Paper Review: Adversarial Regularizers in Inverse Problems

Jevgenija Rudzusika

KTH Royal Institute of Technology

November 20, 2018

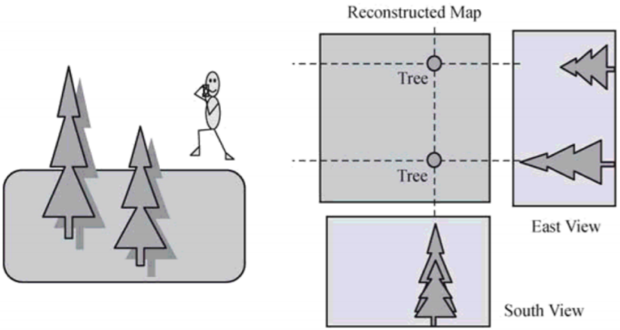
Outline

Lunz, S., Öktem, O. and Schönlieb, C.B., “**Adversarial Regularizers in Inverse Problems**”, NIPS 2018.

- Introduction into CT
- Why end-to-end learning for CT reconstruction is problematic
- Adversarial Regularizers

Introduction into CT

Principle



Milan Zvolský. **Tomographic Image Reconstruction.An Introduction**, Lecture on Medical Physics 28.11.2014

Introduction into CT

Principle

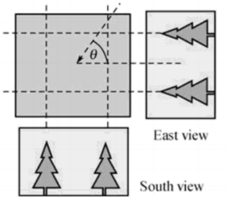


Figure: Two trees seen on *both* views

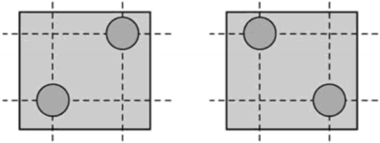
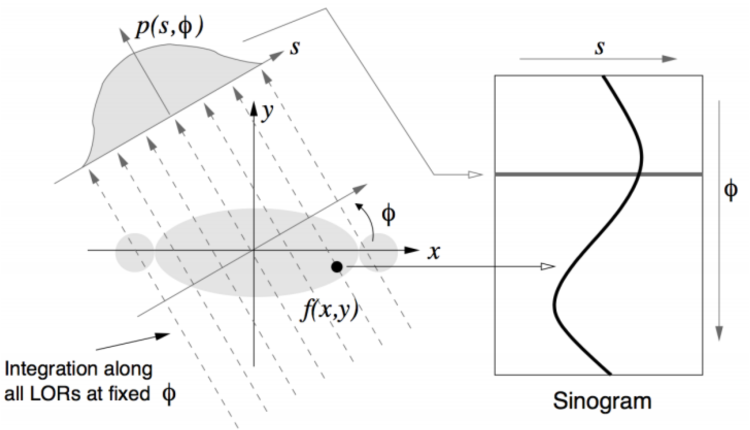


Figure: There are two solutions

Milan Zvolský. **Tomographic Image Reconstruction.An Introduction**, Lecture on Medical Physics 28.11.2014

Introduction into CT

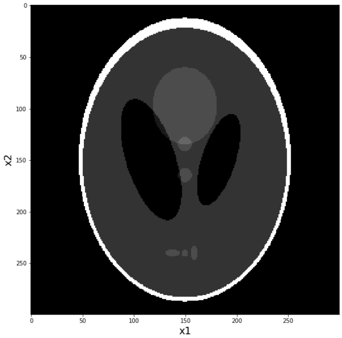
Principle



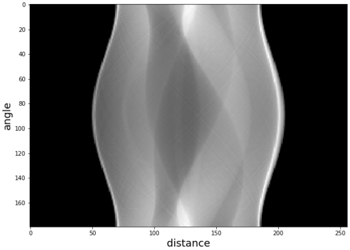
Milan Zvolský. **Tomographic Image Reconstruction.An Introduction**, Lecture on Medical Physics 28.11.2014

Introduction into CT

Sinograms



(a) Image



(b) Projections

Introduction into CT

Beer–Lambert law

- $$I = I_0 \exp\{-\mu \Delta x\} \quad (1)$$

Introduction into CT

Beer–Lambert law

- $$I = I_0 \exp\left\{-\mu \Delta x\right\} \tag{1}$$
- $$I = I_0 \exp\left\{-\int_L \mu(x) dx\right\} \tag{2}$$

Introduction into CT

Beer–Lambert law

- $$I = I_0 \exp\{-\mu \Delta x\} \tag{1}$$

- $$I = I_0 \exp\left\{-\int_L \mu(x) dx\right\} \tag{2}$$

- $$-\ln \frac{I}{I_0} = \int_L \mu(x) dx \tag{3}$$

Introduction into CT

Beer–Lambert law

- $$I = I_0 \exp\{-\mu \Delta x\} \tag{1}$$

- $$I = I_0 \exp\left\{-\int_L \mu(x)dx\right\} \tag{2}$$

- $$-\ln \frac{I}{I_0} = \int_L \mu(x)dx \tag{3}$$

Hsieh, Jiang. **Computed tomography: principles, design, artifacts, and recent advances.** Bellingham, WA: SPIE, 2009.

Introduction to CT

Reconstruction - Inverse problem

- y = $(y_{a,d,z})$ - projections

Introduction to CT

Reconstruction - Inverse problem

- y = $(y_{a,d,z})$ - projections
- x = $(x_{i,j,k})$ - image

Introduction to CT

Reconstruction - Inverse problem

- y = ($y_{a,d,z}$) - projections
- x = ($x_{i,j,k}$) - image
-

$$\begin{aligned} y &= A(x) \\ x &= A^{\dagger}(y) \end{aligned} \tag{4}$$

Introduction to CT

Reconstruction - Inverse problem

- y = ($y_{a,d,z}$) - projections
- x = ($x_{i,j,k}$) - image
- $$\begin{aligned} y &= A(x) \\ x &= A^{\dagger}(y) \end{aligned} \tag{4}$$
- Ill-posed problem (unstable with respect to noise)

Introduction to CT

Reconstruction - Inverse problem

- y = ($y_{a,d,z}$) - projections
- x = ($x_{i,j,k}$) - image
-
$$\begin{aligned} y &= A(x) \\ x &= A^{\dagger}(y) \end{aligned} \tag{4}$$
- Ill-posed problem (unstable with respect to noise)
- But in practice we have

$$y = A(x) + noise \tag{5}$$

- Variational approach:

$$\hat{x} = \arg\min_{x} \left\{ \|A(x) - y\|^2 + \lambda f(x) \right\} \tag{6}$$

- Variational approach:

$$\hat{x} = \arg\min_{x} \left\{ \|A(x) - y\|^2 + \lambda f(x) \right\} \tag{6}$$

- Prior $f(x)$ - ?

Introduction to CT

Inverse problem

- Variational approach:

$$\hat{x} = \arg\min_x \left\{ \|A(x) - y\|^2 + \lambda f(x) \right\} \tag{6}$$

- Prior $f(x)$ - ?
- TV-regularisation:

$$\hat{x} = \arg\min_x \left\{ \|A(x) - y\|^2 + \lambda |\Delta(x)| \right\} \tag{7}$$

Introduction to CT

Inverse problem

- Variational approach:

$$\hat{x} = \arg\min_x \left\{ \|A(x) - y\|^2 + \lambda f(x) \right\} \tag{6}$$

- Prior $f(x)$ - ?
- TV-regularisation:

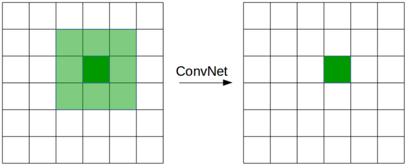
$$\hat{x} = \arg\min_x \left\{ \|A(x) - y\|^2 + \lambda |\Delta(x)| \right\} \tag{7}$$

- $f(x) = NeuralNetwork(x)$?

End-to-end learning

Problems

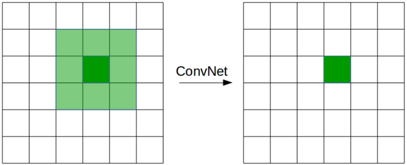
- Compute Vision



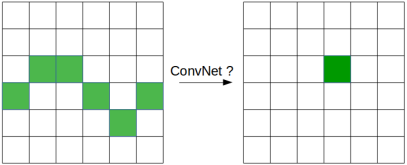
End-to-end learning

Problems

- Compute Vision



- Computed Tomography



End-to-end learning

Problems

- Mayo dataset:
  - $X \sim 512 * 512 * 559$
  - $Y \sim 736 * 64 * 48000 \sim 10GB$

End-to-end learning

Problems

- Mayo dataset:
  - $X \sim 512 * 512 * 559$
  - $Y \sim 736 * 64 * 48000 \sim 10GB$
  - Amount of data $\sim$ 10 samples

End-to-end learning

Problems

- Mayo dataset:
  - $X \sim 512 * 512 * 559$
  - $Y \sim 736 * 64 * 48000 \sim 10GB$
  - Amount of data $\sim$ 10 samples
- Zhu, Bo, et al. **"Image reconstruction by domain-transform manifold learning."** Nature 555.7697 (2018): 487.
  - Fully-connected layers followed by convolutional and de-convolutional layers
  - Used down-sampled 2D image slices

Adversarial Regularizers

Regularization functional as critic

- Learn if x belongs to the space of natural images.

$$\hat{x} = \arg\min_{x} \left\{ \|A(x) - y\|^2 + \lambda \Psi_{\Theta}(x) \right\} \tag{8}$$

Adversarial Regularizers

Regularization functional as critic

- Learn if x belongs to the space of natural images.

$$\hat{x} = \arg\min_{x} \left\{ \|A(x) - y\|^2 + \lambda \Psi_{\Theta}(x) \right\} \tag{8}$$

- Distributions:
  - P_r - ground truth images x_i
  - P_Y - measurements y_i
  - $P_n = A^{\dagger} P_Y$

Adversarial Regularizers

Regularization functional as critic

- Learn if x belongs to the space of natural images.

$$\hat{x} = \arg\min_x \left\{ \|A(x) - y\|^2 + \lambda \Psi_\Theta(x) \right\} \tag{8}$$

- Distributions:
  - P_r - ground truth images x_i
  - P_Y - measurements y_i
  - $P_n = A^\dagger P_Y$
- Goal: tell apart P_n and P_Y

Adversarial Regularizers

Regularization functional as critic

- Minimize

$$\mathbb{E}_{X\sim P_r}[\Psi_\Theta(X)] - \mathbb{E}_{X\sim P_n}[\Psi_\Theta(X)] + \lambda(\|\Delta\Psi_\Theta(X)\| - 1)_+^2 \quad (9)$$

- which corresponds to Wasserstein distance between distributions

$$Wass(P_r, P_n) = \sup_{f\in 1-Lip} \mathbb{E}_{X\sim P_n}[f(X)] - \mathbb{E}_{X\sim P_r}[f(X)] \quad (10)$$

- Original WGANs used clipping to ensure Lipchitz continuity, but then paper[1] proposed a better way.

[1] Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC. **Improved training of wasserstein gans**. InAdvances in Neural Information Processing Systems 2017 (pp. 5767-5777).

Adversarial Regularizers

Regularization functional as critic

Algorithm 1: Learning a regularization functional $\Psi_\Theta(X)$

```
while Θ has not converged do
   for i ∈ 1...m do
      Sample x_r ~ P_r , y ~ P_Y and ε ~ U[0,1]
      x_n ← A† y
      x_i ← ε x_r + (1 − ε) x_n
      L_i ← Ψ_Θ(x_r) − Ψ_Θ(x_n) + μ(‖ΔΨ_Θ(x_i)‖ − 1)_+^2
   end for
   Θ ← Adam(Δ_Θ Σ_{i=1}^m L_i)
end while
```

$$
\begin{aligned}
&\textbf{while } \Theta \text{ has not converged } \textbf{do}\\
&\quad \textbf{for } i \in 1 \dots m \textbf{ do}\\
&\quad\quad \text{Sample } x_r \sim P_r,\ y \sim P_Y \text{ and } \epsilon \sim U[0,1]\\
&\quad\quad x_n \leftarrow A^\dagger y\\
&\quad\quad x_i \leftarrow \epsilon x_r + (1-\epsilon) x_n\\
&\quad\quad L_i \leftarrow \Psi_\Theta(x_r) - \Psi_\Theta(x_n) + \mu(\|\Delta \Psi_\Theta(x_i)\| - 1)_+^2\\
&\quad \textbf{end for}\\
&\quad \Theta \leftarrow Adam(\Delta_\Theta \textstyle\sum_{i=1}^m L_i)\\
&\textbf{end while}
\end{aligned}
$$

Adversarial Regularizers

Regularization functional as critic

Algorithm 2: Applying the regularization functional $\Psi_{\Theta}(X)$ with gradient descent

$x \leftarrow A^{\dagger} y$

```
while stopping criterion not satisfied do
```

$$x \leftarrow x - \epsilon \Delta[\|Ax - y\|^2 + \lambda \Psi_{\Theta}(x)]$$

```
end while
return x
```

Adversarial Regularizers

Distributional Analysis

- **Theorem 1**: Application of Ψ_Θ minimizes Wasserstein distance between P_r and P_η.
- Assume:
  - $g_\eta(x) = x - \eta \Delta \Psi_\Theta(x)$
  - P_η distribution after one gradient step
  - Ψ_Θ has been trained to perfection

Adversarial Regularizers

Distributional Analysis

- **Theorem 1**: Application of Ψ_Θ minimizes Wasserstein distance between P_r and P_η.
- Assume:
  - $g_\eta(x) = x - \eta \Delta \Psi_\Theta(x)$
  - P_η distribution after one gradient step
  - Ψ_Θ has been trained to perfection
- Then

$$\begin{aligned} \frac{d}{d\eta} Wass(P_r, P_\eta)|_{\eta=0} =& \frac{d}{d\eta} \mathbb{E}_{X \sim P_n} \Psi_\Theta(g_\eta(X))|_{\eta=0} \\ =& \mathbb{E}_{X \sim P_n} \frac{d}{d\eta} \Psi_\Theta(g_\eta(X))|_{\eta=0} \\ =& -\mathbb{E}_{X \sim P_n} \|\Delta \Psi_\Theta(X)\|^2 = -1 \end{aligned} \tag{11}$$

- Ψ_Θ gives the strongest decay of the Wasserstein loss

Adversarial Regularizers

Analysis under data manifold assumption

- **Theorem 2**: $\Psi_\Theta(x)$ takes form of $d_\mathcal{M}(x) = \min_{y \in \mathcal{M}} \|x - y\|_2$
- Assume:
  - P_r supported on weakly compact set $\mathcal{M}$
  - $Proj_\mathcal{M}(P_n) = P_r$
- Then a maximizer to

$$\sup_{f \in 1-Lip} \mathbb{E}_{X \sim P_n} f(X) - \mathbb{E}_{X \sim P_r} f(X) = Wass(P_n, P_r) \tag{12}$$

is given by $f(x) = d_\mathcal{M}(x) = \min_{y \in \mathcal{M}} \|x - y\|_2$

Adversarial Regularizers

Analysis under data manifold assumption

- **Theorem 2**: $\Psi_\Theta(x)$ takes form of $d_{\mathcal{M}}(x) = \min_{y \in \mathcal{M}} \|x - y\|_2$
- Assume:
  - P_r supported on weakly compact set $\mathcal{M}$
  - $Proj_{\mathcal{M}}(P_n) = P_r$
- Then a maximizer to

$$\sup_{f \in 1-Lip} \mathbb{E}_{X \sim P_n} f(X) - \mathbb{E}_{X \sim P_r} f(X) = Wass(P_n, P_r) \qquad (12)$$

  is given by $f(x) = d_{\mathcal{M}}(x) = \min_{y \in \mathcal{M}} \|x - y\|_2$
- Assuming that perfectly trained Ψ_Θ is also a maximizer, does it prove that $\Psi_\Theta(x)$ takes form of $d_{\mathcal{M}}(x)$?

Adversarial Regularizers

Stability

- **Theorem 3**: The algorithm converges and is stable
- Assume:
  - f - 1 Lipchitz
  - $y_n \to y$
  - $x_n = \arg\ \min_x \|Ax - y_n\| + \lambda f(x)$
- Then $x_n \to x$

$$x = \arg\ \min_x \|Ax - y\| + \lambda f(x) \tag{13}$$

Adversarial Regularizers

Measures

- I - image, NI - noisy image
- "PSNR is an approximation to human perception of reconstruction quality"[2]

$$PSNR = 10 \log_{10} \left(\frac{MAX(I)^2}{MSE(NI, I)} \right) \tag{14}$$

- "SSIM is a perception-based model that considers image degradation as perceived change in structural information."[3]

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{15}$$

[2] https://en.wikipedia.org/wiki/Peak$_s$$ignal - to - noise_r atio$

[3] https://en.wikipedia.org/wiki/Structural$_s$$imilarity$

Adversarial Regularizers

Results, denoising

Table 1: Denoising results on BSDS dataset

Method	PSNR (dB)	SSIM
Noisy Image	20.3	.534
MODEL-BASED		
Total Variation [23]	26.3	.836
UNSUPERVISED		
Adversarial Regularizer (ours)	28.2	.892
SUPERVISED		
Denoising N.N. [28]	28.8	.908

Adversarial Regularizers

Results, CT

Table 2: CT reconstruction on LIDC dataset

(a) High noise

Method	PSNR (dB)	SSIM
MODEL-BASED		
Filtered Backprojection	14.9	.227
Total Variation [18]	27.7	.890
UNSUPERVISED		
Adversarial Reg. (ours)	30.5	.927
SUPERVISED		
Post-Processing [15]	31.2	.936

(b) Low noise

Method	PSNR (dB)	SSIM
MODEL-BASED		
Filtered Backprojection	23.3	.604
Total Variation [18]	30.0	.924
UNSUPERVISED		
Adversarial Reg. (ours)	32.5	.946
SUPERVISED		
Post-Processing [15]	33.6	.955

The data was simulated! (1018 images)

Adversarial Regularizers

Extensions

- Regularize small patches to have more data.
- Add partially reconstructed images to the training set

Adversarial Regularizers

Extensions

- Regularize small patches to have more data.
- Add partially reconstructed images to the training set
- Anomaly detection?

Adversarial Regularizers

Conclusions

- \+ Unsupervised - means it is possible to take reconstructed data from another scanner and projections from a new scanner
- \- Not learning from the projection data
- \+ Good theoretical support
- \- Extensive experiments are necessary

Thank you!