去噪去模糊方法的数学原理

主要参考：Stanford EE367

Image Deconvolution with the Half-quadratic Splitting (HQS) Method

EE367/CS448I: Computational Imaging
stanford.edu/class/ee367
Lecture 10

Gordon Wetzstein
Stanford University

Image Deconvolution – Brief Review

- Image formation model: $b = c * x + \eta$

2D measurements

known 2D convolution kernel

2D target image

additive noise

Image Deconvolution – Brief Review

- Image formation model: $b = c * x + \eta$
- Convolution theorem: $b = \mathcal{F}^{-1}\{\mathcal{F}\{c\} \cdot \mathcal{F}\{x\}\} + \eta$
- Inverse filtering: $\tilde{x}_{\text{if}} = \mathcal{F}^{-1}\left\{\dfrac{\mathcal{F}\{b\}}{\mathcal{F}\{c\}}\right\}$
- Wiener filtering: $\tilde{x}_{\text{wf}} = \mathcal{F}^{-1}\left\{\dfrac{|\mathcal{F}\{c\}|^2}{|\mathcal{F}\{c\}|^2 + {}^{1}/_{SNR}} \cdot \dfrac{\mathcal{F}\{b\}}{\mathcal{F}\{c\}}\right\}$
- Duality of “signal processing” and “algebraic” interpretation:

$$b = c * x \;\Leftrightarrow\; \boldsymbol{b} = \boldsymbol{C}\boldsymbol{x} \qquad \boldsymbol{C} \in \mathbb{R}^{N\times N}, \quad \boldsymbol{b}, \boldsymbol{x} \in \mathbb{R}^{N}$$

基本的符号

A Bayesian Perspective of Inverse Problems

- Image formation model: $\boldsymbol{b} = \boldsymbol{Ax} + \boldsymbol{\eta}, \quad \boldsymbol{b} \in \mathbb{R}^M, \boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{A} \in \mathbb{R}^{M \times N}$
- Interpret as random variables: $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{x}_i, 0),\ \boldsymbol{\eta}_i \sim \mathcal{N}(0, \sigma^2)$
  $\boldsymbol{b}_i \sim \mathcal{N}\left((\boldsymbol{Ax})_i, \sigma^2\right)$
- Probability of observation i: $p(\boldsymbol{b}_i | \boldsymbol{x}_i, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\boldsymbol{b}_i - (\boldsymbol{Ax})_i)^2}{2\sigma^2}}$
- Joint probability of all observations: $\boxed{p(\boldsymbol{b}|\boldsymbol{x}, \sigma) = \prod_{i=1}^{M} p(\boldsymbol{b}_i | \boldsymbol{x}_i, \sigma) \propto e^{\overbrace{-\frac{\|\boldsymbol{b} - \boldsymbol{Ax}\|_2^2}{2\sigma^2}}^{m}}}$

因为是正比，所以可以得出

$$\mathrm{x} = \mathrm{argmax}\ \mathrm{P(x|b)} = \mathrm{argmax} \frac{\mathrm{P(b|x)} * \mathrm{P(x)}}{\mathrm{P(b)}}$$

$$\mathrm{x} = \mathrm{argmax}\ \mathrm{P(b|x)} * \mathrm{P(x)}$$

$$\mathrm{x} = \mathrm{argmax}\ \log\left(\mathrm{P(b|x)}\right) + \boxed{\log\left(\mathrm{P(x)}\right)}$$

$$\mathrm{x} = \mathrm{argmax}[\log(\mathrm{e}^{\mathrm{m}})] \boxed{- \Psi(\mathrm{x})}$$

image prior

$$\mathrm{x} = \mathrm{argmin}(-\mathrm{m}) + \Psi(\mathrm{x})$$

$$\boldsymbol{x}_{MAP} = \arg\min_{\boldsymbol{x}} \underbrace{\frac{1}{2\sigma^2} \|\boldsymbol{b} - \boldsymbol{Ax}\|_2^2}_{\text{data fidelity term}} + \underbrace{\Psi(\boldsymbol{x})}_{\text{regularization term}}$$

这里就是讲：如果当成贝叶斯来看待这个问题
论证了**我们为什么要image prior**

Examples of Image Priors / Regularizers

blurry stuff	stars	“natural” image
		
Promote smoothness!	Promote sparsity!	Promote sparse gradients!
$\Psi(x) = \|\Delta x\|_2$	$\Psi(x) = \|x\|_1$	$\Psi(x) = \mathrm{TV}(x)$

Laplace operator

Solving Regularized Inverse Problem

- Objective or "loss" function of general inverse problem: $\text{minimize}_x \ \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\Psi(\boldsymbol{x})$

  weight of regularizer

- Practical #1 go-to solution: Adam solver implemented in PyTorch
- 3 simple steps, will explore in problem session & homework:
  1. Implement evaluation of loss function
  2. Set hyperparameters, including learning rate
  3. Run
- The "fine print": convenient but doesn't always converge well

b 是已知，A 是已知，回归项自己定义

完全可以用回归来推出 x，但是往往很难最终收敛

第一个方法：HQS

The Half-quadratic Splitting (HQS) Method

- Objective or "loss" function of general inverse problem: $$\text{minimize}_x \ \frac{1}{2}\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2^2 + \underbrace{\lambda}_{\text{weight of regularizer}}\Psi(\boldsymbol{x})$$

- Reformulate as: $$\boxed{\begin{aligned}&\text{minimize}_{\{x,z\}} \ \underbrace{\frac{1}{2}\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2^2}_{f(\boldsymbol{x})} + \underbrace{\lambda\Psi(\boldsymbol{z})}_{g(\boldsymbol{z})}\\ &\quad\text{subject to } \ \boldsymbol{D}\boldsymbol{x}-\boldsymbol{z}=0\end{aligned}}$$

- Remove constraints using penalty term (equivalent for large ρ): $$L_\rho(\boldsymbol{x},\boldsymbol{z}) = f(\boldsymbol{x}) + g(\boldsymbol{z}) + \underbrace{\frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x}-\boldsymbol{z}\|_2^2}_{\text{penalty term}}$$

这里也可以直接用scipy.minimize来做，但如果这样，则Dx-z=0是一个**必须要达到的约束**

而之后要讲的方法（HQS和ADMM）则是可以有一定的宽松，因此往往效果会好一点。

第一个方法：HQS

HQS for Image Deconvolution with TV

Generic: $$L_\rho(\boldsymbol{x},\boldsymbol{z}) = f(\boldsymbol{x}) + g(\boldsymbol{z}) + \frac{\rho}{2}\|\boldsymbol{Dx}-\boldsymbol{z}\|_2^2$$

Deconv: $$L_\rho(\boldsymbol{x},\boldsymbol{z}) = \frac{1}{2}\|\boldsymbol{Cx}-\boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{z}\|_1 + \frac{\rho}{2}\|\boldsymbol{Dx}-\boldsymbol{z}\|_2^2$$

$\boldsymbol{x} \in \mathbb{R}^N$	unknown sharp image
$\boldsymbol{C} \in \mathbb{R}^{N\times N}$	circulant convolution matrix for known kernel c
$\boldsymbol{z} \in \mathbb{R}^{2N}$	slack variable, twice the size of $\boldsymbol{x}$!
$\boldsymbol{D} = \begin{bmatrix}\boldsymbol{D}_x \\ \boldsymbol{D}_y\end{bmatrix} \in \mathbb{R}^{2N\times N}$	finite difference gradients, horizontal & vertical

如果使用TV滤波进行去模糊，各个符号如上

1. 注意 x 是个向量，也就是图片拉成一条直线作为输入
2. 这里 Ψ(z) 被当成是 ||z||，z 也是要求解的变量
3. Dx 就是 TV 滤波，最后一项让 Dx 尽量等于 z，第二项则让 z 尽快能的小，这样达到我们【自然图片梯度很小】这个假设。

HQS for Image Deconvolution with Denoiser

Generic: $$L_\rho(\boldsymbol{x},\boldsymbol{z}) = f(\boldsymbol{x}) + g(\boldsymbol{z}) + \frac{\rho}{2}\|\boldsymbol{Dx}-\boldsymbol{z}\|_2^2$$

Deconv: $$L_\rho(\boldsymbol{x},\boldsymbol{z}) = \frac{1}{2}\|\boldsymbol{Cx}-\boldsymbol{b}\|_2^2 + \lambda\Psi(z) + \frac{\rho}{2}\|\boldsymbol{Dx}-\boldsymbol{z}\|_2^2$$

$\boldsymbol{x} \in \mathbb{R}^N$	unknown sharp image
$\boldsymbol{C} \in \mathbb{R}^{N\times N}$	circulant convolution matrix for known kernel c
$z \in R^N$	same size of x
$D \in R^{N*N}$	单位矩阵 I

如果使用 Denoiser 进行去模糊，各个符号如上

1. Denoiser：输入图像，输出为只有噪声的图，如 DnCNN。
2. 这里 Ψ(z) 是对图片处理后，和 denoiser 生成的结果的差值。
3. 此时的 imager prior：图像中的噪声尽可能和 denoiser 生成的噪声图接近。

第一个方法：HQS

HQS for Image Deconvolution with TV

$$L_\rho(\boldsymbol{x},\boldsymbol{z}) = \frac{1}{2}\|\boldsymbol{Cx}-\boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{z}\|_1 + \frac{\rho}{2}\|\boldsymbol{Dx}-\boldsymbol{z}\|_2^2$$

```
while not converged:
```

$$\boldsymbol{x} \leftarrow \mathrm{prox}_{\|\cdot\|_2,\rho}(\boldsymbol{z}) = \arg\min_x \frac{1}{2}\|\boldsymbol{Cx}-\boldsymbol{b}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{Dx}-\boldsymbol{z}\|_2^2$$

$$\boldsymbol{z} \leftarrow \mathrm{prox}_{\|\cdot\|_1,\rho}(\boldsymbol{Dx}) = \arg\min_z \lambda\|\boldsymbol{z}\|_1 + \frac{\rho}{2}\|\boldsymbol{Dx}-\boldsymbol{z}\|_2^2$$

之后就是数学推导，以可以忽略

第一个方法：HQS

HQS for Image Deconvolution with TV

<u>$\boldsymbol{x}$ – update:</u>

$$\boldsymbol{x} \leftarrow \operatorname{prox}_{\|\cdot\|_2,\rho}(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{C}\boldsymbol{x}-\boldsymbol{b}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x}-\boldsymbol{z}\|_2^2$$

reformulate

$$= \frac{1}{2}(\boldsymbol{C}\boldsymbol{x}-\boldsymbol{b})^T(\boldsymbol{C}\boldsymbol{x}-\boldsymbol{b}) + \frac{\rho}{2}(\boldsymbol{D}\boldsymbol{x}-\boldsymbol{z})^T(\boldsymbol{D}\boldsymbol{x}-\boldsymbol{z})$$

$$= \frac{1}{2}\left(\boldsymbol{x}^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{x} - 2\boldsymbol{x}^T\boldsymbol{C}^T\boldsymbol{b} + \boldsymbol{b}^T\boldsymbol{b}\right) + \frac{\rho}{2}\left(\boldsymbol{x}^T\boldsymbol{D}^T\boldsymbol{D}\boldsymbol{x} - 2\boldsymbol{x}^T\boldsymbol{D}^T\boldsymbol{z} + \boldsymbol{z}^T\boldsymbol{z}\right)$$

find solution by setting gradient to 0

$$0 = \nabla_x f(\boldsymbol{x}) = \boldsymbol{C}^T\boldsymbol{C}\boldsymbol{x} - \boldsymbol{C}^T\boldsymbol{b} + \rho\boldsymbol{D}^T\boldsymbol{D}\boldsymbol{x} - \rho\boldsymbol{D}^T\boldsymbol{z}$$

closed-form solution

$$\boldsymbol{x} \leftarrow \left(\boldsymbol{C}^T\boldsymbol{C} + \rho\boldsymbol{D}^T\boldsymbol{D}\right)^{-1}\left(\boldsymbol{C}^T\boldsymbol{b} + \rho\boldsymbol{D}^T\boldsymbol{z}\right)$$

第一个方法：HQS

HQS for Image Deconvolution with TV

<u>$\boldsymbol{x}$ – update:</u>

$$\boldsymbol{x} \leftarrow \mathrm{prox}_{\|\cdot\|_2,\rho}(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{C}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{z}\|_2^2$$

$$\boldsymbol{x} \leftarrow \underline{\left(\boldsymbol{C}^T\boldsymbol{C} + \rho\boldsymbol{D}^T\boldsymbol{D}\right)^{-1}}\underline{\left(\boldsymbol{C}^T\boldsymbol{b} + \rho\boldsymbol{D}^T\boldsymbol{z}\right)}$$

Exploit duality of algebraic & signal processing interpretation

$$\boldsymbol{C}^T\boldsymbol{C} \Leftrightarrow \mathcal{F}^{-1}\{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\}\}$$

$$\boldsymbol{D}^T\boldsymbol{D} \Leftrightarrow \mathcal{F}^{-1}\left\{\mathcal{F}\{d_x\}^* \cdot \mathcal{F}\{d_x\} + \mathcal{F}\{d_y\}^* \cdot \mathcal{F}\{d_y\}\right\}$$

$$\boldsymbol{D}^T\boldsymbol{z} = \boldsymbol{D}_x^T\boldsymbol{z}_1 + \boldsymbol{D}_y^T\boldsymbol{z}_2 \Leftrightarrow \mathcal{F}^{-1}\left\{\mathcal{F}\{d_x\} *\cdot \mathcal{F}\{z_1\} + \mathcal{F}\{d_y\} *\cdot \mathcal{F}\{z_2\}\right\}$$

$$\boldsymbol{C}^T\boldsymbol{b} \Leftrightarrow \mathcal{F}^{-1}\{\mathcal{F}\{c\} *\cdot \mathcal{F}\{b\}\}$$

$$\underline{\boldsymbol{C}^T\boldsymbol{C} + \rho\boldsymbol{D}^T\boldsymbol{D}} \Leftrightarrow \mathcal{F}^{-1}\{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\} + \rho(\mathcal{F}\{d_x\}^* \cdot \mathcal{F}\{d_x\} + \mathcal{F}\{d_y\}^* \cdot \mathcal{F}\{d_y\})\}$$

$$\underline{\boldsymbol{C}^T\boldsymbol{b} + \rho\boldsymbol{D}^T\boldsymbol{z}} \Leftrightarrow \mathcal{F}^{-1}\{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{b\} + \rho(\mathcal{F}\{d_x\}^* \cdot \mathcal{F}\{z_1\} + \mathcal{F}\{d_y\}^* \cdot \mathcal{F}\{z_2\})\}$$

第一个方法：HQS

HQS for Image Deconvolution with TV

$\boldsymbol{x}$ – update:

$$\boldsymbol{x} \leftarrow \mathrm{prox}_{\|\cdot\|_2,\rho}(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{Cx}-\boldsymbol{b}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{Dx}-\boldsymbol{z}\|_2^2$$

$$\boldsymbol{x} \leftarrow \left(\boldsymbol{C}^T\boldsymbol{C} + \rho\boldsymbol{D}^T\boldsymbol{D}\right)^{-1}\left(\boldsymbol{C}^T\boldsymbol{b} + \rho\boldsymbol{D}^T\boldsymbol{z}\right)$$

- Efficient $\boldsymbol{x}$–update operates purely on 2D images with FFTs and element-wise multiplications and divisions:

$$\mathrm{prox}_{\|\cdot\|_2,\rho}(\boldsymbol{z}) = \mathcal{F}^{-1}\left\{\frac{\boxed{\mathcal{F}\{c\}^*\cdot\mathcal{F}\{b\}} + \rho\left(\mathcal{F}\{d_x\}^*\cdot\mathcal{F}\{z_1\} + \mathcal{F}\{d_y\}^*\cdot\mathcal{F}\{z_2\}\right)}{\boxed{\mathcal{F}\{c\}^*\cdot\mathcal{F}\{c\} + \rho\left(\mathcal{F}\{d_x\}^*\cdot\mathcal{F}\{d_x\} + \mathcal{F}\{d_y\}^*\cdot\mathcal{F}\{d_y\}\right)}}\right\}$$

can pre-compute most parts

$z_1 = \boldsymbol{z}(1:N), z_2 = \boldsymbol{z}(N+1:2N)$

HQS for Image Deconvolution with TV

$\boldsymbol{z}$ – update:

$$\boldsymbol{z} \leftarrow \mathrm{prox}_{\|\cdot\|_1,\rho}(\boldsymbol{Dx}) = \arg\min_{\boldsymbol{z}} \lambda\|\boldsymbol{z}\|_1 + \frac{\rho}{2}\|\boldsymbol{Dx}-\boldsymbol{z}\|_2^2$$

- Efficient $\boldsymbol{z}$–update uses element-wise soft thresholding operator $\mathcal{S}_\kappa(\cdot)$:

$$\mathrm{prox}_{\|\cdot\|_1,\rho}(\boldsymbol{v}) = \mathcal{S}_\kappa(\boldsymbol{v}) = \begin{cases} v-\kappa & v>\kappa \\ 0 & |v|\le\kappa \\ v+\kappa & v<-\kappa \end{cases} = (v-\kappa)_+ - (-v-\kappa)_+$$

$\kappa = \lambda/\rho$

$\boldsymbol{v} = \boldsymbol{Dx}$

This element-wise soft thresholding is the proximal operator for anisotropic TV, see course notes on block soft thresholding for isotropic TV.

使用TV作为image prior进行去模糊的推导结果

第一个方法：HQS

HQS for Image Deconvolution with Denoiser

$\boldsymbol{x}$ – update:

$$\boldsymbol{x} \leftarrow \mathrm{prox}_{\|\cdot\|_2,\rho}(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{C}\boldsymbol{x}-\boldsymbol{b}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{x}-\boldsymbol{z}\|_2^2 \qquad \boldsymbol{z} \in \mathbb{R}^N$$

$$\boldsymbol{x} \leftarrow \left(\boldsymbol{C}^T\boldsymbol{C}+\rho\boldsymbol{I}\right)^{-1}\left(\boldsymbol{C}^T\boldsymbol{b}+\rho\boldsymbol{z}\right)$$

no matrix $\boldsymbol{D}$!

- Efficient $\boldsymbol{x}$–update operates purely on 2D images with FFTs and element-wise multiplications and divisions:

$$\mathrm{prox}_{\|\cdot\|_2,\rho}(\boldsymbol{z}) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{b\} + \rho\mathcal{F}\{z\}}{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\} + \rho}\right\}$$

HQS for Image Deconvolution with Denoiser

$\boldsymbol{z}$ – update:

$$\boldsymbol{z} \leftarrow \mathrm{prox}_{\mathcal{D},\rho}(\boldsymbol{x}) = \arg\min_{\boldsymbol{z}} \lambda\Psi(\boldsymbol{z}) + \frac{\rho}{2}\|\boldsymbol{x}-\boldsymbol{z}\|_2^2$$

$$= \arg\min_{\boldsymbol{z}} \Psi(\boldsymbol{z}) + \frac{\rho}{2\lambda}\|\boldsymbol{x}-\boldsymbol{z}\|_2^2$$

- Efficient $\boldsymbol{z}$–update uses arbitrary denoiser $\mathcal{D}(\cdot)$, such as DnCNN and non-local means, using noise variance $\sigma^2 = \frac{\lambda}{\rho}$

$$\mathrm{prox}_{\mathcal{D},\rho}(\boldsymbol{x}) = \mathcal{D}\left(\boldsymbol{x}, \sigma^2 = \frac{\lambda}{\rho}\right)$$

使用 denoiser 作为image prior进行去模糊的推导结果

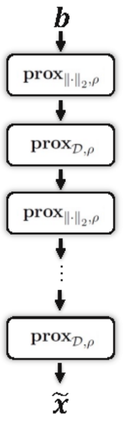
如第6页所阐述的：Ψ 是指和 Denoiser 结果的差距

第一个方法：HQS

Outlook on Unrolled Optimization

- Run or “unroll” HQS for K iterations
- Interpret as unrolled feedforward network:

$$x = \mathbf{prox}_{\|\cdot\|_2,\rho}(z) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{c\}^*\cdot\mathcal{F}\{b\}+\rho\mathcal{F}\{z\}}{\mathcal{F}\{c\}^*\cdot\mathcal{F}\{c\}+\rho}\right\}$$
$$z = \mathbf{prox}_{\mathcal{D},\rho}(\mathbf{x}) = \mathcal{D}\left(\mathbf{x}, \sigma^2 = \frac{\lambda}{\rho}\right)$$
$$x = \mathbf{prox}_{\|\cdot\|_2,\rho}(z) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{c\}^*\cdot\mathcal{F}\{b\}+\rho\mathcal{F}\{z\}}{\mathcal{F}\{c\}^*\cdot\mathcal{F}\{c\}+\rho}\right\}$$
$$z = \mathbf{prox}_{\mathcal{D},\rho}(\mathbf{x}) = \mathcal{D}\left(\mathbf{x}, \sigma^2 = \frac{\lambda}{\rho}\right)$$
$$x = \mathbf{prox}_{\|\cdot\|_2,\rho}(z) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{c\}^*\cdot\mathcal{F}\{b\}+\rho\mathcal{F}\{z\}}{\mathcal{F}\{c\}^*\cdot\mathcal{F}\{c\}+\rho}\right\}$$
$$z = \mathbf{prox}_{\mathcal{D},\rho}(\mathbf{x}) = \mathcal{D}\left(\mathbf{x}, \sigma^2 = \frac{\lambda}{\rho}\right)$$
$$x = \mathbf{prox}_{\|\cdot\|_2,\rho}(z) = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{c\}^*\cdot\mathcal{F}\{b\}+\rho\mathcal{F}\{z\}}{\mathcal{F}\{c\}^*\cdot\mathcal{F}\{c\}+\rho}\right\}$$
$$z = \mathbf{prox}_{\mathcal{D},\rho}(\mathbf{x}) = \mathcal{D}\left(\mathbf{x}, \sigma^2 = \frac{\lambda}{\rho}\right)$$
$$\vdots$$



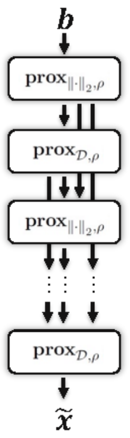
[Diamond et al. 2017]

Outlook on Unrolled Optimization

- Run or “unroll” HQS for K iterations
- Interpret as unrolled feedforward network:

Benefits over unrolled optimization

- Learnable parameters: $\lambda^{(k)}, \rho^{(k)}$, denoiser $\mathcal{D}^{(k)}$
- DenseNet-like skip connections
- Denoiser/regularizer can adapt to matrix $\boldsymbol{C}$
- Can train with advanced loss functions (perceptual, adversarial, other network, …)



[Diamond et al. 2017]

References and Further Reading

Must read: EE367 course notes on Image Deconvolution with the Half-quadratic splitting method.
Optional read: EE367 course notes on Noise, Denoising, and Image Reconstruction with Noise

Adam
- D. Kingma, J. Ba "Adam: A method for stochastic optimization", ICLR 2015

HQS
- D. Geman and C. Yang "Nonlinear image recovery with half-quadratic regularization", IEEE Transactions on Image Processing, 1995

TV Prior and Extensions
- L. Rudin, S. Osher, E. Fatemi "Nonlinear total variation-based noise removal algorithm", Physica D, 1992
- A. Levin, Y. Weiss, F. Durand, W. Freeman "Understanding and evaluating blind deconvolution algorithms", CVPR 2009
- D. Krishnan, R. Fergus "Fast Image Deconvolution using Hyper-Laplacian Priors", NIPS 2009
- K. Bredies, K. Kunisch, T. Pock "Total Generalized Variation", Technical Report 2009
- S. Lefkimmiatis, J. Ward, M. Unser "Hessian Schatten-Norm Regularization for Linear Inverse Problems", IEEE Transactions on Image Processing 2003

Unrolled Optimization
- S. Diamond, V. Sitzmann, F. Heide, G. Wetzstein "Unrolled optimization with deep priors", arxiv, 2017

更好的优化：不细看

第二个方法：ADMM

HQS vs. ADMM

- Objective function: $$\text{minimize}_{\boldsymbol{x}}\ \frac{1}{2}\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2^2+\lambda\Psi(\boldsymbol{x})$$
- Reformulate as: $$\text{minimize}_{\{x,z\}}\ \underbrace{\frac{1}{2}\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2^2}_{f(\boldsymbol{x})}+\underbrace{\lambda\Psi(\boldsymbol{z})}_{g(\boldsymbol{z})}$$
  subject to $\boldsymbol{D}\boldsymbol{x}-\boldsymbol{z}=0$
- Penalty Method of HQS: $$L_\rho^{(\text{HQS})}(\boldsymbol{x},\boldsymbol{z})=f(\boldsymbol{x})+g(\boldsymbol{z})+\frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x}-\boldsymbol{z}\|_2^2$$
- Augmented Lagrangian: $$L_\rho^{(\text{ADMM})}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{y})=f(\boldsymbol{x})+g(\boldsymbol{z})+\boxed{\boldsymbol{y}^T(\boldsymbol{D}\boldsymbol{x}-\boldsymbol{z})}+\frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x}-\boldsymbol{z}\|_2^2$$
  $$\overset{\boldsymbol{u}=(1/\rho)\boldsymbol{y}}{=}f(\boldsymbol{x})+g(\boldsymbol{z})+\boxed{\frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x}-\boldsymbol{z}+\boldsymbol{u}\|_2^2-\frac{\rho}{2}\|\boldsymbol{u}\|_2^2}$$

多加了一个变量，用于控制 Dx-z 一次项

$$\begin{aligned}
&\mathrm{y}^{\mathrm{T}}(Dx-z)+\frac{\rho}{2}*(Dx-z)^2\\
&=\frac{\rho}{2}*(Dx-z)^2+\rho u*(Dx-z)\\
&=\frac{\rho}{2}*[(Dx-z)^2+u*(Dx-z)+u^2-u^2]\\
&=\frac{\rho}{2}*[(Dx-z+u)^2-u^2]
\end{aligned}$$

ADMM方法：多加了一个变量，用于一次项控制

第二个方法：ADMM

ADMM

$$L_\rho^{(\mathrm{ADMM})}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{u}) = f(\boldsymbol{x}) + g(\boldsymbol{z}) + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{z} + \boldsymbol{u}\|_2^2 - \frac{\rho}{2}\|\boldsymbol{u}\|_2^2$$

- Alternating gradient descent approach to solving Augmented Lagrangian:

```
while not converged:
```

$$\boldsymbol{x} \leftarrow \mathrm{prox}_{f,\rho}(\boldsymbol{z}) = \arg\min_x L_\rho^{(\mathrm{ADMM})}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{u}) = \arg\min_x f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{z} + \boldsymbol{u}\|_2^2$$

$$\boldsymbol{z} \leftarrow \mathrm{prox}_{g,\rho}(\boldsymbol{D}\boldsymbol{x}) = \arg\min_z L_\rho^{(\mathrm{ADMM})}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{u}) = \arg\min_z g(\boldsymbol{z}) + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{z} + \boldsymbol{u}\|_2^2$$

$$\boldsymbol{u} \leftarrow \boldsymbol{u} + \boldsymbol{D}\boldsymbol{x} - \boldsymbol{z}$$

第二个方法：ADMM

ADMM

<u>$\boldsymbol{x}$ – update:</u>

$$\boldsymbol{x} \leftarrow \mathrm{prox}_{\|\cdot\|_2,\rho}(\boldsymbol{z}) = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{D}\boldsymbol{x} - \boldsymbol{z} + \boldsymbol{u}\|_2^2,$$

$$\boldsymbol{x} \leftarrow \underbrace{\left(\boldsymbol{A}^T\boldsymbol{A} + \rho\boldsymbol{D}^T\boldsymbol{D}\right)^{-1}}_{\widetilde{\boldsymbol{A}}}\underbrace{\left(\boldsymbol{A}^T\boldsymbol{b} + \rho\boldsymbol{D}^T(\boldsymbol{z} - \boldsymbol{u})\right)}_{\widetilde{\boldsymbol{b}}}$$

- Same general x-update as HQS, use matrix-free iterative solver, such as the conjugate gradient method, to solve $\widetilde{\boldsymbol{A}}\boldsymbol{x} = \widetilde{\boldsymbol{b}}$ (e.g., `scipy.sparse.linalg.cg`)

第二个方法：ADMM

ADMM

<u>$\boldsymbol{z}$ – update for TV regularizer in closed form:</u>

$$\boldsymbol{z} \leftarrow \mathrm{prox}_{\|\cdot\|_1,\rho}(\boldsymbol{v}) = \arg\min_z \lambda\|\boldsymbol{z}\|_1 + \frac{\rho}{2}\|\boldsymbol{v}-\boldsymbol{z}\|_2^2 = \mathcal{S}_\kappa(\boldsymbol{v}),\ \boldsymbol{v} = \boldsymbol{D}\boldsymbol{x} + \boldsymbol{u}$$

<u>$\boldsymbol{z}$ – update for denoising-based regularizer in closed form:</u>

$$\boldsymbol{z} \leftarrow \mathrm{prox}_{\mathcal{D},\rho}(\boldsymbol{x}+\boldsymbol{u}) = \arg\min_z \lambda\Psi(\boldsymbol{z}) + \frac{\rho}{2}\|\boldsymbol{x}-\boldsymbol{z}+\boldsymbol{u}\|_2^2 = \mathcal{D}\left(\boldsymbol{x}+\boldsymbol{u}, \sigma^2 = \frac{\lambda}{\rho}\right)$$

→ Same z-update rules as HQS!

第二个方法：ADMM

ADMM

ADMM for inverse problem with denoiser

initialize ρ and λ
$\mathbf{x} = zeros\,(W, H)\,;$
$\mathbf{z} = zeros\,(W, H)\,;$
$\mathbf{u} = zeros\,(W, H)\,;$
for $k = 1$ **to** max_iters **do**
  $\mathbf{x} = \mathbf{prox}_{\|\cdot\|_2,\rho}(\mathbf{v}) = \text{cg\_solve}\left(\mathbf{A}^T\mathbf{A} + \rho\mathbf{I}, \mathbf{A}^T\mathbf{b} + \rho(\mathbf{z} - \mathbf{u})\right)$
  $\mathbf{prox}_{\mathcal{D},\rho}(\mathbf{x} + \mathbf{u}) = \mathcal{D}\left(\mathbf{x} + \mathbf{u}, \sigma^2 = \frac{\lambda}{\rho}\right)$
  $\mathbf{u} = \mathbf{u} + \mathbf{x} - \mathbf{z}$
end for

ADMM for inverse problem with TV

initialize ρ and λ
$\mathbf{x} = zeros\,(W, H)\,;$
$\mathbf{z} = zeros\,(W, H, 2)\,;$
$\mathbf{u} = zeros\,(W, H, 2)\,;$
for $k = 1$ **to** max_iters **do**
  $\mathbf{x} = \mathbf{prox}_{\|\cdot\|_2,\rho}(\mathbf{z} - \mathbf{u}) = \text{cg\_solve}\left(\mathbf{A}^T\mathbf{A} + \rho\mathbf{D}^T\mathbf{D}, \mathbf{A}^T\mathbf{b} + \rho\mathbf{D}^T(\mathbf{z} - \mathbf{u})\right)$
  $\mathbf{z} = \mathbf{prox}_{\|\cdot\|_1,\rho}(\mathbf{D}\mathbf{x} + \mathbf{u}) = \mathcal{S}_{\lambda/\rho}(\mathbf{D}\mathbf{x} + \mathbf{u})$
  $\mathbf{u} = \mathbf{u} + \mathbf{D}\mathbf{x} - \mathbf{z}$
end for