Robust Variants of Dictionary Learning Exploiting M-Estimators

Carlos A. Loza

***Abstract*—We propose a robust alternative the well known dictionary learning technique K-SVD. Specifically, we exploit the theory behind M-Estimators to incorporate robustness into the sparse coding stage of K-SVD, and hence, decrease the estimation bias that might be introduced when outliers are present. Five different M-Estimators are introduced alongside their optimal hyperparameters in order to avoid parameter tuning by the user. In this way, the proposed framework has the same number of free parameters as K-SVD with the added feature of robustness and improved performance in non–Gaussian environments. We thoroughly demonstrate the superiority of the proposed algorithms via recovery of generating dictionaries for synthetic data and image denoising under two types of non–homogenous noise—salt and pepper noise, and impulsive noise.**

***keywords*——Dictionary Learning, Image Denoising, K-SVD, M-Estimators, Robust Estimation**

I. Introduction

Dictionary learning and sparse coding are two of the main building blocks in sparse modeling. They both harness the principles of parsimony regarding the representation of a given phenomenon with as few variables as possible. In linear terms, sparse coding decomposes the inputs as the sum of weighted contributions from a given basis (usually overcomplete to encourage sparseness), also known as dictionary. Examples of such off–the–shelf dictionaries are the Fourier complex sinusoids, wavelets, and collection of Gabor patterns, either localized in space, time, or frequency. Yet, the real advantage of dictionary learning came with the work of Olshausen and Field [1] in neuroscience—they provided a bona fide proof of concept of a fully data–driven scheme to estimate said bases in an unsupervised fashion.

Sparse modeling has found a wide range of applications in the fields of Image Processing and Computer Vision [2], e.g. denoising [3-4], inpainting [5], and demosaicking [6] to name a few. Most of these applications rely on the well known dictionary learning technique known as K-SVD [7]. Essentially, the algorithm exploits block coordinate descent (BCD) to reach a local stationary point of a constrained linear problem. Results are theoretically optimal under Gaussian noise assumptions. However, if the underlying degradation deviates from normality, e.g. noise drawn from long tail distributions, missing pixels, salt and pepper noise, or impulsive noise, the estimators might introduce a bias in the dictionary elements.

Carlos A. Loza is with the Department of Mathematics. Universidad San Francisco de Quito, Quito, Ecuador (e-mail: cloza@usfq.edu.ec). This work was supported by Universidad San Francisco de Quito (Poligrant No.12311).

Robust M-Estimators are a principled alternative to deal with outliers in linear regimes [8-9]. They exploit cost functions that go beyond the widely used Minimum Squared Error (MSE) criteria, which is optimal only for Gaussian scenarios. In this way, robust estimators adaptively assign lower importance (by means of weighing) to samples deemed as outliers. This option was explored in the sparse coding setting under the term RobOMP [10], where M-Estimators were incorporated into one of the most widely used sparse decomposition algorithms—Orthogonal Matching Pursuit [11]. Now, we propose a fully robust sparse modeling framework where RobOMP is one of the underpinnings of K-SVD in order to replace its inherent MSE criterion with robust measures from the theory of M-Estimators. The result is M-Estimators–based K-SVD, or MeK-SVD for short.

The five variants of MeK-SVD are thoroughly tested and compared to benchmark state of the art algorithms. Recovery of ground truth generating dictionaries with synthetic data and image denoising under two types of non–homogenous noise (salt and pepper, and impulsive noise) confirm the superiority of the proposed techniques over K-SVD. The rest of the paper is organized as follows: Section 2 details the problem of dictionary learning alongside the state of the art. Section 3 introduces M-Estimators and the robust variants of K-SVD. Section 4 summarizes the results, and lastly, Section 5 concludes the paper and outlines potential further work.

II. Dictionary Learning

Let $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N}$, $(\mathbf{y}_i \in \mathbb{R}^n)$ be a set of observations, measurements, or inputs. A sparse model poses each vector as a sparse linear combination of predictors, or atoms, from an overcomplete basis, or dictionary $\mathbf{D} \in \mathbb{R}^{n \times K}$, plus noise:

$$\mathbf{y} = \mathbf{D}\mathbf{x}_0 + \mathbf{n} \qquad \text{s.t.} \qquad \left\|\mathbf{x}_0\right\|_0 = T_0 \tag{1}$$

where T_0 is the support of the ideal sparse decomposition $\mathbf{x}_0$, $\left\|\cdot\right\|_0$ stands for the ℓ_0 pseudonorm (number of non–zero components), and $\mathbf{n}$ is the additive noise. Whereas sparse coding deals with estimating $\mathbf{x}_0$, i.e. inference on the sparse linear model, sparse modeling, or dictionary learning, refers to estimating the dictionary atoms under the conditions of (1).

A. OMP as a solution to the Sparse Coding problem

Orthogonal Matching Pursuit [11] attempts to find a local solution to (1) by iteratively estimating the most correlated atom in **D** to the current residue. Namely, for the *j*-th iteration:

$$\lambda_j = \underset{i\in\Omega}{\operatorname{argmax}} \left| \langle \mathbf{r}_{j-1}, \mathbf{d}_i \rangle \right| \tag{2}$$

where $\mathbf{r}_0 = \mathbf{y}$, $\Omega = \{1, 2, ..., K\}$, $\mathbf{d}_i$ is the *i*-th column of **D**, and $\langle \cdot, \cdot \rangle$ denotes inner product. The locally optimal atom is added to an active set Λ, i.e. $\Lambda_j = \Lambda_{j-1} \bigcup \lambda_j$. Then, the sparse code is estimated as:

$$\mathbf{x}_j = \underset{\mathbf{x}\in\mathbb{R}^K, \operatorname{supp}(\mathbf{x})\subset\Lambda_j}{\operatorname{argmin}} \left\| \mathbf{y} - \mathbf{D}\mathbf{x} \right\|_2 \tag{3}$$

which is solved via Ordinary Least Squares (OLS). The residual is then updated as $\mathbf{r}_j = \mathbf{y} - \mathbf{D}\mathbf{x}_j$. In practice, OMP runs for a fixed number of iterations, *L*, or until the norm of the residual error reaches a set lower bound. As (3) suggests, OMP inherently exploits MSE as the cost function to optimize; therefore, any form of outliers (in the Gaussian sense) would severely bias the resulting sparse code.

It is worth noting that OMP is not the only solution to the sparse coding problem. In fact, OMP is a refinement of Matching Pursuit [12]; whereas the former updates the entire support set via (3), the latter updates it sequentially one atom at the time. OMP also has refinements of its own—Generalized OMP (GOMP) [13], Regularized OMP (ROMP) [14], and CoSaMP [15] to name a few. On the other side of the greedy solutions, we can find relaxations of the model posed in (1). In particular, LASSO [16] or Basis Pursuit [17] showcase the success of ℓ_1–norm–based regularizers (or constraints) in linear problems. However, all of these solutions (either greedy or relaxations), rely on MSE to find the sparse codes and, hence, are prone to biased estimations in the presence of outliers or non–homogenous noise.

B. K-SVD as a solution to the Sparse Modeling problem

Sparse modeling usually refers to the problem of estimating the model parameters given an ensemble of training samples; one instance of such problem is dictionary learning. In particular, the goal is to estimate both sparse codes and dictionary atoms in a data–driven scheme, i.e.:

$$\min_{\mathbf{D},\mathbf{X}} \left\{ \left\| \mathbf{Y} - \mathbf{D}\mathbf{X} \right\|_F^2 \right\} \quad \text{s.t.} \quad \forall i, \ \left\| \mathbf{x}_i \right\|_0 \le T_0 \tag{4}$$

where $\mathbf{x}_i$ is the sparse code corresponding to the sample $\mathbf{y}_i$ and $\| \cdot \|_F$ denotes the Frobenius norm. The performance surface of (4) is non–convex; also, optimizing the linear problem under the ℓ_0 pseudonorm is combinatorial in nature and impractical in most cases. Therefore, greedy techniques are adopted instead. Namely, K-SVD [7] generalizes the well known clustering algorithm k-means to the sparse modeling framework. K-SVD alternates between sparse coding and dictionary update stages. The former admits any off–the–shelf sparse approximators, e.g. OMP, whereas the latter estimates the dictionary atoms in a sequential fashion.

The dictionary update stage starts by assuming that both **X** and K – 1 columns of **D** are fixed. Then, the atom in question, $\mathbf{d}_k$, and its support, $\mathbf{x}_T^k$, i.e. the *k*-th row of **X**, are jointly updated via:

$$\begin{aligned} \left\| \mathbf{Y} - \mathbf{D}\mathbf{X} \right\|_F^2 &= \left\| \mathbf{Y} - \sum_{j=1}^{K} \mathbf{d}_j \mathbf{x}_T^j \right\|_F^2 \\ &= \left\| \left(\mathbf{Y} - \sum_{j\neq k} \mathbf{d}_j \mathbf{x}_T^j \right) - \mathbf{d}_k \mathbf{x}_T^k \right\|_F^2 \\ &= \left\| \mathbf{E}_k - \mathbf{d}_k \mathbf{x}_T^k \right\|_F^2 \end{aligned} \tag{5}$$

where $\mathbf{E}_k$ is the error when the contribution from the *k*-th atom is removed. Lastly, both $\mathbf{d}_k$ and $\mathbf{x}_T^k$ are estimated via Singular Value Decomposition (SVD) of a restricted version of $\mathbf{E}_k$ in order to preserve the sparseness of the solution, i.e. only the columns of $\mathbf{E}_k$ that are currently active for $\mathbf{d}_k$ are part of the optimization of such atom. K-SVD then proceeds to update each dictionary atom sequentially to finish one single dictionary update round. The overall algorithm usually runs for a fixed number of alternating optimizations (sparse coding and dictionary update) or until a convergence criterion is met.

III. M-Estimators–based K-SVD

A. M-Estimators

Let $\Phi \in \mathbb{R}^{n\times k}$ be the active atoms in the dictionary **D** at OMP iteration *k*, and $\beta \in \mathbb{R}^k$ be the vector that solves the following regression problem:

$$\mathbf{y} = \Phi\beta + \mathbf{e} \tag{6}$$

where **e** is an error vector with i.i.d. components drawn from a zero–mean Normal density. The least squares solution is the maximum likelihood estimator for β under the condition of Gaussian errors. After maximizing the corresponding cost function, the well known normal equations give rise to the OLS estimator,

$$\hat{\beta}_{OLS} = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{y}, \tag{7}$$

which is optimal only for Gaussian errors scenarios. If such assumption is no longer valid—e.g. due to outliers or non–Gaussian environments—M-Estimators are a suitable alternative to solve (6).

In particular, M-Estimators exploit a different function to model the statistical properties of the errors:

$$\beta = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \rho\left(\frac{\mathbf{e}[i]}{s} \right) \tag{8}$$

where $\rho(\cdot)$ is a continuous, symmetric function (also known as

the objective function) with a global minimum when the argument is equal to zero [8]. Clearly, the objective function reduces to half the sum of squared errors for the Gaussian case, which confirms the equivalence of the MSE criterion and the OLS estimator under the umbrella of maximum likelihood estimation. s is an estimate of the scale of the errors; it is required in order to avoid biased solutions due to scale differences. Non–robust estimates of the scale—such as the standard deviation—cannot be utilized; thus, the "re–scaled MAD" is the usual choice:

$$s = 1.4826 \times \text{MAD} \tag{9}$$

where MAD (median absolute deviation) is highly robust against outliers because it relies on the median ($\tilde{e}$) of the errors instead of their mean [8]:

$$\text{MAD} = \text{median}\left|\mathbf{e}[i] - \tilde{e}\right|. \tag{10}$$

Likewise OLS, the optimal solution is obtained via partial differentiation of eq. (8) with respect to each of the k parameters in question. In particular, we define the weight function, **w,** as:

$$\mathbf{w}\left(\frac{\mathbf{e}[i]}{s}\right) = \frac{\psi\left(\frac{\mathbf{e}[i]}{s}\right)}{\frac{\mathbf{e}[i]}{s}}. \tag{11}$$

$\psi(\cdot)$ is known as the score function and defined as the derivative of $\rho(\cdot)$. After optimizing the cost function and rearranging terms (more details can be found in [10]), the equation to solve—in matrix form—is:

$$\mathbf{\Phi}^T\mathbf{W}\mathbf{\Phi}\beta = \mathbf{\Phi}^T\mathbf{W}\mathbf{y}. \tag{12}$$

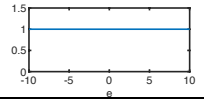
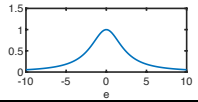
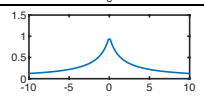
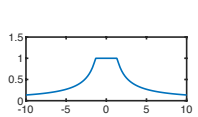
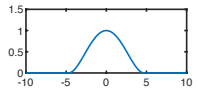
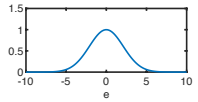
W is the square diagonal matrix with non–zero elements as the entries of the weight function **w**. Lastly, if $\mathbf{\Phi}^T\mathbf{W}\mathbf{\Phi}$ is well–conditioned, the robust M-Estimator is equal to:

$$\hat{\beta}_{M-Est} = (\mathbf{\Phi}^T\mathbf{W}\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{W}\mathbf{y} \tag{13}$$

Comparing equations (7) and (13), we can notice that M-Estimators incorporate a weight matrix that adaptively assigns larger values to samples from the main mode of the residuals and smaller weights, i.e. lower influence, to potential outliers. Many objective functions (and in turn, weight functions) have been proposed in the literature [18]; yet, we focus on five different variants that were thoroughly studied and validated on previous work [10]—Cauchy, Fair, Huber, Tukey and Welsch. Table I details the functional forms of such estimators alongside the benchmark MSE–based estimator: OLS. All the robust flavors under study concentrate higher weights around the zero–error mark and, then, smoothly decay the weights in a symmetrical fashion as the error increases.

Even though the optimal M-Estimator admits a closed form solution, solving for it is not as straightforward as its OLS counterpart. Namely, $\hat{\beta}_{M-Est}$ depends on **W**, which in turn, depends on the residuals and, thus, relies on $\hat{\beta}_{M-Est}$. A plausible solution to find both estimates is Iteratively Reweighted Least Squares or IRLS [8], which exploits BCD to achieve local solutions of (13). Algorithm 1 details the IRLS routine; it normally runs for a fixed number of iterations or until the inter–iteration error of the estimates reaches a set threshold. It is worth mentioning that IRLS uses the OLS solution as initialization to the BCD optimization.

TABLE I. OBJECTIVE AND WEIGHT FUNCTIONS OF OLS AND M-ESTIMATORS. EACH ROBUST VARIANT COMES WITH A HYPERPARAMETER "c". EXEMPLARY PLOTS UTILIZE THE OPTIMAL HYPERPARAMETERS DETAILED IN TABLE II.

Type	$\rho(e)$	$\mathbf{w}(e)$
OLS	$\frac{1}{2}e^2$	
Cauchy	$\frac{c^2}{2}\log\left(1+\left(\frac{e}{c}\right)^2\right)$	
Fair	$c^2\left(\frac{\lvert e\rvert}{c} - \log\left(1+\frac{\lvert e\rvert}{c}\right)\right)$	
Huber	$\begin{cases} \frac{e^2}{2} & ,\lvert e\rvert \le c \\ c\left(\lvert e\rvert - \frac{c}{2}\right) & ,\lvert e\rvert \ge c \end{cases}$	
Tukey	$\begin{cases} \frac{c^2}{6}\left(1-\left(1-\left(\frac{e}{c}\right)^2\right)^3\right) & ,\lvert e\rvert \le c \\ \frac{c^2}{6} & ,\lvert e\rvert > c \end{cases}$	
Welsch	$\frac{c^2}{2}\left(1-\exp\left(-\left(\frac{e}{c}\right)^2\right)\right)$	

Algorithm 1 IRLS–based M–Estimation

function IRLS($\mathbf{y} \in \mathbb{R}^n, \Phi \in \mathbb{R}^{n\times k}, w_c(u)$)
  $t \leftarrow 0$
  $\beta^{(0)} = \beta_{\text{OLS}} \leftarrow (\Phi^T\Phi)^{-1}\Phi^T\mathbf{y}$ ▷ OLS initialization
  $\mathbf{e}^{(0)} \leftarrow \mathbf{y} - \Phi\beta^{(0)}$
  $MAD \leftarrow \text{median}|\mathbf{e}^{(0)}[i] - \tilde{\mathbf{e}}^{(0)}|$
  $s^{(0)} \leftarrow 1.4826 \times MAD$
  $\mathbf{w}^{(0)}[i] \leftarrow w_c\left(\frac{\mathbf{e}^{(0)}[i]}{s^{(0)}}\right) \quad i = 1, 2, \ldots, n$
  $\mathbf{W}^{(0)} \leftarrow \text{diag}\left(\mathbf{w}^{(0)}\right)$
  $t \leftarrow 1$
  while NO CONVERGENCE **do**
    $\beta^{(t)} \leftarrow (\Phi^T\mathbf{W}^{(t-1)}\Phi)^{-1}\Phi^T\mathbf{W}^{(t-1)}\mathbf{y}$ ▷ BCD
    $\mathbf{e}^{(t)} \leftarrow \mathbf{y} - \Phi\beta^{(t)}$
    $MAD \leftarrow \text{median}|\mathbf{e}^{(t)}[i] - \tilde{\mathbf{e}}^{(t)}|$
    $s^{(t)} \leftarrow 1.4826 \times MAD$
    $\mathbf{w}^{(t)}[i] \leftarrow w_c\left(\frac{\mathbf{e}^{(t)}[i]}{s^{(t)}}\right) \quad i = 1, 2, \ldots, n$ ▷ BCD
    $\mathbf{W}^{(t)} \leftarrow \text{diag}\left(\mathbf{w}^{(t)}\right)$
    $t \leftarrow t + 1$
  return $\hat{\beta}_{\text{M–Est}} \leftarrow \beta^{(t)} \quad \hat{\mathbf{w}}_{\text{M–Est}} \leftarrow \mathbf{w}^{(t)}$

Lastly, Table II details the M-Estimator hyperparameters, c, that achieve a 95% asymptotic efficiency on the standard Normal distribution. Throughout this work, we exploit such optimal values to avoid parameter tuning by the user.

TABLE II. OPTIMAL M-ESTIMATOR HYPERPARAMETERS, C.

Cauchy	Fair	Huber	Tukey	Welsch
2.385	1.4	1.345	4.685	2.985

B. MeK-SVD

M-Estimators–based K-SVD incorporates robustness into the sparse coding stage of K-SVD. Specifically, instead of exploiting equation (7) to update the sparse code and solve (3), the proposed approach utilizes equation (13) alongside its IRLS solution procedure to estimate both the robust sparse code and weight vector for all the dimensions in question; this robust sparse inference algorithm was developed in [10] under the name of Robust OMP (RobOMP).

After the robust sparse representation is estimated, the dictionary update stage proceeds as usual until convergence. The overall learning framework has the same number of hyperparameters as its K-SVD predecessor, i.e. a stopping criterion for RobOMP (either a fixed number of iterations, L, or a minimum reconstruction error threshold), and the number of dictionary atoms to be estimated: k.

IV. RESULTS

A. Recovery of Ground Truth Dictionaries

The first set of results deals with estimation of dictionary atoms for synthetic data where the ground truth is available. The dictionary, $\mathbf{D} \in \mathbb{R}^{20\times 50}$, is generated by sampling a zero–mean uniform distribution with support [-1,1]. Each column is further normalized to have unit ℓ_2–norm. Then, the sparse representation coefficients are generated from a uniform distribution with support [0,1]. 3 codes fully represent a synthetic sample, i.e. $T_0 = 3$. 1500 20–dimensional vectors are generated via linear combinations of the sparse codes and dictionary. Lastly, these samples are affected by non–linear, non–homogeneous noise in the form of missing entries, i.e. a percentage of components from each observation is selected at random and set to zero. This percentage is varied from 0 to 50%.

We compare the performance of traditional K-SVD to its M-Estimators counterparts. In particular, we refer to each dictionary learning scheme as its underlying sparse coder, e.g. K-SVD is simply denoted as OMP. Each dictionary learning algorithm performs 40 alternating runs between sparse coding and dictionary update stages. All the sparse coders run for a total of 3 successive iterations, i.e. $L = 3$, and the number of estimated atoms is taken from the ground truth ($k = 50$).

Fig. 1 summarizes the results for a total of 50 independent runs per noise rate/dictionary learning technique in terms of average inner product between the estimated atoms and the ground truth generating dictionary. All algorithms are able to learn the generating dictionary for small amounts of missing entries. However, as the intensity of the noise increases, OMP–based K-SVD yields suboptimal dictionaries, whereas the robust alternatives consistently outperform the state of the art. In particular, the Tukey and Welsch variants rise above the rest of the MeK-SVD flavors. This provides a proof of concept of the robustness achieved when M-Estimators are exploited as part of the dictionary learning framework.

B. Image Denoising Exploiting Redundant Representations

The next set of results focuses on image denoising based on sparse and redundant representations, as proposed in [4]. Essentially, the denoising framework performs dictionary learning over local patches of the noisy image (usually 8×8 pixels in size). First, each patch is sparsely encoded using a stopping criterion based on the residue norm equal to 1.15σ, where σ is the standard deviation of the noise source: homogenous additive Gaussian noise. Then, the dictionary atoms are updated via successive SVD routines that solve (5). Lastly, the denoised image is the result of a combination of local averaging of overlapping patches and global averaging with the original, noisy example. One of the main hyperparameters of the framework, Lagrange multiplier λ, is set equal to 30 according to the authors in [4].

The first set of experiments deals with salt and pepper noise on top of homogenous additive Gaussian noise with known standard deviation. For our case, we chose $\sigma = 10$ and the rate of salt and pepper affected pixels is varied from 0 to 20%. All possible vectorized 8×8 patches constitute the observations, $\mathbf{Y}$, of the model while k is set equal to 256, i.e. $\mathbf{D} \in \mathbb{R}^{64\times 256}$. The overcomplete Discrete Cosine Transform (DCT) basis is chosen as the initialization of the dictionary. Lastly as suggested by [4], we ran a total of 10 alternating optimizations between robust sparse coding and dictionary update stages.

Table III details the grand average PSNRs over 5 independent runs for each salt and pepper noise rate and five different well known 512×512 gray–scale images: Lena, Barbara, Boats, House, and Peppers. For proper comparison purposes, we ran similar denoising schemes exploiting off–the–shelf dictionaries: the overcomplete DCT basis and a "global" dictionary obtained from random sampling of natural images. These two fixed dictionary algorithms are contrasted to the adaptive case, i.e. dictionary learning, for all of the robust variants under study and the K-SVD benchmark.

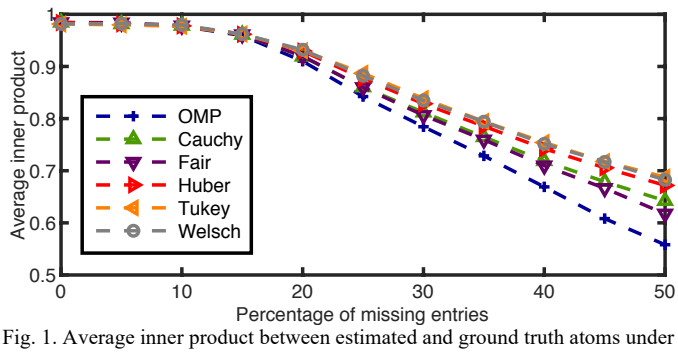


Fig. 1. Average inner product between estimated and ground truth atoms under missing entries type of noise.

TABLE III. SUMMARY OF DENOISING PERFORMANCE, PSNR GRAND AVERAGE (DECIBELS), UNDER DIFFERENT RATES OF SALT AND PEPPER NOISE. EACH CELL REPORTS OVERCOMPLETE DCT (UPPER ROWS), GLOBAL TRAINED DICTIONARY (MIDDLE ROWS), AND ADAPTIVE DICTIONARY, I.E. K-SVD–BASED (BOTTOM ROWS). BEST RESULTS FOR EACH DICTIONARY CASE ARE MARKED BOLD. STAR INDICATES BEST OVERALL RESULT FOR EACH RATE CASE.

S&P Rate	Sparse coder variant					
	OMP	Cauchy	Fair	Huber	Tukey	Welsch
0.00	32.29	**32.34**	32.11	32.27	31.69	31.73
	32.37	**32.43**	32.27	32.35	32.18	32.18
	34.75*	34.25	33.61	34.10	33.67	33.61
0.05	18.51	23.23	20.77	23.83	**24.30**	24.28
	18.48	22.36	20.28	22.91	23.70	**23.71**
	18.70	21.39	19.98	23.92	**24.38***	24.36
0.10	15.47	18.90	16.34	20.11	**21.73**	21.68
	15.45	18.19	16.28	19.25	**20.73**	20.71
	15.68	18.09	16.54	21.21	**24.04***	24.00
0.15	13.69	15.67	14.11	16.83	**19.99**	19.88
	13.68	15.44	14.14	16.47	**18.72**	18.67
	13.84	15.58	14.50	17.17	**24.01***	23.80
0.20	12.43	13.46	12.71	14.26	**18.31**	18.15
	12.43	13.56	12.75	14.32	**17.12**	17.02
	12.54	13.39	12.99	14.36	**21.53***	20.76

It is evident that OMP–based K-SVD degrades quickly in the presence of outliers, e.g. roughly a 16 dB drop from 0% to 5% of salt and pepper affected pixels. Conversely, robust variants (e.g. Tukey and Welsch) do not experience such drastic drop in performance. Also in general, the adaptive dictionary versions yield higher PSNRs than their fixed dictionary counterparts, which confirms the need for data–driven, adaptive solutions. Fig. 2 illustrates the denoising outcomes on the image "House".

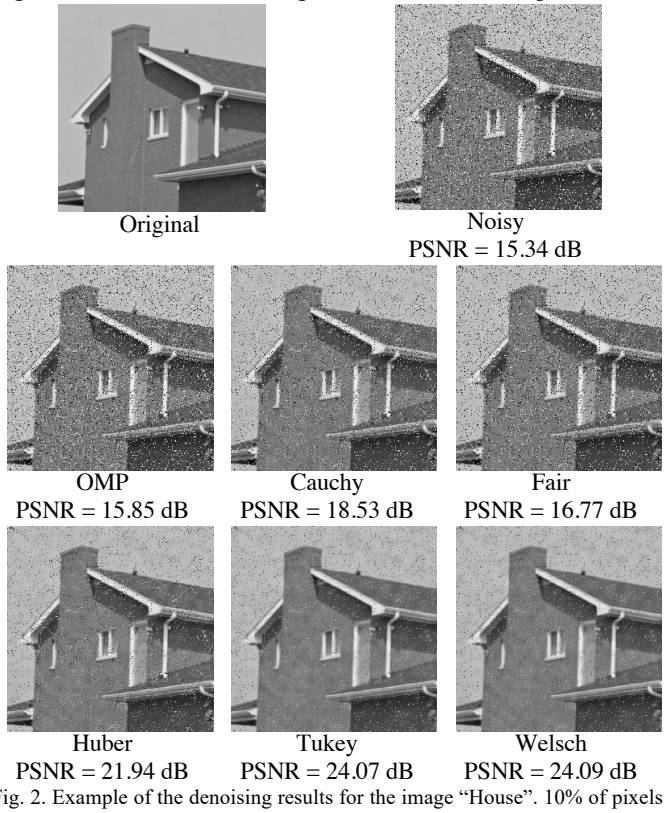


Fig. 2. Example of the denoising results for the image "House". 10% of pixels are affected by salt and pepper noise. Adaptive dictionary cases.

Next, we investigate the effect of impulsive noise, i.e. high–power noise at low rates, on the denoising schemes. We still operate under initial additive homogenous Gaussian noise with $\sigma = 10$. Then, we mimic a non–linear transformation by adding high–variance Gaussian noise to a low percentage of pixels. Due to the restricted dynamic range of the inputs, adding impulsive noise will result in some pixels to saturate to either 0 to 255. Once again, $k = 256$. Table IV summarizes similar grand average PSNRs as Table III's for different impulsive noise powers. The robust variants of K-SVD consistently outperform the state of the art for various impulsive noise powers at 5% of incidence. Moreover, Table V compliments the analysis by providing performance measures with respect to several rates for a given impulsive noise standard deviation of 50.

TABLE IV. SUMMARY OF DENOISING PERFORMANCE, PSNR GRAND AVERAGE (DECIBELS), UNDER DIFFERENT IMPULSIVE NOISE POWERS (5%). EACH CELL REPORTS OVERCOMPLETE DCT (UPPER ROWS), GLOBAL TRAINED DICTIONARY (MIDDLE ROWS), AND ADAPTIVE DICTIONARY, I.E. K-SVD–BASED (BOTTOM ROWS). BEST RESULTS FOR EACH DICTIONARY CASE ARE MARKED BOLD. STAR INDICATES BEST OVERALL RESULT FOR EACH NOISE CASE.

Imp. Noise σ	Sparse coder variant					
	OMP	Cauchy	Fair	Huber	Tukey	Welsch
50	27.25	29.65	29.62	**29.77**	29.62	29.63
	27.16	29.41	29.30	29.52	**29.70**	29.68
	27.48	29.97	29.56	**30.35***	30.15	30.13
60	26.13	28.99	28.91	**29.17**	29.11	29.11
	26.04	28.72	28.49	28.87	**29.13**	29.11
	26.16	28.74	28.28	**29.30***	29.21	29.19
70	25.19	28.42	28.24	28.64	**28.66***	28.66
	25.12	28.11	27.72	28.29	**28.62**	28.59
	25.16	27.76	27.23	**28.44**	28.41	28.36
80	24.41	27.90	27.61	28.17	**28.24***	28.24
	24.36	27.57	27.03	27.78	**28.16**	28.15
	24.35	26.93	26.32	27.74	**27.75**	27.71

TABLE V. SUMMARY OF DENOISING PERFORMANCE, PSNR GRAND AVERAGE (DECIBELS), UNDER DIFFERENT IMPULSIVE NOISE RATES ($\sigma = 50$). EACH CELL REPORTS OVERCOMPLETE DCT (UPPER ROWS), GLOBAL TRAINED DICTIONARY (MIDDLE ROWS), AND ADAPTIVE DICTIONARY, K-SVD–BASED (BOTTOM ROWS). BEST RESULTS FOR EACH DICTIONARY CASE ARE MARKED BOLD. STAR INDICATES BEST OVERALL RESULT FOR EACH NOISE CASE.

Imp. Noise Rate	Sparse coder variant					
	OMP	Cauchy	Fair	Huber	Tukey	Welsch
0.05	27.25	29.65	29.62	**29.77**	29.62	29.63
	27.16	29.41	29.30	29.52	**29.70**	29.68
	27.48	29.97	29.56	**30.35***	30.15	30.13
0.10	24.67	27.60	27.65	28.03	**28.22**	28.22
	24.66	27.36	27.28	27.77	**28.17**	28.14
	25.04	28.19	27.71	29.07	**29.12***	29.08
0.15	23.01	25.76	25.81	26.54	**27.09**	27.07
	22.99	25.58	25.49	26.34	**26.97**	26.94
	23.69	27.09	26.25	28.38	**28.58***	28.54
0.20	21.78	24.08	24.12	25.12	**26.08**	26.03
	21.76	24.00	23.90	25.05	**25.96**	25.91
	22.48	25.99	25.18	27.85	**28.25***	28.20
0.25	20.82	22.63	22.68	23.76	**25.11**	25.02
	20.81	22.67	22.57	23.86	**25.04**	24.97
	21.43	24.49	24.30	26.80	**28.05***	28.02

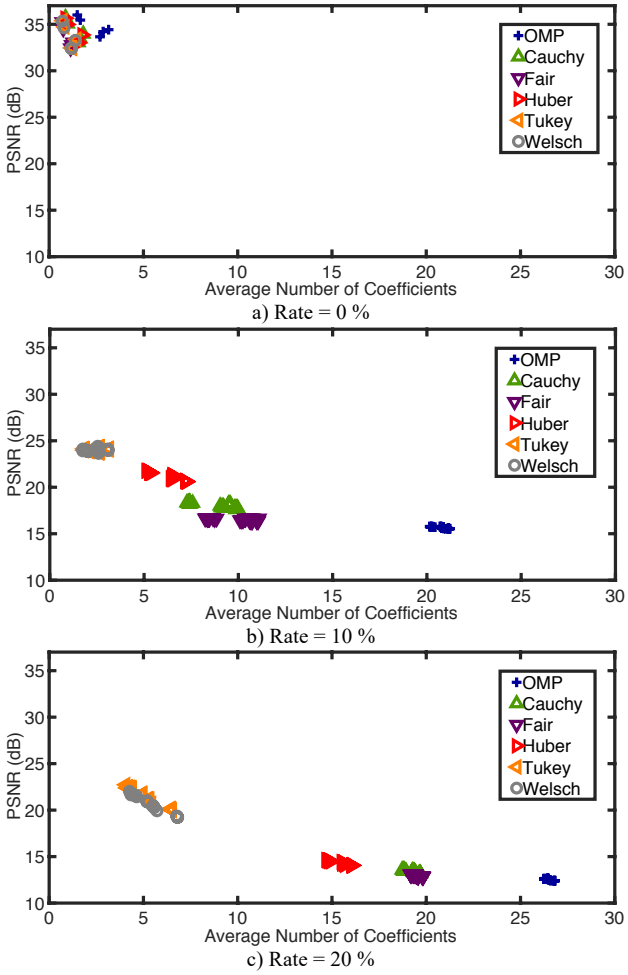


Fig. 3. Relation between denoising performance (PSNR) and sparseness of dictionary learning solutions. Salt and pepper noise. Adaptive dictionary cases.

Lastly, we study the sparseness of the solutions in terms of average number of coefficients per sample, i.e. average ℓ_0 pseudonorm of the support set of the sparse codes per 8×8 patch. We use the salt and pepper noise scenario to illustrate the relation between denoising performance (PSNR) and sparseness of the representation. Fig. 3 summarizes the results for three noise rate cases. The results suggest that, in the presence of outliers, OMP–based K-SVD not only underperforms in terms of denoising, but it also overrepresents the inputs, which essentially defeats the purpose of a sparse modeling framework. On the other hand, the MeK-SVD flavors still encourage sparse solutions (specially Tukey and Welsch) while, at the same time, yield improved outcomes. Consequently, M-Estimator–based dictionary learning techniques might be suitable for image compression and denoising under challenging scenarios. This hypothesis will be explored as further work.

V. Conclusion

We proposed a robust alternative to the well known dictionary learning technique, K-SVD. RobOMP [10] exploits M-Estimators to obtain robust sparse codes and improve not only performance, in terms of estimation bias, but also the overall sparseness of the solutions. Empirically, we found that the Tukey and Welsch variants stand out from the rest. Yet, more detailed analysis is needed in order to assess which weight function (and associated hyperparameter) is indeed the most robust. Further work might involve incorporating robustness into the remaining stage of K-SVD—dictionary update—by means of robust SVD algorithms, such as the one outlined in [19].

References

[1] Olshausen, Bruno A., and David J. Field. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images." *Nature* 381.6583 (1996): 607.

[2] Elad, Michael. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.

[3] Rudin, Leonid I., Stanley Osher, and Emad Fatemi. "Nonlinear total variation based noise removal algorithms." *Physica D: nonlinear phenomena* 60.1-4 (1992): 259-268.

[4] Elad, Michael, and Michal Aharon. "Image denoising via sparse and redundant representations over learned dictionaries." *IEEE Transactions on Image processing* 15.12 (2006): 3736-3745.

[5] Mairal, Julien, Michael Elad, and Guillermo Sapiro. "Sparse representation for color image restoration." *IEEE Transactions on image processing* 17.1 (2007): 53-69.

[6] Mairal, Julien, et al. "Non-local sparse models for image restoration." *ICCV*. Vol. 29. 2009.

[7] Aharon, Michal, Michael Elad, and Alfred Bruckstein. "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation." *IEEE Transactions on signal processing* 54.11 (2006): 4311-4322.

[8] Andersen, Robert. *Modern methods for robust regression*. No. 152. Sage, 2008.

[9] Huber, Peter J. *Robust statistics*. Springer Berlin Heidelberg, 2011.

[10] Loza CA. 2019. RobOMP: Robust variants of Orthogonal Matching Pursuit for sparse representations. *PeerJ Computer Science* 5:e192

[11] Tropp, Joel A., and Anna C. Gilbert. "Signal recovery from random measurements via orthogonal matching pursuit." *IEEE Transactions on information theory* 53.12 (2007): 4655-4666.

[12] Mallat, Stéphane G., and Zhifeng Zhang. "Matching pursuits with time-frequency dictionaries." *IEEE Transactions on signal processing* 41.12 (1993): 3397-3415.

[13] Wang, Jian, Seokbeop Kwon, and Byonghyo Shim. "Generalized orthogonal matching pursuit." *IEEE Transactions on signal processing* 60.12 (2012): 6202-6216.

[14] Needell, Deanna, and Roman Vershynin. "Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit." *arXiv preprint arXiv:0712.1360*(2007).

[15] Needell, Deanna, and Joel A. Tropp. "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples." *Applied and computational harmonic analysis* 26.3 (2009): 301-321.

[16] Tibshirani, Robert. "Regression shrinkage and selection via the lasso." *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996): 267-288.

[17] Chen, Scott Shaobing, David L. Donoho, and Michael A. Saunders. "Atomic decomposition by basis pursuit." *SIAM review*43.1 (2001): 129-159.

[18] Zhang, Zhengyou. "Parameter estimation techniques: A tutorial with application to conic fitting." *Image and vision Computing*15.1 (1997): 59-76.

[19] Loza, Carlos A., and Jose C. Principe. "A robust maximum correntropy criterion for dictionary learning." *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2016.