{(t)}$:

$$\nabla_D f\big(D^{(t)}\big) = \Phi^T\big(\Phi D^{(t)} A^{(t)} - Y\big){A^{(t)}}^T + C_d^{-1} D^{(t)} \qquad (4.3)$$

Taking a step in the negative gradient direction results in the maximum rate of reduction of the cost function. However, to find the optimum step size, we need to solve a 1D line search problem. Since $D^{(t+1)} = D^{(t)} + \mu^{(t)} \nabla_D f\left(D^{(t)}\right)$, we must take the derivative of (24) with $D = D^{(t)} + \mu^{(t)} \nabla_D f\left(D^{(t)}\right)$ with respect to $\mu^{(t)}$ and set it equal to zero to find the optimum step size $\mu^{(t)}$. The final solution is:

$$\mu^{(t)^*} = \frac{Tr\left(B^T E\right) - Tr\left(\Delta^T C_{\tilde{d}}^{-1} D^{(t)}\right)}{Tr\left(B^T B\right) + Tr\left(\Delta^T C_{\tilde{d}}^{-1} \Delta\right)} \tag{4.4}$$

where we have defined $\Delta = \nabla_D f\left(D^{(t)}\right)$, $B = \Phi \Delta A^{(t)}$ and $E = \Phi D^{(t)} A^{(t)}$.

To summarize, we use a steepest descent approach to update the dictionary (by taking a single step in the descent direction) for the dictionary update stage. The step size is also optimized in our framework. By iterating between the sparse coding stage and the dictionary update stage, the dictionary is adapted to the underlying image, merely using the linear measurements, resulting in an image recovery that is superior to the non-adaptive (fixed dictionary) case.

ALGORITHM

Here, we present our algorithm for adapting AR dictionaries.

Algorithm 2. The algorithm for adapting AR dictionaries from measurements.

Input: Starting frame Ψ, Measurements $Y_{m\times N}$, the sampling matrix $\Phi_{\mathrm{m\times n}}$, the dictionary covariance C_d,

LASSO regularization parameter λ, number of iterations T

Outputs: learned dictionary $\widehat{D}^*$, Estimated patches $\hat{X}^*$

Initialization: $D^{(0)} = \Psi$

Do for t from 0 to $T-1$:

Compute $a_j^{(t)}$ for $j = 1,2,\dots,N$ using LASSO:

$$a_j^{(t)} = \arg\min_{a_j} \frac{1}{2}\left\|y_j - \Phi D^{(t)} a_j\right\|_2^2 + \lambda\left\|a_j\right\|_1$$

Update $D^{(t+1)}$ by taking a step in the steepest descent direction:

$$D^{(t+1)} = D^{(t)} + \mu^{(t)} \nabla_D f\left(D^{(t)}\right)$$

where $\nabla_D f\left(D^{(t)}\right)$ and $\mu^{(t)}$ are given in (25) and (26).

End

Return the dictionary $\widehat{D}^* = D^{(T)}$ and the estimated patches: $\hat{X}^* = D^{(T)} A^{(T)}$

SIMULATION RESULTS

The set of test images that we use in this work are shown in Figure 4 from the previous chapter. Similar to the previous chapter, we consider 1-in-5 sampling (as many as $\frac{n}{5}$ random measurements) and 1-in-2 sampling with repeated-block sampling matrices and the blocks are 9×9. Furthermore, we utilize the same definition of the learning curve for analyzing the performance and the efficiency of the algorithm. To reiterate, the empirical covariance matric needed for the dictionary model cannot be computed from a dataset that contains the test image.

A simple strategy would be to use cross-validation, i.e. removing the test image from the dataset and computing the covariance matrix for the remaining images for the adaptive dictionary.

Results

The average results for 1-in-5 sampling are presented in Table 4. The PSNR results are averaged over 10 trials, each with 50 iterations of the algorithm.

Table 4 Average PSNR results for the 1-in-5 sampling (adaptive AR dictionary). PSNRs are in dBs.

Image	Nonadaptive PSNR	Adaptive PSNR	Improvement (dB)
Barbara	21.49	22.97	1.48
Lena	23.35	27.82	4.46
house	22.62	26.24	3.62
rocket	25.04	29.21	4.18
boat	22.01	25.90	3.89
fingerprint	17.00	20.08	3.08
matches	20.42	23.30	2.89
the man	22.86	26.86	4.00

Table 5 contains the results for the case of 1-in-2 sampling. Similarly, each result is the average of 10 trials with 50 iterations each.

Table 5. Average PSNR results for the 1-in-2 sampling (adaptive AR dictionary). PSNRs are in dBs.

Image	Nonadaptive PSNR	Adaptive PSNR	Improvement (dB)
Barbara	26.77	26.23	-0.53
Lena	29.21	32.21	3.00
house	27.97	30.73	2.76
rocket	33.29	34.90	1.61
boat	27.25	29.59	2.33
fingerprint	22.53	23.08	0.55
matches	25.92	26.36	0.45
the man	27.48	30.41	2.93

Furthermore, the learning curve for a 1-in-2 sampling for the image Lena is shown in Figure 8 below.

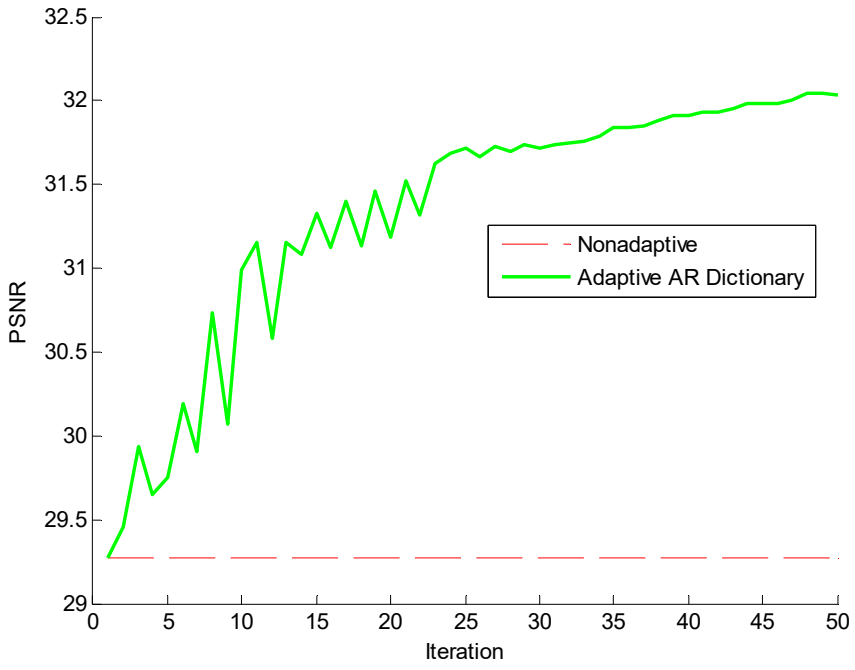


Figure 8. the learning curve for 1-in-2 sampling (Lena) for the adaptive AR dictionary.

The recovered images are shown in Figure 9 below, where clearly, the adaptive recovery presents a more visually pleasing image. Finally, the adapted dictionary is displayed in Figure 10. As expected, the learned AR dictionary appears smooth. The degree of smoothness, however, can be controlled through the dictionary covariance matrix. In fact, the covariance matrix used in this experiment is exponentially scaled so that the pixel correlation falls more drastically with distance, resulting in sharper atomic structures.

Figure 9. Recovered images (Lena) for 1-in-2 sampling using nonadaptive (left image) and adaptive AR dictionary (right image).

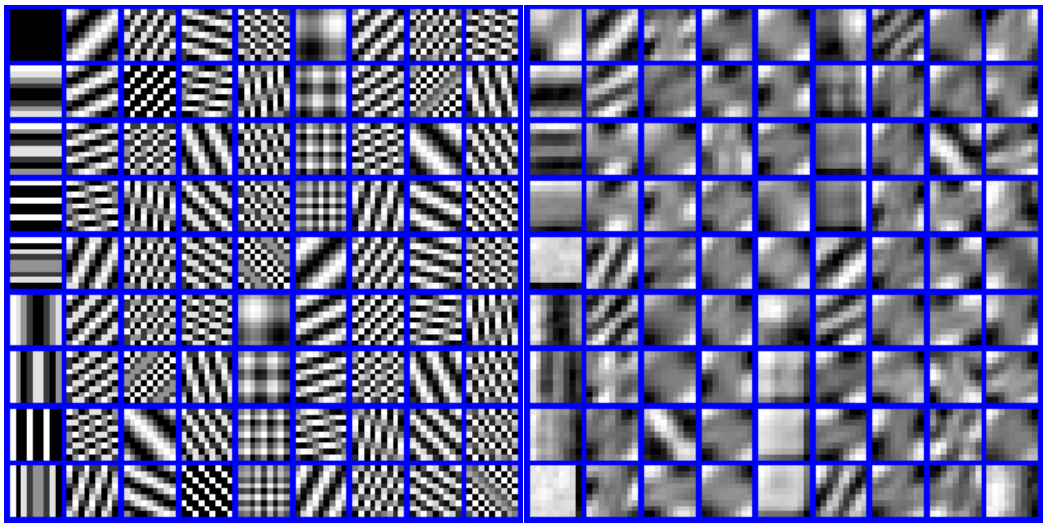


Figure 10. the starting dictionary (left image) versus the adapted dictionary (right image) for Lena based on 1-in-2 sampling after 50 iterations of the algorithm.

CHAPTER 5

UNCONSTRAINED DL FROM RANDOM-BLOCK MEASUREMENTS

The main contribution of this work lies in the study of the relationships between sensing matrices and the dictionary optimization process. Most of the existing imaging systems employ the same sampling matrix throughout the scene and do not consider the more general, arguably more efficient, random designs of the sensing operators. We argue that random-block measurements can be exploited directly within the dictionary learning algorithm without significantly increasing its computational complexity. Given that natural images contain self-similarities [28], the information contents of a set of similar patches, each projected onto a different low-dimensional space, is collectively more significant compared to the case where all patches are projected onto the same subspace. The information coming from different patches is merged at the decoder during the dictionary optimization as we explain in detail in the following sections.

RANDOM-BLOCK VS. REPEATED-BLOCK SAMPLING

We consider two sampling scenarios for images: repeated-block sampling and random-block sampling. In repeated-block sampling the (randomly generated) sampling matrix is fixed across the image. In mathematical terms, the measurements are expressed as $y_j = \Phi x_j$ where Φ is an $m \times n$ matrix (usually with $m \ll n$) and x_j is the block number j reshaped into a vector. In random-block sampling, an independently generated random matrix is used to sample each image block, i.e. $y_j = \Phi_j x_j$ where Φ_j's are independently generated for $j = 1,2, \dots, N$. Clearly, it makes no difference to use the repeated-block sampling or the random-block sampling if the image recovery is to be performed locally (block-by-block) as in normal CS. However, our goal is to learn the representation dictionary which involves combining the information from non-local measurements during the minimization of the total error sum of squares. Therefore, it is essential to study the impact of the sampling mode on the feasibility of the dictionary adaptation process. In particular, during the dictionary update stage using the random measurements, the objective function takes the form:

$$\sum_{j=1}^{N} \frac{1}{2} \left\| y_j - \Phi_j D a_j \right\|_2^2 = \sum_{j=1}^{N} \frac{1}{2} \left(x_j - D a_j \right)^T \Phi_j^T \Phi_j (x_j - D a_j) \qquad (5.1)$$

which we refer to by the Projected Residual Error Sum of Squares or PRESS.

We would like to know the relationship between the surrogate objective function and the True Residual Error Sum of Squares (TRESS):

$$\sum_{j=1}^{N} \frac{1}{2} \left\| x_j - D a_j \right\|_2^2 = \sum_{j=1}^{N} \frac{1}{2} \left(x_j - D a_j \right)^T (x_j - D a_j) \qquad (5.2)$$

under each sampling scenario.

Both objectives represent the sum of energies of the residual vectors. However, PRESS only captures part of the residual energy that is within the row space of the sampling matrices. In other words, PRESS represents the observable part of TRESS. Meanwhile, TRESS can have components in the null space of the sampling matrices that are hidden from the learner and may not be incorporated in the learned dictionary.

Our goal here is to study the ratio between the energy that is captured in the measurements, that is part of the residual energy that can be observed and minimized, to the true residual energy. This translates into the degree to which the measured objective function is concentrated around the true objective function value for different values of residual vectors. Unfortunately, however, residuals depend on the specific choice of the dictionary which makes the analysis case-dependent and less generic. Therefore, instead of working with residual vectors $\Phi_j(x_j - Da_j)$ and $x_j - Da_j$, we directly work with the measurement and signal vectors $\Phi_j x_j$ and x_j in the manner that is described below. Note that, generally speaking, residual vectors are (measurable) parts of signal vectors that are subject to sparse representation by updating the current dictionary. Therefore, using $\Phi_j x_j$ and x_j in place of $\Phi_j(x_j - Da_j)$ and $x_j - Da_j$ corresponds to assuming $a_j = 0$ (which is the sparsest representation of x_j with respect to any D with a large residual).

The total energy of an image with N blocks is $\sum_j^N \|x_j\|_2^2$ [8]. In an experiment, we divide the sample image 'Barbara' into non-overlapping 8 by 8 blocks and sample each block using 1) a fixed sampling matrix Φ and 2) a distinct sampling matrix Φ_j. Specifically, Φ and each of Φ_j's are generated according to a random i.i.d. Gaussian process and normalized so that each row has

[8] Without loss of generality, we assume the signal is zero-mean (we subtract the dc value from the image blocks) and we normalize each block to have unit norm.

unit norm. For each sampling scenario (repeated versus random-block), we run the experiment for 1000 times and compute the empirical distribution of the energy captured in the measurements $\sum_j^N \left\|\Phi_j x_j\right\|_2^2$. We are interested in the following ratio:

$$\frac{\left(\frac{n}{m}\right)\sum_j^N \|\Phi_j x_j\|_2^2}{\sum_j^N \|x_j\|_2^2} \tag{5.3}$$

which represents the concentration around the true signal energy. A concentration ratio of one implies that the measurements capture (preserve) the signal energy while ratios of larger or smaller magnitude imply deviations from the true signal energy and less accurate sensing. We have plotted the concentration ratio for the two sampling scenarios in Figure 11 and Figure 12.

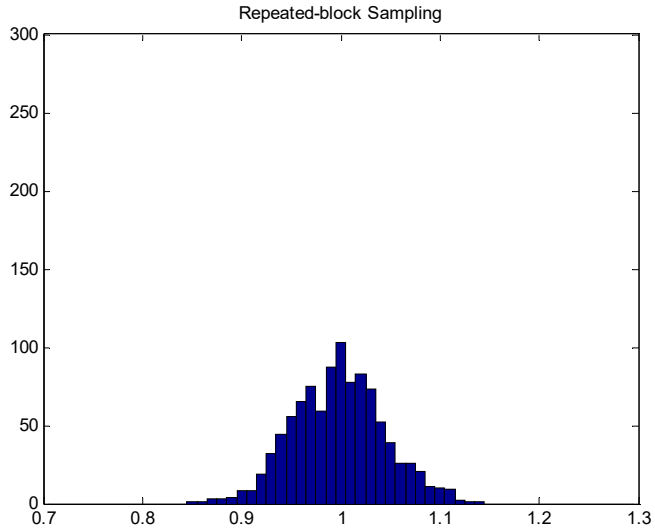


Figure 11. The empirical distribution for the concentration ratio for repeated-block sampling. The test signal comprises of the nonoverlapping 8 by 8 blocks of the 'Barbara' image. The empirical distribution is computed over 1000 trials.

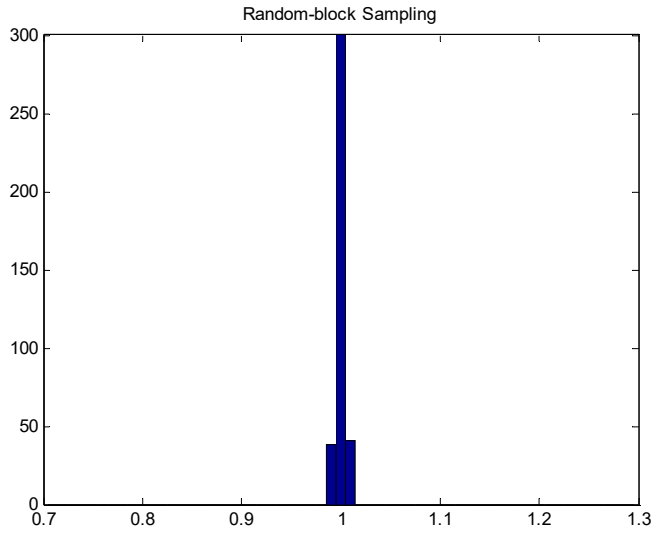


Figure 12. The empirical distribution for the concentration ratio for random-block sampling. Similar to Figure 11, the test signal comprises of the nonoverlapping 8 by 8 blocks of the 'Barbara' image and the distribution is computed over 1000 trials.

Looking at the above results, it can be clearly seen that the random-block measurements are more concentrated around the true signal energy and present a more stable approximation of the underlying signal. On the other hand, the repeated-block measurements are distributed almost like a Gaussian function with a relatively large variance around the true signal value.

Note that, during the dictionary learning, it is crucial to have a stable estimate of TRESS since the dictionary is optimized to minimize the total residual energy which makes it the driving force for the DL algorithm. According to the empirical study above, we can get a more stable estimate of this energy using random-block sampling. In fact, as we show in the following section, the dictionary can be directly fitted over the measurements without imposing artificial structural constraints over the dictionary.

DIRECT LEARNING FROM RANDOM-BLOCK MEASUREMENTS

The sparse coding stage is the same as before:

$$a_j^{(t+1)} = \arg\min_{a_j} \left\| y_j - D^{(t)} a_j \right\|_2^2 + \lambda \left\| a_j \right\|_1 \qquad (5.4)$$

However, the dictionary update stage is going to be different from the approaches described in previous chapters. The gradient of the empirical objective is computed at each iteration and an update step is taken along the opposite gradient direction. The step size is calculated by searching for the point on the negative gradient direction that yields the least function value[9]. The gradient of the surrogate objective (5.1) is:

$$\nabla_D \sum_{j=1}^{N} \frac{1}{2} \left(x_j - D a_j \right)^T \Phi_j^T \Phi_j (x_j - D a_j) = \sum_{j=1}^{N} \Phi_j^T \Phi_j \left(D a_j - x_j \right) a_j^T \qquad (5.5)$$

Therefore, the dictionary is updated as $D^{(t+1)} = D^{(t)} + \mu^{(t)} \nabla_D f(D^{(t)})$ where $\mu^{(t)}$ is optimized as in the steepest descent algorithm [29].

As before, we present an empirical study to showcase the performance of the dictionary learning using random linear and incomplete measurements. The formal algorithm description and more simulation results are presented in the following sections. For our experiments, we work with the Barbara's image due to its texture patterns that makes it difficult to represent compactly using ordinary image dictionaries.

In this experiment, non-overlapping 8×8 patches from the Barbara's image are sampled at half rate ($m = \left\lfloor \frac{n}{2} \right\rfloor = 32$). Specifically, we are interested in studying the learning curve of the iterative DL algorithm which plots the true PSNR (which is unknown to the algorithm) over the

[9] A simple line search yields the optimum step size.

course of iterations. For the random-block sampling case, sampling matrices are generated randomly for each patch according to independent and identically distributed Gaussian distribution. Furthermore, we orthogonalize and normalize rows of each generated sampling matrix so that measurements are weighted equally in the projected error sum of squares cost function. The employed dictionary for this experiment is the redundant offline-learned dictionary presented in [5] which serves as the benchmark among redundant dictionaries for natural images. Although this experiment is carried out for a specific sampling ratio, complete tests presented in the simulations section show the results are consistent within all ranges of sampling ratios. Figure 13 shows the learning curve for both cases of repeated-block and random-block sampling matrices as well as the recovery result using the fixed dictionary.

Some crucial observations can be made from the learning curve in Figure 13. Although a slight improvement is achieved after the first few iterations of the learning algorithm with the repeated-block sampling matrix, the PSNR is decreased subsequently due to overfitting. More specifically, using a fixed sampling matrix creates a 'global' null space (a null-space that exists in all blocks) that prevents the dictionary to improve in the dimensions masked by the sampling matrix. As can be seen in Figure 13, the overfitting associated with the repeated-block sampling significantly damages its performance to the point that its recovery results drop below the non-adaptive recovery.

Clearly, the adaptive recovery results using random-block measurements shown in Figure 13 significantly outperform the non-adaptive recovery just after few iterations. As mentioned before, this is due to the fact random-block sampling is more reliable in preserving the residual sum of squares. It must be mentioned that, however, adaptation using random-block measurements is not completely immune from overfitting. In images with significant

irregularities or randomness where the improvements due to dictionary adaptation is less significant, we have observed degradations in PSNR after a large number of iterations. This shows that the concentration of objective function is signal dependent and a rather complicated phenomenon. To combat such issues, the algorithm must be terminated before overfitting starts which itself very difficult to detect.

Another type of analysis would be to visually inspect the dictionaries before and after adaptation. These dictionaries are shown in Figure 14. The top dictionary in Figure 14 shows the offline-learned dictionary that was computed in [5] and has $d = 256$ atoms (columns) of size $n = 64$. The adapted dictionary for the Barbara's image (using $m = 32$) is shown at the bottom of the Figure 14. Although a visual inspection of the dictionaries does not explain the complexities associated with sparse coding, the results are coherent with the expectation that input image's texture patterns must be captured in the adapted dictionary for an accurate sparse recovery. These texture patterns could not be sparsely represented using the initial dictionary which is why the dictionary must adapt to capture such structures. The suboptimality of the initial dictionary in sparse representation of the textures result in visible artifacts after CS as shown in Figure 15. The adaptive CS recovery result is shown in Figure 16 for comparison. Note the improvements in the cloth texture in the recovered image compared to the initial CS recovery using the universal offline-learned dictionary.

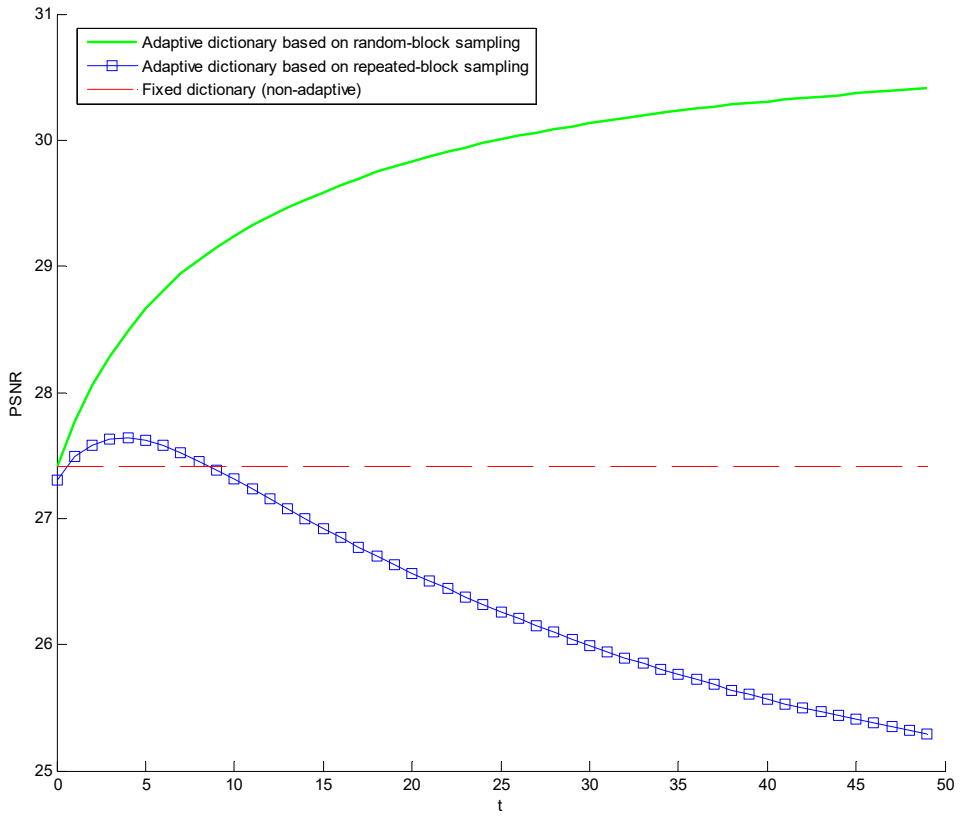


Figure 13. Learning curve for unconstrained learning from random-block sampling versus repeated-block sampling, compared to the nonadaptive recovery.

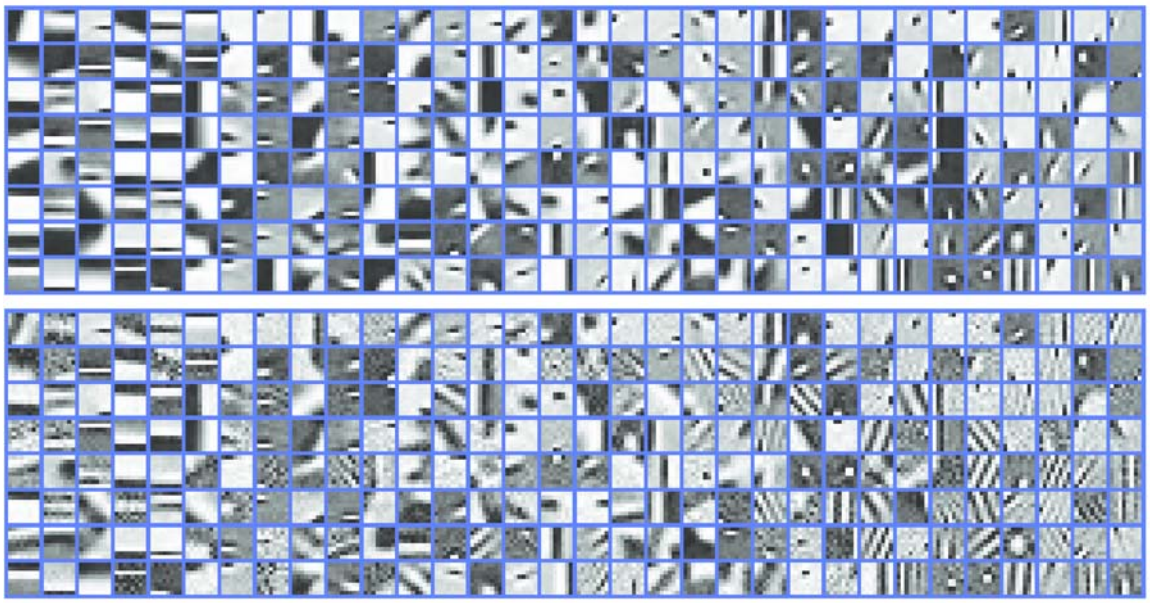


Figure 14. Images of the dictionaries. Top: offline-learned dictionary (the starting point of the learning algorithm). Bottom: the learned dictionary for the image Barbara.

Figure 15. Recovery result using non-adaptive CS with offline-learned dictionary (Barbara).

Figure 16. Recovery result using CS with adaptive dictionary based on random-block sampling (Barbara).

ALGORITHM

We have presented our algorithm below.

Algorithm 3. The algorithm for learning dictionaries from random-block measurements.

Input: Starting frame Ψ, Measurements $Y_{m\times N}$, the sampling matrices Φ_j for $j = 1,2,\dots,N$,

LASSO regularization parameter λ, number of iterations T

Outputs: learned dictionary $\widehat{D}^* = D^{(T)}$, Estimated patches $\hat{X}^*$

Initialization: $D^{(0)} = \Psi$

Do for t from 1 to T:

Compute $a_j^{(t)}$ for $j = 1,2,\dots,N$ using LASSO:

$$a_j^{(t)} = \arg\min_{a_j} \frac{1}{2}\left\|y_j - \Phi_j D^{(t)} a_j\right\|_2^2 + \lambda\left\|a_j\right\|_1$$

Update $D^{(t+1)}$ by taking a step in the steepest descent direction:

$$D^{(t+1)} = D^{(t)} + \mu^{(t)} \nabla_D f\left(D^{(t)}\right)$$

where $\nabla_D f\left(D^{(t)}\right)$ is given in (30) and $\mu^{(t)}$ is computed using:

$\mu^{(t)} = \frac{\mathrm{Tr}\left(\Delta^{(t)^T}\Delta^{(t)}\right)}{\mathrm{Tr}\left(\Delta^{(t)^T} Q^{(t)}\right)}$ with $\Delta^{(t)} = \nabla_D f\left(D^{(t)}\right)$ and $Q^{(t)} = \sum_j^N \Phi_j^T \Phi_j \Delta^{(t)} a_j^{(t)} a_j^{(t)^T}$

End

Return the dictionary $\widehat{D}^* = D^{(T)}$ and the estimated patches: $\hat{X}^* = D^{(T)} A^{(T)}$

CONNECTIONS WITH GROUP SPARSITY

Unlike the learning approaches that force the dictionary atoms to have a unit ℓ_2-norm or lie inside the unit sphere [4, 30, 31], we do not impose such constraints on the atom norms. The steepest descent algorithm for dictionary update naturally reaches a point (close to the local minimum) where the representation error becomes insignificant and the dictionary stops changing significantly. At this point, each atom has a different norm which results in some atoms

having more priority than others. Intuitively, atom norms increase at a rate proportional to their share in reducing the total error sum of squares. If $D^{(t)}$ is factorized as $U^{(t)}V^{(t)}$, where $V^{(t)}$ is the diagonal matrix of atom norms and $U^{(t)}$ is the normalized dictionary, the Lasso problem in (5.4) can be written as:

$$b_j^{(t)} = \arg\min_{b_j} \frac{1}{2}\left\|y_j - \Phi U^{(t)} b_j\right\|_2^2 + \lambda\left\|V^{(t)^{-1}} b_j\right\|_1 \qquad (5.6)$$

where we would use $a_j^{(t)} = V^{(t)^{-1}} b_j^{(t)}$ to compute the original coefficients. The objective function based on b_j (and $U^{(t)}$ as the dictionary) corresponds to a weighted Lasso regression where the weights are inversely proportional to the corresponding atom norms. Coefficients with smaller weights have less uncertainty and are encouraged to take non-zero values, meaning that the corresponding atoms have more priority in the representation. Note that the same structure of uncertainty is used for all patches in the image. This kind of unbalanced uncertainty that 'harmonizes' the sparse coefficients resembles the *multiple measurement vector* framework in [32] where different blocks are forced to take the same sparsity pattern. The difference is that in our framework this harmony emerges naturally and without enforcing group-sparsity constraints.

SIMULATION RESULTS

To test the algorithm discussed in the previous section, we select the dictionary that produces the best recovery to start from. Through trial and error we found that the redundant offline-learned dictionary of [5], shown in Figure 14 (top row), provides the best recovery results for 8×8 patches. Down-scaled versions of the test images are shown in Figure 4.

Selecting the LASSO parameter λ

The first challenge in testing and comparing different dictionaries is the dependence of the optimal λ in (5.4) on the choice of the dictionary and the test images. When a dictionary perfectly fits an image, it is best to use $\lambda = 0$. However, since this is almost never the case, λ is often selected to be a small positive number close to zero accounting for a small amount of tolerable error. The optimal λ for a specific dictionary also depends on the choice of the test image. Due to its data-dependent nature, in most works, a typical value of λ is selected empirically to work well using cross-validation [31].

Another challenge in comparing the recovery using non-normalized dictionaries with the recovery based on normalized dictionaries is again the choice of λ. In the weighted LASSO in (5.6), the multiplier for each coefficient $b_j^{(t)}$ is

$$\lambda_j^{(t)} = \lambda \left(\left\| d_j^{(t)} \right\|_2 \right)^{-1}$$

For the corresponding unweighted Lasso, we can compute $\bar{\lambda}^{(t)}$ based on either of the three different strategies listed below:

- $\bar{\lambda}^{(t)} = \min_j \lambda \left(\left\| d_j^{(t)} \right\|_2 \right)^{-1}$: This value of $\bar{\lambda}^{(t)}$ which is less than the other two values below, implies maximum uncertainty for the coefficients. We utilized this value as a lower bound for $\bar{\lambda}^{(t)}$.
- $\bar{\lambda}^{(t)} = \lambda \left(\frac{1}{k} \sum_j \left(\left\| d_j^{(t)} \right\|_2 \right)^{-1} \right)$: In this case, $\bar{\lambda}^{(t)}$ is the mean of weights for all coefficients. In other words, the sum of uncertainties for each case of weighted and unweighted Lasso will be the same if this quantity is used.

- $\bar{\lambda}^{(t)} = \lambda \sqrt{\frac{1}{k}\sum_j \left(\left\| d_j^{(t)} \right\|_2^2\right)^{-1}}$: This value represents the root mean square of weights. Therefore, in this case, the total energy of weights is the same for the two cases of weighted and unweighted Lasso problems.

We performed Lasso based on all three values and found the

$$\bar{\lambda}^{(t)} = \lambda \left(\frac{1}{k} \sum\nolimits_j \left(\left\| d_j^{(t)} \right\|_2 \right)^{-1} \right)$$

to produce the best PSNR values for most images and sampling rates. The third method of selecting $\bar{\lambda}^{(t)}$ produced very similar results. As a result, we report the results only for this selection of $\bar{\lambda}^{(t)}$. We selected $\lambda = .01$ which represents the upper bound on weight of coefficients given that atom norms are larger than unity.

Results

We have summarized the PSNR results in Table 6. Although the application of algorithm generally improves the recovery results, forcing the total sum of squares to very small values can lead to overfitting depending on the image and the initial dictionary. Upper bounds on how much the results can be improved without introducing overfitting is subject of our ongoing work. For this simulation, we set $T = 10$ (the maximum number of iterations of the algorithm).

Table 6 Average PSNR results for adaptive recovery for different sampling rates (random-block sampling). PSNRs are in dBs.

	Nonadaptive recovery					Adaptive recovery				
Sampling rate / Image	10%	15%	20%	30%	50%	10%	15%	20%	30%	50%
Barbara	21.81	22.61	23.31	24.76	27.43	21.93	22.84	23.65	25.49	29.20
Lena	25.34	26.87	28.40	31.48	35.17	26.11	27.58	29.01	31.74	35.30
house	24.29	25.66	27.08	30.22	34.87	24.60	26.69	28.99	32.46	36.22
rocket	26.86	29.19	31.25	35.93	41.28	28.56	30.97	32.60	36.76	41.73
boat	23.32	24.62	25.90	28.52	32.22	23.46	24.93	26.18	28.74	32.49
fingerprint	18.53	20.39	22.41	25.71	30.36	20.41	23.20	25.58	29.21	34.17
matches	21.84	23.58	25.15	27.74	31.14	22.51	24.53	25.84	28.03	31.35
the man	24.13	25.48	26.63	29.01	32.45	24.31	25.76	26.91	29.09	32.48

CHAPTER 6

MATHEMATICAL ANALYSIS

In this chapter we intend to establish a systematic mathematical understanding of the previous empirical results. In a nutshell, theories of Concentration of Measure (CM) [3] assert that increasing the number of independent random variables of a smooth function reduces its variance and makes it more concentrated around its expected value. This, in turn, implies that using independent measurement matrices for different signal blocks would result in a more accurate approximation of the objective function and arguably a more accurate estimation of the dictionary. Throughout the following sections, we formalize these notions under different measurement scenarios.

The outline of this chapter is as follows. In Section 6.1, we compute and compare tail bounds of the dictionary update objective under repeated-block and random-block sampling schemes which becomes crucial for further assessment of DL under random projections of data. Section 6.2 presents a generic framework for bounding the error in parametric estimation when tail bounds of the empirical risk are available. Finally, the estimation accuracy of dictionary adaptation is studied in Section 6.3.

STOCHASTIC CM ANALYSIS FOR DICTIONARY UPDATE

This section draws important connections between a typical engineering problem, dictionary update in this case, and analytical tools from the area of the Concentration of Measure (CM) [3] theory and serves as a gateway to the rest of this chapter.

Motivations and overview

In the previous chapter we argued that having a reliable approximation of the true residual sum of squares from random projections of data enhances the accuracy of dictionary learning and eventually the quality of the reconstructed image. However, these conclusions were intuitive and empirical as opposed to being analytical and generalizable to other learning problems. Several decades of CM research, specifically in area of probability theory, has resulted in useful analytical tools and countless theorems for bounding stochastic tails of functions of random variables in the form of scalars, vectors or matrices. Over the time, however, these theorems have become very difficult to track and less intuitive for engineers. Our goal in this section is to utilize some of the fundamental CM inequalities, such as the scalar Chernoff inequality [50], to derive tail bounds of the residual sum of squares under both settings of having repeated-block and random-block measurements. Later on, we report some of the relevant (but more general) tail bounds that have been developed by researchers in the area of CS.

Recall the Projected Residual Error Sum of Squares (PRESS):

$$\sum_{j=1}^{N} \left\| y_j - \Phi_j D a_j \right\|_2^2 = \sum_{j=1}^{N} \left\| \Phi_j \left(x_j - D a_j \right) \right\|_2^2$$

and the True Residual Error Sum of Squares (TRESS):

$$\sum_{j=1}^{N} \left\| x_j - D a_j \right\|_2^2$$

It must be clear that $E\{\text{PRESS}\} = \text{TRESS}$. In other words, PRESS is an unbiased estimator of TRESS. However, there is no guarantee that $|\text{PRESS} - \text{TRESS}|$ is small for different values of the residual vectors. In fact, we are interested in the tail bound of the distribution of the random variable PRESS or formally:

$$\Pr\{|\text{PRESS} - \text{TRESS}| \geq \epsilon \text{TRESS}\}$$

or (with some abuse of notation)

$$\Pr\{(1 - \epsilon)\text{TRESS} \geq \text{PRESS} \geq (1 + \epsilon)\text{TRESS}\}$$

As discussed earlier, a drawback of working with the residual vectors is that they depend on the specific choice of the dictionary and the sparse coefficients. Thus, we proposed before that instead of comparing the projected and the true residual vectors $\Phi_j \left(x_j - D a_j \right)$ and $x_j - D a_j$, we directly compare the measurement and the signal vectors $\Phi_j x_j$ and x_j. This makes the following analysis more general and independent of the particular choice of the dictionary and the sparse coefficients. Specifically, residual vectors are expected to have similar stochastic characteristics as signal vectors when they are not dominated by noise.

In what follows, we review some of the basic CM tools that will be used in the rest of this chapter. We start by reviewing the Chernoff bounding method which results from the basic Markov's inequality $\Pr\{X \geq t\} \leq t^{-1}E\{X\}$ for nonnegative X [50].

Theorem 6.1 (Chernoff bounding method). [50] *For any random variable X and $t > 0$, the following inequalities hold:*

$$Pr\{X \geq t\} \leq \min_{s>0} \frac{E\{e^{sX}\}}{e^{st}} \tag{6.1}$$

and

$$Pr\{X \leq t\} \leq \min_{s>0} \frac{E\{e^{-sX}\}}{e^{-st}} \tag{6.2}$$

conditioned on the fact that the right hand side exists.

In the following, we overview some relevant examples where the Chernoff inequality becomes useful.

Tail bound for random vector norm (uncorrelated Gaussian)

Suppose $X \in \mathbb{R}^n$ is distributed according to a Gaussian distribution with zero-mean and covariance matrix $\sigma^2 I$, i.e. $X \sim N(0, \sigma^2 I)$. Then, $E\{\|X\|_2^2\} = n\sigma^2$ and the Chernoff bound can be computed as:

$$\Pr\left\{\|X\|_2^2 \geq (1+\epsilon)E\{\|X\|_2^2\}\right\} \leq \min_{s>0} \frac{E\left\{e^{s\|X\|_2^2}\right\}}{e^{s(1+\epsilon)n\sigma^2}} = \min_{s>0} \frac{(1-2s\sigma^2)^{-\frac{n}{2}}}{e^{-s(1+\epsilon)n\sigma^2}}$$

where we used $E\left\{e^{s\|X\|_2^2}\right\} = (1-2s\sigma^2)^{-\frac{n}{2}}$ for $X \sim N(0, \sigma^2 I)$ [50].

The tightest bound can be computed by taking the derivative of the right hand side with respect to s and setting it equal to zero which results in $s^* = \frac{\epsilon}{2(1+\epsilon)\sigma^2}$. After some basic calculus, it can be shown that for $0 < \epsilon < \frac{1}{2}$ [50]:

$$\Pr\left\{\|X\|_2^2 \geq (1+\epsilon)E\{\|X\|_2^2\}\right\} \leq e^{-\frac{n\epsilon^2}{6}}$$

and

$$\Pr\left\{\|X\|_2^2 \leq (1-\epsilon)E\{\|X\|_2^2\}\right\} \leq e^{-\frac{n\epsilon^2}{4}}$$

Therefore, using the union bound principle,

$$\Pr\left\{(1-\epsilon)E\{\|X\|_2^2\} \geq \|X\|_2^2 \geq (1+\epsilon)E\{\|X\|_2^2\}\right\} \leq 2e^{-\frac{n\epsilon^2}{6}} \qquad (6.3)$$

A simple consequence of this inequality is that a random Gaussian (uncorrelated) ℓ_2 vector norm approaches its expected value as the ambient dimension n approaches infinity. However, the above inequality, and generally any CM inequality, offers more than just an asymptotic interpretation. The non-asymptotic implication of the Chernoff inequality is that, at a fixed probability, $\epsilon = O(n^{-\frac{1}{2}})$ which implies the *rate* at which ϵ decays as the dimension of X is increased.

Tail bound for linear measurement vector norm

Consider the linear measurement vector $Y = \Phi x$ where $x \in \mathbb{R}^n$ is a fixed vector and entries of the random matrix $\Phi \in \mathbb{R}^{m \times n}$ are distributed independently according to a Gaussian distribution with zero mean and variance m^{-1}. For the expected value, we have

$$E\{\|Y\|_2^2\} = E\{x^T \Phi^T \Phi x\} = x^T E\{\Phi^T \Phi\} x = \|x\|_2^2$$

Using the fact that $Y \in \mathbb{R}^m$ is an uncorrelated Gaussian vector with a zero mean and entry-wise variance $\sigma^2 = m^{-1}\|x\|_2^2$, we can use a similar analysis as before and derive the following tail bound for the distribution of the random variable $\|Y\|_2^2 = \|\Phi x\|_2^2$:

$$\Pr\{(1-\epsilon)\|x\|_2^2 \geq \|\Phi x\|_2^2 \geq (1+\epsilon)\|x\|_2^2\} \leq 2e^{-\frac{m\epsilon^2}{6}} \qquad (6.4)$$

for $0 < \epsilon < \frac{1}{2}$. This is an intermediate result of the well-known Johnson-Lindenstrauss lemma (after which a union bound over $\binom{N}{2}$ vectors is carried to derive the J-L bound for N points) [13].

Computing tail bounds of sum of squares for random-block measurements

As the first step, we specify the random-block measurement model. It must be noted that these specifications are mainly for simplicity of the analysis and can be usually extended to include more general scenarios without much trouble.

Specification of random-block measurements

In random-block measurements, each block sampling matrix $\Phi_j \in \mathbb{R}^{m\times n}$ is generated independently. Entries of the Φ_j are distributed according to a joint but uncorrelated Gaussian distribution with zero-mean and variance m^{-1} (for each entry). Clearly, under such distribution:

$$E\{\Phi_i^T \Phi_j\} = \begin{cases} I & \text{when } i = j \\ 0 & \text{when } i \neq j \end{cases} \tag{6.5}$$

Chernoff bound for the sum of squares (random-block measurement)

Consider the sum of squares $\sum_{j=1}^{N} \|Y_j\|_2^2 = \sum_{j=1}^{N} \|\Phi_j x_j\|_2^2$ where, as before, each $\Phi_j \in \mathbb{R}^{m\times n}$ is independently generated and consists of uncorrelated Gaussian entries with zero mean and variance m^{-1}. Similarly, each Y_j is an uncorrelated Gaussian vector with entry-wise variance $\sigma_j^2 = m^{-1}\|x_j\|_2^2$. For the expected value, we have

$$E\left\{\sum_{j=1}^{N} \|\Phi_j x_j\|_2^2\right\} = \sum_{j=1}^{N} \|x_j\|_2^2 = m\sum_{j=1}^{N} \sigma_j^2$$

Therefore, the Chernoff bound would be

$$\Pr\left\{\sum_{j=1}^{N} \|\Phi_j x_j\|_2^2 \geq (1+\epsilon)\sum_{j=1}^{N} \|x_j\|_2^2\right\} \leq \min_{s>0} \frac{E\left\{e^{s\sum_{j=1}^{N} \|\Phi_j x_j\|_2^2}\right\}}{e^{s(1+\epsilon)m\sum_{j=1}^{N} \sigma_j^2}} \tag{6.6}$$

Given that each summand is independent of the other summands, we can expand the expect value term in the numerator of the right hand side:

$$E\left\{e^{s\sum_{j=1}^{N}\left\|\Phi_j x_j\right\|_2^2}\right\} = E\left\{\prod_{j=1}^{N} e^{s\left\|\Phi_j x_j\right\|_2^2}\right\} = \prod_{j=1}^{N} E\left\{e^{s\left\|\Phi_j x_j\right\|_2^2}\right\}$$

Therefore, the right hand side can be written as:

$$\min_{s>0} \frac{\prod_{j=1}^{N} E\left\{e^{s\left\|\Phi_j x_j\right\|_2^2}\right\}}{\prod_{j=1}^{N} e^{s(1+\epsilon)m\sigma_j^2}} = \min_{s>0} \prod_{j=1}^{N}\left(1-2s\sigma_j^2\right)^{-\frac{m}{2}} e^{-s(1+\epsilon)m\sigma_j^2} \tag{6.7}$$

Taking the derivative with respect to s and setting it equal to zero results in the following equation:

$$\sum_{j=1}^{N} \frac{\sigma_j^2}{1-2s^*\sigma_j^2} = (1+\epsilon)\sum_{j=1}^{N}\sigma_j^2 \tag{6.8}$$

For the special case of having $\sigma = \sigma_1 = \sigma_2 = \cdots = \sigma_N$, we arrive at $s^* = \frac{\epsilon}{2(1+\epsilon)\sigma^2}$ (similar to before) and the Chernoff bound becomes:

$$\Pr\left\{(1-\epsilon)\sum_{j=1}^{N}\left\|x_j\right\|_2^2 \geq \sum_{j=1}^{N}\left\|\Phi_j x_j\right\|_2^2 \geq (1+\epsilon)\sum_{j=1}^{N}\left\|x_j\right\|_2^2\right\} \leq 2e^{-\frac{mN\epsilon^2}{6}} \tag{6.9}$$

Note, here, $\epsilon = O\left((mN)^{-\frac{1}{2}}\right)$ while in the single measurement case $\epsilon = O\left(m^{-\frac{1}{2}}\right)$. In simple words, having access to N independent measurements *compensates* for the low-dimensionality of the measurement space.

It turns out that having equal signal energies, i.e. $\sigma_1 = \sigma_2 = \cdots = \sigma_N$, corresponds to the best-case scenario and generally the Chernoff bound may not be as small [13]. This best-case scenario corresponds to the case of having a constant $m^{-1}\left\|x_j\right\|_2^2$, i.e. blocks having equal energies. If we were allowed to allocate different numbers of measurements to different blocks, then the optimal way to distribute m_j would be to select m_j such that $m_j^{-1}\left\|x_j\right\|_2^2$ would stay

(approximately) constant. More precisely, the number of measurements per block must scale linearly with the block-wise energies.

Unfortunately, it is not possible to compute a closed-form solution for s^* for the general case. However, later in this section, we report some of the other works that have computed upper bounds for this Chernoff bound.

Computing tail bounds of sum of squares for repeated-block measurements

This subsection has roughly the same structure as the previous subsection only for the case of having repeated-block measurements. At the end, we will show that repeated-block measurements have heavier sum of squares tail bounds compared to random-block measurements when the data is strongly correlated.

Specification of repeated-block measurements

In the repeated-block scheme, every block is sampled using the same sampling matrix Φ which is distributed similarly to the random-block scheme for a single Φ_j. That is entries of Φ are independent and each of them is a Gaussian random variable with zero mean and variance m^{-1}.

Chernoff bound for the sum of squares (repeated-block measurement)

Consider the sum of squares function:

$$\sum_{j=1}^{N}\left\|Y_j\right\|_2^2=\sum_{j=1}^{N}\left\|\Phi x_j\right\|_2^2$$

As before, for the expected value, we have

$$E\left\{\sum_{j=1}^{N}\left\|\Phi x_j\right\|_2^2\right\}=\sum_{j=1}^{N}\left\|x_j\right\|_2^2=m\sum_{j=1}^{N}\sigma_j^2$$

Therefore, the Chernoff bound would be

$$\Pr\left\{\sum_{j=1}^{N}\left\|\Phi x_j\right\|_2^2\geq(1+\epsilon)\sum_{j=1}^{N}\left\|x_j\right\|_2^2\right\}\leq\min_{s>0}\frac{E\left\{e^{s\sum_{j=1}^{N}\left\|\Phi x_j\right\|_2^2}\right\}}{e^{s(1+\epsilon)m\sum_{j=1}^{N}\sigma_j^2}} \qquad (6.10)$$

However, this time, we cannot expand the numerator of the right hand side since the vectors Φx_j are dependent –in fact, linearly correlated Gaussian vectors. To tackle this problem, we will need to write the sum of squares as a sum of uncorrelated parts.

$$\sum_{j=1}^{N}\left\|\Phi x_j\right\|_2^2=\|\Phi\mathrm{X}\|_F^2=Tr(X^T\Phi^T\Phi X)=Tr(\Phi XX^T\Phi^T) \qquad (6.11)$$

Consider the spectral decomposition of the correlation matrix $XX^T=U\Lambda U^T$ (or the sample covariance matrix under the assumption of zero-mean signal vectors) where $U=[u_1,u_2,\dots,u_n]$ and $\Lambda=diag(\lambda_1,\lambda_2,\dots\lambda_n)$ and that $\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_n$. Note that the matrix product ΦU consists of independent Gaussian column vectors because $E\{(\Phi U)^T(\Phi U)\}=U^TE\{\Phi^T\Phi\}U=U^TU=I$. Therefore, the sum

$$Tr(\Phi XX^T\Phi^T)=Tr\big((\Phi\mathrm{U})\Lambda(\Phi U)^T\big)=Tr(\textstyle\sum_{i=1}^{n}\lambda_{\mathrm{i}}(\Phi u_i)(\Phi u_i)^T)=$$

$$\sum_{i=1}^{n}\lambda_i Tr\big((\Phi u_i)(\Phi u_i)^T\big)=\sum_{i=1}^{n}\lambda_i Tr\big((\Phi u_i)^T(\Phi u_i)\big)=\sum_{i=1}^{n}\lambda_i\|\Phi u_i\|_2^2 \qquad (6.12)$$

represents a weighted sum of stochastically independent variables $\|\Phi u_i\|_2^2$.

Now, the numerator on the right hand side of the Chernoff inequality can be written as:

$$E\left\{e^{s\sum_{j=1}^{N}\|\Phi x_j\|_2^2}\right\} = E\left\{\prod_{i=1}^{N} e^{s\lambda_i\|\Phi u_j\|_2^2}\right\} = \prod_{i=1}^{N} E\left\{e^{s\lambda_i\|\Phi u_j\|_2^2}\right\}$$

Also, we use the equality $m\sum_{j=1}^{N}\sigma_j^2 = Tr(XX^T) = \sum_{i=1}^{n}\lambda_i$ in the denominator, resulting in

$$\min_{s>0}\frac{\prod_{i=1}^{N} E\left\{e^{s\lambda_i\|\Phi u_j\|_2^2}\right\}}{\prod_{i=1}^{N} e^{s(1+\epsilon)\lambda_i}} = \min_{s>0}\prod_{i=1}^{N}(1-2sm^{-1}\lambda_i)^{-\frac{m}{2}}e^{-s(1+\epsilon)\lambda_i} \qquad (6.13)$$

Taking the derivative with respect to s and setting it equal to zero results in the following familiar equation:

$$\sum_{i=1}^{N}\frac{\lambda_i}{1-2s^*m^{-1}\lambda_i} = (1+\epsilon)\sum_{i=1}^{N}\lambda_i \qquad (6.14)$$

This must be compared with (6.8) which we have rewritten below

$$\sum_{j=1}^{N}\frac{\sigma_j^2}{1-2s^*\sigma_j^2} = (1+\epsilon)\sum_{j=1}^{N}\sigma_j^2$$

Therefore, similar to the random-block case, the Chernoff bound is the smallest when XX^T has a uniform spectrum, that is when $\lambda_1 = \lambda_2 = \cdots = \lambda_n$. However, unlike block-wise energies, eigenvalues of the data matrix are expected to decay when the data is correlated. For most natural images and general natural signals, eigenvalues of X^TX decay exponentially leading to heavy tail bounds for the sum of squares when repeated-block measurements are used.

Closed-form tail bounds for the sum of squares

Fortunately, it is possible to obtain closed-form and interpretable tail bounds for the cases of random-block and repeated-block measurements. Specifically, consider a modified version of the problem, given below, which was used in [34].

Assume that we are measuring a signal vector $x \in \mathbb{R}^{Nn}$ which is composed of N blocks, each of size n. The sampling matrix Φ is block-diagonal where each block on the main diagonal is $\Phi_j \in \mathbb{R}^{m_j \times n}$ (distribution of each Φ_j is specified later in theorems):

$$\Phi = \begin{pmatrix} \Phi_1 & & \\ & \ddots & \\ & & \Phi_N \end{pmatrix}$$

Similarly, the measurement vector $y = \Phi x \in \mathbb{R}^{\sum_{j=1}^{N} m_j}$ consists of N blocks, each of size m_j for $j \in \{1,2, \dots, N\}$. Define the matrix $M = \mathrm{diag}(m_1, \dots, m_N) \in \mathbb{R}^{N \times N}$.

It must be clear that the new formulation is equivalent to our previous formulation when $m_1 = m_2 = \cdots = m_N = m$ and $x = vec(X)$. Given the current problem definition, we have the following theorems about the concentration of $\|y\|_2^2$ around its expected value $\|x\|_2^2$. Specifically, the tail bound for random-block measurements is computed for the general class of subgaussian distributions for the measurement matrix but can be easily customized for Gaussian measurements.

Theorem 6.2 (*CM for random-block measurement*). ([34], Theorem III.1) *Let ϕ denote a subgaussian random variable with zero mean and unit variance and subgaussian norm $\|\phi\|_{\psi_2}$.*

Let $\{\Phi_j\}_{j=1}^{N}$ *be random matrices drawn independently, where each* Φ_j *is populated with i.i.d. realizations of the renormalized random variable* $\frac{\phi}{\sqrt{m_j}}$*. Then,*

$$Pr\{\|y\|_2^2 - \|x\|_2^2 > \epsilon\|x\|_2^2\} \leq 2\, exp\left\{-C_1\, min\left(\frac{C_2^2\epsilon^2}{\|\phi\|_{\psi_2}^4}\Gamma_2(x,M), \frac{C_2\epsilon}{\|\phi\|_{\psi_2}^2}\Gamma_\infty(x,M)\right)\right\} \quad (6.15)$$

where C_1 *and* C_2 *are absolute constants.* $\Gamma_2(x,M)$ *and* $\Gamma_\infty(x,M)$ *are defined below.*

$$\Gamma_2(x,M) = \frac{\|\gamma\|_1^2}{\left\|M^{-\frac{1}{2}}\gamma\right\|_2^2} = \frac{\left(\sum_j^N \|x_j\|_2^2\right)^2}{\sum_j^N \frac{\|x_j\|_2^4}{m_j}} \quad (6.16)$$

$$\Gamma_\infty(x,M) = \frac{\|\gamma\|_1}{\|M^{-1}\gamma\|_\infty} = \frac{\sum_j^N \|x_j\|_2^2}{\max_j \frac{\|x_j\|_2^2}{m_j}} \quad (6.17)$$

where $\gamma = [\|x_1\|_2^2, \|x_2\|_2^2, \ldots, \|x_N\|_2^2]^T \in \mathbb{R}^N$ *represents the vector of block-wise energies.*

Similarly, for repeated block measurements with a Gaussian distribution, the following theorem is provided.

Theorem 6.3 (*CM for repeated-block measurement*). ([34], Theorem III.2) *Let* $\widetilde{\Phi}$ *be a random* $m \times n$ *matrix populated with i.i.d. Gaussian entries having variance* $\sigma^2 = m^{-1}$*, and let* Φ *be an* $mN \times nN$ *block-diagonal matrix as defined above and* $\Phi_j = \widetilde{\Phi}$ *for all* j*. Then, for* $y = \Phi x$*,*

$$Pr\{\|y\|_2^2 - \|x\|_2^2 > \epsilon\|x\|_2^2\} \leq 2\, exp\{-C_1\, min(C_3^2\epsilon^2\Lambda_2(x,m), C_3\epsilon\Lambda_\infty(x,m))\} \quad (6.18)$$

where C_1 *and* C_3 *are absolute constants.* $\Lambda_2(x,m)$ *and* $\Lambda_\infty(x,m)$ *are defined below.*

$$\Lambda_2(x,m) = \frac{m\|\lambda\|_1^2}{\|\lambda\|_2^2} \quad (6.19)$$

$$\Lambda_\infty(x, M) = \frac{m\|\lambda\|_1}{\|\lambda\|_\infty} \qquad (6.20)$$

where $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]^T \in \mathbb{R}^n$ *represents the vector of eigenvalues of* XX^T *when* $X = [x_1, \dots, x_N]$.

Importantly, the following inequality was derived [34] which implies that random-block measurements result in shorter tail bounds than repeated-block measurements:

$$\Lambda_2(x, m) = \frac{m\|\lambda\|_1^2}{\|\lambda\|_2^2} = \frac{m\|\gamma\|_1^2}{\|\gamma\|_2^2 + 2\sum_{i>j}\langle x_i, x_j\rangle^2} \leq \frac{m\|\gamma\|_1^2}{\|\gamma\|_2^2} = \Gamma_2(x, M) \qquad (6.21)$$

when $M = mI_N$.

Therefore, the presence of cross-correlation between blocks $\langle x_i, x_j\rangle$ increases the gap between random-block and repeated-block tail bounds.

ESTIMATION ACCURACY FOR GENERAL CONVEX PROBLEMS

In this section, we present our general learning results. Generally speaking, our goal is to show that optimizing the parameters (in a learning problem) based on random projections of data would still be close to the case if they were optimized over the complete data.

Problem definition

Most learning tasks involve minimizing the expected value of a loss function (with respect to some parameters) over the data distribution. However, when the data distribution is not

known, learning must be carried over a set of training data that is randomly sampled from the (hypothetical) data distribution. The (sample) average of the loss function over the training data is sometimes called the empirical risk and data-driven learning algorithms work by minimizing the empirical risk function.

In our problem, training data consists of random measurements of data and each unit of training data consists of the collection of inner products between every block of data $x_j \in \mathbb{R}^n$ and an outcome of a random sampling operator $\phi_j \in \mathbb{R}^n$. Therefore, the collection of block measurements $\Phi_j x_j$, each of size m, corresponds to m training units and minimizing the average loss function over this training data may be referred to as empirical risk minimization.

We denote the empirical risk function as $R_S(\theta)$: $\mathbb{R}^p \rightarrow \mathbb{R}$ where $\theta \in \mathbb{R}^p$ is the parameter vector to be estimated and S represents the random sampling operator for generating the training data which depends on the set of measurement matrices $\Phi_1, \Phi_2, \ldots, \Phi_N$. Put simply, $R_S(\theta)$ is the loss over the training data. For example,

$$R_S(\theta) = \sum_j^N \ell\big(\Phi_j x_j, \theta\big) \tag{6.22}$$

The true risk is denoted by $R(\theta)$ which relies on the complete data matrix X (which is assumed to be fixed and is not shown as a variable). Note that, we require that the empirical risk be an unbiased estimator of the true risk. Expressed formally,

$$E_S\{R_S(\theta)\} = R(\theta) \tag{6.23}$$

Without additional constraints about the parameter vector, estimating θ involves minimizing the empirical risk function:

$$\hat{\theta}^* = \min_{\theta \in \mathbb{R}^p} R_S(\theta) \tag{6.24}$$

The hat notation over the estimated parameter vector emphasizes the fact that the estimation is based on the training data, rather than the complete data. In other words, $\hat{\theta}^*$ is an approximation of the optimal parameter vector θ^* which is defined as:

$$\theta^* = \min_{\theta \in \mathbb{R}^p} R(\theta) \tag{6.25}$$

We refer to θ^* as the true optimum parameter vector and to $\hat{\theta}^*$ as its estimation.

Naming convention: In some of the more recent works, $R_S(\theta)$ is sometimes called the *sketch* of $R(\theta)$ using the *sketching* S [35]. Similarly, $\hat{\theta}^*$ could be called the sketch of θ^*.

Specification of the training data

In the linear measurement model, each data vector (or signal block) x_j is projected onto a linear subspace, usually of lower dimension than the ambient signal space. By definition, a linear projection does not change the dimensionality of the signal vector. It must be noted that it is more common to work with the vector of linear measurements $y = \Phi x \in \mathbb{R}^m$ $(m < n)$. Nonetheless, the input to the estimation problem of this chapter is the actual linear projection of x (for reasons that become clear later in the chapter):

$$\hat{x} = P_\Phi(x) = [\Phi^T(\Phi\Phi^{\mathrm{T}})^{-1}\Phi]x \tag{6.26}$$

Clearly, $\hat{x}$ contains the same amount of information about the underlying signal as y does in $y = \Phi x$ for a known measurement matrix Φ and y can always be calculated from $\hat{x}$ by $y = \Phi\hat{x}$. Let $S = \Phi^T(\Phi\Phi^{\mathrm{T}})^{-1}\Phi \in \mathbb{R}^{n\times n}$ denote the projection operator. For simplicity, without loss of generality, assume that $\Phi\Phi^{\mathrm{T}} = I$ (which is a typical assumption in most CS works) resulting

in $S = \Phi^T\Phi$. Requiring $\Phi\Phi^T = I$ is equivalent to requiring that rows of Φ are orthonormal, in addition to being independent[10].

Bounding the MSE risk for unconstrained strongly convex problems

Well-posed learning problems involve minimization of a convex risk function. Furthermore, in the class of convex functions, there are different degrees of convexity. *Strong convexity* at $\theta = \theta_1$ presents a form of convexity that requires the objective function to be lower-bounded by a quadratic function with a constant concavity that is tangent to the objective function at $\theta = \theta_1$. The formal definition of strong convexity is given below.

Definition 6.1 (*strong convexity*). $f(\theta)\colon \mathbb{R}^p \to \mathbb{R}$ is a strongly convex function with constant α or equivalently an α-strongly convex function if

$$f(\theta_2) - f(\theta_1) \geq \nabla_\theta f(\theta_1)^T(\theta_2 - \theta_1) + \frac{\alpha}{2}\|\theta_2 - \theta_1\|_2^2 \qquad (6.27)$$

for all θ_1 and θ_2 in its domain.

Strong convexity can be specialized for smooth functions that are at least twice differentiable, leading to an intuitive and useful understanding of strong convexity for smooth functions.

Corollary 6.1 (*strong convexity for smooth functions*). If $f(\theta)$ is twice continuously differentiable with respect to θ, then $f(\theta)$ is strongly convex with parameter α if and only if $\nabla_\theta^2 f(\theta) \succcurlyeq \alpha I$ for any θ in its domain.

[10] Note that we always assume Φ is full-rank for sensing efficiency; that is rows of Φ are independent.

Similar to the conventional estimation frameworks where the estimation error is presented in terms of the Mean Squared Error (MSE), we express the accuracy of estimation from random projections of data in terms of MSE. However, there are fundamental differences between the traditional notion of MSE and the MSE that is used here. Traditionally, MSE is the expected value of squared error with regard to the randomness in x, i.e. when x is withdrawn from a random distribution, and signifies the risk associated with optimizing the parameter vector over the limited outcome(s) of x. Meanwhile, in our framework, x is assumed to be fixed and randomness is with regard to the projection matrix S. Therefore, we are interested in the MSE $E_S\left\{\left\|\hat{\theta}^* - \theta^*\right\|_2^2\right\}$ over the distribution of S.

Clearly, evaluating $E_S\left\{\left\|\hat{\theta}^* - \theta^*\right\|_2^2\right\}$ can be very difficult given that $\hat{\theta}^*$ may not have a closed-form expression as a function of S. Even with such closed-form solution for $\hat{\theta}^*$, it is usually extremely difficult to analytically integrate the error $\left\|\hat{\theta}^* - \theta^*\right\|_2^2$ over the distribution of S. However, in many applications, one might be only interested in an analytical upper bound for the MSE which can be computed with a few simplifying assumptions.

Theorem 6.4. *Given that:*

1) *$R(\theta)$ is an α-strongly convex function of θ*
2) *For any outcome of S, we have $|R_S(\theta) - R(\theta)| \leq \epsilon R(\theta)$ with $\epsilon < 1$*

the squared error is bounded as:

$$\left\|\theta^* - \hat{\theta}^*\right\|_2^2 \leq \frac{4\epsilon}{(1-\epsilon)\alpha} \min_{\theta} R(\theta) \qquad (6.28)$$

Proof. We use the result of the (upcoming) Theorem 6.5 which states that, when $|R_S(\theta) - R(\theta)| \leq \epsilon R(\theta)$ for $\theta \in \mathbb{R}^p$, we have $R\left(\hat{\theta}^*\right) - R(\theta^*) \leq \frac{2\epsilon}{1-\epsilon} R(\theta^*)$ (see (6.32)). Also,

note that for a strongly convex risk, we have $\frac{\alpha}{2}\left\|\hat{\theta}^* - \theta^*\right\|_2^2 \leq R\left(\hat{\theta}^*\right) - R(\theta^*)$ since according to the optimality condition $\nabla_\theta R(\theta^*)^T(\theta - \theta^*) \geq 0$ for any θ (moving in any feasible direction away from θ^* would increase the value of true risk function $R(\theta)$). ■

In simple words, this theorem implies that the estimation error is small given that the objective function is convex and does not depend significantly on the specific choice of S.

Unfortunately, the assumption $|R_S(\theta) - R(\theta)| \leq \epsilon R(\theta)$ may not hold for every choice of S. However, for certain distributions of S, this assumption holds with a high probability. More formally, instead of the deterministic inequality of (6.28), we would have the following stochastic inequality:

$$\Pr\left\{\left\|\theta^* - \hat{\theta}^*\right\|_2^2 > \frac{4\epsilon}{(1-\epsilon)\alpha} \min_\theta R(\theta)\right\} \leq e^{-f(\epsilon)}$$

Clearly, the function $f(\epsilon)$ above or generally the empirical risk tail bound

$$\Pr\{|R_S(\theta) - R(\theta)| > \epsilon R(\theta)\} \leq e^{-f(\epsilon)}$$

depends on the specific choice of the loss function. In the following section, we consider the dictionary update loss function based on the compressive training data.

Bounding the regret, the general case

Define the regret as $R\left(\hat{\theta}^*\right) - \min_\theta R(\theta) = R\left(\hat{\theta}^*\right) - R(\theta^*)$. Regret is an important concept in online learning theory. However, in here, we use a similar definition to assess the quality of the learned parameters using empirical risk minimization. For example, in the

dictionary update problem, regret would represent the difference in the sparse representation error between the case of having only random projections of data and the case of having the complete data matrix.

Furthermore, consider the general constrained optimization problems:

$$\hat{\theta}^* = \arg\min_{\theta \in C} R_S(\theta) \tag{6.29}$$

and

$$\theta^* = \arg\min_{\theta \in C} R(\theta) \tag{6.30}$$

Where $C \subset \mathbb{R}^p$ represents an arbitrary constraint set for the parameters. The following theorem results in a bound on the regret function which combined with the tail bound of the empirical risk function can be used to evaluate the quality of learning from random projection of data.

Theorem 6.5. *Suppose* $|R_S(\theta) - R(\theta)| \leq \epsilon R(\theta)$ *for every* $\theta \in C$ *and* $\epsilon < 1$. *Assume that* $\hat{\theta}^*$ *and* θ^* *are minimizers of* $R_S(\theta)$ *and* $R(\theta)$ *over* C. *Then,*

$$R\left(\hat{\theta}^*\right) \leq \left(\frac{1+\epsilon}{1-\epsilon}\right) R(\theta^*) \tag{6.31}$$

Proof.

$$\begin{aligned}
&R\left(\hat{\theta}^*\right) - R(\theta^*) \\
&= R\left(\hat{\theta}^*\right) - R_S(\theta^*) + R_S(\theta^*) - R(\theta^*) \\
&\leq R\left(\hat{\theta}^*\right) - R_S\left(\hat{\theta}^*\right) + R_S(\theta^*) - R(\theta^*) \\
&\leq \epsilon R\left(\hat{\theta}^*\right) + \epsilon R(\theta^*)
\end{aligned}$$

Therefore

$$(1-\epsilon)R\left(\hat{\theta}^*\right) \leq (1+\epsilon)R(\theta^*) \qquad \blacksquare$$

Therefore, an upper bound for the regret, when $|R_S(\theta) - R(\theta)| \leq \epsilon R(\theta)$, is:

$$R\left(\hat{\theta}^*\right) - R(\theta^*) \leq \frac{2\epsilon}{1-\epsilon} R(\theta^*) \tag{6.32}$$

Note that requiring $|R_S(\theta) - R(\theta)| \leq \epsilon R(\theta)$ over $\theta \in C$ is a weaker condition than requiring $|R_S(\theta) - R(\theta)| \leq \epsilon R(\theta)$ over every $\theta \in \mathbb{R}^p$. For example, for any vector $x \in \mathbb{R}^n$, the squared norm of its random projection $\|\Phi x\|_2^2$ is concentrated around $\|x\|_2^2$ and according to (6.4)

$$\Pr\{|\|\Phi x\|_2^2 - \|x\|_2^2| > \epsilon \|x\|_2^2\} \leq 2e^{-\frac{m\epsilon^2}{6}}$$

This probability of error may not be acceptable for recovery applications where x must be recovered from Φx. However, as discovered in CS, it is possible to perfectly recover a sparse x from random measurements of the form Φ. What happens is that the value of $\Pr\{|\|\Phi x\|_2^2 - \|x\|_2^2| \geq \epsilon \|x\|_2^2\}$ becomes smaller when x is restricted to the k sparse vectors. In fact, ϵ here is precisely the definition of the Restricted Isometry Constant [1] for k sparse vectors which is the cornerstone of CS.

This theorem has significant values and summarizes the intuition behind our research. Specifically, both regularization and diversification attempt to make ϵ smaller. Regularization works for constraining the feasible set C and diversification works by reducing the tail bound $\Pr\{|R_S(\theta) - R(\theta)| > \epsilon R(\theta)\}$.

MSE BOUND FOR DICTIONARY UPDATE FROM RANDOM PROJECTIONS

In this section, we compute stochastic bounds for the MSE risk of the dictionary update step assuming the sparse coefficient are given as part of the training data.

As the first step, the dictionary update loss function must be expressed in the vectorized format. Let $\boldsymbol{d} \in \mathbb{R}^{nd}$ denote the vectorized dictionary, i.e. $\boldsymbol{d} = vec(D)$. For a single block, we can write:

$$\left\|y_j - \Phi_j D a_j\right\|_2^2 = \left(y_j - \Phi_j D a_j\right)^T \left(y_j - \Phi_j D a_j\right) = y_j^T y_j - 2 y_j^T \Phi_j D a_j + a_j^T D^T \Phi_j^T \Phi_j D a_j$$

We can further write;

$$y_j^T \Phi_j D a_j = vec\left(\Phi_j^T y_j a_j^T\right)\boldsymbol{d}$$

and

$$a_j^T D^T \Phi_j^T \Phi_j D a_j = \boldsymbol{d}^T \left(a_j a_j^T \otimes \Phi_j^T \Phi_j\right)\boldsymbol{d}$$

Therefore the dictionary update problem can be written in the standard quadratic form as follows[11]:

$$\widehat{\boldsymbol{d}}^* = \arg\min_{\boldsymbol{d}} \frac{1}{2}\boldsymbol{d}^T Q \boldsymbol{d} + f^T \boldsymbol{d} + c \tag{6.33}$$

Where

$$Q = \sum_{j=1}^N a_j a_j^T \otimes \Phi_j \Phi_j^T \tag{6.34}$$

$$f = vec\left(-\sum_{j=1}^N \Phi_j^T y_j a_j^T\right) \tag{6.35}$$

[11] Of course, this quadratic problem is subject to the constraint $\|\boldsymbol{d}\|_2 \leq \sqrt{n}$. This constraint is typically handled individually by a projection onto the constraint set in the last step of the dictionary update stage.

$$c = \frac{1}{2}\sum_{j=1}^{N} y_j^T y_j \tag{6.36}$$

Correspondingly, the original dictionary update problem (in the presence of complete measurements) can be expressed as:

$$\boldsymbol{d}^* = \arg\min_{\boldsymbol{d}} \frac{1}{2}\boldsymbol{d}^T \bar{Q}\boldsymbol{d} + \bar{f}^T\boldsymbol{d} + \bar{c} \tag{6.37}$$

Where

$$\bar{Q} = \sum_{j=1}^{N} a_j a_j^T \otimes I_n \tag{6.38}$$

$$\bar{f} = vec\left(-\sum_{j=1}^{N} x_j a_j^T\right) \tag{6.39}$$

$$\bar{c} = \frac{1}{2}\sum_{j=1}^{N} x_j^T x_j \tag{6.40}$$

Below, the stochastic construction of the block measurement matrices is specified. For a single block measurement matrix, each entry is sampled from a random i.i.d. Gaussian distribution with zero mean and variance $\frac{1}{m}$. The $\frac{1}{m}$ variance assures that

$$E\left\{\Phi_j^T \Phi_j\right\} = I_n$$

and therefore, empirical risk becomes an unbiased estimator of the true risk as desired.

Although the presented analysis focuses on Gaussian measurements, it is straightforward to extend it to the larger class of sub-Gaussian measurements which includes the Rademacher as well as the general class of (centered) bounded random variables [14].

Define $\ell_S(d) = \frac{1}{2}d^T Q d + f^T d + c$ and $\ell(d) = \frac{1}{2}d^T \bar{Q} d + \bar{f}^T d + \bar{c}$. For the case having a uniform distribution over the residual vectors for random-block measurements, in Section 6.1.2, we computed the tail bound for $\ell_S(d)$, i.e. the upper bound for the stochastic measure:

$$\Pr\{(1-\epsilon)\ell(d) \geq \ell_S(d) \geq (1+\epsilon)\ell(d)\} \leq 2e^{-\frac{mN\epsilon^2}{6}} \tag{6.41}$$

Therefore, we are able to find upper bounds for the MSE $E_S\left\{\left\|\hat{d}^* - d^*\right\|_2^2\right\}$ using the results of Section 6.2.3. In particular,

Theorem 6.6. A*ssuming* $\left\|x_j - Da_j\right\|_2$ *is the same for all* $j \in \{1,2,\dots,N\}$*, with probability* $1 - 2e^{-\frac{mN\epsilon^2}{6}}$ *for* $0 < \epsilon < \frac{1}{2}$,

$$\left\|\hat{d}^* - d^*\right\|_2^2 \leq \frac{4\epsilon}{(1-\epsilon)\mu}\ell(d^*) \tag{6.42}$$

where $\mu = \lambda_{min}(\bar{Q}) = \lambda_{min}(AA^T)$.

Note that the upper bound on the right hand side of the above inequality is deterministic. The above result can be easily generalized to the case of unbalanced residual energies using the close-form bounds described in Section 6.1.4.

CHAPTER 7

HYPERSPECTRAL REMOTE SENSING AND CLASSIFICATION BASED ON RANDOM PROJECTIONS

Hyperspectral images are extensions of monochrome images that correspond to tensors widely known as hyperspectral cubes. In spite of their high dimensionality, hyperspectral cubes are highly redundant and compressible data structures. As a result, there have been numerous proposals for compressive architectures for hyperspectral imaging among which are whisk-broom and push-broom scanners [37], reviewed later in this chapter, that represent practical designs.

In this chapter, we study the problem of hyperspectral pixel classification based on the recently proposed architectures for compressive whisk-broom imagers [37] without the need to reconstruct the hyperspectral cube and by directly estimating the classifier from the random measurements. A clear advantage of classification based on compressively sensed data is its suitability for real-time on-site processing of the sensed data. Moreover, it is assumed that the learning process also takes place in the compressed domain, completely isolating the classification unit from the recovery unit at the ground station. We show that, using distinct

measurement matrices for different pixels results in better accuracy of the learned classifier and gives consistent classification performance, supporting the role of information diversity. At the same time, we show that classification based on using a fixed (but random) measurement matrix is less reliable in general.

INTRODUCTION

A hyperspectral cube (HSC) consists of numerous layers of monochrome images where each layer corresponds to a specific electromagnetic frequency or what is known as a spectral band. An example of an HSC is shown in Figure 17.

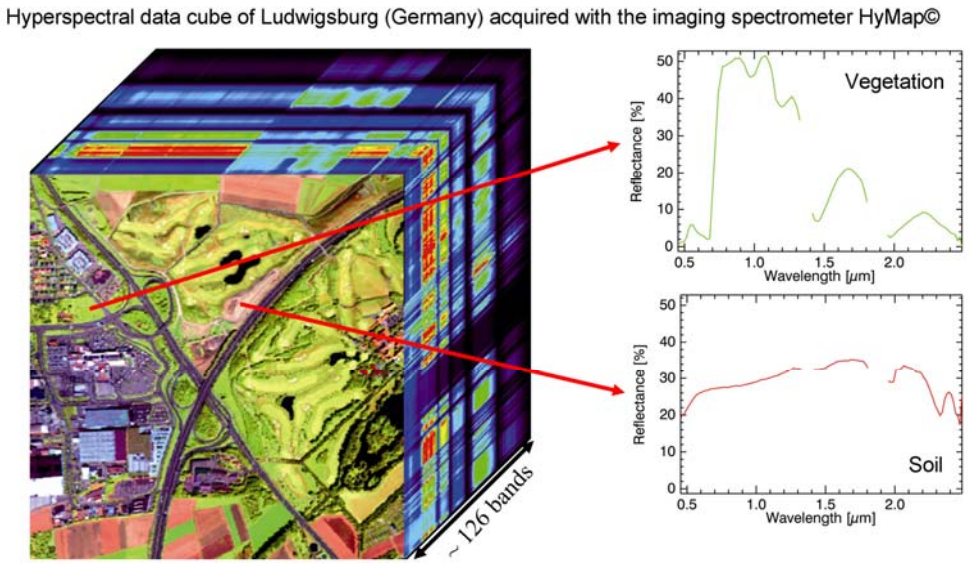


Figure 17. The hyperspectral cube or HSC of an earth patch and the spectral reflectances of two pixels corresponding to vegetation and soil. CO_2 absorption bands are omitted.

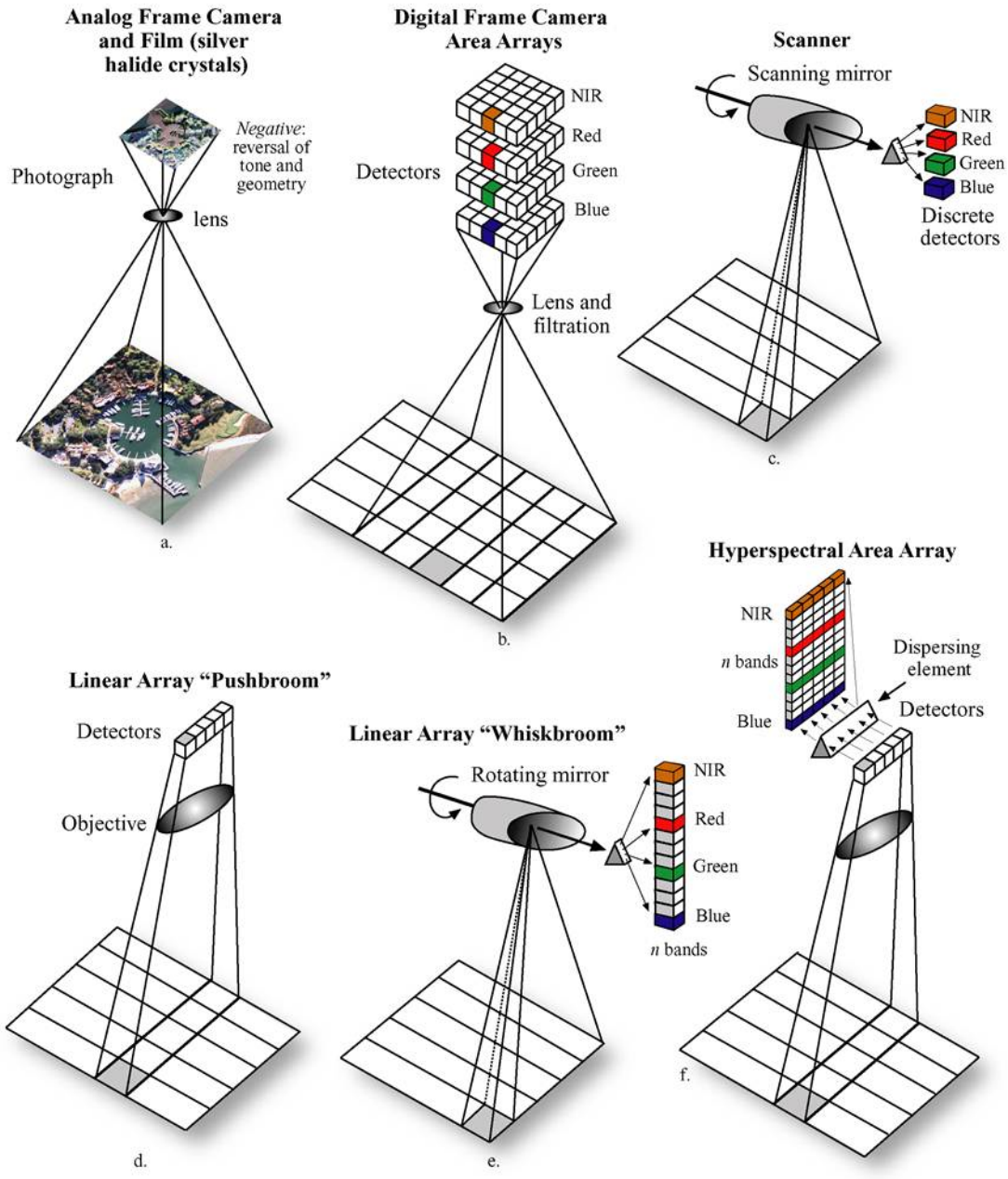


Figure 18. Conceptual diagrams of different types of MSI and HSI sensors, including whisk-broom (e) and push-broom (f) HSI designs. (Photo credit: J. R. Jenson 2007, "Remote Sensing of the Environment: An Earth Resource Perspective," Prentice Hall).

A close family of hyperspectral imaging (HSI) sensors are multispectral imaging (MSI) sensors that consist of coarser spectral resolutions. Specifically, while a typical HSI bandwidth is in the order of $10nm$ (nano-meters), MSI bandwidths are in the order of $100nm$. Figure 18 shows diagrams of the well-known types of MSI and HSI sensors.

Existing challenges in classification of hyperspectral signatures

In remote sensing applications, each spectral signature or endmember is associated with a specific type of material. However, due to the variability of imaging conditions such as the direction of light, variable sizes of objects and atmospheric noise, measured endmembers are subject to variations. This problem, which is widely known as the *endmember variability* problem, makes it difficult to identify objects based on spectral libraries that are usually produced in controlled laboratory conditions [38, 39]. Consequently, classifications algorithms such as the Support Vector Machine (SVM) must be trained for each scenario, possibly for each particular HSC. This necessitates that the training be robust with respect to the noise or missing values. In this chapter, we focus on the latter case, that is when the measurements are not complete and are in the form of linear projections of data vectors onto random low-dimensional subspaces.

Compressive architectures for hyperspectral imaging and their impacts on hyperspectral classification

Recently, there has been a great deal of interest over compressive architectures for hyperspectral imaging and remote sensing. This is mainly due to the increasing amount of hyperspectral data that is being collected by high-resolution airborne imagers such as NASA's AVIRIS (http://aviris.jpl.nasa.gov) and the fact that a large portion of data is discarded during compression or during feature mining prior to learning.

It has been noted in [37] that many of the proposed compressive architectures are based on the spatial mixture of pixels across each frame and correspond to physically costly or impractical operations while most existing airborne hyperspectral imagers employ scanning methods to acquire a pixel or a line of pixels at a time. To address this issue, practical designs of compressive whisk-broom and push-broom cameras were suggested in [37]. These designs are illustrated in Figure 19 and Figure 20.

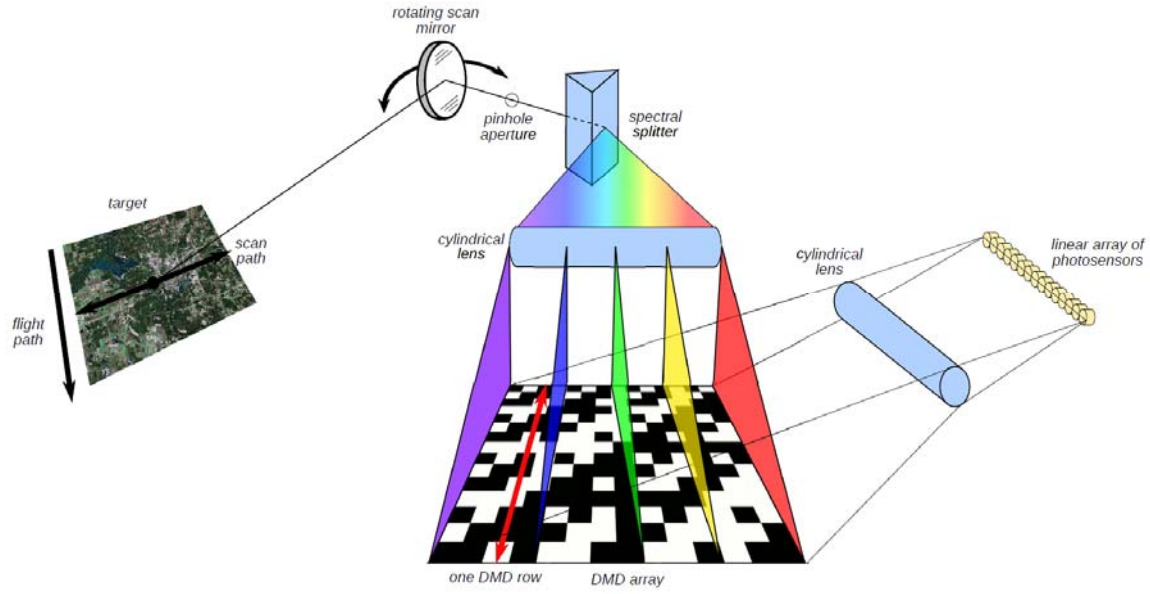


Figure 19. The conceptual compressive whisk-broom camera of [37].

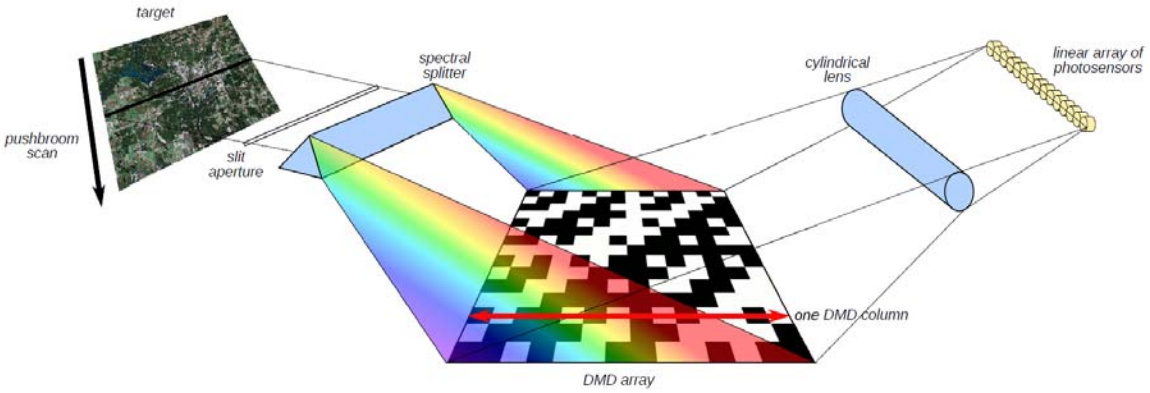


Figure 20. The conceptual compressive push-broom camera of [37].

In this chapter, we study the problem of hyperspectral pixel classification based on compressive whisk-broom sensors; i.e. each pixel is measured at a time using an individual random measurement matrix. Meanwhile, the presented analysis would also apply to compressive push-broom cameras. There have been other efforts focused on the problem of classification from compressive hyperspectral data [40]. To set this work apart from those

efforts, such as [40], we must mention two issues with their typical indirect approach of applying the classification algorithms to the recovered data:

a) The sensed data cannot be decoded at the sender's side (airborne device) due to the heavy computational cost of compressive recovery, making on-site classification infeasible.

b) The number of measurements (per pixel) may not be sufficient for a reliable signal recovery.

It has been established that classification in the compressed domain would succeed with far less number of random measurements than it is required for a full data recovery [11]. However, the compressive framework of [11] corresponds to using a fixed projection matrix for all pixels which limits the measurement diversity that has been promoted by several recent studies for data recovery and learning [41, 42]. Rather than devising new classification algorithms, this work is focused on studying the relationship between the camera's sensing mechanism, namely the employed random measurement matrix, and the common Support Vector Machine (SVM) classifier. It must be emphasized that the general problem of classification based on compressive measurements has been addressed for the case where a fixed measurement matrix is used [11]. However, our aim is to study the impact of measurement diversity on the learned classifier.

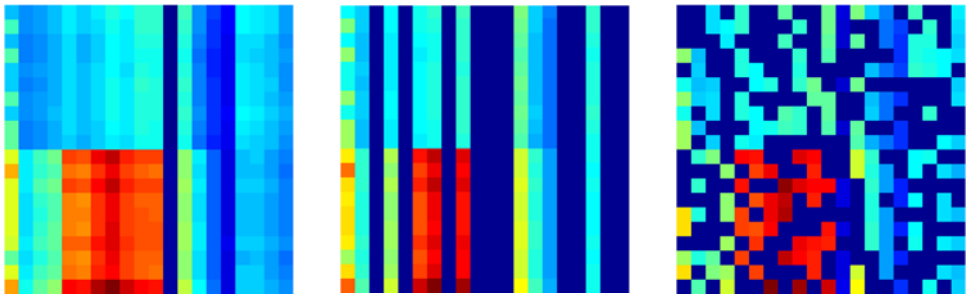
REPEATED-BLOCK AND RANDOM-BLOCK ARCHITECTURES FOR COMPRESSIVE HSI

We investigate two different sensing mechanisms that were introduced in [37]:

1) FCA-based sensor: A Fixed Coded Aperture (FCA) is used to modulate the dispersed light before it is collected at the linear sensor array. This case corresponds to using a fixed measurement matrix for each pixel and a low-cost alternative to the DMD system below.

2) DMD-based sensor: A Digital Micro-mirror Device (DMD) is used to modulate the incoming light according to an arbitrary pattern that is changed for each measurement. Unlike the previous case, DMD adds the option of sensing each pixel using a different measurement matrix.

These two cases are illustrated in Figure 21.



Complete data FCA-sensed data DMD-sensed data

Figure 21. FCA-based versus DMD-based sensing. Rows represent pixels and columns represent spectral bands.

SVM CLASSIFICATION PROBLEM FORMULATION

Support vector machine or SVM has been shown to be a suitable classifier for hyperspectral data. Specifically, we employ an efficient linear SVM classifier with the exponential loss function that gives a smooth approximation to the hinge-loss. To train the classifier in the compressed domain, we must sketch the SVM loss function using the acquired measurements for which we employ some of the techniques developed in [43]. Furthermore, given that the sketched loss function gives a close approximation to the true loss function and that the learning objective function is smooth, it is expected that the learned classifier is close to the ground-truth classifier based on the complete hyperspectral data (which is unknown). As it has been discussed in [44], recovery of the classifier is of independent importance in some applications.

Overview of SVM for spectral pixel classification

In a supervised hyperspectral classification task, a subset of pixels are labeled by a specialist who may have access to the side information about the imaged field such as being physically present at the field for measurement. The task of learning is then to employ the labeled samples for tuning the parameters of the classification machine to predict the pixel labels for a field with similar material compositions. For subpixel targets, an extra stage of spectral unmixing is required to separate different signal sources involved in generating a pixel's spectrum [45]. For simplicity, we assume that the pixels are homogeneous (consist of single objects).

Recall that most classifiers are inherently composed of binary decision rules. Specifically, in multi-categorical classification, multiple binary classifiers are trained according to either One-Against-All (OAA) or One-Against-One (OAO) schemes and voting techniques are employed to combine the results [21]. In a OAA-SVM classification problem, a decision hyperplane is computed between each class and the rest of the training data, while in a OAO scheme, a hyperplane is learned between each pair of classes. As a consequence, most studies focus on the canonical binary classification. Similarly in here, our analysis is presented for the binary classification problem which can be extended to multi-categorical classification.

In the linear SVM classification problem, we are given a set of training data points (corresponding to hyperspectral pixels) $x_j \in \mathbb{R}^d$ for $j \in \{1,2,\dots,N\}$ and the associated labels $z_j \in \{-1,+1\}$. The inferred class label for x_j is $sign\left(x_j^T\omega - b\right)$ that depends on the classifier $\omega \in \mathbb{R}^d$ and the bias term $b \in \mathbb{R}$. The classifier ω is the normal vector to the affine hyperplane that divides the training data in accordance with their labels. When the training classes are inseparable by an affine hyperplane, maximum-margin soft-margin SVM is used which relies on a loss function to penalize the amount of misfit. For example, a widely used loss function is $\ell(r) = (\max\{0,1-r\})^p$ with $r = z_j\left(x_j^T\omega - b\right)$. For $p = 1$, this loss function is known as the hinge loss, and for $p = 2$, it is called the squared hinge loss or simply the quadratic loss. The optimization problem for soft-margin SVM becomes[12]

$$(\omega^*, b^*) = \arg\min_{\omega,b} \frac{1}{n}\sum_{j=1}^{n} \ell\left(z_j\left(x_j^T\omega - b\right)\right) + \frac{\lambda}{2}\|\omega\|_2^2 \qquad (7.1)$$

[12] Discussion: Similar results can be obtained using the dual form. Recent works have shown that advantages of the dual form can be obtained in the primal as well [24]. As noted in [24], the primal form convergences faster to the optimal parameters (ω^*, b^*) than the dual form. For the purposes of this work, it is more convenient to work with the primal form of SVM although the analysis can be properly extended to the dual form.

In this paper, we use the smooth exponential loss function, which can be used to approximate the hinge loss while retaining its margin-maximization properties [46]:

$$\ell(r) = e^{-\gamma r} \tag{7.2}$$

where γ controls the smoothness. We use $\gamma = 1$.

SVM in the compressed domain

Let $y_j = \Phi_j x_j \in \mathbb{R}^{d'}$ denote the low-dimensional measurement vector for pixel j where $d' \leq d$ is size of the photosensor array in the compressive whisk-broom camera [37]. As explained in [47], a DMD architecture can be used to produce a Φ_j with random entries in the range $[0,1]$ or random ± 1 entries, resulting in a sub-Gaussian measurement matrix that satisfies the isometry conditions with a high probability [14]. Recall that the measurement matrix Φ_j is fixed in a FCA-based architecture while it can be distinct for each pixel in a DMD-based architecture.

As noted in [43], the orthogonal projection onto the row space of Φ_j can be computed as $P_j = \Phi_j^T \left(\Phi_j \Phi_j^T\right)^{-1} \Phi_j$. Consequently, an (unbiased) estimator for the inner product $x_j^T \omega$ (assuming a fixed x_j and ω) based on the compressive measurements would be $y_j^T \left(\Phi_j \Phi_j^T\right)^{-1} \Phi_j \omega$. As a result, the soft-margin SVM based on the compressive measurements can be expressed as:

$$\widehat{\omega}^* = \arg\min_{\omega} \frac{1}{n} \sum_j^n \ell\left(z_j y_j^T \left(\Phi_j \Phi_j^T\right)^{-1} \Phi_j \omega\right) + \frac{\lambda}{2} \|\omega\|_2^2 \tag{7.3}$$

We have omitted the bias term b for simplicity for now.

It must be noted that the formulation in (7.3) is different from what was suggested in [11] for a fixed measurement matrix. In particular, we solve for $\widehat{\omega}^*$ in the d-dimensional space. Meanwhile, the methodology in [11] would result in the following optimization problem:

$$\widetilde{\omega}^* = \arg\min_{\omega} \frac{1}{n}\sum_j^n \ell\left(z_j y_j^T \omega\right) + \frac{\lambda}{2}\|\omega\|_2^2 \tag{7.4}$$

which solves for $\widetilde{\omega}^*$ in the low-dimensional column-space of Φ. Also note that, in the case of fixed measurement matrices, (7.3) and (7.4) correspond to the same problem with the relationship $\widehat{\omega}^* = \Phi^T(\Phi\Phi^T)^{-1}\widetilde{\omega}^*$ (because of the ℓ_2 regularization term which zeros the components of $\widehat{\omega}^*$ which lie in the null-space of Φ). In other words, (7.3) represents a generalization of (7.4) for the case when the measurement matrices are not necessarily the same. This allows us to compare the two cases of a) having a fixed measurement matrix and b) having a distinct measurement matrix for each pixel, which is the subject of this paper. For simplicity, assume that each Φ_j consists of a subset of d' rows from a random orthonormal matrix, or equivalently $\Phi_j\Phi_j^T = I_{d'}$; thus, $P_j = \Phi_j^T\Phi_j$. Also assume that, in the case of DMD-based sensing, each Φ_j is generated independently of the other measurement matrices.

Following the recent line of work in the area of randomized optimization, for example [35], we refer to the new loss $\ell\left(z_j x_j^T \Phi_j^T\left(\Phi_j\Phi_j^T\right)^{-1}\Phi_j\omega\right)$ as the sketch of the loss, or simply the sketched loss to distinguish it from the true loss $\ell\left(z_j x_j^T\omega\right)$. Similarly, we refer to $\widehat{\omega}^*$ as the sketched classifier as opposed to the ground-truth classifier ω^*.

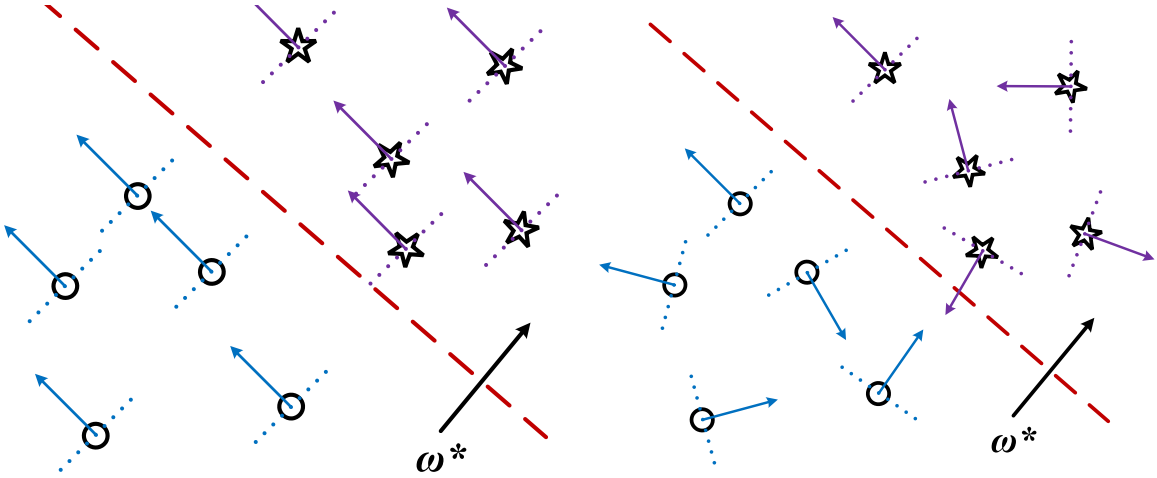


Figure 22. Linear SVM classification depicted for $d = 2$ and $d' = 1$. Each arrow attached to a data point represents the direction of the random projection for that point.

Figure 22 depicts the two cases of using a fixed measurement matrix (FCA-sensed data) and distinct measurement matrices (DMD-sensed data) for training a linear classifier. It is helpful to imagine that, in the sketched problem, each x_j is multiplied with $P_j\omega$ (the projection of ω onto the column-space of Φ_j) since $y_j^T\Phi_j\omega = \left(P_j x_j\right)^T\omega = x_j^T\left(P_j\omega\right)$. As shown in Figure 22 (left) with $P_j = P$ for all j, there is a possibility that ω^* would nearly align with the null-space of the random low-rank matrix $P = \Phi^T\Phi$. For such P, any vector $P\omega$ may not well discriminate between the two classes and ultimately result in the classification failure. Figure 22 (right) depicts the case when a distinct measurement is used for each point. When Φ_j is symmetrically distributed in the space and n is large, there is always a bunch of Φ_j's that nearly align with ω^* whereas other Φ_j's can be nearly orthogonal to ω^* or somewhere between the two extremes. This intuitive example hints about how measurement diversity pays off by making the optimization process more stable with respect to the variations in the random measurements and the separating hyperplane.

Below, we present a simulation to quantify these remarks. Specifically, we look at the distributions of the sketched loss at ω^* for the two sampling cases. Clearly, for a fixed training data and some ω, the sketched loss is a random variable that is a function of the measurement operator. For the optimization to be stable, the sketched loss must not depend significantly on the specific outcome of this measurement operator.

In this (preliminary) simulation we use the Indian Pines dataset [48]. In Figure 23, we have plotted the distributions of sketched loss for the FCA-based sampling (representing measurement without diversity) and DMD-based sampling (measurement with diversity) for a pair of classes with $d = 200$, $d' = 100$ and $n = 200$. Although only evaluated at ω^* in this case, we observe that the pixel-varying measurement results in a more stable (concentrated) sketch of the loss and arguably a closer approximation of the ground-truth classifier.

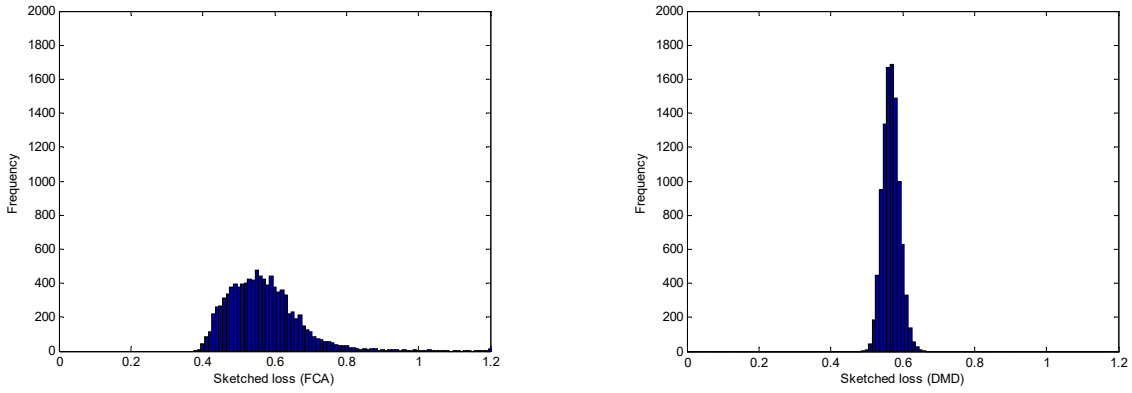


Figure 23. Distributions of the sketched loss for the FCA-based sampling and DMD-based sampling for a pair of classes with $d = 200$, $d' = 100$ and $n = 200$.

THE CLASSIFICATION ALGORITHM

In this section, we provide the details of the optimization algorithm for the smooth soft-margin SVM. However, before that, we need to address the issue of bias variability in the diverse projection scheme.

Handling the bias term

It is not difficult to see that employing a distinct Φ_j for each data vector x_j necessitates having distinct values of bias b_j (for each Φ_j). Note that in the case of fixed measurement matrix, i.e. when $\Phi_j = \Phi$ for all j, bias terms would be all the same and linear SVM works normally as noted in [11]. However, using a customized bias term for each point would clearly result in overfitting and the learned $\widehat{\omega}^*$ would be of no practical value. Furthermore, the classifier cannot be used for prediction since the bias is unknown for the new input samples. In the following, we address these issues.

First, let S denote a set of k distinct measurement matrices, that is

$$S = \left\{\Phi^{(1)}, \Phi^{(2)}, \dots, \Phi^{(k)}\right\} \tag{7.5}$$

Instead of using an arbitrary measurement matrix for each pixel, we draw an entry from S for each pixel. Given that $n \gg k$, each element of S is expected be utilized for more than once. This allows us to learn the bias for each outcome of measurement matrix (without the overfitting issue). Note that k signifies the degree of measurement diversity: $k = 1$ refers to the least

diversity, i.e. using a fixed measurement matrix, and measurement diversity is increased with k. The new optimization problem becomes:

$$(\widehat{\omega}^*, b_1^*, \dots, b_k^*) = \arg \min_{\omega, b_1, \dots, b_k} \frac{\lambda}{2} \|\omega\|_2^2 + \frac{1}{n} \sum_j^n \ell \left(z_j y_j^T \Phi^{(t_j)} \omega + b_{t_j} \right) \qquad (7.6)$$

where t_j randomly (uniformly) maps each $j \in \{1,2, \dots, n\}$ to an element of $\{1,2, \dots, k\}$. The overfitting issue can now be restrained by tuning k; reducing k results in less overfitting. In our simulations, we use $k \geq \left\lceil \frac{d}{d'} \right\rceil$ to ensure that S spans $\mathbb{R}^d$ with a probability close to one.

For prediction, the corresponding bias term is selected from the set $\{b_1^*, b_2^*, \dots, b_k^*\}$.

Implementation of gradient descent for SVM

Here, we describe the details of the Newton's method for optimizing the linear SVM classification problem (with the exponential loss) for the unbiased case. The biased version is quite similar and follows the same path with the additional bias variable(s).

The objective function for the linear SVM classification problem is:

$$\widehat{\omega}^* = \arg \min_{\omega} \frac{1}{n} \sum_{j=1}^n \exp\left(-z_j y_j^T \Phi_j \omega \right) + \frac{\lambda}{2} \omega^T \omega \qquad (7.7)$$

For Newton's method, we need to compute the gradient vector and the Hessian matrix at each intermediate solution $\omega^{(t)}$. Let $\nabla_\omega f\left(\omega^{(t)}\right)$ denote the gradient at step t and let $H_\omega\left(\omega^{(t)}\right)$ denote the Hessian matrix at step t. Using basic vector calculus,

$$\nabla_\omega f\left(\omega^{(t)}\right) = -\frac{1}{n} \sum_j^n z_j \Phi_j^T y_j \exp\left(-z_j y_j^T \Phi_j \omega^{(t)}\right) + \lambda \omega^{(t)} \qquad (7.8)$$

and

$$H_{\omega}\left(\omega^{(t)}\right) = \frac{1}{n}\sum_{j}^{n} \Phi_j^T y_j y_j^T \Phi_j \exp\left(-z_j y_j^T \Phi_j \omega^{(t)}\right) + \lambda I \tag{7.9}$$

According to the update rule of the Newton's method,

$$\omega^{(t+1)} = \omega^{(t)} - H_{\omega}^{-1}\left(\omega^{(t)}\right)\nabla_{\omega} f\left(\omega^{(t)}\right) \tag{7.10}$$

Note that $\lambda_{min}(H_{\omega}) \geq \lambda$ guarantees the numerical stability of the algorithm. The algorithm is stopped when the desired numerical precision is achieved for $\omega^{(t)}$.

SIMULATION RESULTS

The dataset used in this section is the well-known Pavia University dataset [49] which is available with the ground-truth labels[13] [14]. For each experiment, we perform a 2-fold cross-validation with 1000 training and 1000 testing samples. As discussed earlier, multi-categorical SVM classification algorithms typically rely on pair-wise or One-Against-One (OAO) classification results. Hence, we evaluate the sketched classifier on a OAO basis by reporting the pair-wise performances in a table. Finally, since the measurement operator is random and subject to variation in each experiment, we repeat each experiment for 1000 times and perform a worst-case analysis of the results.

Consider the case where a single measurement is made from each pixel, i.e. $d' = 1$ and $\Phi_j \in \mathbb{R}^{1\times d}$ is a random vector in the d-dimensional spectral space. Clearly, this case represents

[13] http://www.ehu.eus/ccwintco/

[14] The Indian Pines dataset was not included due to the small size of the image which is not sufficient for a large-scale cross-validation study.

an extreme scenario where the signal recovery would not be reliable and classification in the compressed domain becomes crucial, even at the receiver's side where the computational cost is not of greatest concern. For performance evaluation, we are interested in two aspects: (a) the prediction accuracy over the test dataset, (b) the recovery accuracy of the classifier (with respect to the ground-truth classifier) –whose importance has been discussed in [44].

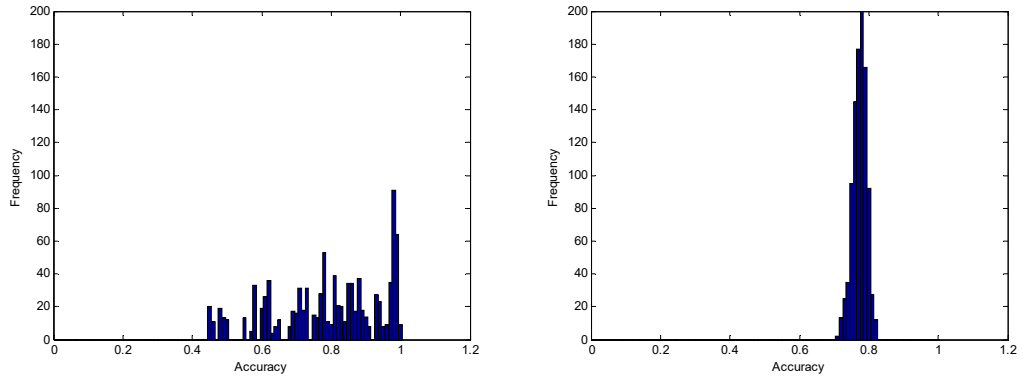


Figure 24. Distributions of the classification accuracy (Asphalt vs. Meadows) for the Pavia University dataset ($d' = 1$).

We define the classification accuracy as the minimum (worst) of the True Positive Rate (sensitivity) and the True Negative Rate (specificity). Figure 24 shows an instance of the distribution of the classification accuracy for a pair of classes over 1000 random trials. As it can be seen, in the presence of measurement diversity, classification results are more consistent (reflected in the low variance of accuracy). Due to the limited space, we only report the worst-case OAO accuracies (i.e. the minimum pair-wise accuracies among 1000 trials) for the Pavia scene.

The results for the case of one-measurement-per-pixel ($d' = 1$) are shown in Table 7 and Table 8. Similarly, the results for the case of $d' = 3$ (which is equivalent to the sampling rate of a typical RGB color camera) are shown in Table 9 and Table 10. Note that the employed SVM

classifier is linear and would not result in perfect accuracy (i.e. accuracy of one) when the classes are not linearly separable. To see this, we have reported ground-truth accuracies in Table 11.

To measure the classifier recovery accuracy, we compute the cosine similarity, or equivalently the correlation, between $\widehat{\omega}^*$ and ω^*:

$$C(\widehat{\omega}^*, \omega^*) = \frac{\langle \widehat{\omega}^*, \omega^* \rangle}{\|\widehat{\omega}^*\|_2 \|\omega^*\|_2} \tag{7.11}$$

In Table 12 and Table 13, we have reported the average recovery accuracy for the case of three-measurements-per-pixel (i.e. $d' = 3$).

Table 7. One FCA measurement per pixel: worst-case classification accuracies (for 1000 trials) for the Pavia scene.

Classes	Meadow	Gravel	Trees	Soil	Bricks
Asphalt	0.45	0.38	0.42	0.36	0.44
Meadow		0.48	0.48	0.41	0.47
Gravel			0.44	0.44	0.44
Trees				0.42	0.53
Soil					0.44

Table 8. One DMD measurement per pixel: worst-case classification accuracies (for 1000 trials) for the Pavia scene

Classes	Meadow	Gravel	Trees	Soil	Bricks
Asphalt	0.71	0.64	0.79	0.60	0.71
Meadow		0.72	0.61	0.46	0.73
Gravel			0.79	0.60	0.44
Trees				0.69	0.79
Soil					0.60

Table 9. Three FCA measurement per pixel: worst-case classification accuracies (for 1000 trials) for the Pavia scene

Classes	Meadow	Gravel	Trees	Soil	Bricks
Asphalt	0.61	0.80	0.94	0.63	0.86
Meadow		0.67	0.82	0.50	0.62
Gravel			0.94	0.62	0.54
Trees				0.89	0.93
Soil					0.66

Table 10. Three DMD measurement per pixel: worst-case classification accuracies (for 1000 trials) for the Pavia scene

Classes	Meadow	Gravel	Trees	Soil	Bricks
Asphalt	0.91	0.76	0.96	0.87	0.84
Meadow		0.90	0.82	0.57	0.91
Gravel			0.95	0.82	0.49
Trees				0.93	0.96
Soil					0.80

Table 11. Ground-truth accuracies for the Pavia scene.

Classes	Meadow	Gravel	Trees	Soil	Bricks
Asphalt	1.00	0.97	0.97	1.00	0.94
Meadow		0.99	0.96	0.89	0.99
Gravel			1.00	1.00	0.86
Trees				0.98	1.00
Soil					0.99

Table 12. Three FCA measurement per pixel: average recovery accuracy (for 1000 trials) for the Pavia scene

Classes	Meadow	Gravel	Trees	Soil	Bricks
Asphalt	0.051	0.055	0.113	0.056	0.048
Meadow		0.100	0.033	0.019	0.077
Gravel			0.122	0.064	0.050
Trees				0.017	0.123
Soil					0.031

Table 13. Three DMD measurement per pixel: average recovery accuracy (for 1000 trials) for the Pavia scene

Classes	Meadow	Gravel	Trees	Soil	Bricks
Asphalt	0.164	0.189	0.483	0.129	0.132
Meadow		0.468	0.147	0.140	0.380
Gravel			0.617	0.272	0.197
Trees				0.102	0.582
Soil					0.128

CONCLUSION

In the field of ensemble learning, it has been discovered that the diversity among the base learners enhances the overall learning performance [21]. Meanwhile, our aim has been to exploit the diversity that can be efficiently built into the sensing system. Both measurement schemes of pixel-invariant (measurement without diversity) and pixel-varying (measurement with diversity) have been suggested as practical designs for compressive hyperspectral cameras [37]. The presented analysis indicates that employing a DMD would result in more accurate recovery of the classifier and a more stable classification performance compared to the case when an FCA is used. Meanwhile, for tasks that only concern class prediction (and not the recovery of the classifier), FCA is (on average) a suitable low-cost alternative to the DMD architecture.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] E. J. Candes, J. Romberg and T. Tao. "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Information Theory*, Vol. 52(2006), 489-509.

[2] D. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, Vol. 52(4), pp. 1289 - 1306, April 2006.

[3] Ledoux, M, "The concentration of measure phenomenon," *Mathematical Surveys and Monographs 89, . American Mathematical Society*, Providence, RI, 2001.

[4] M. Aharon, M. Elad and A. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," IEEE *Transactions on 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 *Transactions on Image Processing*, vol.15, no.12, pp.3736-3745, Dec 2006.

[6] S. Gleichman and Y. Eldar, "Blind compressed sensing," IEEE Transactions on Information Theory, vol. 57, no. 10, pp. 69586975, 2011.

[7] J. Silva, M. Chen, Y. C. Eldar, G. Sapiro and L. Carin, "Blind compressed sensing over a structured union of subspaces," arXiv:1103.2469v1, 2011.

[8] B. Recht, "A simpler approach to matrix completion," arXiv:0910.0651, 2009.

[9] C. Studer and R. Baraniuk, "Dictionary learning from sparsely corrupted or compressed signals," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 33413344, 2012.

[10] G. Peyre, "Best basis compressed sensing," IEEE *Transactions on Signal Processing*, vol.58, no.5, pp.2613-2622, May 2010.

[11] Robert Calderbank, Sina Jafarpour, and Robert Schapire, "Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain," .

[12] R. Rubinstein, M. Zibulevsky and M. Elad, "Double sparsity: learning sparse dictionaries for sparse signal approximation," IEEE *Transactions on Signal Processing*, vol.58, no.3, pp.1553-1564, March 2010.

[13] M. B. Wakin, J. Y. Park, H. L. Yap, and C. J. Rozell, "Concentration of measure for block diagonal measurement matrices," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, March 2010.

[14] R. Baraniuk, M. Davenport, R. DeVore and M. Wakin, “A simple proof of the restricted isometry property for random matrices,” Constructive Approx., vol. 28, no. 3, pp. 253–263, Dec. 2008.

[15] E. J. Candès, “The restricted isometry property and its implications for compressed sensing,” *Compte Rendus de l’Academie des Sciences*, vol. Serie I, no. 346, pp. 589–592, 2008.

[16] Efron, T. Hastie, and R. Tibshirani, “Least angle regression,” *Annals of Statistics*, 32:407–499, 2004.

[17] D. Donoho and Y. Tsaig, “Fast Solution of L1-Norm Minimization Problems When the Solution May be Sparse.” Inst. Comput. Math. Eng., Stanford Univ., Stanford, CA, 2006 [Online]. Available: http://www.stanford.edu/~tsaig, Tech. Rep.

[18] B. A. Olshausen and D. J. Field, “Sparse coding with an overcomplete basis set: A strategy employed by V1?,” Vis. Res., vol. 37, pp. 311325, 1997.

[19] R. Gribonval and K. Schnass, “Dictionary identification-sparse matrix-factorisation via l1-minimisation,” IEEE Transactions on Information Theory, 56(7):3523-3539, 2010.

[20] M. Aharon and M. Elad, “Sparse and redundant modeling of image content using an image signature dictionary,” SIAM Journal on Imaging Sciences, 1(3):228-247, July 2008.

[21] B. Waske, S. Van Der Linden, J.A. Benediktsson, A. Rabe and P. Hostert, “Sensitivity of support vector machines to random feature selection in classification of hyperspectral data,” IEEE Transactions on Geoscience and Remote Sensing, vol. 48, pp. 28802889, 2010.

[22] F. Melgani and L. Bruzzone, “Classification of hyperspectral remote sensing images with support vector machines,” IEEE Transactions on Geoscience and Remote Sensing, vol. 42, no. 8, pp. 17781790, August 2004.

[23] S. Gunn, “Support vector machines for classification and regression,” Technical report, Image Speech and Intelligent Systems Group, Department of Electronics and Computer Science, University of Southampton, 1998.

[24] O. Chapelle, “Training a support vector machine in the primal,” Neural Computing, vol. 19(5), pp. 11551178, 2007.

[25] M. Aghagolzadeh and H. Radha, “Compressive dictionary learning for image recovery,” *IEEE International Conference on Image Processing (ICIP)*, Sep 2012.

[26] Williams C.K.I., “Prediction with Gaussian processes: From linear regression to linear prediction and beyond,” In: Jordan M.I. (Ed.), Learning and Inference in Graphical Models, Kluwer Academic, pp. 599–621, 1998.

[27] Tarantola A, 2005, Inverse Problem Theory, Society for Industrial and Applied Mathematics, ISBN 0-89871-572-5.

[28] A. Buades, B. Coll and J. Morel, “A non-local algorithm for image denoising,” *in Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2005.

[29] K. Deb, “Optimization for Engineering Design: Algorithms and Examples,” Prentice-Hall, New Delhi, 1995.

[30] I. Tosic and P. Frossard, “Dictionary learning,” IEEE Signal Process. Mag., vol. 28, no. 2, pp. 27–38, Mar. 2011.

[31] J. Mairal, F. Bach, J. Ponce and G. Sapiro, “Online dictionary learning for sparse coding,” In Proceedings of the 26th *Annual International Conference on Machine Learning* (ICML ’09), ACM, pp.689-696, New York, NY, USA, 2009.

[32] S. Cotter, B. D. Rao, K. Engan and K. Kreutz-Delgado, “Sparse solutions of linear inverse problems with multiple measurement vectors,” *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2477Ð2488, July 2005.

[33] R.M. Willett, M.F. Duarte, M.A. Davenport, and R.G. Baraniuk, “Sparsity and structure in hyperspectral imaging: Sensing, reconstruction, and target detection,” Signal Processing Magazine, IEEE, vol. 31, no. 1, pp. 116–126, Jan 2014.

[34] J.Y. Park, H.L. Yap, C.J. Rozell and M.B.Wakin, “Concentration of measure for block diagonal matrices with applications to compressive signal processing,” IEEE Transactions on Signal Processing, 59(12):5859-5875, 2011.

[35] M. Pilanci, Martin J. Wainwright, “Randomized Sketches of Convex Programs with Sharp Guarantees,” arXiv:1404.7203 [cs.IT], April 2014.

[36] M. Golbabaee and P. Vandergheynst, “Hyperspectral image compressed sensing via low-rank and joint sparse matrix recovery,” In Indernational Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011.

[37] J.E. Fowler, “Compressive pushbroom and whiskbroom sensing for hyperspectral remote-sensing imaging,” in Proceedings of the International Conference on Image Processing, IEEE, ICIP 2014, October 2014, pp. 684–688.

[38] B. Somers, G. P. Asner, L. Tits and P. Coppin, “Endmember variability in spectral mixture analysis: A review,” Remote Sensing of Environment, vol. 115, no. 7, pp. 16031616, July 2011.

[39] A. Zare and K.C. Ho, "Endmember Variability in Hyperspectral Analysis: Addressing Spectral Variability During Spectral Unmixing," Signal Processing Magazine, IEEE, vol.31, no.1, pp.95,104, January 2014.

[40] J.E. Fowler, Qian Du, Wei Zhu, and N.H. Younan, "Classification performance of random-projection-based dimensionality reduction of hyperspectral imagery," in Geoscience and Remote Sensing Symposium, 2009 IEEE International, IGARSS 2009, July 2009, vol. 5, pp. V–76–V–79.

[41] M. Aghagolzadeh and H. Radha, "Adaptive dictionaries for compressive imaging," in Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE, Dec 2013, pp. 1033–1036.

[42] A. Krishnamurthy, M. Azizyan, and A. Singh, "Subspace Learning from Extremely Compressed Measurements," ArXiv e-prints, Apr. 2014.

[43] M.A. Davenport, P.T. Boufounos, M.B. Wakin, and R.G. Baraniuk, "Signal processing with compressive measurements," Selected Topics in Signal Processing, IEEE Journal of, vol. 4, no. 2, pp. 445–460, April 2010.

[44] Lijun Zhang, M. Mahdavi, Rong Jin, Tianbao Yang, and Shenghuo Zhu, "Random projections for classification: A recovery approach," Information Theory, IEEE Transactions on, vol. 60, no. 11, pp. 7300– 7316, Nov 2014.

[45] W.K. Ma, J.M. Bioucas Dias, Tsung Han Chan, N. Gillis, P. Gader, A.J. Plaza, A. Ambikapathi and Chong-Yung Chi, "A Signal Processing Perspective on Hyperspectral Unmixing: Insights from Remote Sensing," Signal Processing Magazine, IEEE, vol.31, no.1, pp.67,81, January 2014.

[46] Saharon Rosset, Ji Zhu, and Trevor Hastie, "Margin maximizing loss functions," in In NIPS, 2004.

[47] M.F. Duarte, M.A. Davenport, D. Takhar, J.N. Laska, Ting Sun, K.F. Kelly, and R.G. Baraniuk, "Single-pixel imaging via compressive sampling," Signal Processing Magazine, IEEE, vol. 25, no. 2, pp. 83–91, March 2008.

[48] This dataset was gathered by AVIRIS sensor over the Indian Pines test site in North-western Indiana and consists of 145 by 145 pixels and 224 spectral reflectance bands in the wavelength range 0.4 to 2.5e-6 meters. Indian Pines data are available through Pursue's university MultiSpec site.

[49] This scene was acquired by the ROSIS sensor during a flight campaign over Pavia, northern Italy. The number of spectral bands is 103 and the spatial resolution is 610 by 610 pixels. Ground-truth consists of 9 classes.

[50] Tyrone Vincent, Luis Tenorio and Michael Wakin, "Concentration of measure: fundamentals and tools," lecture notes.