Closed-Form MMSE Estimation for Signal Denoising Under Sparse Representation Modeling Over a Unitary Dictionary

Matan Protter, Irad Yavneh, and Michael Elad, *Senior Member, IEEE*

***Abstract*—This paper deals with the Bayesian signal denoising problem, assuming a prior based on a sparse representation modeling over a unitary dictionary. It is well known that the maximum *a posteriori* probability (MAP) estimator in such a case has a closed-form solution based on a simple shrinkage. The focus in this paper is on the better performing and less familiar minimum-mean-squared-error (MMSE) estimator. We show that this estimator also leads to a simple formula, in the form of a plain recursive expression for evaluating the contribution of every atom in the solution. An extension of the model to real-world signals is also offered, considering heteroscedastic nonzero entries in the representation, and allowing varying probabilities for the chosen atoms and the overall cardinality of the sparse representation. The MAP and MMSE estimators are redeveloped for this extended model, again resulting in closed-form simple algorithms. Finally, the superiority of the MMSE estimator is demonstrated both on synthetically generated signals and on real-world signals (image patches).**

***Index Terms*—Maximum *a posteriori* probability (MAP), minimum mean squared error (MMSE), sparse representations, unitary dictionary.**

I. Introduction

ONE OF THE most fundamental and extensively studied problems in signal processing is the removal of additive noise, known as denoising. In this task, it is assumed that the measured signal $\boldsymbol{y} \in \mathbb{R}^n$ is the result of a clean signal $\boldsymbol{x} \in \mathbb{R}^n$ being contaminated by noise,[1] $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e}$. As is often done, we limit the discussion to zero-mean independent identically distributed (i.i.d.) Gaussian noise.

In order to be able to distinguish the signal from the noise, it is important to characterize the signal family as well. One very successful model, that has attracted attention in recent years, leans on the signal's sparsity with respect to some transform. In such a model, the signal is assumed to be representable as a linear combination of a few basic signal building blocks known as *atoms*. Formally put, $\boldsymbol{x}$ can be represented as $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}$, where $\boldsymbol{D} \in \mathbb{R}^{n \times m}$ is a known dictionary (set of atoms $\{\boldsymbol{d}_j\}_{j=1}^m$) and $\boldsymbol{\alpha}$ is a *sparse* vector of coefficients. "Sparse" here means that $\boldsymbol{\alpha}$ contains a small number (compared to n) of nonzero coefficients. In general, the dictionary may be redundant, containing more atoms than the dimension of the signal ($m \geq n$).

How can this model be used for recovering $\boldsymbol{x}$ from the measurement $\boldsymbol{y}$? A commonly used method (see [1] and references therein) is to seek a signal $\hat{\boldsymbol{x}}$ that is both sparse with respect to $\boldsymbol{D}$ (i.e., has a sparse representation) and close enough to the measured signal. This task can be written as seeking the representation $\hat{\boldsymbol{\alpha}}$ defined by

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 + \lambda \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 \tag{1}$$

where $\|\boldsymbol{\alpha}\|_0$ counts the number of nonzeros in $\boldsymbol{\alpha}$ and λ is a positive parameter. This energy functional contains two terms, the first promoting sparsity of the signal and the second promoting proximity to the measurement. This minimization task can be shown to be related to the maximum *a posteriori* (MAP) probability estimator [1].

Solving the minimization task is in general NP-hard [2], and therefore approximate solvers are required. One approach can be to replace the ℓ_0-norm with ℓ_1, leading to a family of algorithms known as Basis Pursuit [3]. Another commonly used approach is a greedy algorithm, such as the Orthogonal Matching Pursuit (OMP) [4]–[6]. In this algorithm, one atom is selected at each step, such that the norm of the residual (that portion of the signal not yet represented) is best decreased.

While MAP estimation, as manifested above, promotes seeking the single sparsest representation, recent work shows that a better result (in the L_2 sense) is possible using the minimum-mean-squared-error (MMSE) estimator [7]–[10]. The MMSE estimator requires a weighted average of all the possible sparse representations that may explain the signal, with weights related to their probability. Just like the MAP in the general setting, this estimation is infeasible to compute, and thus an approximation is proposed. For example, the work reported in [7] and [8] offers approximations based on a tree search for candidate solutions with pruning of ones less likely to explain well the signal. Similarly, the work reported in [9] suggests a random version of the OMP for getting several representations, followed by plain averaging.

More broadly, in the realm of sparse representations, mixing several estimators to get a better estimate has been studied in various directions in the past decade. One such direction considers fusion of estimators that use different dictionaries [11], [12]. The machine-learning and the statistics literature offers several recent contributions (see [13]–[15] for representative

Manuscript received April 07, 2009; accepted February 23, 2010. Date of publication March 25, 2010; date of current version June 16, 2010. This work was supported in part by the European Community's FP7-FET program, in part by the SMALL project, under Grant Agreement 225913, and was in part by the Israel Science Foundation under Grant 599/08.

The authors are with the Computer Science Department, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: matanpr@cs.technion.ac.il; ira@cs.technion.ac.il; elad@cs.technion.ac.il).

Digital Object Identifier 10.1109/TSP.2010.2046596

[1]While the discussion in this paper is over the reals, all the derivations here apply over the complex field just as well.

work), where a group of competing estimators are combined (aggregated) using exponential weights, leading to an estimate that goes beyond the best of the group. Clearly, there is a growing interest in Bayesian estimators that go beyond the MAP, and in non-Bayesian techniques that provide an alternative motivation for aggregation of sparse estimators.

In this paper we focus on the special case where the dictionary $\boldsymbol{D}$ is square $(n = m)$ and unitary $\boldsymbol{D}^T\boldsymbol{D} = \boldsymbol{I}$. In such a case, the problem formed in (1) need not be approximated, as there is a closed-form noniterative solution in the form of shrinkage over simple inner products [16]–[19]. Furthermore, the OMP becomes exact in such a case. Naturally, these facts make the MAP estimator a very appealing approach for the unitary case.

The question we address in this paper is the following: Does the MMSE estimator also enjoy a simple closed-form solution for the unitary case? We show that this is indeed the case, and develop a recursive formula that leads to the exact MMSE estimation. We start our treatment with a simple model that assumes that all the nonzero entries in the representation are drawn from the same distribution (i.i.d), and with a fixed and known cardinality. We then present a more general signal model based on a sparse representation, considering heteroscedastic nonzero entries in the representation, allowing varying probabilities for the chosen atoms, and imposing a probability rule on the cardinality of sparse representation. We extend both the MAP and MMSE estimators to this more complex model, and derive simple and exact algorithms for obtaining these estimators. We test these estimators on both synthetic and real-world signals (image patches) and demonstrate the superior performance of the proposed MMSE estimator in these tests.

We note that a preliminary version of this paper has appeared in [10], showing the core recursive formula for the MMSE computation for a simple sparse representation modeling, and demonstrating it on elementary synthetic experiments. Here we present a much-extended version of this work that includes: 1) a richer model that better fits general content signals; 2) full development of the MAP and MMSE closed-form solutions for this extended model, and with more details; 3) a numerical stability analysis of the recursive formula; and 4) a wider experimental part with new tests on real-world signals, where the parameters of the model are estimated as part of the denoising process.

The structure of the paper is as follows. In the next section we formulate the denoising problem and review the prior work on the MAP and the MMSE estimators. Section III derives the closed-form recursive formula for the MMSE estimator for a simple signal model and analyzes its numerical behavior. In Section IV we propose an extended generative signal model and develop the MAP and MMSE estimators for it, resulting in simple and exact algorithms for their recovery. We also discuss the need and means to estimate the many free parameters of this model. Section V presents an empirical study on both synthetic and real-world signals, demonstrating the various algorithms developed, and Section VI presents conclusions.

II. Prior Work

In order to deploy the MAP and MMSE estimators for the denoising task, we need to start by defining the signal creation process. The literature on sparse representation modeling, and orthogonal wavelet coefficients in particular, is rich with ideas on how to model signals. A hierarchical Bernoulli-Gaussian mixture is commonly used to model such coefficients, in order to derive the shrinkage to be applied on them [20]–[24]. Alternatively, Generalized Gaussians have also been used to model these coefficients [25], [26]. Such models assume independence between these coefficients, which makes the consequent estimation task easier. In this work we take a different path, and follow closely the source model considered in [7]–[9], where a nonuniform prior on the selection of the nonzero coefficients is considered, with a subsequent coupling between the different coefficients.

We assume that $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}$ is generated by first choosing the support of $\boldsymbol{\alpha}$ (locations of nonzero coefficients), denoted by S, using the probability function $P(S)$. Following [9] and [10] we shall restrict our treatment *for now* to the case where all supports with $|S| = k$ are equally probable, and all the others have zero probability. In Section IV we remove this limiting assumption and extend the analysis to the more general case. We denote this set of permissible supports by Ω_k. Once S is chosen, the representation's nonzeros are formed as a set of k random i.i.d. entries drawn from the Normal distribution $\mathcal{N}(0, \sigma_x^2)$.[2] As explained above, the signal $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}$ is then contaminated by a random i.i.d. Gaussian noise vector $\boldsymbol{e}$, resulting in the measured noisy vector $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e}$.

We define the $|S| \times m$ matrix $\boldsymbol{P}_S$ that extracts the $|S|$ nonzero entries from a sparse vector $\boldsymbol{\alpha} \in \mathbb{R}^m$, i.e., $\boldsymbol{P}_S\boldsymbol{\alpha} = \boldsymbol{\alpha}_S \in \mathbb{R}^{|S|}$. We further denote by $\boldsymbol{D}_S = \boldsymbol{D}\boldsymbol{P}_S^T$ the submatrix of $\boldsymbol{D}$ that contains only the columns corresponding to the support. We introduce the following two additional notations for simplicity of later expressions and analysis:

$$\boldsymbol{G}_S = \frac{1}{\sigma^2}\boldsymbol{D}_S^T\boldsymbol{D}_S + \frac{1}{\sigma_x^2}\boldsymbol{I} \qquad \boldsymbol{v}_S = \frac{1}{\sigma^2}\boldsymbol{D}_S^T\boldsymbol{y}.$$

For the signal model described herein, if the support S is known, the MMSE estimator for $\boldsymbol{x}$ (termed the *oracle*) is obtained by minimizing

$$J(S) = \frac{1}{\sigma^2}\|\boldsymbol{D}_S\boldsymbol{\alpha}_S - y\|^2 + \frac{1}{\sigma_x^2}\|\boldsymbol{\alpha}_S\|^2$$

and is given by

$$\begin{aligned}\hat{\boldsymbol{\alpha}}_{oracle} &= \left(\frac{1}{\sigma^2}\boldsymbol{D}_S^T\boldsymbol{D}_S + \frac{1}{\sigma_x^2}\boldsymbol{I}\right)^{-1}\frac{1}{\sigma^2}\boldsymbol{D}_S^T\boldsymbol{y} = \boldsymbol{G}_S^{-1}\boldsymbol{v}_S \\ \hat{\boldsymbol{x}}_{oracle} &= \boldsymbol{D}_S\hat{\boldsymbol{\alpha}}_{oracle}.\end{aligned} \tag{2}$$

This result can easily be obtained by observing that $P(\boldsymbol{\alpha}_S|\boldsymbol{y})$ is proportional to $P(\boldsymbol{y}|\boldsymbol{\alpha}_S)P(\boldsymbol{\alpha}_S)$ (using Bayes's rule). Due to the Gaussian noise, we have $P(\boldsymbol{y}|\boldsymbol{\alpha}_S) \propto \exp\{-\|\boldsymbol{D}_S\boldsymbol{\alpha}_S - \boldsymbol{y}\|_2^2/2\sigma^2\}$. Similarly, the Gaussian distribution of the nonzero entries in $\boldsymbol{\alpha}_S$ implies $P(\boldsymbol{\alpha}_S) \propto \exp\{-\|\boldsymbol{\alpha}_S\|_2^2/2\sigma_x^2\}$. Thus, $P(\boldsymbol{\alpha}_S|\boldsymbol{y})$ is a Gaussian distribution, and its mean (or maximum, as the two align) yields the oracle estimation of $\boldsymbol{\alpha}_S$. A multiplication by $\boldsymbol{D}_S$ leads to the oracle estimation of the corresponding signal $\boldsymbol{x}$, as in (2).

[2]We depart from the Zellner g-prior as used in [9]. This prior assumes orthogonalization of the columns of the support as part of the signal generation. See [8] and [9] for more details.

As the support in the actual problem is random and unknown, the MMSE estimation becomes an expectation over all possible supports. This is a weighted average of many such "oracles," $\boldsymbol{x}_S$, as given in (2), each considering one possible support. Those are to be weighted by their probability to explain $\boldsymbol{y}$, which leads to

$$\hat{\boldsymbol{x}}_{MMSE} = \sum_{S\in\Omega_k} P(S|\boldsymbol{y})\boldsymbol{x}_S = \sum_{S\in\Omega_k} P(S|\boldsymbol{y})\boldsymbol{D}_S\boldsymbol{G}_S^{-1}\boldsymbol{v}_S. \quad (3)$$

It can be shown [9] that, up to a normalization factor, $P(S|\boldsymbol{y})$ is given by

$$P(S|\boldsymbol{y}) \propto \exp\left\{\frac{1}{2}\boldsymbol{v}_S^T\boldsymbol{G}_S^{-1}\boldsymbol{v}_S + \frac{1}{2}\log\left(\det\left(\boldsymbol{G}_S^{-1}\right)\right)\right\}. \quad (4)$$

Roughly speaking, if we assume that $\boldsymbol{G}_S^{-1}$ is approximately proportional to $\boldsymbol{I}$, this expression suggests that highly probable supports are those with high energy remaining in the projection of $\boldsymbol{y}$ onto $\boldsymbol{D}_S$. For a more elaborate derivation of these terms we refer the interested reader to [9, eqs. (7) and (8)].

The MAP estimator is obtained by choosing the support S that maximizes the above probability, $P(S|\boldsymbol{y})$, and computing the oracle estimation for this support. Both this estimation and the MMSE one require in general a sweep through all $\binom{m}{k}$ supports in Ω_k, which is an infeasible task in general, due to the exponentially growing size of this set. Thus, OMP is used to approximate the MAP by solving an exact MAP estimator for $k = 1$ (one atom), peeling the portion of the signal found, and repeating the process [4].

Similarly, the MMSE needs to be approximated, and several methods have been proposed for this task in recent years. The work in [7] and [8] proposes a deterministic process of selecting a small group of well-chosen supports over which to average. Those are found in a greedy fashion, by forming a tree search and pruning less likely solutions. The Random-OMP algorithm [9] repeats the OMP several times, with a random choice of the next atom, based on $P(S|\boldsymbol{y})$ for $k = 1$. This yields an approximate Gibbs sampler for this distribution, and thus plain averaging of the representations found leads to a good approximation of the MMSE estimation. It is important to note in this context that the MMSE estimator and the Random-OMP that approximates it, generally do not result in a sparse representation, but they are still better than the MAP (as shown in [9]), even though the original signal is in fact sparse. This property of the estimators results from the aggregation of many (or in fact all) different supports, leading to an equivalent support which is not sparse. For a more detailed discussion of the phenomenon, see [9].

In the unitary case, any subset of columns from $\boldsymbol{D}$ is orthogonal (i.e., $\boldsymbol{D}_S^T\boldsymbol{D}_S = \boldsymbol{I}$), and thus the above expressions can be further simplified. Starting with the matrix $\boldsymbol{G}_S$, it becomes

$$\boldsymbol{G}_S = \frac{1}{\sigma^2}\boldsymbol{D}_S^T\boldsymbol{D}_S + \frac{1}{\sigma_x^2}\boldsymbol{I} = \left(\frac{1}{\sigma^2} + \frac{1}{\sigma_x^2}\right)\boldsymbol{I} = \frac{\sigma^2+\sigma_x^2}{\sigma^2\sigma_x^2}\boldsymbol{I}. \quad (5)$$

Similarly, the weights $P(S|\boldsymbol{y})$ become

$$P(S|\boldsymbol{y}) \propto \exp\left\{\frac{1}{2}\boldsymbol{v}_S^T\boldsymbol{G}_S^{-1}\boldsymbol{v}_S + \frac{1}{2}\log\left(\det\left(\boldsymbol{G}_S^{-1}\right)\right)\right\}$$

$$\propto \exp\left\{\frac{1}{2\sigma^2}\cdot\frac{\sigma_x^2}{\sigma^2+\sigma_x^2}\left\|\boldsymbol{D}_S^T\boldsymbol{y}\right\|_2^2\right\}. \quad (6)$$

Note that the log-factor has been removed as it is equal for all the supports in Ω_k. Furthermore, this probability is computed only up to a factor which equals $1/P(\boldsymbol{y})$. Instead of computing it directly, we use the fact that the sum of probabilities must equal to 1 in order to normalize the probabilities correctly.

Equation (6) clarifies that the MAP support is the one that maximizes $\left\|\boldsymbol{D}_S^T\boldsymbol{y}\right\|^2$, and is easily found by computing $\boldsymbol{D}^T\boldsymbol{y}$, sorting the resulting vector by (absolute) size, and choosing the first k elements. Thus, MAP for this case can be computed exactly with a simple algorithm. Furthermore, OMP in such a case is also exact, as the sequential detection of the largest inner product leads to the same outcome.

Naturally, we should wonder whether the unitary case offers such a simple and closed-form solution for the MMSE, which bypasses the need for the above described approximations (e.g., the Random-OMP). This is the topic of the next two sections.

III. Case of a Unitary Dictionary

A. MMSE Over a Unitary Dictionary—Fundamentals

The development in this section follows the one in [10] with important modifications to make the derivation clearer, more precise, and more general. Our goal is to show that for a unitary dictionary $\boldsymbol{D}$, the MMSE estimation can be computed exactly (up to rounding errors) while avoiding combinatorial computations. Recall that for a unitary matrix $\boldsymbol{D}$, we have

$$\|\boldsymbol{D}\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 = \|\boldsymbol{\alpha} - \boldsymbol{D}^T\boldsymbol{y}\|_2^2 = \sum_{i=1}^{n}(\alpha_i - \beta_i)^2$$

where $\boldsymbol{\beta} = \boldsymbol{D}^T\boldsymbol{y}$ and β_i the ith entry of $\boldsymbol{\beta}$. This will be helpful in later derivations.

The MMSE estimation in (3) can be read differently. Every possible support in the summation provides a candidate representation vector $\boldsymbol{\alpha}_S = \boldsymbol{G}_S^{-1}\boldsymbol{v}_S \in \mathbb{R}^k$. Multiplication of the form $\boldsymbol{P}_S^T\boldsymbol{\alpha}_S$ provides a sparse vector of length m that contains the entries of $\boldsymbol{\alpha}_S$ as its nonzeros. Thus, the MMSE estimator is given by

$$\hat{\boldsymbol{x}}_{MMSE} = \sum_{S\in\Omega_k} P(S|\boldsymbol{y})\boldsymbol{D}_S\boldsymbol{\alpha}_S = \boldsymbol{D}\cdot\sum_{S\in\Omega_k} P(S|\boldsymbol{y})\boldsymbol{P}_S^T\boldsymbol{G}_S^{-1}\boldsymbol{v}_S. \quad (7)$$

Here we have used the relation $\boldsymbol{D}_S = \boldsymbol{D}\boldsymbol{P}_S^T$, and thus the multiplication by $\boldsymbol{D}$ is performed outside the summation. This expression suggests that there is one effective representation that governs the estimated outcome, given (removing the multiplication by $\boldsymbol{D}$) by

$$\hat{\boldsymbol{\alpha}}_{MMSE} = \sum_{S\in\Omega_k} P(S|\boldsymbol{y})\boldsymbol{P}_S^T\boldsymbol{\alpha}_S = \sum_{S\in\Omega_k} P(S|\boldsymbol{y})\boldsymbol{P}_S^T\boldsymbol{G}_S^{-1}\boldsymbol{v}_S.$$

This implies that every one of the n (recall that $m = n$) atoms contributes a prespecified portion to the overall MMSE estimation. We shall exploit the fact that the matrix $\boldsymbol{D}$ is unitary, and construct a closed-form formula for these n contributions, thus turning this estimator into a practical algorithm.

Denote $c^2 = \sigma_x^2/(\sigma^2+\sigma_x^2)$. Returning to (6), we observe that in the unitary case

$$P(S|\boldsymbol{y}) \propto \exp\left\{\frac{c^2\left\|\boldsymbol{D}_S^T\boldsymbol{y}\right\|^2}{2\sigma^2}\right\} = \prod_{i\in S}\exp\left\{\frac{c^2\beta_i^2}{2\sigma^2}\right\} \propto \prod_{i\in S} q_i$$

where we have denoted

$$q_i = \frac{\exp\left(c^2\beta_i^2/2\sigma^2\right)}{\sum_{j=1}^{n}\exp\left(c^2\beta_j^2/2\sigma^2\right)}. \tag{8}$$

Thus we have

$$P(S|\boldsymbol{y}) = A_k\prod_{j\in S} q_j$$

where A_k is a normalizing constant yielding $\sum_{S\in\Omega_k} P(S|\boldsymbol{y}) = 1$. Note that for $k=1$, the probability of the support being the jth atom is simply $P(S=\{j\}|\boldsymbol{y}) = q_j$, hence $A_1 = 1$, since the q_j's are properly normalized. Now we can obtain a simpler formulation for the MMSE estimator. Using the notations of c^2 and β_i, we can write $\boldsymbol{G}_S^{-1} = \sigma^2c^2\boldsymbol{I}$ [from (5)] and $\boldsymbol{v}_S = \boldsymbol{P}_S\boldsymbol{\beta}/\sigma^2$. Assigning these and the formula for $P(S|\boldsymbol{y})$ into (7), we get that

$$\begin{aligned}\hat{\boldsymbol{x}}_{MMSE} &= \sum_{S\in\Omega_k} P(S|\boldsymbol{y})\cdot\boldsymbol{D}_S\boldsymbol{G}_S^{-1}\boldsymbol{v}_S\\ &= \sum_{S\in\Omega_k}\left[A_k\left(\prod_{i\in S} q_i\right)\boldsymbol{D}_S(\sigma^2c^2\boldsymbol{I})\left(\frac{1}{\sigma^2}\boldsymbol{P}_S\boldsymbol{\beta}\right)\right]\\ &= c^2A_k\sum_{S\in\Omega_k}\left[\left(\prod_{i\in S} q_i\right)\cdot(\boldsymbol{D}_S\boldsymbol{P}_S\boldsymbol{\beta})\right]\\ &= c^2A_k\sum_{S\in\Omega_k}\left[\left(\prod_{i\in S} q_i\right)\cdot\left(\sum_{i\in S}\beta_i\boldsymbol{d}_i\right)\right].\end{aligned} \tag{9}$$

Computing this formula in a straightforward manner requires a prohibitive $O(n^k)$ operations, as every group of $k=|S|$ atoms has to be considered and summed. In order to simplify this expression, we introduce the indicator function

$$I_S(i) = \begin{cases}1 & i\in S\\ 0 & i\notin S\end{cases}$$

and rewrite (9) as

$$\hat{\boldsymbol{x}}_{MMSE} = c^2A_k\sum_{S\in\Omega_k}\left[\left(\prod_{j\in S} q_j\right)\left(\sum_{i=1}^{n}\beta_i\boldsymbol{d}_iI_S(i)\right)\right].$$

Rearranging the order of summations and multiplications in this equation yields the equivalent expression

$$\begin{aligned}\hat{\boldsymbol{x}}_{MMSE} &= c^2A_k\sum_{i=1}^{n}\left[\left(\sum_{S\in\Omega_k} I_S(i)\left(\prod_{j\in S} q_j\right)\right)\beta_i\boldsymbol{d}_i\right]\\ &= c^2\sum_{i=1}^{n} q_i^k\beta_i\boldsymbol{d}_i\end{aligned} \tag{10}$$

where we have introduced the notation

$$q_i^k = A_k\sum_{S\in\Omega_k} I_S(i)\left(\prod_{j\in S} q_j\right). \tag{11}$$

The straightforward way to compute this scalar value would be by sweeping through all supports S in Ω_k that contain the ith atom (there are $\binom{n-1}{k-1}$ of those), computing for each of them

$$A_k\prod_{j\in S} q_j, \quad \text{i.e., } P(S|\boldsymbol{y})$$

and summing these up. Thus, q_i^k is nothing but the probability that atom i will be included in the support. This computation is still exponential and thus prohibitive, but, as we show next, an efficient recursive formula for these values is within reach. Note that, using this notation, the MMSE estimator can be written as $\hat{\boldsymbol{x}}_{MMSE} = c^2\boldsymbol{D}diag(\boldsymbol{q}^k)\boldsymbol{\beta}$, where $\boldsymbol{q}^k$ is a vector of length n comprised of the probabilities $\{q_i^k\}_{i=1}^n$, and $diag(\boldsymbol{v})$ is a diagonal matrix containing the values of $\boldsymbol{v}$ along its main diagonal.

B. Obtaining a Closed-Form MMSE Formula

We proceed toward our goal of a closed-form formula by considering by way of analogy the following game. Suppose that k balls are tossed independently at a group of n buckets of various sizes. Suppose that, if we were to toss a single ball, the probability that it would land in bucket i would be q_i (with $0 < q_i < 1$ for all i, and $\sum_i q_i = 1$, i.e., the ball always lands in some bucket). This "round" of k tosses is repeated over and over again. If the k balls fall into k different buckets in a given round, this round is declared valid and this k-tuple of buckets is tallied. However, if two or more balls fall into any single bucket in a given round of k tosses, the round is void and nothing is tallied. The task is to calculate the q_i^k—the probability that some ball will fall into bucket i in a valid round of k tosses—for $i = 1,\ldots,n$.

Why is this game relevant? A valid round consists of k independent tosses landing in k different buckets, and therefore the probability of any particular k-tuple of buckets is clearly proportional to the product of its q_i's. The probability of each bucket participating in the k-tuple is therefore the sum of probabilities of the k-tuples (S) that contain it, analogously to (11). Based on this analogy, we make the following observations, which will be useful later:

- **Base**: For $k=1$ (only a single toss) we get $\boldsymbol{q}^1 = \boldsymbol{q}$, the vector whose elements are q_i (the individual probabilities of each bucket), as defined in (8).
- **Bounds on $\boldsymbol{q}^k$**: Since every bucket has a nonzero probability, and at most participates in all tuples, we have $\boldsymbol{0} < \boldsymbol{q}^k < \boldsymbol{1}$ elementwise for $k = 1,\ldots,n-1$, and $\boldsymbol{q}^n = \boldsymbol{1}$, where $\boldsymbol{0}$ and $\boldsymbol{1}$ are the n-vectors of all zeros and all ones, respectively.
- **Preservation of order**: If $q_j \geq q_i$ then $q_j^k \geq q_i^k$ for $k = 1,\ldots,n$, with equality occurring if and only if $q_j = q_i$ or $k = n$. That is, a more likely bucket (with greater probability of being hit in a single toss) remains more likely as we increase the number of balls per round.

- **Monotonicity in** k: For $k = 2, \ldots, n$, $\boldsymbol{q}^k > \boldsymbol{q}^{k-1}$ elementwise, because increasing the number of balls increases the probability of every one of the buckets.
- **Monotonicity of ratios in** k: If $q_j^k > q_i^k$ then $q_j^k/q_i^k < q_j^{k-1}/q_i^{k-1}$, for $k = 2, \ldots, n$. This claim is nontrivial, and its proof is given in Appendix A.[3]
- **Symmetry**: Assume henceforth that one of the balls (only) is colored red. Since the color has no effect of any significance, the probability that the red ball will fall into bucket i in a given round is clearly equal to the probability that any one of the other $k-1$ balls will fall into this bucket.
- **Normalization**: The vectors $\boldsymbol{q}^k$ satisfy the normalization condition $\sum_{i=1}^n q_i^k = k$, that is, the sum of probabilities of all buckets is equal to the number of balls per round. This allows us to determine the A_k's. This property is implied by the Symmetry property, by which the probability that the red ball will fall into bucket i in a valid round is q_i^k/k. Since the overall probability that the red ball will fall into *some* bucket in a valid round is 1, we have $\sum_{i=1}^n q_i^k/k = 1$, from which the Normalization property follows.

We next derive the recursive formula for computing $\boldsymbol{q}^k$. For $k = 1$ we have $\boldsymbol{q}^1 = \boldsymbol{q}$ by the Base property, and for $j = 2, \ldots, k$, we have that q_i^j is proportional to the probability that the red ball will fall into bucket i (which is q_i) times the probability that the remaining balls will comprise a valid round of $j-1$ balls that *does not* include bucket i (which is $1-q_i^{j-1}$). This product needs to be normalized so as to satisfy the Normalization property, yielding

$$q_i^j = j\frac{q_i\left(1-q_i^{j-1}\right)}{1-\sum_{\ell=1}^n q_\ell q_\ell^{j-1}}. \tag{12}$$

The full vector of probabilities is thus given by

$$\boldsymbol{q}^j = F^j(\boldsymbol{q}^{j-1}) \equiv j\frac{diag(\boldsymbol{q})(\mathbf{1}-\boldsymbol{q}^{j-1})}{1-\boldsymbol{q}^T\boldsymbol{q}^{j-1}} \tag{13}$$

with $\boldsymbol{q}$ given in (8)

$$q_i = \frac{\exp\left(c^2\beta_i^2/2\sigma^2\right)}{\sum_{j=1}^n \exp\left(c^2\beta_j^2/2\sigma^2\right)}.$$

C. Numerical Instability

Unfortunately, the recursive formula (13) tends to suffer from instability, manifest in a fast growth of numerical errors during the iterations when k is not small. To study this effect, we perform a linear stability analysis. Suppose that $\boldsymbol{q}^{j-1}$ contains an error (vector), $\boldsymbol{\delta}^{j-1}$. Then, ignoring the (typically machine-accuracy, hence negligible) numerical errors in $\boldsymbol{q}$ and in the arithmetic operations of (13), we obtain by taking the first term of the Taylor series of F^j, given by

$$\boldsymbol{\delta}^j \approx \boldsymbol{C}^j\boldsymbol{\delta}^{j-1} \tag{14}$$

where $\boldsymbol{C}^j = \partial F^j(\boldsymbol{q}^{j-1})$ is the gradient matrix of F^j, which can be computed easily from (12). After rearrangement, the elements of $\boldsymbol{C}^j$ can be written as

$$C_{\ell,m}^j = \begin{cases} -\frac{\left(j-q_\ell^j\right)q_\ell^j}{j\left(1-q_\ell^{j-1}\right)} & \text{if } \ell = m \\ \frac{\left(q_\ell^j\right)^2 q_m}{j\left(1-q_\ell^{j-1}\right)q_\ell} & \text{otherwise.} \end{cases} \tag{15}$$

The error propagation per iteration is determined by the spectral properties of $\boldsymbol{C}^j$. These are hard to compute in general, but we can clearly see the source of the numerical trouble by considering a special case where two elements of $\boldsymbol{q}$ happen to be exactly the same. Without loss of generality, assume that these are the first two elements, i.e., $q_1 = q_2$, and therefore, by the Preservation of order property, $q_1^j = q_2^j$ for all j. For all j we then have by (15)

$$\begin{aligned} C_{1,1}^j &= C_{2,2}^j = -\frac{\left(j-q_1^j\right)q_1^j}{j\left(1-q_1^{j-1}\right)} \\ C_{2,1}^j &= C_{1,2}^j = \frac{\left(q_1^j\right)^2}{j\left(1-q_1^{j-1}\right)} \\ C_{\ell,1}^j &= C_{\ell,2}^j \quad \text{for all} \quad \ell > 2 \\ C_{1,m}^j &= C_{2,m}^j \quad \text{for all} \quad m > 2. \end{aligned}$$

It is now immediate to verify that the vector of size n given by $\boldsymbol{v} = (1, -1, 0, 0, \ldots, 0)^T$ is an eigenvector of $\boldsymbol{C}^j$ for all j, with eigenvalue given by

$$\lambda_j = C_{1,1}^j - C_{1,2}^j = -\frac{q_i^j}{1-q_i^{j-1}}.$$

For $q_1^j < 0.5$, we get $|\lambda_j| < 1$, and the iteration is stable with respect to errors of the form $\boldsymbol{v}$. However, by the Monotonicity property, q_1^j grows with j, eventually reaching 1 at $j = n$. Once q_1^{j-1} crosses 0.5, an oscillating (since $\lambda_j < -1$) pairwise antisymmetric divergence of the error kicks in, with the divergence rate growing with each iteration, because $|\lambda_j|$ grows with j. A key feature here is that the eigenvector $\boldsymbol{v}$ is shared by all the $\boldsymbol{C}^j$'s, so it grows in absolute value at each iteration (once $\lambda_j < -1$).

Although this analysis assumes a pair of equal elements, the unstable behavior it implies is quite general. Nevertheless, the instability can largely be kept at bay by enforcing the known constraints implied by the properties above on solutions obtained from the recursive formula. Imposing these constraints at each iteration of the recursive formula is a relatively cheap method of keeping the numerical errors under control. Furthermore, if at stage j during the calculation of the formula it is determined that one (or more) probability q_i^k attains a value sufficiently close to 1 (which also means that it is a source of numerical instability, cf., discussion above), we can set this value to 1 at all subsequent iterations due to the Monotonicity property. To improve the numerical accuracy for the rest of the entries, we may eliminate this element of $\boldsymbol{q}$ and then recalculate $\boldsymbol{q}^{k-1}$ for the remaining entries.

[3]A particular implication of this property is that it shows that the Random-OMP algorithm [9] remains inexact, even if given an infinite number of iterations to run. This is because the Random-OMP selects the atoms with probabilities according to the initial ratios q_j^1/q_i^1, while those ratios should decrease as k increases. This also hints that the inexactness of the Random-OMP increases with k.

TABLE I
GENERAL SIGNAL GENERATION MODEL, AND THE IMPLIED MAP AND MMSE ESTIMATORS

General Signal Generation Model

Parameters of the Model

1) $P_C(k)$: the probability of each support size $P_C(k) = P(|S| = k)$.
2) $\{P_i^a\}_{i=1}^n$: the probability of each atom to be chosen at each step.
3) σ_i^2: the variance of the coefficients for each atom.

Signal Generation Model

1) The support size $k = |S|$ is chosen according to $P_C(k)$.
2) Repeat until k unique atoms are chosen: Select k atoms sequentially, each time selecting one according to $\{P_i^a\}_{i=1}^n$
3) For the atoms selected $i \in S$, draw coefficients $\alpha_i \sim \mathcal{N}(0, \sigma_i^2)$
4) Construct the signal $\boldsymbol{x} = \sum_{i \in S} \alpha_i \boldsymbol{d}_i$.

MAP Estimator

$$\begin{aligned} q_i &= \frac{\exp(c^2\beta_i^2/2\sigma^2)}{\sum_{j=1}^n \exp(c^2\beta_j^2/2\sigma^2)} \quad \text{for} \quad i = 1, 2, \ \ldots \ , n \\ \tilde{q}_i &= q_i \cdot P_i^a \quad \text{for} \quad i = 1, 2, \ \ldots \ , n \\ T_k &= \sum_{S \in \Omega_k} \prod_{i \in S} P_i^a \quad \text{for} \quad k = 1, 2, \ \ldots \ , n \\ \hat{S}_{MAP} &= \arg\max_S \frac{P_C(|S|)}{T_{|S|}} \cdot \prod_{i \in S} \tilde{q}_i \\ \hat{\boldsymbol{x}}_{MAP} &= \sum_{i \in \hat{S}_{MAP}} c_i^2 \beta_i \boldsymbol{d}_i \end{aligned}$$

MMSE Estimator

$$\begin{aligned} \hat{\boldsymbol{x}}_{MMSE} &= \sum_{k=1}^n \frac{P_C(|S|)}{T_k} \sum_{i=1}^n q_i^k c_i^2 \beta_i \boldsymbol{d}_i \\ q_i^k &= k \frac{\tilde{q}_i (1 - q_i^{k-1})}{1 - \sum_{\ell=1}^n \tilde{q}_\ell q_\ell^{k-1}} \\ q_i^1 &= q_i \, . \end{aligned}$$

IV. EXTENDING THE MODEL TO REAL-WORLD SIGNALS

The model we have relied on so far has simplified the analysis and the derivation of the MAP and MMSE estimators. However, this model is far too limited for handling real-world signals. More specifically, we have relied on three assumptions that we cannot generally make:

- All coefficients in the support are assumed to be drawn according to the same normal distribution with the same variance σ_x^2.
- The size of the support $|S|$ is fixed and known.
- Given that $|S|$ is known, $P(S)$ is equal for all supports of this size, and hence all atoms are (*a priori*) equally likely to be selected.

Unfortunately, these assumptions are too simplistic for faithfully describing real-world signals (such as image patches), and thus cannot function as a good prior signal model for denoising. In order to construct a model fitting real-world signals, these assumptions must be relaxed and generalized, and the formulas for the MAP and MMSE estimators must be adapted accordingly. The assumption regarding the equal distribution of the coefficients is the first we choose to tackle. We relax the remaining two assumptions together by proposing a general signal generation model. The resulting model is general enough to describe a wide range of signals, and can be successfully harnessed for image denoising, as will be shown in Section V. We now describe in detail the required extensions and adaptations, which are then summarized in Table I.

A. Treating a Heteroscedastic Coefficient Set

Previously it was assumed that all coefficients share the same prior variance σ_x^2. Assuming that all coefficients behave identically is unrealistic, so we now allow the variance to be atom-dependent and denote it by σ_i^2. Accordingly, we define $c_i^2 = \sigma_i^2/(\sigma_i^2 + \sigma^2)$, which also becomes atom dependent. The oracle in the unitary case becomes

$$\hat{\boldsymbol{x}}_{oracle} = \sum_{i \in S} c_i^2 \beta_i \boldsymbol{d}_i. \tag{16}$$

This is easily verified following the explanation given in Section II for the derivation of the oracle formula in the general case. Using the fact that for the nonzero portion of $\boldsymbol{\alpha}$ we now have $P(\boldsymbol{\alpha}_S) \propto \exp\{-\sum_{i=1}^n \alpha_i^2/2\sigma_i^2\}$, we observe that the posterior probability $P(\boldsymbol{\alpha}_S|\boldsymbol{y})$ is Gaussian, and the expression given in (16) is its mean.

A second effect of the different variances per atom appears in the posterior probability $P(S|\boldsymbol{y})$. Using (4) and (6) one notices that the log-factor cannot be discarded, and this expression becomes

$$P(S|\boldsymbol{y}) \propto \prod_{i \in S} \exp\left\{ \frac{c_i^2 \beta_i^2}{2\sigma^2} + \frac{\log\left(c_i^2\right)}{2} \right\} \propto \prod_{i \in S} q_i \tag{17}$$

which implies a somewhat modified definition for q_i.

The MAP estimator selects the k atoms with the largest q_i values and projects onto them using the oracle formula in (16). The MMSE estimator uses a formula very similar to the one introduced before in (10)

$$\hat{\boldsymbol{x}}_{MMSE} = A_k \sum_{i=1}^{n} \left[\left(\sum_{S \in \Omega_k} I_S(i) \left(\prod_{j \in S} q_j \right) \right) c_i^2 \beta_i \boldsymbol{d}_i \right] = \sum_{i=1}^{n} q_i^k c_i^2 \beta_i \boldsymbol{d}_i \tag{18}$$

with the two changes being the redefinition of q_i and the atom-dependent value c_i replacing the constant c. Interestingly, the recursive formula for the update of q_i^k remains the same as in (12), as do the constraints that are employed in stabilizing its numerical evaluation.

B. Extending the Signal Generation Model

The assumption that only a specific cardinality exists, and moreover, that all supports of this cardinality are equally likely, is unrealistic. For example, smooth and slowly varying signals may have a very sparse representation, while highly textured signals may require many more atoms for an adequate representation. Furthermore, some atoms are expected to appear more frequently than others, increasing the probability of some supports and reducing the probability of other supports. These observations lead to the generative signal model we now consider.

Assume that the size of the support is chosen randomly according to a known probability $P_C(k) = P(|S| = k)$, thus relaxing the fixed support size constraint introduced in Section II. Then, k atoms are chosen sequentially, where atom i has a probability P_i^a of being selected (normalized such that $\sum_i P_i^a = 1$). If the resulting group consists of k distinct atoms, this support draw is considered valid; otherwise (i.e., in the event of at least one repetition) it is discarded and the random atom selection process is restarted. Lastly, the active coefficients for the selected support are drawn at random from the distributions $\mathcal{N}(0, \sigma_i^2)$, as before.

In order to adapt the estimators to this more general model, we should update the definition of $P(S)$ to reflect the new signal model. The probability of a specific support to be chosen is proportional to the probability of the size of the support multiplied by the individual probabilities of the atoms to be chosen $P(S) \propto P_C(|S|) \cdot \prod_{i \in S} P_i^a$, with a normalization such that for every k, $\sum_{S \in \Omega_k} P(S) = P_C(k)$. This implies that the choice of atoms is independent of the choice of support size. Denoting $T_k = \sum_{S \in \Omega_k} \prod_{i \in S} P_i^a$, the probability of a specific support S is given by

$$P(S) = \frac{P_C(|S|)}{T_{|S|}} \cdot \prod_{i \in S} P_i^a. \tag{19}$$

The formula for the normalization factor T_k is reminiscent of the formula for q_i^k given in (11). Indeed, in order to compute T_k we need to apply the recursive formula on the values $\{P_i^a\}_{i=1}^n$, and for each k, sum the resulting values (after undoing the numerically stabilizing normalization), and divide by k (as each possible support contributes to k entries). Note that in the general case, in which the signal model is to be applied to a large set of signals, this procedure is needed only once, as it is a property of the model and does not depend on the specific signal.

Using the *a priori* probability of each support, the overall posterior probability of each support becomes

$$P(S|\boldsymbol{y}) \propto P(\boldsymbol{y}|S) \cdot P(S) \propto \frac{P_C(|S|)}{T_{|S|}} \cdot \prod_{i \in S} (q_i \cdot P_i^a) = \frac{P_C(|S|)}{T_{|S|}} \cdot \prod_{i \in S} \tilde{q}_i \tag{20}$$

with $\tilde{q}_i = q_i \cdot P_i^a$, and q_i taken from (17).

The MAP estimator for this more general model is simply the one that maximizes the probability given in (20). Recovering it starts by computing $\tilde{q}_i$ for each atom. Then, at each step, one atom is added to the current representation, in descending order of magnitude of $\tilde{q}_i$, and the relative posterior probability of this support is computed according to (20). Of the n supports generated in this procedure, the likeliest one (which is also the likeliest over all supports) is selected, and by computing the oracle for this support, the MAP estimator emerges. Note that the value computed by (20) is not normalized, and therefore it does not represent a true probability. This has no effect on the MAP estimator, however, as we seek the support with the largest probability, and the order is not changed by the lack of normalization.

For the MMSE estimator, all cardinalities with their appropriate probabilities must be considered. Going back to (9), this translates into

$$\begin{aligned} \hat{\boldsymbol{x}}_{MMSE} &= \sum_{S} P(S|\boldsymbol{y}) \sum_{i \in S} c_i^2 \beta_i \boldsymbol{d}_i \\ &= \sum_{k} \sum_{S \in \Omega_k} P(S|\boldsymbol{y}) \cdot \sum_{i \in S} c_i^2 \beta_i \boldsymbol{d}_i \\ &= \sum_{k} \frac{P_C(k)}{T_k} \sum_{S \in \Omega_k} \prod_{i \in S} \tilde{q}_i \cdot \sum_{i \in S} c_i^2 \beta_i \boldsymbol{d}_i. \end{aligned}$$

The summation over $S \in \Omega_k$ is exactly the same as was developed for the single cardinality case in (18), with the slight redefinition of $\tilde{q}_i$ instead of q_i. Therefore, the recursive formula developed in the previous section can be used to obtain the MMSE estimator, by obtaining the MMSE estimate for each support size, and merging them with appropriate weights.

Some care is required, however, in the application of the recursive formula. In the single cardinality case, the various stabilizing normalizations could be ignored, as a normalization by the sum of weights was to be applied in the end. When applying the formula to this more general model, this normalization must be tracked and then undone, in order to properly reflect the relative weights of the different cardinalities.

C. Model Summary and Parameter Estimation

The generative model and its estimators are summarized in Table I. There are several parameters that govern the behavior of the model, and those are assumed to be known for the estimation task to complete. The model requires explicit and *a priori* knowledge of the variances $\{\sigma_i\}_{i=1}^n$ per atom, the prior probability of each support size $\{P_C(k)\}_{k=1}^n$ and the prior probability of each atom to be chosen $\{P_i^a\}_{i=1}^n$.

This model has a large number of parameters, and when applying the MAP and MMSE estimators to real-world signals,

these are unknown to begin with. Therefore, some method of estimating these parameters must be used, or else, the estimators are rendered useless.[4] One approach can be to use a set of high-quality (i.e., almost noise-free) signals in order to learn the parameters, and then apply the estimators based on these parameters. This approach, however, assumes that different images share the same parameter set.

An alternative method is to estimate these parameters from the noisy data directly. We assume that when facing a denoising task, many noisy signal instances are to be denoised together. For example, for image denoising, as will be the case in the experimental section, each 8×8 patch extracted from the image is considered as one noisy signal. Taking all these signals together, we may ask what would be the best set of parameters that describe these signals (taking into account that they are also noisy).

From the noisy signals, a direct maximum-likelihood (ML) approach can be undertaken to find the most likely set of parameters to have generated the noisy signals. Unfortunately, the maximization task obtained is quite complex. Instead, we adopt a block-coordinate-descent like approach, where the signals are first denoised by a parameter-less method (hard-thresholding, which is equivalent to a specific set of parameters that includes equal P_i^a and σ_i for all atoms), and from the cleaner signals we estimate the parameters using an ML formulation, which is built on clean-data. This approach of predenoising has been suggested elsewhere, such as in [27]. In principle, this method should be iterated, updating the parameters after the denoising. However, we found that one such iteration is sufficient to get a reliable set of estimates for the parameters, and this is indeed the way we operate in subsequent experiments. Therefore, the only manually set parameter is the parameter that controls the initial denoising, the threshold under which coefficients are considered to be zeros. More details on the parameter estimation process are given in Appendix B.

V. Experimental Results

We now proceed to demonstrate the proposed exact MMSE estimator and its superiority over the MAP. We also present one possible approximation of the MMSE, the Random-OMP algorithm [9], to illustrate the gain achieved by using the closed-form solution proposed. Our tests are performed first on synthetic signals, where the model parameters are known and are used by the estimators. We also introduce tests on real-world signals (image patches), where the parameters are unknown and therefore the estimation of the model parameters is required for the estimators as part of the overall treatment.

A. Synthetic Experiments

When performing synthetic tests, we have complete control over the signal generation process and its parameters. Since the parameters are known, as well as the standard deviation of the noise, their exact values are given to the estimators in order to check the estimators' performance in "optimal" settings. The dictionary used in all of the synthetic tests is generated randomly and then orthogonalized, to create a random unitary dictionary. We start by focusing on the simplest model (introduced in Section III), in which the support size k is known and fixed, and all supports are equally likely. Generating a signal according to this model is done by randomly choosing a set of k unique atoms, using a uniform probability over all $\binom{n}{k}$ possibilities. For the selected atoms, coefficients α_i are drawn independently from a Normal distribution $\mathcal{N}(0, \sigma_x^2)$. The resulting sparse vector of coefficients is multiplied by the dictionary to obtain the *ground-truth* signal. Each entry is independently contaminated by white Gaussian noise $(\mathcal{N}(0, \sigma^2))$ to create the *input* signal (note that due to $\boldsymbol{D}$ being unitary, this is equivalent to contaminating the coefficients themselves with additive white Gaussian noise with the same parameters). For all tests, the dimension of the signals is $n = 64$.

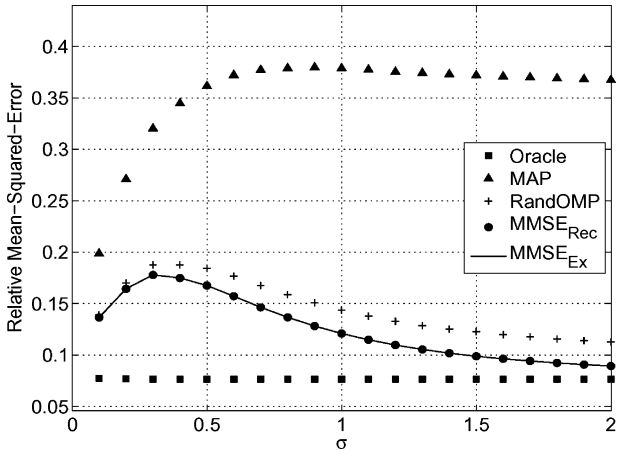

Fig. 1. Relative denoising achieved (compared to the noisy signal), averaged over a 1000 signals, by several methods, for different noise amplitudes and $|S| = 5$.

The noisy signal is denoised by several methods: 1) MAP estimator; 2) Random-OMP that approximates the MMSE [9] (averaging 20 representations); 3) an exact and exhaustive MMSE using (3) (the complexity of this estimator is exponential in k); 4) the recursive MMSE formula; and 5) an oracle that knows the exact support. This process is repeated for 1000 signals, and the mean L_2 error is averaged over all signals to obtain an estimate of the expected quality of each estimator. The denoising effect is quantified by the relative mean squared error (RMSE), which is obtained by dividing the MSE of each sample by the standard deviation of the noise, averaged over all signals. The RMSE reflects exactly the ratio between the noise energies in the reconstructed image and the initial one (e.g., an RMSE of 0.1 implies that the noise has been attenuated by a factor of 10).

In order to test the performance of these estimators under different noise conditions, several such tests are run, with $\sigma_x = 1$ kept constant in all tests, and the noise level σ varying in the range 0.1–2. This is sufficient, since the important parameter is the ratio σ_x/σ, and not their individual absolute values. Fig. 1 shows the denoising effect achieved by each method, when $|S| = 5$.

Next, we slightly generalize the generation model, by keeping the support size fixed $(|S| = 5)$ as before, but using a heteroscedastic coefficient set (where σ_i are linearly spaced in the range 0.5–2), and allowing each atom to have a different probability to appear ($P_i^a = 0.5^i$, normalized to 1 and randomly assigned to the atoms). The result of such a test appears in Fig. 2. It

[4]Note that σ, which characterizes the noise, is not part of these parameters, and in this work it is assumed as known.

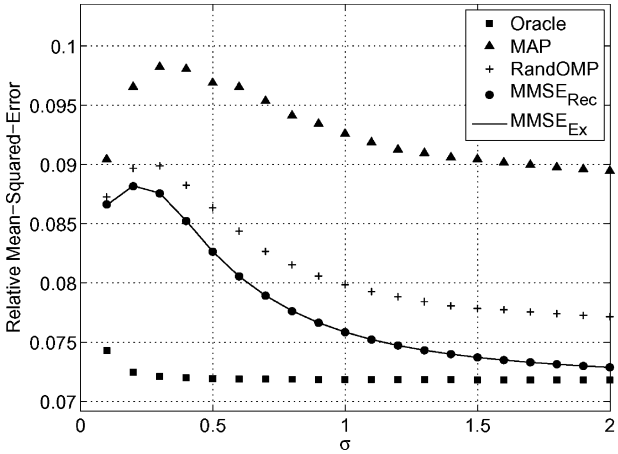

Fig. 2. Relative denoising achieved (compared to the noisy signal), averaged over a 1000 signals, by several methods, for different noise amplitudes, for $|S| = 5$, a heteroscedastic coefficient set and different probabilities for each atom.

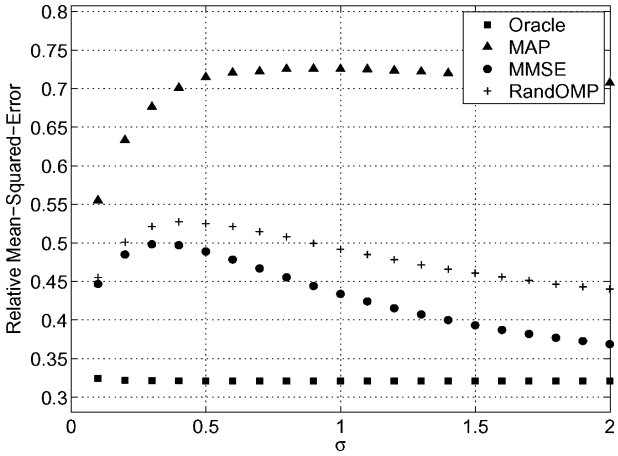

Fig. 3. Relative denoising achieved (compared to the noisy signal), averaged over a 1000 signals, by several methods, for different noise amplitudes, for $|S| = 20$, a heteroscedastic coefficient set and different probabilities for each atom.

is apparent that there is quite a big gap in performance between the MMSE estimator and its approximation via the Random-OMP, demonstrating the importance of the closed-form formula presented here.

The same test, but when the signals are not very sparse ($|S| = 20$) is displayed in Fig. 3, showing similar behavior. This test does not feature the exhaustive MMSE, due to its exponential complexity. In the last synthetic test, we apply the most general model, where the probability of each cardinality is given by $P_C(|S|) = 0.8^{|S|}$, $|S| = 1, \ldots, 5$ (normalized to sum to 1), with σ_i and P_i^a as in the previous test. The results for this test appear in Fig. 4, and are an average over five different random assignments (all of which yield similar results). This test does not include the Random-OMP estimator, which was originally developed only for the fixed support size scenario. As the focus of the paper is the exact estimator, we chose to avoid extending the approximate (and inferior) Random-OMP to the most general signal model.

To better understand the differences between the estimators, we show in Fig. 5 the effective representation achieved by each

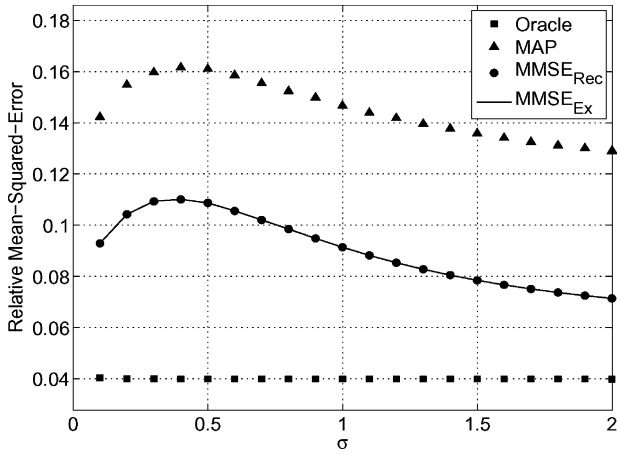

Fig. 4. Relative denoising achieved (compared to the noisy signal), averaged over a 1000 signals, by several methods, for the most general signal model (averaged over five different random assignments of atom probabilities).

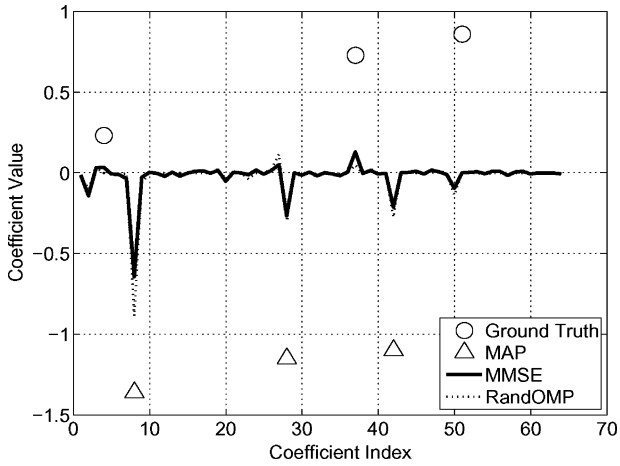

Fig. 5. Effective representation achieved by different methods for one example signal, with noise standard deviation $\sigma = 0.6$.

method (for $|S| = 3$), for one example signal. The MAP estimator selects the wrong atoms, due to the relatively strong noise ($\sigma = 0.6$).

B. Real-World Signals

In order to present experiments on real-world signals, we use 8×8 image patches drawn without overlap from an image, to which white Gaussian noise has been added. These are selected to compose the real-world data-set for our experiments. The unitary dictionary for these experiments is the discrete cosine transform (DCT) dictionary, which is known to serve natural image content adequately (i.e., sparsely).

It is important to note that there is no attempt to compare the estimators to the state of the art in image denoising. This is because our building blocks—such as a nonadaptive and unitary dictionary—are too limited for this comparison to be fair. Our goal is to demonstrate the superior performance of the MMSE estimator, and to offer the possibility that incorporating it into more complex denoising mechanisms may indeed improve denoising results.

Unlike the synthetic experiments detailed in the previous section, when working on real-world images the various parameters

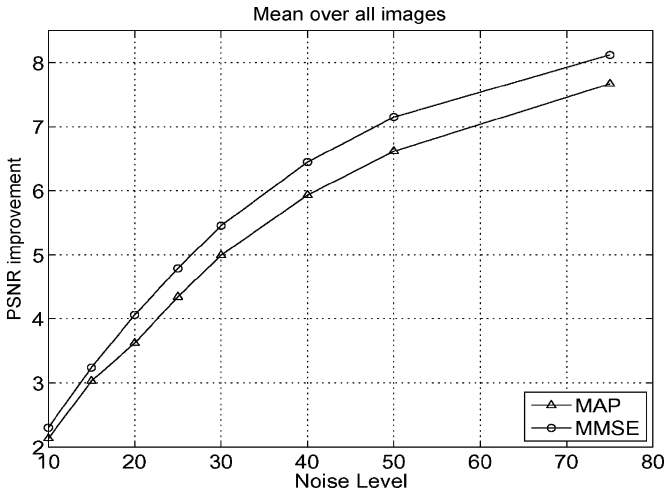

Fig. 6. Relative denoising achieved (compared to the noisy signal), averaged over all blocks in seven images, by the MAP and MMSE estimators.

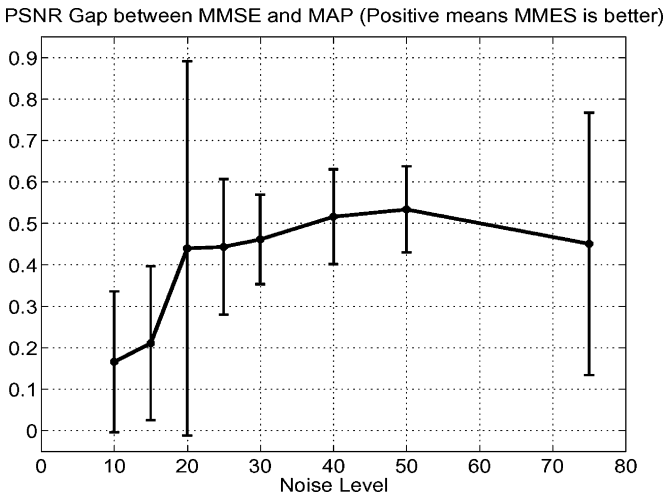

Fig. 7. PSNR gap between MMSE and MAP (with positive values indicating the MMSE is better performing), averaged over all blocks in seven images. The error bars indicate one standard deviation of the gap.

of the model are unknown, and must be estimated from the data. We note that we assume the noise variance σ^2 is known (or estimated using other methods), and the values of the parameters of the signal generation model are estimated from the noisy data, as detailed in Section IV and in Appendix B. Only one parameter is to be set by the user, and that is the parameter controlling the hard-thresholding in the initial denoising that is used to estimate the parameters.

The test set for the experiments includes seven different images (15th frame from "garden," "tennis," and "mobile" sequences, and the images "Barbara," "boat," "fingerprint," and "peppers"), and various noise-levels: $\sigma = 10$, 15, 20, 25, 30, 40, 50, 75 (which are equivalent to PSNR[5] of 28.12, 24.64, 22.10, 20.16, 18.60, 16.11, 14.14, and 10.61 dB, respectively), with pixel values in the range [0, 255]. The average (over all images) improvement in PSNR of the cleaned image compared to the noisy image appears in Fig. 6. The MMSE estimator outperforms the MAP estimator by about 0.5 dB on average, with this gap being fairly consistent over the different images. In order to highlight the gap between the MMSE and the MAP, Fig. 7 displays the advantage in PSNR of the MMSE estimator over the MAP estimator. The error bars in this figure indicate one standard deviation of this gap. Comparisons using the Structural Similarity Index (SSIM) [30] were also carried out, displaying similar behavior—a slight advantage for the MMSE estimator.

As discussed above, the parameter estimation relies on a setting of a single parameter—the amount of energy to remove from the signals in the crude preliminary denoising stage—and the performance of the estimators relies on the quality of the parameter estimation. In order to run a fair comparison, we varied the value of this parameter in order to optimize the average performance (over all the images in the set) of each estimator individually (i.e., one optimization for the MAP estimator, and another for the MMSE estimator).

One conclusion from these experiments is that for weak noise levels ($\sigma \leq 20$), it is beneficial to remove relatively little energy in the crude denoising stage, e.g., $T = 0.1 \cdot \sigma$, for both estimators. When working on moderate and strong noise, the best choice is $T = 1.05 \cdot \sigma$. This phenomena can be explained mostly by model mismatch, as the model we force on the signals (sparse representation over a unitary dictionary) in itself inserts some "noise" into the estimation process, and therefore the denoising performance when the noise is weak is limited.

A further analysis of the sensitivity to the setting of this parameter has been carried out. When deviating from the optimal choice, even considerably, the MMSE estimator loses at most 0.1 dB on average PSNR performance, while the MAP estimator displays a more considerable drop in performance, up to 0.5 dB. This hints that the MMSE may be more robust to errors in the parameter estimation stage, i.e., it is more robust to model mismatches.

A visual comparison of the results of the different estimators is presented in Fig. 8, for "Boat" image to which white Gaussian noise with $\sigma = 30$ has been added. The images are constructed by returning the processed patches to their original location (again, with no overlap). It is well known that increasing the overlap between the patches improves results [28], [29]. We choose to refrain from doing this, as a large overlap between patches introduces an MMSE flavor, regardless of the estimator itself, and it thus partly obscures the differences between the estimators. In order to complete the picture, the parameters estimated for this image appear in Fig. 9.

VI. Summary

In this work we discuss the problem of denoising a signal known to have a sparse representation, studying the MAP and the MMSE estimators. We focus on unitary dictionaries, for which we show that a closed-form, exact, and simple recursive formula exists for the MMSE estimator. This replaces the need for an approximation, such as the Random-OMP algorithm. We show experimentally that this exact MMSE formula outperforms the Random-OMP and the OMP (which is the exact MAP). We also discuss several numerical issues which arise when this formula is implemented in practice.

[5] $PSNR = 10\log_{10}((255^2 \cdot p)/\|\hat{\boldsymbol{X}} - \boldsymbol{X}\|_2^2)$ [dB], where $\hat{\boldsymbol{X}}$ and $\boldsymbol{X}$ are the clean and reconstructed images, respectively, and p is the number of pixels in the image.

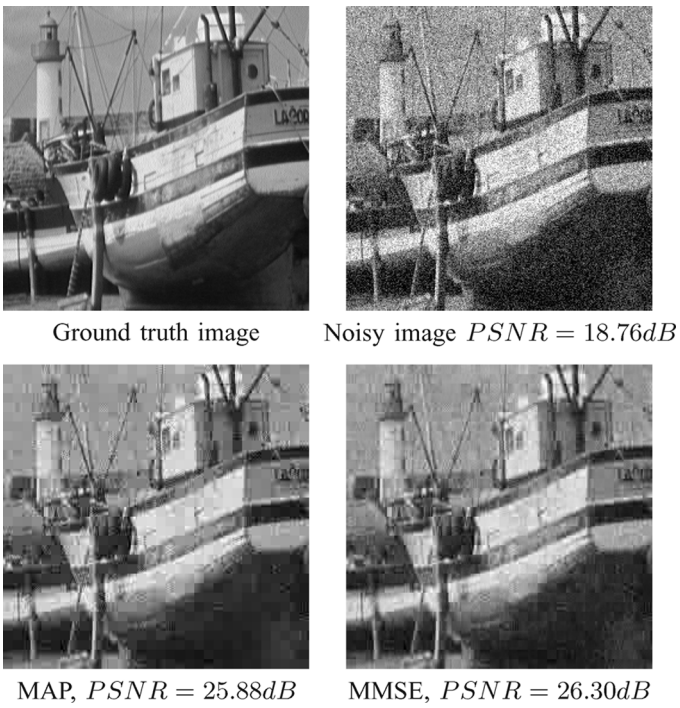

Fig. 8. Visual comparison of the reconstructed image by the MAP and MMSE estimators, for the different training options, on the center portion of the "boat" image with noise level $\sigma = 30$.

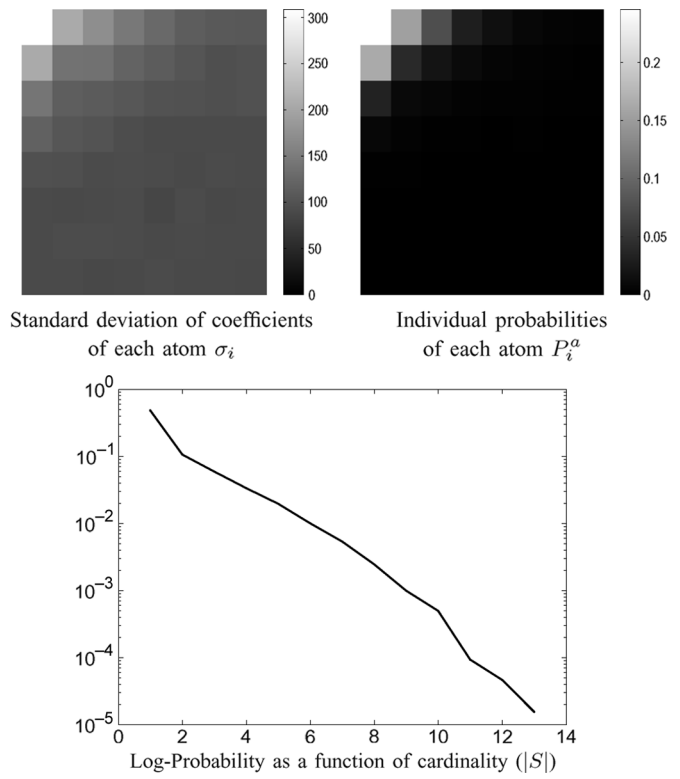

Fig. 9. Values of the estimated model parameters, using the block-coordinate-descent method described in Appendix B, for the "Boat" image with noise $\sigma = 30$. The values for σ_i and P_i^a are arranged as 8×8 arrays, corresponding to the increasing vertical and horizontal frequencies that construct the DCT dictionary. Note that the value for the top-left atom (the DC atom) is much larger than the rest, for both σ_i and P_i^a, and its value is "saturated" in both figures.

This work then extends the somewhat limited signal generation model to accommodate real-world signals. We describe how the parameters of this model are estimated, and present experiments in which the parameters are estimated from the noisy signals themselves. We show the clear advantage of the MMSE estimator over the MAP estimator in these tests, both objectively and visually.

The main drawbacks of the work presented here is the complexity of the signal generation model. This complexity leads to two difficulties. The first is that the recursive formula, while relatively efficient, still requires a large number of computations. This also limits the dimensions of the signals to work on, inducing us to work on image patches instead of a full-scale image. The second problem that arises is that the parameter estimation process, while mathematically justifiable, is still relatively weak.

We believe that future work should address different sparse signal generation models, and by doing so, find an even more efficient way to compute the MMSE, and perhaps gain a more stable estimation of the parameters involved. Another possible future direction is obtaining efficient optimal estimators for different types of risks, such as the mean over absolute errors.

APPENDIX A
RECURSIVE FORMULA—MONOTONICITY OF RATIOS PROOF

In this appendix we aim to prove the monotonicity property as described in Section III. Given that $q_j > q_i$, then

$$\frac{q_j^k}{q_i^k} < \frac{q_j^{k-1}}{q_i^{k-1}}, \qquad \text{for all} \quad k.$$

Proof: Let us start by writing out q_j^k as in (11). The tuples the jth element participates in are divided into two groups, based on whether the ith element also participates in the tuple or not

$$\begin{aligned} q_j^k &= \sum_{\{S \in \Omega_k | j \in S\}} \prod_{l \in S} q_l \\ &= \sum_{\{S \in \Omega_k | i,j \in S\}} \prod_{l \in S} q_l + q_j \cdot \sum_{\{S \in \Omega_{k-1} | i,j \notin S\}} \prod_{l \in S} q_l \\ &= A^k + q_j \cdot B^{k-1} \end{aligned} \tag{A1}$$

where we have denoted

$$A^k = \sum_{\{S \in \Omega_k | i,j \in S\}} \prod_{l \in S} q_l, \quad B^k = \sum_{\{S \in \Omega_k | i,j \notin S\}} \prod_{l \in S} q_l$$

and note that the two terms are related through $A^k = q_i \cdot q_j \cdot B^{k-2}$.

Let us analyze B^k more closely. This is in fact the sum of products over all tuples of size k from the elements of $\boldsymbol{q}$, excluding q_i and q_j. The sum of these elements, due to the normalization $\sum_{l=1}^n q_l = 1$, is $t = 1 - q_i - q_j$. Let us now denote by

$$\{r_m\}_{m=1}^{n-2}$$

all the elements of $\boldsymbol{q}$, excluding q_i and q_j, and after being multiplied by $1/t$. The multiplication leads to $\sum_{m=1}^{n-2} r_m = 1$. Substituting into B^k we obtain that

$$B^k = t^k \cdot \sum_{S\in\Omega_k} \prod_{l\in S} r_l.$$

Now, we observe that the second term is the sum of products over all k-tuples of elements from $\boldsymbol{r}$. Since $\sum_{m=1}^{n-2} r_m = 1$, this is exactly the game described in Section III-B. Therefore, we can use the normalization property to claim that

$$\sum_{S\in\Omega_k} \prod_{l\in S} r_l = k$$

and arrive at

$$B^k = t^k \cdot k \quad A^k = q_i \cdot q_j \cdot t^{k-2} \cdot (k-2) \tag{A2}$$

Going back to what we set to prove, and substitute into (A1), we now need to prove that

$$\frac{A^k + q_j \cdot B^{k-1}}{A^k + q_i \cdot B^{k-1}} < \frac{A^{k-1} + q_j \cdot B^{k-2}}{A^{k-1} + q_i \cdot B^{k-2}}.$$

Multiplying along the diagonals, removing common terms and rearranging, our new goal to prove is

$$\begin{gathered}(q_j - q_i) \cdot A^{k-1} \cdot B^{k-1} < (q_j - q_i) \cdot A^k \cdot B^{k-2} \\ A^{k-1} \cdot B^{k-1} < A^k \cdot B^{k-2}\end{gathered}$$

where the last step is valid since we know $q_j > q_i$.

Now we substitute the formulas for A^k and B^k given in (A2), changing what we need to prove the statement shown at the bottom of the page, which is a true statement, and therefore all the inequalities are true, and we have proven what we set out to prove.

Appendix B
Parameter Estimation

A. Direct Maximum-Likelihood Approach

In this appendix, we discuss more elaborately how the parameter estimation process described in Section IV-C is developed. The goal of this stage is to estimate the values of the different parameters of the signal generation model, given a set of noisy signals $\{\boldsymbol{y}^m\}_{m=1}^M$, or equivalently (due to the unitary dictionary) a set of noisy coefficients $\{\boldsymbol{\beta}^m\}_{m=1}^M$ (where $\boldsymbol{\beta}^m = \boldsymbol{D}^T\boldsymbol{y}^m$). We note again that we assume the variance σ^2 of the noise is known or was estimated by other means.

A possible approach to this problem is the ML approach. Our goal is to find a set of parameters $\boldsymbol{\Theta} = \{\{\sigma_i\}_{i=1}^n, \{P_C(k)\}_{k=1}^n, \{P_i^a\}_{i=1}^n\}$ which is the most likely to have produced this set of noisy signals. The parameters are then found by solving the following maximization problem:

$$\hat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} \prod_{m=1}^{M} P(\boldsymbol{\beta}^m|\boldsymbol{\Theta}) = \arg\max_{\boldsymbol{\Theta}} \sum_{m=1}^{M} \log\left(P(\boldsymbol{\beta}^m|\boldsymbol{\Theta})\right) \tag{B1}$$

where the second step is the maximization of the log-likelihood. The probability of a specific signal to be generated given this set of parameters is

$$P(\boldsymbol{\beta}^m|\boldsymbol{\Theta}) = \sum_{S\in\Omega} P(\boldsymbol{\beta}^m|S,\boldsymbol{\Theta}) \cdot P(S|\boldsymbol{\Theta}).$$

This probability is obtained by considering each possible support, and computing the probability that this support generated the signal, multiplied by the probability of the support to have been chosen. The sum over all possible supports is the actual probability of the signal to have been generated from the set of parameters $\boldsymbol{\Theta}$.

The probability of a signal to be generated, given a known support S, a set of parameters $\boldsymbol{\Theta}$ and with the Gaussian noise assumption is

$$P(\boldsymbol{\beta}^m|S,\boldsymbol{\Theta}) = \prod_{j\in S^m} \frac{1}{\sqrt{2\pi\left(\sigma^2+\sigma_j^2\right)}} \cdot e^{-\frac{\left(\beta_j^m\right)^2}{2\left(\sigma^2+\sigma_j^2\right)}} \tag{B2}$$

and the probability of a support to be generated given the set of parameters $\boldsymbol{\Theta}$ is given in (19)

$$P(S^m|\boldsymbol{\Theta}) = \frac{P_C\left(|S^m|\right)}{T_{|S_m|}} \cdot \prod_{i\in S^m} P_i^a$$

with $T_k = \sum_{S\in\Omega_k} \prod_{i\in S} P_i^a$ a normalization factor. Note that the probability in (B2) is computed only over the coefficients in the support. Since the support is assumed to be known, we focus only on the coefficients inside the support, and the probability of them being generated, while ignoring the coefficients outside the support (which are known to be 0).

Unfortunately, assigning those into the full ML expression in (B1) yields a highly complex argument, the maximization of which is very complicated (due to both the summation over all supports and the normalization factors T_k). Therefore, we change our course slightly and turn to a block coordinate descent approach, which may assume that the data it operates on is clean.

B. Block Coordinate Descent Approach

In the block coordinate descent approach, the denoising stage and the parameter estimation stage are carried out alternatingly, where one is estimated while the other is considered known, and

$$\begin{gathered}q_i \cdot q_j \cdot t^{k-3} \cdot (k-3) \cdot t^{k-1} \cdot (k-1) < q_i \cdot q_j \cdot t^{k-2} \cdot (k-2) \cdot t^{k-2} \cdot (k-2) \\ q_i \cdot q_j \cdot t^{2k-4} \cdot (k-3) \cdot (k-1) < q_i \cdot q_j \cdot t^{2k-4} \cdot (k-2) \cdot (k-2) \\ k^2 - 4k + 3 < k^2 - 4k + 4\end{gathered}$$

vice versa. In practice, the first stage is a simple denoising mechanism, such as hard-thresholding. An initial crude denoising stage prior to a more complex denoising mechanism is common; see for example [27].

The set of denoised signals—or in fact, the supports found—can then be used to estimate the parameters, again in an ML formulation. Then, a denoising stage can be again carried out, using the explicit signal generation model, obtaining a better result. The new denoised set can then be used to better estimate the parameters, and so on. In the experiments described above, the parameters were estimated only using the initial crude denoising.

Given the denoised signals, how can we estimate the parameters? We assume that instead of the denoised signals, we obtain the hypothesized support for each signal S^m. Inserting the known support for each signal into (B1), we no longer need to sum over all supports and we can use (B2), arriving at the following maximization problem:

$$\begin{aligned}\hat{\boldsymbol{\Theta}} &= \arg\max_{\boldsymbol{\Theta}} \sum_{m=1}^{M} \log\left(P(\boldsymbol{\beta}^m|\boldsymbol{\Theta})\right) \\ &= \arg\max_{\boldsymbol{\Theta}} \sum_{m=1}^{M} \log\left(\left(\prod_{j\in S^m} \frac{1}{\sqrt{2\pi\left(\sigma^2+\sigma_j^2\right)}} \cdot e^{-\frac{\left(\beta_j^m\right)^2}{2\left(\sigma^2+\sigma_j^2\right)}}\right)\right. \\ &\qquad \left. \cdot \frac{P_C\left(|S^m|\right)}{T_{|S^m|}} \cdot \prod_{i\in S^m} P_i^a\right) \\ &= \arg\max_{\boldsymbol{\Theta}} \sum_{m=1}^{M} \left[\sum_{j\in S^m} \log\left(\frac{1}{\sqrt{2\pi\left(\sigma^2+\sigma_j^2\right)}} \cdot e^{-\frac{\left(\beta_j^m\right)^2}{2\left(\sigma^2+\sigma_j^2\right)}}\right)\right. \\ &\qquad +\log\left(P_C\left(|S^m|\right)\right) - \log\left(T_{|S^m|}\right) \\ &\qquad \left. + \sum_{i\in S^m} \log\left(P_i^a\right)\right].\end{aligned}$$

This argument can be divided into four separate sums: the first depending only on $\{\sigma_i\}_{i=1}^n$, the second depending only on $\{P_C(k)\}_{k=1}^n$, and the third and fourth depending only on $\{P_i^a\}_{i=1}^n$. Therefore, we can separate the maximization problem into three parts, and recover each set of parameters.

The values of $\{\sigma_i\}_{i=1}^n$ are recovered by taking a derivative of the first term, and finding the zero crossing. This gives rise to n independent maximization problems. We omit this straightforward (but tedious) procedure, which eventually leads to

$$\hat{\sigma}_k = \sqrt{\frac{\sum_{\{m|k\in S^m\}} \left(\beta_k^m\right)^2}{|\{m|k\in S^m\}|} - \sigma^2}.$$

Maximizing over $\{P_C(k)\}_{k=0}^n$, we must remember the constraint that $\sum_{k=1}^n P_C(k) = 1$. For this constrained maximization, we use lagrange multipliers. Again, we jump straight to the result, being

$$\hat{P}_C(k) = \frac{|\{m||S^m| = k\}}{M}.$$

The last part of finding $\{P_i^a\}_{i=1}^n$ is more challenging compared to the first two, and a closed-form solution is not available, because of the existence of $T_{|S^m|}$ inside the formula, which requires the application of the recursive formula at each evaluation of the function. Instead, we shall try to maximize the function value using gradient ascent. In order to simplify the notation, we now denote $z_i \equiv P_i^a$, and the function to maximize is

$$F = \sum_{m=1}^{M} \left[-\log\left(T_{|S^m|}\right) + \sum_{i\in S^m} \log(z_i)\right].$$

We denote by M_k^a the number of supports containing the kth atom, and by M_k^C the number of supports of size k. Now we can rewrite this function as

$$F = -\sum_{k=1}^{n} M_k^C \cdot \log(T_k) + \sum_{k=1}^{n} M_k^a \cdot \log(z_k). \tag{B3}$$

We rewrite T_k as a function of z_i

$$T_k = \sum_{S\in\Omega_k} \prod_{l\in S} z_l = z_i \cdot \sum_{\{S\in\Omega_{k-1}|i\notin S\}} \prod_{l\in S} z_l.$$

From this last step it can be seen that only the first term depends on z_i. Taking a derivative of T_k with respect to z_i leads therefore to

$$\begin{aligned}\frac{\partial T_k}{\partial z_i} &= \sum_{\{S\in\Omega_{k-1}|i\notin S\}} \prod_{l\in S} z_l \\ &= \sum_{S\in\Omega_{k-1}} \prod_{l\in S} z_l - \sum_{\{S\in\Omega_{k-1}|i\in S\}} \prod_{l\in S} z_l \\ &= T_{k-1} - \sum_{\{S\in\Omega_{k-1}|i\in S\}} \prod_{l\in S} z_l.\end{aligned}$$

Now, we remind ourselves of the recursive formula introduced in Section III-A. This formula allows us to efficiently compute $Z_i^k = \sum_{\{S\in\Omega_k|i\in S\}} \prod_{l\in S} z_l$. Observing that $T_k = (1/k)\sum_{l=1}^N Z_l^k$, we get a simple formula for computing the function value in (B3), as well as a simple way to compute the derivative

$$\frac{\partial F}{\partial z_i} = -\sum_{k=1}^{n} M_k^C \cdot \frac{T_{k-1} - Z_i^{k-1}}{T_k} + \sum_{k=1}^{n} \frac{M_k^a}{z_i}$$

An efficient initialization for this maximization problem is $z_i = M_i^a/M$, which is the relative number of supports the ith atom appears in divided by the total number of supports. While this is not the optimal choice, it is quite near, and in both synthetic experiments (done to validate the parameter estimation process) and real-world experiments, the change of the values in the optimization problems was very mild. Since the initial denoising is inaccurate, it makes sense not to try to obtain extreme accuracy for $\{P_i^a\}_{i=1}^n$, and instead remain with the initial estimate suggested here. The synthetic and real-world experiments demonstrate that indeed, the results obtained by the two options are extremely close.

REFERENCES

[1] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, Feb. 2009.

[2] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, pp. 227–234, 1995.

[3] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–59, 2001.

[4] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[5] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[6] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *27th Asilomar Conf. Signals, Systems and Computers*, Nov. 1993.

[7] E. G. Larsson and Y. Selen, "Linear regression with a sparse parameter vector," *IEEE Trans. Signal Process.*, vol. 55, no. 2, pp. 451–460, Feb. 2007.

[8] P. Schnitter, L. C. Potter, and J. Ziniel, "Fast Bayesian matching pursuit," presented at the Workshop on Information Theory and Applications (ITA), La Jolla, CA, Jan. 2008.

[9] M. Elad and I. Yavneh, "A plurality of sparse representations is better than the sparsest one alone," *IEEE Trans. Inf. Theory*, vol. 55, no. 10, pp. 4701–4714, Oct. 2009.

[10] M. Protter, I. Yavneh, and M. Elad, "Closed-form MMSE estimator for denoising signals under sparse reconstruction modelling," in *Proc. IEEE 25th Convention of Electrical Eng. in Israel*, Eilat, Israel, Dec. 2008, pp. 580–584.

[11] M. J. Fadili, J.-L. Starck, and L. Boubchir, "Morphological diversity and sparse image denoising," in *Proc. IEEE ICASSP*, Honolulu, HI, Apr. 2007, vol. I, pp. 589–592.

[12] J. L. Starck, D. L. Donoho, and E. Candes, "Very high quality image restoration by combining wavelets and curvelets," in *Proc. Wavelet Applications in Signal and Image Processing IX*, A. Aldroubi, A. F. Laine, and M. A. Unser, Eds., 2001, vol. 4478, Proc. SPIE.

[13] A. Dalalyan and A. B. Tsybakov, "Aggregation by exponential weighting, sharp PAC-Bayesian bounds and sparsity," *Mach. Learning*, vol. 72, no. 1–2, pp. 39–61, 2008.

[14] A. Dalalyan and A. B. Tsybakov, "Sparse regression learning by aggregation and Langevin Monte-Carlo," presented at the 22nd Annu. Conf. Learning Theory (COLT 2009), Quebec, ON, Canada, Jun. 18–21, 2009 [Online]. Available: http://www.cs.mcgill.ca/~colt2009/papers/009.pdf

[15] A. Juditsky, P. Rigollet, and A. B. Tsybakov, "Learning by mirror averaging," *Ann. Stat.*, vol. 36, no. 5, pp. 2183–2206, 2008.

[16] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrica*, vol. 81, no. 3, pp. 425–455, Sep. 1994.

[17] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *J. Amer. Stat. Assoc.*, vol. 90, no. 432, pp. 1200–1224, Dec. 1995.

[18] A. Antoniadis and J. Q. Fan, "Regularization of wavelet approximations," *J. Amer. Stat. Assoc.*, vol. 96, no. 455, pp. 939–955, Sep. 2001.

[19] M. Elad, "Why simple shrinkage is still relevant for redundant representations?," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5559–5569, Dec. 2006.

[20] M. Clyde and E. I. George, "Empirical Bayes estimation in wavelet nonparametric regression," in *Bayesian Inference in Wavelet Based Models*, P. Muller and B. Vidakovic, Eds. New York: Springer-Verlag, 1998, vol. 141, Lecture Notes in Statistics, pp. 309–322.

[21] M. Clyde and E. I. George, "Flexible empirical Bayes estimation for wavelets," *J. R. Stat. Soc. B*, vol. 62, pp. 681–698, 2000.

[22] M. Clyde, G. Parmigiani, and B. Vidakovic, "Multiple shrinkage and subset selection in wavelets," *Biometrika*, vol. 85, pp. 391–401, 1998.

[23] F. Abramovich, T. Sapatinas, and B. W. Silverman, "Wavelet thresholding via a Bayesian approach," *J. R. Stat. Soc. B*, vol. 60, pp. 725–749, 1998.

[24] A. Antoniadis, J. Bigot, and T. Sapatinas, "Wavelet estimators in nonparametric regression: A comparative simulation study," *J. Stat. Softw.*, vol. 6, no. 6, 2001.

[25] P. Moulin and J. Liu, "Analysis of multiresolution image denoising schemes using generalized-Gaussian priors," *IEEE Trans. Inf. Theory*, vol. 45, no. 3, pp. 909–919, Apr. 1999.

[26] M. N. Do and M. Vetterli, "Wavelet-based texture retreival using generalized Gaussian density and Kullback–Leibler distance," *IEEE Trans. Image Process.*, vol. 11, no. 2, pp. 146–158, Feb. 2002.

[27] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3-D transform-domain collaborative filtering," presented at the Eur. Signal Process. Conf. (EUSIPCO-2007), Poznan, Poland, Sep. 2007.

[28] M. Aharon, M. Elad, and A. M. Bruckstein, "On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them," *J. Linear Alg. Appl.*, vol. 416, pp. 48–67, Jul. 2006.

[29] M. Aharon, M. Elad, and A. M. Bruckstein, "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[30] Z. Wang, A. C. Bovik, H. R. Sheikk, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

Matan Protter received the B.Sc. degree in mathematics, physics, and computer sciences from the Hebrew University of Jerusalem, Jerusalem, Israel, in 2003. He is currently pursuing the Ph.D. degree in computer sciences at the Computer Sciences Department, The Technion—Israel Institute of Technology, Haifa, Israel.

From 2003–2009, he has concurrently served in the Israeli Air Force, as a Senior R&D Engineer. His research interests are in the area of image processing, focusing on example-based image models and their application to various inverse problems.

Mr. Protter is the recipient of the Jacobs-Qualcomm Fellowship in 2010 and of the Wolf Foundation Research Students Excellence prize in 2010.

Irad Yavneh received the Ph.D. degree in applied mathematics from the Weizmann Institute of Science, Rehovot, Israel, in 1991.

He is currently a Professor in the faculty of computer science at the Technion—Israel Institute of Technology, Haifa, Israel. His research interests include multiscale computational techniques, scientific computing and computational physics, image processing and analysis, and geophysical fluid dynamics.

Michael Elad (M'98–SM'08) received the B.Sc. (1986), M.Sc., and D.Sc. degrees from the Technion—Israel Institute of Technology, Haifa, Israel.

From 1988–1993 he served in the Israeli Air Force. From 1997–2000 he worked at Hewlett-Packard Laboratories Israel as an R&D Engineer. During 2000–2001 he headed the research division at Jigami Corporation, Israel. During the years 2001–2003 he spent a postdoctoral period at Stanford University. Since late 2003 he has been a faculty member in computer science at the Technion. On May 2007 he was tenured to an Associate Professorship. He works currently in the field of signal and image processing, specializing in particular on inverse problems, sparse representations, and overcomplete transforms.

Prof. Elad received the Technion's best lecturer award six times. He is the recipient of the Solomon Simon Mani Award for Excellence in Teaching in 2007, and he is the recipient of the Henri Taub Prize for Academic Excellence (2008). He currently is serving as an Associate Editor for both the IEEE TRANSACTIONS ON IMAGE PROCESSING and the *SIAM Journal on Imaging Sciences*.