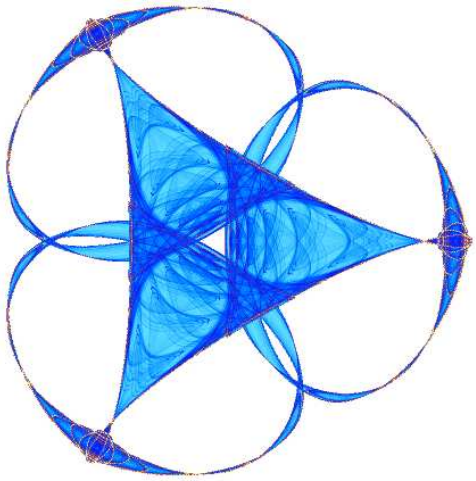
UNIVERSAL SPARSE MODELING

By

Ignacio Ramírez
and
Guillermo Sapiro

IMA Preprint Series # 2303

(March 2010)



INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA
400 Lind Hall
207 Church Street S.E.
Minneapolis, Minnesota 55455–0436

Phone: 612-624-6066 Fax: 612-626-7370
URL: http://www.ima.umn.edu

Universal Sparse Modeling

Ignacio Ramírez and Guillermo Sapiro
Department of Electrical and Computer Engineering
University of Minnesota
{ramir048,guille}@umn.edu

Abstract

Sparse data models, where data is assumed to be well represented as a linear combination of a few elements from a dictionary, have gained considerable attention in recent years, and their use has led to state-of-the-art results in many signal and image processing tasks. It is now well understood that the choice of the sparsity regularization term is critical in the success of such models. In this work, we use tools from information theory, and in particular universal coding theory, to propose a framework for designing sparsity regularization terms which have several theoretical and practical advantages when compared to the more standard ℓ_0 or ℓ_1 ones, and which lead to improved coding performance and accuracy in reconstruction and classification tasks. We also report on further improvements obtained by imposing low mutual coherence and Gram matrix norm on the corresponding learned dictionaries. The presentation of the framework and theoretical foundations is complemented with examples in image denoising and classification.

EDICS: MLR-INFO, SSP-APPL, SSP-SNMD, SSP-SSAN.

I. Introduction

Sparse modeling calls for constructing a succinct representation of some data as a combination of a few typical patterns (*atoms*) learned from the data itself. Significant contributions to the theory and practice of learning such collections of atoms (usually called *dictionaries* or *codebooks*), e.g., [1], [14], [31], and of representing the actual data in terms of them, e.g., [9], [11], [12], have been developed in recent years, leading to state-of-the-art results in many signal and image processing tasks [24], [26], [27], [32]. We refer the reader for example to [5] for a recent review on the subject.

A critical component of sparse modeling is the actual sparsity of the representation, which is controlled by a regularization term (*regularizer* for short) and its associated parameters. The choice of the functional form of the regularizer and its parameters is a challenging task. Several solutions to this problem have

been proposed in the literature, ranging from the automatic tuning of the parameters [20] to Bayesian models, where these parameters are themselves considered as random variables [17], [20], [46].

In this paper we address this challenge by proposing a family of regularizers that are robust under the choice of their parameters. These regularizers are derived using tools from information theory, more specifically, from universal coding theory. The main idea is to consider sparse modeling as a codelength minimization problem, where the regularizers define, through a probability assignment model, the codelength associated with the description of the sparse representation coefficients. We then design these regularizers so that the associated codelength for describing the coefficients generating any possible signal (from a given class of signals), is close to the best possible codelength achievable for that particular instance of coefficients. Encoding schemes with this property, and their associated regularizers, are called *universal* [29], where universality is defined with respect to the set of signals that can be observed in practice in a given application. These concepts will be formally introduced in the paper.

The universal regularizers that we obtain with our framework have several desirable theoretical and practical properties such as statistical consistency, improved robustness to outliers in the data, and lead to a better sparse signal recovery (e.g., decoding of sparse signals in compressive sensing) than ℓ_0 and ℓ_1-based techniques in practice. This new family of models is complemented by imposing incoherence in the learned dictionary. We illustrate with tasks from image processing the power of these new models. Finally, the introduction of tools from universal modeling into the sparse world permits to bring a fundamental and well supported theoretical angle to this very important and popular area of research.

The remainder of this paper is organized as follows: in Section II we introduce the standard framework of sparse modeling. Section III is dedicated to the derivation of our proposed universal modeling framework, while Section IV deals with its implementation. Section V presents experimental results showing the practical benefits of the proposed framework for image representation and classification. Concluding remarks are given in Section VI.

II. Sparse modeling and the need for better models

Let $\mathbf{X} \in \mathbb{R}^{M \times N}$ be a set of N column data samples $\mathbf{x}_j \in \mathbb{R}^M$, $\mathbf{D} \in \mathbb{R}^{M \times K}$ a dictionary of K atoms represented as columns $\mathbf{d}_k \in \mathbb{R}^M$, and $\mathbf{A} \in \mathbb{R}^{K \times N}, \mathbf{a}_j \in \mathbb{R}^K$, a set of reconstruction coefficients such that $\mathbf{X} = \mathbf{D}\,\mathbf{A}$. We use $\mathbf{a}_k^T$ to denote the k-th row of $\mathbf{A}$, which corresponds to the coefficients associated to the k-th atom in $\mathbf{D}$. For each $j = 1, \ldots, N$ we define the *active set* of $\mathbf{a}_j$ as $\mathcal{A}_j = \{k : a_{kj} \neq 0, 1 \leq k \leq K\}$, and $\|\mathbf{a}_j\|_0 = |\mathcal{A}_j|$ as its cardinality. The goal of sparse modeling is to design a dictionary $\mathbf{D}$ such that for all or most data samples $\mathbf{x}_j$, there exists a coefficients vector $\mathbf{a}_j$ such that $\mathbf{x}_j \approx \mathbf{D}\,\mathbf{a}_j$ and $\|\mathbf{a}_j\|_0$ is small

(usually below some threshold $L \ll K$). Formally, we would like to solve the following optimization problem in $(\mathbf{D}, \mathbf{A})$,

$$\min_{\mathbf{D},\mathbf{A}} \sum_{j=1}^{N} \psi(\mathbf{a}_j) \quad \text{s.t.} \quad \|\mathbf{x}_j - \mathbf{D}\,\mathbf{a}_j\|_2^2 \leq \epsilon, \quad j = 1, \ldots, N \tag{1}$$

where $\psi(\cdot)$ is a regularization function, or regularizer, which induces sparsity in the columns of the solution $\mathbf{A}$. Usually the constraint $\|\mathbf{d}_k\|_2 \leq 1,\ k = 1, \ldots, K$, is added, since otherwise we can always decrease the cost function arbitrarily by multiplying $\mathbf{D}$ by a large constant and dividing $\mathbf{A}$ by the same constant. When $\mathbf{D}$ is fixed, the problem of finding a sparse $\mathbf{a}_j$ for each sample $\mathbf{x}_j$ is called sparse coding,

$$\mathbf{a}_j = \arg\min_{\mathbf{a}} \psi(\mathbf{a}_j) \quad \text{s.t.} \quad \|\mathbf{x}_j - \mathbf{D}\,\mathbf{a}_j\|_2^2 \leq \epsilon. \tag{2}$$

Among possible choices of $\psi(\cdot)$ are the ℓ_0 pseudo-norm, $\psi(\cdot) = \|\cdot\|_0$, and the ℓ_1 norm. The former tries to solve directly for the sparsest $\mathbf{a}_j$, but since it is non-convex, it is commonly replaced by the ℓ_1 norm, which is its closest convex approximation. Furthermore, under certain conditions on (fixed) $\mathbf{D}$ and the sparsity of $\mathbf{a}_j$, the solutions to the ℓ_0 and ℓ_1-based sparse coding problems coincide (see for example [6]). The problem (1) is also usually formulated in Lagrangian form,

$$\min_{\mathbf{D},\mathbf{A}} \sum_{j=1}^{N} \|\mathbf{x}_j - \mathbf{D}\,\mathbf{a}_j\|_2^2 + \lambda\psi(\mathbf{a}_j), \tag{3}$$

along with its respective sparse coding problem when $\mathbf{D}$ is fixed,

$$\mathbf{a}_j = \arg\min_{\mathbf{a}} \|\mathbf{x}_j - \mathbf{D}\,\mathbf{a}\|_2^2 + \lambda\psi(\mathbf{a}). \tag{4}$$

Even when the regularizer $\psi(\cdot)$ is convex, the sparse modeling problem, in any of its forms, is jointly non-convex in $(\mathbf{D}, \mathbf{A})$. Therefore, the standard approach to find an approximate solution is to use alternate minimization: starting with an initial dictionary $\mathbf{D}^{(0)}$, we minimize (3) alternatively in $\mathbf{A}$ via (2) or (4) (sparse coding step), and then $\mathbf{D}$ (dictionary update step). The sparse coding step can be solved efficiently when $\psi(\cdot)$ is the ℓ_1 norm, using for example Iterative Shrinkage [11] or LARS [12], or with OMP [28] when the regularizer is the ℓ_0 pseudo-norm. The dictionary update step can be done using for example MOD [14] or K-SVD [1].

A. Interpretations of the sparse coding problem

We now turn our attention to the sparse coding problem: given a fixed dictionary $\mathbf{D}$, for each sample vector $\mathbf{x}_j$, compute the sparsest vector of coefficients $\mathbf{a}_j$ that yields a good approximation of $\mathbf{x}_j$. The sparse coding problem admits several interpretations. What follows is a summary of these interpretations and the insights that they provide into the properties of the sparse models.

1) Model selection in statistics: Using the ℓ_0 norm as $\psi(\cdot)$ in (4) is known in the statistics community as the Akaike's Information Criterion (AIC) when $\lambda = 1$, or the Bayes Information Criterion (BIC) when $\lambda = \frac{1}{2}\log M$, two popular forms of model selection (see [22, Chapter 7]). In this context, the ℓ_1 regularizer was introduced in [38], again as a convex approximation of the above model selection methods, and is commonly known (either in its constrained or Lagrangian forms) as the *Lasso*. Note however that, in the regression interpretation of (4), the interpretation of $\mathbf{D}$ and $\mathbf{X}$ is very different.

2) Compressive sensing: The sparse coding problem also appears, usually in its constrained form (2), at the core of compressive sensing [6]. In this case, the vectors $\mathbf{x}_j$ are often random projections onto the rows of the *sensing matrix* $\mathbf{D}$ of the unknown sparse data to be recovered $\mathbf{a}_j$ (note the very different role of $\mathbf{x}_j$, $\mathbf{a}_j$ and $\mathbf{D}$ in this context). One of the main results in compressive sensing states that, under certain conditions on $\mathbf{D}$ and $\mathbf{X}$, one can obtain the exact solution to the ℓ_0 sparse coding problem by solving the ℓ_1-based problem instead, which can be done using standard optimization techniques [6].

3) Maximum a posteriori: Another interpretation of (4) is that of a maximum a posteriori (MAP) estimation of $\mathbf{a}_j$ in the logarithmic scale, that is

$$\begin{aligned} \mathbf{a}_j &= \arg\max_{\mathbf{a}} \{\log P(\mathbf{a}|\mathbf{x}_j)\} = \arg\max_{\mathbf{a}} \{\log P(\mathbf{x}_j|\mathbf{a}) + \log P(\mathbf{a})\} \\ &= \arg\min_{\mathbf{a}} \{-\log P(\mathbf{x}_j|\mathbf{a}) - \log P(\mathbf{a})\}, \end{aligned} \tag{5}$$

where the observed samples $\mathbf{x}_j$ are assumed to be contaminated with additive, zero mean, IID Gaussian noise with variance σ^2,

$$P(\mathbf{x}_j|\mathbf{a}) \propto e^{-\frac{1}{2\sigma^2}\|\mathbf{x}_j - \mathbf{D}\,\mathbf{a}\|_2^2}, \tag{6}$$

and a *prior probability model* on $\mathbf{a}$ with the form

$$P(\mathbf{a}) \propto e^{-\theta\psi(\mathbf{a})} \tag{7}$$

is considered. The energy term in Equation (4) follows by plugging (6) and (7) into (5) and factorizing $2\sigma^2$ into $\lambda = 2\sigma^2\theta$. According to (5) and (7) we have that the ℓ_1 regularizer corresponds to an IID Laplacian prior with mean 0 and inverse-scale parameter θ, $P(\mathbf{a}) = \prod_{k=1}^{K} \theta e^{-\theta|a_k|} = \theta^K e^{-\theta\|\mathbf{a}\|_1}$, which has a special meaning in signal processing tasks such as image or audio compression. This is due to the widely accepted fact that representation coefficients derived from predictive coding of continuous-valued signals, and, more generally, responses from zero-mean filters, are well modeled using Laplacian distributions. For example, for the special case of DCT coefficients of image patches, an analytical study of this phenomenon is provided in [25], along with further references on the subject.

4) Codelength minimization: Sparse coding, in all its forms, has yet another important interpretation. Suppose that we have a fixed dictionary $\mathbf{D}$ and that we want to use it to compress an image, either losslessly by encoding the reconstruction coefficients $\mathbf{A}$ and the residual $\mathbf{X} - \mathbf{DA}$, or in a lossy manner, by obtaining a good approximation $\mathbf{X} \approx \mathbf{DA}$ and encoding only $\mathbf{A}$. Consider for example the latter case. Most modern compression schemes consist of two parts: a probability assignment stage, where the data, in this case $\mathbf{A}$, is assigned a probability $P(\mathbf{A})$, and an encoding stage, where a code $C(\mathbf{A})$ of length $L(\mathbf{A})$ bits is assigned to the data given its probability, so that $L(\mathbf{A})$ is as short as possible. The techniques known as Arithmetic and Huffman coding provide the best possible solution for the encoding step, which is to approximate the Shannon ideal codelength $L(\mathbf{A}) = -\log P(\mathbf{A})$ [10, Chapter 5]. Therefore, modern compression theory deals with maximizing $P(\mathbf{A})$, or, equivalently, minimizing $-\log P(\mathbf{A})$. Now, to encode $\mathbf{X}$ lossily we obtain coefficients $\mathbf{A}$ such that each data sample $\mathbf{x}_j$ is approximated up to a certain distortion ϵ, $\|\mathbf{x}_j - \mathbf{D}\mathbf{a}_j\|_2 \leq \epsilon$. Therefore, given a model $P(\mathbf{a})$ for a vector of reconstruction coefficients, and assuming that we encode each sample independently, the optimum vector of coefficients $\mathbf{a}_j$ for each sample $\mathbf{x}_j$ will be the solution to the optimization problem

$$\mathbf{a}_j = \arg\min_{\mathbf{a}} \; -\log P(\mathbf{a}) \quad \text{s.t.} \quad \|\mathbf{x}_j - \mathbf{D}\,\mathbf{a}_j\|_2^2 \leq \epsilon, \tag{8}$$

which, for the choice $P(\mathbf{a}) \propto e^{-\psi(\mathbf{a})}$ coincides with the error constrained sparse coding problem (2). Suppose now that we want lossless compression. In this case we also need to encode the reconstruction residual $\mathbf{x}_j - \mathbf{D}\mathbf{a}_j$. Since $P(\mathbf{x}, \mathbf{a}) = P(\mathbf{x}|\mathbf{a})P(\mathbf{a})$, the combined codelength will be

$$L(\mathbf{x}_j, \mathbf{a}_j) = -\log P(\mathbf{x}_j, \mathbf{a}_j) = -\log P(\mathbf{x}_j|\mathbf{a}_j) - \log P(\mathbf{a}_j). \tag{9}$$

Therefore, obtaining the best coefficients $\mathbf{a}_j$ amounts to solving $\min_{\mathbf{a}} L(\mathbf{x}_j, \mathbf{a}_j)$, which is precisely the MAP formulation of (5), which in turn, for proper choices of $P(\mathbf{x}|\mathbf{a})$ and $P(\mathbf{a})$, leads to the Lagrangian form of sparse coding (4).

As one can see, the codelength interpretation of sparse coding is able to unify and interpret both the constrained and unconstrained formulations into one consistent framework. Furthermore, this framework offers a natural and objective measure for comparing the quality of different models $P(\mathbf{x}|\mathbf{a})$ and $P(\mathbf{a})$ in terms of the codelengths obtained. The Laplacian model for $\mathbf{A}$ is widely used in image compression, for example in the JPEG-LS standard, which applies a Laplacian model to describe prediction errors [43].

Before continuing, we need to clarify an important technical detail: Laplacian models, as well as Gaussian models, are probability distributions over $\mathbb{R}$, characterized by continuous probability density functions, $f(a) = F'(a)$, $F(a) = P(x \leq a)$. If the reconstruction coefficients are considered real numbers,

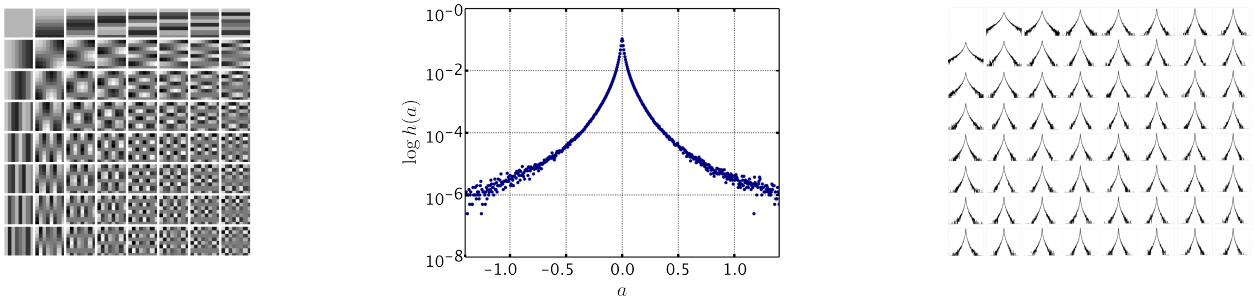


Fig. 1: Standard 8×8 DCT dictionary (left), global empirical distribution of the coefficients in $\mathbf{A}$ (center, log scale) and the $K = 64$ empirical distributions of the coefficients associated to each of the $K = 64$ DCT atoms (right, log scale). The coefficients were obtained from encoding around 10^6 8×8 patches (after removing their DC component) randomly sampled from the Pascal 2006 dataset of natural images [15]. Note the difference in variance as we move from the low frequencies (upper left corner) to the higher frequencies (lower right corner). Note also that even for each single atom, the distribution of the coefficients is not a perfect Laplacian, but something with heavier tails.

under any of these distributions, any instance of $\mathbf{A} \in \mathbb{R}^{K\times N}$ will have measure 0, that is, $P(\mathbf{A}) = 0$. In order to use such distributions as our models for the data, we assume that the coefficients in $\mathbf{A}$ are quantized to a precision Δ, small enough for the density function $f(a)$ to be approximately constant in any interval $[a-\Delta/2, a+\Delta/2]$, $x \in \mathbb{R}$, so that we can approximate $P(a) \approx \Delta f(a),\ a \in \mathbb{R}$. Under these assumptions, $-\log P(a) \approx -\log f(a) - \log \Delta$, and the effect of Δ on the codelength produced by any model is the same. Therefore, we will omit Δ in the sequel, and treat density functions and probability distributions interchangeably as $P(\cdot)$. Of course, in real compression applications, Δ needs to be tuned.

B. The need for a better model

As explained in the previous subsection, the use of the ℓ_1 regularizer implies that all the coefficients in $\mathbf{A}$ share the same Laplacian parameter θ. However, as noted in [25] and references therein, the empirical variance of coefficients associated to different atoms, that is, of the different rows $\mathbf{a}_k^T$ of $\mathbf{A}$, varies greatly with $k = 1 \ldots, K$. This is clearly seen in Figure 1, which shows the empirical distribution of DCT coefficients of 8×8 patches. As the variance of a Laplacian is $2/\theta^2$, different variances indicate different underlying θ. The histogram of the set $\left\{\hat{\theta}_k, k = 1, \ldots, K\right\}$ of estimated Laplacian parameters for each row k, Figure 2(left), shows that this is indeed the case, with significant occurrences of values of $\hat{\theta}$ in a range of 5 to 25.

A first refinement is thus an independent but not identically distributed Laplacian model, where each

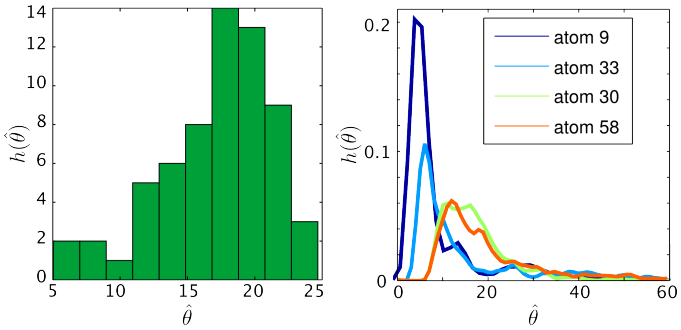


Fig. 2: Left: Histogram of the $K = 64$ different $\hat{\theta}_k$ values obtained by fitting a Laplacian distribution to each row $\mathbf{a}_k^T$ of $\mathbf{A}$ for the data in Figure 1. Note that there are significant occurrences between $\hat{\theta} = 5$ to $\hat{\theta} = 25$, indicating very different widths of the empirical distributions for different atoms, as hinted from Figure 1. Right: Histograms showing the variability of the best local estimations of $\hat{\theta}_k$ for a few rows of $\mathbf{A}$ across different regions of an image. The coefficients $\mathbf{A}$ correspond to the sparse encoding of all 8×8 patches from a single image, in scan-line order. For each k, each value of $\hat{\theta}_k$ was computed from a random contiguous block of 250 samples from $\mathbf{a}_k^T$. The procedure was repeated 4000 times to obtain an empirical distribution. In all cases shown, the wide supports of the empirical distributions indicate that the estimated $\hat{\theta}$ can have very different values, even for the same atom, depending on the region of the data from where the coefficients are taken.

coefficient a_k associated to atom $\mathbf{d}_k$ has a different Laplacian parameter θ_k,

$$P(\mathbf{a}) = \prod_{k=1}^{K} \theta_k e^{-\theta_k |a_k|}. \tag{10}$$

Plugging the model (10) in, for example, (4), results in a weighted ℓ_1 sparse coding formulation,

$$\mathbf{a}_j = \arg\min_{\mathbf{a}} \|\mathbf{x}_j - \mathbf{D}\,\mathbf{a}\|_2^2 + \sum_{k=1}^{K} \lambda_k |a_k|, \tag{11}$$

where $\lambda_k := 2\sigma^2\theta_k$. The weighted ℓ_1 model (11), also called weighted Lasso, has been used for example in [2], [36]. Assume now that $\mathbf{a}_j^0$ is the true underlying coefficients vector generating $\mathbf{x}_j$, and that $\mathcal{A}^0$ is its active set. From a statistics perspective, it has been shown that, with a proper choice of the weights $\{\lambda_k\}_{k=1,\ldots,K}$, the weighted Lasso is an oracle estimator, meaning that $P(\mathcal{A}_j \neq \mathcal{A}_j^0) \to 0$ and that $\mathbf{a}_j \to \mathbf{a}_j^0$ in probability, both limits taken when $M \to \infty$ [46].[1]

From a modeling perspective, instead of a single parameter, the weighted Lasso has K parameters to deal with. This explosion of parameters is often undesirable and, yet, it may not be accurate enough if we consider that real images, or other types of signals such as audio samples, are far from stationary. This

[1]Unfortunately, for sparse modeling and compressive sensing applications, M is relatively small for the results of [46] to be meaningful (the asymptotic results in [46] require that $M \geq K$ so that $\mathbf{D}^T\mathbf{D}$ is positive definite, but we usually have $M \leq K$).

implies that, in reality, even if each atom k is associated to its own θ_k (λ_k), the optimal value of θ_k can have significant local variations at different positions or times. This effect is shown in Figure 2(right), where, for each k, each θ_k was re-estimated several times using samples from different regions of an image, and the histogram of the different estimated values of $\hat{\theta}_k$ was computed. Here again we used the DCT basis as the dictionary $\mathbf{D}$.

The need for a flexible model which at the same time has a small number of parameters leads naturally to Bayesian formulations where the different possible λ_k are "marginalized out" by imposing an hyper-prior distribution on λ, sampling λ using its posterior distribution, and then averaging the estimates obtained with the sampled sparse-coding problems. Examples of this recent line of work, and the closely related Bayesian Compressive Sensing, are developed for example in [23], [39], [45], [44]. Despite of its promising results, the Bayesian approach is often criticized due to the potentially expensive sampling process (something which can be reduced for certain choices of the priors involved [23]), arbitrariness in the choice of the priors, and lack of proper theoretical justification for the proposed models [44].

In this work we pursue the same goal of deriving a more flexible and accurate sparse model than the traditional ones, while avoiding an increase in the number of parameters and the burden of possibly solving several sampled instances of the sparse coding problem. For this, we deploy tools from the very successful information-theoretic field of *universal coding*, which is an extension of the compression scenario summarized above in Section II-A, when the probability model for the data to be described is itself unknown and has to be described as well.

The result is a framework and a corresponding family of models that are as accurate as (11) (or even better, as we will see), while maintaining the computational cost of sparse coding at a small multiple of that required for solving (2) with an ℓ_1-regularizer.

III. Universal models for sparse coding

Following the discussion in the preceding section, we now have several possible scenarios to deal with. First, we may still want to consider a single value of θ to work well for all the coefficients in $\mathbf{A}$, and try to design a sparse coding scheme that does not depend on prior knowledge on the value of θ. Secondly, we can consider an independent (but not identically distributed) Laplacian model where the underlying parameter θ can be different for each atom $\mathbf{d}_k$, $k = 1, \ldots, K$. In the most extreme scenario, we can consider each single coefficient a_{kj} in $\mathbf{A}$ to have its own unknown underlying θ_{kj} and yet, we would like to encode each of these coefficients (almost) as if we knew its hidden parameter.

The first two scenarios are the ones which fit the original purpose of universal coding theory [29],

which is the design of optimal codes for data whose probability models are unknown, and the models themselves are to be encoded as well in the compressed representation. The theory of universal coding also plays a central role in the Minimum Description Length principle (MDL) [3], [21], a powerful model selection tool, of particular relevance in all the applications presented here.

Now we develop the basic ideas and techniques of universal coding applied to the first scenario, where the problem is to describe $\mathbf{A}$ as an IID Laplacian with unknown parameter θ. Assuming a known parametric form for the prior, with unknown parameter θ, leads to the concept of a *model class*. In our case, we consider the class $\mathcal{M} = \{P(\mathbf{A}|\theta) : \theta \in \Theta\}$ of all IID Laplacian models over $\mathbf{A} \in \mathbb{R}^{K\times N}$, where

$$P(\mathbf{A}|\theta) = \prod_{j=1}^{N}\prod_{k=1}^{K} P(a_{kj}|\theta), \quad P(a_{kj}|\theta) = \theta e^{-\theta|a_{kj}|}$$

and $\Theta \subseteq \mathbb{R}^+$. The goal, following the universal coding framework, is to find a probability model $Q(\mathbf{A})$ which can fit $\mathbf{A}$ as well as the model in $\mathcal{M}$ that best fits $\mathbf{A}$ after having observed it. A model $Q(\mathbf{A})$ with this property is called *universal* (with respect to the model $\mathcal{M}$).

For simplicity, in the following discussion we consider the coefficient matrix $\mathbf{A}$ to be arranged as a single long column vector of length $n = K\times N$, $\mathbf{a} = (a_1, \ldots, a_n)$. We also use the letter a without sub-index to denote the value of a random variable representing coefficient values.

First we need to define a criterion for comparing the fitting quality of different models. In universal coding theory this is done in terms of the codelengths $L(\mathbf{a})$ required by each model to describe $\mathbf{a}$.

If the model consists of a single probability distribution $P(\cdot)$, we known from Section II-A4 that the optimum codelength corresponds to $L_P(\mathbf{a}) = -\log P(\mathbf{a})$. Moreover, this relationship defines a one-to-one correspondence between distributions and codelengths, so that for any coding scheme $L_Q(\mathbf{a})$, $Q(\mathbf{a}) = 2^{-L_Q(\mathbf{a})}$. Now suppose that we are restricted to a class of models $\mathcal{M}$, and that we need choose the model $\hat{P} \in \mathcal{M}$ that assigns the shortest codelength to a particular instance of $\mathbf{a}$. We then have that $\hat{P}$ is the model in $\mathcal{M}$ that assigns the maximum probability to $\mathbf{a}$. For a class $\mathcal{M}$ parameterized by θ, this corresponds to $\hat{P} = P(\mathbf{a}|\hat{\theta}(\mathbf{a}))$, where $\hat{\theta}(\mathbf{a})$ is the maximum likelihood estimator (MLE) of the model class parameter θ given $\mathbf{a}$ (we will usually omit the argument and just write $\hat{\theta}$). Unfortunately, we also need to include the value of $\hat{\theta}$ in the description of $\mathbf{a}$ for the decoder to be able to reconstruct it from the code $C(\mathbf{a})$. Thus, we have that any model $Q(\mathbf{a})$ inducing valid codelengths $L_Q(\mathbf{a})$ will have $L_Q(\mathbf{a}) > -\log P(\mathbf{a}|\hat{\theta})$. The overhead of $L_Q(\mathbf{a})$ with respect to $-\log P(\mathbf{a}|\hat{\theta})$ is known as the *codelength regret*,

$$\mathcal{R}(\mathbf{a}, Q) := L_Q(\mathbf{a}) - (-\log P(\mathbf{a}|\hat{\theta}(\mathbf{a}))) = -\log Q(\mathbf{a}) + \log P(\mathbf{a}|\hat{\theta}(\mathbf{a}))).$$

A model $Q(\mathbf{a})$ (or, more precisely, a sequence of models, one for each data length n) is called *universal* if $\mathcal{R}(\mathbf{a}, Q)$ grows sublinearly in n for all possible realizations of $\mathbf{a}$, that is

$$\frac{1}{n}\mathcal{R}(\mathbf{a}, Q) \to 0\,,\ \forall\, \mathbf{a} \in \mathbb{R}^n,$$

so that the codelength regret with respect to the MLE becomes asymptotically negligible.

There are a number of ways to construct universal probability models. The simplest one is the so called *two-part code*, where the data is described in two parts. The first part describes the optimal parameter $\hat{\theta}(\mathbf{a})$ and the second part describes the data according to the model with the value of the estimated parameter $\hat{\theta}$, $P(\mathbf{a}|\hat{\theta}(\mathbf{a}))$. For uncountable parameter spaces Θ, such as a compact subset of $\mathbb{R}$, the value of $\hat{\theta}$ has to be quantized in order to be described with a finite number of bits d. We call the quantized parameter $\hat{\theta}_d$. The regret for this model is thus

$$\mathcal{R}(\mathbf{a}, Q) = L(\hat{\theta}_d) + L(\mathbf{a}|\hat{\theta}_d) - L(\mathbf{a}|\hat{\theta}) = L(\hat{\theta}_d) - \log P(\mathbf{a}|\hat{\theta}_d) - (-\log P(\mathbf{a}|\hat{\theta})).$$

The key for this model to be universal is in the choice of the quantization step for the parameter $\hat{\theta}$, so that both its description $L(\hat{\theta}_d)$, and the difference $-\log P(\mathbf{a}|\hat{\theta}_d) - (-\log P(\mathbf{a}|\hat{\theta}))$, grow sublinearly. This can be achieved by letting the quantization step shrink as $O(1/\sqrt{n})$, requiring then $d = O(0.5 \log n)$ bits to describe $\hat{\theta}_d$, as detailed in the proof of universality of two-part codes given in [21, Theorem 10.1]. In this case the regret for the two-part codes grows as $\frac{\dim(\Theta)}{2} \log n$, where $\dim(\Theta)$ is the dimension of the parameter space Θ.

Another important universal code is the so called *Normalized Maximum Likelihood* (NML) [37]. In this case the universal model $Q^*(\mathbf{a})$ corresponds to the model that minimizes the worst case regret,

$$Q^*(\mathbf{a}) = \min_{Q} \max_{\mathbf{a}} \{-\log Q(\mathbf{a}) + \log P(\mathbf{a}|\hat{\theta}(\mathbf{a}))\},$$

which can be written in closed form as $Q^*(\mathbf{a}) = \frac{P(\mathbf{a}|\hat{\theta}(\mathbf{a}))}{\mathcal{C}(\mathcal{M},n)}$, where the normalization constant $\mathcal{C}(\mathcal{M}, n) := \sum_{\mathbf{a}\in\mathbb{R}^n} P(\mathbf{a}|\hat{\theta}(\mathbf{a}))d\mathbf{a}$ is the value of the minimax regret and depends only on $\mathcal{M}$ and the length of the data n. The fact that $Q^*(\mathbf{a})$ is minimax worst case optimal derives from the fact that it defines a complete uniquely decodable code for all data $\mathbf{a}$ of length n, that is, it satisfies the Kraft inequality with equality[2] $\sum_{\mathbf{a}\in\mathbb{R}^n} 2^{-L_{Q^*}(\mathbf{a})} = 1$. Since every uniquely decodable code with length $L_Q(\mathbf{a})$ has to satisfy the Kraft inequality (see [10, Chapter 5]), if there exists a value of $\mathbf{a}$ such that $L_Q(\mathbf{a}) < L_{Q^*}(\mathbf{a})$ (that is $2^{-L_Q(\mathbf{a})} > 2^{-L_{Q^*}(\mathbf{a})}$), then there must exist at least some $\mathbf{a}'$ for which $L_Q(\mathbf{a}') > L_{Q^*}(\mathbf{a}')$ for the Kraft

[2]Recall that we are actually dealing with (finely) quantized $\mathbf{a}$, so that the expression is really a sum and not an integral. Nevertheless, we write $\mathbf{a} \in \mathbb{R}^n$ since the distributions that we are using are defined over $\mathbb{R}$.

inequality to hold. Therefore the regret of Q for $\mathbf{a}'$ is necessarily greater than $\mathcal{C}(\mathcal{M}, n)$, which shows that Q^* is minimax optimal. Note that the NML model requires that $\mathcal{C}(\mathcal{M}, n)$ be finite, something which is often not the case.

The two previous examples are good for assigning a probability to coefficients $\mathbf{a}$ that have already been computed, but they cannot be used as a model for computing the coefficients themselves since they depend on having observed them in the first place. For this and other reasons that will become clearer later, we concentrate our work on a third important family of universal codes derived from the so called *mixture models* (also called *Bayesian mixtures*). In a mixture model, $Q(\mathbf{a})$ is a convex mixture of all the models $P(\mathbf{a}|\theta)$ in $\mathcal{M}$, indexed by the model parameter θ, $Q(\mathbf{a}) = \int_\Theta P(\mathbf{a}|\theta)w(\theta)d\theta$, where $w(\theta)$ specifies the weight of each model. Being a convex mixture implies that $w(\theta) \geq 0$ and $\int_\Theta w(\theta)d\theta = 1$, thus $w(\theta)$ is itself a probability measure over Θ. We will restrict ourselves to the particular case when $\mathbf{a}$ is considered a sequence of independent random variables,[3]

$$Q(\mathbf{a}) = \prod_{j=1}^{n} Q_j(a_j), \quad Q_j(a_j) = \int_\Theta P(a_j|\theta)w_j(\theta)d\theta, \tag{12}$$

where the mixing function $w_j(\theta)$ can be different for each sample j. An important particular case of this scheme is the so called *Sequential Bayes* code, in which $w_j(\theta)$ is computed sequentially as a posterior distribution based on previously observed samples, that is $w_j(\theta) = P(\theta|a^{(n-1)})$ [21, Chapter 6]. In this work, for simplicity, we restrict ourselves to the case where $w_j(\theta) = w(\theta)$ is the same for all j. The result is an IID model where the probability of each sample a_j is a mixture of some probability measure over $\mathbb{R}$,

$$Q_j(a_j) = Q(a_j) = \int_\Theta P(a_j|\theta)w(\theta)d\theta, \ \forall\, j = 1, \ldots, N. \tag{13}$$

A central result in universal coding theory for IID mixture (Bayesian) codes states that their asymptotic regret is $O(\frac{\dim(\Theta)}{2} \log n)$, thus stating their universality, as long as the weighting function $w(\theta)$ is positive, continuous and unimodal over Θ. This gives us great flexibility on the choice of a weighting function $w(\theta)$ that guarantees universality. Of course, the results are asymptotic and the $o(\log n)$ terms can be large, so that the choice of $w(\theta)$ can have practical impact.

In the following discussion we derive several IID mixture models for the Laplacian model class $\mathcal{M}$. For this purpose, it will be convenient to consider the corresponding one-sided counterpart of the Laplacian, which is the exponential distribution over the absolute value of the coefficients, $|a|$, and then symmetrize back to obtain the final distribution over the signed coefficients a.

[3]More sophisticated models which include dependencies between the elements of $\mathbf{a}$ are out of the scope of this work.

A. The conjugate prior

In general, a closed form solution of (13) if $w(\theta)$ is the conjugate prior of $P(a|\theta)$. When $P(a|\theta)$ is an exponential (one-sided Laplacian), the conjugate prior is the Gamma distribution,

$$w(\theta|\kappa,\beta) = \Gamma(\kappa)^{-1}\theta^{\kappa-1}\beta^{\kappa}e^{-\beta\theta},\ \theta \in \mathbb{R}^{+},$$

where κ and β are its *shape* and *scale* parameters respectively. Plugging this in (13) we obtain the *Mixture of exponentials* model (MOE), which has the following form (see Appendix A for the full derivation),

$$Q_{\text{MOE}}(a|\beta,\kappa) = \kappa\beta^{\kappa}(a+\beta)^{-(\kappa+1)},\ a \in \mathbb{R}^{+}. \tag{14}$$

With some abuse of notation, we will also denote the symmetric distribution on a as MOE,

$$Q_{\text{MOE}}(a|\beta,\kappa) = \frac{1}{2}\kappa\beta^{\kappa}(|a|+\beta)^{-(\kappa+1)},\ a \in \mathbb{R}. \tag{15}$$

Although the resulting prior has two parameters to deal with instead of one, we know from universal coding theory that, in principle, any choice of κ and β will give us a model whose codelength regret is asymptotically small.

Furthermore, being IID models, each coefficient of $\mathbf{a}$ itself is modeled as a mixture of exponentials, which makes the resulting model over $\mathbf{a}$ very well suited to the most flexible scenario where the "underlying" θ can be different for each a_j. In Section V-B we will show that a single MOE distribution can fit each of the K rows of $\mathbf{A}$ better than K separate Laplacian distributions fine-tuned to these rows, with a total of K parameters to be estimated. Thus, not only we can deal with one single unknown θ, but we can actually achieve maximum flexibility with only two parameters (κ and β). This property is particular of the mixture models, and does not apply to the other universal models presented.

Finally, if desired, both κ and β can be easily estimated using the method of moments (see Appendix A). Given sample estimates of the first and second non-central moments, $\hat{\mu}_1 = \frac{1}{n}\sum_{j=1}^{n}|a_j|$ and $\hat{\mu}_2 = \frac{1}{n}\sum_{j=1}^{n}|a_j|^2$, we have that

$$\hat{\kappa} = 2(\hat{\mu}_2 - \hat{\mu}_1^2)/(\hat{\mu}_2 - 2\hat{\mu}_1^2) \quad \text{and} \quad \hat{\beta} = (\hat{\kappa}-1)\hat{\mu}_1. \tag{16}$$

When the MOE prior is plugged into (5) instead of the standard Laplacian, the following new sparse coding formulation is obtained,

$$a_j = \arg\min_{\mathbf{a}} \|\mathbf{x}_j - \mathbf{D}\mathbf{a}\|_2^2 + \lambda_{\text{MOE}} \sum_{k=1}^{K} \log\left(|a_k| + \beta\right), \tag{17}$$

where $\lambda_{\text{MOE}} = 2\sigma^2(\kappa+1)$. An example of the MOE regularizer, and the thresholding function it induces, is shown in Figure 3 (center column) for $\kappa = 2.5, \beta = 0.05$. Smooth, differentiable non-convex regularizers

such as the one in in (17) have become a mainstream robust alternative to the ℓ_1 norm in statistics [16], [46]. Furthermore, it has been shown that the use of such regularizers in regression leads to consistent estimators which are able to identify the relevant variables in a regression model (oracle property) [16]. This is not always the case for the ℓ_1 regularizer, as was proved in [46]. The MOE regularizer has also been recently proposed in the context of compressive sensing [7], where it is conjectured to be better than the ℓ_1-term at recovering sparse signals in compressive sensing applications.[4] This conjecture was partially confirmed recently for non-convex regularizers of the form $\psi(\mathbf{a}) = \|\mathbf{a}\|_r$ with $0 < r < 1$ in [35], [18], and for a more general family of non-convex regularizers including the one in (17) in [42]. In all cases, it was shown that the conditions on the sensing matrix (here $\mathbf{D}$) can be significantly relaxed to guarantee exact recovery if non-convex regularizers are used instead of the ℓ_1 norm, provided that the exact solution to the non-convex optimization problem can be computed. In practice, this regularizer is being used with success in a number of applications here and in [8], [41].[5] Our experimental results in Section V provide further evidence on the benefits of the use of non-convex regularizers, leading to a much improved recovery accuracy of sparse coefficients compared to ℓ_1 and ℓ_0, which in turn yields improvements in applications such as denoising and classification. We also show in Section V that the MOE prior is much more accurate than the standard Laplacian to model the distribution of reconstruction coefficients drawn from a large database of image patches.

B. The Jeffreys prior

The Jeffreys prior for a parametric model class $\mathcal{M} = \{P(a|\theta),\, \theta \in \Theta\}$, is defined as

$$w(\theta) = \frac{\sqrt{I(\theta)}}{\int_\Theta \sqrt{I(\xi)} d\xi},\ \theta \in \Theta, \tag{18}$$

where $I(\theta)$ is the *Fisher information matrix*:

$$I(\theta) = \left\{ E_{P(a|\tilde{\theta})} \left[-\frac{\partial^2}{\partial \tilde{\theta}^2} \log P(a|\tilde{\theta}) \right] \right\} \Bigg|_{\tilde{\theta}=\theta}. \tag{19}$$

The Jeffreys prior is well known in Bayesian theory due to three important properties: it virtually eliminates the hyper-parameters of the model, it is invariant to the original parametrization of the distribution, and it is a "non-informative prior," meaning that it represents well the lack of prior information

[4] In [7], the logarithmic regularizer arises from approximating the ℓ_0 pseudo-norm as an ℓ_1-normalized element-wise sum, without the insight and theoretical foundation here reported.

[5] While these works support the use of such non-convex regularizers, none of them formally derives them using the universal coding framework as in this paper.

on the unknown parameter θ [4]. It turns out that, for quite different reasons, the Jeffreys prior is also of paramount importance in the theory of universal coding. For instance, it has been shown in [3] that the worst case regret of the mixture code obtained using the Jeffreys prior approaches that of the NML as the number of samples n grows. Thus, by using Jeffreys, one can attain the minimum worst case regret asymptotically, while retaining the advantages of a mixture (not needing hindsight of $\mathbf{a}$), which in our case means to be able to use it as a model for computing $\mathbf{a}$ via sparse coding.

For the exponential distribution we have that $I(\theta) = \frac{1}{\theta^2}$. Clearly, if we let $\Theta = (0, \infty)$, the integral in (18) evaluates to ∞. Therefore, in order to obtain a proper integral, we need to exclude 0 and ∞ from Θ (note that this was not needed for the conjugate prior). We choose to define $\Theta = [\theta_1, \theta_2]$, $0 < \theta_1 < \theta_2 < \infty$, leading to

$$w(\theta) = \frac{1}{\ln(\theta_2/\theta_1)} \frac{1}{\theta}, \ \theta \in [\theta_1, \theta_2].$$

The resulting mixture, after being symmetrized around 0, has the following form (see Appendix B):

$$Q_{\text{JOE}}(a|\theta_1, \theta_2) = \frac{1}{2\ln(\theta_2/\theta_1)} \frac{1}{|a|} \left(e^{-\theta_1|a|} - e^{-\theta_2|a|} \right), \ a \in \mathbb{R}^+. \tag{20}$$

We refer to this prior as a *Jeffreys mixture of exponentials* (JOE), and again overload this acronym to refer to the symmetric case as well. Note that although Q_{JOE} is not defined for $a = 0$, its limit when $a \to 0$ is finite and evaluates to $\frac{\theta_2 - \theta_1}{2\ln(\theta_2/\theta_1)}$. Thus, by defining $Q_{\text{JOE}}(0) = \frac{\theta_2 - \theta_1}{2\ln(\theta_2/\theta_1)}$, we obtain a prior that is well defined and continuous for all $a \in \mathbb{R}$. When plugged into (5), we get the JOE-based sparse coding formulation,

$$\min_{\mathbf{a}} \|\mathbf{x}_j - \mathbf{D}\mathbf{a}\|_2^2 + \lambda_{\text{JOE}} \sum_{k=1}^{K} \{\log|a_k| - \log(e^{-\theta_1|a_k|} - e^{-\theta_2|a_k|})\}, \tag{21}$$

where, according to the convention just defined for $Q_{\text{JOE}}(0)$, we define $\psi_{\text{JOE}}(0) := \log(\theta_2 - \theta_1)$. According to the MAP interpretation we have that $\lambda_{\text{JOE}} = 2\sigma^2$, coming from the Gaussian assumption on the approximation error as explained in Section II-A.

As with MOE, the JOE-based regularizer, $\psi_{\text{JOE}}(\cdot) = -\log Q_{\text{JOE}}(\cdot)$, is continuous and differentiable in $\mathbb{R}^+$, and its derivative converges to a finite value at zero, $\lim_{a\to 0} \psi'_{\text{JOE}}(a) = \frac{\theta_2^2 - \theta_1^2}{\theta_2 - \theta_1}$. As we will see later in Section IV, these properties are important to guarantee the convergence of sparse coding algorithms using non-convex priors.

Note from (21) that we can rewrite the JOE regularizer as

$$\psi_{\text{JOE}}(a_k) = \log|a_k| - \log e^{-\theta_1|a|}(1 - e^{-(\theta_2 - \theta_1)|a|}) = \theta_1|a_k| + \log|a_k| - \log(1 - e^{-(\theta_2 - \theta_1)|a_k|}).$$

For sufficiently large $|a_k|$, $\log(1 - e^{-(\theta_2 - \theta_1)|a_k|}) \approx 0$, $\theta_1|a_k| \gg \log|a_k|$, and we have that $\psi_{\text{JOE}}(|a_k|) \approx \theta_1|a_k|$. Thus, for large $|a_k|$, the JOE regularizer behaves like ℓ_1 with $\lambda' = 2\sigma^2\theta_1$. In terms of the

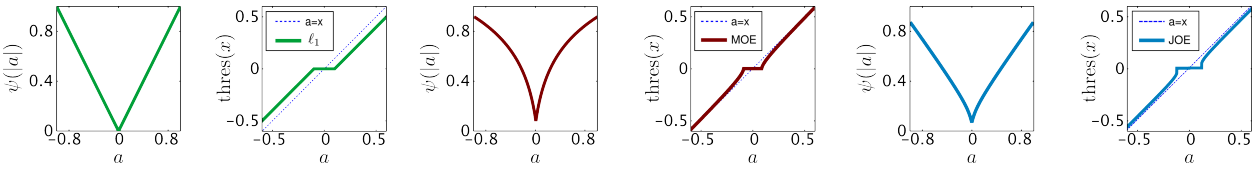


Fig. 3: Left to right: ℓ_1 (green), MOE (red) and JOE (blue) regularizers and their corresponding thresholding functions $\text{thres}(x) := \arg\min_a\{(x-a)^2 + \lambda\psi(|a|)\}$. The unbiasedness of MOE is due to the fact that large coefficients are not shrank by the thresholding function. Also, although the JOE regularizer is biased, the shrinkage of large coefficients can be much smaller than the one applied to small coefficients.

probability model, this means that the tails of the JOE mixture behave like a Laplacian with $\theta = \theta_1$, with the region where this happens determined by the value of $\theta_2 - \theta_1$. The fact that the non-convex region of $\psi_{\text{JOE}}(\cdot)$ is confined to a neighborhood around 0 could help to avoid falling in bad local minima during the optimization (see Section IV for more details on the optimization aspects). Finally, although having Laplacian tails means that the estimated $\mathbf{a}$ will be biased [16], the sharper peak at 0 allows us to perform a more aggressive thresholding of small values, without excessively clipping large coefficients, which leads to the typical over-smoothing of signals recovered using an ℓ_1 regularizer. See Figure 3 (rightmost column) for an example regularizer based on JOE with parameters $\theta_1 = 20, \theta_2 = 100$, and the thresholding function it induces.

The JOE regularizer has two hyper-parameters (θ_1, θ_2) which define Θ and that, in principle, need to be tuned. One possibility is to choose θ_1 and θ_2 based on the physical properties of the data to be modeled, so that the possible values of θ never fall outside of the range $[\theta_1, \theta_2]$. For example, in modeling patches from grayscale images with a limited dynamic range of $[0, 255]$ in a DCT basis, the maximum variance of the coefficients can never exceed 128^2. The same is true for the minimum variance, since unprocessed images always have a small (albeit unnoticeable) base noise.

Having said this, in practice it is advantageous to adjust $[\theta_1, \theta_2]$ to the data at hand. In this case, although no closed form solutions exist for estimating $[\theta_1, \theta_2]$ using maximum likelihood or the method of moments, standard optimization techniques can be easily applied to obtain them. See Appendix B for details.

C. The conditional Jeffreys

A recent approach to deal with the case when the integral over Θ in the Jeffreys prior is improper, is the *conditional Jeffreys* [21, Chapter 11]. The idea is to construct a proper prior, based on the improper

Jeffreys prior and the first few n_0 samples of $\mathbf{a}$, $(a_1, a_2, \ldots, a_{n_0})$, and then use it for the remaining data. The key observation is that although the normalizing integral $\int \sqrt{I(\theta)}d\theta$ in the Jeffreys prior is improper, the unnormalized prior $w(\theta) = \sqrt{I(\theta)}$ can be used as a measure to weight $P(a_1, a_2, \ldots, a_{n_0}|\theta)$,

$$w(\theta) = \frac{P(a_1, a_2, \ldots, a_{n_0}|\theta)\sqrt{I(\theta)}}{\int_\Theta P(a_1, a_2, \ldots, a_{n_0}|\xi)\sqrt{I(\xi)}d\xi}. \tag{22}$$

It turns out that the integral in (22) usually becomes proper for small n_0 in the order of $\dim(\Theta)$. In our case we have that for any $n_0 \geq 1$, the resulting prior is a $\text{Gamma}(\kappa_0, \beta_0)$ distribution with $\kappa_0 := n_0$ and $\beta_0 := \sum_{j=1}^{n_0} a_j$ (see Appendix C for details). Therefore, using the conditional Jeffreys prior in the mixture leads to a particular instance of MOE, which we denote by CMOE (although the functional form is identical to MOE), where the Gamma parameters κ and β are automatically selected from the data. This may explain in part why the Gamma prior performs so well in practice, as we will see in Section V.

Furthermore, we observe that the value of β obtained with this approach (β_0) coincides with the one estimated using the method of moments for MOE if the κ in MOE is fixed to $\kappa = \kappa_0 + 1 = n_0 + 1$. Indeed, if computed from n_0 samples, the method of moments for MOE gives $\beta = (\kappa - 1)\mu_1$, with $\mu_1 = \frac{1}{n_0}\sum a_j$, which gives us $\beta = \frac{n_0+1-1}{n_0}\sum a_j = \beta_0$. It turns out in practice that the value of κ estimated using the method of moments gives a value between 2 and 3 for the type of data that we deal with (see Section V), which is just above the minimum acceptable value for the CMOE prior to be defined, which is $n_0 = 1$. This justifies our choice of $n_0 = 2$ when applying CMOE in practice.

As n_0 becomes large, so does $\kappa_0 = n_0$, and the Gamma prior $w(\theta)$ obtained with this method converges to a Kronecker delta at the mean value of the Gamma distribution, $\delta_{\kappa_0/\beta_0}(\cdot)$. Consequently, when $w(\theta) \approx \delta_{\kappa_0/\beta_0}(\theta)$, the mixture $\int_\Theta P(a|\theta)w(\theta)d\theta$ will be close to $P(a|\kappa_0/\beta_0)$. Moreover, from the definition of κ_0 and β_0 we have that κ_0/β_0 is exactly the maximum likelihood estimator of θ for the Laplacian distribution. Thus, for large n_0, the conditional Jeffreys method approaches the maximum likelihood Laplacian model.

Although from a strict universal coding point of view this is not a problem, for large n_0 the conditional Jeffreys model will loose its flexibility to deal with the case when different coefficients in $\mathbf{A}$ have different underlying θ. On the other hand, using a small n_0 can lead to a prior $w(\theta)$ that is severely overfitted to the local properties of the first samples, which for non-stationary data such as image patches, can be problematic. Ultimately, n_0 defines a trade-off between the degree of flexibility and the accuracy of the resulting model.

IV. Optimization and implementation details

All of the mixture models discussed so far yield non-convex regularizers, rendering the sparse coding problem non-convex in $\mathbf{a}$. It turns out however that these regularizers satisfy certain conditions which make the resulting sparse coding optimization well suited to be approximated using a sequence of successive convex sparse coding problems, a technique known as *Local Linear Approximation* (LLA) [47] (see also [41], [19] for alternative optimization techniques for such non-convex sparse coding problems). In a nutshell, suppose we need to obtain an approximate solution to

$$\mathbf{a}_j = \arg\min_{\mathbf{a}} \|\mathbf{x}_j - \mathbf{D}\,\mathbf{a}\|_2^2 + \lambda \sum_{k=1}^{K} \psi(|a_k|), \tag{23}$$

where $\psi(\cdot)$ is a non-convex function over $\mathbb{R}^+$. At each LLA iteration, we compute $\mathbf{a}_j^{(t+1)}$ by doing a first order expansion of $\psi(\cdot)$ around the K elements of the current estimate $a_{kj}^{(t)}$,

$$\tilde{\psi}_k^{(t)}(|a|) = \psi(|a_{kj}^{(t)}|) + \psi'(|a_{kj}^{(t)}|)\left(|a| - |a_{kj}^{(t)}|\right) = \psi'(|a_{kj}^{(t)}|)|a| + c_k,$$

and solving the convex weighted ℓ_1 problem that results after discarding the constant terms c_k,

$$\begin{aligned}\mathbf{a}_j^{(t+1)} &= \arg\min_{\mathbf{a}} \|\mathbf{x}_j - \mathbf{D}\,\mathbf{a}\|_2^2 + \lambda \sum_{k=1}^{K} \tilde{\psi}_k^{(t)}(|a_k|) \\ &= \arg\min_{\mathbf{a}} \|\mathbf{x}_j - \mathbf{D}\,\mathbf{a}\|_2^2 + \lambda \sum_{k=1}^{K} \psi'(|a_{kj}^{(t)}|)|a_k| = \arg\min_{\mathbf{a}} \|\mathbf{x}_j - \mathbf{D}\,\mathbf{a}\|_2^2 + \sum_{k=1}^{K} \lambda_k^{(t)}|a_k|. \end{aligned} \tag{24}$$

where we have defined $\lambda_k^{(t)} := \lambda\psi'(|a_{kj}^{(t)}|)$. If $\psi'(\cdot)$ is continuous in $(0, +\infty)$, and right-continuous and finite at 0, then the LLA algorithm converges to a stationary point of (23) [46]. These conditions are met for both the MOE and JOE regularizers. Although, for the JOE prior, the derivative $\psi'(\cdot)$ is not defined at 0, it converges to the limit $\frac{\theta_2^2-\theta_1^2}{2(\theta_2-\theta_1)}$ when $|a| \to 0$, which is well defined for $\theta_2 \neq \theta_1$. If $\theta_2 = \theta_1$, the JOE mixing function is a Kronecker delta and the prior becomes a Laplacian with parameter $\theta = \theta_1 = \theta_2$. Therefore we have that for all of the mixture models studied, the LLA method converges to a stationary point. In practice, we have observed that 5 iterations are enough to converge. Thus, the cost of sparse coding, with the proposed non-convex regularizers, is at most 5 times that of a single ℓ_1 sparse coding, and could be less in practice if warm restarts are used to begin each iteration.

Of course we need a starting point $\mathbf{a}_j^{(0)}$, and, being a non-convex problem, this choice will influence the approximation that we obtain. One reasonable choice, used in this work, is to define $a_{kj}^{(0)} = a_0$, $k = 1, \ldots, K, j = 1, \ldots, N$, where a_0 is a scalar so that $\psi'(a_0) = E_w[\theta]$, that is, so that the first sparse coding corresponds to a Laplacian regularizer whose parameter is the average value of θ as given by the mixing prior $w(\theta)$.

Finally, note that although the discussion here has revolved around the Lagrangian or MAP formulation to sparse coding, this technique is also applicable to the constrained formulation of sparse-coding given by Equation (1) for a fixed dictionary $\mathbf{D}$.

Comments on parameter estimation: All the universal models presented so far, with the exception of the conditional Jeffreys, depend on hyper-parameters which in principle should be tuned for optimal performance (remember that they do not influence the universality of the model). If tuning is needed, it is important to remember that the proposed universal models are intended for reconstruction coefficients of *clean data*, and thus their hyper-parameters should be computed from statistics of clean data, or either by compensating the distortion in the statistics caused by noise (see for example [30]). Finally, note that when $\mathbf{D}$ is linearly dependent and $\text{rank}(\mathbf{D}) = \mathbb{R}^M$, the coefficients matrix $\mathbf{A}$ resulting from an exact reconstruction of $\mathbf{X}$ will have many zeroes which are not properly explained by any continuous distribution such as a Laplacian. We sidestep this issue by computing the statistics only from the non-zero coefficients in $\mathbf{A}$. Dealing properly with the case $P(a = 0) > 0$ is beyond the scope of this work.

V. Experimental results

In the following experiments, the testing data $\mathbf{X}$ are 8×8 patches drawn from the Pascal VOC2006 *testing* subset,[6] which are high quality 640×480 RGB images with 8 bits per channel. For the experiments, we converted the 2600 images to grayscale by averaging the channels, and scaled the dynamic range to lie in the $[0, 1]$ interval. Similar results to those shown here are also obtained for other patch sizes.

A. Dictionary learning

For the experiments that follow, unless otherwise stated, we use a "global" overcomplete dictionary $\mathbf{D}$ with $K = 4M = 256$ atoms trained on the full VOC2006 *training* subset using an extension of the model (3), where we add an additional dictionary regularization term to the cost function,[7]

$$\min_{\mathbf{D},\mathbf{A}} \frac{1}{N} \sum_{j=1}^{N} \left\{ \|\mathbf{x}_j - \mathbf{D}\,\mathbf{a}_j\|_2^2 + \lambda\psi(\mathbf{a}_j) \right\} + \mu \left\|\mathbf{D}^T\mathbf{D}\right\|_F^2 . \tag{25}$$

The additional term, $\mu \left\|\mathbf{D}^T\mathbf{D}\right\|_F^2$, encourages incoherence in the learned dictionary, that is, it forces the atoms to be as orthogonal as possible. Dictionaries with lower coherence are well known to have several

[6] http://pascallin.ecs.soton.ac.uk/challenges/VOC/databases.html#VOC2006

[7] While we could have used off-the-shelf dictionaries such as DCT in order to test our universal sparse coding framework, it is important to use dictionaries that lead to the state-of-the-art results in order to show the additional potential improvement of our proposed regularizers.

theoretical advantages such as improved ability to recover sparse signals [11], [40], and faster and better convergence to the solution of the sparse coding problems (1) and (3) [13]. In particular, the technique of imposing incoherence using the formulation in (25) was introduced in [33], where it was shown to lead to improvements in a variety of sparse modeling applications, including the ones discussed below.

For dictionary learning we use an ℓ_1 regularizer with $\lambda = 0.1$, which is typical in sparse coding applications, producing dictionaries $\mathbf{D}$ that lead to state-of-the-art results [1], [26]. For the incoherence we use $\mu = 1$, which has also been observed to be a good value in practice for this case. See [1], [26], [33] for details on the optimization of (3) and (25).

B. MOE *as a prior for sparse coding coefficients*

We begin by comparing the performance of the Laplacian and MOE models (priors) for fitting a single global distribution to the whole matrix $\mathbf{A}$. We compute $\mathbf{A}$ using (1) with $\epsilon \approx 0$ and then, following the discussion in Section IV, restrict our study to the nonzero elements of $\mathbf{A}$.

The empirical distribution of $\mathbf{A}$ is plotted in Figure 4(left), along with the best fitting Laplacian, MOE, JOE, and a particularly good example of the conditional Jeffreys (CMOE) distributions.[8] The MLE for the Laplacian fit is $\hat{\theta} = N_1 / \|\mathbf{A}\|_1 = 27.2$ (here N_1 is the number of nonzero elements in $\mathbf{A}$). For MOE, using (16), we obtained $\kappa = 2.8$ and $\beta = 0.07$. For JOE, $\theta_1 = 2.4$ and $\theta_2 = 371.4$. According to the discussion in Section III-C, we used the value $\kappa = 2.8$ obtained using the method of moments for MOE as a hint for choosing $n_0 = 2$ ($\kappa_0 = n_0 + 1 = 3 \approx 2.8$), yielding $\beta_0 = 0.07$, which coincides with the β obtained using the method of moments. As observed in Figure 4(left), in all cases the proposed mixture models fit the data better, significantly better for both Gamma-based mixtures, MOE and CMOE, and slightly better for JOE. This is further confirmed by the Kullback-Leibler divergences (KL) obtained in each case. As a reference, the empirical entropy of the quantized data is $H(\mathbf{A}) = 3.00$ bits. Note that JOE fails to significantly improve on the Laplacian mode due to the excessively large estimated range $[\theta_1, \theta_2]$. In this sense, it is clear that the JOE model is very sensitive to its hyper-parameters, and a better and more robust estimation would be needed for it to be useful in practice.

Given these results, we concentrate the rest of the experiments on the best (and simplest) case which is the MOE prior (which, as detailed above, can be derived from the conditional Jeffreys as well, thus representing both approaches).

[8] To compute the empirical distribution, we quantized the elements of $\mathbf{A}$ uniformly in steps of 2^{-8}, which for the amount of data available, gives us enough detail and at the same time reliable statistics for all the quantized values.

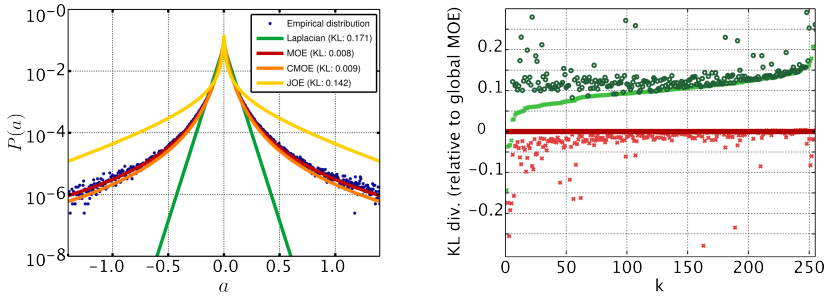


Fig. 4: Left: Empirical distribution of the sparse coding coefficients a for image patches (blue dots), best fitting Laplacian (green), MOE (red), CMOE (orange) and JOE (yellow) distributions. The Laplacian is clearly not fitting the tails properly, and is not sufficiently peaked at zero either. The two models based on a Gamma prior, MOE and CMOE, provide an almost perfect fit. The fitted JOE is the most sharply peaked at 0, but doest not fit the tails as tight as desired. The KL divergences are 0.171 bits for the Laplacian, 0.008 for MOE, 0.009 for CMOE and 0.142 for JOE. Right: KL divergence for the best fitting global Laplacian (dark green), per-atom Laplacian (light green), global MOE (dark red) and per-atom MOE (light red), relative to the KL divergence between the globally fitted MOE distribution and the empirical distribution. The horizontal axis represents the indexes of each atom, $k = 1, \ldots, K$, ordered according to the difference in KL divergence between the global MOE and the per-atom Laplacian model. Note how the global MOE outperforms both the global and per-atom Laplacian models in all but the first 4 cases.

From Figure 2(right) we know that the optimal $\hat{\theta}$ varies locally across different regions, thus, we expect the mixture models to perform well also on a per-atom basis. This is confirmed in Figure 4(right), where we show, for each row $\mathbf{a}^k, k = 1, \ldots, K$, the difference in KL divergence between the globally fitted MOE distribution and the best per-atom fitted MOE, the globally fitted Laplacian, and the per-atom fitted Laplacians respectively. The horizontal axis, which represents atom index, is sorted by increasing KL divergence difference between the per-atom fitted Laplacians and the globally fitted MOE distribution. As can be observed, the KL obtained with the *global* MOE is significantly smaller than the global Laplacian in all cases, and even the *per-atom* Laplacians in most of the cases. This shows that MOE, with only two parameters (which can be easily estimated, as detailed in the text), is a much better model than K Laplacians (requiring K critical parameters) fitted specifically to the coefficients associated to each atom. Whether these modeling improvements have a practical impact is explored in the next experiments.

C. Recovery of noisy sparse signals

Here we compare the active set recovery properties of the MOE prior, compared to those of the ℓ_1-based one, on data for which the sparsity assumption $|\mathcal{A}_j| \leq L$ holds exactly for all j, for a small L. To this end, we obtain sparse approximations to each sample $\mathbf{x}_j$ using the ℓ_0-based Orthogonal Matching Pursuit

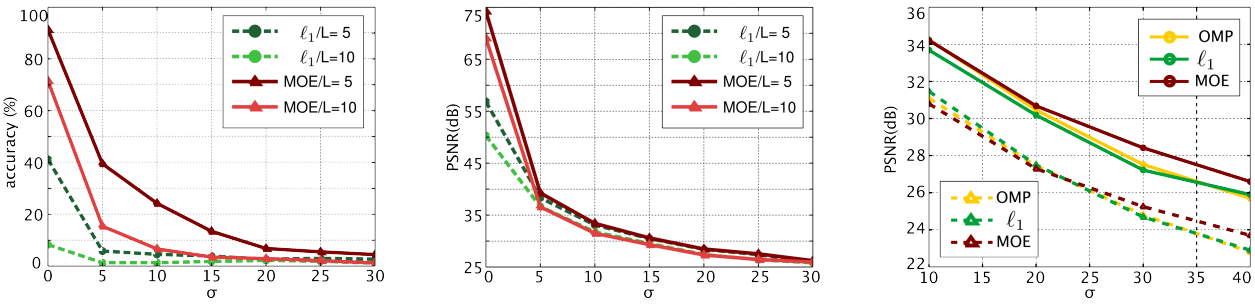


Fig. 5: Left: active set recovery accuracy of ℓ_1 and MOE for truly sparse $\mathbf{A}$, as defined in Section V-C, for two sparsity levels L, as a function of the noise variance σ. The improvement of the proposed model over ℓ_1 is a factor of 5 to 9. Center: PSNR of the recovered signals with respect to the clean signals, again for the case where $\mathbf{A}$ is truly sparse. In this case significant improvements can be observed at the low SNR range, specially for highly sparse ($L = 5$) signals. The performance of both methods is practically the same for $\sigma \geq 10$. Right: denoising of real data. Results are relative to OMP using a globally trained dictionary. The proposed mixture model offers the best of both worlds: good patch-level reconstruction and final image reconstruction after averaging. While loosing to the ℓ_1-based dictionary solution for $\sigma = 10$ at the patch level, the gap is closed after averaging. For high noise, the improved robustness of our method gives significantly better results.

algorithm (OMP) on $\mathbf{D}$ [28], and record the resulting active sets $\mathcal{A}_j$ as ground truth. The data is then contaminated with additive Gaussian noise of variance σ and the recovery is performed by solving (1) for $\mathbf{A}$ with $\epsilon = CM\sigma^2$ and either the ℓ_1 or the MOE-based regularizer for $\psi(\cdot)$. We use $C = 1.32$, which is a standard value in denoising applications (see for example [27]).

For each sample j, we measure the error of each method in recovering the active set as the Hamming distance between the true and estimated support of the corresponding reconstruction coefficients. The accuracy of the method is then given as the percentage of the samples for which this error falls below a certain threshold T. Results are shown in Figure 5(left) for $L = (5, 10)$ and $T = (2, 4)$ respectively, for various values of σ. Note the very significant improvement obtained with the proposed model.

Given the estimated active set $\mathcal{A}_j$, the estimated clean patch is obtained by projecting $\mathbf{x}_j$ onto the subspace defined by the atoms that are active according to $\mathcal{A}_j$, using least squares (which is the standard procedure for denoising once the active set is determined). We then measure the PSNR of the estimated patches with respect to the true ones. The results are shown in Figure 5(center), again for various values of σ. As can be observed, the MOE-based recovery is significantly better, specially in the high SNR range. Notoriously, the more accurate active set recovery of MOE does not seem to improve the denoising performance in this case. However, as we will see next, it does make a difference when denoising real life signals, as well as for classification tasks.

D. Recovery of real signals with simulated noise

This experiment is an analogue to the previous one, when the data are the original natural image patches (without forcing exact sparsity). Since for this case the sparsity assumption is only approximate, and no ground truth is available for the active sets, we compare the different methods in terms of their denoising performance.

A critical strategy in image denoising is the use of overlapping patches, where for each pixel in the image a patch is extracted with that pixel as its center. The patches are denoised independently as M-dimensional signals and then recombined into the final denoised images by simple averaging. Although this consistently improves the final result in all cases, the improvement is very different depending on the method used to denoise the individual patches. Therefore, we now compare the denoising performance of each method at two levels: individual patches and final image.

To denoise each image, the global dictionary learned as described in Section V-A is further adapted to the noisy image patches to be processed for a few iterations of the learning algorithm, using ℓ_1 or MOE respectively plugged into (25), with a coherence penalty of $\mu = 1$. For each of these dictionaries, we solve (2) with $\epsilon = CM\sigma^2$, [27], and two different regularizers: first, with the corresponding regularization term used to learn the dictionary, and then with $\psi(\cdot) = \|\cdot\|_0$ (using OMP). We do the latter since it is the one that often yields better denoising results in practice (see [1], [27] for examples). The results for the four combinations are summarized in Table I. For the final image results, we also include results from [1] (K-SVD), when available, as a reference in the table, and a graph corresponding to the average final image performance in Figure 5(right). We also show results for two sample images in Figure 6.

For $\sigma < 20$, at the patch level, our method gives results that are very close to those of the (learning+coding) ℓ_1+OMP combination, which in turn are worse than the ℓ_1+ℓ_1 solution by 0.4dB on average. However, both our method and OMP give better results after patch averaging. This is a well known empirical effect which is attributed in part to the biasedness of ℓ_1-regularized regression. The OMP, on the other hand, gives less accurate per-patch estimations, but the artifacts it produces are more "random" and thus they are canceled-out when the patches are averaged to build the final image. In this sense, the MOE-based method offers the best of both worlds, since it gives better results than OMP at a patch level, and less biased, which helps in the final result. The benefits become clearer as we move into the high noise region $\sigma \geq 30$, where we obtain significant improvements in all cases, which are clearly visible in Figure 6, whereas differences of less than $0.3 - 0.5$dB are hard to distinguish even by close inspection.

Fig. 6: Sample image denoising results. Top: Barbara, $\sigma = 30$. Bottom: Boats, $\sigma = 40$. From left to right: noisy, ℓ_1/OMP, ℓ_1/ℓ_1, MOE/MOE. The reconstruction obtained with the proposed model is more accurate, as evidenced by a better reconstruction of the texture in Barbara, and sharp edges in Boats, and does not produce the artifacts seen in both the ℓ_1 and ℓ_0 reconstructions, which appear as black/white speckles all over Barbara, and ringing on the edges in Boats.

E. Classification with universal sparse models

In this section we apply our proposed universal models to a classification problem where each sample $\mathbf{x}_j$ is to be assigned a class label $y_j = 1, \ldots, c$, which serves as an index to the set of possible classes, $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_c\}$. The sample problem presented is the Graz'02 bike detection problem,[9] where each pixel of each testing image has to be classified as either background or as part of a bike. We follow the classification framework given in [34], where a statistical model for each class $\mathcal{C}_i$ is learned by adapting a dictionary $\mathbf{D}_i$ to the training samples belonging to that class using (25). Given the learned dictionaries, the label assigned to a sample $\mathbf{x}_j$ is given by maximum likelihood, $y_j = \arg\max_i P(\mathbf{x}_j|\mathcal{C}_i)$, where the likelihood of $\mathbf{x}_j$ belonging to class $\mathcal{C}_i$ is given in terms of its sparse code given $\mathbf{D}_i$, $\mathbf{a}_j^i$, computed via (4). Assuming that $\mathbf{a}_j^i$ is unique for all $i = 1, \ldots, c$, we have that $P(\mathbf{x}_j|\mathcal{C}_i) = P(\mathbf{x}_j, \mathbf{a}_j^i|\mathbf{D}_i)$, so that

$$\begin{aligned} y_j &= \arg\max_i P(\mathbf{x}_j, \mathbf{a}_j^i|\mathbf{D}_i) = \arg\max_i P(\mathbf{x}_j|\mathbf{a}_j^i, \mathbf{D}_i) P(\mathbf{a}_j^i|\mathbf{D}_i) \\ &= \arg\min_i \left\{ \arg\min_{\mathbf{a}} \{ -\log P(\mathbf{x}_j|\mathbf{a}, \mathbf{D}_i) - \log P(\mathbf{a}|\mathbf{D}_i) \} \right\}. \end{aligned} \tag{26}$$

[9] http://lear.inrialpes.fr/people/marszalek/data/ig02/

	$\sigma = 10$					$\sigma = 30$				
learning	ℓ_1		MOE		[1]	ℓ_1		MOE		[1]
coding	ℓ_1	ℓ_0	MOE	ℓ_0		ℓ_1	ℓ_0	MOE	ℓ_0	
camera	30.7/**34.0**	**31.6**/33.5	30.5/33.9	30.8/**34.0**	-	24.2/27.2	24.5/27.0	24.5/27.6	**24.7/27.7**	-
barbara	30.9/**34.7**	**31.3**/33.9	30.3/34.3	30.7/34.4	34.4	23.0/25.5	23.0/25.2	24.2/27.0	**24.4/27.9**	-
boat	30.7/33.6	**31.0**/32.9	30.4/33.7	30.7/**33.8**	33.6	24.7/27.2	24.5/26.8	25.1/28.0	**25.2/28.2**	-
goldhil	30.4/33.4	**30.6**/33.1	30.3/33.4	30.5/**33.5**	-	25.2/27.8	24.7/27.6	25.3/28.1	**25.4/28.2**	-
lena	32.3/35.3	**32.5**/34.7	32.0/**35.5**	32.2/**35.5**	**35.5**	26.4/29.5	26.2/29.1	26.5/30.0	**26.6/30.2**	-
peppers	31.9/**34.8**	**32.1**/34.4	31.7/**34.8**	31.9/**34.8**	34.3	26.4/29.0	26.3/28.9	26.5/29.7	**26.6/29.8**	-
AVER.	*31.1/34.2*	***31.5****/33.7*	*30.8/34.2*	*31.1/**34.3***	-	*24.8/27.5*	*24.7/27.2*	*25.2/28.4*	***25.4/28.6***	-
	$\sigma = 20$					$\sigma = 40$				
learning	ℓ_1		MOE		[1]	ℓ_1		MOE		[1]
coding	ℓ_1	ℓ_0	MOE	ℓ_0		ℓ_1	ℓ_0	MOE	ℓ_0	
camera	26.7/29.6	**27.2**/29.6	26.6/29.9	26.8/**30.0**	-	21.7/24.7	22.6/25.3	22.8/25.6	**23.0/25.8**	-
barbara	26.8/30.4	**27.2**/30.3	26.5/30.4	26.8/30.6	**30.8**	21.6/24.3	21.6/24.5	22.6/25.5	**22.7/25.7**	-
boat	27.1/30.1	27.0/29.9	27.0/30.2	**27.2**/30.3	**30.4**	22.6/25.5	22.6/25.6	**23.6**/26.3	**23.6/26.4**	-
goldhil	27.1/29.4	26.8/28.8	27.1/30.0	**27.2/30.1**	-	23.7/26.2	23.4/26.2	**24.2/26.7**	**24.2/26.7**	-
lena	**28.7**/32.1	28.6/31.5	28.5/32.2	**28.7**/32.3	**32.4**	24.3/27.6	24.2/27.6	**25.0**/28.2	**25.0/28.3**	-
peppers	**28.8**/31.9	28.7/31.6	28.5/**32.0**	28.7/**32.0**	30.8	23.5/26.7	23.8/27.0	24.8/27.8	**24.9/27.9**	-
AVER.	***27.5****/30.5*	***27.5****/30.2*	*27.3/30.7*	***27.5/30.8***	-	*22.8/25.7*	*22.9/25.9*	*23.7/26.6*	***23.8/26.7***	-

TABLE I: Denoising results: in each table, each column shows the denoising performance of a learning+coding combination. Results are shown in pairs, where the left number is the PSNR between the clean and recovered individual patches, and the right number is the PSNR between the clean and recovered images. Best results are in bold. Our method gives the best results in all cases at both the patch and image levels for $\sigma \geq 30$, on average for $\sigma = 20$, and in most cases at the image level for $\sigma = 10$.

Since the decision depends on the measured likelihoods, it is clear that the overall detection scheme would benefit from accurate models for $P(\mathbf{x}|\mathbf{a}, \mathbf{D}_i)$ and $P(\mathbf{a}|\mathbf{D}_i)$. The idea is then to use a universal model for $P(\mathbf{a}|\mathbf{D}_i)$, as here proposed, instead of the implied Laplacian model used in [34].

In the Graz'02 dataset, each of the pixels can belong to one of two classes: bike or background. On each of the training images (which by convention are the first 150 even-numbered images), we are given a mask that tells us whether each pixel belongs to a bike or to the background. We then train a dictionary for bike patches and another for background patches. Patches that contain pixels from both classes are assigned to the class corresponding to the majority of their pixels.

In Figure 7 we show the *precision vs. recall curves* obtained with the detection framework when either the ℓ_1 or the MOE regularizers are used for learning $\mathbf{D}$ using (25), and then computing $\mathcal{R}_i$ based on

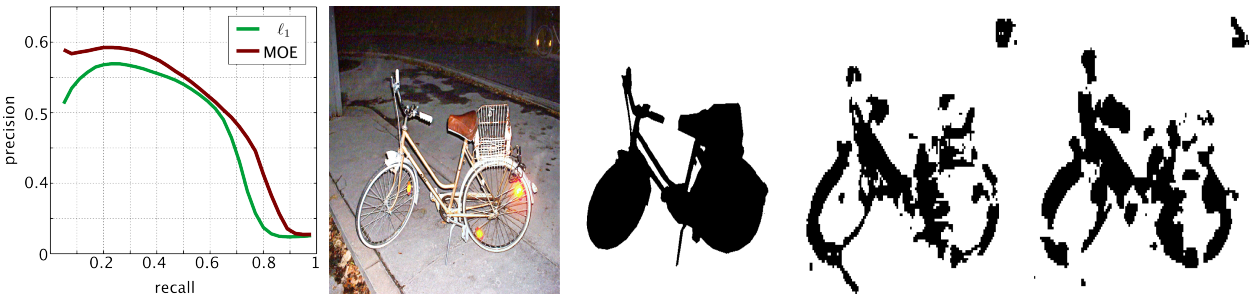


Fig. 7: Classification results. Left to right: precision vs. recall curve, a sample image from the Graz'02 dataset, its ground truth, and the corresponding estimated maps obtained with ℓ_1 and MOE for a fixed threshold. The precision vs. recall curve shows that the mixture model gives a better precision in all cases. In the example, the classification obtained with MOE yields less false positives and more true positives than the one obtained with ℓ_1.

the respective learned dictionary $\mathbf{D}$. The parameters for the ℓ_1 prior (λ), the MOE model (λ_{MOE}) and the incoherence term (μ) were all adjusted by cross validation. The only exception is the MOE parameter β, which was chosen based on the fitting experiment as $\beta = 0.07$. As can be seen, the MOE-based model outperforms the ℓ_1 in this classification task as well, giving a better precision for all recall values.

VI. Concluding remarks

A framework for designing sparse modeling priors was introduced in this work, using tools from universal coding. The priors obtained lead to models with both theoretical and practical advantages over the traditional ℓ_0 and ℓ_1-based ones. In all derived cases, the designed non-convex problems are suitable to be efficiently (approximately) solved via a few iterations of (weighted) ℓ_1 subproblems. We also showed that these priors are able to fit the empirical distribution of sparse codes of image patches significantly better than the traditional IID Laplacian model, and even the non-identically distributed independent Laplacian model where a different Laplacian parameter is adjusted to the coefficients associated to each atom, thus showing the flexibility and accuracy of these proposed models. The additional flexibility, furthermore, comes at a small cost of only 2 parameters to be easily tuned (either (κ, β) in the MOE model, or (θ_1, θ_2) in the JOE model), instead of K (dictionary size), as in weighted Lasso models. The additional accuracy of the proposed models was shown to have significant practical impact in active set recovery of sparse signals, image denoising, and classification applications. Compared to the Bayesian approach, we avoid the potential burden of solving several sampled sparse problems, or being forced to use a conjugate prior for computational reasons (although in our case, *a fortiori*, the conjugate prior does

provide us with a good model). Finally, for the presented models, parameter estimation is straightforward to implement and very fast to compute. Overall, as demonstrated in this paper, the introduction of information theory tools can lead to formally addressing critical aspects of sparse modeling.

Future work in this direction includes the design of priors that take into account the nonzero mass at $a = 0$ that appears in overcomplete models, and online learning of the model parameters from noisy data, following for example the technique in [30]. We also aim at applying this framework for model selection via the MDL principle.

ACKNOWLEDGMENTS

Work partially supported by NGA, ONR, ARO, NSF and FUNDACIBA-ANTEL. We wish to thank Julien Mairal for providing us with his fast sparse modeling toolbox, SPAMS.[10] We also thank Federico Lecumberry for his participation on the incoherent dictionary learning method, and helpful comments.

APPENDIX

A. *Derivation of the MOE model*

For the MOE case we have

$$P(a|\theta) = \theta e^{-\theta a}, \qquad \text{and} \qquad w(\theta|\kappa,\beta) = \frac{1}{\Gamma(\kappa)}\theta^{\kappa-1}\beta^{\kappa}e^{-\beta\theta},$$

which, when plugged into (13), gives

$$Q(a|\beta,\kappa) \;=\; \int_{\theta=0}^{\infty} \theta e^{-\theta a}\frac{1}{\Gamma(\kappa)}\theta^{\kappa-1}\beta^{\kappa}e^{-\beta\theta}d\theta \;=\; \frac{\beta^{\kappa}}{\Gamma(\kappa)}\int_{\theta=0}^{\infty} e^{-\theta(a+\beta)}\theta^{\kappa}d\theta.$$

After the change of variables $u := (a+\beta)\theta$ ($u(0) = 0$, $u(\infty) = \infty$), the integral can be written as

$$\begin{aligned} Q(a|\beta,\kappa) \;&=\; \frac{\beta^{\kappa}}{\Gamma(\kappa)}\int_{\theta=0}^{\infty} e^{-u}\left(\frac{u}{a+\beta}\right)^{k}\frac{du}{a+\beta} \;=\; \frac{\beta^{\kappa}}{\Gamma(\kappa)}(a+\beta)^{-(\kappa+1)}\int_{\theta=0}^{\infty} e^{-u}u^{k}du \\ &=\; \frac{\beta^{\kappa}}{\Gamma(\kappa)}(a+\beta)^{-(\kappa+1)}\Gamma(\kappa+1) \;=\; \frac{\beta^{\kappa}}{\Gamma(\kappa)}(a+\beta)^{-(\kappa+1)}\kappa\Gamma(\kappa), \end{aligned}$$

obtaining $Q(a|\beta,\kappa) = \kappa\beta^{\kappa}(a+\beta)^{-(\kappa+1)}$, since the integral on the second line is precisely the definition of $\Gamma(\kappa+1)$. The symmetrization is obtained by substituting a by $|a|$ and dividing the normalization constant by two, $Q(|a||\beta,\kappa) = 0.5\kappa\beta^{\kappa}(|a|+\beta)^{-(\kappa+1)}$.

[10] http://www.di.ens.fr/willow/SPAMS/

The mean of the MOE distribution (which is defined only for $\kappa > 1$) can be easily computed using integration by parts,

$$\begin{aligned}\mu(\beta,\kappa) &= \kappa\beta^\kappa \int_0^\infty \frac{u}{(u+\beta)^{(\kappa+1)}} du = \kappa\beta \left[-\frac{u}{\kappa(u+\beta)^\kappa}\Big|_0^\infty + \frac{1}{\kappa}\int_0^\infty \frac{du}{(u+\beta)^k}\right] \\ &= \beta^\kappa \left(-\frac{1}{(\kappa-1)(u+\beta)^{(k-1)}}\Big|_0^\infty \right) = \frac{\beta}{\kappa-1}\end{aligned}$$

In the same way, it is easy to see that the non-central moments of order i are $\mu_i = \frac{\beta}{\binom{\kappa-1}{i}}$.

The MLE estimates of κ and β can be obtained using any nonlinear optimization technique such as Newton method, using for example the estimates obtained with the method of moments as a starting point. In practice, however, we have not observed any significant improvement in using the MLE estimates over the moments-based ones.

B. Derivation of the constrained Jeffreys (JOE) model

In the case of the exponential distribution, the Fisher Information Matrix in (19) evaluates to

$$I(\theta) = \left\{ E_{P(\cdot|\tilde\theta)} \left[\frac{\partial^2}{\partial\tilde\theta^2}(-\log\theta + \theta\log a)\right]\right\}\Big|_{\tilde\theta=\theta} = \left\{ E_{P(\cdot|\tilde\theta)} \left[\frac{1}{\tilde\theta^2}\right]\right\}\Big|_{\tilde\theta=\theta} = \frac{1}{\theta^2}.$$

By plugging this result into (18) with $\Theta = [\theta_1,\theta_2]$, $0<\theta_1<\theta_2<\infty$ we obtain $w(\theta) = \frac{1}{\ln(\theta_2/\theta_1)}\frac{1}{\theta}$. We now derive the (one-sided) JOE probability density function by plugging this $w(\theta)$ in (13),

$$\begin{aligned}Q(a) &= \int_{\theta_1}^{\theta_2} \theta e^{-\theta a} \frac{1}{\ln(\theta_2/\theta_1)}\frac{d\theta}{\theta} = \frac{1}{\ln(\theta_2/\theta_1)}\int_{\theta_1}^{\theta_2} e^{-\theta a} d\theta \\ &= \frac{1}{\ln(\theta_2/\theta_1)}\left(-\frac{1}{a}e^{-\theta a}\Big|_{\theta_1}^{\theta_2}\right) = \frac{1}{\ln(\theta_2/\theta_1)}\frac{1}{a}\left(e^{-\theta_1 a} - e^{-\theta_2 a}\right).\end{aligned}$$

Although $Q(a)$ cannot be evaluated at $a=0$, the limit for $a\to 0$ exists and is finite, so we can just define $Q(0)$ as this limit, which is

$$\begin{aligned}\lim_{a\to 0} Q(a) &= \lim_{a\to 0}\frac{1}{\ln(\theta_2/\theta_1)a}\left[1-\theta_1 a + o(a^2) - (1-\theta_2 a + o(a^2))\right] \\ &= \lim_{a\to 0}\frac{1}{\ln(\theta_2/\theta_1)a}(\theta_2-\theta_1)a = \frac{\theta_2-\theta_1}{\ln(\theta_2/\theta_1)}.\end{aligned}$$

Again, if desired, parameter estimation can be done for example using maximum likelihood (via nonlinear optimization), or using the method of moments. However, in this case, the method of moments does not provide a closed form solution for (θ_1,θ_2). The non-central moments of order i are

$$\mu_i = \int_0^{\infty+} \frac{a^i}{\ln(\theta_2/\theta_1)}\frac{1}{a}\left[e^{-\theta_1 a} - e^{-\theta_1 a}\right] da = \frac{1}{\ln(\theta_2/\theta_1)}\left\{\int_0^{+\infty} a^{i-1}e^{-\theta_1 a} da - \int_0^{+\infty} a^{i-1}e^{-\theta_2 a} da\right\}. \tag{27}$$

For $i=1$, both integrals in (27) are trivially evaluated, yielding $\mu_1 = \frac{1}{\ln(\theta_2/\theta_1)}(\theta_1^{-1} - \theta_2^{-1})$. For $i>1$, these integrals can be solved using integration by parts:

$$\begin{aligned}\mu_i^+ &= \int_0^{+\infty} a^{i-1}e^{-\theta_1 a}da = a^{i-1}\frac{1}{(-\theta_1)}e^{-\theta_1 a}\Big|_0^{+\infty} - \frac{1}{(-\theta_1)}(i-1)\int_0^{+\infty} a^{i-2}e^{-\theta_1 a}da \\ \mu_i^- &= \int_0^{+\infty} a^{i-1}e^{-\theta_2 a}da = a^{i-1}\frac{1}{(-\theta_2)}e^{-\theta_2 a}\Big|_0^{+\infty} - \frac{1}{(-\theta_2)}(i-1)\int_0^{+\infty} a^{i-2}e^{-\theta_2 a}da,\end{aligned}$$

where the first term in the right hand side of both equations evaluates to 0 for $i>1$. Therefore, for $i>1$ we obtain the recursions

$$\mu_i^+ = \frac{i-1}{\theta_1}\mu_{i-1}^+, \quad \mu_i^- = \frac{i-1}{\theta_2}\mu_{i-1}^-,$$

which, combined with the result for $i=1$, gives the final expression for all the moments of order $i>0$

$$\mu_i = \frac{(i-1)!}{\ln(\theta_2/\theta_1)}\left(\frac{1}{\theta_1^i} - \frac{1}{\theta_2^i}\right),\ i=1,2,\ldots.$$

In particular, for $i=1$ and $i=2$ we have

$$\theta_1 = \left(\ln(\theta_2/\theta_1)\mu_1 + \theta_2^{-1}\right)^{-1}, \quad \theta_2 = \left(\ln(\theta_2/\theta_1)\mu_2 + \theta_1^{-2}\right)^{-1},$$

which, when combined, give us

$$\theta_1 = \frac{2\mu_1}{\mu_2 + \ln(\theta_2/\theta_1)\mu_1^2}, \quad \theta_2 = \frac{2\mu_1}{\mu_2 - \ln(\theta_2/\theta_1)\mu_1^2}. \tag{28}$$

One possibility is to solve the nonlinear equation $\theta_2/\theta_1 = \frac{\mu_2+\ln(\theta_2/\theta_1)\mu_1^2}{\mu_2-\ln(\theta_2/\theta_1)\mu_1^2}$ for $u=\theta_1/\theta_2$ by finding the roots of the nonlinear equation $u = \frac{\mu_2+\ln u\mu_1^2}{\mu_2-\ln u\mu_1^2}$ and choosing one of them based on some side information. Another possibility is to simply fix the ratio θ_2/θ_1 beforehand and solve for θ_1 and θ_2 using (28).

C. Derivation of the conditional Jeffreys (CMOE) model

The conditional Jeffreys method defines a proper prior $w(\theta)$ by assuming that n_0 samples from the data to be modeled $\mathbf{a}$ were already observed. Plugging the Fisher information for the exponential distribution, $I(\theta)=\theta^{-2}$, into (22) we obtain

$$w(\theta) = \frac{P(a^{n_0}|\theta)\theta^{-1}}{\int_\Theta P(a^{n_0}|\xi)\xi^{-1}d\xi} = \frac{(\prod_{j=1}^{n_0}\theta e^{-\theta a_j})\theta^{-1}}{\int_0^{+\infty}(\prod_{j=1}^{n_0}\xi e^{-\xi a_j})\xi^{-1}d\xi} = \frac{\theta^{n_0-1}e^{-\theta\sum_{j=1}^{n_0}a_j}}{\int_0^{+\infty}\xi^{n_0-1}e^{-\xi\sum_{j=1}^{n_0}a_j}d\xi}.$$

Denoting $S_0 = \sum_{j=1}^{n_0} a_j$ and performing the change of variables $u := S_0\xi$ we obtain

$$w(\theta) = \frac{\theta^{n_0-1}e^{-S_0\theta}}{S_0^{-n_0}\int_0^{+\infty}u^{n_0-1}e^{-u}du} = \frac{S_0^{n_0}\theta^{n_0-1}e^{-S_0\theta}}{\Gamma(n_0)},$$

where the last equation derives from the definition of the Gamma function, $\Gamma(n_0)$. We see that the resulting prior $w(\theta)$ is a Gamma distribution Gamma(κ_0,β_0) with $\kappa_0=n_0$ and $\beta_0=S_0=\sum_{j=1}^{n_0}a_j$.

REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein. The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations. *IEEE Trans. SP*, 54(11):4311–4322, Nov. 2006.

[2] D. Angelosante and G. Giannakis. RLS-weighted lasso for adaptive estimation of sparse signals. In *IEEE ICASSP, Taipei, Taiwan*, April 2009.

[3] A. Barron, J. Rissanen, and B. Yu. The minimum description length principle in coding and modeling. *IEEE Trans. IT*, 44(6):2743–2760, 1998.

[4] J. Bernardo and A. Smith. *Bayesian Theory*. Wiley, 1994.

[5] A. Bruckstein, D. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, Feb. 2009.

[6] E. J. Candès. Compressive sampling. *Proc. of the International Congress of Mathematicians*, 3, Aug. 2006.

[7] E. J. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *J. Fourier Anal. Appl.*, 14(5):877–905, Dec. 2008.

[8] R. Chartrand. Fast algorithms for nonconvex compressive sensing: MRI reconstruction from very few data. In *IEEE ISBI*, June 2009.

[9] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[10] T. Cover and J. Thomas. *Elements of information theory*. John Wiley and Sons, Inc., 2 edition, 2006.

[11] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. on Pure and Applied Mathematics*, 57:1413–1457, 2004.

[12] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

[13] M. Elad. Optimized projections for compressed-sensing. *IEEE Trans. SP*, 55(12):5695–5702, Dec. 2007.

[14] K. Engan, S. Aase, and J. Husoy. Multi-frame compression: Theory and design. *Signal Processing*, 80(10):2121–2140, Oct. 2000.

[15] M. Everingham, A. Zisserman, C. Williams, and L. Van Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results. http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf.

[16] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal Am. Stat. Assoc.*, 96(456):1348–1360, Dec. 2001.

[17] M. Figueiredo. Adaptive sparseness using Jeffreys prior. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Adv. NIPS*, pages 697–704. MIT Press, Dec. 2001.

[18] S. Foucart and M. Lai. Sparsest solutions of underdetermined linear systems via ℓ_q-minimization for $0 < q \leq 1$. *Applied and Computational Harmonic Analysis*, 3(26):395–407, 2009.

[19] G. Gasso, A. Rakotomamonjy, and S. Canu. Recovering sparse signals with non-convex penalties and DC programming. *IEEE Trans. SP*, 57(12):4686–4698, 2009.

[20] R. Giryes, Y. Eldar, and M. Elad. Automatic parameter setting for iterative shrinkage methods. In *IEEE 25-th Convention of Electronics and Electrical Engineers in Israel (IEEEI'08)*, Dec. 2008.

[21] P. Grünwald. *The Minimum Description Length Principle*. MIT Press, June 2007.

[22] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, 2 edition, Feb. 2009.

[23] S. Ji, Y. Xue, and L. Carin. Bayesian compressive sensing. *IEEE Trans. SP*, 56(6):2346–2356, 2008.

[24] B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Trans. PAMI*, 27(6):957–968, 2005.

[25] E. Lam and J. Goodman. A mathematical analysis of the DCT coefficient distributions for images. *IEEE Trans. IP*, 9(10):1661–1666, 2000.

[26] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Adv. NIPS*, volume 21, Dec. 2009.

[27] J. Mairal, G. Sapiro, and M. Elad. Learning multiscale sparse representations for image and video restoration. *SIAM MMS*, 7(1):214–241, April 2008.

[28] S. Mallat and Z. Zhang. Matching pursuit in a time-frequency dictionary. *IEEE Trans. SP*, 41(12):3397–3415, 1993.

[29] N. Merhav and M. Feder. Universal prediction. *IEEE Trans. IT*, 44(6):2124–2147, Oct. 1998.

[30] G. Motta, E. Ordentlich, I. Ramirez, G. Seroussi, and M. Weinberger. The DUDE framework for grayscale image denoising. Technical report, HP laboratories, 2009. http://www.hpl.hp.com/techreports/2009/HPL-2009-252.html.

[31] B. Olshausen and D. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.

[32] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng. Self-taught learning: transfer learning from unlabeled data. In *ICML*, pages 759–766, June 2007.

[33] I. Ramirez, F. Lecumberry, and G. Sapiro. Universal priors for sparse modeling. In *CAMSAP*, Dec. 2009.

[34] I. Ramírez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *CVPR*, June 2010.

[35] R. Saab, R. Chartrand, and O. Yilmaz. Stable sparse approximation via nonconvex optimization. In *ICASSP*, April 2008.

[36] T. Shimamura, S. Imoto, R. Yamaguchi, and S. Miyano. Weighted lasso in graphical gaussian modeling for large gene network estimation based on microarray data. In *Genome Informatics*, volume 19, pages 142–153, 2007.

[37] Y. Shtarkov. Universal sequential coding of single messages. *Probl. Inform. Transm.*, 23(3):3–17, July 1987.

[38] R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.

[39] M. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning*, 1:211–244, 2001.

[40] J. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. IT*, 50(10):2231–2242, Oct. 2004.

[41] J. Trzasko and A. Manduca. Highly undersampled magnetic resonance image reconstruction via homotopic ℓ_0-minimization. *IEEE Trans. MI*, 28(1):106–121, Jan. 2009.

[42] J. Trzasko and A. Manduca. Relaxed conditions for sparse signal recovery with general concave priors. *IEEE Trans. SP*, 57(11):4347–4354, 2009.

[43] M. Weinberger, G. Seroussi, and G. Sapiro. The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. *IEEE Trans. IP*, 9, 2000.

[44] D. Wipf, J. Palmer, and B. Rao. Perspectives on sparse bayesian learning. In *Adv. NIPS*, Dec. 2003.

[45] D. Wipf and B. Rao. An empirical bayesian strategy for solving the simultaneous sparse approximation problem. *IEEE Trans. IP*, 55(7-2):3704–3716, 2007.

[46] H. Zou. The adaptive LASSO and its oracle properties. *Journal Am. Stat. Assoc.*, 101:1418–1429, 2006.

[47] H. Zou and R. Li. One-step sparse estimates in nonconcave penalized likelihood models. *Annals of Statistics*, 36(4):1509–1533, 2008.