Weekly Report 2

Further literature review and brief introduction about future plan

Yuchen Jin

Dept. of ECE, University of Houston

February 8, 2019

Outline

Outline

Introduction

- **Recent works**:
  - Arrange two beamer templates and upload them to Github.
    `https://cainmagi.github.io/projects/latex_templates/`
  - Update a new article, now it has already include 8 brief reviews on different papers.
    1. note20190129: A collection of researches about inverse problem..
  - Read 3 more papers in this week. Until now I have reviewed all papers that are directly related to inverse problems from NIPS-2010 to NIPS-2018. The papers about dictionary learning and general optimization have not been reviewed yet.
    1. About an inspection on Vector-AMP [1].
    2. About learning a regularization term [2].
    3. About deep-learning support ADMM algorithm [3].
- **Future plan**:
  1. How to feed offset into the network.
  2. How to make use of derivative.
  3. How to use transfer learning.

Outline

Recent reviews

Plug-in Estimation in High-Dimensional Linear ... [1]

AMP Iteration

$$b_t = \frac{\|\mathbf{x}_t\|_0}{M}. \quad (1\text{-}1)$$

$$\mathbf{v}_t = \mathbf{y} - \mathbf{A}\mathbf{x}_t + b_t \mathbf{v}_{t-1}. \quad (1\text{-}2)$$

$$\lambda_t = \frac{\alpha \|\mathbf{v}_t\|_2}{\sqrt{M}}. \quad (1\text{-}3)$$

$$\mathbf{x}_{t+1} = \mathrm{prox}_{\lambda_t|\cdot|}(\mathbf{x}_t + \mathbf{A}^T \mathbf{v}_t). \quad (1\text{-}4)$$

- Recall the AMP algorithm which is used to solve $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda|\mathbf{x}|$.
- AMP could converge much faster than ISTA & FISTA.
- LAMP converges faster than LISTA.
- AMP/LAMP requires the **A** to be a (large) i.i.d. (sub)Gaussian random matrix.

Recent reviews

Plug-in Estimation in High-Dimensional Linear ... [1]

- We call $\text{prox}_{\lambda|\cdot|}(\cdot)$ proximal operator, which solution is soft thresholding. This is the denoiser activation.
- Repeating the AMP steps in each iteration but using different activations, we have VAMP.

Algorithm 1 Vector AMP (LMMSE form)

Require: LMMSE estimator $\mathbf{g}_2(\cdot, \gamma_{2k})$ from (4), denoiser $\mathbf{g}_1(\cdot, \gamma_{1k})$, and number of iterations K_{it}.

```
Select initial r_10 and γ_10 ≥ 0.
for k = 0, 1, ..., K_it do
    // Denoising
    x̂_1k = g_1(r_1k, γ_1k)
    α_1k = ⟨∇g_1(r_1k, γ_1k)⟩
    η_1k = γ_1k/α_1k, γ_2k = η_1k − γ_1k
    r_2k = (η_1k x̂_1k − γ_1k r_1k)/γ_2k

    // LMMSE estimation
    x̂_2k = g_2(r_2k, γ_2k)
    α_2k = ⟨∇g_2(r_2k, γ_2k)⟩
    η_2k = γ_2k/α_2k, γ_1,k+1 = η_2k − γ_2k
    r_1,k+1 = (η_2k x̂_2k − γ_2k r_2k)/γ_1,k+1
end for
Return x̂_1K_it.
```

Recent reviews

Plug-in Estimation in High-Dimensional Linear ... [1]

- VAMP has two different activating functions $\mathbf{g}_1(\cdot)$ and $\mathbf{g}_2(\cdot)$.
- $\mathbf{g}_1(\cdot)$ could be an arbitrary denoiser. In (1-4), we use $\mathrm{prox}_{\lambda|\cdot|}(\cdot)$ as $\mathbf{g}_1(\cdot)$.
- $\mathbf{g}_2(\cdot)$ is a solution to the L2-penalized linear inverse problem. See (2)

- VAMP do not requires $\mathbf{A}$ to be Gaussian random. Instead, $\mathbf{A}$ only needs to be an arbitrary right rotationally invariant matrix.
- Conventionally VAMP requires $\mathbf{g}_1$ to be a separable denoiser.
- This paper works on the issue that if the noise in the original problem is in Gaussian distribution, VAMP could use non-separable denoiser including Group-Based Denoiser, Convolutional Denoiser, CNN and Singular-Value Thresholding (SVT) Denoiser.

$$\mathbf{g}_2(\mathbf{r}_{2k},\ \gamma_{2k}) := \left(\gamma_\omega \mathbf{A}^T\mathbf{A} + \gamma_{2k}\mathbf{I}\right)^{-1}\left(\gamma_{2k}\mathbf{A}^T\mathbf{y} + \gamma_{2k}\mathbf{r}_{2k}\right). \tag{2}$$

Recent reviews

Adversarial Regularizers in Inverse Problems [2]

- The problem is:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda f(\mathbf{x}). \tag{3}$$

- This paper proposes a method to learn a network $\Psi_{\mathbf{\Theta}}(\cdot)$ to replace the regularization f in (3).
- First, we need to have a fast method to predict the inverse directly.
  - Inspired by [4], the author use pseudo-inverse to calculate $\tilde{\mathbf{x}} = \mathbf{A}^*\mathbf{y}$.
  - The author claims that this inverse could be computed fast.
  - The set of ground truth is $\mathbb{P}_r$.
  - The set of observation is $\mathbb{P}_y$.
  - The set of pseudo-inverse is $\mathbb{P}_n$.
- In the testing phase, we could use the gradient descent based methods like:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha \nabla_{\mathbf{x}} \left(\|\mathbf{y} - \mathbf{A}\mathbf{x}^k\|_2^2 + \lambda \Psi_{\mathbf{\Theta}}(\mathbf{x}^k) \right). \tag{4}$$

Recent reviews

Adversarial Regularizers in Inverse Problems [2]

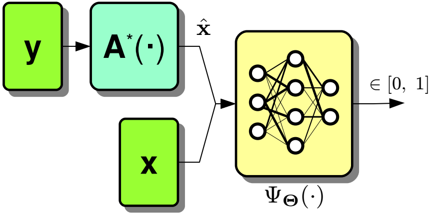

Figure 1: GAN structure for training $\Psi_{\Theta}(\cdot)$.

- The training phase could be viewed as training a GAN with fixed and predefined generator.
- The loss function is defined in (5).
- $\mathbf{x}_r \sim \mathbb{P}_r$, $\mathbf{x}_n \sim \mathbb{P}_n$ and $\mathbf{x}_i = \varepsilon\mathbf{x}_r + (1 - \varepsilon\mathbf{x}_n)$.
- The last term is used to preserve Lipschitz continuity.

$$\mathscr{L} = \Psi_{\Theta}(\mathbf{x}_r) - \Psi_{\Theta}(\mathbf{x}_n) + \mu \max\left(\|\nabla_{\mathbf{x}_i}\Psi_{\Theta}(\mathbf{x}_i)\|_2^2,\ 0\right)^2. \tag{5}$$

Inf-ADMM-ADNN

An inner-loop free solution to inverse problems ... [3]

- Considering a linear inverse problem in a generalized form:

$$\arg\min_{\mathbf{x}} \|\mathbf{y}-\mathbf{A}\mathbf{z}\| + \lambda\mathscr{R}(\mathbf{x},\ \mathbf{y}), \quad \text{s.t. } \mathbf{z}=\mathbf{x}. \tag{6}$$

- To solve this problem, we may need Lagrange multiplier method which decomposes the gradient descent into 3 steps in each iteration:

Lagrange multiplier method

$$\begin{cases} \mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \beta \left\| \mathbf{x} - \mathbf{z}^k + \dfrac{\mathbf{u}^k}{2\beta} \right\|^2 + \lambda\mathscr{R}(\mathbf{x},\ \mathbf{y}), & \text{(7-1)} \\ \mathbf{z}^{k+1} = \arg\min_{\mathbf{z}} \|\mathbf{y}-\mathbf{A}\mathbf{z}\|^2 + \beta \left\| \mathbf{x}^{k+1} - \mathbf{z} + \dfrac{\mathbf{u}^k}{2\beta} \right\|^2, & \text{(7-2)} \\ \mathbf{u}^{k+1} = \mathbf{u}^k + 2\beta\left(\mathbf{x}^{k+1} - \mathbf{z}^{k+1}\right) & \text{(7-3)} \end{cases}$$

Inf-ADMM-ADNN

An inner-loop free solution to inverse problems ... [3]

- The solution to (7-2) is $\mathbf{z}^{k+1} = \mathbf{K}\left(\mathbf{A}^T\mathbf{y} + \beta\mathbf{x}^{k+1} + \mathbf{u}^k/2\right)$.
- $\mathbf{K} = \left(\mathbf{A}^T\mathbf{A} + \beta\mathbf{I}\right)^{-1} = \beta^{-1}\left(\mathbf{I} - \mathbf{A}^T\mathbf{B}\mathbf{A}\right)$, where $\mathbf{B} = \left(\beta\mathbf{I} + \mathbf{A}\mathbf{A}^T\right)^{-1}$.

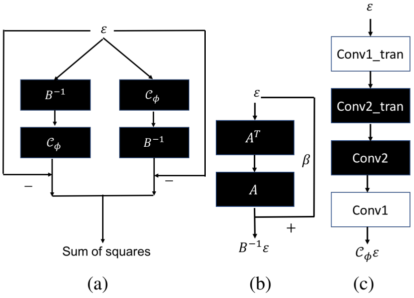

Figure 2: Learning the inverse of matrix **B**.

- Use a four-layer surrogate to learn $\mathbf{C}_\phi \rightarrow \mathbf{B}$.
- For any random variable ε, the training loss is $\mathscr{L} = \|\varepsilon - \mathbf{C}_\phi\mathbf{B}^{-1}\varepsilon\|_2^2 + \|\varepsilon - \mathbf{B}^{-1}\mathbf{C}_\phi\varepsilon\|_2^2$.

Inf-ADMM-ADNN

An inner-loop free solution to inverse problems ... [3]

- The solution to (7-1) may not be closed-form due to $\mathscr{R}(\cdot)$. But we still have a conclusion:
  1. This problem is essentially solving the proximal operator $\arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{v}\|_2^2 + \mathscr{R}(\mathbf{x}, \mathbf{y})$.
  2. The derivative shows that $\mathbf{v}-\mathbf{x} \propto \partial\mathscr{R}$.
  3. Then we could denote that $\mathbf{v} = \mathscr{F}(\mathbf{x})$.
  4. By using a network as the surrogate, we could solve the inverse $\mathbf{x} = \mathscr{F}^{-1}(\mathbf{v})$.

- We use an example-based GAN to solve the inverse of $\mathscr{F}$. The generator is a denoise network.

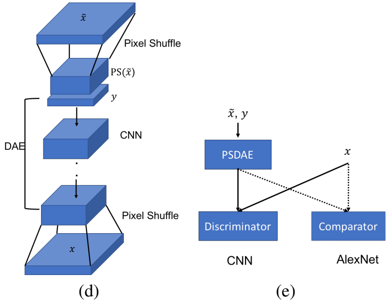

Figure 3: Learning the inverse of proximal operator.

Outline

Future Plan

How to feed offset into the network.

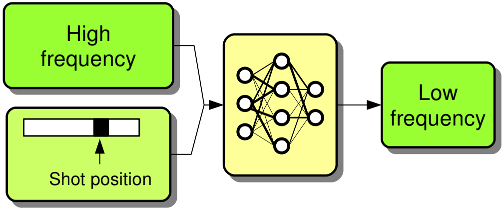

Figure 4: Incorporate offsets directly by one-hot vector.

- This method has been proved to be ineffective.
- In this method, we only give the position of the shot.

Future Plan

How to feed offset into the network.

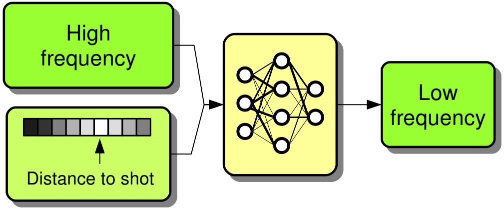

Figure 5: Incorporate offsets by distance to shot.

- This method is not expected to be effective.
- It is a similar approach. Dr. Hu has suggested me to do that, I would have a try.

Future Plan

How to feed offset into the network.

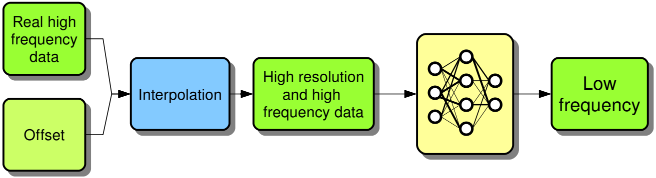

Figure 6: Adjust the signals by normalization.

- This method is expected to be effective, but it depends on pre- and post-processing.
- First, interpolate the signal into high-resolution.
- Second, let network predict high-resolution low-frequency data.
- Finaaly, downsample the prediction to the same resolution of input.

Future Plan

How to make use of derivative.

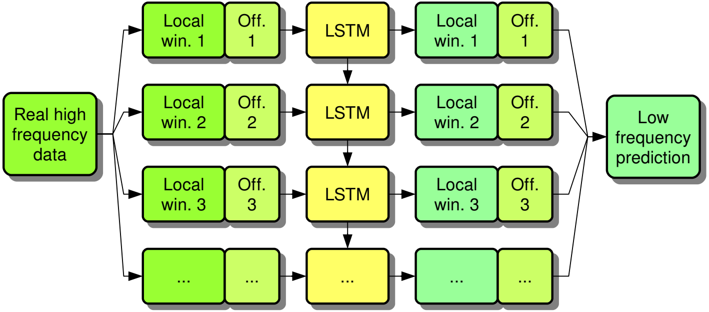

Figure 7: Incorporate offsets by window-wise LSTM.

- Extract the original signal into small windows. Each window is companied with a begin offset.
- Use local window to remove the sampling effect, and use LSTM to learn global feature.

Future Plan

How to make use of derivative.

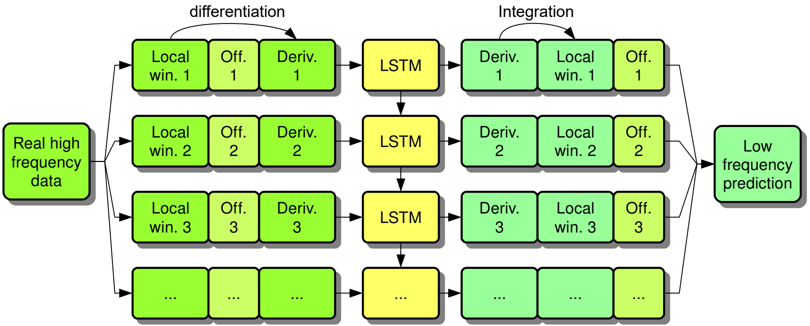

Figure 8: Incorporate offsets by window-wise LSTM and derivative calculation.

- Derivative is easy to calculate $\partial \mathbf{f}(\mathbf{x}) = {}^{\mathbf{f}(\mathbf{x}+1)-\mathbf{f}(\mathbf{x})}/_{\mathbf{Off}(\mathbf{x}+1)-\mathbf{Off}(\mathbf{x})}$.
- We use local begin value of the window to correct the deviation.
- The cumulative errors in the local window is small.

Outline

Reference I

- A. K. Fletcher, P. Pandit, S. Rangan, S. Sarkar, and P. Schniter, "Plug-in estimation in high-dimensional linear inverse problems: A rigorous analysis," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 7451–7460. [Online]. Available: http://papers.nips.cc/paper/7973-plug-in-estimation-in-high-dimensional-linear-inverse-problems-a-rigorous-analysis.pdf
- S. Lunz, C. Schoenlieb, and O. Öktem, "Adversarial regularizers in inverse problems," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 8516–8525. [Online]. Available: http://papers.nips.cc/paper/8070-adversarial-regularizers-in-inverse-problems.pdf

Reference II

- K. Fan, Q. Wei, L. Carin, and K. A. Heller, "An inner-loop free solution to inverse problems using deep neural networks," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 2370–2380. [Online]. Available: http://papers.nips.cc/paper/6831-an-inner-loop-free-solution-to-inverse-problems-using-deep-neural-networks.pdf
- K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, Sep. 2017.

Thank you for Listening

It's time for Q & A