

M*-REGULARIZED DICTIONARY LEARNING

MATHIEU BARRÉ AND ALEXANDRE D'ASPREMONT

ABSTRACT. Classical dictionary learning methods simply normalize dictionary columns at each iteration, and the impact of this basic form of regularization on generalization performance (e.g. compression ratio on new images) is unclear. Here, we derive a tractable performance measure for dictionaries in compressed sensing based on the low M^* bound and use it to regularize dictionary learning problems. We detail numerical experiments on both compression and inpainting problems and show that this more principled regularization approach consistently improves reconstruction performance on new images.

1. INTRODUCTION

Dictionary learning seeks to decompose signals on a few atoms using a dictionary learned from the data set, instead of a predefined one formed by e.g. wavelet transforms [Mallat, 1999]. This learning approach has significantly improved state-of-the-art performance on various signal processing tasks such as image denoising [Elad and Aharon, 2006] or inpainting [Mairal et al., 2009] for example. Dictionary learning is an inherently hard problem and the references above use alternating minimization to find good solutions. Furthermore, in all these cases, the dictionary learning problem is only regularized by a simple normalization constraint on the matrix columns. Our main point here is that beyond its simplicity, it is unclear how this normalization affects dictionary performance. *Classical methods thus learn dictionaries without proper regularization, which can hurt generalization performance.*

In a similar vein, structured acquisition seeks to design dictionaries to maximize signal recovery performance while satisfying design constraints [Boyer et al., 2016, 2017] in e.g. magnetic resonance imaging. This means for example ensuring samples follow a continuous path in Fourier space for compressed sensing MRI [Lustig et al., 2008]. More recently, structured acquisition procedures in e.g. [Chauffert et al., 2014, Boyer et al., 2016, 2017] use a sampling approach based on the results of [Lustig et al., 2008, Candes and Plan, 2011].

Instead of this sampling approach, we focus on producing a tractable metric of dictionary performance and use it to regularize dictionary learning problems. In the classical compressed sensing setting, we let $A \in \mathbb{R}^{m \times n}$ be a full rank matrix, we are given m observations Ax_0 of a signal $x_0 \in \mathbb{R}^n$, and we seek to decode it by solving

$$\begin{aligned} & \text{minimize} && \text{Card}(x) \\ & \text{subject to} && Ax = Ax_0, \end{aligned} \tag{1}$$

in the variable $x \in \mathbb{R}^n$. Problem (1) is combinatorially hard, but under certain conditions on the matrix A (see e.g. [Candès and Tao, 2005, Donoho and Tanner, 2005, Kashin and Temlyakov, 2007, Cohen et al., 2009]), we can reconstruct the signal by solving instead

$$\begin{aligned} & \text{minimize} && \|x\|_1 \\ & \text{subject to} && Ax = Ax_0, \end{aligned} \tag{2}$$

which is a convex problem in the variable $x \in \mathbb{R}^n$. Given a sensing matrix A , [Candès and Tao, 2005, Donoho and Tanner, 2005] showed that there is a *recovery threshold* k such that solving problem (2) will always recover the solution to (1) provided the signal has at most k nonzero coefficients. While many results allow us to bound k with high probability for certain classes of random matrices A , computing this threshold k given A is a hard problem [Bandeira et al., 2013, Wang et al., 2016, Weed, 2017].

Date: October 8, 2018.

Our first objective here is to *learn dictionary matrices* A to reconstruct a sample data set with minimal loss, while maximizing the recovery threshold k of the dictionary matrix. Several relaxations have been derived to approximate the recovery threshold [d’Aspremont and El Ghaoui, 2011, Juditsky and Nemirovski, 2011, d’Aspremont et al., 2014], but these approximations typically only certify recovery up to signal size \sqrt{k} when the optimal threshold is k . In fact, there is substantial evidence that this is the best that can be achieved in polynomial time, in the regimes that are relevant for compressed sensing. It can be shown for example that certifying recovery using the restricted isometry property is equivalent to solving a sparse PCA problem, which is hard in a broad sense by reduction of the planted-clique problem [Berthet and Rigollet, 2013].

On the other hand, a simple result in Kashin and Temlyakov [2007] shows that the sparse recovery threshold satisfies $k \geq S(A)^{-2}$, where $S(A)$ is the radius of a section of the ℓ_1 -ball by the nullspace of the sampling matrix A . Directly approximating the radius of a convex polytope is of course hard [Freund and Orlin, 1985, Lovasz and Simonovits, 1992], but the low- M^* bound in e.g. [Pajor and Tomczak-Jaegermann, 1986] shows that we can accurately quantify the performance of a slightly enlarged sampling matrix (A, G) , with high probability, where G is a matrix containing a few additional Gaussian samples.

Our contribution here is twofold. In the spirit of smoothed analysis [Spielman and Teng, 2004], we first show how to use the recovery threshold of the perturbed sampling matrix (A, G) as a proxy for the performance of the original dictionary A . We then use the M^* of this perturbed matrix to regularize dictionary learning problems using an alternating minimization algorithm, to improve generalization performance.

We detail numerical experiments on both image compression and inpainting problems, comparing the PSNR of reconstructed images both inside the training set and on new images. We observe that M^* -regularized dictionary learning often significantly improves reconstruction PSNR and SSIM compared to classically normalized dictionary learning algorithms, both inside and outside the training set.

The paper is organized as follows. In Section 2 we recall the low- M^* bounds and its application to sparse recovery. In Section 3, we detail our M^* -regularized dictionary learning algorithms. Finally, we detail numerical experiments in Section 4.

2. LOW M^* DICTIONARIES

Our starting point is the following result by Kashin and Temlyakov [2007], linking signal recovery thresholds and the radius of a section of the ℓ_1 ball. We will see that this last quantity is hard to estimate but provides accurate bounds on the recovery threshold k of a given sampling matrix A .

Proposition 2.1. [Kashin and Temlyakov, 2007, Th. 2.1] *Given a coding matrix $A \in \mathbb{R}^{m \times n}$, suppose that there is some $k > 0$ such that*

$$S(A) \triangleq \sup_{Ax=0} \frac{\|x\|_2}{\|x\|_1} \leq \frac{1}{\sqrt{k}} \quad (3)$$

then $x^{\text{LP}} = x_0$ if $\text{Card}(x_0) \leq k/4$, and

$$\|x_0 - x^{\text{LP}}\|_1 \leq 4 \min_{\{\text{Card}(y) \leq k/16\}} \|x_0 - y\|_1$$

where x^{LP} solves the ℓ_1 -recovery problem in (2) and x_0 is the original signal.

This result means that the ℓ_1 -minimization problem in (2) will recover exactly all sparse signals x_0 satisfying $\text{Card}(x_0) \leq k/4$ and that the ℓ_1 reconstruction error for other signals will be at most four times larger than the ℓ_1 error corresponding to the best possible approximation of x_0 by a signal of cardinality at most $k/16$. The quantity $\sup_{Ax=0} \|x\|_2/\|x\|_1$ is the radius of a section of the ℓ_1 ball written

$$\mathcal{K}(A) \triangleq \{x \in \mathbb{R}^n : \|x\|_1 \leq 1, Ax = 0\} \quad (4)$$

and thus controls the recovery threshold of matrix A , i.e. the largest signal size that can provably be recovered using the observations in A .

2.1. Deterministic Approximation Bounds on the Radius. In line with previous results producing bounds on RIP and nullspace property constant which also control the recovery threshold k [d’Aspremont and El Ghaoui, 2011, Juditsky and Nemirovski, 2011, d’Aspremont et al., 2014], we now derive convex relaxations to efficiently approximate the radius defined in (3). As discussed above, we will see that the approximation bounds for these relaxations are relatively coarse.

2.2. Semidefinite relaxation. We now show how to compute tractable bounds on the ratio

$$S(A) = \max_{Ax=0} \frac{\|x\|_2}{\|x\|_1},$$

defined in (3). We first formulate a semidefinite relaxation of this problem as follows.

Lemma 2.2. Let $A \in \mathbb{R}^{m \times n}$,

$$S(A)^2 \leq SDP(A) \equiv \max_{\substack{\text{Tr}(A^T AX)=0 \\ \|X\|_1 \leq 1, X \succeq 0}} \text{Tr} X \quad (5)$$

where $SDP(A)$ is computed by solving a semidefinite program in the variable $X \in \mathbf{S}_n$.

Proof. Writing $X = xx^T$, we have

$$S(A)^2 = \max_{\substack{\text{Tr}(A^T AX)=0, \|X\|_1^2 \leq 1, \\ \text{Rank}(X)=1, X \succeq 0}} \text{Tr} X$$

and dropping the rank constraint yields the desired result. ■

We now connect the value of $S(A)$ with that of the function $\alpha_1(A)$ defined in [Juditsky and Nemirovski, 2011, d’Aspremont and El Ghaoui, 2011] as

$$\alpha_1(A) \equiv \max_{Ax=0} \frac{\|x\|_\infty}{\|x\|_1}, \quad (6)$$

which can be computed by solving either a linear program [Juditsky and Nemirovski, 2011] or a semidefinite program [d’Aspremont and El Ghaoui, 2011]. The following lemma bounds $S(A)$ using $\alpha_1(A)$.

Lemma 2.3. Let $A \in \mathbb{R}^{m \times n}$, we have

$$\alpha_1(A) \leq S(A) \leq \sqrt{SDP(A)} \leq \sqrt{\alpha_1(A)}$$

Proof. The first inequality simply follows from $\|x\|_\infty \leq \|x\|_2$, the second from Lemma 2.2. If X solves (5), $\text{Tr}(A^T AX) = 0$ implies $AX = 0$, which means that the columns of X are in the nullspace of A . By definition of $\alpha_1(A)$, we then have $X_{ii} = \|X_i\|_\infty \leq \alpha_1(A) \|X_i\|_1$, hence $\text{Tr}(X) \leq \alpha_1(A) \|X\|_1 \leq \alpha_1(A)$, which yields the desired result. ■

The following proposition shows that if a matrix allows recovery of all signals of cardinality less than k^* , then the SDP relaxation above will efficiently certify recovery of all signals up to cardinality $O(k^*/\sqrt{n})$. This is a direct extension of Lemma 2.3 and Proposition 2.1.

Proposition 2.4. Suppose $A \in \mathbb{R}^{m \times n}$ satisfies condition (3) for some $k > 0$, the semidefinite relaxation will satisfy

$$S(A) \leq \sqrt{SDP(A)} \leq k^{-\frac{1}{4}} \quad (7)$$

and the semidefinite relaxation will certify exact decoding of all signals of cardinality at most \sqrt{k} .

Proof. From Lemma 2.3, we know that $\alpha_1 \leq S(A)$ hence $\sqrt{SDP(A)} \leq \sqrt{S(A)}$. We conclude using Proposition 2.1. ■

We can produce a second proof of this last result, which uses the norm ratio in (3) directly.

Proposition 2.5. Suppose $A \in \mathbb{R}^{m \times n}$ satisfies condition (3) for some $k > 0$, the semidefinite relaxation will satisfy

$$S(A) \leq \sqrt{SDP(A)} \leq k^{-\frac{1}{4}} \quad (8)$$

and the semidefinite relaxation will certify exact decoding of all signals of cardinality at most \sqrt{k} .

Proof. If X solves the SDP relaxation in (5), then the rows of X are in the nullspace of A , and satisfy $\|X_i\|_2 \leq \|X_i\|/\sqrt{S}$. Then, with $\|X\|_1$,

$$\mathbf{Tr} X \leq \sum_{i=1}^n \|X_i\|_\infty \leq \sum_{i=1}^n \|X_i\|_2 \leq \frac{\|X\|_1}{\sqrt{S}} \leq \frac{1}{\sqrt{S}}$$

hence the desired result. ■

Note that we are not directly using $X \succeq 0$ in this last proof, so the approximation ratio also holds for a linear programming bound written

$$LP(A) \equiv \begin{array}{ll} \max. & \mathbf{Tr} X \\ \text{s.t.} & AX = 0 \\ & \|X\|_1 \leq 1 \end{array} \quad (9)$$

We now show that the $k^{-1/4}$ bound is typically the best we can hope for from the relaxation in (5).

Proposition 2.6. Suppose $A \in \mathbb{R}^{m \times n}$ with $n = 2m$, then

$$\frac{1}{\sqrt{2n}} \leq SDP(A) \quad (10)$$

and the semidefinite relaxation will certify exact decoding of all signals of cardinality at most $O(\sqrt{m})$.

Proof. Let Q be the orthoprojector on the nullspace of A . We have $Q \succeq 0$, $\mathbf{Tr}(Q) = m$, $\|Q\|_F = \sqrt{m}$ and $\|Q\|_1 \leq \sqrt{n^2} \|Q\|_F \leq n\sqrt{m}$, which means that $X = Q/(n\sqrt{m})$ is a feasible point of the SDP relaxation in (5) with $\mathbf{Tr} X = \sqrt{m}/n = 1/\sqrt{2n}$ which yields the required bound on the optimal value of (5). ■

This means that if the matrix A allows exact recovery of signals with up to (an unknown number) k nonzero coefficients, then our relaxation will only certify recovery of signals with cardinality $O(\sqrt{k})$. The fact that approximating the recovery threshold k is hard is not entirely surprising, indeed k in (3) is the Euclidean radius of the centrally symmetric polytope $\{x \in \mathbb{R}^n : Ax = 0, \|x\|_1 \leq 1\}$. Computing the radius of generic convex polytopes is NP-Complete [Freund and Orlin, 1985, Lovasz and Simonovits, 1992, Gritzmann and Klee, 1993, Brieden et al., 2001]. In particular, Lovasz and Simonovits [1992] show that if we only have access to an oracle for K , then there is no randomized polynomial time algorithm to compute the radius of a convex body K within a factor $n^{1/4}$. In that sense, the approximation ratio obtained above is optimal. This question is also directly connected to that of efficiently testing Kashin decompositions (see [Szarek, 2010, §4.1] for a discussion).

Here of course, we have some additional structural information on the set K (it is a section of the ℓ_1 ball) so there is a (slight) possibility that this bound could be improved. On the other hand, in the next section, we will see that if we are willing to add a few random experiments to A , then the radius can be bounded with high probability by a randomized polynomial time algorithm.

2.3. Probabilistic Approximation Bounds on the Radius. Proposition 2.1 links the sparse recovery threshold k of a matrix A and the radius of the polytope $\{x \in \mathbb{R}^n : Ax = 0, \|x\|_1 \leq 1\}$. In this section, we first recall some classical results from geometric functional analysis and use these to quantify the sparse recovery thresholds of arbitrary matrices A .

2.3.1. *Dvoretzky's Theorem.* We first recall some concentration results on the sphere as well as classical results in geometric functional analysis which control, in particular, the radius of *random* sections of the ℓ_1 ball (i.e. where A is chosen randomly). Let σ be the unique rotation invariant probability measure on the unit sphere \mathbb{S}^{n-1} of \mathbb{R}^n , and $\|\cdot\|_K$ be a norm on \mathbb{R}^n with unit ball K , then

$$\sigma \{x \in \mathbb{S}^{n-1} : |||x|| - M(K)| \geq tM(K)\} \leq e^{-k(K)t^2} \quad (11)$$

with

$$k(K) = cn \left(\frac{M(K)}{b(K)} \right)^2 \quad (12)$$

where $c > 0$ is a universal constant, and

$$M(K) = \int_{\mathbb{S}^{n-1}} \|x\| d\sigma(x) \quad \text{and} \quad b(K) = \sup_{x \in \mathbb{S}^{n-1}} \|x\|. \quad (13)$$

Klartag and Vershynin [2007] call $k(K)$ the *Dvoretzky dimension* of the convex set K . Part of the proof of Dvoretzky's theorem states that random sections of K with dimension $k = k(K)$ are approximately spherical with high probability (w.r.t. the uniform measure on the Grassman $\mathcal{G}_{n,k}$). We write B_p^n the ℓ_p ball of \mathbb{R}^n .

Theorem 2.7. (*General Dvoretzky*) *In a Banach space with unit ball K , let $E \subset \mathbb{R}^n$ be a subspace of dimension $l \leq k(K)$ defined in (12), chosen uniformly at random w.r.t. to the Haar measure on $\mathcal{G}_{n,k}$, then*

$$\frac{c_1}{M(K)} (B_2^n \cap E) \subset (K \cap E) \subset \frac{c_2}{M(K)} (B_2^n \cap E)$$

with probability $1 - e^{-c_3 l}$, where $c_1, c_2, c_3 > 0$ are absolute constants.

Proof. See [Milman and Schechtman, 1986, §4] or [Vershynin, 2011, Th. 6.4] for example. ■

This result means that random sections of convex bodies with dimension k are approximately spherical with high probability. Milman and Schechtman [1997] show that the threshold $k(K)$ is sharp in the sense that random sections of dimension greater than $k(K)$ are typically not spherical. Because projections of sphere are spheres, there is thus a phase transition at $k(K)$: random sections of K become increasingly spherical until they reach dimension $k(K)$ below which they are approximately spherical with high probability.

2.3.2. *Low M^* Bounds.* The radius follows a similar phase transition, and the following result characterizes its behavior as the dimension of the subspace decreases (we write K^* the polar of K).

Theorem 2.8. (*Low M^**) *In a Banach space with unit ball K , let $E \subset \mathbb{R}^n$ be a subspace of codimension k chosen uniformly at random w.r.t. to the Haar measure on $\mathcal{G}_{n,n-k}$, then*

$$\text{radius}(K \cap E) \leq c \sqrt{\frac{n}{k}} M(K^*)$$

with probability $1 - e^{-k}$, where $c > 0$ is an absolute constant.

Proof. See [Pajor and Tomczak-Jaegermann, 1986] for example. ■

The value of $M(K^*)$ is known for many convex bodies, including l_p balls. In particular, $(B_1^n)^* = B_\infty^n$ and $M(B_\infty^n) \sim \sqrt{\log n/n}$ asymptotically. This means that random sections of the ℓ_1 ball with dimension $n - k$ have radius bounded by

$$\text{radius}(B_1^n \cap E) \leq c \sqrt{\frac{\log n}{k}}$$

with high probability, where c is an absolute constant (a more precise analysis allows the log term to be replaced by $\log(n/k)$). This, combined with the result of Proposition 2.1 is one way to prove optimal bounds on the sparse recovery threshold of random matrices A , and we will apply it below to characterize the performance of randomly perturbed deterministic ones.

2.3.3. *Sparse Recovery Thresholds.* Proposition 2.1 shows that the sparse recovery threshold associated with the m linear observations stored in $A \in \mathbb{R}^{m \times n}$, i.e. the largest signal cardinality for which all sparse signals can be recovered exactly by solving the ℓ_1 -minimization problem in (2), is given by the radius of the centrally symmetric convex polytope $\{x \in \mathbb{R}^n : Ax = 0, \|x\|_1 \leq 1\}$ with

$$k \geq \frac{1}{\text{radius}(\{x \in \mathbb{R}^n : Ax = 0, \|x\|_1 \leq 1\})^2} \quad (14)$$

By homogeneity, this is also equivalent to producing lower bounds on $\|Fy\|_1$ over \mathbb{S}^{n-m-1} , the unit sphere of \mathbb{R}^{n-m} .

The low M^* estimate in Proposition 2.8 together with the fact that $M(B_\infty^n) \sim \sqrt{\log n/n}$ and Proposition 2.1 then show that choosing m linear samples $A \in \mathbb{R}^{m \times n}$ uniformly at random in the Grassman will allow us, with high probability, to recover all signals with at most $\frac{m}{c \log n}$ nonzero coefficients, by solving the ℓ_1 minimization problem in (2) (again, the log term can be replaced by $\log(n/k)$).

As we have seen above, finding good compressed sensing experiments means finding matrices $A \in \mathbb{R}^{m \times n}$ for which $\|Fy\|_1$ is almost spherical, where F is any basis for the nullspace of A . Bad matrices are matrices for which the norm ball of $\|Fy\|_1$ is much closer to a cross-polytope. The key difficulty in high dimensions is that all centrally symmetric convex bodies look like spheres, except for a few ‘‘spikes’’ (or tentacles in Vershynin [2011]) with negligible volume, hence precisely characterizing the radius using only probabilistic arguments is delicate.

2.3.4. *Approximating the Radius of a Perturbed Matrix A .* Crucially here, if we notice that $\|Fy\|_1$ defines a norm on \mathbb{R}^{n-m} , we can apply the low- M^* bound in Theorem 2.8 to the normed space $(\mathbb{R}^{n-m}, \|Fy\|_1)$ with unit ball

$$K = \{y \in \mathbb{R}^{n-m} : \|Fy\|_1 \leq 1\}$$

instead of the space $(\mathbb{R}^n, \|x\|_1)$. Applying Theorem 2.8 requires computing $M^*(K)$ and, since an affine section of an affine section is itself an affine section, this then produces bounds on the radius

$$\text{radius}(\{x \in \mathbb{R}^n : Ax = 0, Gx = 0, \|x\|_1 \leq 1\})$$

where $G \in \mathbb{R}^{q \times n}$ is a i.i.d Gaussian matrix, with high probability. Thus $M^*(K)$ estimates the recovery threshold of the perturbed sampling matrix (A, G) and we use this last quantity as a proxy for the actual recovery threshold of A .

2.3.5. *Estimating $M^*(K)$.* Computing the dual norm of $\|Fy\|_1$ for any orthogonal matrix F is a convex problem, hence we can simply approximate M^* by simulation. In the particular case of $(\mathbb{R}^{n-m}, \|Fy\|_1)$, this means computing

$$M^* \triangleq \mathbf{E} \left[\max_{\|Fy\|_1 \leq 1} y^T g \right] = \mathbf{E} \left[\min_{F^T x = g} \|x\|_\infty \right] = \mathbf{E} \left[\min_{F^T x = 0} \|Fg + x\|_\infty \right] \quad (15)$$

by duality, where $g \sim \mathcal{N}(0, \mathbf{I}_{n-m})$ (assuming $F^T F = \mathbf{I}_{n-m}$). Sampling both terms simply means solving one linear program per sample. Also, a simple Cauchy inequality shows that $M(K^*)$ is bounded above by $O(1/\sqrt{k})$. Since the target precision for our estimate of $M(K^*)$ is always larger than $1/\sqrt{n}$, this produces a recipe for a randomized polynomial time algorithm for estimating S . In fact, following [Bourgain et al., 1988, Giannopoulos and Milman, 1997, Giannopoulos et al., 2005], we have the following bound.

Proposition 2.9. *If $K \subset \mathbb{R}^n$ is a symmetric convex body, $0 < \delta, \beta < 1$ and we pick N points x_i uniformly at random on the sphere \mathbb{S}^{n-1} with*

$$N = \frac{c \log(2/\beta)}{\delta^2} + 1$$

where c is an absolute constant, then

$$\left| M(K^*) - \frac{1}{N} \sum_{i=1}^N \|x_i\|_{K^*} \right| \leq \delta M(K^*)$$

with probability $1 - \beta$.

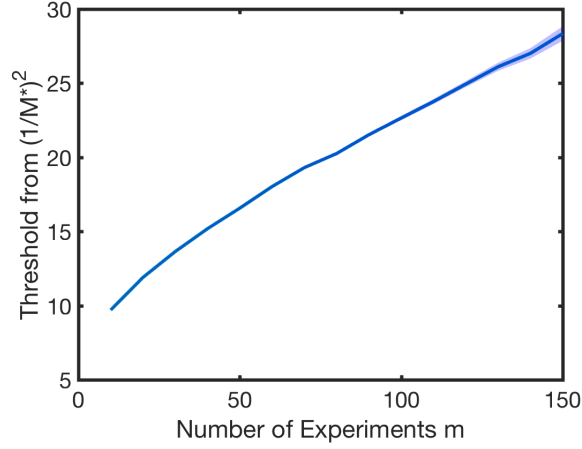


FIGURE 1. Recovery threshold k versus number of samples m , by estimating M^* in dimension 200. Shaded region at plus and minus one standard deviation.

3. ALGORITHMS FOR M^* -REGULARIZED DICTIONARY LEARNING

Let $A \in \mathbb{R}^{m \times n}$ be a dictionary matrix with linearly independent lines and F be a basis of $\text{Nullspace}(A)$. Then $M^*(A)$ was defined in (15) as

$$M^*(A) = \mathbf{E} \left[\min_{F^T x = g} \|x\|_\infty \right] \quad (16)$$

with $g \sim \mathcal{N}(0, \mathbf{I}_{n-m})$. We have seen above that this quantity is tractable and we will use it as a penalty in various dictionary learning problems. In general, given a dictionary learning task which involves minimizing a loss $l : A \in \mathbb{R}^{m \times n} \rightarrow l(A) \in \mathbb{R}$ with respect to a dictionary matrix A , on an admissible set \mathcal{C} , we solve the penalized loss minimization problem

$$\begin{aligned} & \text{minimize} && l(A) + \lambda M^*(A) \\ & \text{subject to} && A \in \mathcal{C} \end{aligned} \quad (17)$$

in the variable $A \in \mathbb{R}^{m \times n}$.

3.1. Optimizing M^* . In order to solve the regularized loss minimization problem (17) using e.g. stochastic gradient descent, we first need to compute a subgradient of M^* with respect to A . In practice, M^* is an explicit function of $F = \text{null}(A)$, so we will start by computing a subgradient with respect to F . We will then add a coupling constraint to link F and A . We have the following result.

Lemma 3.1. *The regularized function*

$$\begin{aligned} \nu_g(F) \triangleq & \min. && \|x\|_\infty + \frac{\lambda}{2} \|x\|_2^2 + \frac{\sigma}{2} \|r\|_2^2 \\ & \text{s.t.} && F^T x + r = g \end{aligned} \quad (18)$$

in the variables $x \in \mathbb{R}^n$, $r \in \mathbb{R}^{n-m}$, with $\lambda, \sigma > 0$ is differentiable and its gradient with respect to F is given by

$$\nabla \nu_g(F) = -x_g^*(F) y_g^{*T}(F), \quad (19)$$

where $x_g^*(F)$ and $y_g^*(F)$ are the primal and dual solutions of problem (18).

Proof. We consider here the regularized version of the linear program appearing in the formula of M^* in (15) in order to enforce uniqueness of the solution, which is does not hold for (15) when the intersection between the nullspace of F^T and the hypercube have some common directions. The differentiability result then follows from [Bonnans and Shapiro, 2013, Th. 4.24]. ■

To account for the fact that $M^*(A)$ is only explicitly computed from F , (17) can be rewritten as

$$\begin{aligned} \min. \quad & l(A) + \lambda M^*(F) \\ \text{s.t.} \quad & A \in \mathcal{C}, AF = 0 \\ & F^T F = I \end{aligned} \quad (20)$$

Imposing $AF = 0$ may yield numerical issues and we can replace the hard constraint by a penalty on $\|AF\|_F^2$, where $\|\cdot\|_F$ is the Frobenius norm. The problem then becomes

$$\begin{aligned} \text{minimize} \quad & l(A) + \lambda M^*(F) + \mu \|AF\|_F^2 \\ \text{subject to} \quad & A \in \mathcal{C} \\ & F^T F = I \end{aligned} \quad (21)$$

in the variables $A \in \mathbb{R}^{m \times n}$ and $F \in \mathbb{R}^{n \times n-m}$.

Assuming $l(A)$ is easy to minimize, a classical technique to solve the above problem involve alternate minimization in A and F . The variable F lies in the Stiefel manifold, hence minimizing in F can be performed using stochastic gradient descent on the Stiefel manifold

$$\mathcal{M} = \{F \in \mathbb{R}^{n \times n-m} : F^T F = I_{n-m}\}. \quad (22)$$

using the partial derivative obtained in Lemma 3.1.

Updates consist in projecting the partial gradient on $T_F \mathcal{M}$, the tangent space of \mathcal{M} at the current point F , make a gradient step in this tangent space and finally re-project the result on the manifold to get the new point. The resulting stochastic gradient descent algorithm is described in Algorithm 1.

Algorithm 1 Stochastic Gradient Descent on \mathcal{M} , $\mathbf{SGD}_{M^*}(null(A), \tau, n_{sgd})$

Input: Initial $F_0 \in \mathbb{R}^{n \times n-m}$, penalty μ , stepsize τ , number of gradient steps n_{sgd} .

$F := F_0$.

for 1 to n_{sgd} **do**

 Sample $g \sim \mathcal{N}(0, \mathbf{I}_{n-m})$,

 Update $F := \text{Proj}_{\mathcal{M}}(F - \tau \text{Proj}_{T_F \mathcal{M}}(\nabla \nu_g(F) + 2\mu A^T AF))$,

end for

Output: Matrix F .

In practice, the penalization coefficient μ is set close to 0 in the SGD steps. To obtain a F that is not too far from the nullspace of A , the initial F_0 belongs to the nullspace of the current iterate A . All the algorithms described in the next parts will then follow the generic alternating minimization structure of Algorithm 2.

Algorithm 2 Generic Alternate Minimization Algorithm

Input: Initial A , number of iterations $niter$, penalty μ , stepsize τ , number of gradient steps n_{sgd} .

for 1 to $niter$ **do**

 Update $A := \text{argmin}_{A \in \mathcal{C}} l(A) + \mu \|AF\|_F^2$,

 Update $F := \mathbf{SGD}_{M^*}(null(A), \tau, n_{sgd})$,

end for

Output: Dictionary matrix A .

We now apply this method to dictionary learning problems.

3.2. Compression by Dictionary Learning. Our objective here is to learn an over-complete dictionary that has good compressed sensing properties, i.e. a low M^* in our setting, to optimize dictionary performance out-of-sample.

3.2.1. *Dictionary Learning.* Let us recall the dictionary learning problem and its classical formulation for compression. Given a set of training observations $Y = (Y_1, \dots, Y_m) \in \mathbb{R}^{n \times m}$ and a sparsity target S , the goal is to find an over-complete dictionary $D = (D_1, \dots, D_p) \in \mathbb{R}^{n \times p}$ ($n < p \ll m$) and a representation $X = (X_1, \dots, X_m) \in \mathbb{R}^{p \times m}$ which minimize the training loss

$$\sum_i \|Y_i - DX_i\|_2^2 = \|Y - DX\|_F^2.$$

with $\|X_i\|_0 \leq S$. One classical regularization strategy is to normalize the columns of the dictionary, often called atoms. This prevents the problem to be ill-posed with infinitely many solutions differing only by the norms of their atoms. The problem then becomes

$$\begin{aligned} & \text{minimize} && \|Y - DX\|_F^2 \\ & \text{subject to} && \|D_i\|_2 = 1, \quad i = 1, \dots, p \\ & && \|X_j\|_0 \leq S, \quad j = 1, \dots, m \end{aligned} \quad (23)$$

in the variables $D \in \mathbb{R}^{n \times p}$, $X \in \mathbb{R}^{p \times m}$. This corresponds to choosing

$$\begin{aligned} l(A) := & \min. && \|Y - AX\|_F^2 \\ & \text{s.t.} && \|X_j\|_0 \leq S, \quad j = 1, \dots, m \end{aligned} \quad (24)$$

in our generic problem (21), where the minimization is performed with respect to $X \in \mathbb{R}^{p \times m}$ on the set of matrices with normalized columns $\mathcal{C} = \{A \in \mathbb{R}^{n \times p} \mid \|A_i\|_2 = 1, j = 1, \dots, p\}$.

A standard way to deal with problem (23) is the KSVD Algorithm (see Elad and Aharon [2006]). This is an alternate minimization algorithm between the dictionary D and the representation X . In the minimization with respect to X , Orthogonal Matching Pursuit (see e.g. Cai and Wang [2011]) is used to find an approximate solution with a given cardinality. For the minimization with respect to the dictionary D , the updates are made column by column. Given X , an update for the column j then consists in solving

$$\begin{aligned} & \text{minimize} && \|Y - \sum_{i \neq j} D_i X^{(i)} - D_j X^{(j)}\|_F^2 \\ & \text{subject to} && \|D_j\|_2 = 1 \end{aligned} \quad (25)$$

in the variable $D_j \in \mathbb{R}^n$. If one allows the minimization to include the variables $(D_j, X^{(j)})$, this comes down to finding the rank one matrix $D_j X^{(j)}$ that best approximates $M = Y - \sum_{i \neq j} D_i X^{(i)}$ in term of Frobenius norm. This can be obtain by performing a rank one SVD of M . One significant advantage of this method is that it directly gives a normalized update for D_j and also guarantees that the dictionary updates are descent steps. KSVD is detailed in Algorithm 3

Algorithm 3 KSVD

Input: Initial Dictionary D_0 , training patches $Y = [Y_1, \dots, Y_m]$, sparsity level S , number of iterations $niter$.

$D = D_0, K = K_0$.

for 1 to $niter$ **do**

for $j := 1$ to m **do**

$X_j := \text{OMP}(D, Y_j, S)$,

end for

for $l := 1$ to p **do**

$\Omega := \text{supp}(X^{(l)})$, where $X^{(l)}$ is the l -th line of X ,

$E := Y - \sum_{i \neq l} d_i X^{(i)}$,

$[U, S, V] := \text{SVD}(E_\Omega)$,

$D_l := U_1$,

$X^{(l)} := S_{11} V_1^T$,

end for

end for

Output: Dictionary $D = [d_1, \dots, d_p]$.

3.2.2. *Dictionary Learning with M^* Penalization.* The penalized formulation introduced in (21) can be used to learn a dictionary with low M^* . The penalized learning problem is then written

$$\begin{aligned}
& \text{minimize} && \|Y - DX\|_F^2 + \lambda M^*(F) + \mu \|DF\|_F^2 \\
& \text{subject to} && \|D_i\|_2 = 1, \quad i = 1, \dots, p \\
& && \|X_j\|_0 \leq S, \quad j = 1, \dots, m \\
& && F^T F = I_{p-n}
\end{aligned} \tag{26}$$

in the variables $X \in \mathbb{R}^{p \times m}$, $D \in \mathbb{R}^{n \times p}$, $F \in \mathbb{R}^{p \times p-n}$.

There is no change in the updates of the representation X when everything else is fixed compared to the classical setting. However for the dictionary updates in the variable D , the addition of the penalty term $\mu \|DF\|_F^2$ prevents the use of the SVD approach from Algorithm 3. Instead, the new dictionary is chosen to annihilate the gradient of the loss with respect to D and then projected on the admissible set \mathcal{C} . If \mathcal{C} is chosen to be the set of dictionaries with normalized columns as in KSVD, the projection changes the current value of M^* , since normalizing each column changes the nullspace of the matrix. To avoid this effect, we can take $\mathcal{C}_* = \{D \mid \max(\|D_i\|_2) = 1\}$. This set contains the previous one and the projection on it simply reduces to divide all the coefficients of the dictionary by $\max(\|D_i\|_2)$ which has no effect on the M^* . Overall, the M^* -regularized dictionary learning problem is then written

$$\begin{aligned}
& \text{minimize} && \|Y - DX\|_F^2 + \lambda M^*(F) + \mu \|DF\|_F^2 \\
& \text{subject to} && \max(\|D_i\|_2)_{i \in [1:p]} = 1 \\
& && \|X_j\|_0 \leq S, \quad j = 1, \dots, m \\
& && F^T F = I_{p-n}
\end{aligned} \tag{27}$$

in the variables $X \in \mathbb{R}^{p \times m}$, $D \in \mathbb{R}^{n \times p}$, $F \in \mathbb{R}^{p \times p-n}$.

Finally, the update with respect to F is done as in part 3.1, using a stochastic gradient descent to minimize M^* on the Stiefel Manifold. The complete M^* -penalized dictionary learning algorithm is then detailed as Algorithm 4.

Algorithm 4 Penalized Dictionary Learning

Input: Initial Dictionary D_0 , Initial nullspace F_0 , training patches $Y = [Y_1, \dots, Y_m]$, sparsity level S , number of iterations n_{iter} , regularization parameter μ , stepsize τ , number of gradient iterations n_{sgd}
 $D := D_0$, $K := K_0$.
for 1 to n_{iter} **do**
 for $j := 1$ to m **do**
 $X_j := \text{OMP}(D, Y_j, S)$,
 end for
 $D := \text{proj}_{\mathcal{C}_*}(Y X^T (X X^T + \mu F F^T)^{-1})$,
 $F := \text{SGD}_{M^*}(\text{null}(D), \tau, n_{sgd})$,
end for
Output: Dictionary D .

3.3. **Inpainting by Dictionary Learning.** Inpainting is a particular class of denoising problems for imaging. This is a situation where the noise is multiplicative and takes its values in $\{0, 1\}$. For an image I of size $n \times m$ the noise matrix is called a mask denoted $B \in \{0, 1\}^{p \times m}$. What is observed is a noisy version of the image $I \odot B$ (where \odot is the Hadamard product of matrices) which is basically I with missing parts appearing as black holes. Dictionary learning by patches has been adapted to the inpainting problem giving good results (see e.g. Mairal et al. [2008]). In this section, we adapt the M^* penalized algorithm to the inpainting setting.

3.3.1. *Inpainting Problems.* Given some training patches $Y = [Y_1, \dots, Y_m] \in \mathbb{R}^{n \times m}$ and a mask $B = [B_1, \dots, B_m] \in \mathbb{R}^{n \times m}$, the idea of inpainting by patches is essentially the same as the classical dictionary learning principle. It seeks to find a sparse representation of the training patches using a few learned atoms. However, only $B \odot Y$ is accessible, meaning that information is only available on some pixels of each patch. Due to the intrinsic sparse structure of natural images it is reasonable to think that there is enough information in the visible pixels to learn a good dictionary to fill the masked parts of the image. The learning task in this case is then simply

$$\begin{aligned} & \text{minimize} && \|B \odot (Y - DX)\|_F^2 \\ & \text{subject to} && \|D_i\|_2 = 1, \quad i = 1, \dots, p \\ & && \|X_j\|_0 \leq S, \quad j = 1, \dots, m \end{aligned} \quad (28)$$

in the variables $X \in \mathbb{R}^{p \times m}$, $D \in \mathbb{R}^{n \times p}$.

Due to the Hadamard product with B , the KSVD algorithm cannot be directly applied to solve this problem. Mairal et al. [2008] presented a weighted KSVD algorithm that will be referred to as wKSVD in the following. It uses an iterative algorithm detailed in Srebro and Jaakkola [2003] to approximate a solution of the weighted rank one approximation problem encountered when trying to update the dictionary column by column as in KSVD. This consists in solving the following

$$\begin{aligned} & \text{minimize} && \|W \odot (M - A)\|_F^2 \\ & \text{subject to} && \text{rank}(A) = 1 \end{aligned} \quad (29)$$

with respect to the matrix $A \in \mathbb{R}^{n \times m}$, with $M \in \mathbb{R}^{n \times m}$, $W \in \mathbb{R}_+^{n \times m}$. Pseudo code for wKSVD is detailed as Algorithm 5.

Algorithm 5 Weighted KSVD Algorithm

Input: Initial Dictionary D_0 , training patches $Y = [Y_1, \dots, Y_m]$, sparsity level S , number of iterations $niter$, number of intermediate iterations n_{dico} .

$D = D_0$, $K = K_0$.

for 1 to $niter$ **do**

for $j := 1$ to m **do**

$X_j \leftarrow \text{OMP}(\text{diag}(B_j)D, B_j \odot Y_j, S)$,

end for

for $l := 1$ to p **do**

$\Omega := \text{supp}(X^{(l)})$,

$E = Y - \sum_{i \neq l} d_i X^{(i)}$,

for 1 to n_{dico} **do**

$E_B := B \odot E + (\mathbf{1} - B) \odot d_l X^{(l)}$,

$[U, S, V] := \text{SVD}(E_B, \Omega)$,

$d_l := U_1$,

$X^{(l)} := S_{11} V_1^T$,

end for

end for

end for

Output: Dictionary $D = [d_1, \dots, d_p]$.

3.3.2. *M^* Penalization for Inpainting.* As above, an M^* penalty can be added to the classical loss, with the admissible set becoming $\mathcal{C}_* = \{D \mid \max(\|D_i\|_2) = 1\}$ and the penalized algorithm is modified using an iterative method during the dictionary update step in D . Indeed this update corresponds to solving the problem

$$\begin{aligned} & \text{minimize} && \|B \odot (Y - DX)\|_F^2 + \mu \|DF\|_F \\ & \text{subject to} && \max(\|D_i\|_2)_{i \in [1:p]} = 1 \end{aligned} \quad (30)$$

in the variable $D \in \mathbb{R}^{n \times p}$.

One can set $Y_B = B \odot Y + (\mathbf{1} - B) \odot DX$ and rewrite the loss above as $\|Y_B - DX\|_F + \mu \|DF\|_F$. The variable Y_B takes the values of the training patches matrix Y on the observed pixels and the current values of DX on the masked ones. Minimizing $\|Y_B - DX\|_F + \mu \|DF\|_F$ with respect to D , with Y_B fixed, can be solved as in the classical compression case.

This procedure can be seen as a missing values estimation problem, where given a matrix of observations Y with some missing values (the values of the masked pixels), one tries to find a dictionary D that minimize the previous error. Hence setting Y_B as detailed above consists in an estimation step where the missing values are replaced by the current estimate DX . Then one performs a minimization step on D to update the current estimate. This is done in an iterative setting and pseudo code for M^* penalized inpainting is described in Algorithm 6.

Algorithm 6 Penalized Dictionary Learning for Inpainting

Input: Initial Dictionary D_0 , initial nullspace F_0 , training patches $Y = [Y_1, \dots, Y_m]$, mask for on the training patches $B = [B_1, \dots, B_m]$, sparsity level S , number of iterations n_{iter} , regularization parameter μ , stepsize τ , number of gradient iterations n_{sgd} , number of intermediate iterations n_{dico} .

$D = D_0, K = K_0$.

for 1 to n_{iter} **do**

for $j := 1$ to m **do**

$X_j \leftarrow \text{OMP}(\text{diag}(B_j)D, B_j \odot Y_j, S)$,

end for

for 1 to n_{dico} **do**

$Y_B \leftarrow B \odot Y + (\mathbf{1} - B) \odot DX$,

$D \leftarrow \text{proj}_{C_*}(Y_B X^T (X X^T + \mu F F^T)^{-1})$,

end for

$F \leftarrow \text{SGD}_{M^*}(\text{null}(D), \tau, n_{sgd})$,

end for

Output: Dictionary D .

4. NUMERICAL RESULTS

This section is dedicated to experimental results obtained using the previously described framework. The optimization toolbox Manopt (Boumal et al. [2014]) was used to perform stochastic gradient descent on the Stiefel manifold, together with the SPAMS toolbox (Mairal et al. [2009]) to perform the OMP algorithm. All the tests were done on grayscale images of size 512×512 . The size of the patches has been set to 8×8 , meaning $n = 64$. When not specified, the columns of the initial dictionaries D_0 are normalized independent mean zero and unit variance Gaussian vectors except for the the last one being the constant vector $\frac{1}{\sqrt{n}}$. This last column remains unmodified by the various algorithms to capture the mean information.

4.1. Compression Experiments. This is the setting of Section 3.2.1. The number of atoms in the dictionaries has been set to $p = 4n = 256$. The training set is formed by $200p = 51200$ patches selected randomly in four training images. Both KSVD and penalized dictionary methods are applied for 150 iterations, for different sparsity levels S between 2 and 10.

Figure 2 shows examples of dictionaries obtained using these two algorithms. The dictionary obtained by M^* penalization is not as sharp as the dictionary learned with KSVD, yet has a lot more structure than random Gaussian dictionaries. With Gaussian initialization, the M^* of the dictionaries learned by both methods starts at minimal value. During the iterations of the algorithms dictionaries acquire more and more structure and their M^* increases gradually as shown in Figure 3 on the left.

To measure the quality of a dictionary D in the compression setting, we formed a test set of 21 standard gray scale 512×512 images. Each image is decomposed in non overlapping patches. This means for instance

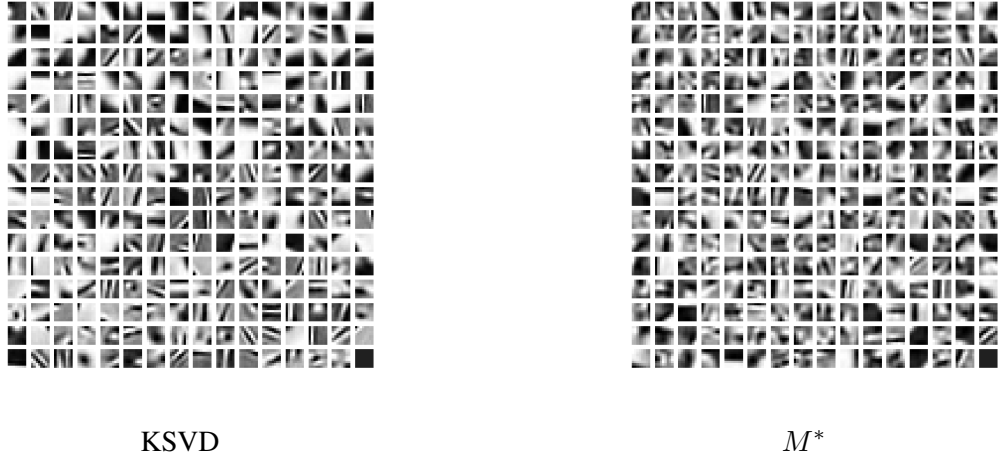


FIGURE 2. Example of dictionaries learned by both KSVD and M^* -Regularization, with a training sparsity $S = 5$ and a regularization parameter $\mu = 10^8$. (Gaussian M^* for this dimensions is 1.517 ± 0.003). Left: KSVD, $M^* = 1.686 \pm 0.004$. Right: M^* penalization, $M^* = 1.558 \pm 0.003$.

that a 512×512 image is cut into $64 \cdot 64 = 4096$ adjacent patches of size 8×8 . Each image is then represented as a set of patches $Y \in \mathbb{R}^{64 \times 4096}$ and is approximated by DX where X is obtained as in (24) with a given reconstruction sparsity k (which is not necessarily the same as the training sparsity S). This corresponds to the compression factor: the smaller k , the more compressed the images are, so compression is measured by the cardinality of the representations of the images in the dictionary.

For a given set of non overlapping patches $Y \in \mathbb{R}^{64 \times 4096}$ representing an image and a cardinality k , we define

$$Y_k(D) \triangleq DX, \quad \text{where } X = \underset{X}{\operatorname{argmin}} \quad \begin{aligned} & \|Y - DX\|_F^2 \\ & \text{s.t.} \quad \|X_j\|_0 \leq k, \quad j = 1, \dots, 4096 \end{aligned} \quad (31)$$

where the minimization is performed with respect to $X \in \mathbb{R}^{256 \times 4096}$. Here, $Y_k(D)$ corresponds to the matrix where each column is an approximation of the corresponding column of Y using a linear combination of k atoms of D .

We write D_S the dictionary obtained by KSVD with a training sparsity S , and D_S^μ the one obtained by the M^* penalized algorithm with sparsity S and regularization parameter μ in problem (27). For a patch representation Y of a test image, a reconstruction sparsity k and a penalization coefficient μ , approximation quality for the KSVD (resp. M^* penalized) algorithm is obtained by computing both PSNR and SSIM between the ground truth Y , and $Y_k(D_S)$ (resp. $Y_k(D_S^\mu)$). SSIM is a measurement of structural similarity designed to describe the perceived quality of an image more faithfully than PSNR, which is a pixel to pixel measurement (Wang et al. [2004]). In order to plot the aggregate curve in Figure 3 on the right, we took the average of the PSNR and SSIM values over all 21 images in the test set, for a range of reconstruction sparsity k between 2 and 30.

When using small penalization μ in problem (27), the two methods had similar compression performance on the test set, with a minor advantage for M^* penalized dictionary for small values of k . In this case, the M^* of the penalized dictionary has an intermediate value between that of the dictionary from KSVD and that of a Gaussian dictionary. Increasing the penalization parameter allows to learn dictionaries with M^* almost as low as Gaussian ones, however these new dictionaries with low M^* don't fit the training data as well and the test PSNR and SSIM become worse than those of KSVD.

Convergence of the algorithms with deterministic initialization has also been experimented, with D_0 a constant matrix with columns of norm 1. The KSVD algorithm converges very slowly in this case (if at all). All the columns of the dictionary remain very close to the initial ones. The penalized algorithm on the other

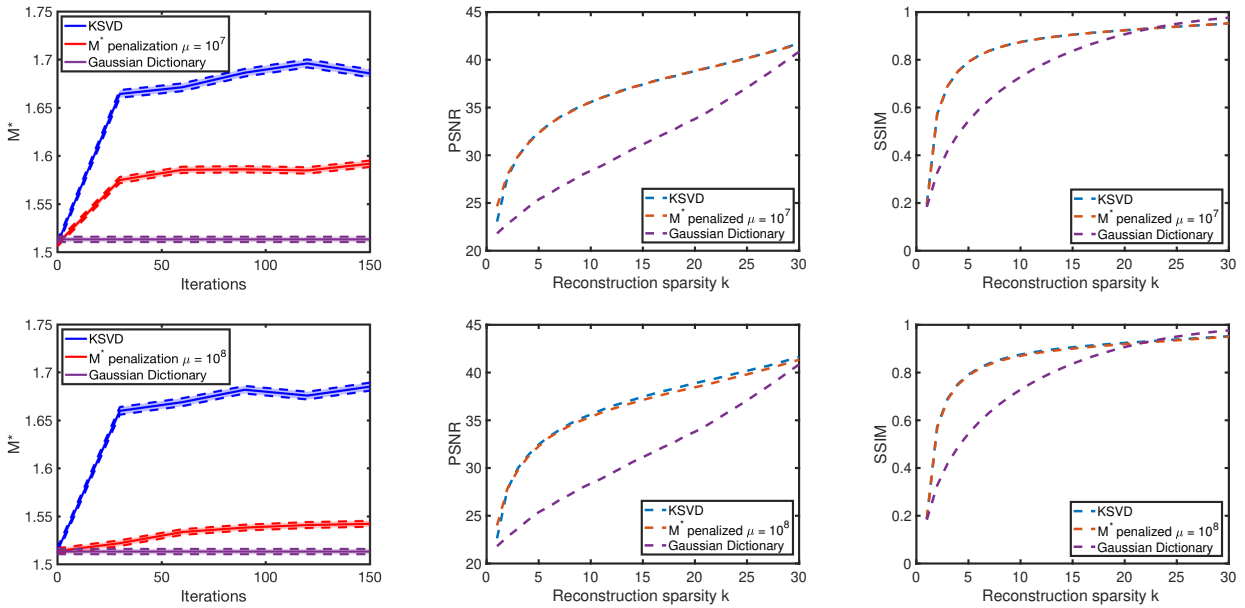


FIGURE 3. Compression experiments with training sparsity $S = 5$. Top: $\mu = 10^7$. Bottom: $\mu = 10^8$. Left: Evolution of the M^* across algorithm iterations. Middle: PSNR of the compressed test images. Right: SSIM of the compressed test images.

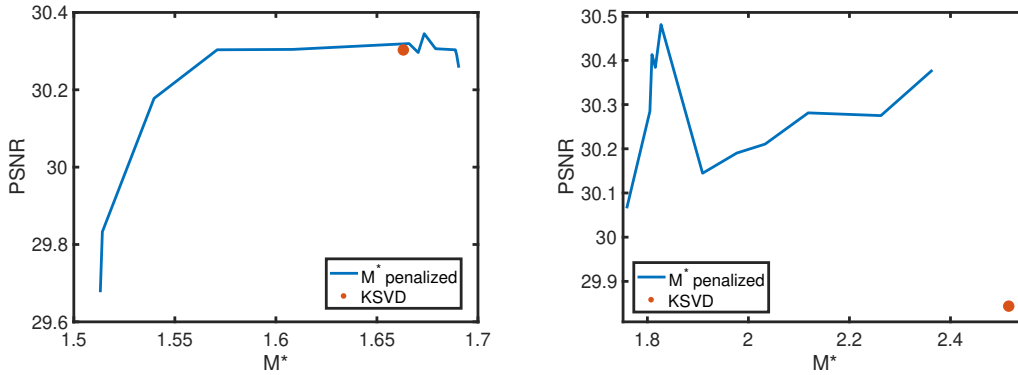


FIGURE 4. Training PSNR versus M^* when varying the regularization parameter μ with training sparsity $S = 4$. Red dot corresponds to the KSVD algorithm. Left: Gaussian initialization. Right: deterministic initialization.

hand achieves similar performances on train and test errors compared to the random initialization setting. It also appears that a stronger M^* regularization is suitable to reach better training error, compared with random initialization (see Figure 4).

For the compression task, it thus appears that dictionaries learned by KSVD are nearly optimal, if randomly initialized, in the sense that learning a dictionary with lower M^* through M^* penalization method does not improve performance. Regularization does improve convergence when starting from a deterministic matrix. The next section is devoted to inpainting experiments where the M^* will play a much more significant role due to the particular structure of the noise introduced by the masks.

4.2. Inpainting Experiments. In the setting of §3.3, the number of atoms in the dictionaries is set to $p = 128$ and the number of training patches is $m = 150p$. The size of the patches used is still 8×8 . This corresponds to dictionaries of size 64×128 . Experiments are based on a set of 14 gray scale 512×512

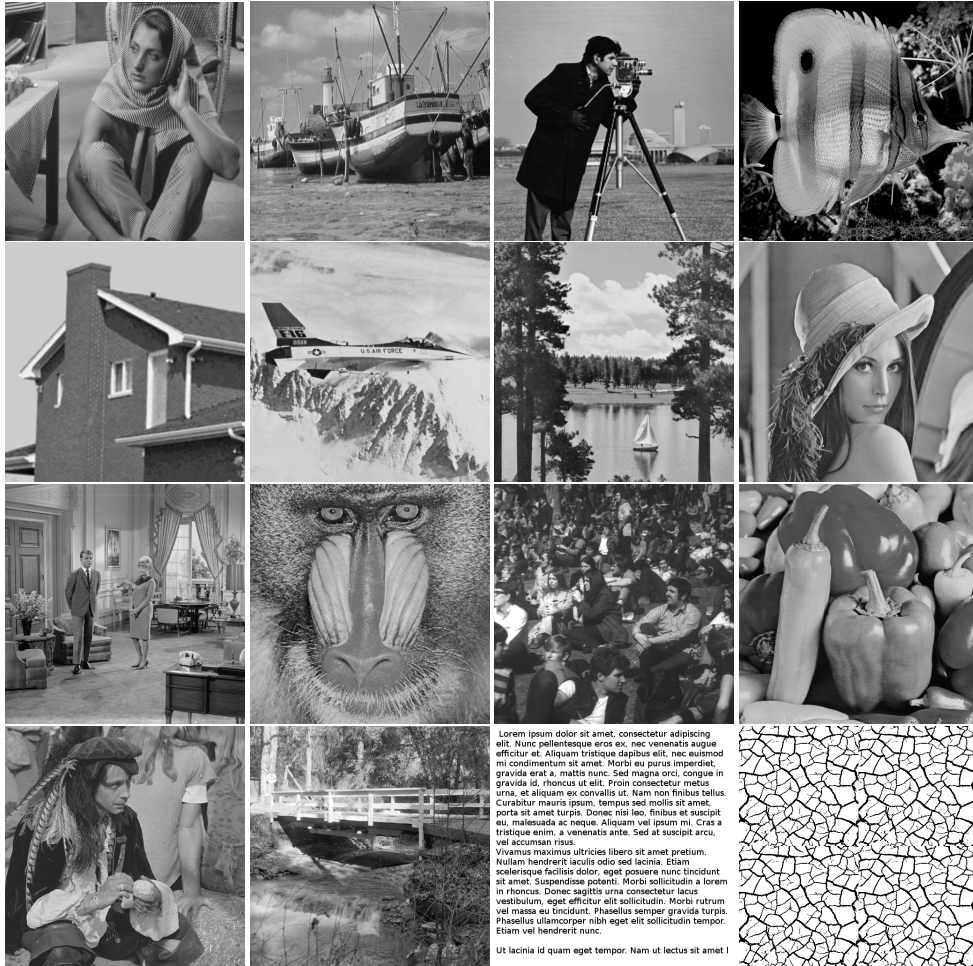


FIGURE 5. Images and masks used for inpainting experiments. From top left to bottom right line by line: "barbara", "boat", "cameraman", "fish", "house", "jetplane", "lake", "lena", "livingroom", "mandril", "people", "peppers", "pirate", "walkbridge", "text", "cracks".

images and two masks, one representing cracks and one with text (see Figure 5). For a given mask B , a given image I and a training sparsity S , m masked patches are selected randomly in the image $B \odot I$. Then both wKSVD and M^* penalized algorithms are run for 50 iterations on these training patches with a training sparsity S to obtain the dictionaries $D_S(I, B)$ and $D_S^\mu(I, B)$. For simplicity, in all these inpainting experiments, the regularization parameter μ has been fixed to 10^8 , the same order of magnitude as the training loss $\|Y - DX\|_F^2$ for our experiments. Of course, results would further improve with μ chosen adaptively.

To reconstruct a new masked image $B' \odot I' \in \mathbb{R}^{512 \times 512}$ thanks to a dictionary D , all the $(512 - 8) \cdot (512 - 8) = 254016$ patches of I' are gathered in a matrix $Y = \mathbb{R}^{64 \times 254016}$. Compared with the compression setting, all the 8×8 patches are used for the reconstruction. For simplicity, $B' \odot Y$ will represent the matrix of masked patches, even if B' has not the right dimension, and corresponds to the set of all patches of the mask. As in (31), we set

$$Y_k(D) \triangleq DX \quad \text{where } X = \underset{\text{s.t.}}{\text{argmin.}} \quad \|B' \odot (Y - DX)\|_F^2 \quad (32)$$

$$\|X_j\|_0 \leq k, \quad j = 1, \dots, 25016$$

$Y_k(D)$ is the approximation of the patches Y through D with a reconstruction sparsity of k when only $B' \odot Y$ is observed. An approximation $I'_{B'}$ of I' is then reconstructed from patches $Y_k(D)$ by recasting them to an

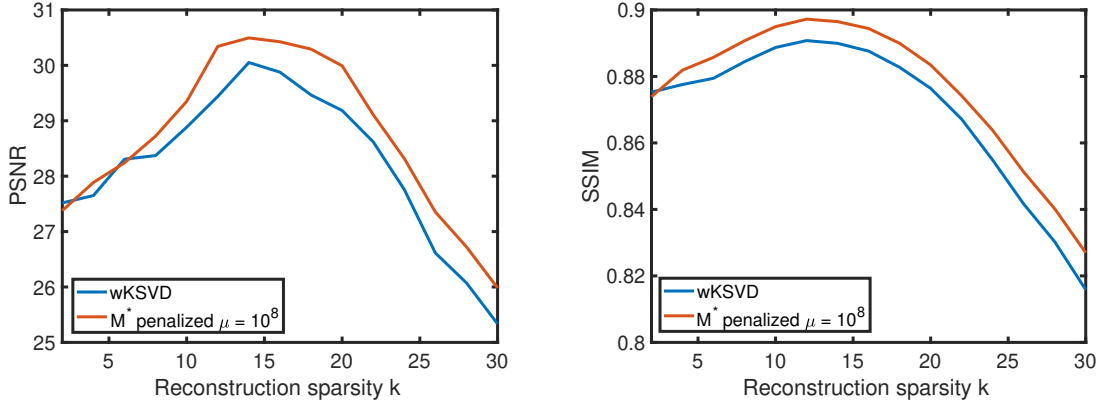


FIGURE 6. Reconstruction errors for the inpainting of 'lake' based on dictionaries learned on 'livingroom' with both methods (with $S = 5$).

image and simply averaging their overlaying parts. The final reconstructed image is $B' \odot I' + (1 - B') \odot I'_{B'}$, since I' was already known on the pixels p where $B'(p) = 1$ and the approximation $I'_{B'}$ is only used for the pixels where $B'(p) = 0$.

The new image reconstructed can be compared to the original using PSNR and SSIM measures. For instance Figure 6 represents the curves of PSNR and SSIM versus reconstruction sparsity for the inpainting of the "lake" image trained on the image "livingroom" and masked by the cracks mask with $S = 5$.

The 14 images are used successively as training image with both masks and for all the sparsity levels S between 4 and 10. This means $2 \times 14 \times 7$ dictionaries computed for each method. The dictionaries obtained by both methods are then used to reconstruct each of the 14 images with reconstruction sparsity k between 2 and 30. This means forming $(\text{number of mask} : 2) \times (\text{number of train images} : 14) \times (\text{number of train sparsity } S : 7) \times (\text{number of test images} : 14) = 2744$ PSNR and SSIM curves (as in Figure 6).

For each plot, the area of the gap between the curve obtained from M^* penalization and the curve from wKSVD gives an indicator of the benefit of regularized methods (the larger the better). This area can be computed as the mean of the difference between the curves. These areas are aggregated over the training sparsity S and over the two masks for each couple of train/test images and the results in terms of PSNR are shown in Figure 7 where the abscissa corresponds to the training images and the ordinate to the test ones. Figure 8 is the same type of figure but comparing the performance of our low- M^* dictionaries to Gaussian ones, to illustrate the impact of the learning step.

Finally the distribution of the SSIM and PSNR gaps between the images reconstructed by the two methods is represented on Figure 9 on the left. This distribution is shifted on the positive side showing globally better reconstruction performances with M^* penalized method. One can also observe the distribution of the M^* of the dictionaries produced by both methods in Figure 9 on the right. The x-axis corresponds to the difference of M^* between the dictionaries and Gaussian matrices. Most of the dictionaries obtained by the penalized method have a nearly optimal M^* .

Compared with the previous setting, the training masked patches should not be fitted exactly. Indeed it would mean that the dictionary learned the noise and the reconstructed image would artifact from the mask. This is a more interesting setting for M^* penalized methods because the dictionary learned must have better generalization performance in order to fill the void in the images. The regularization parameter μ has been set to a constant value for all the inpainting experiments. It could of course be fine tuned to improve reconstruction results. For example, results for inpainting have been presented for S between 4 and 10 with a regularization parameter $\mu = 10^8$. When using the same value of μ for smaller S as 2 and 3 performances can be worse than with wKSVD, however decreasing the regularization parameter to $\mu = 10^6$ allows to retrieve or improve the reconstruction performance of wKSVD.

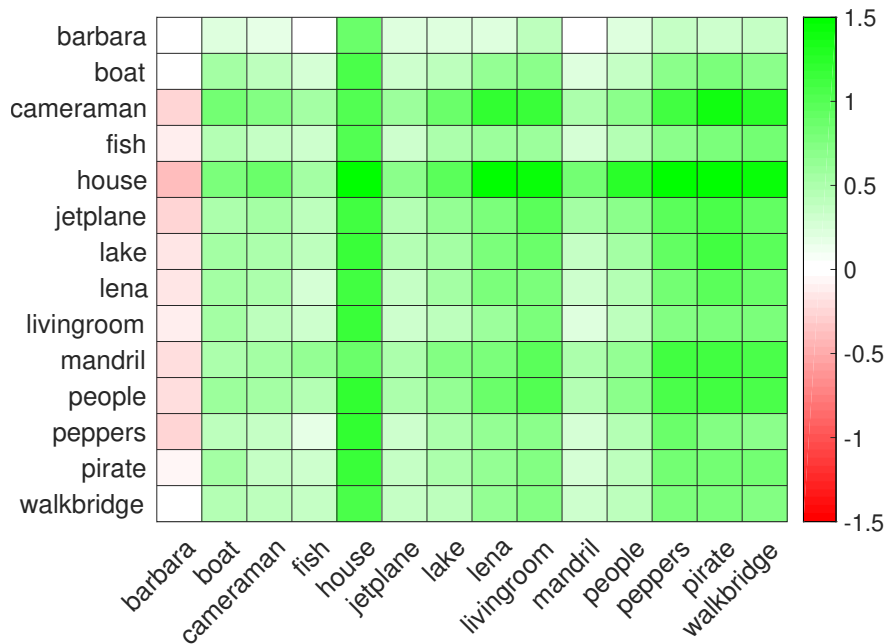


FIGURE 7. Heatmap of the average PSNR gap between the M^* penalized method and wKSVD with on x-axis the training images and on y-axis the test images. Except for the Barbara image which has very particular texture, M^* penalized algorithm globally performs better.

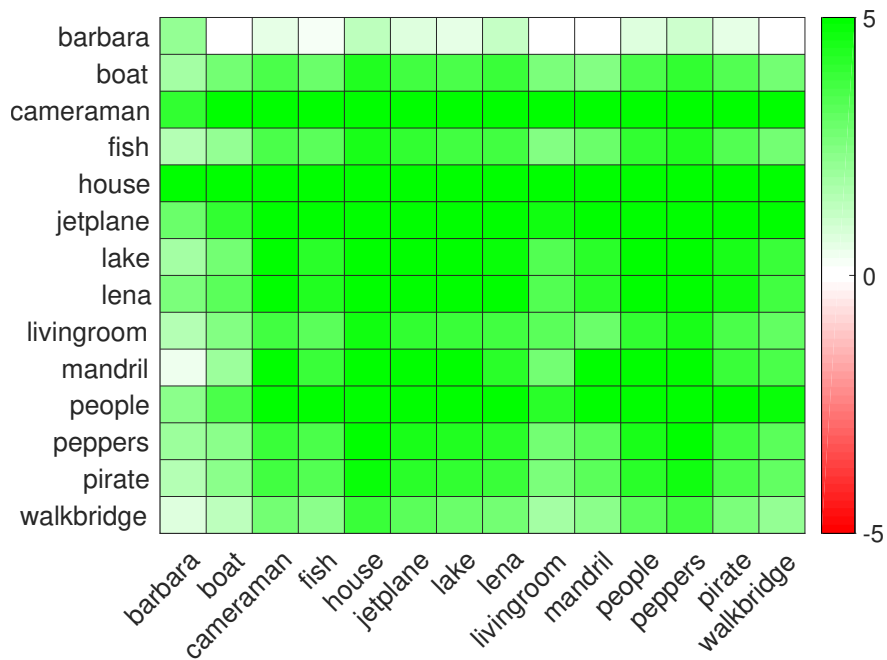


FIGURE 8. Heatmap of the average PSNR gap between the M^* penalized method and random Gaussian dictionaries with on x-axis the training images and on y-axis the test images. Gaussian dictionaries although they are M^* optimal, don't perform as well as low M^* dictionaries learned on training images.

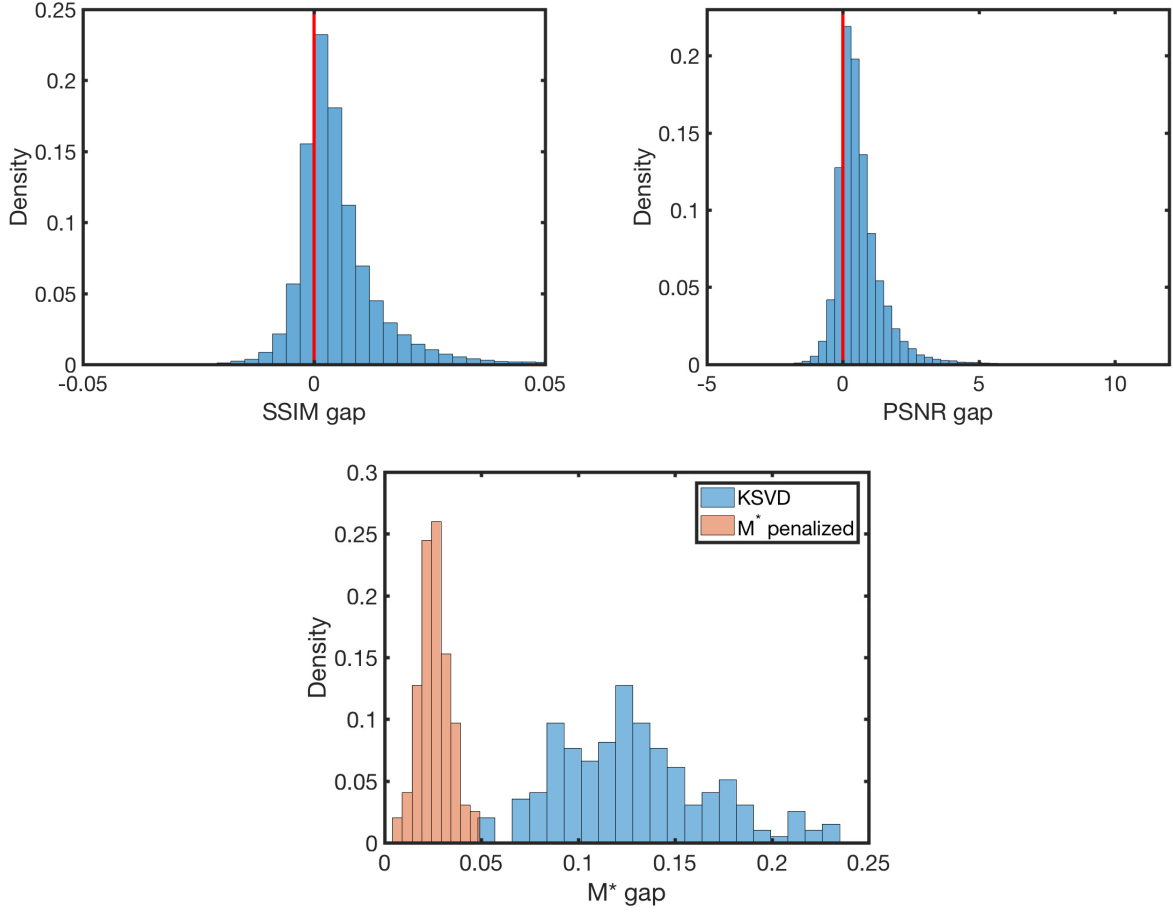


FIGURE 9. Results of the $2 \times 7 \times 14 \times 14$ inpainting experiments. Top: Histogram of the gaps (M^* minus KSVd) in SSIM and PSNR of reconstructed images between both methods. Bottom: Histogram of the gap of M^* regularized solution with Gaussian M^* .

ACKNOWLEDGEMENTS

AA is at CNRS & d epartement d'informatique,  Ecole normale sup erieure, UMR CNRS 8548, 45 rue d'Ulm 75005 Paris, France, INRIA and PSL Research University. The authors would like to acknowledge support from the *data science* joint research initiative with the *fonds AXA pour la recherche* and Kamet Ventures.

5. APPENDIX

We recall some results on a low- M estimate.

Theorem 5.1 (Low M estimate). *Let $\lambda \in (0, 1)$ and $k = \lfloor \lambda n \rfloor$ and $E \subset \mathbb{R}^n$ be a subspace of codimension k chosen uniformly at random w.r.t. to the Haar measure on $\mathcal{G}_{n, n-k}$, suppose $B_2^n \subset K$ and*

$$M(K) \geq \sqrt{\lambda}$$

then

$$\text{radius}(K \cap E) \leq \frac{c\sqrt{1-\lambda}}{M(K) - \sqrt{\lambda}}$$

with probability $1 - c_2 e^{-c_3 \delta^2 (1-\lambda)^n}$, where

$$\delta = \frac{M^2(K) - \lambda}{1 - M^2(K)}$$

and c_1, c_2, c_3 are absolute constants.

Proof. See [Giannopoulos et al., 2005, Th.B]. ■

Note that the condition $B_2^n \subset K$ means the set K needs to be normalized by $b(K)$. Klartag [2004] recently produced a similar result using $M(K)$ together with volume ratios. This result applies to all values of $M(K)/b(K)$, unfortunately, the dependence on k is exponential instead of being polynomial.

We will see below that the quantities $M(K)$ and $b(K)$ which characterize the phase transition for sections of the norm ball of $\|Fy\|_1$ can be approximated efficiently. We first recall a result which can be traced back at least to [Nesterov, 1998a, Nemirovski, 2005], approximating the mixed $\|\cdot\|_{2 \rightarrow 1}$ operator norm by a MAXCUT type relaxation.

Proposition 5.2. Let $F \in \mathbb{R}^{n \times n-m}$, then

$$\frac{2}{\pi} SDP(F) \leq \max_{\|x\|_2 \leq 1} \|Fx\|_1^2 \leq SDP(F) \quad (33)$$

where

$$\begin{aligned} SDP(F) = \max. & \quad \mathbf{Tr}(XFF^T) \\ \text{s.t.} & \quad \mathbf{diag}(X) = \mathbf{1} \\ & \quad X \succeq 0. \end{aligned} \quad (34)$$

Proof. We can write, by conjugacy,

$$\max_{\|x\|_2 \leq 1} \|Fx\|_1^2 = \max_{\|u\|_\infty \leq 1} \|u^T F\|_2^2 = \max_{\|u\|_\infty \leq 1} u^T F F^T u$$

and by convexity of $u^T F F^T u$ this is equal to

$$\max_{u \in \{-1, 1\}^n} u^T F F^T u$$

and Nesterov [1998b] (using again the fact that FF^T is positive semidefinite) shows that this problem can be approximated within a factor $2/\pi$ by the semidefinite relaxation in (34). ■

This means that the mixed norm $b(K)$, which is typically hard to bound in probabilistic arguments, is approximated within a factor $2/\pi$ by solving a MAXCUT semidefinite relaxation when the norm ball is a section of the ℓ_1 ball. We now recall a classical result showing that the spherical average $M(K)$ can be approximated by a Gaussian average.

Lemma 5.3. Let f be a homogeneous function on \mathbb{R}^n , then

$$\int_{\mathbb{S}^{n-1}} f(x) d\sigma(x) = \left(\frac{1}{\sqrt{n}} + \frac{1}{4n^{3/2}} + o(n^{-3/2}) \right) \mathbf{E}[f(g)]$$

where σ is the Haar measure on the sphere and $g \sim \mathcal{N}(0, \mathbf{I}_n)$.

Proof. Because the Gaussian measure γ is invariant by rotation, uniqueness of the Haar measure on \mathbb{S}^{n-1} means that

$$\int_{\mathbb{S}^{n-1}} f(x) d\sigma(x) = \lambda_n \int_{\mathbb{R}^n} \|x\|_2 f(x/\|x\|_2) d\gamma(x) = \lambda_n \int_{\mathbb{R}^n} f(x) d\gamma(x)$$

for some constant λ_n satisfying

$$\lambda_n = \int_{\mathbb{R}^n} \|x\|_2 d\gamma(x)$$

and we conclude using

$$\int_{\mathbb{R}^n} \|x\|_2 d\gamma(x) = \frac{\sqrt{2}\Gamma((n+1)/2)}{\Gamma(n/2)} = \sqrt{n} - \frac{1}{4\sqrt{n}} + o(n^{-1/2})$$

as n goes to infinity. ■

We can now easily compute $M(K)$, when K is the unit ball of $\|Fy\|_1$, with

$$M(K) = \left(\frac{1}{\sqrt{n}} + \frac{1}{4n^{3/2}} + o(n^{-3/2}) \right) \sqrt{\frac{2}{\pi}} \sum_{i=1}^n \|F_i\|_2 \quad (35)$$

where F_i are the rows of the matrix F , with $F \in \mathbb{R}^{n \times n-m}$ satisfying $AF = 0$. The key difficulty with these approximations of the Dvoretzky dimension is that $M(B_1^n)$ is roughly equal to $\sqrt{2n/\pi}$, so the ratio $M(K)/b(K)$ is already constant and the $2/\pi$ approximation ratio for $b(K)$ only produces trivial bounds. Hence, even though we can expect matrices with high approximate ratio $M(K)/SDP(F)$ to be good sensing matrices, there are no guarantees that all such matrices will have high approximate ratios.

REFERENCES

- Afonso S Bandeira, Edgar Dobriban, Dustin G Mixon, and William F Sawin. Certifying the restricted isometry property is hard. *IEEE transactions on information theory*, 59(6):3448–3450, 2013.
- Quentin Berthet and Philippe Rigollet. Computational lower bounds for sparse pca. *arXiv preprint arXiv:1304.0828*, 2013.
- J Frédéric Bonnans and Alexander Shapiro. *Perturbation analysis of optimization problems*. Springer Science & Business Media, 2013.
- Nicolas Boumal, Bamdev Mishra, P-A Absil, and Rodolphe Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459, 2014.
- J. Bourgain, J. Lindenstrauss, and V. Milman. Minkowski sums and symmetrizations. *Geometric aspects of functional analysis*, pages 44–66, 1988.
- Claire Boyer, Nicolas Chauffert, Philippe Ciuciu, Jonas Kahn, and Pierre Weiss. On the generation of sampling schemes for magnetic resonance imaging. *SIAM Journal on Imaging Sciences*, 9(4):2039–2072, 2016.
- Claire Boyer, Jérémie Bigot, and Pierre Weiss. Compressed sensing with structured sparsity and structured acquisition. *Applied and Computational Harmonic Analysis*, 2017.
- A. Brieden, P. Gritzmann, R. Kannan, V. Klee, L. Lovász, and M. Simonovits. Deterministic and randomized polynomial-time approximation of radii. *Mathematika*, 48(1-2):63–105, 2001.
- T Tony Cai and Lie Wang. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information theory*, 57(7):4680–4688, 2011.
- E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.
- Emmanuel J Candes and Yaniv Plan. A probabilistic and riplless theory of compressed sensing. *IEEE transactions on information theory*, 57(11):7235–7254, 2011.
- Nicolas Chauffert, Philippe Ciuciu, Jonas Kahn, and Pierre Weiss. Variable density sampling with continuous trajectories. *SIAM Journal on Imaging Sciences*, 7(4):1962–1992, 2014.
- A. Cohen, W. Dahmen, and R. DeVore. Compressed sensing and best k-term approximation. *Journal of the AMS*, 22(1):211–231, 2009.
- Alexandre d’Aspremont and Laurent El Ghaoui. Testing the nullspace property using semidefinite programming. *Mathematical Programming*, 127:123–144, 2011.
- Alexandre d’Aspremont, Francis Bach, and Laurent El Ghaoui. Approximation bounds for sparse principal component analysis. *Mathematical Programming*, 148(1-2):89–110, 2014.
- D. L. Donoho and J. Tanner. Sparse nonnegative solutions of underdetermined linear equations by linear programming. *Proc. of the National Academy of Sciences*, 102(27):9446–9451, 2005.
- M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

- R.M. Freund and J.B. Orlin. On the complexity of four polyhedral set containment problems. *Mathematical Programming*, 33(2):139–145, 1985.
- A. Giannopoulos, V.D. Milman, and A. Tsolomitis. Asymptotic formulas for the diameter of sections of symmetric convex bodies. *Journal of Functional Analysis*, 223(1):86–108, 2005.
- A. A. Giannopoulos and V. D. Milman. On the diameter of proportional sections of a symmetric convex body. *International Math. Research Notices, No. 1 (1997) 5–19.*, (1):5–19, 1997.
- P. Gritzmann and V. Klee. Computational complexity of inner and outer j -radii of polytopes in finite-dimensional normed spaces. *Mathematical programming*, 59(1):163–213, 1993.
- A. Juditsky and A.S. Nemirovski. On verifiable sufficient conditions for sparse signal recovery via ℓ_1 minimization. *Mathematical Programming Series B*, 127(57-88), 2011.
- B.S. Kashin and V.N. Temlyakov. A remark on compressed sensing. *Mathematical notes*, 82(5):748–755, 2007.
- B. Klartag and R. Vershynin. Small ball probability and Dvoretzky’s theorem. *Israel Journal of Mathematics*, 157(1): 193–207, 2007.
- Bo’az Klartag. A geometric inequality and a low M -estimate. *Proceedings of the American Mathematical Society*, 132(9):2619–2628, 2004.
- L. Lovasz and M. Simonovits. On the randomized complexity of volume and diameter. In *Foundations of Computer Science, 1992. Proceedings., 33rd Annual Symposium on*, pages 482–492. IEEE, 1992.
- Michael Lustig, David L Donoho, Juan M Santos, and John M Pauly. Compressed sensing mri. *IEEE signal processing magazine*, 25(2):72–82, 2008.
- J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696. ACM, 2009.
- Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69, 2008.
- Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.
- V.D. Milman and G. Schechtman. *Asymptotic theory of finite dimensional normed spaces*, volume 1200 of *Lecture notes in mathematics*. Springer Verlag, 1986.
- VD Milman and G. Schechtman. Global vs. local asymptotic theories of finite dimensional normed spaces. *Duke Math. J.*, 90:73–93, 1997.
- A.S. Nemirovski. *Computation of matrix norms with applications to Robust Optimization*. PhD thesis, Technion, 2005.
- Y. Nesterov. *Global quadratic optimization via conic relaxation*. Number 9860. CORE Discussion Paper, 1998a.
- Y. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization methods and software*, 9(1): 141–160, 1998b.
- A. Pajor and N. Tomczak-Jaegermann. Subspaces of small codimension of finite-dimensional banach spaces. *Proceedings of the American Mathematical Society*, 97(4):637–642, 1986.
- Daniel A Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)*, 51(3):385–463, 2004.
- Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 720–727, 2003.
- S. Szarek. Convexity, complexity, and high dimensions. In *International Congress of Mathematicians*, volume 2, pages 1599–1621, 2010.
- R. Vershynin. *Lectures in Geometric Functional Analysis*. In preparation, 2011. URL <http://www-personal.umich.edu/~romanv/papers/GFA-book/GFA-book.pdf>.
- Tengyao Wang, Quentin Berthet, and Yaniv Plan. Average-case hardness of rip certification. In *Advances in Neural Information Processing Systems*, pages 3819–3827, 2016.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Jonathan Weed. Approximately certifying the restricted isometry property is hard. *IEEE Transactions on Information Theory*, 2017.

INRIA & D.I., UMR 8548,
ÉCOLE NORMALE SUPÉRIEURE, PARIS, FRANCE.
E-mail address: mathieu.barre@inria.fr

CNRS & D.I., UMR 8548,
ÉCOLE NORMALE SUPÉRIEURE, PARIS, FRANCE.
E-mail address: aspremon@ens.fr