Technical Report

TR-2007-024

Overcomplete dictionary design by empirical risk minimization

by

Lior Horesh, Eldad Haber

Mathematics and Computer Science

EMORY UNIVERSITY

Overcomplete Dictionary Design by Empirical Risk Minimization

L Horesh* and E Haber†

December 18, 2007

Abstract

Recently, there have been a growing interest in application of sparse representation for inverse problems. Most studies concentrated in devising ways for sparsely representing a solution using a given prototype overcomplete dictionary. Very few studies have addressed the more challenging problem of construction of an optimal overcomplete dictionary, and even these were primarily devoted to the sparse coding application.

In this paper we present a new approach for dictionary learning. Our approach is based on minimizing the empirical risk, given some training models. We present a mathematical formulation and an algorithmic framework to achieve this goal. The proposed framework offers incorporation of non-injective and nonlinear operators, where the data and the recovered parameters may reside in different spaces. We test our algorithm and show that it yields optimal dictionaries for diverse problems.

keywords sparse representation, overcomplete dictionary, empirical risk, constrained optimization, optimal design, non-linear

1 Introduction

We consider a discrete ill-posed inverse problem of the following form

$$J(\boldsymbol{m}) + \boldsymbol{\eta} = \boldsymbol{d}$$

where $\boldsymbol{m} \in \mathbb{R}^n$ is the model, $J : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is the forward operator, the acquired data is $\boldsymbol{d} \in \mathbb{R}^k$ and $\boldsymbol{\eta} \in \mathbb{R}^k$ is the noise assumed to be Gaussian and iid. We assume that the operator J is ill-posed, under-determined and possibly non-linear.

Traditionally, the general aim is to recover the model $\boldsymbol{m}$ from the noisy data $\boldsymbol{d}$. However, since the problem is ill-posed regularization is needed. One possible way to regularize the

*Mathematics and Computer Science, Emory University, Atlanta, 30322, GA, USA
†Mathematics and Computer Science, Emory University, Atlanta, 30322, GA, USA

problem is by using a Tikhonov-like regularization and solve the following optimization problem

$$\widehat{\boldsymbol{m}} = \underset{\boldsymbol{m}}{\operatorname{argmin}} \ \frac{1}{2}\|J(\boldsymbol{m}) - \boldsymbol{d}\|_2^2 + \alpha R(\boldsymbol{m})$$

where R is a regularization functional which imposes a-priori information into the solution and α is a regularization parameter.

A different approach for regularization which gained popularity in the recent years is implicit regularization by using sparse representation. The underlying assumption is that the true solution can be described by a small number of parameters (principle of parsimony), and that the model can be accurately represented by a sparse set of prototype atoms from an overcomplete dictionary D. A common and convenient relation between the dictionary and the sparse code is through a linear generative model

$$\boldsymbol{m} = D\boldsymbol{u}$$

where $\boldsymbol{u}$ is a sparse coefficient vector, i.e. it is mostly zeros besides a small subset. Using the linear model it is possible to solve for $\boldsymbol{m}$ through $\boldsymbol{u}$ by solving the following optimization problem

$$\min_{\boldsymbol{u}} \ \frac{1}{2}\|J(D\boldsymbol{u}) - \boldsymbol{d}\|_2^2 + \alpha\|\boldsymbol{u}\|_1 \tag{1}$$

where α is a regularization parameter.

Equivalently, a noiseless data acquisition problem can be considered, where $\eta \to 0$. In such case the following constrained optimization problem can be formulated

$$\min_{\boldsymbol{u}} \quad \|\boldsymbol{u}\|_1 \tag{2a}$$

$$\text{s.t} \quad J(D\boldsymbol{u}) - \boldsymbol{d} = 0 \tag{2b}$$

Sparse representation offers independence on the one hand, and expressiveness and flexibility in matching the data on the other hand. The idea of getting sparse solutions dates back to the 70's [4]. This concept had gained considerable attention since the work of Olshausen and Field [15, 14] who suggested that the visual cortex in mammalian brain employs a sparse coding strategy.

One can distinguish between two challenges in the field:

- *Sparse representation* - finding a sparse vector $\boldsymbol{u}$ given the data $\boldsymbol{d}$ and a **given** overcomplete dictionary D
- *Dictionary design* - construction of an optimal dictionary D which in conjuncture with the first goal promotes parsimonious representation

Recently a large volume of studies have primarily addressed the first problem, while the more involved problem of dictionary design was seldom tackled.

The goal of this paper is to explore a new approach for dictionary design. To do so we assume the availability of a set of examples $\{\boldsymbol{m}_1, .., \boldsymbol{m}_s\}$. Our goal is to evaluate an optimal dictionary given the set of examples.

The idea of learning a dictionary from a set of models is not new. Numerous algorithms for dictionary learning based on optimization of different probabilistic entities or other heuristics were developed. All the work known to us addressed the degenerated version of the problem above, where J was set to be an identity operator. Olshausen and Field [14] developed an Approximate Maximum Likelihood (AML) approach for updating the dictionary, Lewicki and Sejnowski [11] developed an extension of Independent Component Analysis (ICA) to overcomplete dictionaries, a Maximum A - Posteriori (MAP) framework combined with the notion of relative convexity was suggested by Kneutz-Delgado et al by using the Focal Underdetermined System Solver. This algorithm was further improved by the employment of Column Normalized Dictionary Prior (FOCUSS - CNDL) [10].An Expectation Maximization (EM) approach with Adjustable Variable Gaussian inducing prior was introduced in the Sparse Bayesian Learning algorithm (SBL-AVG) [21, 26]. A similar variational approach was also presented in [7]. However, despite the promising potential of this approach for solving the dictionary design problem, it was employed so far only for sparse coding.Most recently, the K-SVD algorithm was developed by Aharon and Elad [1, 6]. This algorithm facilitated singular value decomposition of an error expression for its reduction. Within the dictionary updating process, each dictionary column (atom) was sequentially and independently updated.

Non of the approaches presented above can natively be modified to handle the situation where J is underdetermined. In fact, the case of underdetermined J is mathematically different than the case of well-posed J. The main issue is that for a well-posed J it is possible to recover the exact model when the noise $\boldsymbol{\eta}$ reduces to 0, while for ill-posed problems this is obviously not the case. For ill-posed problems regularization inevitably introduces bias into the solution. Adding the "correct" bias implies more accurate results, hence, the goal of dictionary design is not only to overcome noise but also to complete missing information in the data.

In order to achieve the above goal we base our approach on minimizing the empirical risk. To solve the optimization problem obtained by the learning process we use Sequential Quadratic Programming (SQP) [12, 8].

The paper is organized as follows, in Section 2, the mathematical and statistical frameworks for a novel dictionary design approach are introduced. Later, in Section 3, a noiseless and noisy data formulations of the forward problem are brought. In Section 4 two corresponding formulations for the dictionary design problem are introduced. Later in this chapter, several computational and numerical aspects of the proposed optimization framework are discussed. In Section 5 we bring some numerical results for problems of different scales, for a non-injective super-resolution transformation J as well as for an injective gaussian kernel. In Section 6 the numerical results are discussed. Finally, in Section 7 the paper is summarized and future challenges are proposed.

2 Mathematical Framework for Dictionary Learning

The goal of this section is to develop a mathematical framework for the estimation of a dictionary D to be used for the solution of the inverse problem via equation (1). The construction of an optimal dictionary requires an optimality criterion. One such obvious criterion is, how well the dictionary works for our particular inverse problem. To do that, we define the loss function L

$$L(\boldsymbol{m}, D) = \frac{1}{2}\|\widehat{\boldsymbol{m}}(D, \boldsymbol{m}) - \boldsymbol{m}\|_2^2 \tag{3}$$

where $\widehat{\boldsymbol{m}}$ is obtained by the solution $\boldsymbol{u}$ of the following optimization problem

$$\begin{aligned} \min_{\boldsymbol{u}} \quad & \|\boldsymbol{u}\|_1 \\ \text{s.t} \quad & J(D\boldsymbol{u}) - J(\boldsymbol{m}) = 0 \end{aligned} \tag{4}$$

for the noiseless case and

$$\min_{\boldsymbol{u}} \ \frac{1}{2}\|J(D\boldsymbol{u}) - J(\boldsymbol{m}) - \boldsymbol{\eta}\|_2^2 + \alpha\|\boldsymbol{u}\|_1 \tag{5}$$

in the noisy case.

Various different loss functions than the one prescribed above can be considered, e.g. a semi-norm that focuses in a specific region of interest, or a distance measure for edges. Obviously, such choice need to be elected individually according to the requirements of the application.

Given a model $\boldsymbol{m}$ and a noise realization $\boldsymbol{\eta}$ an optimal dictionary should provide superior model reconstruction over a dictionary that does not comply with the optimality criterion. There are two problems in using the loss function as a criterion for optimality. First, the function depends on the random variable $\boldsymbol{\eta}$ and second, the problem depends on the particular (unknown) model. It may be that one dictionary is particularly effective in recovering one model but may perform badly for others.

To eliminate the dependency of our function with respect to the noise we take the expected value and define the risk

$$\text{risk}(\boldsymbol{m}, D) = \frac{1}{2}\ \mathbf{E}_{\boldsymbol{\eta}}\, \|\widehat{\boldsymbol{m}}(D, \boldsymbol{m}) - \boldsymbol{m}\|_2^2 \tag{6}$$

where $\mathbf{E}_{\boldsymbol{\eta}}$ is the expectation with respect to the noise. The expectation eliminates the noise but we are left with the unknown model. There are several approaches to eliminate the model from (6). One approach is to assume that the model $\boldsymbol{m}$ belongs to a convex set $\mathcal{M}$ and to look at the worst case scenario. This leads to the minimax estimator [19]. A different approach is to assume that $\boldsymbol{m}$ is associated with some probability measure and to integrate over that probability, that is, to obtain the expected value with respect to $\boldsymbol{m}$ as well. This is known as a Bayesian estimator. The main difficulty in this case is that such a distribution is rarely available in practice. Nevertheless, if we assume that such a probability density

function exists and that we are able to extract s samples out of it, then we can approximate the expected value with respect to $\boldsymbol{m}$ by a simple average. Thus, we define the optimal dictionary as the dictionary that minimizes

$$\widehat{D} = \underset{D}{\operatorname{argmin}} \ \frac{1}{2s} \mathbf{E}_{\boldsymbol{\eta}} \sum_{i=1}^{s} \ \|\widehat{\boldsymbol{m}}_i(D, \boldsymbol{m}_i) - \boldsymbol{m}_i\|_2^2 \tag{7}$$

The fundamental underlying assumption in this process is that we are able to obtain a training set $\{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_s\}$ of plausible models and that these models are samples from some density function. Although this assumption is difficult to verify in practice, it has been used successfully in the past [16]. In fact, such assumption is the basis for empirical risk minimization and to support vector machines (SVM) [23, 5].

A discussion regarding the possible methods for extracting such set is beyond the scope of this study. In some applications the data can be selected by a professional, e.g. a clinician and in others generated by computer simulation. Our approach is also useful for the Bayesian case, where Monte-Carlo sampling is used to obtain the examples.

From the above formulation it is evident that an optimal dictionary for a particular forward model, would differ from an optimal dictionary of another. Thus, the forward problem and the noise model play a primary role in the design of an optimal dictionary. Such a property is absent when the forward problem is well-posed.

We now derive a numerical framework for the solution of both variants of the problem (noisy and noiseless). It is important to acknowledge that for each variant, two problems need to be addressed. The *forward problem* of recovering the model $\boldsymbol{m}$ for a **given** dictionary D and a *design problem* of constructing D given the examples. For the forward problem we either need to solve (1) or (2) while for the design problem we need to solve (7). Since the solution of the design problem is intimately related to the solution of the forward problem, the latter is addressed first.

In the following section we will address the forward problem of the noisy and the noiseless data case independently. This septation will be retained for the design problem as well.

3 Solving the Forward Problem

We now consider solving the forward problem in the noiseless and the noisy case. Although there are many (sophisticated) approaches for the solution of the problems [22] we prefer to use a simple approach. This is because the nonlinear system which is solved for the forward problem is part of the necessary conditions for optimization in the design problem and needs to be differentiated again. We therefore use the Iterated Reweighted Least Squares (IRLS) approach [13]. IRLS was successfully used for ℓ_1 inversion in many practical scenarios [24, 18, 25].

3.1 Solving the Noiseless Forward Problem

Considering noiseless acquisition of the data, a data fit term is imposed as an equality constraint. The forward problem is then expressed by

$$\min_{\boldsymbol{u}} \quad \|u\|_1 \tag{8a}$$

$$\text{s.t} \quad JD\boldsymbol{u} - \boldsymbol{d} = 0 \tag{8b}$$

To use the IRLS we replaced the ℓ_1-norm by a smoothed version of the absolute value function $\|u\|_{1,\epsilon}$ where

$$|t|_\epsilon := \sqrt{t^2 + \epsilon} \text{ and } \|u\|_{1,\epsilon} := \sum_i |u_i|_\epsilon$$

Next, we use a variation of Newton's method to solve the problem. The Lagrangian and its derivatives are brought by

$$\mathcal{L} = \|u\|_{1,\epsilon} + \boldsymbol{\xi}^\top (JD\boldsymbol{u} - \boldsymbol{d})$$

and the Euler-Lagrange equations are

$$\begin{aligned} \mathcal{L}_{\boldsymbol{u}} &= \operatorname{diag}\left(\frac{1}{|\boldsymbol{u}|_\epsilon}\right)\boldsymbol{u} + D^\top J^\top \boldsymbol{\xi} = 0 \\ \mathcal{L}_{\boldsymbol{\xi}} &= JD\boldsymbol{u} - \boldsymbol{d} = 0 \end{aligned}$$

The approximate k^{th} Newton's step is obtained by solving the system

$$\begin{pmatrix} \operatorname{diag}\left(\frac{1}{|\boldsymbol{u}_k|_\epsilon}\right) & D^\top J^\top \\ JD & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta \boldsymbol{\xi} \end{pmatrix} = -\boldsymbol{\nabla}\mathcal{L}$$

We use $\operatorname{diag}\left(\frac{1}{|\boldsymbol{u}_k|_\epsilon}\right)$ as an approximation to the (1,1) block. This is commonly done for IRLS or lagged diffusivity [20]. Each iteration is followed by a line search to provide the desired solution for $\boldsymbol{u}$ and $\boldsymbol{\xi}$. The following merit function was employed within that procedure (see [12])

$$\varphi = \|u\|_{1,\epsilon} + \gamma |\boldsymbol{\xi}^\top (JD\boldsymbol{u} - \boldsymbol{d})|.$$

3.2 Solving the Forward Problem for Noisy Data

Here again we use the IRLS approach. The necessary conditions for a minimum are

$$g(\boldsymbol{u}; D) = D^\top J^\top (JD\boldsymbol{u} - \boldsymbol{d}) + \alpha \operatorname{diag}\left(\frac{1}{|\boldsymbol{u}|_\epsilon}\right)\boldsymbol{u} = 0 \tag{10}$$

At the k^{th} IRLS iteration we solve

$$\left(D^\top J^\top JD + \alpha \operatorname{diag}\left(\frac{1}{|\boldsymbol{u}_k|_\epsilon}\right)\right)\delta\boldsymbol{u} = -g(\boldsymbol{u}; D)$$

Each iteration is followed by a weak line search to guarantee sufficient reduction of the objective function.

4 Solving the Dictionary Design Problem

We now describe a methodology for the solution of the design problem. For simplicity we first introduce the noisy case where smaller number of variables need to be considered. We then extend the framework to the noiseless case.

4.1 Solving the Dictionary Design Problem for Noisy Data

It is possible to use a global dictionary D, however, this can lead to a large dense matrix inversion problem and to an excessively difficult design problem. We therefore use a different approach to construct D. The idea is similar to the one presented in [9, 6]. We divide the model to υ overlapping patches, each of size l. Let P_j extract be j^{th} patch of the image. Then, we have

$$P^j \boldsymbol{m} = D_L \boldsymbol{u}^j$$

where $D_L \in \mathbb{R}^{q\times r}$ denotes a local dictionary which is assumed to be invariant for the entire model. This can also be written as

$$P\boldsymbol{m} = (I \otimes D_L)\boldsymbol{u}$$

where $P = \text{diag}(P^j)$, $I \in \mathbb{R}^{\upsilon\times\upsilon}$ is an identity matrix and $\boldsymbol{u} = [\boldsymbol{u}^1, \ldots, \boldsymbol{u}^\upsilon]$. Assuming consistency we can rewrite $\boldsymbol{m}$ as

$$\boldsymbol{m} = (P^\top P)^{-1} P^\top (I \otimes D_L)\boldsymbol{u} = D\boldsymbol{u}$$

where

$$D := (P^\top P)^{-1} P^\top (I \otimes D_L)$$

This leads to the constrained optimization problem

$$\begin{aligned} \min_{\boldsymbol{u},D} \quad & \frac{1}{2}\sum_{i=1}^{s} \|D\boldsymbol{u}_i - \boldsymbol{m}_i\|_2^2 \\ \text{s.t} \quad & g(\boldsymbol{u}_i, D) = 0 \quad i = 1, \ldots, s \end{aligned}$$

This is a non-linear equality constrained optimization problem. One robust way to solve such problems is by Sequential Quadratic Programming (SQP). The Lagrangian $\mathcal{L}$ and its derivatives w.r.t. $\boldsymbol{u}_i$, D and $\boldsymbol{\lambda}_i$ are brought by

$$\mathcal{L} = \sum_{i=1}^{s} \frac{1}{2}\left\|\hat{D}\boldsymbol{u}_i - \boldsymbol{m}_i\right\|_2^2 + \sum_{i=1}^{s} \boldsymbol{\lambda}_i^\top g(\boldsymbol{u}_i; D)$$

and

$$\mathcal{L}_{\boldsymbol{u}_i} = D^\top (D\boldsymbol{u}_i - \boldsymbol{m}_i) + \left(\frac{\partial g_{\boldsymbol{u}_i}}{\partial \boldsymbol{u}_i}\right)^\top \boldsymbol{\lambda}_i = 0 \tag{12a}$$

$$\mathcal{L}_D = \sum_{i=1}^{s} U_i^\top (D\boldsymbol{u}_i - \boldsymbol{m}_i) + \left(\frac{\partial g_{\boldsymbol{u}_i}}{\partial D}\right)^\top \boldsymbol{\lambda}_i = 0 \tag{12b}$$

$$\mathcal{L}_{\boldsymbol{\lambda}_i} = D^\top J^\top (JD\boldsymbol{u}_i - \boldsymbol{b}_i) + \alpha\, \text{diag}\left(\frac{1}{|\boldsymbol{u}_i|_\epsilon}\right)\boldsymbol{u}_i = 0 \tag{12c}$$

where the derivative of $g(\boldsymbol{u}_i, D)$ w.r.t. $\boldsymbol{u}_i$ is

$$g_{u_i} := \frac{\partial g(\boldsymbol{u}_i, D)}{\partial \boldsymbol{u}_i} = D^\top J^\top J D + \alpha \operatorname{diag}\left(\frac{\epsilon}{|\boldsymbol{u}_i|_\epsilon^3}\right)$$

and the derivative of $g(\boldsymbol{u}_i, D)$ w.r.t. the dictionary D is more involved

$$g_D := \sum_{i=1}^{s} \frac{\partial g(\boldsymbol{u}_i; D)}{\partial D} = T_i^\top + D^\top J^\top J U_i - Y_i^\top = 0 \tag{13}$$

where

$$\begin{aligned} T_i^\top &:= \frac{\partial}{\partial D} D^\top \underbrace{J^\top J D \boldsymbol{u}_i}_{t_i} \\ Y_i^\top &:= \frac{\partial}{\partial D} D^\top \underbrace{J^\top \boldsymbol{b}_i}_{y_i} \\ U_i &:= \frac{\partial}{\partial D} D \underbrace{\boldsymbol{u}_i}_{u_i}. \end{aligned}$$

These matrix derivatives (13) have the following structure

$$U_i = \begin{pmatrix} (U_i^1)^\top & (U_i^2)^\top & \dots & (U_i^\upsilon)^\top \end{pmatrix}^\top$$

where U_i^j is defined through $\frac{\partial D \boldsymbol{u}_i^j}{\partial D} = U_i^j + D\frac{\partial \boldsymbol{u}_i^j}{\partial D}$ and $U_i^j := I_{l\times l} \otimes (\boldsymbol{u}_i^j)^\top$. Likewise, for $\mathbf{t}_i^j := J^\top J D \boldsymbol{u}_i^j$, the derivative $T_i^{j^\top} := \frac{\partial D^\top \mathbf{t}_i^j}{\partial D} + D^\top \frac{\partial \mathbf{t}_i^j}{\partial D}$ is structured as $T_i^{j^\top} = I_{r\times r} \otimes (\mathbf{t}_i^j)^\top$. In a similar manner to $T_i^\top$, $Y_i^\top$ is defined. For notational coherence, multiple example variables are concatenated

$$\begin{aligned} &\boldsymbol{u} = \begin{pmatrix} (\boldsymbol{u}_1)^\top & (\boldsymbol{u}_2)^\top & \dots & (\boldsymbol{u}_s)^\top \end{pmatrix}^\top, \boldsymbol{m} = \begin{pmatrix} (\boldsymbol{m}_1)^\top & (\boldsymbol{m}_2)^\top & \dots & (\boldsymbol{m}_s)^\top \end{pmatrix}^\top, \\ &\boldsymbol{b} = \begin{pmatrix} (\boldsymbol{b}_1)^\top & (\boldsymbol{b}_2)^\top & \dots & (\boldsymbol{b}_s)^\top \end{pmatrix}^\top \end{aligned}$$

and similarly do vector and matrix derivatives, e.g.

$$U = \begin{pmatrix} (U_1)^\top & (U_2)^\top & \dots & (U_s)^\top \end{pmatrix}^\top \tag{15}$$

Respectively, the operator J and the dictionary D were replaced by the Kronecker products $\mathcal{J} := I \otimes J$ and $\mathcal{D} := I \otimes D$, where $I \in \mathbb{R}^{s\times s}$ is the identity.

For some applications, employment of distinct patches may offer computational advantage. However, when patches completely lack overlapping area, an additional penalty term is required in order to ensure smooth transition at the edges between neighboring patches. Here, utilization of an edge-gradient operator $\mathcal{G} := I \otimes G$ where $G \in \mathbb{R}^{2n\times n}$ in a quadratic penalty term of the form $\|\mathcal{G}\mathcal{D}\boldsymbol{u}\|_2^2$ is proposed. The derivative of this term with respect to $\boldsymbol{u}$ can be added to $g(\boldsymbol{u}; \mathcal{D})$ and accordingly its derivatives can be updated.

4.2 Solving the KKT System for the Noisy Case

By linearization the following KKT system can be derived

$$\begin{pmatrix} \mathcal{D}^\top \mathcal{D} & 0 & \tilde{g_{\boldsymbol{u}}}^\top \\ 0 & U^\top U & g_{\mathcal{D}}^\top \\ \tilde{g_{\boldsymbol{u}}} & g_{\mathcal{D}} & 0 \end{pmatrix} \begin{pmatrix} \delta \boldsymbol{u} \\ \delta \mathcal{D} \\ \delta \boldsymbol{\lambda} \end{pmatrix} = - \begin{pmatrix} \mathcal{L}_{\boldsymbol{u}} \\ \mathcal{L}_{\mathcal{D}} \\ \mathcal{L}_{\boldsymbol{\lambda}} \end{pmatrix} \tag{16}$$

where due to conditioning considerations the IRLS approximation for g_u was taken

$$\tilde{g}_u \simeq \mathcal{D}^\top \mathcal{J}^\top \mathcal{J} \mathcal{D} + \alpha \operatorname{diag}\left(\frac{1}{|\boldsymbol{u}|_\epsilon} \right)$$

This large-scale system is symmetric indefinite and typically ill-conditioned. Apart from $g_{\mathcal{D}}$ all other components of the Hessian are block diagonal and therefore can be processed independently. A sparsity pattern of the Hessian can be found in (4.2). A straight forward

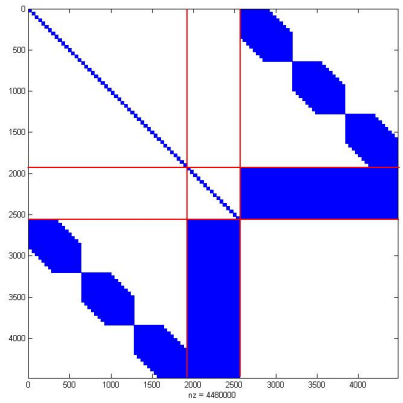

Figure 1: Sparsity pattern of the Hessian of the noisy data KKT system

approach for solving this system would be by employing an appropriate Krylov subspace solver, e.g. FGMRes solver with another Krylov subspace solver as a preconditioner [17]. However, this approach does not exploit the partial block-diagonal sparsity pattern characterizing this matrix [2]. Another approach, which typically handles better the Hessian's ill-conditioning and also do benefit from the partial block-diagonal sparsity pattern is by elimination (reduced SQP) [12]. We begin by elimination of $\delta \boldsymbol{u}$ as follows

$$\tilde{g_u} \delta \boldsymbol{u} + g_{\mathcal{D}} \delta \mathcal{D} = -\mathcal{L}_{\boldsymbol{\lambda}} \tag{17a}$$

$$\delta \boldsymbol{u} = -\tilde{g_u}^{-1} \left(\mathcal{L}_{\boldsymbol{\lambda}} + g_{\mathcal{D}}^\top \delta \mathcal{D} \right). \tag{17b}$$

Further, $\delta \boldsymbol{\lambda}$ can be isolated

$$-\mathcal{D}^\top \mathcal{D} \tilde{g_u}^{-1} \left(\mathcal{L}_{\boldsymbol{\lambda}} + g_{\mathcal{D}} \delta \mathcal{D} \right) + \tilde{g_u}^\top \delta \boldsymbol{\lambda} = -\mathcal{L}_{\boldsymbol{u}} \tag{18a}$$

$$\delta \boldsymbol{\lambda} = \tilde{g_u}^{-\top} \left(\mathcal{D}^\top \mathcal{D} \tilde{g_u}^{-1} \left(\mathcal{L}_{\boldsymbol{\lambda}} + g_{\mathcal{D}} \delta \mathcal{D} \right) - \mathcal{L}_{\boldsymbol{u}} \right) \tag{18b}$$

and lastly, $\delta\mathcal{D}$ can be derived as follows

$$U^\top U\delta\mathcal{D} + g_\mathcal{D}\tilde{g_u}^{-\top}\left(\mathcal{D}^\top\mathcal{D}\tilde{g_u}^{-1}\left(\mathcal{L}_{\boldsymbol{\lambda}} + g_\mathcal{D}\delta\mathcal{D}\right) - \mathcal{L}_{\boldsymbol{u}}\right) = -\mathcal{L}_\mathcal{D} \tag{19a}$$
$$\left(U^\top U + g_\mathcal{D}\tilde{g_u}^{-\top}\mathcal{D}^\top\mathcal{D}\tilde{g_u}^{-1}g_\mathcal{D}\right)\delta\mathcal{D} = \tilde{g_u}^{-\top}\mathcal{L}_{\boldsymbol{u}} - \tilde{g_u}^{-\top}\mathcal{D}^\top\mathcal{D}\tilde{g_u}^{-1}\mathcal{L}_{\boldsymbol{\lambda}} - \mathcal{L}_D$$

we shall denote $K := -\mathcal{D}\tilde{g_u}^{-1}g_\mathcal{D}$, and then obtain the following relation for $\delta\mathcal{D}$

$$\left(U^\top U + K^\top K\right)\delta\mathcal{D} = \tilde{g_u}^{-\top}\mathcal{L}_{\boldsymbol{u}} - K^\top\mathcal{D}\tilde{g_u}^{-1}\mathcal{L}_{\boldsymbol{\lambda}} - \mathcal{L}_D$$

The dimensions of this system are only bounded by the number of atoms in the dictionary, thus, the reduced system is typically substantially smaller than the original SQP system (16). At this point, three strategies can be employed in order to retrieve $\delta\mathcal{D}$. The first approach would be to solve (19b) to obtain $\delta\mathcal{D}$, then substitute $\delta\mathcal{D}$ in (18b) to obtain $\delta\boldsymbol{\lambda}$ and finally, substitute $\delta\boldsymbol{\lambda}$ in (17b) to obtain $\delta\boldsymbol{u}$. The second approach would be to solve (19b) and perform a line search, while $\mathcal{D}$ and $\boldsymbol{\lambda}$ are maintained fixed, i.e. $\mathcal{D}^{(k+1)} \leftarrow \mathcal{D}^{(k)} + \beta^{(k)}\delta\mathcal{D}$. Then use the updated dictionary for solving (13), while $\boldsymbol{\lambda}$ remains fixed. Finally use the relation for $\mathcal{L}_{\boldsymbol{u}}$, given in (12b) to obtain $\boldsymbol{\lambda}$ itself (rather than $\partial\boldsymbol{\lambda}$). The third alternative is sequential elimination of $\boldsymbol{u}$, $\boldsymbol{\lambda}$ and then $\mathcal{D}$. In the first stage $\boldsymbol{u}$ is solved assuming $\boldsymbol{\lambda}$ and $\mathcal{D}$ are given, using relation (12c) $\mathcal{L}_{\boldsymbol{\lambda}} = 0$. Afterwards, $\boldsymbol{\lambda}$ can be obtained from (12b) $\mathcal{L}_{\boldsymbol{u}} = 0$ using the updated $\boldsymbol{u}$ from the previous step. In the last stage, the update for $\mathcal{D}$ can either be obtained using relation (12c) $\mathcal{L}_D = 0$ and the updated $\boldsymbol{u}$ and $\boldsymbol{\lambda}$, or by using the Lagrangian derivative directly within a steepest descent, or L-BFGS optimization scheme, to obtain the updated dictionary $\mathcal{D}^{(k+1)} = \mathcal{D}^{(k)} - \beta^{(k)}\mathcal{L}_D$.

In this study, after experimenting with the different variants, the first strategy was employed, nevertheless, the performance of any optimization scheme is problem-dependent and therefore, implementation of any other scheme may be advantageous on a different setup.

For cases where the Hessian is excessively large or the operator $\mathcal{J}$ is given only in a matrix-vector form (i.e. $\mathcal{J}\boldsymbol{m}$), it is possible to solve this system by an implicit formation of the Hessian and the gradient by using a suitable Krylov subspace solver. Such iterative solver offers control over the desired solution accuracy. Typically, an inexact solution is sufficient and therefore redundant computational effort can be spared [8].

4.3 Solving the Dictionary Design Problem for Noiseless Data

In the noiseless case the following constrained optimization problem is considered

$$\begin{aligned}
\min_{\boldsymbol{u},D} \quad & \frac{1}{2}\sum_{i=1}^{s}\|D\boldsymbol{u}_i - \boldsymbol{m}_i\|_2^2 \\
\text{s.t} \quad & \operatorname{diag}\left(\frac{1}{|\boldsymbol{u}_i|_{1+\epsilon}}\right)\boldsymbol{u}_i + D^\top J^\top\boldsymbol{\xi}_i = 0 \quad i = 1,\ldots,s \\
& JD\boldsymbol{u}_i - \boldsymbol{d}_i = 0 \quad i = 1,\ldots,s
\end{aligned}$$

Similarly to the noisy case SQP is employed for solving this nonlinear equality constrained optimization problem, where this time we have two equality constraints rather than a single

one. The necessary conditions for a minimum are

$$\begin{aligned}
\mathcal{L}_{\boldsymbol{u}_i} &= D^\top (D\boldsymbol{u}_i - \boldsymbol{m}_i) + \operatorname{diag}\left(\frac{\epsilon}{|\boldsymbol{u}_i|_\epsilon^3}\right)\boldsymbol{\rho}_i + D^\top J^\top \boldsymbol{\varrho}_i = 0 \\
\mathcal{L}_D &= \sum_{i=1}^{s} U_i^\top (D\boldsymbol{u}_i - \boldsymbol{m}_i) + U^\top J^\top \boldsymbol{\varrho}_i + Z_i^\top = 0 \\
\mathcal{L}_{\boldsymbol{\rho}_i} &= \operatorname{diag}\left(\frac{1}{|\boldsymbol{u}_i|_\epsilon}\right)\boldsymbol{u}_i = 0 \\
\mathcal{L}_{\boldsymbol{\varrho}_i} &= JD\boldsymbol{u}_i - \boldsymbol{d}_i = 0 \\
\mathcal{L}_{\boldsymbol{\xi}_i} &= JD\boldsymbol{\rho}_i = 0
\end{aligned}$$

where $Z_i^\top := \frac{\partial}{\partial D} D^\top J^\top \boldsymbol{\xi}_i$ and U_i are defined similarly as in (15), and $\boldsymbol{\rho}$, $\boldsymbol{\varrho}$ are Lagrange multipliers.

Similar to the noisy case, in case distinct patches are desired, utilization of an edge-gradient operator $\mathcal{G}$ in a quadratic penalty term $\|\mathcal{G}\mathcal{D}\boldsymbol{u}\|_2^2$ can be incorporated into the Lagrangian.

4.4 Solving the KKT System for the Noiseless Case

By linearization an approximation for the second derivatives can be acquired. Here as well, numerical instabilities are avoided by approximating $\mathcal{L}_{\boldsymbol{u}_i\boldsymbol{\rho}_i}$ to be

$$\mathcal{L}_{\boldsymbol{u}_i\boldsymbol{\rho}_i} \approx \operatorname{diag}\left(\frac{1}{|\boldsymbol{u}_i|_\epsilon}\right)$$

which is again the IRLS approximation used previously for the forward problem. The overall KKT system in multi-image notation gets the form

$$\begin{pmatrix}
\mathcal{D}^\top\mathcal{D} & 0 & \operatorname{diag}\left(\frac{1}{|\boldsymbol{u}|_\epsilon}\right) & \mathcal{D}^\top\mathcal{J}^\top & 0 \\
0 & U^\top U & Z^\top & U^\top\mathcal{J}^\top & 0 \\
\operatorname{diag}\left(\frac{1}{|\boldsymbol{u}|_\epsilon}\right) & Z & 0 & 0 & \mathcal{D}^\top\mathcal{J}^\top \\
\mathcal{J}\mathcal{D} & \mathcal{J}U & 0 & 0 & 0 \\
0 & 0 & \mathcal{J}\mathcal{D} & 0 & 0
\end{pmatrix}
\begin{pmatrix} \delta\boldsymbol{u} \\ \delta\mathcal{D} \\ \delta\boldsymbol{\rho} \\ \delta\boldsymbol{\varrho} \\ \delta\boldsymbol{\xi} \end{pmatrix}
= -
\begin{pmatrix} \mathcal{L}_{\boldsymbol{u}} \\ \mathcal{L}_{\mathcal{D}} \\ \mathcal{L}_{\boldsymbol{\rho}} \\ \mathcal{L}_{\boldsymbol{\varrho}} \\ \mathcal{L}_{\boldsymbol{\xi}} \end{pmatrix}$$

This large - scale system possess similar structure and properties as the one introduced in the noisy case (symmetric, indefinite and ill-conditioned). Apart from $\mathcal{L}_{\boldsymbol{u}\boldsymbol{\rho}}$ and $\mathcal{L}_{\boldsymbol{u}\boldsymbol{\varrho}}$ all other components of the Hessian are block diagonal and therefore can be processed in a block-wised manner. Here, the system was solved using a MinRes Krylov solver.

5 Numerical Studies

Performance evaluation of both the noisy and the noiseless variants of the proposed methodology were committed, where naturally the main emphasis was drawn to the realistic noisy

scenario, while the noiseless case was evaluated mainly for comparison purposes. In addition, the proposed methods were tested in both overlapping and distinct patches settings. The former was tested over a small synthetic data set, whereas the latter which considered distinct patches and incorporated additional patch-edge penalty term, was tested over large-scale realistic problems of natural images.

Testing procedure consisted of two separate stages: dictionary learning and performance assessment, as described in the following.

5.1 Dictionary Learning

On the first stage, a dictionary was trained using an initial prototype dictionary D_0 and training data set $\{\boldsymbol{d}_1, .., \boldsymbol{d}_s\}$, which corresponded to particular training model set $\{\boldsymbol{m}_1, .., \boldsymbol{m}_s\}$ through the transformation J. For noisy data dictionary training was obtained by solving the dictionary design problem given in (4.1), whereas for noiseless data the problem given in (4.3) was solved.

As initial prototype dictionaries Discrete Cosine Transform (DCT), feature and random overcomplete dictionaries were considered. All dictionaries were verified to be of a full column rank.

Following the rational presented in [10] regarding dictionary normalization, the norms of atoms in the dictionary (columns) were normalized to 1, and accordingly, the corresponding values in the sparse code $\boldsymbol{u}$ were adjusted. This procedure assisted in maintaining scaling invariance of the different components of the constraints, and provided a better control over the learning rate.

Termination of the learning process was predefined by two criteria, the ratio between the initial and the current sum of the absolute values of the Lagrangian derivatives, and the absolute value of that current sum. In cases where line search broke, the solution was projected to the constraints to maintain feasibility and from there an attempt to further improve parameter recovery was made.

5.2 Performance Assessment

The performance of the acquired trained dictionary D_t was compared with that of the original prototype dictionary D_0 in solving the forward problem for given various data sets $\boldsymbol{d}$. First, we solved the forward problem for the training models $\{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_s\}$ given $\{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_s\}$ using both D_0 and D_t. By construction of D_t we expect achieving better results in the recovery of the training models. Next, we used a separate set of models $\{\widetilde{\boldsymbol{m}}_1, \ldots, \widetilde{\boldsymbol{m}}_z\}$, we refer to as a validating set and their associated data $\{\widetilde{\boldsymbol{d}}_1, \ldots, \widetilde{\boldsymbol{d}}_z\}$. We now use D_0 and D_t in attempt to recover the validating set from the data $\{\widetilde{\boldsymbol{d}}_1, \ldots, \widetilde{\boldsymbol{d}}_z\}$. Note that the validating set is not used for dictionary training and is only used for assessment purposes.

Let $\widehat{\boldsymbol{m}}_j(D_0)$ be the recovered j^{th} training model using D_0, let $\widehat{\boldsymbol{m}}_j(D_t)$ be the recovered j^{th} model using D_t. Finally, let $\widehat{\widetilde{\boldsymbol{m}}}_j(D_0)$ and $\widehat{\widetilde{\boldsymbol{m}}}_j(D_t)$ be the recovered validating models

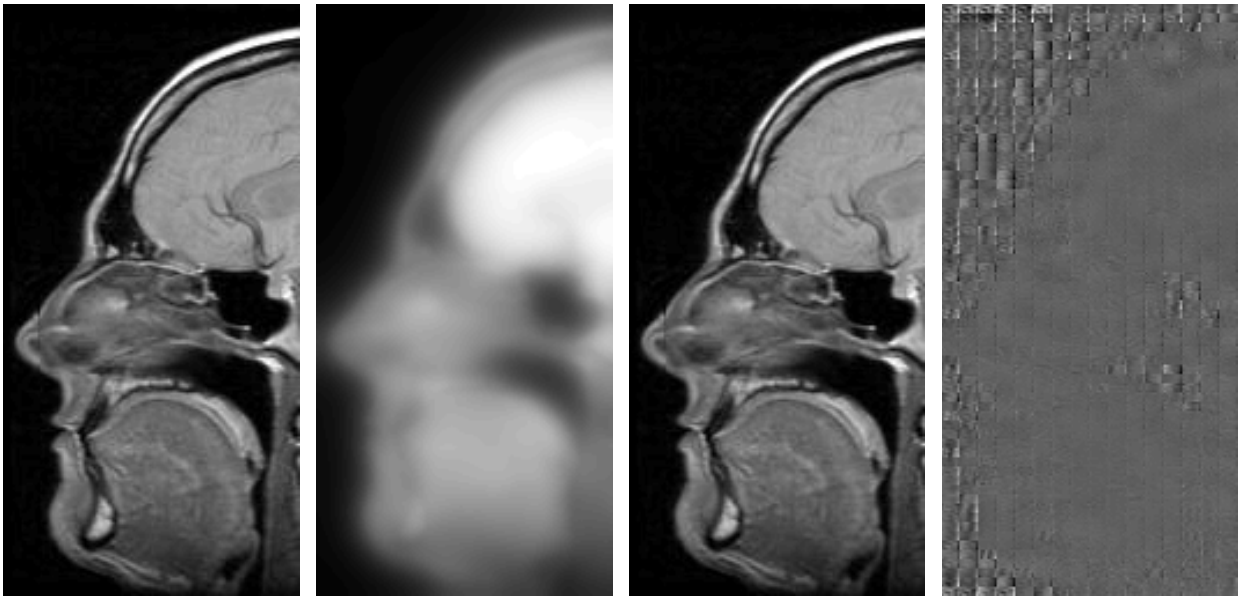

Figure 2: Dictionary learning for MRI image with Gaussian PSF operator. Left to right: true model $\boldsymbol{m}$, data (model after application of J), recovered model using trained dictionary $D_t\boldsymbol{u}$, error $\|\boldsymbol{m} - D_t\boldsymbol{u}\|^2$ (gray scale of the error were rescaled for display purposes)

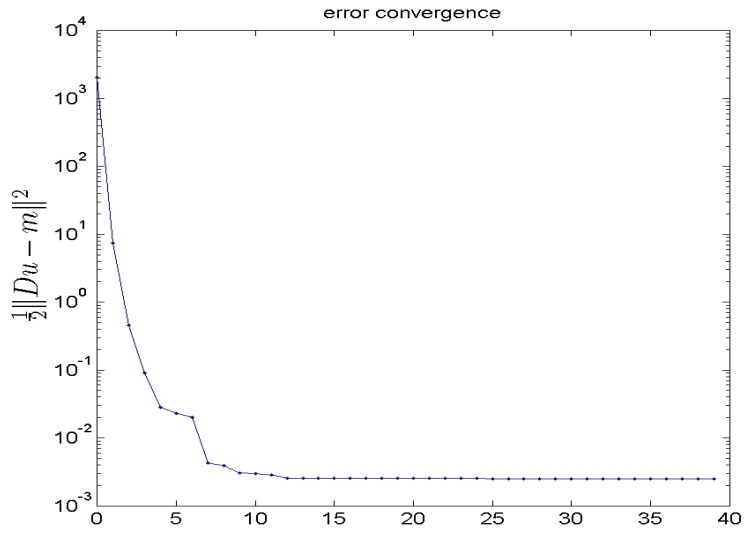

Figure 3: Loss convergence during the dictionary learning process for MRI image with Gaussian PSF operator.

Model	Model Size	J	Data Size	Dictionary Type	Train:Test Sets Size	Training Loss Reduction	Validation Loss Reduction
`Features`	24×24	SR $_{2\times 2}$	12×12	DCT $_{16\times 32}$	1:1	29.1	27.5 %
`Features`	24×24	SR $_{2\times 2}$	12×12	Features $_{16\times 32}$	1:1	3.6	2.3 %

Table 1: Overlapping patches - noisy data recovery performance

using D_0 and D_t accordingly. The average loss L_t and L_v for each of the sets is

$$L_t(D) = \frac{1}{s}\sum_i \|\boldsymbol{m}_i - \widehat{\boldsymbol{m}}_i(D)\|^2$$

$$L_v(D) = \frac{1}{z}\sum_i \|\widetilde{\boldsymbol{m}}_i - \widehat{\widetilde{\boldsymbol{m}}}_i(D)\|^2$$

Obviously, by construction, L_t is minimized for the choice $D = D_t$. We define the loss reduction for the training set as

$$\text{loss reduction} = \frac{L_t(D_0) - L_t(D_t)}{L_t(D_0)}$$

and similarly for the validating set. For the training set loss reduction is smaller than 1. If a similar number is obtained for the validating set then we obtain a reasonable dictionary for the problem. If on the other hand, the loss reduction for the validating test is far from the loss reduction for the training set then we concur that either the training or the validating set fail to represent the problem.

5.3 Overlapping Patches Simulations

Small training and testing data sets, each consisting of two 24×24 images were considered. Each image was generated by a random selection of atoms from a 16×32 feature dictionary. The dimensions of the patches were 4×4, and a single pixel overlap gap was set. A 2×2 averaging (super-resolution) operator, J, was applied over the model to produce the data $\mathbf{d}$. In this setup, a 16×32 overcomplete feature and DCT dictionaries were employed (table 5.3).

5.4 Distinct Patches Simulations - Single Image

Training data sets were generated from the left half of the following models: a 256×256 Siberian tiger cubs image, a 264×504 text art image of an eye, and a 256×256 image of text. A 4×4 averaging operator J was applied over these models in order to produce the data. Similarly, the right halves of these images were used for producing the validation data sets. A 32×128 overcomplete DCT, feature and random dictionaries were trained (table 5.4).

Model	Model Size	J	Data Size	Dictionary Type	Train:Test Sets Size	Training Loss Reduction	Validation Loss Reduction
`Eye`	264×252	SR $_{4\times4}$	66×63	DCT $_{77\times154}$	1:1	1.4 %	9.7 %
`Text`	256×128	SR $_{4\times4}$	64×32	DCT $_{16\times120}$	1:1	7.1 %	6.9 %
`Cubs`	256×128	SR $_{4\times4}$	64×32	DCT $_{32\times128}$	1:1	14.4 %	15.7 %
`Cubs`	256×128	SR $_{4\times4}$	64×32	Features $_{32\times128}$	1:1	32.2 %	31.7 %
`Cubs`	256×128	SR $_{4\times4}$	64×32	Random $_{32\times128}$	1:1	53.1 %	51.4 %

Table 2: Single image - noisy data recovery performance

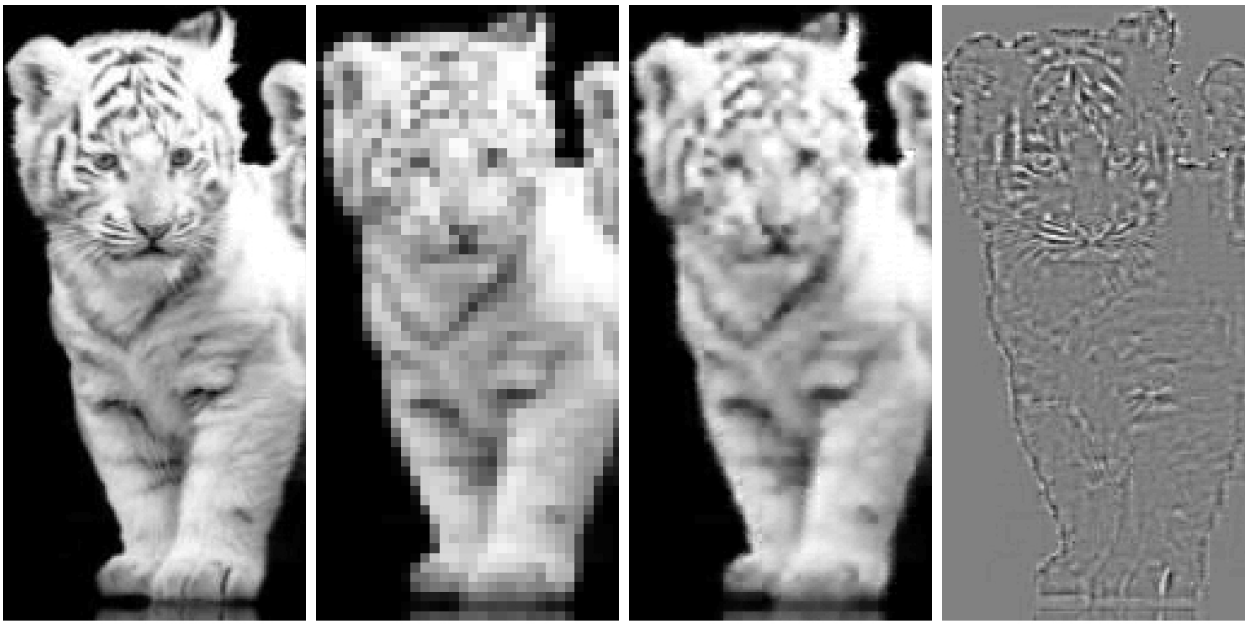

Figure 4: Dictionary learning for Siberian tiger cubs image with a 4×4 averaging operator. Left to right: true model $\boldsymbol{m}$, data $\boldsymbol{d}$ (model after application of J), recovered model $D_t\boldsymbol{u}$, error $\|\boldsymbol{m} - D_t\boldsymbol{u}\|^2$ (gray scale of the error were rescaled for display purposes)

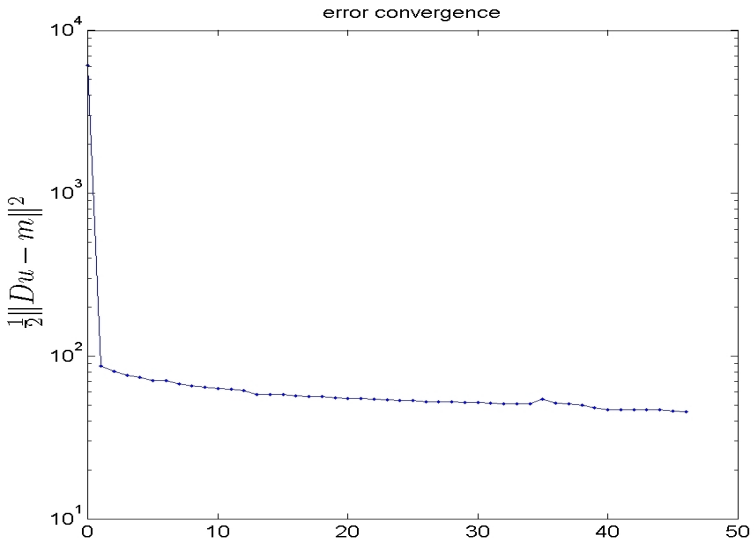

Figure 5: Loss convergence during the dictionary learning process for left half of a Siberian tiger cubs image with a 4×4 averaging operator.

5.5 Distinct Patches Simulations - Multi-Image

The third training set was generated by splitting a 256×256 head MRI scan of a mid-lateral sagittal projection into 64 sub-images of 32×32, out of which subsets of 15 sub-images were randomly chosen. Only sub-images with variance exceeding 20% of the overall mean variance in the training model were considered. This way, sub-images of smooth background which are characterized by poor feature content were excluded from the training set. These sub-images conveyed a portion of 20% of the entire training image. As test data, 17 head MRI slices from lateral sagittal projections of 256×256 were used. Two different operators J were applied over this data set: a 4×4 averaging operator (table 5.5) and a gaussian point spread function operator ([3]) (table 5.5). Dictionary training was performed using 32×128 overcomplete DCT, feature and random prototype dictionaries.

Model	Model Size	J	Data Size	Dictionary Type	Train:Test Sets Size	Training Loss Reduction	Validation Loss Reduction	Validation Variance
MRI	256×256	SR $_{4\times4}$	64×64	DCT $_{32\times128}$	1:85	14.3 %	13.1 %	6.4 %
MRI	256×256	SR $_{4\times4}$	64×64	Features $_{32\times128}$	1:85	34.9 %	32.8 %	2.4 %
MRI	256×256	SR $_{4\times4}$	64×64	Random $_{32\times128}$	1:85	49.3 %	50.4 %	12.5 %
MRI	256×256	PSF $_{128\times128}$	256×256	Features $_{32\times128}$	1:17	47.4 %	47.7 %	2.2 %

Table 3: Multi - image noisy data recovery performance

Model	Model Size	J	Data Size	Dictionary Type	Train:Test Sets Size	Training Loss Reduction	Validation Loss Reduction	Validation Variance
`MRI`	256×256	SR $_{4\times4}$	64×64	DCT $_{32\times128}$	1:17	52.8 %	46.9 %	67.7 %

Table 4: Multi-image noiseless data recovery performance

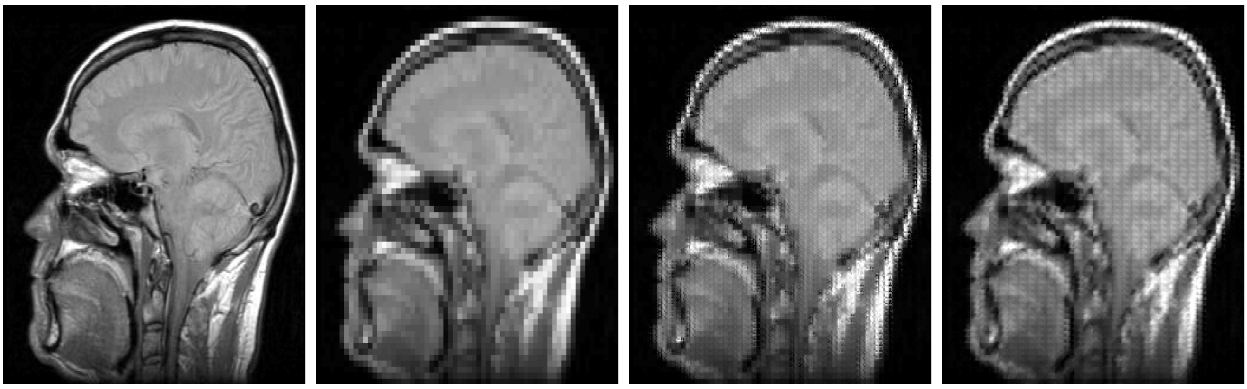

Figure 6: Performance validation for MRI image with averaging operator J. Left to right: true model $\boldsymbol{m}$, data $\boldsymbol{d}$ (model after application of J), recovered model using a prototype dictionary $D_0\boldsymbol{u}(D_0)$, recovered model using the trained dictionary $D_t\boldsymbol{u}(D_t)$

6 Discussion

Regardless of the choice for initial dictionary, the least square errors of images recovered using trained dictionaries were consistently smaller than that of images recovered using the original dictionaries. Moreover, a comparison of the error reduction over the testing data versus the validation data reveals that the acquired trained dictionaries were general enough to provide equivalent results over unseen data.

An important observation is that despite the relatively large percentage improvement in the least square ℓ_2-norm error in using a trained dictionary over a prototype dictionary, such improvement was less apparent when asesed by the appraisal of the eye (sometimes referred to as the "eyeball norm"). This discrepancy can be attributed mainly to the fact that the considered loss measure, i.e. least square ℓ_2-norm, differs substantially from the error measure employed by our vision. There have been many efforts in defining distance measures for mammalian's vision. Due to the complexity of the neuronal system, this challenge still remains unresolved. Nevertheless, on a more qualitative level, it is well known that the visual system in mammalian is more susceptible to changes in context, rather than changes in intensity, contrast or dynamic range. For an instance, our vision may consider two similar images of different gray scale level as almost identical, while images of different content with similar gray-scale would seem much different. Conversely, it is easy to alter the gray scale of an image to generate another, which would provide a greater least square error than the one arising from images of different context. Here, the error in the least square sense was considered and therefore, performance should mainly be judged in that respect. Nevertheless, the methodology proposed here can incorporate any other derivable loss expression.

Different trained dictionaries were obtained from different initial dictionaries. This can be explained by several reasons, first, an optimal dictionary may not be unique, as the recovered images are dictionary permutation invariant. Furthermore, for some data, identical images can be represented by equally sparse representation using different atoms. Observation of the final error figures for images recovered by different trained dictionaries, shows small differences, which may suggest, convergence into multiple minima.

Typically, the dictionary learning phase, i.e. the inverse problem, is far more computationally intensive than the independent phase of parameter recovery (forward problem). However, the former, need to be performed only once for a given set of examples, while the latter, can be facilitated repeatedly for multiple data sets. Accordingly, the learning stage can be conducted offline, and then later, the resulted dictionary can be used multiple times on the offset.

Another issue which was not addressed in this study is preconditioning. For problems of realistic dimensions, where the training set may involve thousands of examples, only implicit preconditioning can be considered. This topic is an active field of research which confers great difficulties and challenges by itself.

One of the cardinal factors that influences performance is the dictionary update rate (sometimes referred to as learning rate), which can be controlled, to some extent, by the regularization hyper-parameter α and also by ϵ. Determining optimal values for these hyper-parameters, can be formulated as an optimization problem independently, or alternatively,

can be recovered in addition to all other recovered parameters. In this study, fixed, predefined α and ϵ were used to avoid further complexity, however, for any specific application a variable and optimized hyper-parameters should be used.

7 Conclusions and Future Challenges

We have introduced a method for designing an overcomplete dictionary for solving parameter estimation inverse problem by means of sparse representation. This framework can be utilized for a broad range of applications such as: multi-modality, compressed sensing, optimal experimental design and inverse source localization.

The implementation introduced here has successfully demonstrated the superiority of trained overcomplete dictionaries in solving parameter estimation problems.

Several future challenges are left to be pursued, such as: exploration of the performance of non-linear sparse models by estimating their performance vs. their computational cost, derivation of a generic method for deduction of an optimal learning rate from the data, and computationally, constructing an effective implicit preconditioner, to be applied both in the learning phase and in the parameter estimation process.

8 Acknowledgements

The authors wish to express their gratitude for Jim Nagy, Michele Benzi and Raya Shindmes for their advice.

This research was supported by NSF grants DMS-0724759, CCF-0427094, CCF-0728877 and by DOE grant DE-FG02-05ER25696.

References

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[2] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, To Appear, 2005.

[3] J. Chung, E. Haber, and J. Nagy. Numerical methods for coupled super-resolution. *Inverse Problems*, 22:1261–1272, 2006.

[4] J. Claerbout and F. Muir. Robust modeling with erratic data. *Geophysics*, 38:826–844, 1973.

[5] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, March 2000.

[6] Michael Elad and Dmitry Datsenko. Example-based regularization deployed to super-resolution reconstruction of a single image. *The Computer Journal (to appear)*, 2006.

[7] Mark Girolami. A variational method for learning sparse and overcomplete representations. *Neural Computation*, 13(11):2517–2532, 2001.

[8] E Haber and U M Ascher. Preconditioned all-at-once methods for large, sparse parameter estimation problems. *Inverse Problems*, 17(6):1847–1864, 2001.

[9] C. Kervrann and J. Boulanger. Optimal spatial adaptation for patch-based image denoising. *IEEE Trans Image Process*, 15(10):2866–78, 2006.

[10] Kenneth Kreutz-Delgado, Joseph F. Murray, Bhaskar D. Rao, Kjersti Engan, Te-Won Lee, and Terrence J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15(2):349–396, February 2003.

[11] Michael S. Lewicki and Terrence J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000.

[12] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer-Verlag, 1999.

[13] Dianne P. O'Leary. Robust regression computation computation using iteratively reweighted least squares. *SIAM J. Matrix Anal. Appl.*, 11(3):466–480, 1990.

[14] B. Olshausen and D. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:33113325, 1997.

[15] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, June 1996.

[16] Alexander Rakhlin. *Applications of empirical processes in learning theory : algorithmic stability and generalization bounds*. PhD thesis, Dept. of Brain and Cognitive Sciences, Massachusetts Institute of Technology, 2006.

[17] Y. Saad. A flexible inner-outer preconditioned gmres algorithm. *SIAM J. Sci. Comput.*, 14:461–469, 1993.

[18] M.D. Sacchi and T.J Ulrych. Improving resolution of radon operators using a model re-weighted least squares procedure. *Journal of Seismic Exploration*, 4:315–328, 1995.

[19] P. Stark. Inference in infinite dimensional inverse problems: discretization and duality. *JGR*, 97:14055–14082, 1992.

[20] James O. Street, Raymond J. Carroll, and David Ruppert. A note on computing robust regression estimates via iteratively reweighted least squares. *The American Statistician*, 42:152–154, 1988.

[21] Michael E. Tipping. Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, 2001.

[22] E. van den Berg and M. P. Friedlander. In pursuit of a root. Technical report, Department of Computer Science, University of British Columbia, June 2007.

[23] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[24] C. Vogel. *Computational methods for inverse problem*. SIAM, Philadelphia, 2001.

[25] K. P. Whittall and D. W. Oldenburg. *Inversion of Magnetotelluric Data for a One Dimensional Conductivity*, volume 5. SEG monograph, 1992.

[26] B.D. Wipf, D.P.; Rao. Sparse bayesian learning for basis selection. *IEEE Transactions on Signal Processing*, 52(8):2153– 2164, 2004.