Sankhyā : The Indian Journal of Statistics
2018, Volume 80-B, Part 1, pp. 178-193

Solution of Linear Ill-Posed Problems Using Random Dictionaries

Pawan Gupta and Marianna Pensky
University of Central Florida, Orlando, USA

Abstract

In the present paper, we consider an application of overcomplete dictionaries to the solution of general ill-posed linear inverse problems. In the context of regression problems, there has been an enormous amount of effort to recover an unknown function using such dictionaries. One of the most popular methods, lasso, and its versions, is based on minimizing the empirical likelihood and unfortunately, requires stringent assumptions on the dictionary, the so-called, compatibility conditions. Though compatibility conditions are hard to satisfy, it is well known that this can be accomplished by using random dictionaries. In the present paper, we show how one can apply random dictionaries to the solution of ill-posed linear inverse problems. We put a theoretical foundation under the suggested methodology and study its performance via simulations and real-data example.

AMS (2000) *subject classification.* Primary 62G05; Secondary 62C10.
Keywords and phrases. Linear inverse problem, Lasso, Random dictionaries

1 Introduction

In this paper, we consider the solution of a general ill-posed linear inverse problem $\mathbf{Qf} = \mathbf{q}$ where $\mathbf{Q}$ is a bounded linear operator that does not have a bounded inverse, and the right-hand side $\mathbf{q}$ is measured with error. In particular, we consider equation

$$\mathbf{y} = \mathbf{q} + \sigma\boldsymbol{\eta}, \quad \mathbf{q} = \mathbf{Qf}, \tag{1.1}$$

where $\mathbf{y}, \mathbf{q}, \mathbf{f}, \boldsymbol{\eta} \in \mathbb{R}^n$, $\mathbf{Q} \in \mathbb{R}^{n\times n}$. Here, $\mathbf{y}$ is observed, $\mathbf{q}$ is unobserved, $\mathbf{f}$ is the function to be estimated, σ is the noise level, and $\boldsymbol{\eta}$ is the noise vector which we assume to have standard normal distribution. Matrix $\mathbf{Q}$ is invertible, but its lowest eigenvalue is very small, especially, when n is relatively large, which makes the problem ill-posed. A general linear inverse problem can usually be reduced to formulation (1.1) by, either expanding $\mathbf{y}$ and $\mathbf{f}$ over some collection of basis functions or by measuring them at some set of points.

Solutions of statistical inverse problem (1.1) usually rely on reduction of the problem to the sequence model by carrying out the singular value decomposition (SVD) (see, e.g., Cavalier et al., 2002, Cavalier and Reiss, 2014, and Tropp and Wright, 2010 and references therein), or its relaxed version, the wavelet-vaguelette decomposition proposed by Donoho (1995) and further studies by Abramovich and Silverman (1998). Another general approach is Galerkin method with subsequent model selection (see, e.g., Cohen et al., 2004).

The advantage of the methodologies listed above is that they are asymptotically optimal in a minimax sense. The function of interest is usually represented via an orthonormal basis which is motivated by the form of matrix $\mathbf{Q}$. However, in spite of being minimax optimal in many contexts, these approaches have drawbacks. In particular, in practical applications, the number of observations n may be low while noise level σ high. In this situation, if the unknown vector $\mathbf{f}$ does not have a relatively compact and accurate representation in the chosen basis, the precision of the resulting estimator will be poor.

In the last decade, a great deal of effort was spent on the recovery of an unknown vector $\mathbf{f}$ in regression setting from its noisy observations using overcomplete dictionaries. In particular, if $\mathbf{f}$ has a sparse representation in some dictionary, a collection of vectors used for the representation of $\mathbf{f}$, then $\mathbf{f}$ can be recovered with a much better precision than, for example, when it is expanded over an orthonormal basis. The methodology is based on the idea that the error of an estimator of $\mathbf{f}$ is approximately proportional to the number of dictionary functions that are used for representing $\mathbf{f}$, therefore, expanding a function of interest over fewer dictionary elements decreases the estimation error. Similar advantages hold in the case of linear inverse problems (see Pensky, 2016). However, in order to represent a variety of functions using a small number of dictionary elements, one needs to consider a dictionary of much larger size than the number of available observations, the, so-called, overcomplete dictionary.

A variety of techniques have been developed for the solution of regression problems using overcomplete dictionaries such as likelihood penalization methods and greedy algorithms. The most popular of those methods (due to its computational convenience), lasso and its versions, have been used for the solution of a number of theoretical and applied statistical problems (see, e.g., Cavalier et al., 2009, and also Tropp and Wright, 2011 and references therein). However, application of lasso is based on maximizing the likelihood and, unfortunately, relies on stringent assumptions on the dictionary $\{\varphi_k\}_{k=1}^{p}$, the, so-called, compatibility conditions, for a proof of its optimality. In regression set up, as long as compatibility conditions hold, lasso identifies a linear combination of the dictionary elements which represent the function of interest best of all at a "price" which is proportional to $\sigma \sqrt{n^{-1} \log p}$ where log stands for the natural logarithm and p is the dictionary size (see, e.g., Bühlmann and van de Geer, 2011). Regrettably, while compatibility conditions may be satisfied for the vectors $\boldsymbol{\varphi}_j$ in the original dictionary, they usually do not hold for their images $\mathbf{Q}\boldsymbol{\varphi}_j$ due to contraction imposed by the operator $\mathbf{Q}$. Pensky (2016) showed how lasso solution can be modified, so that it delivers an optimal solution, however, compatibility assumptions in Pensky (2016) remain very complex and hard to verify.

In the recent years it has been discovered that in the regression setting, one can satisfy compatibility conditions for lasso by simply using random dictionaries. In particular, Vershynin (2012) provided a variety of way for construction of such dictionaries, i.e., dictionaries comprised of random vectors. The purpose of Vershynin (2012), however, is that the methodology is intended for the recovery of a function which is directly observed. The purpose of the present paper is to explain how random dictionaries can be adopted for the solution of ill-posed linear inverse problems.

To the best of our knowledge, application of random dictionaries to signal recovery has not been used since random vectors usually contain "pure noise" and therefore are perceived as unsuitable for representing a meaningful signal. This is indeed true when one needs to estimate a simple smooth function which is best represented by a smooth set of basis functions. However, when a signal has a more complicated structure, it may not be expanded over a small number of basis functions. In this case, a large rich dictionary may be helpful since there is a high chance that the signal of interest has is close to a linear combination of a small number of vectors of the dictionary. The advantage of the random dictionaries is that unlike in the case of a fixed dictionary, one can work with a dictionary of very large size which provides a competitive advantage over standard orthogonal basis based techniques. This advantage may be more pronounced when one needs to solve an inverse ill-posed problem since, as it was noted in Pensky (2016), finding a "low-cost" representation of a function of interest can significantly improve the accuracy of the solution.

The rest of the paper is organized as follows. Section 2 introduces some notations, formulates optimization problem with lasso penalty and lists compatibility conditions of Pensky (2016). Section 3 contains the main results of the paper: it explains how one can obtain fast lasso convergence rates by using random dictionaries. Section 4 includes a simulation study and a real data example which prove that our technique is competitive. Section 5 concludes the paper with the discussion. Finally, Section 5 provides the proofs of the statements in the paper.

2 Construction of the Lasso Estimator and a General Compatibility Condition

In the paper, we use the following notations.

For any vector $\mathbf{t} \in \mathbb{R}^p$, denote its ℓ_2, ℓ_1, ℓ_0 and ℓ_∞ norms by, respectively, $\|\mathbf{t}\|$, $\|\mathbf{t}\|_1$, $\|\mathbf{t}\|_0$ and $\|\mathbf{t}\|_\infty$. For any matrix $\boldsymbol{A}$, denote its i^{th} row and j^{th} column by, $\boldsymbol{A}_{i\cdot}$ and $\boldsymbol{A}_{\cdot j}$ respectively. Denote its spectral and Frobenius norms by, respectively, $\|\boldsymbol{A}\|$ and $\|\boldsymbol{A}\|_2$.

Denote $\mathcal{P} = \{1, \cdots, p\}$. For any subset of indices $J \subseteq \mathcal{P}$, subset J^c is its complement in $\mathcal{P}$ and $|J|$ is its cardinality, so that $|\mathcal{P}| = p$. Let $\mathcal{L}_J = \text{Span}\left\{\boldsymbol{\varphi}_j,\ j \in J\right\}$. If $J \subset \mathcal{P}$ and $\mathbf{t} \in \mathbb{R}^p$, then $\mathbf{t}_J \in \mathbb{R}^{|J|}$ denotes reduction of vector $\mathbf{t}$ to subset of the indices J. Also, $\boldsymbol{\Phi}_J$ denotes the reduction of matrix $\boldsymbol{\Phi}$ to columns $\boldsymbol{\Phi}_{\cdot j}$ with $j \in J$.

Denote by $\lambda_{\min}(m; \boldsymbol{\Phi})$ and $\lambda_{\max}(m; \boldsymbol{\Phi})$ the minimum and the maximum restricted eigenvalues of matrix $\boldsymbol{\Phi}^T\boldsymbol{\Phi}$ given by

$$\lambda_{\min}(m; \boldsymbol{\Phi}) = \min_{\substack{\mathbf{t} \in \mathbb{R}^p \\ \|\mathbf{t}\|_0 \leq m}} \frac{\mathbf{t}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{t}}{\|\mathbf{t}\|_2^2}, \quad \lambda_{\max}(m; \boldsymbol{\Phi}) = \max_{\substack{\mathbf{t} \in \mathbb{R}^p \\ \|\mathbf{t}\|_0 \leq m}} \frac{\mathbf{t}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{t}}{\|\mathbf{t}\|_2^2}. \tag{2.1}$$

Denote by $\boldsymbol{\Phi}$ the dictionary matrix with columns $\boldsymbol{\varphi}_j \in \mathbb{R}^n$, $j = 1, \cdots, p$, where p is possibly much larger than n and

$$\mathbf{f}_\mathbf{t} = \sum_{j=1}^p t_j \boldsymbol{\varphi}_j = \boldsymbol{\Phi}\mathbf{t}. \tag{2.2}$$

Let $\boldsymbol{\theta}$ be the true vector of coefficients of expansion of $\mathbf{f}$ over the dictionary $\boldsymbol{\Phi}$ so that $\mathbf{f} = \boldsymbol{\Phi}\boldsymbol{\theta}$. Let vectors $\boldsymbol{\psi}_j$ be such that $\mathbf{Q}^T\boldsymbol{\psi}_j = \boldsymbol{\varphi}_j$, where $\mathbf{Q}^T$ is the transpose of matrix $\mathbf{Q}$, and $\boldsymbol{\Psi}$ be a matrix with columns $\boldsymbol{\psi}_j$, $j = 1, \cdots, p$. Then,

$$\mathbf{Q}^T\boldsymbol{\Psi} = \boldsymbol{\Phi} \quad \text{and} \quad \boldsymbol{\Psi} = \mathbf{Q}(\mathbf{Q}^T\mathbf{Q})^{-1}\boldsymbol{\Phi}. \tag{2.3}$$

Note that, although $\mathbf{f}$ is unknown,

$$\|\mathbf{f} - \mathbf{f}_\mathbf{t}\|^2 = \|\mathbf{f}\|_2^2 + \mathbf{t}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{t} - 2\mathbf{t}^T\boldsymbol{\Phi}^T\mathbf{f} = \|\mathbf{f}\|_2^2 + \mathbf{t}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{t} - 2\mathbf{t}^T\boldsymbol{\Psi}^T\mathbf{Q}\mathbf{f} \tag{2.4}$$

is the sum of the three components where the first one is independent of $\mathbf{t}$, the second one is completely known, while the last term is of the form $2\mathbf{t}^T\boldsymbol{\Psi}^T\mathbf{Q}\mathbf{f} = 2\mathbf{t}^T\boldsymbol{\Psi}^T\mathbf{q}$ and, hence, can be estimated by $2\mathbf{t}^T\boldsymbol{\Psi}^T\mathbf{y}$. Let $\mathbf{z}$ be such that

$$\boldsymbol{\Psi}^T\mathbf{y} = \boldsymbol{\Phi}^T\mathbf{z}.$$

Therefore, expression $\mathbf{t}^T\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{t} - 2\mathbf{t}^T\mathbf{\Psi}^T\mathbf{y}$ is minimized by the same vector $\mathbf{t}$ that minimizes $\|\mathbf{\Phi}\mathbf{t} - \mathbf{z}\|_2^2$ where

$$\mathbf{z} = (\mathbf{\Phi}\mathbf{\Phi}^T)^{-1}\mathbf{\Phi}\mathbf{\Psi}^T\mathbf{y}. \tag{2.5}$$

Denote $\nu_j = \|\boldsymbol{\psi}_j\|_2$, $j = 1, \cdots, p$, and observe that ν_j is proportional to the standard deviation of the jth component of the vector $\mathbf{\Psi}^T\mathbf{y}$. The values ν_j can be viewed as a "cost" of using a dictionary element $\boldsymbol{\varphi}_j$ in the representation of $\mathbf{f}$. Consider a matrix

$$\mathbf{\Upsilon} = \mathrm{diag}(\nu_1, \cdots, \nu_p) = \mathrm{diag}(\|\boldsymbol{\psi}_1\|_2, \cdots, \|\boldsymbol{\psi}_p\|_2). \tag{2.6}$$

Following Pensky (2016), we estimate the true vector of coefficients $\boldsymbol{\theta}$ as a solution of the quadratic optimization problem with the weighted lasso penalty

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\mathbf{t}} \left\{ \|\mathbf{\Phi}\mathbf{t} - \mathbf{z}\|_2^2 + \alpha\|\mathbf{\Upsilon}\mathbf{t}\|_1 \right\}. \tag{2.7}$$

Here $\mathbf{z}$ is given by (2.5) and $\alpha \geq \alpha_0$ where

$$\alpha_0 = \sigma\,\sqrt{2\,n^{-1}\,(\tau+1)\log p}. \tag{2.8}$$

Parameter $\tau > 0$ is related to the required probability bound (see formula (A.3) in Section 5 for details). Subsequently, we estimate the unknown solution $\mathbf{f}$ by $\widehat{\mathbf{f}} = \mathbf{\Phi}\widehat{\boldsymbol{\theta}}$.

Note that since we are interested in $\mathbf{f} = \mathbf{f}_{\boldsymbol{\theta}} = \mathbf{\Phi}\boldsymbol{\theta}$ rather than in the vector $\boldsymbol{\theta}$ of coefficients themselves, we are using lasso for the solution of the so-called prediction problem where it requires milder conditions on the dictionary. In fact, it is known (see Pensky, 2016) that with no additional assumptions, for $\alpha \geq \alpha_0$, with probability at least $1 - 2p^{-\tau}$, one has

$$n^{-1}\,\|\mathbf{f}_{\widehat{\boldsymbol{\theta}}} - \mathbf{f}\|_2^2 \leq \inf_{\mathbf{t}} \left[n^{-1}\,\|\mathbf{f}_{\mathbf{t}} - \mathbf{f}\|_2^2 + 4\alpha\sum_{j=1}^{p}\nu_j|t_j| \right]. \tag{2.9}$$

It is easy to see that if $\mathbf{t} = \boldsymbol{\theta}$, then $\mathbf{f}_{\mathbf{t}} = \mathbf{f}$. Then, with high probability, the error of the estimator $\mathbf{f}_{\widehat{\boldsymbol{\theta}}}$ is proportional to $\sigma\sqrt{n^{-1}\,(\tau+1)\log p}\;\sum_j \nu_j$. This is the, so called, *slow lasso rate*. In order to attain the *fast lasso rate* proportional to $\sigma^2 n^{-1}\sum_j \nu_j^2$, one needs some kind of compatibility assumption.

Pensky (2016) formulated the following compatibility condition: matrices $\mathbf{\Phi}$ and $\mathbf{\Upsilon}$ are such that for some $\mu > 1$ and any $J \subset \mathcal{P}$

$$\kappa^2(\mu, J) = \min\left\{\mathbf{d} \in \mathcal{J}(\mu, J),\ \|\mathbf{d}\|_2 \neq 0 : \quad \frac{\mathbf{d}^T\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{d}\cdot\mathrm{Tr}(\mathbf{\Upsilon}_J^2)}{\|(\mathbf{\Upsilon}\mathbf{d})_J\|_1^2}\right\} > 0, \tag{2.10}$$

where $\mathcal{J}(\mu, J) = \left\{\mathbf{d} \in \mathbb{R}^p:\ \|(\boldsymbol{\Upsilon}\mathbf{d})_{J_*^c}\|_1 \le \mu \|(\boldsymbol{\Upsilon}\mathbf{d})_J\|_1\right\}$. Pensky (2016) proved that, under assumption (2.10), for $\alpha = \varpi\alpha_0$ where $\varpi \ge (\mu+1)/(\mu-1)$ and α_0 is defined in (2.8), with probability at least $1 - 2p^{-\tau}$, one has

$$\|f_{\widehat{\boldsymbol{\theta}}} - f\|_2^2 \le \inf_{J \subseteq \mathcal{P}} \left[\|f - f_{\mathcal{L}_J}\|_2^2 + \frac{\sigma^2 K_0 (1+\varpi)^2 (\tau+1)}{\kappa^2(\mu, J)} \frac{\log p}{n} \sum_{j \in J} \nu_j^2 \right], \quad (2.11)$$

where $f_{\mathcal{L}_J} = \mathrm{proj}_{\mathcal{L}_J} f$.

Note, however, that unless matrix $\boldsymbol{\Phi}$ has orthonormal columns, assumption (2.10) is hard not only to satisfy but even to verify since it requires checking it for every subset $J \in \mathcal{P}$. Indeed, sufficient conditions listed in Appendix A1 of Pensky (2016) rely on the results of Bickel et al. (2009) and require very stringent conditions on $\lambda_{\min}(m; \boldsymbol{\Phi})$ and entries $\boldsymbol{\Upsilon}$ in (2.6). In the present paper, we offer an alternative to this approach.

3 Lasso Solution to Linear Inverse Problems Using Random Dictionaries

An advantage of using random dictionary lies in the fact that one can ensure, with a high probability, that the dictionary satisfies a restricted isometry condition (see, e.g., Candès et al., 2010 or Foucart and Rauhut, 2013). In particular, if matrix $\boldsymbol{\Phi} \in \mathbb{R}^{n \times p}$ satisfies the restricted isometry property of order $s \ge 1$, then $\lambda_{\min}(s; \boldsymbol{\Phi}) > 0$. The latter allows one to formulate the following results.

THEOREM 1. *Let* $\boldsymbol{\theta}$ *be the solution of the optimization problem* (2.7) *with* $\alpha \ge \alpha_0$ *where* α_0 *is defined in* (2.8). *Let* $\boldsymbol{\Phi} \in \mathbb{R}^{n \times p}$ *be a random dictionary independent of* $\mathbf{y}$ *in* (1.1). *Denote*

$$J_* = \arg\min \left\{ J \subset \mathcal{P}:\ n^{-1} \|\mathbf{f} - \mathbf{f}_{\mathcal{L}_J}\|_2^2 + K_0 \alpha^2 \sum_{j \in J} \nu_j^2 \right\}, \quad (3.1)$$

where $\mathbf{f}_{\mathcal{L}_J} = \mathrm{proj}_{\mathcal{L}_J} \mathbf{f}$ *and assume that* $\boldsymbol{\Phi}$ *is such that for some* s, $1 \le s \le n/2$ *and* $\delta, \epsilon_1, \epsilon_2, \epsilon_3 \in (0,1)$, *the following conditions hold*

$$\mathbb{P}\left(\lambda_{\min}(2s; \boldsymbol{\Phi}) \ge 1 - \delta\right) \ \ge\ 1 - \epsilon_1, \quad (3.2)$$

$$\mathbb{P}\left(|J_*| \le s\right) \ \ge\ 1 - \epsilon_2, \quad (3.3)$$

$$\mathbb{P}\left(\|\widehat{\boldsymbol{\theta}}\|_0 \le s\right) \ \ge\ 1 - \epsilon_3, \quad (3.4)$$

If $K_0 \geq 4/(1-\delta)^2$ *in* (3.1), *then*

$$\mathbb{P}\left(\frac{1}{n}\|\mathbf{f}_{\widehat{\boldsymbol{\theta}}}-\mathbf{f}\|_2^2 \leq \inf_{J\subseteq\mathcal{P}}\left[\frac{1}{n}\|\mathbf{f}-\mathbf{f}_{\mathcal{L}_J}\|_2^2+K_0\alpha^2\sum_{j\in J}\nu_j^2\right]\right) \geq 1-2p^{-\tau}-\epsilon_1-\epsilon_2-\epsilon_3. \tag{3.5}$$

Note that for $\alpha = \alpha_0$ and $K_0 = 4/(1-\delta)^2$, under assumptions (3.2)–(3.4), (3.5) yields the following result

$$\mathbb{P}\left(\frac{1}{n}\|\mathbf{f}_{\widehat{\boldsymbol{\theta}}}-\mathbf{f}\|_2^2 \leq \inf_{J\subseteq\mathcal{P}}\left\{\frac{1}{n}\|\mathbf{f}-\mathbf{f}_{\mathcal{L}_J}\|_2^2+\frac{4\,\sigma^2}{n\,(1-\delta)^2}\sum_{j\in J}\nu_j^2\right\}\right) \geq 1-2p^{-\tau}-\epsilon_1-\epsilon_2-\epsilon_3. \tag{3.6}$$

As Lemma 1 below shows, assumption (3.2) can be guaranteed by choosing a dictionary of a particular type.

Lemma 1. *Let matrix* $\boldsymbol{\Phi} \in \mathbb{R}^{n\times p}$ *be independent of* $\mathbf{y}$ *and satisfy one of the following conditions:*
a) Matrix $\boldsymbol{\Phi}$ *has independent sub-gaussian isotropic random rows;*
b) Matrix $\boldsymbol{\Phi}$ *has independent sub-gaussian isotropic random columns with unit norms;*
c) Matrix $\boldsymbol{\Phi}$ *is obtained as* $\boldsymbol{\Phi} = (c\sqrt{n})^{-1}\mathbf{D}\mathbf{W}$ *where* $\mathbf{W} \in \mathbb{R}^{m\times p}$ *is a matrix with i.i.d. standard Gaussian entries and columns of the matrix* $\mathbf{D} \in \mathbb{R}^{n\times m}$ *form a non-random c-tight frame so that for any vector* $\boldsymbol{x}$, *one has* $\boldsymbol{x}^T\mathbf{D}\mathbf{D}^T\boldsymbol{x} = c^2\|\boldsymbol{x}\|^2$.

If, for some $\delta \in (0,1)$ *and* $1 \leq s \leq n/2$, *one has*

$$n \geq C_1\delta^{-2}s[\log(p/s)+1], \tag{3.7}$$

then condition (3.2) *holds with* $\epsilon_1 \leq 2\exp(-C_2\delta^2 n)$. *Here,* C_1 *and* C_2 *depend on the kind of sub-gaussian variables that are involved in the formation of* $\boldsymbol{\Phi}$ *and are independent of* n, m, p, s, *and* δ.

Finally, conditions (3.3) and (3.4) can be ensured by restricting the set of solutions $\mathbf{t}$ to vectors with cardinality at most s. In this case, $\epsilon_2 = \epsilon_3 = 0$ and the following corollary of Theorem 1 is valid.

Corollary 1. *Let* $\boldsymbol{\theta}$ *be the solution of optimization problem*

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\mathbf{t}:\|\mathbf{t}\|_0\leq s}\left\{\|\boldsymbol{\Phi}\mathbf{t}-\mathbf{z}\|_2^2+\alpha\|\boldsymbol{\Upsilon}\mathbf{t}\|_1\right\}, \tag{3.8}$$

with $\alpha \geq \alpha_0$ *where* α_0 *is defined in* (2.8)*. Let* $\mathbf{\Phi} \in \mathbb{R}^{n\times p}$ *be one of the random dictionaries defined in Lemma 1. If, for some* $\delta \in (0,1)$*, condition* (3.7) *holds, then*

$$\mathbb{P}\left(\frac{1}{n}\|\mathbf{f}_{\widehat{\boldsymbol{\theta}}}-\mathbf{f}\|_2^2 \leq \inf_{\substack{J\subseteq\mathcal{P}\\|J|\leq s}}\left[\frac{1}{n}\|\mathbf{f}-\mathbf{f}_{\mathcal{L}_J}\|_2^2+\frac{4\alpha^2}{(1-\delta)^2}\sum_{j\in J}\nu_j^2\right]\right)\geq 1-2p^{-\tau}-2\exp(-C_2\delta^2 n), \tag{3.9}$$

where C_2 *depends on the kind of sub-gaussian variables that are involved in the formation of* $\mathbf{\Phi}$ *and is independent of* n*,* m*,* p*,* s*, and* δ*.*

Note that case c) above offers a structured random dictionary since each of its elements is a linear combination of smooth functions.

4 Simulation Studies and Real-Data Example

In order to evaluate the performance of the procedure suggested in this paper, we carried out a limited simulation study followed by a real-life example.

For our simulation study, we chose three sample sizes: $n = 32$, $n = 64$ and $n = 128$. We first generated a true vector $\mathbf{f}$ using `MakeSignal` program in the package Wavelab 850. We then generated the matrix $\mathbf{Q}$ in Eq. 1.1 as $\mathbf{Q} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ where $\mathbf{U}$ is an $(n \times n)$ random orthogonal matrix and $\mathbf{\Lambda}$ is a diagonal matrix with entries $\mathbf{\Lambda}_{ii} = 1/\sqrt{i}$, $i = 1, 2, \cdots, n$. Using $\mathbf{Q}$, we obtained the unobserved vector $\mathbf{q}$ as

$$\mathbf{q} = \mathbf{Q}\mathbf{f}.$$

At last, for generating the data $\mathbf{y}$, we added Gaussian random noise to $\mathbf{q}$. For this purpose, we chose particular values of the Signal to Noise Ratio (SNR) and obtained σ as the ratio of the standard deviation of $\mathbf{q}$ and the SNR. Vector $\mathbf{y}$ was then calculated at n observation points as $\mathbf{y} = \mathbf{q} + \sigma\,\boldsymbol{\eta}$ where $\boldsymbol{\eta} \in \mathbb{R}^n$ is a standard normal vector. Finally, we ran simulations for two noise levels: SNR = 3 and SNR = 5.

We compared the estimators of $\mathbf{f}$ based on random dictionaries with the estimator of $\mathbf{f}$ based on the Singular Value Decomposition (SVD). For our simulations we have created three different $n \times p$ random dictionaries with $p = 5000$: (a) purely random dictionaries with, respectively, the i.i.d. standard Gaussian entries and the i.i.d. sparse Bernoulli entries; (b) the fusion of the fixed dictionary and the random dictionary that follows case c) in Lemma 1 with $\mathbf{D}$ being the Haar dictionary. The sparse Bernoulli variable is defined as

$$\mathbf{X} = \begin{cases} -\sqrt{\frac{3}{n}} & \text{with probability } \frac{1}{6} \\ 0 & \text{with probability } \frac{2}{3} \\ \sqrt{\frac{3}{n}} & \text{with probability } \frac{1}{6}. \end{cases} \tag{4.1}$$

For creating the fusion dictionary, we first generated the orthogonal matrix of Haar wavelet transform $\mathbf{D}$ using `MakeWavelet` function, so that $m = n$ and $c = 1$. Then we obtained the dictionary $\mathbf{\Phi}$ following part c) of the Lemma 1 using the $n \times p$ matrix $\mathbf{W}$ with the i.i.d. normal entries.

We obtained matrix $\mathbf{\Psi}$ of the inverse images as the numerical solution of the exact equation $\mathbf{Q}^T\mathbf{\Psi} = \mathbf{\Phi}$ and calculated vector $\mathbf{z}$ with elements (2.5). For the sake of obtaining a solution of the optimization problem (2.7), we used function `LassoWeighted` in SPAMS MatLab toolbox (see Mairal, 2014).

In order to evaluate the value of the lasso parameter α, we calculated $\alpha_{\max}$ as the value of α that guarantees that all coefficients in the model vanish. We created a grid of the values of $\alpha_k = \alpha_{\max} * k/N$, $k = 1, \cdots, N$, with $N = 200$. As a result, we obtained a collection of estimators $\widehat{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}(\alpha_k)$. For the purpose of choosing the most appropriate value of k, we estimated α as $\hat{\alpha} = \alpha_{\widehat{k}}$ in two ways: one using the oracle value of α and another using the estimated value of α. We found oracle value of α as $\alpha_{oracle} = \alpha_{\max} * \widehat{k}_{oracle}/N$ using the value $\widehat{k}_{oracle}$ that guarantees the most accurate estimator of $\mathbf{f}$:

$$\widehat{k}_{oracle} = \arg\min_k \|\mathbf{f} - \mathbf{\Phi}\widehat{\boldsymbol{\theta}}(\alpha_k)\|_2.$$

Since the vector $\mathbf{f}$ is unavailable in real-life, we find the estimated value $\hat{\alpha}_{est} = \alpha_{\max} * \widehat{k}_{est}/N$ of α using

$$\widehat{k}_{est} = \arg\min_k \left[\frac{1}{n}\|\mathbf{y} - \hat{\mathbf{q}}(\alpha_k)\|_2^2 + 2\sigma^2 n^{-1}\hat{p}_k\right],$$

where $\hat{\mathbf{q}}(\alpha_k) = \mathbf{Q}\mathbf{\Phi}\widehat{\boldsymbol{\theta}}(\alpha_k)$ is the estimator of $\mathbf{q}$ based on the lasso estimator obtained with the parameter α_k and $\hat{p}_k$ is the number of nonzero components of $\widehat{\boldsymbol{\theta}}(\alpha_k)$.

We compared the estimators $\hat{\mathbf{f}}_{RN}$, $\hat{\mathbf{f}}_{RB}$, $\hat{\mathbf{f}}_{RH}$ of $\mathbf{f}$ based, respectively, on Gaussian, Bernoulli and Haar fusion random dictionaries described above with $\hat{\mathbf{f}}_{SVD}$, the estimator based on the singular value decomposition (SVD). Initially, we considered wavelet estimator of $\mathbf{f}$ using Daubechies wavelet of order 8, but we discarded it due to its poor performance with respect to the estimators considered for comparison. For finding $\hat{\mathbf{f}}_{SVD}$, we used the oracle number K_{oracle} of eigenbasis functions. We obtained K_{oracle} as the number of eigenbasis functions that minimizes the difference between $\hat{\mathbf{f}}_{SVD}$ and the true function $\mathbf{f}$ which is unavailable in a real-life setting.

Table 1 below compares the accuracies of the estimators based on random dictionaries with the SVD estimator. Precision of an estimator $\hat{\mathbf{f}}$ is measured by the relative error

$$R(\hat{\mathbf{f}}) = \|\hat{\mathbf{f}} - \mathbf{f}\|/\|\mathbf{f}\|, \tag{4.2}$$

averaged over 50 simulation runs (with the standard deviations listed in parentheses). For all the three estimators based on random dictionaries, we report the errors with both the oracle and the estimated values of α, $\hat{\mathbf{f}}_{RN}^{oracle}$, $\hat{\mathbf{f}}_{RB}^{oracle}$, $\hat{\mathbf{f}}_{RH}^{oracle}$ and $\hat{\mathbf{f}}_{RN}^{est}$, $\hat{\mathbf{f}}_{RB}^{est}$, $\hat{\mathbf{f}}_{RH}^{est}$, respectively. We carried out simulations with three types of test functions `WernerSorrows`, `MishMash` and `Chirps`. The test signals are presented in Fig. 1.

Table 1 shows that the random dictionary based estimators are more accurate than the SVD estimator with the 5-10% smaller average errors

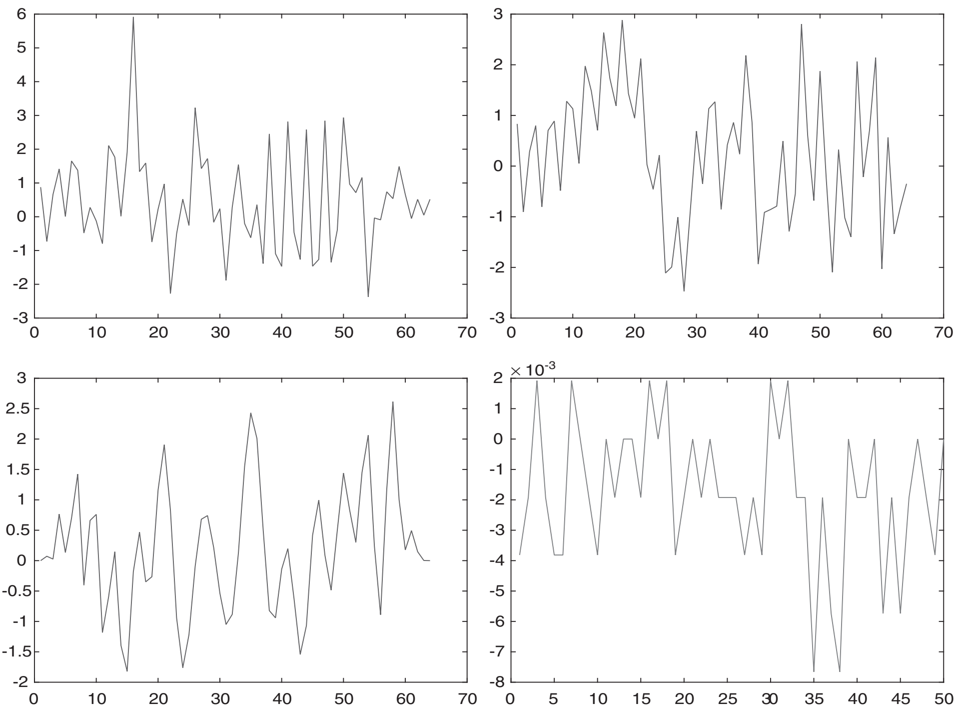

Figure 1: Test signals: `WernerSorrows` (top left), `MishMash` (top right), `Chirps` (bottom left) with $n = 64$ and `Bird's twitter` (bottom right) with $n = 50$

Table 1: The average values of the errors $R(\hat{\mathbf{f}})$ evaluated over 50 simulation runs of the estimators for various test signals (standard deviations of the errors are listed in the parentheses)

	$\hat{\mathbf{f}}_{RN}^{oracle}$	$\hat{\mathbf{f}}_{RN}^{est}$	$\hat{\mathbf{f}}_{RB}^{oracle}$	$\hat{\mathbf{f}}_{RB}^{est}$	$\hat{\mathbf{f}}_{RH}^{oracle}$	$\hat{\mathbf{f}}_{RH}^{est}$	$\hat{\mathbf{f}}_{SVD}$
$WernerSorrows$							
$n = 32,$	0.3416	0.3567	0.3416	0.3566	0.3365	0.3466	0.3587
$SNR = 3$	(0.0497)	(0.0534)	(0.0512)	(0.0559)	(0.0504)	(0.0498)	(0.0502)
$n = 32,$	0.2670	0.2752	0.2651	0.2759	0.2645	0.2725	0.2797
$SNR = 5$	(0.0447)	(0.0440)	(0.0455)	(0.0443)	(0.0433)	(0.0439)	(0.0440)
$n = 64,$	0.3778	0.3944	0.3814	0.3969	0.3739	0.3850	0.3970
$SNR = 3$	(0.0332)	(0.0351)	(0.0308)	(0.0349)	(0.0299)	(0.0331)	(0.0375)
$n = 64,$	0.2047	0.2083	0.2056	0.2084	0.2052	0.2077	0.2113
$SNR = 5$	(0.0228)	(0.0227)	(0.0234)	(0.0236)	(0.0227)	(0.0233)	(0.0233)
$n = 128,$	0.4066	0.4285	0.4079	0.4292	0.4054	0.4164	0.4388
$SNR = 3$	(0.0287)	(0.0304)	(0.0278)	(0.0312)	(0.0277)	(0.0294)	(0.0311)
$n = 128,$	0.2632	0.2687	0.2623	0.2682	0.2623	0.2651	0.2717
$SNR = 5$	(0.0177)	(0.0188)	(0.0172)	(0.0189)	(0.0176)	(0.0183)	(0.0191)
$MishMash$							
$n = 32,$	0.3934	0.4093	0.3951	0.4096	0.3932	0.4045	0.4347
$SNR = 3$	(0.0590)	(0.0634)	(0.0582)	(0.0621)	(0.0617)	(0.0615)	(0.0636)
$n = 32,$	0.2562	0.2615	0.2573	0.2628	0.2569	0.2613	0.2660
$SNR = 5$	(0.0428)	(0.0420)	(0.0408)	(0.0412)	(0.0403)	(0.0407)	(0.0399)
$n = 64,$	0.4621	0.4824	0.4642	0.4817	0.4632	0.4788	0.4974
$SNR = 3$	(0.0392)	(0.0465)	(0.0416)	(0.0448)	(0.0408)	(0.0457)	(0.0468)
$n = 64,$	0.2291	0.2326	0.2291	0.2324	0.2279	0.2312	0.2342
$SNR = 5$	(0.0225)	(0.0237)	(0.0213)	(0.0226)	(0.0221)	(0.0222)	(0.0224)
$n = 128,$	0.4042	0.4152	0.4037	0.4147	0.4040	0.4099	0.4277
$SNR = 3$	(0.0311)	(0.0323)	(0.0301)	(0.0326)	(0.0311)	(0.0318)	(0.0328)
$n = 128,$	0.2676	0.2712	0.2672	0.2711	0.2675	0.2696	0.2735
$SNR = 5$	(0.0193)	(0.0190)	(0.0191)	(0.0191)	(0.0187)	(0.0186)	(0.0194)
$Chirps$							
$n = 32,$	0.3497	0.3603	0.3513	0.3603	0.3502	0.3613	0.3746
$SNR = 3$	(0.0512)	(0.0525)	(0.0490)	(0.0486)	(0.0498)	(0.0507)	(0.0494)
$n = 32,$	0.2336	0.2421	0.2342	0.2407	0.2335	0.2400	0.2454
$SNR = 5$	(0.0375)	(0.0373)	(0.0364)	(0.0364)	(0.0372)	(0.0361)	(0.0360)
$n = 64,$	0.3932	0.4068	0.3935	0.4072	0.3938	0.4017	0.4246

Table 1: (continued)

	$\hat{\mathbf{f}}_{RN}^{oracle}$	$\hat{\mathbf{f}}_{RN}^{est}$	$\hat{\mathbf{f}}_{RB}^{oracle}$	$\hat{\mathbf{f}}_{RB}^{est}$	$\hat{\mathbf{f}}_{RH}^{oracle}$	$\hat{\mathbf{f}}_{RH}^{est}$	$\hat{\mathbf{f}}_{SVD}$
$SNR = 3$	0.0365)	(0.0399)	(0.0388)	(0.0409)	(0.0401)	(0.0399)	(0.0386)
$n = 64$,	0.2702	0.2749	0.2700	0.2745	0.2691	0.2734	0.2756
$SNR = 5$	(0.0310)	(0.0341)	(0.0323)	(0.0347)	(0.0316)	(0.0337)	(0.0343)
$n = 128$,	0.4065	0.4146	0.4054	0.4144	0.4055	0.4144	0.4266
$SNR = 3$	(0.0302)	(0.0325)	(0.0296)	(0.0303)	(0.0303)	(0.0299)	(0.0307)
$n = 128$,	0.2545	0.2579	0.2548	0.2575	0.2545	0.2579	0.2593
$SNR = 5$	(0.0191)	(0.0188)	(0.0189)	(0.0188)	(0.0189)	(0.0190)	(0.0181)

than the oracle SVD estimator. The advantage of using random dictionaries is more noticeable when n is small ($n = 32$) and the noise level is high ($SNR = 3$). In addition, the improvement of $\hat{\mathbf{f}}_{RN}^{oracle}$, $\hat{\mathbf{f}}_{RB}^{oracle}$ and $\hat{\mathbf{f}}_{RH}^{oracle}$ over $\hat{\mathbf{f}}_{SVD}$ is more significant than that of $\hat{\mathbf{f}}_{RN}^{est}$, $\hat{\mathbf{f}}_{RB}^{est}$ and $\hat{\mathbf{f}}_{RH}^{est}$ since the latter estimators loose accuracy because of suboptimal choices of the parameter α. Nevertheless, in the majority of cases, they still exhibit better precision than $\hat{\mathbf{f}}_{SVD}$ although this is not entirely fair comparison since $\hat{\mathbf{f}}_{SVD}$ is based on the oracle choice of parameter K. This is due to the fact that large random dictionaries provide a more sparse representation of $\mathbf{f}$.

To study the performance of the suggested method in a practical setting, we used a real-life signal for $\mathbf{f}$ that consists of a bird's twitter available as an audio signal on the internet at http://www.externalharddrive.com/waves/animal/index.html. We used parts of the signal of lengths $n = 50$ and applied an averaging matrix operator $\mathbf{Q}$ to evaluate $\mathbf{q}$. Matrix $\mathbf{Q}$ is chosen as the Toeplitz matrix with unit diagonal entries, upper diagonal entries equal to 0.5 and the rest of the entries equal to zero. We obtained the noisy observations $\mathbf{y}$ by adding Gaussian random noise to $\mathbf{q}$ as before. In our study we used $p = 3000$ and considered three noise levels: $SNR = 3, SNR = 5$, and $SNR = 7$. We constructed estimators of $\mathbf{f}$ based on two random dictionaries, the sparse Bernoulli dictionary defined in Eq. 4.1 and the symmetric Bernoulli dictionary with i.i.d. random entries given by

$$\mathbf{X} = \begin{cases} 1/n & \text{with probability } 1/2 \\ -1/n & \text{with probability } 1/2. \end{cases} \tag{4.3}$$

We denote estimators based on dictionaries (4.1) and (4.3) by $\hat{\mathbf{f}}_{RB1}$ and $\hat{\mathbf{f}}_{RB2}$, respectively, and characterize estimation precision by $R(\hat{\mathbf{f}})$.

Table 2 below compares the accuracies of the random dictionary based estimators with the SVD estimator using the relative error (4.2) averaged over 50 simulation runs (with the standard deviations of the precision listed

Table 2: The average values of the relative errors $R(\hat{\mathbf{f}})$ evaluated over 50 simulation runs of the estimators for the test signal (standard deviations of the relative errors are listed in the parentheses)

Bird's twitter

	$\hat{\mathbf{f}}^{oracle}_{RB1}$	$\hat{\mathbf{f}}^{est}_{RB1}$	$\hat{\mathbf{f}}^{oracle}_{RB2}$	$\hat{\mathbf{f}}^{est}_{RB2}$	$\hat{\mathbf{f}}_{SVD}$
$n = 50$,	0.4266	0.4458	0.4218	0.4394	0.4319
$SNR = 3$	(0.0557)	(0.0618)	(0.0551)	(0.0598)	(0.0307)
$n = 50$,	0.2711	0.2772	0.2692	0.2758	0.2795
$SNR = 5$	(0.0354)	(0.0365)	(0.0365)	(0.0363)	(0.0341)
$n = 50$,	0.1972	0.2010	0.1960	0.1986	0.2014
$SNR = 7$	(0.0261)	(0.0266)	(0.0269)	(0.0269)	(0.0260)

in parentheses). For all the estimators based on random dictionaries, we report the errors of the estimators with both the oracle and the estimated values of α, denoted by $\hat{\mathbf{f}}^{oracle}_{RB1}$, $\hat{\mathbf{f}}^{oracle}_{RB2}$ and $\hat{\mathbf{f}}^{est}_{RB1}$, $\hat{\mathbf{f}}^{est}_{RB2}$, respectively. From Table 2 it follows that all the random dictionary based estimators are more accurate than the SVD estimators for the real-life signal with approximately 5% better accuracy. Again as in the case of artificial test signals, the random dictionary estimators performed better for high noise level (SNR = 3). Also, similarly to the case of artificial signals, the advantage of $\hat{\mathbf{f}}^{oracle}_{RB1}$ and $\hat{\mathbf{f}}^{oracle}_{RB2}$ over $\hat{\mathbf{f}}_{SVD}$ is more significant than that of $\hat{\mathbf{f}}^{est}_{RB1}$ and $\hat{\mathbf{f}}^{est}_{RB2}$ since the latter estimators loose accuracy because of suboptimal choices of the parameter α. Nevertheless, in the majority of cases, $\hat{\mathbf{f}}^{est}_{RB1}$ and $\hat{\mathbf{f}}^{est}_{RB2}$ still exhibit better precisions than $\hat{\mathbf{f}}_{SVD}$ although this is not entirely fair comparison since $\hat{\mathbf{f}}_{SVD}$ is based on the oracle choice of parameter K.

5 Discussion

In the present paper, we provided a new approach for the solution of a general ill-posed linear inverse problem. The underlying idea is to use lasso technique for estimating the function of interest by representing it as a sparse linear combination of elements of a random overcomplete dictionary. The advantage of choosing a random dictionary over any other overcomplete dictionary is that one can construct it in such a way that it satisfies restricted isometry condition with a high probability and, therefore, ensures that the compatibility conditions, that guarantee fast convergence rates for the lasso, also hold.

We provide theoretical justification for application of lasso technique with the random dictionaries for the solution of the linear inverse problems. We also support our theory by the simulation studies and a real data example which show that the proposed estimators have higher accuracies than the SVD estimators in spite of the fact that the SVD estimators are based on the oracle parameter choices. Therefore, the advantage of the random dictionary based estimators is more pronounced when they are likewise constructed with the oracle choices of parameter α. In fact, this is the place where our method has some room for improvement since our procedure for estimating parameter α is rather elementary and can probably be fine-tuned using more advanced techniques (Appendix).

Acknowledgements. Marianna Pensky and Pawan Gupta were partially supported by National Science Foundation (NSF), grants DMS-1407475 and DMS-1712977.

Appendix: Proofs

PROOF OF THEOREM 1. The beginningof the proof is similar to the proof of Lemma 2 in Pensky (2016). However, for completeness, we provide the complete proof here. Let $\boldsymbol{\theta}$ be the true parameter vector so that $\mathbf{f} = \mathbf{f}_{\boldsymbol{\theta}} = \boldsymbol{\Phi}\boldsymbol{\theta}$. Denote $\boldsymbol{\zeta} = \boldsymbol{\Psi}^T\boldsymbol{\eta}$. Then, it is easy to check that

$$\boldsymbol{\Phi}^T(\mathbf{z} - \mathbf{f}) = \boldsymbol{\Psi}^T(\mathbf{y} - \mathbf{Q}\mathbf{f}) = \sigma\boldsymbol{\zeta}.$$

Following Dalalyan et al. (2014), by K-K-T condition, we derive that for any $\mathbf{t} \in \mathbb{R}^p$

$$\begin{aligned}\widehat{\boldsymbol{\theta}}^T\boldsymbol{\Phi}^T(\mathbf{z} - \boldsymbol{\Phi}\widehat{\boldsymbol{\theta}}) &= \alpha\sum_{j=1}^{p}\nu_j|\widehat{\theta}_j| \\ \mathbf{t}^T\boldsymbol{\Phi}^T(\mathbf{z} - \boldsymbol{\Phi}\widehat{\boldsymbol{\theta}}) &\le \alpha\sum_{j=1}^{p}\nu_j|t_j|,\end{aligned}$$

so that, subtracting the first line from the second, we obtain

$$(\boldsymbol{\Phi}\widehat{\boldsymbol{\theta}} - \boldsymbol{\Phi}\mathbf{t})^T(\boldsymbol{\Phi}\widehat{\boldsymbol{\theta}} - \mathbf{z}) \le \alpha\sum_{j=1}^{p}\nu_j(|t_j| - |\widehat{\theta}_j|). \tag{A.1}$$

Then, (A.1) yields $(\boldsymbol{\Phi}\widehat{\boldsymbol{\theta}} - \boldsymbol{\Phi}\mathbf{t})^T(\boldsymbol{\Phi}\widehat{\boldsymbol{\theta}} - \boldsymbol{\Phi}\boldsymbol{\theta}) \le \sigma(\widehat{\boldsymbol{\theta}} - \mathbf{t})^T\boldsymbol{\zeta} + \alpha\sum_{j=1}^{p}\nu_j(|t_j| - |\widehat{\theta}_j|)$. Since for any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^p$ one has $\mathbf{v}^T\mathbf{u} = \frac{1}{2}\left[\|\mathbf{v}\|^2 + \|\mathbf{u}\|^2 - \|\mathbf{v} - \mathbf{u}\|^2\right]$, choosing $\mathbf{v} = \boldsymbol{\Phi}\widehat{\boldsymbol{\theta}} - \boldsymbol{\Phi}\mathbf{t}$ and $\mathbf{u} = \boldsymbol{\Phi}\widehat{\boldsymbol{\theta}} - \boldsymbol{\Phi}\boldsymbol{\theta}$ for any $\mathbf{t} \in \mathbb{R}^p$ obtain

$$\|\mathbf{f}_{\widehat{\boldsymbol{\theta}}} - \mathbf{f}\|^2 + \|\boldsymbol{\Phi}(\widehat{\boldsymbol{\theta}} - \mathbf{t})\|^2 \le \|\mathbf{f}_{\mathbf{t}} - \mathbf{f}\|^2 + 2\sigma(\widehat{\boldsymbol{\theta}} - \mathbf{t})^T\boldsymbol{\zeta} + 2\alpha\sum_{j=1}^{p}\nu_j(|t_j| - |\widehat{\theta}_j|). \tag{A.2}$$

By definition of $\boldsymbol{\zeta}$, for any $j = 1, \cdots, p$, one has $\zeta_j \sim \mathcal{N}(0, \nu_j^2)$. Hence, on the set

$$\Omega_0 = \left\{ \omega : \max_{1 \le j \le p} (\nu_j^{-1} |\zeta_j|) \le \sqrt{2(\tau+1)\log p} \right\} \quad \text{with} \quad \mathbb{P}(\Omega_0) \ge 1 - 2p^{-\tau} \tag{A.3}$$

one obtains $|(\widehat{\boldsymbol{\theta}} - \mathbf{t})^T \boldsymbol{\zeta}| \le \sqrt{2(\tau+1)\log p} \sum_{j=1}^p \nu_j \, |\widehat{\theta}_j - t_j| = \alpha_0 \sum_{j=1}^p \nu_j \, |\widehat{\theta}_j - t_j|$. Combining the last inequality with (A.2) obtain that, for any $\alpha > 0$, on the set Ω_0,

$$\|\mathbf{f}_{\widehat{\boldsymbol{\theta}}} - \mathbf{f}\|^2 + \|\boldsymbol{\Phi}(\widehat{\boldsymbol{\theta}} - \mathbf{t})\|^2 \le \|\mathbf{f}_{\mathbf{t}} - \mathbf{f}\|^2 + 2\alpha \sum_{j=1}^p \nu_j (|t_j| - |\widehat{\theta}_j|) + 2\alpha_0 \sum_{j=1}^p \nu_j \, |\widehat{\theta}_j - t_j|. \tag{A.4}$$

Denote $\boldsymbol{\Omega}_1 = \{\omega : \lambda_{\min}(2s; \boldsymbol{\Phi}) \ge 1 - \delta\}$, $\boldsymbol{\Omega}_2 = \{\omega : |J_*| \le s\}$ and $\boldsymbol{\Omega}_3 = \left\{\omega : \|\widehat{\boldsymbol{\theta}}\|_0 \le s\right\}$. Choose $\mathbf{t}$ such that $\mathbf{f}_{\mathbf{t}} = \text{proj}_{\mathcal{L}_{J_*}} \mathbf{f} = \mathbf{f}_{\mathcal{L}_{J_*}}$ and note that $t_j = 0$ for $j \in J_*^c$. Then, due to $\alpha \ge \alpha_0$ and $|\widehat{\theta}_j - t_j| \le |\widehat{\theta}_j| + |t_j|$, obtain

$$\|\mathbf{f}_{\widehat{\boldsymbol{\theta}}} - \mathbf{f}\|^2 + \|\boldsymbol{\Phi}(\widehat{\boldsymbol{\theta}} - \mathbf{t})\|^2 \le \|\mathbf{f} - \mathbf{f}_{\mathcal{L}_{J_*}}\|^2 + 4\alpha \sum_{j \in J_*} \nu_j \, |\widehat{\theta}_j - t_j|. \tag{A.5}$$

Consider the set $\boldsymbol{\Omega} = \boldsymbol{\Omega}_0 \cap \boldsymbol{\Omega}_1 \cap \boldsymbol{\Omega}_2 \cap \boldsymbol{\Omega}_3$ and note that $\mathbb{P}(\boldsymbol{\Omega}) \ge 1 - 2p^{-\tau} - \epsilon_1 - \epsilon_2 - \epsilon_3$. If $\omega \in \boldsymbol{\Omega}$, then $\|\widehat{\boldsymbol{\theta}} - \mathbf{t}\|_0 \le 2s$ and, hence,

$$4\alpha \sum_{j \in J_*} \nu_j \, |\widehat{\theta}_j - t_j| \le 4\alpha \left(\sum_{j \in J_*} \nu_j^2 \right)^{1/2} \frac{\|\boldsymbol{\Phi}(\widehat{\boldsymbol{\theta}} - \mathbf{t})\|}{\lambda_{\min}(2s; \boldsymbol{\Phi})} \le \|\boldsymbol{\Phi}(\widehat{\boldsymbol{\theta}} - \mathbf{t})\|^2 + \frac{4\alpha^2}{(1-\delta)^2} \sum_{j \in J_*} \nu_j^2.$$

Plugging the last inequality into (A.5) and recalling the definition of J_*, we derive (3.5).

Proof of Lemma 1. In cases a) and b), $\lambda_{\min}(m; \boldsymbol{\Phi}) \ge 1 - \delta$ is ensured by Theorem 5.65 of Vershynin (2012). In case c), note that entries of matrix $\boldsymbol{\Phi}$ are uncorrelated and, hence, are independent Gaussian variables due to

$$\text{Cov}\,(\boldsymbol{\Phi}_{ik}, \boldsymbol{\Phi}_{jl}) = \frac{1}{c^2} \sum_{r_1=1}^m \sum_{r_2=1}^m \mathbf{d}_{ir_1} \mathbf{d}_{jr_2} I(r_1 = r_2) I(k = l) = I(i = j) I(k = l).$$

Moreover, matrix $\boldsymbol{\Phi}$ has isotropic rows since

$$\text{Cov}\,(\boldsymbol{\Phi}_{ih}, \boldsymbol{\Phi}_{jl}) = \frac{1}{c^2} \sum_{r_1=1}^m \sum_{r_2=1}^m \mathbf{d}_{ir_1} \mathbf{d}_{jr_2} I(r_1 = r_2) I(h = l) = I(i = j) I(h = l).$$

Therefore, $\lambda_{\min}(m; \boldsymbol{\Phi}) \ge 1 - \delta$ by Theorem 5.65 of Vershynin (2012).

References

ABRAMOVICH, F. and SILVERMAN, B. W. (1998). Wavelet decomposition approaches to statistical inverse problems. *Biometrika* **85**, 115–129.

ABRAMOVICH, F., PENSKY, M. and ROZENHOLC, Y. (2013). Laplace deconvolution with noisy observations. *Electron. J. Statist.* **7**, 1094–1128.

BICKEL, P. J., RITOV, Y. and TSYBAKOV, A. (2009). Simultaneous analysis of Lasso and Dantzig selector. *Ann. Statist.* **37**, 1705–1732.

BÜHLMANN, P. and VAN DE GEER, S. (2011). Statistics for High-Dimensional Data, Methods, Theory and Applications. Springer.

CANDÈS, E. J., ELDAR, Y., NEEDELL, D. and RANDALL, P. (2010). Compressed sensing with coherent and redundant dictionaries. *Appl. Computat. Harmonic Anal.* **31**, 59–73.

CAVALIER, L. and REISS, M. (2014). Sparse model selection under heterogeneous noise: Exact penalisation and data-driven thresholding. *Electron. J. Statist.* **8**, 432–455.

CAVALIER, L., GOLUBEV, G. K., PICARD, D. and TSYBAKOV, A. B. (2002). Oracle inequalities for inverse problems. *Ann. Statist.* **30**, 843–874.

COHEN, A., HOFFMANN, M. and REISS, M. (2004). Adaptive wavelet Galerkin methods for linear inverse problems. *SIAM J. Numer. Anal.* **42**, 1479–1501.

COMTE, F., CUENOD, C. -A., PENSKY, M. and ROZENHOLC, Y. (2017). Laplace deconvolution on the basis of time domain data and its application to dynamic contrast enhanced imaging. *J. Royal Stat. Soc., Ser.B* **79**, 69–94.

DALALYAN, A. S., HEBIRI, M. and LEDERER, J. (2014). On the prediction performance of the Lasso. ArXiv:1402.1700.

DONOHO, D. L. (1995). Nonlinear solution of linear inverse problems by wavelet-vaguelette decomposition. *Appl. Comput. Harmon. Anal.* **2**, 101–126.

FOUCART, S. and RAUHUT, H. (2013). *A Mathematical Introduction to Compressive Sensing.* Springer, New York.

MAIRAL, J. (2014). SPAMS: A Sparse Modeling Software, MatLab toolbox. http://spams-devel.gforge.inria.fr.

PENSKY, M. (2016). Solution of linear ill-posed problems using overcomplete dictionaries. *Ann. Statist.* **44**, 1739–1764.

TROPP, J. A. and WRIGHT, S. J. (2010). Computational methods for sparse solution of linear inverse problems. *Proc. IEEE, special issue, Appl. Sparse Represent. Compressive Sensing* **98**, 948–958.

VARESCHI, T. (2013). Noisy Laplace deconvolution with error in the operator. *J. Statist. Plan. Inf.* **157-158**, 16–35.

VERSHYNIN, R. (2012). *Introduction to the non-asymptotic analysis of random matrices.* Cambridge University Press, ELDAR, Y. and KUTYNIOK, G. (eds.),. Chapter 5.

PAWAN GUPTA
MARIANNA PENSKY
DEPARTMENT OF MATHEMATICS,
UNIVERSITY OF CENTRAL FLORIDA,
ORLANDO, FL, USA
E-mail: marianna.pensky@ucf.edu

Paper received: 20 November 2016.