Weekly Report I: recent paper review

YUCHEN JIN
Dept. of ECE
University of Houston

Outline

Introduction

- **Recent works**:
  - Finish two articles on my site.
    1. note20180824special: Derivation of LMA.
    2. note20181129special: Derivation of Lovasz extension [1].
  - Read three papers roughly.
    1. About conjugate gradient descent (propose a new coefficient) [2].
    2. About decentralized FrankWolfe algorithm [3].
    3. About an improvement of LISTA (faster convergence) [4].

Outline

1 Introduction

2 Reviews
- Derivation of LMA
- Proof of Lovasz extension
- Conjugate gradient descent
- Decentralized FrankWolfe algorithm
- Improved LISTA

3 Reference

Derivation of LMA

- Assume we have $g(\mathbf{f}(\mathbf{x}))$, where $\mathbf{z} = \mathbf{f}(\mathbf{x})$.

Expansion for function $g(\cdot)$.

$$g(\mathbf{z} + \Delta\mathbf{z}) - g(\mathbf{z}) \approx \nabla g^T(\mathbf{z})\Delta\mathbf{z} + \frac{1}{2}\Delta\mathbf{z}^T\mathbf{H}(g)\Delta\mathbf{z}, \tag{1}$$

- The problem is $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{f}(\mathbf{x})\|^2$.
- Let $\mathbf{z}$ get first-order expansion: $\Delta\mathbf{z} = \frac{\partial\mathbf{f}}{\partial\mathbf{x}}\Delta\mathbf{x} = \mathbf{J}(\mathbf{f})\Delta\mathbf{x}$.
- Finally we get $\frac{g(\mathbf{x}+\Delta\mathbf{x}) - g(\mathbf{x})}{\Delta\mathbf{x}} \approx \Delta\mathbf{x}^T\mathbf{J}^T\mathbf{J} - 2(\mathbf{y} - \mathbf{f}(\mathbf{x}))^T\mathbf{J} = 0.$.

Sub problem's params into (1).

$$\begin{aligned} g(\mathbf{z}) &= \|\mathbf{y} - \mathbf{z}\|^2, \\ \mathbf{z} &= \mathbf{f}(\mathbf{x}), \\ \nabla g^T(\mathbf{z}) &= -2(\mathbf{y} - \mathbf{z})^T, \\ \mathbf{H}(g) &= 2\mathbf{I}, \\ \mathbf{J}(\mathbf{f}) &:= \mathbf{J}, \end{aligned} \tag{2}$$

Proof of Lovasz extension

The Lovťasz-Softmax loss: A tractable surrogate … [1]

Algorithm 1 Gradient of the Jaccard loss extension $\overline{\Delta_{J_c}}$

Inputs: vector of errors $\boldsymbol{m}(c) \in \mathbb{R}_+^p$
class foreground pixels $\boldsymbol{\delta} = \{\boldsymbol{y}^* = c\} \in \{0,1\}^p$
Output: $\boldsymbol{g}(\boldsymbol{m})$ gradient of $\overline{\Delta_{J_c}}$ (Equation (9))

```
π ← decreasing sort permutation for m
δ_π ← (δ_{π_i})_{i∈[1,p]}
intersection ← sum(δ) − cumulative_sum(δ_π)
union ← sum(δ) + cumulative_sum(1 − δ_π)
g ← 1 − intersection/union
if p > 1 then
    g[2 : p] ← g[2 : p] − g[1 : p − 1]
end if
return g_{π^{-1}}
```

- The primal Jaccard index:
  $$\Delta_c(\hat{\mathbf{y}},\ \mathbf{y}^*) = \frac{|\mathbf{M}_c|}{|\{\mathbf{y}^* = c\} \cup \mathbf{M}_c|}.$$
- The Jaccard index in algorithm:
  $$\Delta_{c\text{Alg}} = \frac{\mathbf{S1}}{\sum(\delta) + \mathbf{S}(1-\delta)}.$$
- Lovasz extension:
  $\bar{\Delta}_c = \sum_{i=1}^{p} m_{\pi_i} g_{\pi_i}$.
- For the index (i) of δ, exist l,
  - When $i \leqslant l$, $m_{\pi_i} = 1,\ \Delta_c = \Delta_{c\text{Alg}}$;
  - When $i > l$, $m_{\pi_i} = 0,\ \Delta_c \neq \Delta_{c\text{Alg}}$.

Proof of Lovasz extension

The Lovťasz-Softmax loss: A tractable surrogate ... [1]

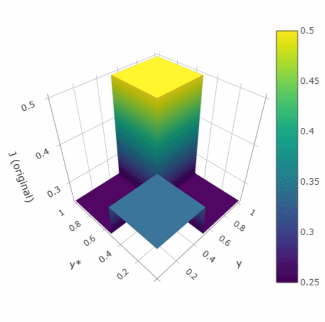


Figure: An example of the primal function of Δ.

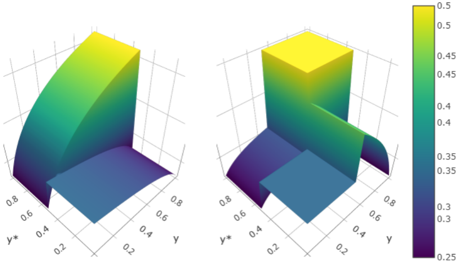


Figure: Compare the Lovasz extensions from algorithm and theory.

Conjugate gradient descent

A New Family of Conjugate Gradient Descent ... [2]

- To solve $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} f(\mathbf{x})$, the algorithm is:

  1. Initialize the input parameter $\mathbf{x}_0, k = 0$.
  2. Calculate first-order gradient $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$.
  3. Compute β_k which is the conjugate gradient coefficient.
  4. Update descent direction: when $k = 0$, let $\mathbf{d}_k = \mathbf{g}_k$; when $k > 0$, $\mathbf{d}_k = -\mathbf{g}_k + \beta_k \mathbf{d}_{k-1}$.
  5. Use line search to find the best update parameter: $\alpha_k = \arg\min_\alpha f(\mathbf{x}_k + \alpha \mathbf{d}_k)$.
  6. Let $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$. If $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$ and $\|\mathbf{g}_k\| < \varepsilon$, stop; otherwise get back to step 2.

- The author proposes a new coefficient that $\beta_k^{\mathrm{RMF}} = \frac{\mathbf{g}_k^T(\mathbf{g}_k - \mathbf{g}_{k-1})}{\|\mathbf{d}_{k-1}} \|^2$.
- Give the proof that $f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) \leqslant -\frac{1}{9L} \frac{(\mathbf{g}_k^T \mathbf{d}_k)^2}{\|\mathbf{d}_k\|}$, then we have $\lim_{k\to\infty} \|\mathbf{g}_k\| = 0$.

Decentralized FrankWolfe algorithm
Decentralized FrankWolfe Algorithm for Convex and Nonconvex Problems [3]

- To solve $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{D}} \sum_i f_i(\mathbf{x})$, suppose that $\sum_j W_{ij} = 1$ for each i, the algorithm is:
  1. For each agent, calculate the local average iterate among its neighbor: $\bar{\mathbf{x}}_i = \sum_j W_{ij}\mathbf{x}_j$, where W_{ij} is an element of the adjacent matrix.
  2. For each agent, calculate the local average gradient among its neighbor: $\overline{\nabla F}_i = \sum_j W_{ij}\nabla f_j(\mathbf{x}_j)$.
  3. Let $\boldsymbol{\alpha}_i = \arg\min_{\boldsymbol{\alpha}_i\in\mathcal{D}} \boldsymbol{\alpha}_i^T \overline{\nabla F}_i$.
  4. Update iterate: $\mathbf{x}_{i+1} = (1-\gamma)\bar{\mathbf{x}}_i + \gamma\boldsymbol{\alpha}_i$.

Decentralized FrankWolfe algorithm
Decentralized FrankWolfe Algorithm for Convex and Nonconvex Problems [3]

- Compare to conventional FW algorithm, DeFW could solve a combination of convex functions f_i with multiple agents and parallel computing.
- The author has proved that the local average with matrix **W** could serve as a global average by convergence. Denote the real average of any $\boldsymbol{\theta}_i$ is $\boldsymbol{\theta}_a$:
  - For any $\bar{\boldsymbol{\theta}}_i = \sum_j W_{ij}\boldsymbol{\theta}_j$, $\sum_{i=1}^N ||\bar{\boldsymbol{\theta}}_i - \boldsymbol{\theta}_a||^2 \leqslant |\lambda_2(\mathbf{W})|^2 \sum_{i=1}^N ||\boldsymbol{\theta}_i - \boldsymbol{\theta}_a||^2$.
  - $\lambda_2(\mathbf{W}) \leqslant \left(\frac{t_0(\alpha)}{t_0(\alpha)+1}\right)^\alpha \frac{1}{1+t_0^{-\alpha}(\alpha)}$, where λ_2 is the 2nd largest eigenvalue.
  - $\max_i||\bar{\mathbf{x}}_i - \mathbf{x}_a|| = \frac{1}{t^\alpha}C_p$, where $C_p = t_0^\alpha(\alpha)\sqrt{N}\bar{\rho}$.
  - $\max_i||\overline{\nabla F}_i - \overline{\nabla F}_a|| = \frac{1}{t^\alpha}t_0^\alpha(\alpha)2\sqrt{N}(2C_p + \bar{\rho})L$.

Improved LISTA

Theoretical Linear Convergence of Unfolded ... [4]

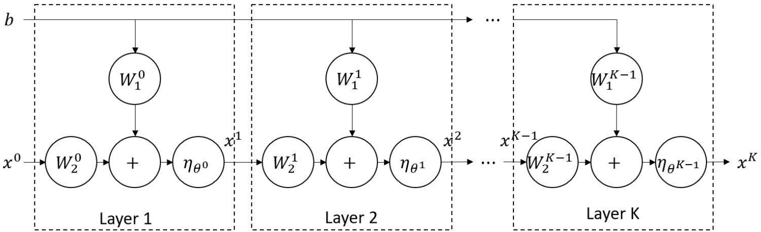


Figure: The LISTA network structure.

- The network is inspired from an algorithm, to learn more, please check 🌍: note20180813special: ISTA and AMP.
- ISTA is used to solve "sparse inverse problem": $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda|\mathbf{x}|$.
- The basic form of the layer is $\mathbf{x}_{k+1} = \eta(\mathbf{W}_{k1}\mathbf{b} + \mathbf{W}_{k2}\mathbf{x}_k, \theta_k)$
- $\eta(\cdot, \theta)$ is the soft thresholding function.

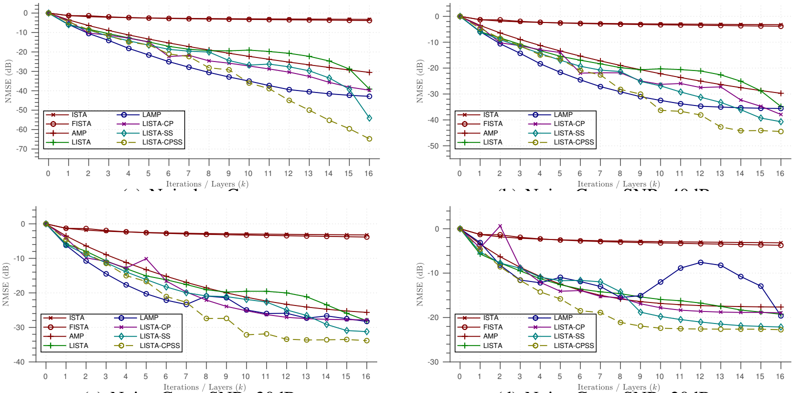
Improved LISTA

Theoretical Linear Convergence of Unfolded ... [4]

- The improvement for partial weight coupling (LISTA-CP):
  - The author prove that to let $\mathbf{x}_k$ converges, we need to have $\mathbf{W}_{k2} = \mathbf{I} - \mathbf{W}_{k1}\mathbf{A}$.
  - Thus the layer need to be adapted to $\mathbf{x}_{k+1} = \eta(\mathbf{x}_k \mathbf{W}_k^T(\mathbf{b} - \mathbf{A}\mathbf{x}_k), \theta_k)$.
- LISTA-CP could converge to $||\mathbf{x}_k - \mathbf{x}^*|| \leqslant sB\exp(ck) + C\sigma$.
- The improvement for support selection technique (LISTA-SS): The thresholding function would be adapted as $\eta_{ss}(\mathbf{v}_k,\ \theta_k,\ \mathbb{S}_k)$, where $\mathbf{v}_k = \mathbf{W}_{k1}\mathbf{b} + \mathbf{W}_{k2}\mathbf{x}_k$.
- First, sort all the values $\mathbf{v}_k$ that need to be threshold, the largest p_k values would remains in set $\mathbb{S}_k$.
  - $\mathbf{v}_k \in \mathbb{S}_k$, use hard thresholding as η_{ss}.
  - $\mathbf{v}_k \notin \mathbb{S}_k$, use soft thresholding as η_{ss}.
- LISTA-CPSS could converge to $||\mathbf{x}_k - \mathbf{x}^*|| \leqslant sB\exp(-\sum_{t=0}^{k-1} c_{ss}^t) + C_{ss}\sigma$, where $c_{ss}^t \geqslant c$ but $C_{ss} \leqslant C$.

Improved LISTA

Theoretical Linear Convergence of Unfolded ... [4]



(c) Noisy Case: SNR=30dB (d) Noisy Case: SNR=20dB

Figure: The performance of LISTA.

Outline

Reference I

- M. Berman and M. B. Blaschko, "Optimization of the jaccard index for image segmentation with the lovász hinge," *CoRR*, vol. abs/1705.08790, 2017. [Online]. Available: http://arxiv.org/abs/1705.08790
- M. Rivaie, M. Fauzi, and M. Mamat, "A new family of conjugate gradient methods for unconstrained optimization," in *2011 Fourth International Conference on Modeling, Simulation and Applied Optimization*, April 2011, pp. 1–4.
- H. Wai, J. Lafond, A. Scaglione, and E. Moulines, "Decentralized frankwolfe algorithm for convex and nonconvex problems," *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5522–5537, Nov 2017.
- X. Chen, J. Liu, Z. Wang, and W. Yin, "Theoretical linear convergence of unfolded ista and its practical weights and thresholds," 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. 9079–9089.

Thank you for listening!

IT'S TIME FOR Q&A.