

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

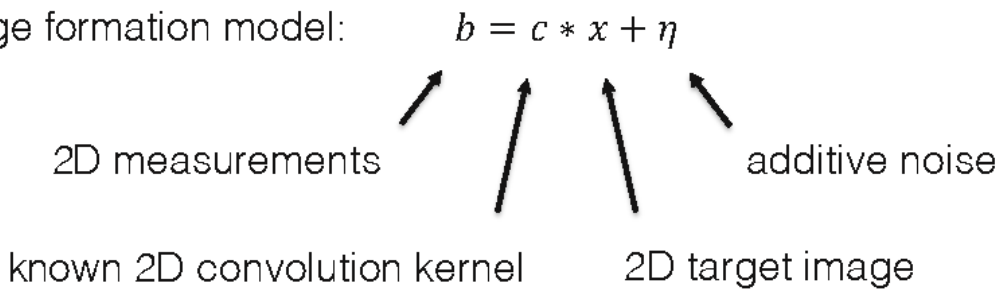
- Image formation model: $b = c * x + \eta$ 

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: $\tilde{x}_{\text{if}} = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{b\}}{\mathcal{F}\{c\}}\right\}$
- Wiener filtering: $\tilde{x}_{\text{wf}} = \mathcal{F}^{-1}\left\{\frac{|\mathcal{F}\{c\}|^2}{|\mathcal{F}\{c\}|^2 + 1/SNR} \cdot \frac{\mathcal{F}\{b\}}{\mathcal{F}\{c\}}\right\}$
- Duality of “signal processing” and “algebraic” interpretation:
$$b = c * x \Leftrightarrow \mathbf{b} = \mathbf{C}\mathbf{x} \quad \mathbf{C} \in \mathbb{R}^{N \times N}, \quad \mathbf{b}, \mathbf{x} \in \mathbb{R}^N$$

基本的符号

A Bayesian Perspective of Inverse Problems

- Image formation model: $\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta}$, $\mathbf{b} \in \mathbb{R}^M, \mathbf{x} \in \mathbb{R}^N, \mathbf{A} \in \mathbb{R}^{M \times N}$

- Interpret as random variables:
 $\mathbf{x}_i \sim \mathcal{N}(\mathbf{x}_i, 0)$, $\boldsymbol{\eta}_i \sim \mathcal{N}(0, \sigma^2)$
 $\mathbf{b}_i \sim \mathcal{N}((\mathbf{A}\mathbf{x})_i, \sigma^2)$

- Probability of observation i :

$$p(\mathbf{b}_i | \mathbf{x}_i, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\mathbf{b}_i - (\mathbf{A}\mathbf{x})_i)^2}{2\sigma^2}}$$

- Joint probability of all observations:

$$p(\mathbf{b} | \mathbf{x}, \sigma) = \prod_{i=1}^M p(\mathbf{b}_i | \mathbf{x}_i, \sigma) \propto e^{-\frac{\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2}{2\sigma^2}}$$



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

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

$$\mathbf{x} = \operatorname{argmax} \log(P(\mathbf{b} | \mathbf{x})) + \log(P(\mathbf{x}))$$

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

image prior

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

因为是正比, 所以可以得出

$$\mathbf{x}_{MAP} = \operatorname{argmin}_x \frac{1}{2\sigma^2} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 + \Psi(\mathbf{x})$$

data fidelity term
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



Solving Regularized Inverse Problem

- Objective or “loss” function of general inverse problem:
$$\underset{x}{\text{minimize}} \frac{1}{2} \|b - Ax\|_2^2 + \lambda \Psi(x)$$

↑
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:
$$\underset{x}{\text{minimize}} \frac{1}{2} \|b - Ax\|_2^2 + \lambda \Psi(x)$$

↑
weight of regularizer

- Reformulate as:

$$\underset{\{x,z\}}{\text{minimize}} \underbrace{\frac{1}{2} \|b - Ax\|_2^2}_{f(x)} + \underbrace{\lambda \Psi(z)}_{g(z)}$$
$$\text{subject to } Dx - z = 0$$

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。

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

第一个方法: HQS

HQS for Image Deconvolution with TV

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

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

$\mathbf{x} \in \mathbb{R}^N$ unknown sharp image

$\mathbf{C} \in \mathbb{R}^{N \times N}$ circulant convolution matrix for known kernel c

$\mathbf{z} \in \mathbb{R}^{2N}$ slack variable, twice the size of $\mathbf{x}$!

$\mathbf{D} = \begin{bmatrix} \mathbf{D}_x \\ \mathbf{D}_y \end{bmatrix} \in \mathbb{R}^{2N \times N}$ 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 $\mathbf{x}$ 是个向量, 也就是图片拉成一条直线作为输入
2. 这里 $\Psi(\mathbf{z})$ 被当成是 $\|\mathbf{z}\|$, $\mathbf{z}$ 也是要求解的变量
3. $\mathbf{D}\mathbf{x}$ 就是 TV 滤波, 最后一项让 $\mathbf{D}\mathbf{x}$ 尽量等于 $\mathbf{z}$, 第二项则让 $\mathbf{z}$ 尽快的小, 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: $L_\rho(\mathbf{x}, \mathbf{z}) = f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{D}\mathbf{x} - \mathbf{z}\|_2^2$

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

$\mathbf{x} \in \mathbb{R}^N$ unknown sharp image

$\mathbf{C} \in \mathbb{R}^{N \times N}$ circulant convolution matrix for known kernel c

$\mathbf{z} \in \mathbb{R}^N$ same size of $\mathbf{x}$

$\mathbf{D} \in \mathbb{R}^{N \times N}$ 单位矩阵 $\mathbf{I}$

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 $\Psi(\mathbf{z})$ 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

$$L_{\rho}(\mathbf{x}, \mathbf{z}) = \frac{1}{2} \|\mathbf{C}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{z}\|_1 + \frac{\rho}{2} \|\mathbf{D}\mathbf{x} - \mathbf{z}\|_2^2$$

while not converged:

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

$$\mathbf{z} \leftarrow \text{prox}_{\|\cdot\|_1, \rho}(\mathbf{D}\mathbf{x}) = \arg \min_{\mathbf{z}} \lambda \|\mathbf{z}\|_1 + \frac{\rho}{2} \|\mathbf{D}\mathbf{x} - \mathbf{z}\|_2^2$$

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

x - update:

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

$$\begin{aligned} & \swarrow \text{reformulate} \\ &= \frac{1}{2} (\mathbf{C}\mathbf{x} - \mathbf{b})^T (\mathbf{C}\mathbf{x} - \mathbf{b}) + \frac{\rho}{2} (\mathbf{D}\mathbf{x} - \mathbf{z})^T (\mathbf{D}\mathbf{x} - \mathbf{z}) \\ &= \frac{1}{2} (\mathbf{x}^T \mathbf{C}^T \mathbf{C} \mathbf{x} - 2\mathbf{x}^T \mathbf{C}^T \mathbf{b} + \mathbf{b}^T \mathbf{b}) + \frac{\rho}{2} (\mathbf{x}^T \mathbf{D}^T \mathbf{D} \mathbf{x} - 2\mathbf{x}^T \mathbf{D}^T \mathbf{z} + \mathbf{z}^T \mathbf{z}) \end{aligned}$$

↓ find solution by setting gradient to 0

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

↓ closed-form solution

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

第一个方法: HQS

HQS for Image Deconvolution with TV

$\mathbf{x}$ - update:

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

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

Exploit duality of algebraic & signal processing interpretation

$$\begin{aligned} \mathbf{C}^T \mathbf{C} &\Leftrightarrow \mathcal{F}^{-1}\{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{c\}\} & \mathbf{D}^T \mathbf{z} = \mathbf{D}_x^T \mathbf{z}_1 + \mathbf{D}_y^T \mathbf{z}_2 &\Leftrightarrow \mathcal{F}^{-1}\{\mathcal{F}\{d_x\}^* \cdot \mathcal{F}\{z_1\} + \mathcal{F}\{d_y\}^* \cdot \mathcal{F}\{z_2\}\} \\ \mathbf{D}^T \mathbf{D} &\Leftrightarrow \mathcal{F}^{-1}\{\mathcal{F}\{d_x\}^* \cdot \mathcal{F}\{d_x\} + \mathcal{F}\{d_y\}^* \cdot \mathcal{F}\{d_y\}\} & \mathbf{C}^T \mathbf{b} &\Leftrightarrow \mathcal{F}^{-1}\{\mathcal{F}\{c\}^* \cdot \mathcal{F}\{b\}\} \end{aligned}$$

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

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

第一个方法: HQS

HQS for Image Deconvolution with TV

$\mathbf{x}$ -update:

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

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

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

$$\text{prox}_{\|\cdot\|_2, \rho}(\mathbf{z}) = \mathcal{F}^{-1} \left\{ \begin{array}{l} \mathcal{F}\{\mathbf{c}\}^* \cdot \mathcal{F}\{\mathbf{b}\} + \rho(\mathcal{F}\{d_x\}^* \cdot \mathcal{F}\{z_1\} + \mathcal{F}\{d_y\}^* \cdot \mathcal{F}\{z_2\}) \\ \mathcal{F}\{\mathbf{c}\}^* \cdot \mathcal{F}\{\mathbf{c}\} + \rho(\mathcal{F}\{d_x\}^* \cdot \mathcal{F}\{d_x\} + \mathcal{F}\{d_y\}^* \cdot \mathcal{F}\{d_y\}) \end{array} \right\}$$



can pre-compute most parts

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

HQS for Image Deconvolution with TV

$\mathbf{z}$ -update:

$$\mathbf{z} \leftarrow \text{prox}_{\|\cdot\|_1, \rho}(\mathbf{D}\mathbf{x}) = \arg \min_{\mathbf{z}} \lambda \|\mathbf{z}\|_1 + \frac{\rho}{2} \|\mathbf{D}\mathbf{x} - \mathbf{z}\|_2^2$$

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

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

$$\kappa = \lambda/\rho$$

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

$$\mathbf{v} = \mathbf{D}\mathbf{x}$$

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

$\mathbf{x}$ - update:

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

$$\mathbf{x} \leftarrow (\mathbf{C}^T \mathbf{C} + \rho \mathbf{I})^{-1} (\mathbf{C}^T \mathbf{b} + \rho \mathbf{z}) \quad \text{no matrix } \mathbf{D}!$$

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

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

HQS for Image Deconvolution with Denoiser

$\mathbf{z}$ - update:

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

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

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

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

使用 denoiser 作为image prior进行去模糊的推导结果

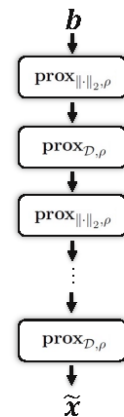
如第6页所阐述的: Ψ 是指和 Denoiser 结果的差距

第一个方法: HQS

Outlook on Unrolled Optimization

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

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



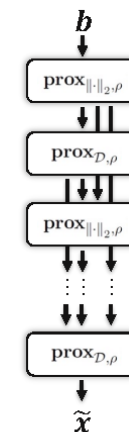
[Diamond et al. 2017]

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 $\mathcal{C}$
- Can train with advanced loss functions (perceptual, adversarial, other network, ...)



[Diamond et al. 2017]

References and Further Reading

Must read: EE357 course notes on Image Deconvolution with the Half-quadratic splitting method.

Optional read: IT367 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: $\text{minimize}_x \frac{1}{2} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \Psi(\mathbf{x})$
- Reformulate as:
$$\text{minimize}_{\{\mathbf{x}, \mathbf{z}\}} \underbrace{\frac{1}{2} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2}_{f(\mathbf{x})} + \underbrace{\lambda \Psi(\mathbf{z})}_{g(\mathbf{z})}$$

subject to $\mathbf{D}\mathbf{x} - \mathbf{z} = 0$

- Penalty Method of HQS: $L_\rho^{(\text{HQS})}(\mathbf{x}, \mathbf{z}) = f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{D}\mathbf{x} - \mathbf{z}\|_2^2$

- Augmented Lagrangian:
$$L_\rho^{(\text{ADMM})}(\mathbf{x}, \mathbf{z}, \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^T(\mathbf{D}\mathbf{x} - \mathbf{z}) + \frac{\rho}{2} \|\mathbf{D}\mathbf{x} - \mathbf{z}\|_2^2$$

$$\stackrel{u = (1/\rho)\mathbf{y}}{=} f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{D}\mathbf{x} - \mathbf{z} + \mathbf{u}\|_2^2 - \frac{\rho}{2} \|\mathbf{u}\|_2^2$$

多加了一个变量, 用于控制 $\mathbf{D}\mathbf{x} - \mathbf{z}$ 一次项

$$\begin{aligned} & \mathbf{y}^T(\mathbf{D}\mathbf{x} - \mathbf{z}) + \frac{\rho}{2} \|\mathbf{D}\mathbf{x} - \mathbf{z}\|_2^2 \\ &= \frac{\rho}{2} \|\mathbf{D}\mathbf{x} - \mathbf{z}\|_2^2 + \rho \mathbf{u}^T (\mathbf{D}\mathbf{x} - \mathbf{z}) \\ &= \frac{\rho}{2} \left[(\mathbf{D}\mathbf{x} - \mathbf{z})^2 + 2\mathbf{u}^T (\mathbf{D}\mathbf{x} - \mathbf{z}) + \mathbf{u}^T \mathbf{u} \right] \\ &= \frac{\rho}{2} \left[(\mathbf{D}\mathbf{x} - \mathbf{z} + \mathbf{u})^2 - \mathbf{u}^T \mathbf{u} \right] \end{aligned}$$

ADMM方法: 多加了一个变量, 用于一次项控制

第二个方法: ADMM

ADMM

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

- Alternating gradient descent approach to solving Augmented Lagrangian:

while not converged:

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

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

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

第二个方法: ADMM

ADMM

$\mathbf{x}$ -update:

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

$$\mathbf{x} \leftarrow \underbrace{(\mathbf{A}^T \mathbf{A} + \rho \mathbf{D}^T \mathbf{D})}^{\tilde{\mathbf{A}}}^{-1} \underbrace{(\mathbf{A}^T \mathbf{b} + \rho \mathbf{D}^T (\mathbf{z} - \mathbf{u}))}_{\tilde{\mathbf{b}}}$$

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

第二个方法: ADMM

ADMM

$\mathbf{z}$ – update for TV regularizer in closed form:

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

$\mathbf{z}$ – update for denoising-based regularizer in closed form:

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

→ Same $\mathbf{z}$ -update rules as HQS!

第二个方法: ADMM

ADMM

ADMM for inverse problem with denoiser

```
1: initialize  $\rho$  and  $\lambda$ 
2:  $\mathbf{x} = \text{zeros}(W, H)$ ;
3:  $\mathbf{z} = \text{zeros}(W, H)$ ;
4:  $\mathbf{u} = \text{zeros}(W, H)$ ;
5: for  $k = 1$  to  $\text{max\_iters}$  do
6:    $\mathbf{x} = \text{prox}_{\|\cdot\|_{2,\rho}}(\mathbf{v}) = \text{cg\_solve}(\mathbf{A}^T \mathbf{A} + \rho \mathbf{I}, \mathbf{A}^T \mathbf{b} + \rho(\mathbf{z} - \mathbf{u}))$ 
7:    $\text{prox}_{\mathcal{D},\rho}(\mathbf{x} + \mathbf{u}) = \mathcal{D}\left(\mathbf{x} + \mathbf{u}, \sigma^2 = \frac{\lambda}{\rho}\right)$ 
8:    $\mathbf{u} = \mathbf{u} + \mathbf{x} - \mathbf{z}$ 
9: end for
```

ADMM for inverse problem with TV

```
1: initialize  $\rho$  and  $\lambda$ 
2:  $\mathbf{x} = \text{zeros}(W, H)$ ;
3:  $\mathbf{z} = \text{zeros}(W, H, 2)$ ;
4:  $\mathbf{u} = \text{zeros}(W, H, 2)$ ;
5: for  $k = 1$  to  $\text{max\_iters}$  do
6:    $\mathbf{x} = \text{prox}_{\|\cdot\|_{2,\rho}}(\mathbf{z} - \mathbf{u}) = \text{cg\_solve}(\mathbf{A}^T \mathbf{A} + \rho \mathbf{D}^T \mathbf{D}, \mathbf{A}^T \mathbf{b} + \rho \mathbf{D}^T(\mathbf{z} - \mathbf{u}))$ 
7:    $\mathbf{z} = \text{prox}_{\|\cdot\|_1,\rho}(\mathbf{D}\mathbf{x} + \mathbf{u}) = \mathcal{S}_{\lambda/\rho}(\mathbf{D}\mathbf{x} + \mathbf{u})$ 
8:    $\mathbf{u} = \mathbf{u} + \mathbf{D}\mathbf{x} - \mathbf{z}$ 
9: end for
```
