Lecture # 3: Plug & play methods

DEep Learning for Image REstoration and Synthesis

Andrés ALMANSA   Saïd LADJAL   Alasdair NEWSON

Cours de Master M2 MVA
http://delires.wp.imt.fr

February 2nd, 2021

1. Introduction
2. Bayesian & Variational Methods
3. Plug & Play methods
   - RED - Regularisation by Denoising
   - HQS - Half Quadratic Splitting
   - ADMM - Alternate Direction Method of Multipliers

Image acquisition model

- ideal image $u : \mathbb{R}^2 \to [0, \infty)$
- Measurements $\tilde{u} : \mathbb{Z}^2 \to [0, 255] \cap \mathbb{Z}$

$$\tilde{u} = g(\Delta_{\mathbb{Z}^2}((u \circ \phi) * h + n))$$

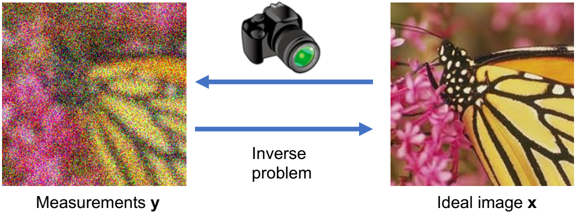
- Geometric deformation ϕ
- blur kernel h
- contrast change g

Simplified model

- $\mathbf{x} = \Delta_{\mathbb{Z}^2 \cap [0,N)^2}(u * \mathrm{sinc})$ (ideal discrete image $\mathbf{x} \in \mathbb{R}^{N^2}$)
- $\mathbf{y} = A(\mathbf{x}) + \mathbf{n}$ (degraded measurements)

Inverse Problems in Imaging

Estimate clean image $\mathbf{x} \in \mathbb{R}^N$
from noisy, degraded measurements $\mathbf{y} \in \mathbb{R}^m$.



Known degradation model (usually log-concave):

$$p_{Y|X}(\mathbf{y} \,|\, \mathbf{x}) \propto e^{-F(\mathbf{x},\mathbf{y})} \quad \text{where} \quad F(\mathbf{x},\mathbf{y}) = \frac{1}{2\sigma^2}\|A\mathbf{x} - \mathbf{y}\|^2. \qquad (1)$$

Inverse Problems in Imaging

Estimate clean image $\mathbf{x} \in \mathbb{R}^N$
from noisy, degraded measurements $\mathbf{y} \in \mathbb{R}^m$.
Known degradation model (usually log-concave):

$$p_{Y|X}(\mathbf{y} \mid \mathbf{x}) \propto e^{-F(\mathbf{x},\mathbf{y})} \quad \text{where} \quad F(\mathbf{x},\mathbf{y}) = \frac{1}{2\sigma^2}\|A\mathbf{x} - \mathbf{y}\|^2. \tag{1}$$

Variational/Bayesian Approach

Use image prior $p_X(\mathbf{x}) \propto e^{-\lambda R(\mathbf{x})}$ to compute estimator

$$\hat{\mathbf{x}}_{\text{MAP}} = \arg\max_{\mathbf{x}} p_{X|Y}(\mathbf{x} \mid \mathbf{y}) = \arg\min_{\mathbf{x}} \{F(\mathbf{x},\mathbf{y}) + \lambda R(\mathbf{x})\} \tag{2}$$

$$\hat{\mathbf{x}}_{\text{MMSE}} = \arg\min_{\mathbf{x}} \mathrm{E}\left[\|X - \mathbf{x}\|^2 \;\middle|\; Y = \mathbf{y} \right] \tag{3}$$

Bayesian Estimators

Simplified model

- $\mathbf{x} = \Delta_{\mathbb{Z}^2 \cap [0,\sqrt{N})^2}(u * \mathrm{sinc})$ (ideal discrete image $\mathbf{x} \in \mathbb{R}^N$)
- $\mathbf{y} = A(\mathbf{x}) + \mathbf{n}$ (degraded measurements)

Ill-posed $A \Rightarrow$ need for regularization (prior knowledge)

- risk minimization (MMSE): $\min_{\mathbf{x}} \mathrm{E}\left[\, \|X - \mathbf{x}\|^2 \,\middle|\, Y = \mathbf{y} \,\right]$
- posterior maximisation (MAP) $\max_{\mathbf{x}} \mathbb{P}[\, X = \mathbf{x} \mid Y = \mathbf{y} \,]$

Example of MAP estimation :

- Observation model:

$$\mathbb{P}[\, Y = \mathbf{y} \mid X = \mathbf{x} \,] = \mathbb{P}[N = (\mathbf{y} - A(\mathbf{x}))] = Ce^{-\frac{1}{2\sigma^2}\|\mathbf{y}-A(\mathbf{x})\|_2^2}$$

- Prior model

$$\mathbb{P}[X = \mathbf{x}] = C' e^{-\lambda R(\mathbf{x})}$$

- Posterior (using Bayes Theorem):

$$\mathbb{P}[\, X = \mathbf{x} \mid Y = \mathbf{y} \,] = \mathbb{P}[\, Y = \mathbf{y} \mid X = \mathbf{x} \,]\, \mathbb{P}[X = \mathbf{x}]$$

- log-Posterior

$$-\log \mathbb{P}[\, X = \mathbf{x} \mid Y = \mathbf{y} \,] = E(\mathbf{x}) = \underbrace{\frac{1}{2\sigma^2}\|\mathbf{y} - A(\mathbf{x})\|_2^2}_{F(\mathbf{x})} + \lambda R(\mathbf{x}) + C''$$

Bayesian Estimators

Simplified model

- $\mathbf{x} = \Delta_{\mathbb{Z}^2 \cap [0,\sqrt{N})^2}(u * \text{sinc})$ (ideal discrete image $\mathbf{x} \in \mathbb{R}^N$)
- $\mathbf{y} = A(\mathbf{x}) + \mathbf{n}$ (degraded measurements)

Ill-posed $A \Rightarrow$ need for regularization (prior knowledge)

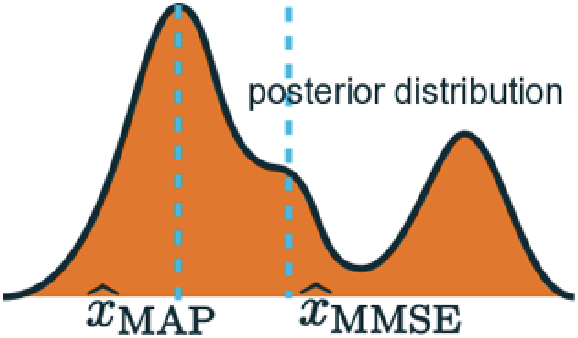
- risk minimization (MMSE): $\min_{\mathbf{x}} \mathrm{E}\left[\ \|X - \mathbf{x}\|^2 \ \middle|\ Y = \mathbf{y}\ \right]$
- posterior maximisation (MAP) $\max_{\mathbf{x}} \mathbb{P}\left[\ X = \mathbf{x} \mid Y = \mathbf{y}\ \right]$

Special case $\mathbf{n} \sim N(0, \sigma^2)$ and $\mathbf{x} \sim N(\mu, \Sigma)$ are Gaussian

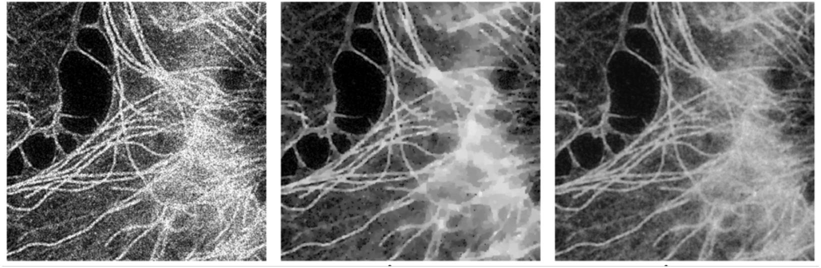
Both MMSE & MAP lead to the same closed form (Wiener filter):
$\hat{x} = (A^* A + \sigma^2 \Sigma^{-1})^{-1}(A^* \mathbf{y} + \sigma^2 \Sigma^{-1} \mu)$

WANING!! This is a very special case !!

Bayesian Estimators



Bayesian Estimators



Noisy image (Poisson noise) MAP TV-ICE

Choosing a prior for the clean image $\mathbf{x}$

- Tikhonov regularization: *(convex, smooth)*

$$R(\mathbf{x}) = \sum_i \|\nabla x_i\|_2^2$$

- Total Variation regularization: *(convex, non-smooth)*

$$R(\mathbf{x}) = \sum_i \|\nabla x_i\|_2$$

- Wavelet shrinkage: *(convex, non-smooth)*

$$R(\mathbf{x}) = \|W\mathbf{x}\|_1$$

- Patch based regularization (EPLL): *(non-convex, smooth)*

Choosing a prior for the clean image $\mathbf{x}$

- Tikhonov regularization: *(convex, smooth)*

$$R(\mathbf{x}) = \sum_i \|\nabla x_i\|_2^2$$

- Total Variation regularization: *(convex, non-smooth)*

$$R(\mathbf{x}) = \sum_i \|\nabla x_i\|_2$$

- Wavelet shrinkage: *(convex, non-smooth)*

$$R(\mathbf{x}) = \|W\mathbf{x}\|_1$$

- Patch based regularization (EPLL): *(non-convex, smooth)*
- Neural network to learn $R(\mathbf{x})$?

Convex Optimization Methods

We want to minimize $E(\mathbf{x}) = F(\mathbf{x}) + R(\mathbf{x}).$

Gradient Descent Method

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha \nabla E(\mathbf{x}^k)$$

Converges to stationary point $\mathbf{x}^*$
if E convex, ∇E is $L - Lipschitz$, and $\alpha \in [0, 2/L]$

FBS / ISTA / PGD

1. $\mathbf{v}^k = \mathbf{x}^k - \alpha \nabla R(\mathbf{x}^k)$
2. $\mathbf{x}^{k+1} = \operatorname{prox}_{\alpha F}(\mathbf{v}^k) = \arg\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{v}^k\|^2 + \alpha F(\mathbf{x})$

Converges to stationary point $\mathbf{x}^*$
if F, R convex, ∇R is $L - Lipschitz$, and $\alpha \in [0, 2/L]$

Other splitting methods: HQS, DRS, ADMM, Chambolle-Pock

Convex Optimization Methods

We want to minimize $E(\mathbf{x}) = F(\mathbf{x}) + R(\mathbf{x}).$

Gradient Descent Method

FBS / ISTA / PGD

ADMM

1. $\mathbf{v}^{k+1} = \mathrm{prox}_{\alpha R}(\alpha\overline{\mathbf{u}}^k - \mathbf{x}^k)$
2. $\mathbf{x}^{k+1} = \mathrm{prox}_{\alpha F}(\alpha\overline{\mathbf{u}}^k + \mathbf{v}^k)$
3. $\overline{\mathbf{u}}^{k+1} = \overline{\mathbf{u}}^k + \frac{1}{\alpha}(\mathbf{v}^{k+1} - \mathbf{x}^{k+1})$

Other splitting methods: HQS, DRS, ADMM, Chambolle-Pock

Neural Networks for inverse problems: Two paradigms

- Learning-based approach : find a sufficient number of image pairs $(\mathbf{x}^i, \mathbf{y}^i)$ and train a neural network f_θ to invert A by minimizing the empirical risk $\sum_i \|f_\theta(\mathbf{y}^i) - \mathbf{x}^i\|_2^2$
  - ✓ no need to model A, $\mathbf{n}$ nor prior for $\mathbf{x}$
  - ✗ needs retraining if A or $\mathbf{n}$ change
- Bayesian approach : Model separately
  1. conditional probability $\mathbb{P}[\, Y = \mathbf{y} \mid X = \mathbf{x}\,]$ (using physical model, calibration)
  2. prior model $\mathbb{P}[X = \mathbf{x}]$ (through NN learning)
  3. Use Bayes theorem to estimate $\mathbf{x}$ via MAP or MMSE

Using NNs to learn a prior for the clean image $\mathbf{x}$

- Neural network to learn $R(\mathbf{x})$?
  - ⚠ difficult to train !
  - ⚠ properties of R ? (we need to minimise $F(\mathbf{x}) + \lambda R(\mathbf{x})$!!)

Using NNs to learn a prior for the clean image $\mathbf{x}$

- Indirect (Plug & Play) approach :
  - Train a neural network to solve a simpler (denoising) problem:

$$D_\sigma : \mathbf{y} \mapsto \arg\min_{\mathbf{x}} \frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{y}\|_2^2 + R(\mathbf{x})$$

  - $D_\sigma = \text{prox}_{\sigma^2 R}$ is the proximal operator of the regularizer ... or $\nabla R(\mathbf{x}) = f(D_\sigma(\mathbf{x}))$
  - To regularize the original problem use a splitting scheme and
    - use D_σ instead of $\text{prox}_{\sigma^2 R}$ ... or
    - use $f(D_\sigma(\mathbf{x}))$ instead of $\nabla R(\mathbf{x})$

Plug & Play approach

A myriad of solutions have been proposed:

- RED - Regularization by Denoising (Reehorst and Schniter, 2019; Romano et al., 2017)
- HQS - Half Quadratic Splitting (Geman & Yang 2002; Zoran & Weiss 2012)
- ADMM - Alternated Direction Method of Multipliers (Boyd, 2010; Ryu et al., 2019)
- ISTA Proximal-gradient algorithm (Xu et al., 2020)

Questions

- Does the scheme converge?
- Does there exist a regularizer R such that $\text{prox}_{\sigma^2 R} = D_\sigma$?

RED - Regularisation by Denoising

RED constructs an explicit regularizer from a denoiser:

$$R_{RED}(\mathbf{x}) = \frac{1}{2}\langle \mathbf{x}, \mathbf{x} - D_\sigma(\mathbf{x})\rangle$$

Theorem (RED's gradient (Reehorst & Schnitter 2018))

If the denoiser D_σ :

- *is locally homogeneous (i.e. $(1+\varepsilon)D_\sigma(\mathbf{x}) = D_\sigma((1+\varepsilon)\mathbf{x})$) and*
- *has symmetric Jacobian*

then R_{RED}'s gradient is writes $\nabla R_{RED}(\mathbf{x}) = \mathbf{x} - D_\sigma(\mathbf{x})$

RED - Regularisation by Denoising

$$R_{RED}(\mathbf{x}) = \frac{1}{2}\langle \mathbf{x}, \mathbf{x} - D_\sigma(\mathbf{x})\rangle$$

Theorem (RED's gradient (Reehorst & Schnitter 2018)**)**

If the denoiser D_σ :

- *is locally homogeneous (i.e. $(1+\varepsilon)D_\sigma(\mathbf{x}) = D_\sigma((1+\varepsilon)\mathbf{x})$) and*
- *has symmetric Jacobian*

then R_{RED}'s gradient is writes $\nabla R_{RED}(\mathbf{x}) = \mathbf{x} - D_\sigma(\mathbf{x})$

Collorary A gradient descent scheme (without splitting) is easy to implement:

$$\nabla E(\mathbf{x}) = (\nabla F + Id - D_\sigma)(\mathbf{x})$$

RED - Regularisation by Denoising

$$R_{RED}(\mathbf{x}) = \frac{1}{2}\langle \mathbf{x}, \mathbf{x} - D_\sigma(\mathbf{x})\rangle$$

Theorem (RED's gradient (REEHORST & SCHNITTER 2018))

If the denoiser D_σ :

- *is locally homogeneous (i.e. $(1+\varepsilon)D_\sigma(\mathbf{x}) = D_\sigma((1+\varepsilon)\mathbf{x})$) and*
- *has symmetric Jacobian*

then R_{RED}'s gradient is writes $\nabla R_{RED}(\mathbf{x}) = \mathbf{x} - D_\sigma(\mathbf{x})$

Lemma 1: If the denoiser D is locally homogeneous then $[J_D(\mathbf{x})]\mathbf{x} = D(\mathbf{x})$

Lemma 2: R_{RED}'s gradient is $\nabla R(\mathbf{x}) = \mathbf{x} - 0.5 D_\sigma(\mathbf{x}) - 0.5[J_D(\mathbf{x})]^T\mathbf{x}$

Lemma 3: If the denoiser D is locally homogeneous and has symmetric Jacobian then $\nabla R_{RED}(\mathbf{x}) = \mathbf{x} - D(\mathbf{x})$

RED - Regularisation by Denoising

$$R_{RED}(\mathbf{x}) = \frac{1}{2}\langle \mathbf{x}, \mathbf{x} - D_\sigma(\mathbf{x})\rangle$$

Theorem (RED's gradient (Reehorst & Schnitter 2018)**)**

If the denoiser D_σ :

- *is locally homogeneous (i.e. $(1+\varepsilon)D_\sigma(\mathbf{x}) = D_\sigma((1+\varepsilon)\mathbf{x})$) and*
- *has symmetric Jacobian*

then R_{RED}'s gradient is writes $\nabla R_{RED}(\mathbf{x}) = \mathbf{x} - D_\sigma(\mathbf{x})$

Reality check

$$e_f^J = \frac{\|J_f - J_f^T\|_F^2}{\|J_f\|_F^2}$$

	TDT	MF	NLM	BM3D	TNRD	DnCNN
$e_f^J(\boldsymbol{x})$	5.36e-21	1.50	0.250	1.22	0.0378	0.0172

TABLE I
AVERAGE JACOBIAN-SYMMETRY ERROR ON 16×16 IMAGES

Alternative to RED: Tweedie's formula (Efron 2011)

Tweedie's formula

If $X \sim p_X$, $N \sim N(0, \sigma^2 Id)$ and $D_\sigma(\mathbf{y}) = \mathrm{E}[\, X \mid X + N = \mathbf{y}\,]$ then

$$(D_\sigma - Id)(\mathbf{x}) = \sigma^2 \nabla \log(p_X * g_\sigma)(\mathbf{x})$$

Using the smooth prior $R_\sigma(\mathbf{x}) = -\log(p_X * g_\sigma)$
We want to minimize:

$$E(\mathbf{x}) = F(\mathbf{x}) + \lambda R_\sigma(\mathbf{x})$$

And the gradient writes exactly:

$$\nabla E(\mathbf{x}) = (\nabla F + \frac{\lambda}{\sigma^2}(Id - D_\sigma))(\mathbf{x})$$

This justifies the RED algorithm (for $\frac{\lambda}{\sigma^2} = 1$) for any denoiser that is computed as an MMSE. For $\frac{\lambda}{\sigma^2} \neq 1$ the above scaling is required.

Tweedie's formula and posterior sampling (Guo et al., 2019)

Tweedie's formula

If $X \sim p_X$, $N \sim N(0, \sigma^2 Id)$ and $D_\sigma(\mathbf{y}) = \mathrm{E}\left[\, X \mid X + N = \mathbf{y} \,\right]$ then

$$(D_\sigma - Id)(\mathbf{x}) = \sigma^2 \nabla \log(p_X * g_\sigma)(\mathbf{x})$$

Instead of maximizing $\pi(\mathbf{x}) = p_{X|Y}(\mathbf{x} \,|\, \mathbf{y})$
We can try to take samples $X_k \sim \pi$
using the Langevin algorithm

$$X_{k+1} = X_k + \delta \nabla \log \pi(X_k) + \sqrt{2\delta} Z_k \quad \text{with } Z_k \sim N(0, Id)$$

Using Tweedie's formula we can write

$$\nabla \log \pi(\mathbf{x}) = (-\nabla F + \frac{1}{\sigma^2}(D_\sigma - Id))(\mathbf{x})$$

HQS - Half Quadratic Splitting

Instead of minimizing the original problem

$$E(\mathbf{x}) = F(\mathbf{x}) + \lambda R(\mathbf{x})$$

we introduce an auxiliary variable $\mathbf{v}$ and we aim at solving the equivalent *constrained minimization problem* :

$$\min_{\mathbf{x},\mathbf{v}} F(\mathbf{x}) + \lambda R(\mathbf{v}) \quad \text{under the constraint } \mathbf{x} = \mathbf{v}$$

The constraint can be added back to the energy (preceded by a Lagrange multiplier β)

$$E_1(\mathbf{x}, \mathbf{v}, \beta) = F(\mathbf{x}) + \beta\|\mathbf{x} - \mathbf{v}\|^2 + \lambda R(\mathbf{v})$$

We know that for β large enough joint minimization of E_1 is equivalent to minimizing E.

HQS - Half Quadratic Splitting

$$E_1(\mathbf{x}, \mathbf{v}, \beta) = F(\mathbf{x}) + \beta\|\mathbf{x} - \mathbf{v}\|^2 + \lambda R(\mathbf{v})$$

To ensure that β is large enough we can use a *continuation scheme*:

$$\arg\min_{\mathbf{x}} E(\mathbf{x}) = \lim_{\beta\to\infty} \arg\min_{\mathbf{x}} \min_{\mathbf{v}} E_1(\mathbf{x}, \mathbf{v}, \beta)$$

Which inspires this alternating a minimization algorithm:

1. Initialization: $\mathbf{x}^0, \mathbf{v}^0, \beta^0 > 0, \gamma > 0$
2. For k=1, ... until convergence
3. $\mathbf{x}^k = \arg\min_{\mathbf{x}} E_1(\mathbf{x}, \mathbf{v}^{k-1}, \beta_{k-1})$. // inverse problem
4. $\mathbf{v}^k = \arg\min_{\mathbf{v}} E_1(\mathbf{x}^k, \mathbf{v}, \beta_{k-1})$. // regularization
5. $\beta_k = \gamma\beta^{k-1}$
6. End for

This method (which was proposed by Geman & Yang (2002)) was notably used by Zoran & Weiss (2012) to optimize their EPLL method.

HQS Plug & Play

The fourth step in the HQS method ...

4. $\mathbf{v}^k = \arg\min_{\mathbf{v}} E_1(\mathbf{x}^k, \mathbf{v}) = \arg\min_{\mathbf{v}} \beta_k \|\mathbf{x}^k - \mathbf{v}\|^2 + R(\mathbf{v})$

... can be interpreted as a **denoising** of $\mathbf{x}^k$ with noise variance $\sigma_k^2 = 1/(2\beta^k)$

Replace step 4 by a trained denoiser D_{σ_k}:

1. Initialization: $\mathbf{x}^0, \mathbf{v}^0, \beta^0$
2. For k=1, ... until convergence
3. $\mathbf{x}^k = \arg\min_{\mathbf{x}} E_1(\mathbf{x}, \mathbf{v}^{k-1})$. (inverse problem)
4. $\mathbf{v}^k = D_{\sigma_k}(\mathbf{x}^k)$. (regularization)
5. $\beta_k = \gamma\beta^{k-1}$
6. End for

Obs 1: In this scheme the denoiser needs to be trained for several values of σ_k

Obs 2: The convergence of the P&P HQS method has not been established.

We refer to it here as an historical introduction.

Method of Multipliers

The HQS method requires β to increase during the iterations.

A more proper method which can ensure convergence with a fixed parameter is the *method of multipliers*, which introduces a Lagrange multiplier w_i for each constraint $\mathbf{x}_i = \mathbf{v}_i$:

$$E_2(\mathbf{x}, \mathbf{v}, \mathbf{u}) = F(\mathbf{x}) + \mathbf{u}^T(\mathbf{x} - \mathbf{v}) + \lambda R(\mathbf{v})$$

We can show that

$$\arg\min_{\mathbf{x}} \min_{\mathbf{v}} \max_{\mathbf{u}} E_2(\mathbf{x}, \mathbf{v}, \mathbf{u}) = \arg\min_{\mathbf{x}} E(\mathbf{x})$$

This results in the following algorithm

1. Initialization: $\mathbf{x}^0, \mathbf{v}^0, \mathbf{u}^0, \alpha > 0,$
2. For k=1, ... until convergence
3. $\mathbf{x}^k = \arg\min_{\mathbf{x}} E_2(\mathbf{x}, \mathbf{v}^{k-1}, \mathbf{u}^{k-1}).$ // inverse problem
4. $\mathbf{v}^k = \arg\min_{\mathbf{v}} E_2(\mathbf{x}^k, \mathbf{v}, \mathbf{u}^{k-1}).$ // regularization
5. $\mathbf{u}^k = \mathbf{u}^{k-1} + \alpha(\mathbf{x}^k - \mathbf{v}^k)$ // gradient ascent on multiplier
6. End for

ADMM - unscaled version

A more flexible version of this algorithm can be constructed, which:

- has milder conditions on D and R
- converges faster

It is obtained by adding the ℓ^2 norm of the constraint to build the augmented Lagrangian:

$$E_3(\mathbf{x}, \mathbf{v}, \mathbf{u}) = F(\mathbf{x}) + \mathbf{u}^T(\mathbf{x} - \mathbf{v}) + \lambda R(\mathbf{v}) + \frac{\rho}{2}\|\mathbf{x} - \mathbf{v}\|^2$$

The corrresponding alternated optimization (where $\alpha = \rho$ is the *unscaled version of ADMM*:

1. Initialization: $\mathbf{x}^0, \mathbf{v}^0, \mathbf{u}^0, \rho > 0,$
2. For k=1, ... until convergence
3. $\mathbf{x}^k = \arg\min_{\mathbf{x}} E_3(\mathbf{x}, \mathbf{v}^{k-1}, \mathbf{u}^{k-1}).$ // inverse problem
4. $\mathbf{v}^k = \arg\min_{\mathbf{v}} E_3(\mathbf{x}^k, \mathbf{v}, \mathbf{u}^{k-1}).$ // regularization
5. $\mathbf{u}^k = \mathbf{u}^{k-1} + \rho(\mathbf{x}^k - \mathbf{v}^k)$ // gradient ascent on multiplier
6. End for

ADMM - scaled version

ADMM is often written in a more convenient way as follows:
Defining the residual of the constraint as $r = \mathbf{x} - \mathbf{v}$
the "constraint" part of the Lagrangian can be rewritten

$$\mathbf{u}^T r + \frac{\rho}{2}\|r\|^2 = \frac{\rho}{2}\|r + \frac{1}{\rho}\mathbf{u}\|^2 - \frac{1}{2\rho}\|\mathbf{u}\|^2 = \frac{\rho}{2}\|r + \overline{\mathbf{u}}\|^2 - \frac{\rho}{2}\|\overline{\mathbf{u}}\|^2$$

where $\overline{\mathbf{u}} := \frac{1}{\rho}\mathbf{u}$ is the scaled multiplier
With this modification the *scaled ADMM* that results from minimizing

$$E_4(\mathbf{x}, \mathbf{v}, \overline{\mathbf{u}}) = F(\mathbf{x}) + \rho\overline{\mathbf{u}}^T(\mathbf{x} - \mathbf{v}) + \lambda R(\mathbf{v}) + \frac{\rho}{2}\|\mathbf{x} - \mathbf{v}\|^2$$

and the algorithm becomes:

1. Initialization: $\mathbf{x}^0, \mathbf{v}^0, \overline{\mathbf{u}}^0, \rho > 0$,
2. For k=1, ... until convergence
3. $\mathbf{x}^k = \arg\min_{\mathbf{x}} E_4(\mathbf{x}, \mathbf{v}^{k-1}, \overline{\mathbf{u}}^{k-1})$. // inverse problem
4. $\mathbf{v}^k = \arg\min_{\mathbf{v}} E_4(\mathbf{x}^k, \mathbf{v}, \overline{\mathbf{u}}^{k-1})$. // regularization
5. $\overline{\mathbf{u}}^k = \overline{\mathbf{u}}^{k-1} + (\mathbf{x}^k - \mathbf{v}^k)$. // gradient ascent on multiplier
6. End for

ADMM - scaled version

ADMM is often written in a more convenient way as follows:
Defining the residual of the constraint as $r = \mathbf{x} - \mathbf{v}$
the "constraint" part of the Lagrangian can be rewritten

$$\mathbf{u}^T r + \frac{\rho}{2}\|r\|^2 = \frac{\rho}{2}\|r + \frac{1}{\rho}\mathbf{u}\|^2 - \frac{1}{2\rho}\|\mathbf{u}\|^2 = \frac{\rho}{2}\|r + \overline{\mathbf{u}}\|^2 - \frac{\rho}{2}\|\overline{\mathbf{u}}\|^2$$

where $\overline{\mathbf{u}} := \frac{1}{\rho}\mathbf{u}$ is the scaled multiplier
With this modification the *scaled ADMM* that results from minimizing

$$E_4(\mathbf{x}, \mathbf{v}, \overline{\mathbf{u}}) = F(\mathbf{x}) + \lambda R(\mathbf{v}) + \frac{\rho}{2}\|\mathbf{x} - \mathbf{v} + \overline{\mathbf{u}}\|^2 - \frac{\rho}{2}\|\overline{\mathbf{u}}\|^2$$

or expanding the two minimization steps:

1. Initialization: $\mathbf{x}^0, \mathbf{v}^0, \overline{\mathbf{u}}^0, \rho > 0$,
2. For k=1, ... until convergence
3. $\mathbf{x}^k = \arg\min_{\mathbf{x}} F(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{x} - (\mathbf{v}^{k-1} - \overline{\mathbf{u}}^{k-1})\|^2$. // prox on inverse problem
4. $\mathbf{v}^k = \arg\min_{\mathbf{v}} \lambda R(\mathbf{v}) + \frac{\rho}{2}\|(\mathbf{x}^k + \overline{\mathbf{u}}^{k-1}) - \mathbf{v}\|^2$. // prox on regularization
5. $\overline{\mathbf{u}}^k = \overline{\mathbf{u}}^{k-1} + (\mathbf{x}^k - \mathbf{v}^k)$. // gradient ascent on multiplier
6. End for

Plug & Play ADMM

As in the HQS case, step 4 in this algorithm is substituted by a NN-based denoiser that was trained independently of this problem.
Instead of:

4. $\mathbf{v}^k = \arg\min_{\mathbf{v}} \lambda R(\mathbf{v}) + \frac{\rho}{2}\|(\mathbf{x}^k + \overline{\mathbf{u}}^{k-1}) - \mathbf{v}\|^2.$

we write:

4. $\mathbf{v}^k = D_\sigma(\mathbf{x}^k + \overline{\mathbf{u}}^{k-1}).$

where $\sigma = \sqrt{\lambda/\rho}$

Theorem (Plug & Play ADMM (Sreehari et al., 2016))

If D_σ is differentiable and its Jacobian J_{D_σ} is **symmetric** *with* **eigenvalues in** $[0, 1]$, *plus some mild technical conditions, then:*

- *D_σ is the proximal operator of some energy function R.*
- *Plug & Play ADMM converges to the global infimum of $F(\mathbf{x}) + R(\mathbf{x})$*

Plug & Play ADMM

Theorem (Plug & Play ADMM (Sreehari et al., 2016))

If D_σ is differentiable and its Jacobian J_{D_σ} is **symmetric** *with* **eigenvalues in** $[0,1]$, *plus some mild technical conditions, then:*

- *D_σ is the proximal operator of some energy function R.*
- *Plug & Play ADMM converges to the global infimum of $F(\mathbf{x}) + R(\mathbf{x})$*

The proof of this result is based on a result of J.J. Moreau (1965):

Theorem ((J.J. Moreau, 1965))

A denoiser D is the proximal operator of an energy R, iff

- *D is non-expansive $\|D(\mathbf{x}) - D(\mathbf{v})\|_2 \leq \|\mathbf{x} - \mathbf{v}\|_2$, and*
- *there exists φ such that $D(\mathbf{x}) \in \partial\varphi(\mathbf{x})$*

Non-expansiveness results from the eigenvalue condition. And the existence of φ from the symmetric Jacobian and Green's theorem.

Plug & Play ADMM (bounded denoisers) (Chan et al., 2017)

A slight variation of the P&P-ADMM algorithm allows it to converge under more reasonable conditions.
The modification is similar to the continuation scheme used for HQS:

1. Initialization: $\mathbf{x}^0, \mathbf{v}^0, \overline{\mathbf{u}}^0, \rho^0 > 0, \eta < 1, \gamma > 1$
2. For k=1, ... until convergence
3. $\mathbf{x}^k = \arg\min_{\mathbf{x}} F(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{x} - (\mathbf{v}^{k-1} - \overline{\mathbf{u}}^{k-1})\|^2.$
   // proximal descent on inverse problem
4. $\mathbf{v}^k = D_{\sigma^k}(\mathbf{x}^k + \overline{\mathbf{u}}^{k-1})$ where $\sigma^k = \sqrt{\lambda/\rho^{k-1}}$
5. $\overline{\mathbf{u}}^k = \overline{\mathbf{u}}^{k-1} + (\mathbf{x}^k - \mathbf{v}^k)$. // gradient ascent on multiplier
6. If $\Delta_k \geq \eta\Delta_{k-1}$ then $\rho^k = \gamma\rho^{k-1}$
7. Else $\rho^k = \rho^{k-1}$
8. End if
9. End for

Plug & Play ADMM convergence for bounded denoisers

Theorem ((Chan et al., 2017))

- *H1: If F has bounded gradients, and*
- *H2: D_σ is bounded (there exists $C > 0$ such that for any image $\mathbf{x} \in \mathbb{R}^N$ $\frac{1}{N}\|D_\sigma(\mathbf{x}) - \mathbf{x}\|^2 \leq \sigma^2 C$)*

Then the iterates $\theta^k = (\mathbf{x}^k, \mathbf{v}^k, \overline{\mathbf{u}}^k)$ of Plug & Play ADMM converge to a fixed point in ℓ^2 norm.

The convergence proof proceeds by showing that $\theta^k = (\mathbf{x}^k, \mathbf{v}^k, \overline{\mathbf{u}}^k)$ is a Cauchy sequence by showing that
$D(\theta^{k+1}, \theta^k) \leq C\delta^k$
for some $C > 0$ and $\delta \in (0, 1)$
This is trivial in case 1 (else case when $\Delta_k < \eta\Delta_{k-1}$).
Otherwise the update rule $\rho_{k+1} = \gamma\rho_k$ and boundedness of the denoiser play an important role:
Indeed $\sigma^k < \gamma^{-1/2}\sigma^{k-1}$, and therefore $\|D_{\sigma^k}(\mathbf{x}^k + \overline{\mathbf{u}}^{k-1}) - (\mathbf{x}^k + \overline{\mathbf{u}}^{k-1}))\|$ also decreases because it is bounded.

PnP-ADMM - Non-expansive residual (Ryu et al., 2019)

Previous assumptions:

- D_σ contractive => too strong
- $2D_\sigma - I$ contractive => too strong

Assumption A (ε-Lipschitz residual):

$$\|(D_\sigma - I)(x) - (D_\sigma - I)(y)\|^2 \leq \varepsilon^2 \|x - y\|^2$$

Obs: D_σ can be trained to satisfy (A), via Residual Networks and Spectral Normalization.

PnP-ADMM - Non-expansive residual (Ryu et al., 2019)

Assumption A (ε-Lipschitz residual):

$$\|(D_\sigma - I)(x) - (D_\sigma - I)(y)\|^2 \leq \varepsilon^2 \|x - y\|^2$$

Idée de la preuve:
PnP ADMM has the same fixed points as PnP-DRS:

PnP-ADMM:

1. $\mathbf{x}^k = \mathrm{prox}_{F/\rho}(\mathbf{v}^{k-1} - \overline{\mathbf{u}}^{k-1})$.
2. $\mathbf{v}^k = D_\sigma(\mathbf{x}^k + \overline{\mathbf{u}}^{k-1})$.
3. $\overline{\mathbf{u}}^k = \overline{\mathbf{u}}^{k-1} + (\mathbf{x}^k - \mathbf{v}^k)$.

PnP-DRS

1. $\mathbf{v}^k = \mathrm{prox}_{F/\rho}(\overline{\mathbf{u}}^{k-1})$.
2. $\mathbf{x}^k = D_\sigma(2\mathbf{v}^k - \overline{\mathbf{u}}^{k-1})$.
3. $\overline{\mathbf{u}}^k = \overline{\mathbf{u}}^{k-1} + (\mathbf{x}^k - \mathbf{v}^k)$.
multiplier

PnP-ADMM - Non-expansive residual (Ryu et al., 2019)

Assumption A (ε-Lipschitz residual):

$$\|(D_\sigma - I)(x) - (D_\sigma - I)(y)\|^2 \leq \varepsilon^2 \|x - y\|^2$$

Idée de la preuve:
PnP ADMM has the same fixed points as PnP-DRS:
$\overline{\mathbf{u}}^k = T(\overline{\mathbf{u}}^{k-1})$ avec $T = \frac{1}{2} I - \frac{1}{2}(2D_\sigma - I)(2\,\mathrm{prox}_{F/\rho} - I)$
Objective: Show that T is non-expansive => convergent sequence.
If:

- Assumption A holds
- F is μ-strongly convex
- $\varepsilon < 1$
- $\rho < \frac{\mu(1+\varepsilon-2\varepsilon^2)}{\varepsilon}$

Then T is contractive and both PnP-DRS and PnP-ADMM converge.

PnP-ISTA (Xu et al., 2020)

If D_σ is an MMSE denoiser with a non-degenerate prior p_X
and ∇F is L-Lipschitz
then the PnP-ISTA scheme with $\alpha \in [0, 1/L]$ converges to a stationary point of $F + R^*$
where $D_\sigma = \mathrm{prox}_{\alpha R^*}$

PnP-ISTA

1. $\mathbf{v}^k = \mathbf{x}^k - \alpha \nabla F(\mathbf{x}^k)$
2. $\mathbf{x}^{k+1} = \mathrm{prox}_{\alpha R^*}(\mathbf{v}^k) = D_\sigma(\mathbf{v}^k)$

No contractivity condition on D_σ
No strongly convex condition on F
Proof: Based on (Gribonval, 2011)

Conclusion

- RED requires either :
  - locally homogeneous denoiser with symmetric Jacobian (Reehorst and Schniter, 2019), OR
  - careful parameter choice and denoiser with non-expansive residual (Tweedie interpretation)
- Plug & Play ADMM (adaptive ρ) requires bounded denoiser
  - difficult to achieve for small σ.
  - Convergence is forced by decreasing ρ artificially
- Plug & Play ADMM (fixed ρ) requires either
  - non-expansive denoiser (Wavelet Thresholding, symmetrized NLM (Sreehari et al., 2016)), or
  - denoiser with non-expansive residual ((Ryu et al., 2019) retrains denoiser to satisfy this condition) and strongly convex data-fitting.
- Plug & Play ISTA (Xu et al., 2020) converges for ANY MMSE denoiser.

Reading Guide

The Plug & Play methods we shall experiment with in the lab session are discussed here:

- PnP ADMM (Ryu et al., 2019)
- PnP ISTA (Xu et al., 2020)

A critical review of the RED algorithm is provided by (Reehorst and Schniter, 2019), and its Tweedie interpretation shall appear soon.
For a review of splitting methods in convex optimization see Emilie Chouzenoux's course or (Parikh and Boyd, 2014).
For a more in-depth review of the theory of monotone operators behind the proofs of PnP ADMM and PnP ISTA see (Ryu and Boyd, 2016) or the more comprehensive monograph (Bauschke and Combettes, 2017).

The bibliography below provides the doi link (official version). If you do not have access to the original (via your university library's online subscriptions) you can download the arXiv or HAL preprint or the PDF link to the author's page.

Bauschke, Heinz H. and Patrick L. Combettes (2017). *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer International Publishing. ISBN: 978-3-319-48310-8. DOI: 10.1007/978-3-319-48311-5 (cit. on p. 41).

Boyd, Stephen (2010). "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". In: *Foundations and Trends® in Machine Learning* 3.1, pp. 1–122. ISSN: 1935-8237. DOI: 10.1561/2200000016. URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.360.1664&rep=rep1&type=pdf (cit. on p. 18).

Chambolle, Antonin and Thomas Pock (2016). "An introduction to continuous optimization for imaging". In: *Acta Numerica* 25, pp. 161–319. ISSN: 0962-4929. DOI: 10.1017/S096249291600009X.

Chan, Stanley H., Xiran Wang, and Omar A. Elgendy (2017). "Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications". In: *IEEE Transactions on Computational Imaging* 3.1, pp. 84–98. ISSN: 2333-9403. DOI: 10.1109/TCI.2016.2629286. arXiv: 1605.01710 (cit. on pp. 34, 35).

Efron, Bradley (2011). "Tweedie's Formula and Selection Bias". In: *Journal of the American Statistical Association* 106.496, pp. 1602–1614. ISSN: 0162-1459. DOI: 10.1198/jasa.2011.tm11181.

González, Mario, Andrés Almansa, Mauricio Delbracio, Pablo Musé, and Pauline Tan (2019). "Solving Inverse Problems by Joint Posterior Maximization with a VAE Prior". In: arXiv: 1911.06379. URL: http://arxiv.org/abs/1911.06379.

Gribonval, Rémi (2011). "Should Penalized Least Squares Regression be Interpreted as Maximum A Posteriori Estimation?" In: *IEEE Transactions on Signal Processing* 59.5, pp. 2405–2410. ISSN: 1053-587X. DOI: 10.1109/TSP.2011.2107908. URL: https://hal.inria.fr/inria-00486840 (cit. on p. 39).

Guo, Bichuan, Yuxing Han, and Jiangtao Wen (2019). "AGEM : Solving Linear Inverse Problems via Deep Priors and Sampling". In: *(NeurIPS) Advances in Neural Information Processing Systems*. Ed. by Curran Associates Inc., pp. 545–556. URL: http://papers.nips.cc/paper/8345-agem-solving-linear-inverse-problems-via-deep-priors-and-sampling (cit. on p. 24).

J.J. Moreau (1965). "Proximité et dualité dans un espace hilbertien". In: *Bulletin de la S.M.F.* 93.3, pp. 273–299. ISSN: 02782626. URL: https://hal.archives-ouvertes.fr/hal-01740635 (cit. on p. 33).

Meinhardt, Tim, Michael Moeller, Caner Hazirbas, and Daniel Cremers (2017). "Learning Proximal Operators: Using Denoising Networks for Regularizing Inverse Imaging Problems". In: *(ICCV) International Conference on Computer Vision*, pp. 1781–1790. arXiv: 1704.03488.

Parikh, Neal and Stephen Boyd (2014). "Proximal Algorithms". In: *Foundations and Trends® in Optimization* 1.3, pp. 127–239. ISSN: 2167-3888. DOI:

10.1561/2400000003. URL: https://pdfs.semanticscholar.org/e1e8/be9ce5a7c0c4b41b384bbb84367fa3b6122c.pdf (cit. on p. 41).

Reehorst, Edward T. and Philip Schniter (2019). "Regularization by Denoising: Clarifications and New Interpretations". In: *IEEE Transactions on Computational Imaging* 5.1, pp. 52–67. ISSN: 2333-9403. DOI: 10.1109/TCI.2018.2880326. arXiv: 1806.02296 (cit. on pp. 18, 40, 41).

Romano, Yaniv, Michael Elad, and Peyman Milanfar (2017). "The Little Engine That Could: Regularization by Denoising (RED)". In: *SIAM Journal on Imaging Sciences* 10.4, pp. 1804–1844. ISSN: 1936-4954. DOI: 10.1137/16M1102884. arXiv: 1611.02862 (cit. on p. 18).

Ryu, Ernest K and Stephen Boyd (2016). "a Primer on Monotone Operator Methods Survey". In: *Appl. Comput. Math* 1, pp. 3–43. URL: https://web.stanford.edu/~boyd/papers/pdf/monotone_primer.pdf (cit. on p. 41).

Ryu, Ernest K., Jialin Liu, Sicheng Wang, Xiaohan Chen, Zhangyang Wang, and Wotao Yin (2019). "Plug-and-Play Methods Provably Converge with Properly Trained Denoisers". In: *ICML*. arXiv: 1905.05406 (cit. on pp. 18, 36–38, 40, 41).

Sreehari, Suhas, Singanallur V. Venkatakrishnan, Brendt Wohlberg, Gregery T. Buzzard, Lawrence F. Drummy, Jeffrey P. Simmons, and Charles A. Bouman (2016). "Plug-and-Play Priors for Bright Field Electron Tomography and Sparse Interpolation". In: *IEEE Transactions on Computational*

Imaging 2.4, pp. 1–1. ISSN: 2333-9403. DOI: 10.1109/TCI.2016.2599778. arXiv: 1512.07331 (cit. on pp. 32, 33, 40).

Xu, Xiaojian, Yu Sun, Jiaming Liu, Brendt Wohlberg, and Ulugbek S. Kamilov (2020). "Provable Convergence of Plug-and-Play Priors with MMSE denoisers". In: 4, pp. 1–10. arXiv: 2005.07685 (cit. on pp. 18, 39–41).

Yan, Ming and Wotao Yin (2016). "Self Equivalence of the Alternating Direction Method of Multipliers". In: pp. 165–194. DOI: 10.1007/978-3-319-41589-5_5. arXiv: 1407.7400.

Zoran, Daniel and Yair Weiss (2011). "From learning models of natural image patches to whole image restoration". In: *(ICCV) International Conference on Computer Vision*. IEEE, pp. 479–486. ISBN: 978-1-4577-1102-2. DOI: 10.1109/ICCV.2011.6126278. URL: http://people.csail.mit.edu/danielzoran/EPLLICCVCameraReady.pdf.