MMSE Approximation For Sparse Coding Algorithms Using Stochastic Resonance

Dror Simon, Jeremias Sulam, Yaniv Romano, Yue M. Lu and Michael Elad

***Abstract*—Sparse coding refers to the pursuit of the sparsest representation of a signal in a typically overcomplete dictionary. From a Bayesian perspective, sparse coding provides a Maximum a Posteriori (MAP) estimate of the unknown vector under a sparse prior. In this work, we suggest enhancing the performance of sparse coding algorithms by a deliberate and controlled contamination of the input with random noise, a phenomenon known as stochastic resonance. The proposed method adds controlled noise to the input and estimates a sparse representation from the perturbed signal. A set of such solutions is then obtained by projecting the original input signal onto the recovered set of supports. We present two variants of the described method, which differ in their final step. The first is a provably convergent approximation to the Minimum Mean Square Error (MMSE) estimator, relying on the generative model and applying a weighted average over the recovered solutions. The second is a relaxed variant of the former that simply applies an empirical mean. We show that both methods provide a computationally efficient approximation to the MMSE estimator, which is typically intractable to compute. We demonstrate our findings empirically and provide a theoretical analysis of our method under several different cases.**

***Index Terms*—Sparse coding, stochastic resonance, basis pursuit, orthogonal matching pursuit, MMSE estimation**

I. Introduction

In signal processing, often times we have access to a corrupted signal and we wish to estimate its clean version. This process includes a wide variety of problems, such as denoising, where we wish to remove noise from a noisy signal; deblurring where we look to sharpen an image that has been blurred or was taken out of focus; and inpainting in which we fill-in missing data that have been removed from the image. All the aforementioned tasks, and many others, include a linear degradation operator and a stochastic corruption. The forward model can be described by $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\nu}$, where $\boldsymbol{x}$ is the clean signal, $\boldsymbol{H}$ is the linear degradation operator, $\boldsymbol{\nu}$ denotes additive noise, and $\boldsymbol{y}$ stands for the noisy measurements.

In order to provide a *good estimate* of $\boldsymbol{x}$, it is useful to incorporate both the statistical properties of the corruption as well as prior knowledge on the signal. In image processing in particular, many such priors have been developed over the years, such as total-variation, self-similarity, sparsity, and many others [1–3].

In this work we focus our attention on the sparse model prior, which assumes that the clean signal $\boldsymbol{x} \in \mathbb{R}^n$ is a linear combination of a small number of columns from an overcomplete dictionary $\boldsymbol{D} \in \mathbb{R}^{n\times m}$, where $n < m$, referred to as *atoms*. In this case, we can write $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}$, where the representation vector $\boldsymbol{\alpha} \in \mathbb{R}^m$ is sparse. One of the most fundamental problems in this model is termed *sparse coding*: Given $\boldsymbol{x}$, find the sparsest $\boldsymbol{\alpha}$ such that $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}$. Formally, this calls for solving

$$(P_0): \quad \hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \ ||\boldsymbol{\alpha}||_0 \ \text{s.t.} \ \boldsymbol{D}\boldsymbol{\alpha} = \boldsymbol{x},$$

where $\|\cdot\|_0$ stands for the l_0 pseudo-norm that counts the number of non-zero elements in the vector.

Returning to the real measurements setting, and in particular for a denoising task, the degradation is simply given by additive noise $\boldsymbol{\nu}$, typically assumed Gaussian or with bounded energy $\|\boldsymbol{\nu}\|_2 \leq \epsilon$, resulting in $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\nu}$. Hence, the above problem is naturally modified to

$$(P_0^\epsilon): \quad \hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \ ||\boldsymbol{\alpha}||_0 \ \text{s.t.} \ ||\boldsymbol{D}\boldsymbol{\alpha} - \boldsymbol{y}||_2 \leq \epsilon.$$

The solution of (P_0^ϵ) can be then used to provide an estimate of $\boldsymbol{x}$, in the form of $\hat{\boldsymbol{x}} = \boldsymbol{D}\hat{\boldsymbol{\alpha}}$.

Without further assumptions, (P_0^ϵ) is non-convex and NP-Hard in general [4], as it requires searching through all the possible supports of $\boldsymbol{\alpha}$. Nonetheless, different approximation or pursuit algorithms have been developed in order to manage this task effectively. Some of these include greedy strategies, such as the *Orthogonal Matching Pursuit* (OMP) [5], or relaxation alternatives, like *Basis-Pursuit* (BP) [6].

These approximation algorithms have been accompanied by theoretical guarantees for finding a sparse representation $\hat{\boldsymbol{\alpha}}$ that is close to the original representation, e.g. in an ℓ_2 sense, $\|\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\|_2$. Additionally, they often assure the correct recovery of the support [7]. Such results rely on the cardinality of $\boldsymbol{\alpha}$, the range of the non-zero values and properties of the dictionary $\boldsymbol{D}$. These algorithms succeed not only in cases of noise with bounded energy, but also accurate solutions with high probability in more general settings [8, 9].

From a Bayesian point of view, pursuit algorithms provide an approximation to a Maximum a Posteriori (MAP) estimator [14, 15]. Indeed, the objective seeks the most likely signal under the sparse prior, subject to the noise deviation. Such an estimator does not coincide with the Minimum Mean Squared Error (MMSE) estimate, which is of great interest in many cases. Unfortunately, however, exact MMSE estimation has been shown to be computationally intractable and unfeasible in practice [14, 15].

In this paper, we provide an efficient way to approximate the MMSE by means of contaminating the input signal with a controlled amount of (further) noise. The proposed method

D. Simon and M. Elad are with the Department of Computer Science at the Technion, Israel.

J. Sulam is with the Biomedical Engineering Department and the Mathematical Institute of Data Science of Johns Hopkins University, USA.

Y. Romano is with the Statistics Department of Stanford University, USA.

Y. M. Lu is with the John A. Paulson School of Engineering and Applied Sciences, Harvard University, USA.

leans on the *Stochastic Resonance* (SR) phenomenon, in which the addition of noise to a weak signal can increase its output Signal-to-Noise Ratio (SNR) in the context of a non-linear transfer function. This field has been broadly developed to improve the performance of sub-optimal detectors [10], non-linear parametric estimators [11] and image processing algorithms [12]. As we will careful comment later, our method shares similarities with the Supra-threshold SR (SSR) algorithm [13] while providing a generalization thereof. After providing a provably convergent algorithm, we will explore an alternative relaxation that will enable the deployment of this idea to general pursuit methods and in more practical scenarios. The proposed approach provides a general tool that can improve performance of different sparse coding algorithms, both in synthetic and real data settings, as we demonstrate numerically.

The rest of the paper is structured as follows. Section II explores and comments on related previous work, and Section III reviews Bayesian estimation under the sparse prior. Then, in Section IV we present and analyze our provably convergent MMSE approximation algorithm, followed by a practical variation in Section V. We study the properties of the general algorithm under specific cases in Sections VI and VII. In Section VIII we explore the application of different SR noise distributions, before showcasing our proposed approach for image denoising in Section IX. Finally, we conclude our work in Section X.

II. Related Work

Providing solutions to inverse problems with minimal MSE has remained a problem of great interest for many years. In the context of sparse modeling in particular, the work in [14] suggested an MMSE approximation in terms of the *Random-OMP* (RandOMP) algorithm. This approach consists in running a stochastic variant of OMP several times, introducing some randomness in the choice of the supports at every iteration, and finally averaging the results. RandOMP was shown to coincide with the MMSE estimator under a unitary dictionary assumption, or when $\boldsymbol{D}$ is overcomplete and the cardinality of the representations is restricted to one atom. RandOMP also improves the MSE empirically in more general cases, where the MMSE cannot be practically computed. On the other hand, this approach inherits the limitation of OMP, and specifically the gradual increase of the support one element at a time. As a result, this method is impractical in cases where the cardinality of the solution is in the order of hundreds and beyond – as for many real world signals.

In [15] a pursuit based on a Bayesian approach was suggested. The *Fast Bayesian Matching Pursuit* (FBMP) is a method that seeks the most probable supports and then approximates their posterior probabilities in order to provide an estimate of the MMSE. FBMP has been developed under a specific sparse prior model and relies on it in order to properly compute its estimate. This limits this approach to cases that follow the assumed signal model and less applicable to real applications where the prior is unknown.

A closely related method to our work is that of Supra-threshold Stochastic Resonance, first described in [13]. SSR consists in the addition of noise to a signal before it is passed through a set of thresholds, or other analytic non-linearities [16]. All outputs are then averaged to obtain a final result. Just as in SR, the amount of noise to be added is a parameter that needs careful tuning, and it can be set to maximize some statistical measure (e.g. SNR, Mutual Information, among others). On the other hand, SSR is usually motivated by a fixed physical system (e.g. sensory neurons [17]) where one has only limited control over the input signal.

As we will explore in the following sections, our work is inspired by SSR and it can be understood as a generalization of it. In particular, we will regard pursuit algorithms as more general non-linear functions, moving beyond simple thresholding rules. Moreover, the output of our algorithm will subtract some of the effect of the added noise, providing a better estimate, asymptotically converging to the MMSE estimator. Finally, the proposed approach will be general, making it possible to consider convex relaxation alternatives, this way making MMSE approximation plausible for large dimensional signals. Before moving to the presentation of the main algorithm, however, we review some general results in Bayesian sparse estimation in the following Section.

III. Bayesian Estimation Under the Sparse Prior

Let us formulate our model in more details. The matrix $\boldsymbol{D} \in \mathbb{R}^{n\times m}$ is an overcomplete dictionary, while the representation $\boldsymbol{\alpha} \in \mathbb{R}^m$ is a sparse vector with either fixed cardinality $\|\boldsymbol{\alpha}\|_0 = M$ or with a prior probability p_i for each entry to be non-zero – we will use both alternatives in this work. The generative model consists of first drawing a support S from a distribution P_S. We denote the set of all probable supports by $\Omega = \{S|P_S(S) > 0\}$. The non-zero elements in $\boldsymbol{\alpha}$, denoted as $\boldsymbol{\alpha}_S$, are then drawn from a distribution P_{α_S}, which is assumed to be a white Gaussian distribution $\mathcal{N}(\boldsymbol{0}, \sigma_\alpha^2 \boldsymbol{I})$. A signal $\boldsymbol{x}$ is constructed by a linear combination of atoms $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}$, and the measured samples are given by $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\nu}$, where $\boldsymbol{\nu} \sim \mathcal{N}(\boldsymbol{0}, \sigma_\nu^2 \boldsymbol{I})$ is additive Gaussian noise. Under such a generative model, a Bayesian formulation for estimating $\boldsymbol{\alpha}$ deploys the prior on the representations $\boldsymbol{\alpha}$ in different ways. We now describe different Bayesian estimators that can be formulated in this context, as described in [18].

We begin with the Oracle estimator, which seeks to estimate the clean signal when the support is assumed to be known. This is a simplistic assumption, as retrieving the correct support S of the original sparse representation $\boldsymbol{\alpha}$ is the essence of the combinatorial (P_0^ϵ) problem. In such a case, the MMSE estimator is simply the conditional expectation $\hat{\boldsymbol{\alpha}}_S^{\text{Oracle}} = \mathbb{E}\left[\boldsymbol{\alpha}|\boldsymbol{y}, S\right]$, given by

$$\hat{\boldsymbol{\alpha}}_S^{\text{Oracle}}(\boldsymbol{y}) = \frac{1}{\sigma_\nu^2}\boldsymbol{Q}_S^{-1}\boldsymbol{D}_S^T\boldsymbol{y}, \tag{1}$$

where $\boldsymbol{D}_S$ is the sub-dictionary containing only the atoms indexed by S, and $\boldsymbol{Q}_S$ is given by

$$\boldsymbol{Q}_S = \frac{1}{\sigma_\alpha^2}\boldsymbol{I}_{|S|} + \frac{1}{\sigma_\nu^2}\boldsymbol{D}_S^T\boldsymbol{D}_S.$$

We refer to this estimator as the *Oracle*, as there is no possible way of knowing the true support beforehand.

Next is the MAP estimator, which searches for the most probable support $\hat{S}$ given the measurements and uses it to estimate the signal[1]. The relevant posterior probability for this estimation is given by [18]

$$P(S|y) = \frac{t_S}{t}, \qquad t \triangleq \sum_{S\in\Omega} t_S \tag{2}$$

where

$$t_S \triangleq \frac{1}{\sqrt{\det(\boldsymbol{C}_S)}} \exp\left\{-\frac{1}{2}\boldsymbol{y}^T\boldsymbol{C}_S^{-1}\boldsymbol{y}\right\} \prod_{i\in S} p_i \prod_{i\notin S}(1-p_i)\,,$$

and $\boldsymbol{C}_S^{-1} = \frac{1}{\sigma_\nu^2}\boldsymbol{I}_n - \frac{1}{\sigma_v^4}\boldsymbol{D}_S\boldsymbol{Q}_S^{-1}\boldsymbol{D}_S^T$.

The MAP estimator is obtained by maximizing $P(S|\boldsymbol{y})$ with respect to S, which (employing Bayes' rule) can be written as

$$\hat{S} = \arg\max_S P(S|\boldsymbol{y}) = \arg\max_S P(\boldsymbol{y}|S)P(S).$$

By replacing the conditional probability and the prior on the support, one can show [18] that this corresponds to

$$\hat{S} = \arg\max_S \frac{1}{2}\left|\left|\frac{1}{\sigma_v^2}\boldsymbol{Q}_S^{-\frac{1}{2}}\boldsymbol{D}_S^T\boldsymbol{y}\right|\right|_2^2 - \frac{1}{2}\log(\det(\boldsymbol{C}_S)) + \sum_{i\in S}\log(p_i) + \sum_{j\notin S}\log(1-p_j).$$

In the case where the cardinality of $\boldsymbol{\alpha}$ is constant, $\|\boldsymbol{\alpha}\|_0 = M$, and all supports are equally likely, the last two terms above can be omitted. Once the most probable support is recovered, the oracle formula can then be employed to estimate the corresponding coefficients as

$$\hat{\boldsymbol{\alpha}}^{\text{MAP}}(\boldsymbol{y}) = \hat{\boldsymbol{\alpha}}^{\text{Oracle}}_{\hat{S}^{\text{MAP}}}(\boldsymbol{y}).$$

The last estimator we discuss here is the MMSE, which is given by the conditional expectation,

$$\hat{\boldsymbol{\alpha}}^{\text{MMSE}}(\boldsymbol{y}) = \mathbb{E}\left[\boldsymbol{\alpha}|\boldsymbol{y}\right] = \sum_{S\in\Omega} P(S|\boldsymbol{y})\mathbb{E}\left[\boldsymbol{\alpha}|\boldsymbol{y},S\right].$$

This is a weighted sum over all the possible supports. Moreover, each element calls for the oracle estimator over the candidate support, allowing us to write

$$\hat{\boldsymbol{\alpha}}^{\text{MMSE}}(\boldsymbol{y}) = \sum_{S\in\Omega} P(S|\boldsymbol{y})\hat{\boldsymbol{\alpha}}^{\text{Oracle}}_S(\boldsymbol{y}). \tag{3}$$

Because of this massive averaging of all the possible supports, somewhat surprisingly, the MMSE under a sparse prior is actually a dense vector.

Both estimators, $\hat{\boldsymbol{\alpha}}^{\text{MMSE}}$ and $\hat{\boldsymbol{\alpha}}^{\text{MAP}}$, are NP hard to obtain in general, as they require either a sum over all the possible supports or the computation of all posterior probabilities and selecting the highest one. Either option is prohibitive as the number of possible supports is exponentially large. This is the reason for approximation algorithms or pursuits, which typically attempt to approximate the MAP estimate. Nonetheless, as noted in [15], the posterior probabilities $P(S|\boldsymbol{y})$ have an exponential nature – see Equation (2), and thus the sum in Equation (3) is practically dominated by a much smaller number of terms. Put formally, this suggests that there exists a subset of the supports, $\omega \subset \Omega$, $|\omega| \ll |\Omega|$ such that $P(S_\omega|\boldsymbol{y}) \gg P(S_{\Omega\setminus\omega}|\boldsymbol{y})$ where $S_\omega \in \omega, S_{\Omega\setminus\omega} \in \Omega\setminus\omega$. If one could obtain these most significant elements and their proper weights, then an MMSE approximation would be attainable. This is the rationale in earlier work on MMSE approximations [14, 15], and the motivation behind our proposed approach.

[1]In fact, this is the MAP of the support. We use it to avoid the probable case where the recovered signal is the $\mathbf{0}$ vector as described in [18].

Algorithm 1 Prior-based SR algorithm

procedure PRIORBASED-SR($\boldsymbol{y}$, $\boldsymbol{D}$, PursuitAlg, σ_n, K)
  $\boldsymbol{\mathcal{S}} \leftarrow \Phi$
  for k=1...K **do**
    $\boldsymbol{n}_k \leftarrow$ SampleNoise(σ_n)
    $\tilde{\boldsymbol{\alpha}}_k \leftarrow$ PursuitAlg($\boldsymbol{y}+\boldsymbol{n}_k, \boldsymbol{D}$)
    $\hat{S}_k \leftarrow$ Support($\tilde{\boldsymbol{\alpha}}_k$)
    $\boldsymbol{\mathcal{S}} \leftarrow \boldsymbol{\mathcal{S}} \cup \{\hat{S}_k\}$
  end for
  $\hat{\boldsymbol{\alpha}} \leftarrow \frac{\sum_{S\in\boldsymbol{\mathcal{S}}} P(\boldsymbol{y}|S)P(S)\hat{\boldsymbol{\alpha}}^{\text{Oracle}}_S(\boldsymbol{y})}{\sum_{S\in\boldsymbol{\mathcal{S}}} P(\boldsymbol{y}|S)P(S)}$
  return $\hat{\boldsymbol{\alpha}}$
end procedure

IV. The Proposed Algorithm

We now present the proposed approach in Algorithm 1. This algorithm consists of K iterations, where in each a small amount of noise $\boldsymbol{n}_k$ is added to the already noisy signal $\boldsymbol{y}$, and a (greedy or relaxation-based) pursuit algorithm is employed. The final estimation is computed as a weighted average over the obtained supports: For each recovered support, the corresponding oracle estimator is obtained w.r.t. the original measurements $\boldsymbol{y}$ (i.e., without the influence of $\boldsymbol{n}_k$) and weighed according to its un-normalized posterior probability.

As we show next, Algorithm 1 asymptotically converges to the MMSE estimator. Before presenting this result, however, we present in the following lemma an alternative expression for the MMSE estimator that will prove useful later on.

Lemma 1. *Let $\Omega = \{S|P(S)>0\}$ be the set of all the possible supports, then*

$$\hat{\boldsymbol{\alpha}}^{MMSE}(\boldsymbol{y}) = \frac{\sum_{S\in\Omega} P(\boldsymbol{y}|S)P(S)\hat{\boldsymbol{\alpha}}^{Oracle}_S(\boldsymbol{y})}{\sum_{S\in\Omega} P(\boldsymbol{y}|S)P(S)}.$$

Proof. From Bayes' theorem and the law of total probability, we can write

$$P(S|\boldsymbol{y}) = \frac{P(\boldsymbol{y}|S)P(S)}{P(\boldsymbol{y})} = \frac{P(\boldsymbol{y}|S)P(S)}{\sum_{S\in\Omega} P(\boldsymbol{y}|S)P(S)}.$$

Combining this with Equation (3), we have:

$$\hat{\boldsymbol{\alpha}}^{\text{MMSE}}(\boldsymbol{y}) = \frac{\sum_{S\in\Omega} P(\boldsymbol{y}|S)P(S)\hat{\boldsymbol{\alpha}}^{\text{Oracle}}_S(\boldsymbol{y})}{\sum_{S\in\Omega} P(\boldsymbol{y}|S)P(S)}.$$

□

The result that we present below is general, in the sense that it is applicable to any pursuit that provides a stable approximation of $\hat{\boldsymbol{\alpha}}$. For this reason, we now formalize this in the following definition, followed by the statement of the main theorem.

Definition 1. A pursuit method PM is a *stable pursuit* w.r.t. the prior P_S, $P_{\boldsymbol{\alpha}_S}$ and the dictionary $\boldsymbol{D}$ if $\forall\ S \in \Omega$ and a respective $\boldsymbol{\alpha} \in \mathbb{R}^m$ such that $\text{Support}(\boldsymbol{\alpha}) = S$, and $P_{\boldsymbol{\alpha}_S}(\boldsymbol{\alpha}_S) > 0$, then $\exists \epsilon > 0$ such that $\forall \boldsymbol{v} \in \{\boldsymbol{v} \in \mathbb{R}^n | \|\boldsymbol{v} - \boldsymbol{D}\boldsymbol{\alpha}\|_2 < \epsilon\}$ the pursuit method PM recovers the correct support, i.e. $\text{Support}\{PM(\boldsymbol{v})\} = S$.

Theorem 2. *Let $\boldsymbol{n}_k \sim \mathcal{N}\left(\boldsymbol{0}, \sigma_n^2 \boldsymbol{I}\right)$ be a white Gaussian SR noise, $\sigma_n > 0$ and PM a stable pursuit method. Then, as $K \to \infty$, Algorithm 1 asymptotically converges to the MMSE estimator with probability* 1.

Proof. Assume by contradiction that Algorithm 1 does not converge to the MMSE. From Lemma 1 this means that $\Omega \setminus \boldsymbol{\mathcal{S}} \neq \Phi$, where $\boldsymbol{\mathcal{S}}$ are the gathered supports by Algorithm 1. Let S_i be a support such that $S_i \in \Omega \setminus \boldsymbol{\mathcal{S}}$, and let $\boldsymbol{\alpha}_i \in \mathbb{R}^m$ be such that $\text{Support}\{\boldsymbol{\alpha}_i\} = S_i$ and $P_{\boldsymbol{\alpha}_S}(\boldsymbol{\alpha}_{S_i}) > 0$. Since the pursuit method is stable, $\exists \epsilon > 0$ such that $\forall \boldsymbol{v} \in \{\boldsymbol{v} \in \mathbb{R}^n | \|\boldsymbol{v} - \boldsymbol{D}\boldsymbol{\alpha}_i\|_2 < \epsilon\}$ and $\text{Support}\{PM(\boldsymbol{v})\} = S_i$.

In each iteration, the algorithm sparse codes $\boldsymbol{y} + \boldsymbol{n}_k$ and since $\boldsymbol{n}_k$ is Gaussian $P(\|\boldsymbol{y} + \boldsymbol{n}_k - \boldsymbol{D}\boldsymbol{\alpha}\|_2 < \epsilon) > 0$ for any $\boldsymbol{\alpha} \in \mathbb{R}^m$. Hence, as $K \to \infty$, $\exists k_i$ such that $\|\boldsymbol{y} + \boldsymbol{n}_{k_i} - \boldsymbol{D}\boldsymbol{\alpha}_i\|_2 < \epsilon$ and from the stability of the pursuit $\text{Support}\{PM(\boldsymbol{y} + \boldsymbol{n}_{k_i})\} = S_i$. Therefore, at the k_i-th iteration, the support S_i is added to the accumulated set $\boldsymbol{\mathcal{S}}$, contradicting the false assumption. □

A few remarks are in place. First, if $K \geq |\Omega|$, using Algorithm 1 is clearly ineffective since one may simply compute the MMSE from (3). Nevertheless, fast convergence occurs when $\boldsymbol{\mathcal{S}}$ contains the most likely supports, since their weight is much larger than the weight of the other elements [15]. Using the MAP estimator – or its approximation in terms of pursuit algorithms – promotes highly likely supports more often than less likely ones. The SR idea provides a way of accumulating a set of highly probable supports by perturbing the measurements before running the pursuit.

Second, the result above is somewhat limited as it does not inform about *how fast* $\hat{\boldsymbol{\alpha}}$ converges to $\hat{\boldsymbol{\alpha}}^{\text{MMSE}}$. Such an analysis must depend on the energy of the added noise. Indeed, when σ_n is too large, the SR measurements $\boldsymbol{y} + \boldsymbol{n}_k$ significantly deviate from $\boldsymbol{y}$, reducing the chances to retrieve probable supports. On the other hand, when the added noise is weak, the signal $\boldsymbol{y} + \boldsymbol{n}_k$ hardly varies, reducing the chances to cover the set of supports quickly. The analysis of this convergence rate is challenging, and we defer it to future work. As we will extensively corroborate numerically, however, a relatively low number of iterations K suffices to provide a good approximation to the MMSE in practice.

To empirically examine the performance of Algorithm 1, we performed the following experiments. We drew a random dictionary $\boldsymbol{D} \in \mathbb{R}^{50 \times 100}$ and normalized its columns. Then, we generated sparse vectors containing a single non-zero element at a random location, where its value was sampled from the normal distribution $\mathcal{N}(0, 1)$. Signals were then created by multiplying the sparse vectors with the dictionary $\boldsymbol{D}$. Finally, we added a Gaussian noise $\boldsymbol{\nu} \sim \mathcal{N}(\boldsymbol{0}, \sigma_\nu^2 \boldsymbol{I})$, $\sigma_\nu = 0.2$ to the signals resulting with noisy measurements. We then used

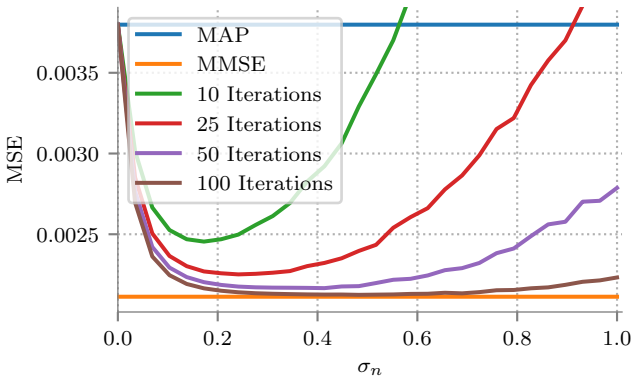


Fig. 1: PriorBased-SR for various K and σ_n values. Sparse vector cardinality $\|\boldsymbol{\alpha}\|_0 = 1$.

Algorithm 1 to estimate the clean signals and compared its MSE to the MAP and the MMSE estimators for a varying number of iterations K and standard deviation σ_n. Note that in this case, the OMP algorithm is also the MAP, and therefore we used it as our pursuit method. The results averaged over $10,000$ realizations can be seen in Figure 1. Note that in this experiment, the number of possible supports is $|\Omega| = 100$. When $K = 100$ and $\sigma_n \approx 0.5$, the MSE of Algorithm 1 almost reaches that of the MMSE estimator, suggesting that only a few improbable supports are missing from the accumulated set $\boldsymbol{\mathcal{S}}$. That said, even when K is much smaller than 100, for a reasonable amount of SR noise $\sigma_n \approx \sigma_\nu$, the MSE of Algorithm 1 is close to that of the MMSE estimator.

To further demonstrate the efficiency of this technique, we repeat the same protocol, only this time the number of non-zero elements in the sparse vector is 3, increasing the number of possible supports to $|\Omega| = \binom{100}{3} = 161,700$, all apriori equally likely. The results can be seen in Figure 2. Note that while the MAP and the MMSE estimators require iterating through all the possible supports, Algorithm 1 efficiently retrieve the most significant ones, leading to superior denoising results over the standard OMP and the MAP estimator.

These results show that not only does the algorithm asymptotically converge to the MMSE, but also a significant improvement can be achieved over the MAP estimator (or its approximation) even with a relatively small number of iterations. A limitation of this approach, however, is that this method requires full knowledge of the generative model and its parameters, similar to the FBMP algorithm, thus limiting its use in real settings. In the following sections we suggest a practical variation of the presented algorithm that can be used in more general cases.

V. A Practical Variant

We present the practical variant of Algorithm 1 in Algorithm 2. Note that we split the algorithm into two cases. In the first we assume that even though we have no knowledge regarding the probability mass function of the support $P(S)$, we do know the probability density function of the non-zero elements given their indices and the measurements, making the oracle estimator obtainable. The second case only assumes

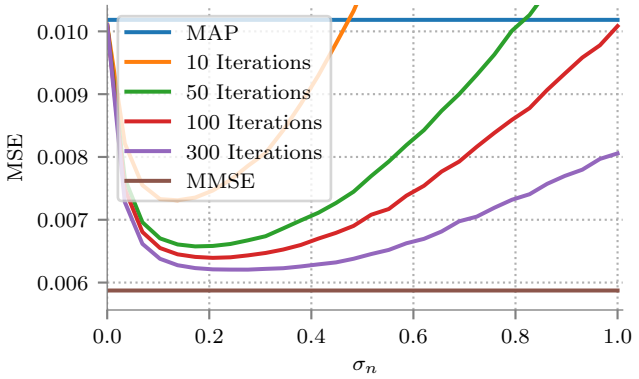


Fig. 2: PriorBased-SR for various K and σ_n values. Sparse vector cardinality $\|\boldsymbol{\alpha}\|_0 = 3$.

Algorithm 2 General SR algorithm

procedure GENERAL-SR($\boldsymbol{y}, \boldsymbol{D}$, PursuitAlg, σ_n, K)
  for k=1...K **do**
    $\boldsymbol{n}_k \leftarrow$ SampleNoise(σ_n)
    $\tilde{\boldsymbol{\alpha}}_k \leftarrow$ PursuitAlg($\boldsymbol{y} + \boldsymbol{n}_k, \boldsymbol{D}$)
    $\hat{S}_k \leftarrow$ Support($\tilde{\boldsymbol{\alpha}}_k$)
    if $P(\boldsymbol{\alpha}|\boldsymbol{y}, S)$ is known **then**
      $\hat{\boldsymbol{\alpha}}_k \leftarrow \hat{\boldsymbol{\alpha}}^{\text{Oracle}}_{\hat{S}_k}(\boldsymbol{y})$
    else
      $\hat{\boldsymbol{\alpha}}_k \leftarrow \left(\boldsymbol{D}_S^T \boldsymbol{D}_S\right)^{-1} \boldsymbol{D}_S^T \boldsymbol{y}$
    end if
  end for
  $\hat{\boldsymbol{\alpha}} = \frac{1}{K} \sum_{k=1}^{K} \hat{\boldsymbol{\alpha}}_k$
  return $\hat{\boldsymbol{\alpha}}$
end procedure

knowledge regarding the dictionary $\boldsymbol{D}$, replacing the oracle with a simple Least Squares (LS) operation.

Computing the MMSE estimator in Equation (3) requires a complete knowledge regarding the generative model. Therefore, in cases where the prior is only partially known, the MMSE estimator cannot be obtained, and achieving its MSE performance is generally not feasible. Nevertheless, as we empirically show here and in the following sections, Algorithm 2 succeeds to effectively approximate the MMSE estimator.

Before we go through the analytic arguments and empirical evidence provided in the coming sections, we present some intuition behind the proposed method. Similar to the previous method, the SR noise added will introduce a small perturbation in the signal $\boldsymbol{y} + \boldsymbol{n}_k$ in each iteration. Therefore, in each iteration the pursuit will extract supports that are likely to fit the original signal $\boldsymbol{x}$. The final estimator is an arithmetic mean of all the recovered supports, meaning that the supports that have higher posterior probability will be retrieved more often, making their weight greater than other, less probable supports. If K is large enough and the occurrence of each support resembles its un-normalized posterior probability, then the averaged result coincides with the MMSE.

We propose the following experiments in order to demonstrate Algorithm 2's performance. As in the previous section, we use a normalized random Gaussian dictionary, this time of size 25×50, and generate random sparse vectors with 3 non-zero elements and Gaussian coefficients. We multiply the sparse vectors by the dictionary and add a Gaussian noise to the signals. To obtain clean estimates we use Algorithm 2, once with BP and once with OMP. Since we assume no knowledge regarding the prior probability of the support of the sparse vector, we use the bounded noise formulation of the pursuit algorithms, i.e.

$$\begin{aligned} &\text{(OMP)} \quad \min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad \text{s.t.} \quad \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{\alpha}\|_2 \leq \epsilon, \\ &\text{(BP)} \quad \min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \quad \text{s.t.} \quad \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{\alpha}\|_2 \leq \epsilon. \end{aligned}$$

where ϵ is chosen to be optimal. We repeat the experiment twice. Once we assume we know the distribution of $\boldsymbol{\alpha}_S|S, \boldsymbol{y}$ allowing us to use the oracle in Equation (1), and once using the plain LS variant. We compare the results to those obtained by Algorithm 1 with OMP used as a pursuit, as well as to the MMSE and MAP estimators.

In a second experiment we further examine the effectiveness of our method compared to standard pursuit algorithms for a varying number non-zero elements. For this experiment, we increased the dimensions of the dictionary to 50×100 to allow for a large number of non-zero elements in $\boldsymbol{\alpha}$, while keeping it sparse. In all the described experiments we use 300 iterations for both algorithms and average over $10,000$ realizations. The results can be seen in Figure 3 and 4 respectively.

Examining the results obtained in Figure 4, our LS variant improves over standard pursuit algorithms for various cardinality values. Furthermore, Figure 3 demonstrates that while algorithm 2 in its LS and oracle variant improved both BP's and OMP's MSE, the two perform differently. At first, the oracle seems slightly favorable and indeed achieves better results for the optimal σ_n. Surprisingly however, when too much noise is added their difference diminishes. Comparing Algorithm 1 to Algorithm 2, the former is much more robust to the standard deviation of the SR noise. The source for this difference is the averaging operator. While Algorithm 1 uses the prior in order to weigh the solutions correctly, in Algorithm 2 each support is weighed by its probability to be chosen in the SR process. In general, the weights given by the SR process do not match the MMSE weights in Equation (3).

In the following sections we introduce additional assumptions in order to expand the theoretical and empirical analysis presented. First we analyze the case in which the dictionary consists of a unitary matrix and then we analyze the case of a general normalized dictionary, while restricting the cardinality of the sparse vector to be 1.

VI. General SR in the Unitary Case

A. The Unitary Sparse Estimators

When the dictionary is a unitary $n \times n$ matrix, we can simplify the expressions associated with the oracle, MAP and MMSE estimators as suggested in [18, 19]. Given a support S, the oracle estimator is a constant shrinkage applied on the projected measurements $\boldsymbol{\beta}_S = \boldsymbol{D}_S^T \boldsymbol{y}$

$$\hat{\boldsymbol{\alpha}}^{\text{Oracle}}_S(\boldsymbol{y}) = c^2 \boldsymbol{\beta}_S,$$

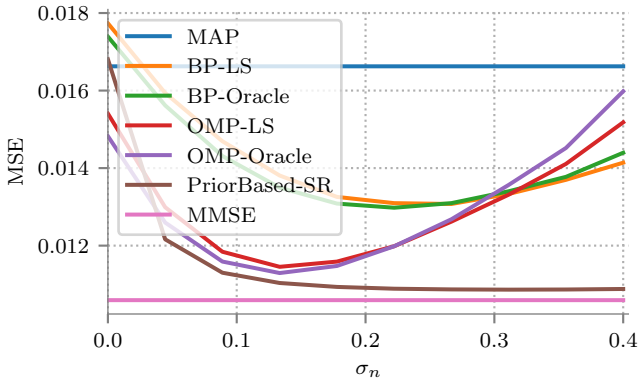


Fig. 3: MSE comparison between Algorithm 1 with OMP, and Algorithm 2 with both OMP and BP in its oracle and LS forms.

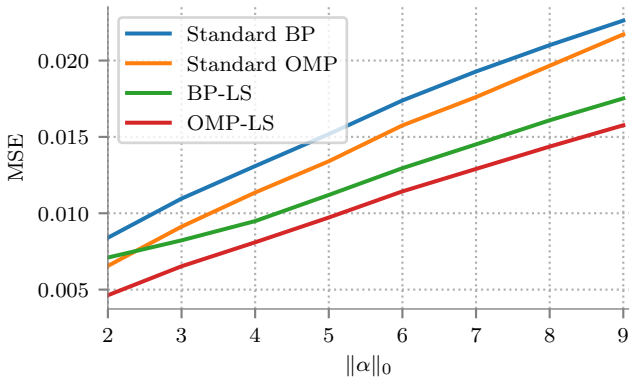


Fig. 4: Algorithm 2 with OMP and BP in its LS form compared to standard OMP and BP pursuits vs. the cardinality of the sparse representation vector. σ_n is optimal and obtained empirically.

where $c^2 = \sigma_\alpha^2/(\sigma_\alpha^2+\sigma_\nu^2)$.

The MAP estimator is reduced to the element-wise hard thresholding operator applied on the projected measurements $\boldsymbol{\beta} = \boldsymbol{D}^T\boldsymbol{y}$, given by

$$\hat{\alpha}_{MAP}(\beta) = \mathcal{H}_{\lambda_{MAP}}(\beta) = \begin{cases} c^2\beta & \text{if } |\beta| \geq \lambda_{MAP}, \\ 0 & \text{otherwise} \end{cases},$$

where $\lambda_{MAP} \triangleq \frac{\sqrt{2}\sigma_\nu}{c}\sqrt{\log\left(\frac{1-p_i}{p_i\sqrt{1-c^2}}\right)}$, and α and β are the elements of the vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

The MMSE estimator in the unitary case is a simple elementwise shrinkage operator of the following form:

$$\hat{\alpha}_{MMSE} = \psi(\beta) = \frac{\exp\left(\frac{c^2}{2\sigma_\nu^2}\beta^2\right)\frac{P_i}{1-P_i}\sqrt{1-c^2}}{1+\exp\left(\frac{c^2}{2\sigma_\nu^2}\beta^2\right)\frac{P_i}{1-P_i}\sqrt{1-c^2}}c^2\beta.$$

Note that this shrinkage operator does not result in a sparse vector, just as in the general case. The above scalar operators are extended to act on vectors in an entry-wise manner.

B. The Unitary SR Estimator

We now analyze the estimator suggested in Algorithm 2, under the unitary dictionary assumption.

Proposition 3. *Let $\boldsymbol{D}$ be a unitary matrix and denote by $Q(\cdot)$ the tail probability of the standard normal distribution*[2]*. Suppose that we use Algorithm 2 with the MAP estimator $\mathcal{H}_\lambda$ and white Gaussian SR noise $\boldsymbol{n}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma_n^2\boldsymbol{I})$ as a pursuit. Then, asymptotically, $\hat{\alpha} = \left[Q\left(\frac{\lambda+\beta}{\sigma_n}\right) + Q\left(\frac{\lambda-\beta}{\sigma_n}\right)\right]c^2\beta$.*

Proof. When using the MAP estimator in Algorithm 2, the thresholding operator is only used to recover the support itself. Once the support is extracted, the final estimator computes the oralce estimator w.r.t. the obtained supports before applying an empirical mean. This process can be equivalently described by the following elementwise *subtractive hard thresholding* operator[3] $\mathcal{H}_\lambda^-(\cdot)$:

$$\hat{\alpha}_k(\beta, \tilde{n}_k) = \mathcal{H}^-(\beta, \tilde{n}_k) = \begin{cases} c^2\beta & \text{if } |\beta+\tilde{n}_k| \geq \lambda_{MAP}, \\ 0 & \text{otherwise} \end{cases},$$

where $\tilde{\boldsymbol{n}}_k = \boldsymbol{D}^T\boldsymbol{n}_k$. Clearly, since $\boldsymbol{D}$ is unitary, $\tilde{\boldsymbol{n}}_k \sim \mathcal{N}\left(\boldsymbol{0}, \sigma_n^2\boldsymbol{I}\right)$. The final estimator is then achieved by an empirical mean,

$$\hat{\boldsymbol{\alpha}} = \frac{1}{K}\sum_{k=1}^{K}\hat{\boldsymbol{\alpha}}_k = \frac{1}{K}\sum_{k=1}^{K}\mathcal{H}_\lambda^-\left(\boldsymbol{\beta}+\tilde{\boldsymbol{n}}_k\right).$$

As $K \to \infty$, the described process asymptotically converges to the expectation

$$\begin{aligned}\mathbb{E}_{\tilde{n}}\left[\mathcal{H}_\lambda^-(\beta,\tilde{n})\right] &= \int_{-\infty}^{\infty}\mathcal{H}_\lambda^-(\beta,\tilde{n})\,p(\tilde{n})\,d\tilde{n} \\ &= \left[Q\left(\frac{\lambda+\beta}{\sigma_n}\right) + Q\left(\frac{\lambda-\beta}{\sigma_n}\right)\right]c^2\beta. \quad (4)\end{aligned}$$

The full derivation can be found in Appendix A. □

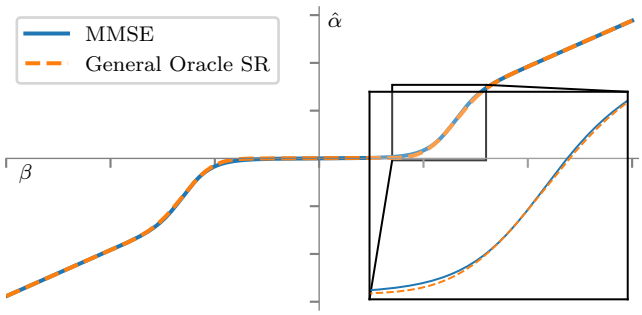
Unfortunately, analytically bounding the MSE between (4) and the MMSE estimator proves to be challenging. However, as we will see shortly, Equation (4) numerically approximates the MMSE very accurately. Note that there are two parameters yet to be set: σ_n and λ. The former tunes the magnitude of the added noise, while the latter controls the value of the thresholding operation. The original MAP threshold λ_{MAP} might be sub-optimal due to the addition of SR noise and therefore, we leave λ as a free parameter. We will suggest a method to set these parameters later in this section.

C. Unitary SR Estimation Results

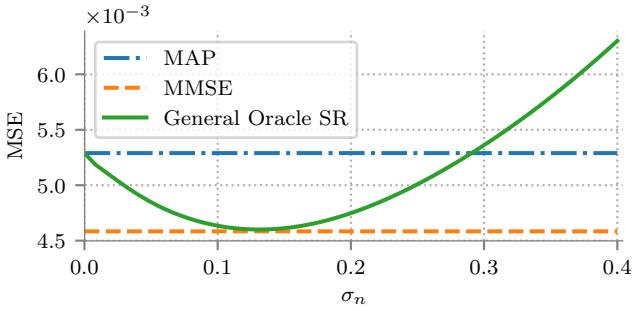
In order to demonstrate the similarity of the proposed estimator to the MMSE estimator, we compare their shrinkage curves in Figure 5a. One can see that, while the curves do not overlap completely, for the right choice of parameters (λ and σ_n), the curves are indeed quite close to each other. In terms of MSE, in Figure 5b we compare the performance of the general SR method and the MMSE as a function of σ_n (with λ fixed at the optimal value). Indeed, for the optimal parameters, the two are almost identical. In Appendix B, the performance of the general SR estimator is demonstrated as a function of both λ and σ_n.

[2] $Q(\cdot)$ is given by $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-\frac{u^2}{2}}du$.

[3] Notice that the written equation operates on the the vectors elementwise.



(a) Shrinkage curves comparison.



(b) General SR estimator's MSE for varying σ_n.

Fig. 5: The asymptotic general SR estimator vs. the MMSE. $\boldsymbol{D}$ is a unitary 100×100 dictionary, $\boldsymbol{\nu} \sim \mathcal{N}\left(\mathbf{0}, \sigma_{\boldsymbol{\nu}}^2 \boldsymbol{I}\right)$, $\sigma_{\boldsymbol{\nu}} = 0.2$, $P_i = 0.05\ \forall i \in [m]$ and $\boldsymbol{\alpha}_s \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{I}\right)$.

We now discuss how to set the parameters in order to reach these optimal results.

D. Finding the Optimal Parameters for the Unitary Case

To optimize the free parameters λ and σ_n, we propose to use Stein's Unbiased Risk Estimate (SURE) [20] which measures an estimator's MSE up to a constant when the additive noise is Gaussian. The SURE formulation of the expected MSE is given by

$$\mu(\mathcal{H}_\lambda^-(\boldsymbol{\beta}, \sigma_n), \boldsymbol{\beta}) = \|\mathcal{H}_\lambda^-(\boldsymbol{\beta}, \sigma_n)\|_2^2 - 2\mathcal{H}_\lambda^-(\boldsymbol{\beta}, \sigma_n)^T \boldsymbol{\beta} + 2\sigma_{\boldsymbol{\nu}}^2 \nabla_{\boldsymbol{\beta}} \mathcal{H}_\lambda^-(\boldsymbol{\beta}, \sigma_n).$$

In the unitary case this is further simplified to an element-wise sum:

$$\mu(\mathcal{H}_\lambda^-(\boldsymbol{\beta}, \sigma_n), \boldsymbol{\beta}) = \sum_i \mu(\mathcal{H}_\lambda^-(\beta_i, \sigma_n), \beta_i).$$

Expanding the estimator, one gets

$$\mu(\mathcal{H}_\lambda^-(\boldsymbol{\beta}, \sigma_n), \boldsymbol{\beta}) = \sum_i \mathcal{H}_\lambda^-(\beta_i, \sigma_n)^2 - 2\mathcal{H}_\lambda^-(\beta_i, \sigma_n)\beta_i + 2\sigma_{\boldsymbol{\nu}}^2 \frac{d}{d\beta_i} \mathcal{H}_\lambda^-(\beta_i, \sigma_n), \tag{5}$$

and we wish to optimize for σ_n and λ:

$$\sigma_n, \lambda = \underset{\sigma_n, \lambda}{\arg\min}\ \mu(\mathcal{H}_\lambda^-(\boldsymbol{\beta}, \sigma_n), \boldsymbol{\beta}).$$

This can be further simplified, as shown in Appendix B. Also, this Appendix depicts the surface $\mathbb{E}_{\boldsymbol{n}}\mu$ for a specific experiment. Interestingly, we observe that the empirically obtained optimal λ is quite close to the threshold suggested by the MAP estimator.

Note that in this process, we obtain an MMSE approximation without explicitly knowing the prior distribution of each element P_i. Furthermore, if σ_α is not known, one can easily estimate it as follows: The dictionary is unitary and therefore the mean energy of the signal $\boldsymbol{y}$ is $\sigma_\nu^2 + \sigma_\alpha^2$. Assuming we know σ_ν one can easily obtain σ_α.

VII. General SR in the Single Atom Case

A. Cardinality 1 Performance

While the unitary case is simpler to analyze, most applications rely on overcomplete dictionaries. In Section V, we already showed that empirically, Algorithm 2 improves the MSE performance of standard pursuit algorithms in the general case. We now try to further analyze Algorithm 2 by introducing a single atom assumption, i.e. assuming that the cardinality of the sparse vectors is restricted to one. From [14] we have that in this case, the MAP estimator described in Section III boils down to the following form:

$$\hat{\alpha}_i^{\text{MAP}}(\boldsymbol{y}) = \begin{cases} c^2 \beta_{\hat{S}}, & i = \hat{S} \\ 0, & i \neq \hat{S} \end{cases}, \quad \beta_{\hat{S}} = \boldsymbol{d}_{\hat{S}}^T \boldsymbol{y}, \quad c^2 = \frac{\sigma_\alpha^2}{\sigma_\alpha^2 + \sigma_\nu^2},$$

where $\hat{\alpha}_i^{\text{MAP}}$ is the i-th index in the vector $\hat{\boldsymbol{\alpha}}^{\text{MAP}}$, $\boldsymbol{d}_i$ is the ith atom and $\hat{S}$ represents the chosen atom index:

$$\hat{S} = \underset{S}{\arg\min} \|\beta_S \boldsymbol{d}_S - \boldsymbol{y}\|_2^2 = \underset{S}{\arg\max} \left|\boldsymbol{d}_S^T \boldsymbol{y}\right|.$$

Proposition 4. *Let $\boldsymbol{D}$ be a dictionary with normalized atoms and $\boldsymbol{\alpha}$ a sparse representation vector such that $\|\boldsymbol{\alpha}\|_0 = 1$, and suppose we use Algorithm 2 with the MAP estimator as a pursuit algorithm. Then, asymptotically, one obtains $\hat{\boldsymbol{\alpha}}$ such that its i-th index is $\hat{\alpha}_i = c^2 \boldsymbol{d}_i^T \boldsymbol{y} \cdot P(\hat{S} = i)$, where $P(\hat{S} = i)$ is the probability of the SR process to retrieve the support $\hat{S}$.*

Proof. Following the subtractive hard thresholding concept suggested in the previous section, we introduce the following SR estimator:

$$\hat{\alpha}_{k_i}(\boldsymbol{y}, \boldsymbol{n}_k) = \begin{cases} c^2 \beta_{\hat{S}_{\text{SR}}}, & i = \hat{S}_{\text{SR}} \\ 0, & i \neq \hat{S}_{\text{SR}} \end{cases}, \quad \beta_{\hat{S}_{\text{SR}}} = \boldsymbol{d}_{\hat{S}_{\text{SR}}}^T \boldsymbol{y},$$

where this time the chosen index $\hat{S}_{\text{SR}}$ is affected by an additive SR noise:

$$\begin{aligned} \hat{S}_{\text{SR}} &= \underset{S}{\arg\min} \|\boldsymbol{d}_S^T(\boldsymbol{y} + \boldsymbol{n}) - (\boldsymbol{y} + \boldsymbol{n})\|_2^2 \\ &= \underset{S}{\arg\max} \left|\boldsymbol{d}_S^T (\boldsymbol{y} + \boldsymbol{n}_k)\right|. \end{aligned} \tag{6}$$

Hence, asymptotically, Algorithm 2 converges to

$$\begin{aligned} \mathbb{E}_{\boldsymbol{n}}\left[\hat{\boldsymbol{\alpha}}(\boldsymbol{y}, \boldsymbol{n})\right] &= \mathbb{E}_S\left[\mathbb{E}_{\boldsymbol{n}|S}\left[\hat{\boldsymbol{\alpha}}(\boldsymbol{y}, \boldsymbol{n})|S\right]\right] \\ &= \sum_{i=1}^m \mathbb{E}_{\boldsymbol{n}|S}\left[\hat{\boldsymbol{\alpha}}|S = i\right] P\left(\hat{S} = i\right) \\ &= c^2 \begin{bmatrix} \beta_1 P\left(\hat{S} = 1\right) \\ \vdots \\ \beta_m P\left(\hat{S} = m\right) \end{bmatrix}. \end{aligned}$$

as claimed. □

As before, the difference between the general MMSE estimator (3) and Algorithm in Equation 2 in its oracle form is in the weight assigned to each solution $P(\hat{S})$. We now further analyze the weight obtained in the SR process under the single atom assumption. As stated in (6), the chosen atom i is the most correlated one with the input SR noisy signal:

$$\begin{aligned} P(\hat{S}=i) &= P\left(\left|\boldsymbol{d}_i^T(\boldsymbol{y}+\boldsymbol{n})\right| > \max_{j\neq i}\left|\boldsymbol{d}_j^T(\boldsymbol{y}+\boldsymbol{n})\right|\right) \\ &= P\left(|\tilde{n}_i| > \max_{j\neq i}|\tilde{n}_j|\right), \end{aligned} \tag{7}$$

where we defined $\tilde{\boldsymbol{n}} \triangleq \boldsymbol{D}^T(\boldsymbol{y}+\boldsymbol{n})$. Choosing $\boldsymbol{n} \sim \mathcal{N}\left(\boldsymbol{0}, \sigma_n^2\boldsymbol{I}\right)$ then $\tilde{\boldsymbol{n}}$ is Gaussian as well:

$$\tilde{\boldsymbol{n}} = \begin{bmatrix} \tilde{n}_1 \\ \vdots \\ \tilde{n}_m \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{D}^T\boldsymbol{y}, \sigma_n^2\boldsymbol{D}^T\boldsymbol{D}\right). \tag{8}$$

Therefore, the probability of choosing the i-th atom is distributed as the probability of the maximum value of a random Gaussian vector with *correlated* variables, since in the non-unitary case, $\boldsymbol{D}^T\boldsymbol{D}$ is not a diagonal matrix. Facing this difficulty, we propose to tackle it as follows:

1) Instead of adding the SR noise to $\boldsymbol{y}$, we can add it to the projected signal $\boldsymbol{D}^T\boldsymbol{y}$, thus avoiding the variables $\{\tilde{n}_i\}_{i=1}^m$ being correlated.
2) Add statistical assumptions regarding the dictionary $\boldsymbol{D}$, leading to average case conclusions.
3) Change the pursuit used. Intuitively, using the MAP will produce the optimal results, since it retrieves the most probable support for any given signal. However, changing the pursuit might ease the analysis of the asymptotic estimator. We leave the study of this option for future work.

We now analyze the first two proposed alternatives.

B. Add Noise to the Representation Domain

Proposition 5. *Let the same conditions as Proposition 4 hold. Moreover, let $\boldsymbol{n} \sim \mathcal{N}\left(\boldsymbol{0}, \sigma_n^2\boldsymbol{I}\right)$ be an SR noise added to the representation domain. Then, the probability to retrieve the i-th support is given by*

$$\begin{aligned} P(\hat{S}=i) = \int_0^\infty \frac{1}{\sqrt{2\pi}\sigma_n}\left[e^{-\frac{(t+\beta_i)^2}{2\sigma_n^2}} + e^{-\frac{(t-\beta_i)^2}{2\sigma_n^2}}\right] \\ \times \prod_{j\neq i}\left[1-\left(Q\left(\frac{t-\beta_j}{\sigma_n}\right)+Q\left(\frac{t+\beta_j}{\sigma_n}\right)\right)\right]dt. \end{aligned} \tag{9}$$

Proof. We continue from (7), only now the noise $\tilde{n}_i$ is white and has the following properties:

$$\tilde{\boldsymbol{n}} \sim \mathcal{N}\left(\boldsymbol{D}^T\boldsymbol{y}, \sigma_n^2\boldsymbol{I}_{m\times m}\right).$$

Plugging this into (7):

$$\begin{aligned} P(\hat{S}=i) &= P\left(\left|\boldsymbol{d}_i^T\boldsymbol{y}+n_i\right| > \max_{j\neq i}\left|\boldsymbol{d}_j^T\boldsymbol{y}+n_j\right|\right) \\ &= P\left(|\tilde{n}_i| > \max_{j\neq i}|\tilde{n}_j|\right) \\ &= \int_0^\infty P\left(\max_{j\neq i}|\tilde{n}_j| < t \middle| |\tilde{n}_i| = t\right) P\left(|\tilde{n}_i| = t\right) dt \\ &= \int_0^\infty P\left(\max_{j\neq i}|\tilde{n}_j| < t\right) P\left(|\tilde{n}_i| = t\right) dt. \end{aligned} \tag{10}$$

For the first term, the elements of $\tilde{\boldsymbol{n}}$ are independent, and therefore

$$\begin{aligned} P\left(\max_{j\neq i}|\tilde{n}_j| < t\right) &= \prod_{j\neq i} P\left(|\tilde{n}_j| < t\right) \\ &= \prod_{j\neq i}\left[1 - P\left(|\tilde{n}_j| > t\right)\right] \\ &= \prod_{j\neq i}\left[1-\left(Q\left(\frac{t+\beta_j}{\sigma_n}\right)+Q\left(\frac{t-\beta_j}{\sigma_n}\right)\right)\right], \end{aligned}$$

where the last equality follows similar steps as in Appendix A. The second term in (10) is simply the PDF of the absolute value of a Gaussian variable, therefore

$$P\left(|\tilde{n}_i| = t\right) = \frac{1}{\sqrt{2\pi\sigma_n^2}}\left(e^{-\frac{(t-\beta_i)^2}{2\sigma_n^2}} + e^{-\frac{(t+\beta_i)^2}{2\sigma_n^2}}\right).$$

Putting the two terms back into (10) we obtained the claimed relation in Equation (9). □

The obtained expression cannot be solved analytically but can be computed numerically. We now empirically examined the properties of the derived estimator. We generated a random dictionary of size 25×50 with iid Gaussian elements and normalized the atoms. Then, we generated sparse vectors with a single non-zero element with a Gaussian value $\alpha_S \sim \mathcal{N}(0,1)$. Noisy measurements were generated by multiplying the dictionary with the sparse vectors and adding noise $\boldsymbol{\nu} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{I}\sigma_\nu^2\right)$, $\sigma_\nu = 0.2$. In Figure 6 we compare the MSE of the MMSE and MAP estimators to Algorithm 2 when the noise is added to the representation domain. Indeed, the proposed method improves the MSE of the MAP estimator and almost achieves the MMSE estimator's performance for the right choice of σ_n.

In Figure 7 we show the probability of recovering the true support P_{success} as a function of σ_n, both from (9) and from iterating Algorithm 2 100 times, each time picking the most correlated atom. We also compare it to the MMSE weight from (2). The optimal σ_n in terms of MSE is drawn as a vertical dashed line. Notice that for the optimal choice of σ_n, the probability of the true support to be chosen is similar to that given in the MMSE solution. In other words, the optimal σ_n is the one that approximates the weight of the support to the weight given by the MMSE expression. The trend in (9) shows, unsurprisingly, that as we add noise, the probability of successfully recovering the true support decreases. In the limit, when $\sigma_n \to \infty$ the signal will be dominated by the noise and

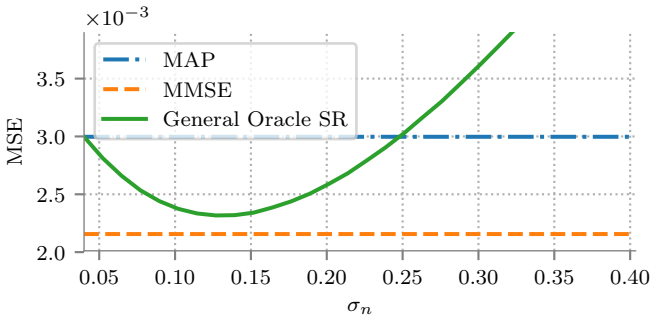


Fig. 6: MSE comparison of the MAP, MMSE and 100 iterations of General Oracle SR with non unitary overcomplete dictionary and a sparse representation with 1 non-zero element.

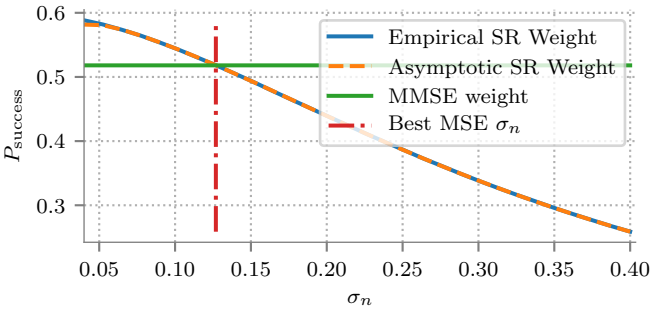


Fig. 7: Numerical integration of $P(S = \text{True Suport}|y)$ and the empirical weight achieved by 100 iterations of SR.

the success probability will be uniform among all the atoms, i.e. equal to $P_{success} = \frac{1}{m}$.

Due to these findings, and since the optimal MSE is comparable to that of the MMSE estimator, one might expect a similar behavior for most of the other possible supports. To examine this idea, we carried out the following experiment. We randomized many representations $\boldsymbol{\alpha}$, all containing a non-zero coefficient in the same index. Then, we ploted the histogram of the average empirical probability of each element in the vector $\boldsymbol{\alpha}$ to be non-zero (obtained by pursuit). Finally, we compared these probabilities to the weights of the MMSE from (2). This experiment was repeated for various σ_n values and for each such value we compared the entire support histogram. We expect the two histograms (SR and MMSE) to fit for the right choice of added noise parameter σ_n. In Figure 8 we see the results of the described experiment.

Analyzing the results obtained, we see that when no noise is added (this is the average case of the MAP estimator), apart from the true support, the elements have a much lower weight than the MMSE. As noise is added, the true support's probability decreases and its weight is divided among the other elements. At some point the two histograms almost match each other completely. At that point, the SR MSE almost equals that of the MMSE. As we add more noise, the true support's probability keeps decreasing and the other elements keep increasing and the histograms are now farther apart from each other. When we reach $\sigma_n \to \infty$ we obtain uniform probability for all the supports.

To further demonstrate their similarity, the left axis in Figure 9 is the $D_{K\|L}$ distance (Kullback-Leibler divergence) between the two histograms, and the right axis is the MSE. As expected, when their $K\|L$ divergence is the smallest, the MSE is minimal.

C. Statistical Assumptions on the Dictionary $\boldsymbol{D}$

In this section we will try to simplify the expression in (7) by assuming that the columns of the dictionary are statistically uncorrelated. Formally, our assumption is that the atoms $\boldsymbol{d}_i$ are drawn from some random distribution that obeys the following properties:

$$\mathbb{E}\left[\boldsymbol{d}_i^T \boldsymbol{d}_j\right] = 0, \quad i \neq j \quad 1 \leq i, j \leq m, \tag{11}$$

and that the atoms are normalized:

$$\|\boldsymbol{d}_i\|_2 = 1, \quad 1 \leq i \leq m. \tag{12}$$

Proposition 6. *Let the same conditions as Proposition 4 hold and furthermore suppose that the dictionary's atoms are statistically uncorrelated. Then, when using Algorithm 2 with the MAP estimator, adding white Gaussian SR noise to the signal domain with standard deviation σ_n is equivalent to adding white Gaussian SR noise to the representation domain with the same standard deviation σ_n.*

Proof. From (8), we have $\tilde{\boldsymbol{n}} = \boldsymbol{D}^T(\boldsymbol{y} + \boldsymbol{n})$. Observe that given the dictionary $\boldsymbol{D}$, each of the elements in this vector is a Gaussian random variable:

$$\tilde{n}_i|\boldsymbol{d}_i = \boldsymbol{d}_i^T(\boldsymbol{y} + \boldsymbol{n}) = \sum_{k=1}^{n} d_{i,k}(y_k + n_k) = \sum_{k=1}^{n} d_{i,k} y_k + \sum_{k=1}^{n} d_{i,k} n_k = \mu_i + \sum_{k=1}^{n} d_{i,k} n_k.$$

Given the measurements and the dictionary, the first sum $\sum_{k=1}^{n} d_{i,k} y_k \triangleq \mu_i$ is some constant. The second term in the expression is a weighted sum of n iid Gaussian random variables $\{n_k\}_{k=1}^{n}$, hence it is Gaussian. Clearly, its mean value is 0, and its standard deviation is σ_n, hence $\tilde{n}_i|\boldsymbol{d}_i \sim \mathcal{N}(\boldsymbol{d}_i^T \boldsymbol{y}, \sigma_n)$ for $n \to \infty$.

Now we turn to analyze the properties of the entire vector $\tilde{\boldsymbol{n}}$. From the previous analysis, given the dictionary $\boldsymbol{D}$, $\tilde{\boldsymbol{n}}$ is a random Gaussian vector with the mean vector $\boldsymbol{\mu}_{\tilde{\boldsymbol{n}}}|\boldsymbol{D} = \boldsymbol{D}^T\boldsymbol{y}$. Using the properties of the noise $\mathbb{E}\left[\boldsymbol{n}\boldsymbol{n}^T\right] = \sigma_n^2 I$, the auto-correlation matrix of $\tilde{\boldsymbol{n}}|\boldsymbol{D}$ is by definition:

$$\boldsymbol{\Sigma}|\boldsymbol{D} = \mathbb{E}\left[\boldsymbol{D}^T\boldsymbol{n}\boldsymbol{n}^T\boldsymbol{D}|\boldsymbol{D}\right] = \boldsymbol{D}^T\mathbb{E}\left[\boldsymbol{n}\boldsymbol{n}^T\right]\boldsymbol{D} = \sigma_n^2\boldsymbol{D}^T\boldsymbol{D}.$$

Analyzing the average case, the mean vector is of the form:

$$\boldsymbol{\mu}_{\tilde{\boldsymbol{n}}} = \mathbb{E}_{\boldsymbol{D}}\left[\boldsymbol{D}^T\boldsymbol{y}\right],$$

and the auto-correlation matrix is simply diagonal:

$$\boldsymbol{\Sigma} = \mathbb{E}_{\mathbb{D}}[\boldsymbol{\Sigma}|\boldsymbol{D}] = \mathbb{E}\left[\sigma_n^2\boldsymbol{D}^T\boldsymbol{D}\right] = \sigma_n^2\boldsymbol{I},$$

where we used the assumptions in (11) and (12). □

From Proposition 6, the uncorrelated atoms assumption leads $\tilde{\boldsymbol{n}}$ to have the same properties as in Proposition 5.

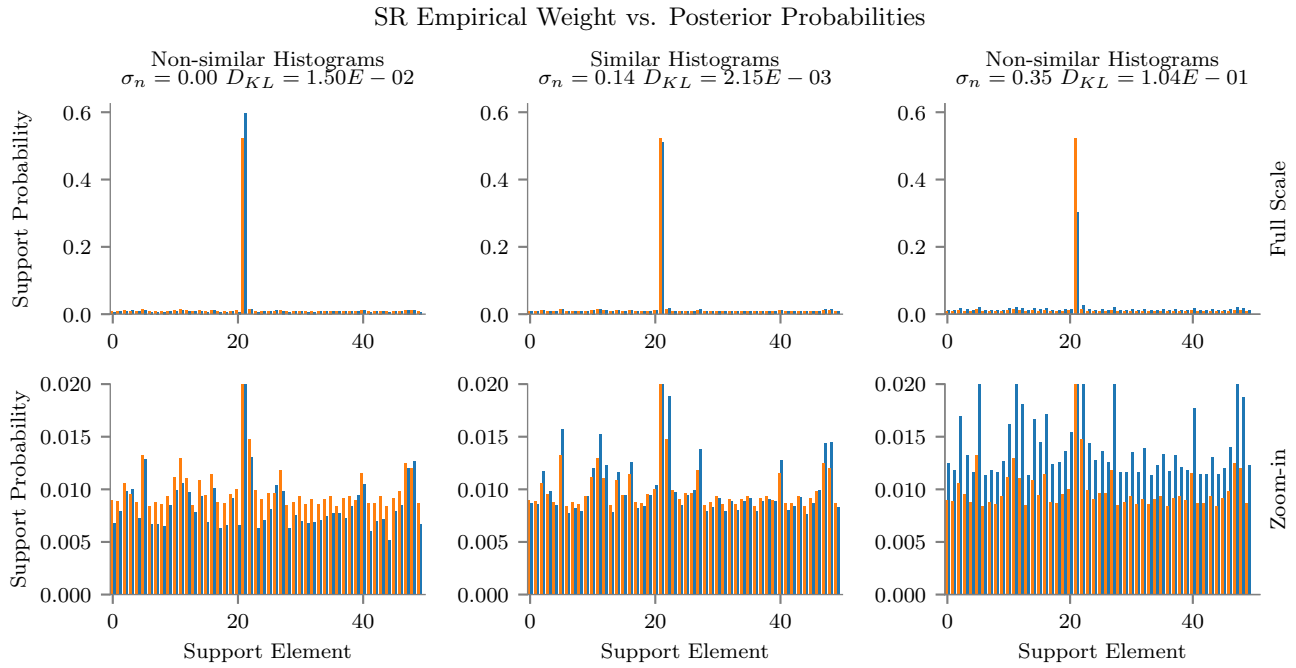


Fig. 8: MMSE (in orange) and 100 iterations of General-SR (in blue) support weights histograms for varying values of σ_n; Top row presents the entire scale; Bottom row is zoomed-in to emphasize the differences in the smaller weights; Title for each column shows the σ_n value and KL divergence between the histograms; Atom number 21 is the true support.

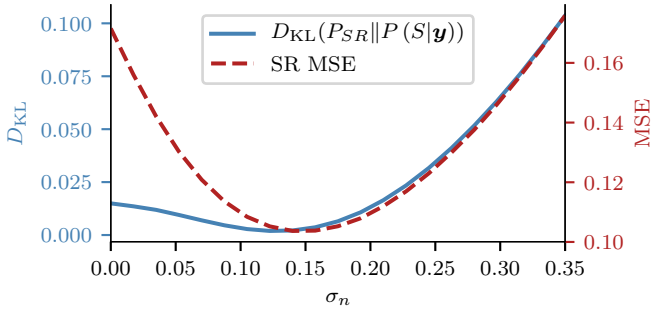


Fig. 9: Subtractive SR MSE and $D_{K\|L}$ divergence between the MMSE and SR weights. When the divergence is small so is the MSE.

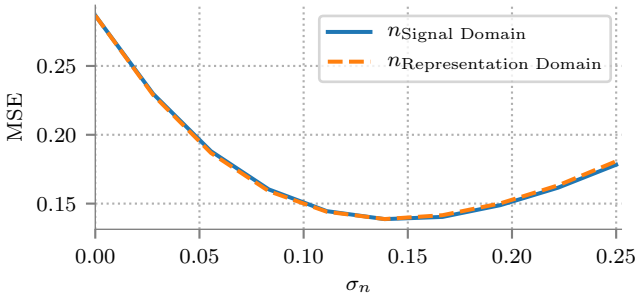


Fig. 10: Noise location comparison. 500 iterations of General Oracle SR with MAP estimator as a pursuit method. $\boldsymbol{\nu} \sim \mathcal{N}\left(\mathbf{0}, \sigma_{\boldsymbol{\nu}}^2 I\right)$, $\sigma_{\boldsymbol{\nu}} = 0.2$, $||\boldsymbol{\alpha}||_0 = 1$ and $\boldsymbol{\alpha}_s \sim \mathcal{N}(0,1)$. The SR noises are $\boldsymbol{n}_{\text{Signal Domain}} \sim \mathcal{N}(0, \sigma_n I_{n\times n})$. $\boldsymbol{n}_{\text{Representation Domain}} \sim \mathcal{N}(0, \sigma_n I_{m\times m})$.

Therefore, the empirical analysis following Proposition 5 holds for this case as well.

To demonstrate empirically that indeed the two are the same, we performed the following experiment. We sampled a random dictionary $\boldsymbol{D}$ and random sparse representations $\boldsymbol{\alpha}$ with cardinality of 1 as the generative model described earlier suggests. In this experiment we used a dictionary $\boldsymbol{D}$ of size 200×400 and 2000 random sparse representations. Using the generated vectors and dictionary we created signals $\boldsymbol{y}$: $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{\alpha} + \boldsymbol{\nu}$. To denoise the signals, we once used Algorithm 2 with noise $\boldsymbol{n}_{\text{sig.}} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \boldsymbol{I}_{n\times n})$ added to the signal vectors $\boldsymbol{y} + \boldsymbol{n}_{\text{sig.}}$, and once with noise $\boldsymbol{n}_{\text{rep.}} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \boldsymbol{I}_{m\times m})$ added to the representation domain $\boldsymbol{D}^T\boldsymbol{y} + \boldsymbol{n}_{\text{rep.}}$. As before, we use the MAP estimator as the chosen pursuit. In Figure 10 we see that the MSE of the two cases result in an almost identical curve. Small differences exist due to the finite dimensions used in the experiment.

Note that the noise energy added in the representation domain is much larger than that of the noise added to the signal, i.e. $E\|\boldsymbol{n}_{\text{rep.}}\|_2^2 = m\sigma_n^2 > n\sigma_n^2 = E\|\boldsymbol{n}_{\text{sig.}}\|_2^2$ but the results remain the same due to the unit norm of the dictionary $\|\boldsymbol{d}_i\|_2 = 1$, and of course, the uncorrelated assumption.

To conclude this subsection, statistically uncorrelated atoms provide a way to further our theoretical analysis. In this case, adding noise in the signal domain converges to the analysis addressed in the previous subsection, where noise is instead added in the representation domain. As a result, similar results and conclusions can be drawn.

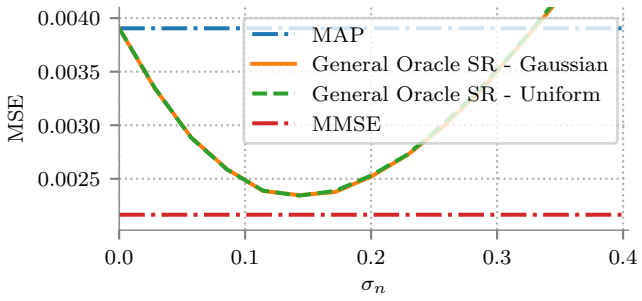


Fig. 11: Uniform vs. Gaussian SR noise. $\boldsymbol{D} \in \mathbb{R}^{50\times100}$, $\boldsymbol{\nu} \sim \mathcal{N}\left(\boldsymbol{0}, \sigma_{\boldsymbol{\nu}}^2 I\right)$, $\sigma_{\boldsymbol{\nu}} = 0.2$, $\|\boldsymbol{\alpha}\|_0 = 1$ and $\boldsymbol{\alpha}_s \sim \mathcal{N}(0,1)$. 100 iterations of Algorithm 2 with the MAP estimator as a pursuit.

VIII. What Noise Should be Used?

Throughout this work we naturally used white Gaussian noise by default. In this section we question this decision and wonder whether we can use noise models with different distributions and whether it affects the performance of the stochastic resonance estimator.

Theorem 7. *Let $\boldsymbol{n}$ be a random vector with iid elements sampled from a distribution with finite mean and variance and used as SR noise in Algorithm 2. Then, as the dimension of the sparse representation grows asymptotically, the estimate given by Algorithm 2 is not affected by $\boldsymbol{n}$'s distribution.*

Proof. Denoting $\tilde{\boldsymbol{n}} \triangleq \boldsymbol{D}^T\boldsymbol{n}$, each element $\tilde{n}_i$ is:

$$\tilde{n}_i = \boldsymbol{d}_i^T\boldsymbol{n} = \sum_{j=1}^{m} d_{i,j}n_j.$$

Without loss of generality, we assume that the atoms are normalized, i.e., $\|\boldsymbol{d}_i\|_2 = 1$. The above expression is a weighted average of the variables $\{n_j\}_{j=1}^m$. Since $\{n_j\}_{j=1}^m$ are iid and have bounded mean and variance, then the central limit theorem holds. Therefore, as m increases, $\tilde{n}_i$ is asymptotically Gaussian regardless of the distribution of the original additive noise n_j. □

Following the previous statement, we experimented with a different distribution for a random noise vector. We employed an element-wise iid uniform noise with 0 mean $n_{\mathcal{U}} \sim \mathcal{U}[-r, r]$. In order to be compatible with a Gaussian noise $n_{\mathcal{N}} \sim \mathcal{N}(0, \sigma_n^2)$ we chose $r = \sqrt{3}\sigma_n$ thus assuring the same standard deviation for the two cases. In Figure 11 we compare the random Gaussian noise with the uniform one as described, and indeed, the curves overlap.

In Appendix C we further experiment with a different form of SR noise, leading to similar performance in terms of MSE, while reducing the computations performed.

IX. Image Denoising

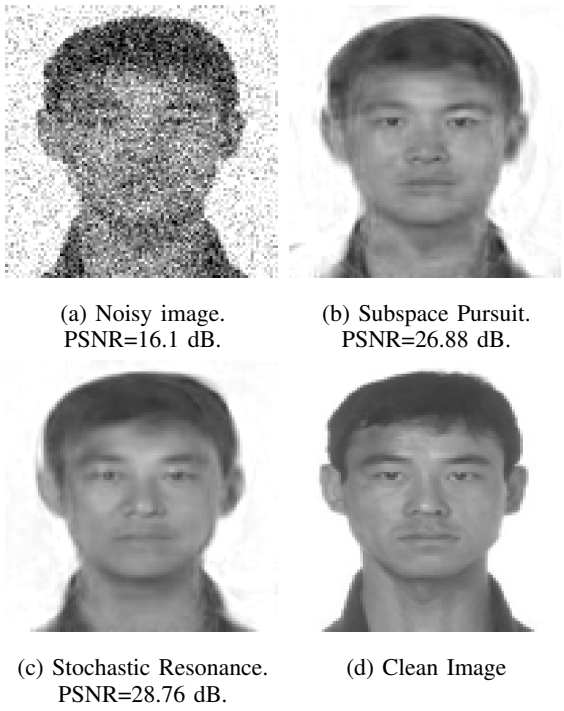
In this section we demonstrate the benefits of using Algorithm 2 in image denoising. We use the Trainlets [21] dictionary trained on facial images from the Chinese Passport dataset as described in [22]. In the dataset, each image is of size 100×100 pixels and contains a gray-scale aligned face. The application we address is denoising, that is approximating the following optimization problem:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{D}\boldsymbol{\alpha} - \boldsymbol{y}\|_2 \quad \text{s.t.} \quad \|\boldsymbol{\alpha}\|_0 = L.$$

Since this problem is intractable, an approximation is achieved using the Subspace Pursuit (SP) algorithm [23], which provides a fast converging algorithm for a fixed number of non-zeros L. The particular choice of using SP follows from the prohibitive cardinality and dimensions for, rendering the OMP as a highly non-efficient alternative.

In our experiment, we corrupt an unseen image from the dataset with additive white Gaussian noise, using various standard deviation σ_ν values. Then, we denoise the image using SP, where the number of non-zeros L is empirically set to maximize the denoising performance. We then apply Algorithm 2 in its LS form, using 200 iterations and the same SP settings. Note that we do not seek state of the art denoising results but rather to show that our method can be easily applied to improve real image processing tasks.

In Figure 12 the results for $\sigma_\nu = 40$ can be seen. Importantly, the SR result yield a clearer image with much less artifacts. Figure 13 presents the effectiveness of SR under varying SR noise σ_n. We see that a gain of almost 2 dB is achieved by using SR with a proper σ_n over the regular pursuit. Figure 14 presents a comparison of the PSNR values obtained by SP and Algorithm 2 with SP for varying values of standard deviation σ_ν. In all of the described experiments, Algorithm 2 improved the denoising results. Generally we observe that as the noise increases, the improvement becomes more significant.



(a) Noisy image. PSNR=16.1 dB.

(b) Subspace Pursuit. PSNR=26.88 dB.

(c) Stochastic Resonance. PSNR=28.76 dB.

(d) Clean Image

Fig. 12: Denoising results comparison. $\sigma_\nu = 40$, $L = 90$.

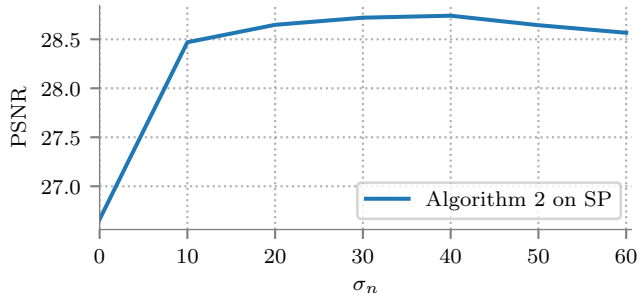


Fig. 13: SR results with varying σ_n for a noisy image with $\sigma_\nu = 40$, PSNR=16.1 dB. $\sigma_n = 0$ effectively does not use SR.

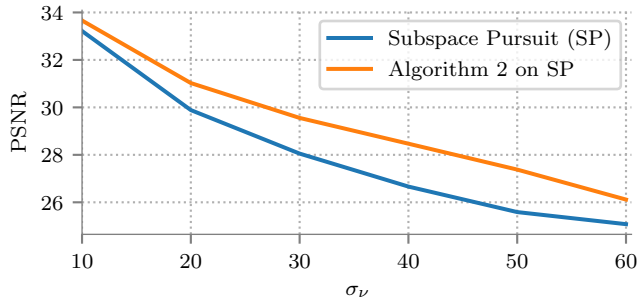


Fig. 14: SP and SP + Algorithm 2: Comparison for varying standard deviation values σ_v.

X. Conclusion

In this work we suggested two algorithms leveraging the idea of stochastic resonance under the context of sparse coding. We analyzed their theoretical properties and showed that they enable to efficiently deploy arbitrary pursuit algorithms while boosting their performance with SR, providing an approximation to the MMSE estimator. While the first method we suggested is provably convergent to the MMSE, it is not directly applicable in cases where the prior of the sparse vectors is not known. This brought us to introduce a relaxed and more practical alternative. We have analyzed the properties of this second path in several cases and demonstrated its superiority over standard pursuit algorithms in both synthetic cases and on a natural image denoising task. In contrast to previous MMSE approximation methods, the ones suggested in this work have the ability to use any pursuit algorithm as a "black box", thus opening the door for MMSE approximation in large dimension regimes for the first time.

Appendix A
Unitary General SR Asymptotic Estimator

Placing the normal distribution function into (4), we obtain:

$$\begin{aligned}
\mathbb{E}_n\left[\mathcal{H}_\lambda^-(\beta,n)\right] &= \int_{-\infty}^{\infty}\mathcal{H}_\lambda^-(\beta+n)\,p(n)\,dn \\
&= \int_{|\beta+n|\geq\lambda} c^2\beta p(n)\,dn \\
&= c^2\left[\int_{-\infty}^{-\lambda-\beta}\beta p(n)\,dn + \int_{\lambda-\beta}^{\infty}\beta p(n)\,dn\right] \\
&= c^2\beta\left[\int_{-\infty}^{-\lambda-\beta}\frac{1}{\sqrt{2\pi\sigma_n^2}}e^{-\frac{n^2}{2\sigma_n^2}}dn + \int_{\lambda-\beta}^{\infty}\frac{1}{\sqrt{2\pi\sigma_n^2}}e^{-\frac{n^2}{2\sigma_n^2}}\right] \\
&= c^2\beta\left[Q\left(\frac{\lambda+\beta}{\sigma_n}\right)+Q\left(\frac{\lambda-\beta}{\sigma_n}\right)\right].
\end{aligned}$$

Appendix B
The SURE Surface for the Unitary Case

Plugging the subtractive hard thresholding $\mathcal{H}_\lambda^-$ into (5) leads to the following expression:

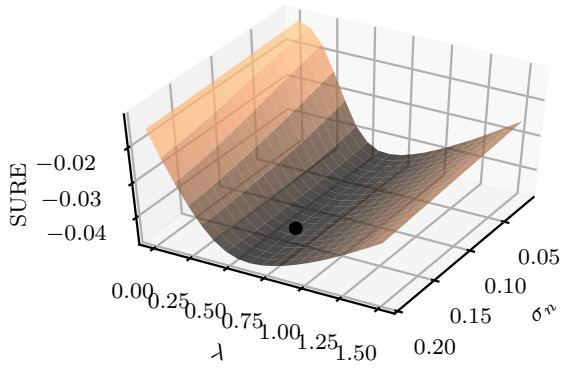
$$\begin{aligned}
\mathbb{E}_n\left[\mu\left(\mathcal{H}_\lambda^-\right)\right] &= \sum_i\left(c^2\beta_i\left[Q\left(\frac{\lambda+\beta_i}{\sigma_n}\right)+Q\left(\frac{\lambda-\beta_i}{\sigma_n}\right)\right]\right)^2 \\
&- \sum_i 2c^2\beta_i^2\left[Q\left(\frac{\lambda+\beta_i}{\sigma_n}\right)+Q\left(\frac{\lambda-\beta_i}{\sigma_n}\right)\right] \\
&+ \sum_i 2\sigma_\nu^2c^2\left[Q\left(\frac{\lambda+\beta_i}{\sigma_n}\right)+Q\left(\frac{\lambda-\beta_i}{\sigma_n}\right)\right] \\
&+ \sum_i 2\sigma_\nu^2c^2\beta_i\left[\frac{1}{\sqrt{2\pi}\sigma_n}e^{-\frac{(\lambda-\beta_i)^2}{2\sigma_n^2}} - \frac{1}{\sqrt{2\pi}\sigma_n}e^{-\frac{(\lambda+\beta_i)^2}{2\sigma_n^2}}\right].
\end{aligned}$$

In order to show that it is indeed easy to optimize λ and σ on the SURE surface, we demonstrate it by the following experiment. We generate sparse vectors with probability of $P_i = 0.01$ for any coefficient to be non-zero. The coefficients of the non-zero entries are drawn from a Gaussian distribution $\mathcal{N}(0,1)$. We then generate signals using a unitary dictionary and add random Gaussian noise $\mathcal{N}\left(0,0.2^2\right)$. We then compute $\mu\left(\mathcal{H}_\lambda^-\right)$ for various λ and σ_n values. Figure 15 presents the surface for these values, and Figure 16 shows the MSE results respectively. We can see that the SURE surface behaves just like the true MSE up to an additive constant and that it is smooth and rather easy to optimize. In terms of MSE, we see the superiority of the proposed estimator over the MAP estimator, and that it is quite close to the MMSE.

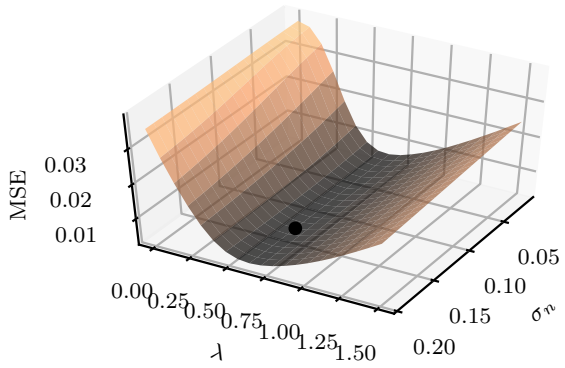
Appendix C
Multiplicative Bernoulli SR Noise

In this section we seek for an SR distribution from which we can benefit more than others in terms of computational efficiency. To do so, consider the following. Given the signal $\boldsymbol{y}$, we define the subsampling noise $\boldsymbol{n}_{\text{subsample}}$ in the following way:

$$n_i = \begin{cases} 1 & \text{w.p.} \quad p \\ 0 & \text{w.p.} \quad 1-p \end{cases},$$

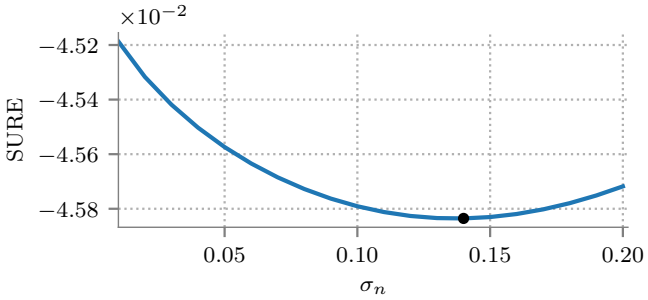


(a) SURE Surface. The minimum is located at •

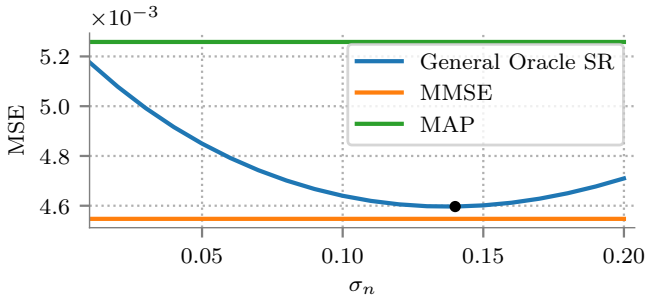


(b) MSE Surface. The minimum is located at •

Fig. 15: SURE and MSE values for a unitary dictionary with varying λ and σ_n.



(a) SURE curve for the optimal λ.



(b) MSE curve for the optimal λ extracted using SURE.

Fig. 16: SURE and MSE values for the optimal λ, extracted from SURE. SURE's optimal σ_n marked in •.

and the SR samples will now follow the following distribution:

$$y_i \cdot n_i = \begin{cases} y_i & \text{w.p.} \quad p \\ 0 & \text{w.p.} \quad 1-p \end{cases}.$$

Therefore, for an input signal of size n, only pn samples will remain on average. This distribution is interesting because of the following reason. When zeroing out an element in the vector $\boldsymbol{y}$, the matching row in the dictionary $\boldsymbol{D}$ will always be multiplied by the zero element when calculating the correlations $\boldsymbol{D}^T\boldsymbol{y}_{\text{SR}}$ as done in most pursuits. This multiplication obviously has no contribution to the inner product and we might as well omit the zero elements from $\boldsymbol{y}_{\text{SR}}$ and the corresponding rows from $\boldsymbol{D}$, leading to a subsampled version of the signal $\boldsymbol{y}$ and the dictionary $\boldsymbol{D}$. In other words, in each of the SR iterations we simply subsample random elements with probability p from the signal $\boldsymbol{y}$ and the matching rows from the dictionary $\boldsymbol{D}$, which leads to $\boldsymbol{y}_{\text{subsample}}$ of size $pn \times 1$ (on average) and a dictionary $\boldsymbol{D}_{\text{subsample}}$ of size $pn \times m$ (on average). Finally we sparse code the subsampled vectors. Just like in previous cases, we use the pursuit's result only as a support estimator in order to compute the oracle estimator. Hence, we then revert to the full sized signal $\boldsymbol{y}$ and dictionary $\boldsymbol{D}$ and compute either oracle or LS.

Note that when using the Bernoulli noise, each pursuit has a computational benefit over the previously presented additive SR noise due to the decreased size of the signal's dimension. In Figure 17 we show the results of the multiplicative Bernoulli noise compared to Gaussian additive noise. In this figure the x axis represents the probability p of the Bernoulli noise. We see that the two noise distributions lead to similar MSE results for the optimal choice of σ_n and p, while using multiplicative Bernoulli SR noise is computationally efficient compared to additive Gaussian SR noise.

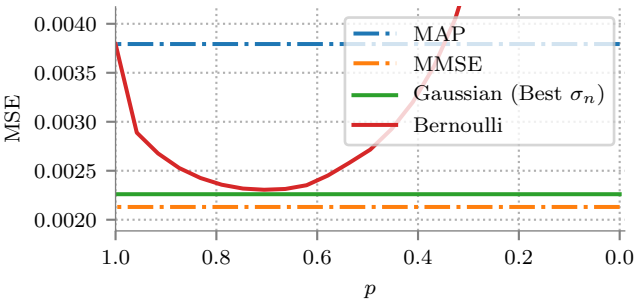


Fig. 17: Multiplicative Bernoulli SR noise vs. additive Gaussian SR noise with 100 iterations of Algorithm 1. $\boldsymbol{D} \in \mathbb{R}^{50\times 100}$, $\boldsymbol{\nu} \sim \mathcal{N}\left(\boldsymbol{0}, \sigma_{\boldsymbol{\nu}}^2 I\right)$, $\sigma_{\boldsymbol{\nu}} = 0.2$, $||\boldsymbol{\alpha}||_0 = 1$ and $\alpha_S \sim \mathcal{N}(0,1)$.

Acknowledgment

The authors thank Prof. Pramod K. Varshney, for his inspiring keynote talk at ICASSP 2015 in Brisbane, which inspired the initial ideas of this paper. The research leading to these results has received funding from the European Research Council under European Unions Seventh Framework Programme, ERC Grant agreement no. 320649 and the Israel Science Foundation (ISF) Grant no. 335/18.

REFERENCES

[1] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[3] M. Elad and M. Aharon, "Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, 2006.

[4] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM journal on computing*, vol. 24, no. 2, pp. 227–234, 1995.

[5] J. A. Tropp and A. C. Gilbert, "Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, 2007.

[6] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[7] D. L. Donoho, M. Elad, and V. N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 6–18, 2006.

[8] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[9] J. A. Tropp, "Average-case analysis of greedy pursuit," in *Proc. of SPIE Vol*, vol. 5914, pp. 591412–1, 2005.

[10] S. Kay, J. H. Michels, H. Chen, and P. K. Varshney, "Reducing Probability of Decision Error Using Stochastic Resonance," *IEEE Signal Processing Letters*, vol. 13, no. 11, 2006.

[11] H. Chen, P. K. Varshney, and J. H. Michels, "Noise enhanced parameter estimation," *IEEE Transactions on Signal Processing*, vol. 56, no. 10 II, pp. 5074–5081, 2008.

[12] H. Chen, L. R. Varshney, and P. K. Varshney, "Noise-enhanced information systems," *Proceedings of the IEEE*, vol. 102, no. 10, pp. 1607–1621, 2014.

[13] N. Stocks, "Suprathreshold stochastic resonance in multilevel threshold systems," *Physical Review Letters*, vol. 84, no. 11, p. 2310, 2000.

[14] M. Elad and I. Yavneh, "A Weighted Average of Sparse Representations is Better than the Sparsest One Alone," *IEEE Transactions on Information Theory*, vol. 55, no. 10, pp. 1–35, 2009.

[15] P. Schniter, L. C. Potter, and J. Ziniel, "Fast bayesian matching pursuit," in *Information Theory and Applications Workshop, 2008*, pp. 326–333, IEEE, 2008.

[16] F. Chapeau-Blondeau and D. Rousseau, "Noise-aided snr amplification by parallel arrays of sensors with saturation," *Physics letters A*, vol. 351, no. 4-5, pp. 231–237, 2006.

[17] N. Stocks and R. Mannella, "Generic noise-enhanced coding in neuronal arrays," *Physical Review E*, vol. 64, no. 3, p. 030902, 2001.

[18] J. S. Turek, I. Yavneh, and M. Elad, "On MMSE and MAP denoising under sparse representation modeling over a unitary dictionary," *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pp. 3526–3535, 2011.

[19] M. Protter, I. Yavneh, and M. Elad, "Closed-form mmse estimation for signal denoising under sparse representation modeling over a unitary dictionary," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3471–3484, 2010.

[20] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *The annals of Statistics*, pp. 1135–1151, 1981.

[21] J. Sulam, B. Ophir, M. Zibulevsky, and M. Elad, "Trainlets: Dictionary learning in high dimensions," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3180–3193, 2016.

[22] J. Sulam and M. Elad, "Large inpainting of face images with trainlets," *IEEE Signal Processing Letters*, vol. 23, no. 12, pp. 1839–1843, 2016.

[23] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.