Regularization by Denoising: Clarifications and New Interpretations

Phil Schniter and Ted Reehorst

The Ohio State University

With support from: NSF CCF-1527162 and NSF CCF-1716388

SIAM Conference on Imaging Science (Bologna, Italy) — June 7, 2018

Outline

- Introduction to RED
- Clarifications on RED
- New Interpretation of RED
- Fast RED Algorithms

Inverse Problems in Imaging

Inverse problems in imaging:

$$\text{Recover } \boldsymbol{x}^0 \text{ from measurements } \boldsymbol{y} = \text{corrupted}(\boldsymbol{A}\boldsymbol{x}^0)$$

where $\boldsymbol{A}$ is a known linear operator.

In this talk, we'll focus on additive white Gaussian noise (AWGN):

$$\text{Recover } \boldsymbol{x}^0 \text{ from measurements } \boxed{\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^0 + \boldsymbol{e} \text{ with } \boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}).}$$

Other corruptions include loss of phase, quantization, Poisson arrivals...

The Variational Approach and MAP Estimation

The variational approach to recovering $\boldsymbol{x}$ solves an optimization problem:

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left\{ \ell(\boldsymbol{x};\boldsymbol{y}) + \lambda\rho(\boldsymbol{x}) \right\} \text{ with } \begin{cases} \ell(\boldsymbol{x};\boldsymbol{y}): & \text{loss function} \\ \rho(\boldsymbol{x}): & \text{regularization} \\ \lambda > 0: & \text{tuning parameter} \end{cases}$$

Can be interpreted as Bayesian MAP estimation:

$$\widehat{\boldsymbol{x}}_{\mathsf{map}} = \arg\min_{\boldsymbol{x}} \left\{ -\ln p(\boldsymbol{y}|\boldsymbol{x}) - \ln p(\boldsymbol{x}) \right\} \text{ with } \begin{cases} p(\boldsymbol{y}|\boldsymbol{x}): & \text{likelihood} \\ p(\boldsymbol{x}): & \text{prior} \end{cases}$$

AWGN likelihood implies quadratic loss $\ell(\boldsymbol{x};\boldsymbol{y}) = \frac{1}{2\sigma^2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2$.
But how should we choose the regularization $\rho(\cdot)$?

Regularization by Denoising (RED)

Recently, Romano, Elad and Milanfar[1] proposed the RED regularization

$$\boxed{\rho_{\text{red}}(\boldsymbol{x}) \triangleq \frac{1}{2}\boldsymbol{x}^\top(\boldsymbol{x}-\boldsymbol{f}(\boldsymbol{x}))},$$

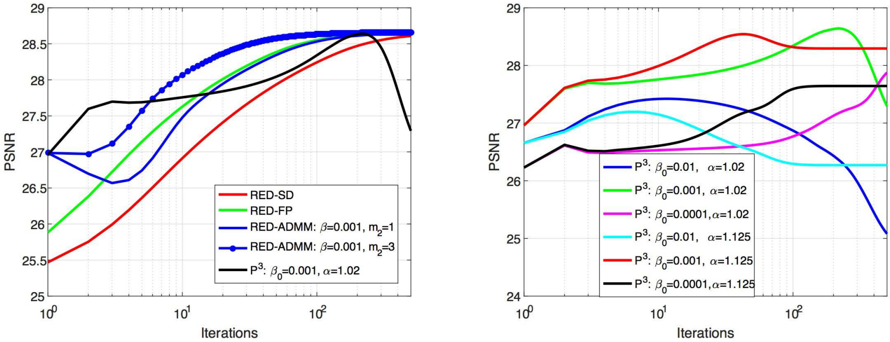
where $\boldsymbol{f}:\mathbb{R}^N \to \mathbb{R}^N$ is an image denoising function (e.g., BM3D).

RED leads to a family of "plug-and-play" (PnP) algorithms, similar to those proposed by Bouman et al.[2] and Metzler et al.[3], but with some advantages.

[1] Romano,Elad,Milanfar'17, [2] Venkatakrishnan,Bouman,Wolhberg'13, [3] Metzler,Maleki,Baraniuk'15

RED versus PnP

Experiments in the RED paper[1] show advantages of RED algs over PnP:



Above represents super-resolution recovery averaged over 10 test images.

Claims about RED

The RED paper[1] claims ...

1. If $\boldsymbol{f}(\cdot)$ is locally homogeneous (LH), i.e.,

$$\boxed{\boldsymbol{f}\big((1+\epsilon)\boldsymbol{x}\big) = (1+\epsilon)\boldsymbol{f}(\boldsymbol{x}) \quad \text{for small } \epsilon},$$

   and differentiable, then gradient of $\rho_{\text{red}}(\boldsymbol{x}) \triangleq \frac{1}{2}\boldsymbol{x}^\top(\boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x}))$ obeys

$$\boxed{\nabla \rho_{\text{red}}(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x})}.$$

2. If the Jacobian $J\boldsymbol{f}(\boldsymbol{x})$ is strongly passive, i.e.,

$$\|J\boldsymbol{f}(\boldsymbol{x})\|_2 \leq 1,$$

   then the RED regularization $\rho_{\text{red}}(\boldsymbol{x})$ is convex.

Implications of RED Claims

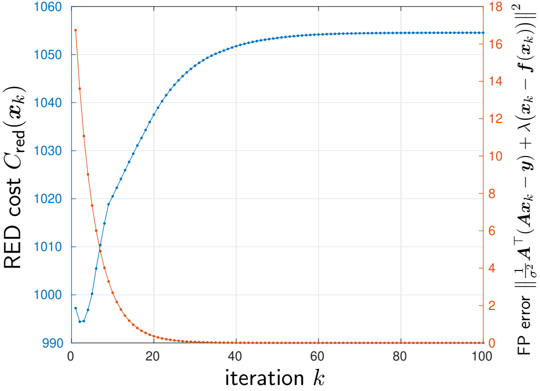
- The convexity claim on $\rho_{\text{red}}(\cdot)$ implies that minimization of
$$C_{\text{red}}(\boldsymbol{x}) \triangleq \frac{1}{2\sigma^2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|^2 + \lambda\rho_{\text{red}}(\boldsymbol{x})$$
can be easily tackled by many algs (e.g., SD, ADMM, etc.).

- The gradient claim $\nabla\rho_{\text{red}}(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x})$ implies the minimizers obey
$$\boxed{\text{RED fixed-point condition:} \quad \frac{1}{\sigma^2}\boldsymbol{A}^\top(\boldsymbol{A}\widehat{\boldsymbol{x}}-\boldsymbol{y}) + \lambda\big(\widehat{\boldsymbol{x}} - \boldsymbol{f}(\widehat{\boldsymbol{x}})\big) = \boldsymbol{0}}$$
The RED algorithms find exactly these $\widehat{\boldsymbol{x}}$.

Mysterious Behavior

Surprisingly, the RED algorithms do not always behave as expected!

We expect SD to decrease the (convex) RED cost, but it is increasing it!



$$\text{RED-SD:} \quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \mu \nabla C_{\text{red}}(\boldsymbol{x}_k)$$

Clarifications on RED Gradient

It can be shown that...

- differentiability in $\boldsymbol{f}(\cdot)$ implies

$$\nabla\rho_{\text{red}}(\boldsymbol{x}) \stackrel{\text{D}}{=} \boldsymbol{x} - \frac{1}{2}\boldsymbol{f}(\boldsymbol{x}) - \frac{1}{2}[J\boldsymbol{f}(\boldsymbol{x})]^\top \boldsymbol{x}.$$

- adding local-homogeneity (LH) gives

$$\nabla\rho_{\text{red}}(\boldsymbol{x}) \stackrel{\text{D,LH}}{=} \boldsymbol{x} - \frac{1}{2}[J\boldsymbol{f}(\boldsymbol{x})]\boldsymbol{x} - \frac{1}{2}[J\boldsymbol{f}(\boldsymbol{x})]^\top \boldsymbol{x}.$$

- adding Jacobian symmetry (JS) finally leads to

$$\nabla\rho_{\text{red}}(\boldsymbol{x}) \stackrel{\text{D,LH,JS}}{=} \boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x}) \quad \text{...which yields the RED algorithms.}$$

So both LH and JS are needed to link RED cost to RED algs.

Which Denoisers Yield Jacobian Symmetry?

Clear that these yield JS:

- Linear denoisers $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x}$ with $\boldsymbol{W} = \boldsymbol{W}^\top$.
- Transform-domain-thresholding (TDT) denoisers $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W}^\top \boldsymbol{g}(\boldsymbol{W}\boldsymbol{x})$.
- MAP or MMSE denoisers under any assumed prior $\boldsymbol{x} \sim \widehat{p}_{\mathsf{x}}$.

Not clear that these yield JS:

- Pseudo-linear denoisers $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W}(\boldsymbol{x})\boldsymbol{x}$ with non-linear $\boldsymbol{W}(\cdot)$.
- *Approximately* MAP or MMSE denoisers.

Most state-of-the-art denoisers fall into the 2nd category.

Jacobian Symmetry Experiments

Avg JS error on suite of 16×16 images:

	TDT	MF	NLM	BM3D	TNRD	DnCNN
$\frac{\lVert \widehat{J\boldsymbol{f}}(\boldsymbol{x}) - [\widehat{J\boldsymbol{f}}(\boldsymbol{x})]^\top \rVert_F^2}{\lVert \widehat{J\boldsymbol{f}}(\boldsymbol{x}) \rVert_F^2}$	4.11e-21	1.35	0.118	0.186	0.0151	0.194

Avg gradient error on suite of 16×16 images:

$\frac{\lVert \nabla \rho_{\text{red}}(\boldsymbol{x}) - \widehat{\nabla \rho_{\text{red}}}(\boldsymbol{x}) \rVert^2}{\lVert \widehat{\nabla \rho_{\text{red}}}(\boldsymbol{x}) \rVert^2}$	TDT	MF	NLM	BM3D	TNRD	DnCNN
$\nabla \rho_{\text{red}}(\boldsymbol{x})$ with D	3.39e-19	2.65e-15	6.17e-21	2.14e-13	5.42e-17	1.02e-12
$\nabla \rho_{\text{red}}(\boldsymbol{x})$ with D,LH,JS	0.565	0.966	0.913	1.00	0.957	0.852

Key points:

1. Large JS error for all but TDT.
2. Large gradient error under JS & LH assumptions for all denoisers!
3. Even TDT has large gradient error! Is LH the problem?

Local Homogeneity Experiments

Avg LH error on suite of 16×16 images:

	TDT	MF	NLM	BM3D	TNRD	DnCNN
$\frac{\\|\boldsymbol{f}((1+\epsilon)\boldsymbol{x})-(1+\epsilon)\boldsymbol{f}(\boldsymbol{x})\\|^2}{\\|(1+\epsilon)\boldsymbol{f}(\boldsymbol{x})\\|^2}$	7.99e-10	0	5.60e-9	1.52e-13	5.09e-10	2.06e-9
$\frac{\\|[\widehat{J\boldsymbol{f}}(\boldsymbol{x})]\boldsymbol{x}-\boldsymbol{f}(\boldsymbol{x})\\|^2}{\\|\boldsymbol{f}(\boldsymbol{x})\\|^2}$	4.10e-4	2.14e-15	5.63e-3	0.214	2.60e-4	8.02e-3

Avg gradient error on suite of 16×16 images:

$\frac{\\|\nabla\rho_{\text{red}}(\boldsymbol{x})-\widehat{\nabla\rho_{\text{red}}}(\boldsymbol{x})\\|^2}{\\|\widehat{\nabla\rho_{\text{red}}}(\boldsymbol{x})\\|^2}$	TDT	MF	NLM	BM3D	TNRD	DnCNN
$\nabla\rho_{\text{red}}(\boldsymbol{x})$ with D	3.39e-19	2.65e-15	6.17e-21	2.14e-13	5.42e-17	1.02e-12
$\nabla\rho_{\text{red}}(\boldsymbol{x})$ with D,LH	0.565	6.09e-15	0.0699	0.344	0.139	1.20

Key points:

- It is important how LH is quantified.
- The RED gradient is very sensitive to small imperfections in LH.

Implications of our Findings

We found:

- The RED algorithms solve a fixed-point equation corresponding to $\nabla\rho(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x})$.
- $\boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x})$ is very different from $\nabla\rho_{\text{red}}(\boldsymbol{x})$ under practical $\boldsymbol{f}(\cdot)$, such as TDT, MF, NLM, BM3D, TNRD, and DnCNN.

Implication:

- $\rho_{\text{red}}(\cdot)$ does not explain the RED algorithms under practical $\boldsymbol{f}(\cdot)$.

A bigger problem:

- For non-JS $\boldsymbol{f}(\cdot)$, can show that there exists no explicit regularizer $\rho(\cdot)$ for which $\nabla\rho(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x})$, i.e., explaining the RED algorithms!

How to Explain the RED Algorithms?

The RED algorithms assume $\boxed{\nabla\rho(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x})}$ and work very well.

Can we justify this $\nabla\rho(\boldsymbol{x})$?
Even when $\boldsymbol{f}(\cdot)$ is not LH and/or JS?

Yes! Using score matching. We explain this in 3 steps:

1. regularization by log-likelihood (RLL),
2. RLL as kernel density estimation (KDE),
3. score matching.

Regularization by Log-Likelihood (RLL)

- Consider noisy pseudo-measurements

$$\boldsymbol{r} = \boldsymbol{x}^0 + \mathcal{N}(0, \nu \boldsymbol{I}).$$

- Suppose we adopt the prior pdf $\widehat{p}_{\mathsf{x}}$. Then the likelihood of $\boldsymbol{r}$ is

$$\widehat{p}_{\mathsf{r}}(\boldsymbol{r}; \nu) = \int_{\mathbb{R}^N} \mathcal{N}(\boldsymbol{r}; \boldsymbol{x}, \nu \boldsymbol{I})\, \widehat{p}_{\mathsf{x}}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}.$$ "Gaussian blurred prior"

- Define the RLL regularization as

$$\boxed{\rho_{\mathsf{LL}}(\boldsymbol{r}; \nu) \triangleq -\nu \ln \widehat{p}_{\mathsf{r}}(\boldsymbol{r}; \nu)}$$

- Then it can be shown using Tweedie's formula[4] that

$$\nabla \rho_{\mathsf{LL}}(\boldsymbol{r}; \nu) = \boldsymbol{r} - \widehat{\boldsymbol{f}}_{\mathsf{mmse},\nu}(\boldsymbol{r}),$$

which is consistent with the RED algorithms!

[4] Robbins'56

RLL as Kernel Density Estimation

- Given training data $\{\boldsymbol{x}_t\}_{t=1}^T$, consider the empirical prior

$$\widehat{p}_{\mathsf{x}}(\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T}\delta(\boldsymbol{x}-\boldsymbol{x}_t).$$

- A better match to the true p_{x} is obtained via KDE or Parzen windowing:

$$\widetilde{p}_{\mathsf{x}}(\boldsymbol{x};\nu) = \frac{1}{T}\sum_{t=1}^{T}\mathcal{N}(\boldsymbol{x};\boldsymbol{x}_t,\nu\boldsymbol{I}). \qquad \text{“blurred empirical prior”}$$

- Using this $\widetilde{p}_{\mathsf{x}}$ for MAP/variational optimization yields

$$\begin{aligned}\widehat{\boldsymbol{x}} &= \arg\min_{\boldsymbol{x}} \tfrac{1}{2\sigma^2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|^2 - \ln \widetilde{p}_{\mathsf{x}}(\boldsymbol{x};\nu)\\ &= \arg\min_{\boldsymbol{x}} \tfrac{1}{2\sigma^2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|^2 + \lambda\rho_{\mathsf{LL}}(\boldsymbol{x};\nu) \text{ for } \lambda=\tfrac{1}{\nu}.\end{aligned}$$

So RLL arises naturally in non-parametric estimation via KDE!

Score-Matching by Denoising

- The above RLL/KDE framework encompasses only JS denoisers $\boldsymbol{f}(\cdot)$. We now generalize.
- First note that, for large # of examples T, gradient is very expensive:

$$\nabla \ln \widetilde{p}_{\mathsf{x}}(\boldsymbol{x};\nu) = \frac{\widehat{\boldsymbol{f}}_{\mathsf{mmse},\nu}(\boldsymbol{x}) - \boldsymbol{x}}{\nu} \text{ with } \widehat{\boldsymbol{f}}_{\mathsf{mmse},\nu}(\boldsymbol{x}) = \frac{\sum_{t=1}^{T}(\boldsymbol{x}_t - \boldsymbol{x})\mathcal{N}(\boldsymbol{x};\boldsymbol{x}_t,\nu\boldsymbol{I})}{\sum_{t=1}^{T}\mathcal{N}(\boldsymbol{x};\boldsymbol{x}_t,\nu\boldsymbol{I})}.$$

- Practical idea:[5] use best match to "score" $\nabla \ln \widetilde{p}_{\mathsf{x}}(\boldsymbol{x})$ among computationally friendly functions $\psi(\boldsymbol{x};\boldsymbol{\theta})$:

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\widetilde{p}_{\mathsf{x}}}\left\{\left\|\boldsymbol{\psi}(\boldsymbol{x};\widehat{\boldsymbol{\theta}}) - \nabla \ln \widetilde{p}_{\mathsf{x}}(\boldsymbol{x};\nu)\right\|^2\right\}.$$

- Vincent[6] connected to denoising: if $\psi(\boldsymbol{x};\boldsymbol{\theta}) = [\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}) - \boldsymbol{x}]/\nu$, then

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\{\left\|\boldsymbol{f}_{\boldsymbol{\theta}}\big(\boldsymbol{x}_t + \mathcal{N}(0,\nu\boldsymbol{I})\big) - \boldsymbol{x}_t\right\|^2\right\},$$

where $\boldsymbol{f}_{\widehat{\boldsymbol{\theta}}}(\cdot)$ is MMSE optimal $\boldsymbol{f}_{\boldsymbol{\theta}} \in \mathcal{F}$, where $\mathcal{F} \triangleq \{\boldsymbol{f}_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$.

[5] Hyvärinen'05, [6] Vincent'11

Score-Matching by Denoising (SMD)

Key points:

1. SMD interpretation yields $\nabla\rho(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{f}(\boldsymbol{x})$, thus explaining RED algs.
2. SMD interpretation holds for any $\widehat{p}_{\mathsf{x}}$, any denoiser class $\mathcal{F}$ (i.e., $\boldsymbol{f}_{\boldsymbol{\theta}}$ may be non-JS and/or non-LH), and any $\boldsymbol{\theta}$ (maybe not MMSE).
3. SMD arises naturally via non-parametric estimation and KDE. Matches construction of *learned* denoisers liked TNRD and DnCNN.

Related work:

Alain and Bengio[7] recently showed that learned auto-encoders can be explained by score-matching and *not* by minimization of an energy function.

[7] Alain/Bengio'14

Fast RED Algorithms

Until now we focused on how to *explain* the RED algorithms, which solve

$$\boxed{\text{RED fixed-point condition:} \quad \frac{1}{\sigma^2}\boldsymbol{A}^\top(\boldsymbol{A}\widehat{\boldsymbol{x}} - \boldsymbol{y}) + \lambda\big(\widehat{\boldsymbol{x}} - \boldsymbol{f}(\widehat{\boldsymbol{x}})\big) = \boldsymbol{0}}$$

We now focus on interpretation/design of fast RED algorithms.

In the RED paper, three algorithms were described:

1. Steepest-Descent
2. ADMM with I inner iters (to solve $\arg\min_{\boldsymbol{x}}\{\lambda\rho(\boldsymbol{x}) + \frac{\beta}{2}\|\boldsymbol{x} - \boldsymbol{r}_k\|^2\}$)
3. A "fixed-point" method (we show equivalence to proximal gradient alg[8])

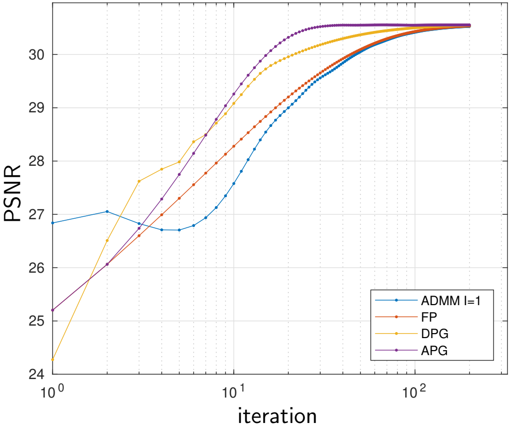
We propose a couple more. . .

[8] Combettes/Pesquet'11

Algorithm Comparison: Image Deblurring

New algorithms:

- DPG: "Dynamic" proximal gradient, which schedules the stepsize.
- APG: Accelerated proximal gradient, similar to FISTA.[9]



In this experiment, APG is about $3\times$ faster than the Fixed-Point method.

[9] Beck/Teboulle'09

Conclusions

- The RED algorithms work very well in practice.
- But they do not minimize $C_{\text{red}}(\boldsymbol{x}) = \ell(\boldsymbol{x};\boldsymbol{y}) + \lambda\rho_{\text{red}}(\boldsymbol{x})$ for many $\boldsymbol{f}(\cdot)$.
  - Why? Practical denoisers $\boldsymbol{f}(\cdot)$ are not sufficiently LH and JS.
  - Can construct examples of RED-SD *increasing* $C_{\text{red}}(\boldsymbol{x})$ over the iterations.
- We explained RED algorithms as "score-matching by denoising".
- We proposed new RED algorithms with faster convergence.

http://arxiv.org/abs/1806.02296