AMATH 840: Advanced Numerical Methods for Computational and Data Sciences

Giang Tran

Department of Applied Mathematics, University of Waterloo

Jan 26, 2021

Lecture 09: ℓ_1-Optimization

$\min_x g(x) + h(x) \to$ non-diff

Method of Multipliers / ADMM / Split Bregman

constrained $\to$ unconstrained

$|\nabla x|$

Primal-Dual Alg.

Wrap up Topic 1

CS - Sparse Opt

2 Topic 2: Neural Networks

Fully Connected NN

Universal Approximations : Shallow Network

Recall: Proximal Gradient Algorithm

- Consider

$$\min_{x\in\mathbb{R}^n} g(x) + h(x),$$

  where

  - $g(x)$ is convex, differentiable
  - $h(x)$ is convex, (possibly non-differentiable), with an inexpensive proximal mapping.

- Proximal gradient algorithm: Initialization $x_0 \in \mathbb{R}^n$ from infeasible set

$$(k+1)^{\text{th}} \text{ iterate } \to x_{k+1} = \operatorname{prox}_{t_k h}\left(\underline{x_k - t_k \nabla g(x_k)}\right), \quad \|$$

  where $t_k > 0$ is the step size.

- The proximal mapping of a convex function h is

$$\operatorname{prox}_h(x) := \operatorname*{argmin}_u \left(h(u) + \frac{1}{2}\|u - x\|_2^2 \right).$$

- For $h(x) = \lambda\|x\|_1$ with $\lambda > 0$, prox_h is the shrink operator (component-wise):

$$\operatorname{prox}_h(x) = \operatorname{sign}(x)\max(|x| - \lambda, 0). = \begin{cases} x - \lambda, & \text{if } x > \lambda \\ x + \lambda, & \text{if } x < -\lambda \\ 0, & |x| \le \lambda \end{cases}$$

Proximal Gradient Algorithm (cont'd)

Theorem 1: Consider

$$\min_{x \in \mathbb{R}^n} f(x) = g(x) + h(x).$$

Assume:

- h is convex and closed (so that prox_{th} is well-defined)
- g is differentiable with $dom(g) = \mathbb{R}^n$ and g is L-smooth: = L-Lipschitz of ∇g

$$g(y) \leq g(x) + \underbrace{\nabla g(x)^T (y-x)}_{\text{Taylor series expansion}} + \frac{L}{2}\|y-x\|_2^2.$$

- There exists a constant $m \geq 0$ such that

$$g(y) \geq g(x) + \nabla g(x)^T (y-x) + \frac{m}{2}\|y-x\|_2^2.$$

  Note: if $m = 0$, this means g is convex; if $m > 0$, this means g is strongly convex.

- The optimal value f^* is finite and attained at some x^* (may not be unique).

Proximal Gradient Algorithm (cont'd)

m, L

$\min f(x) = g(x) + h(x)$

Theorem 1 (cont'd). Then with fixed step size $t_k = 1/L$, we have:

1. Each proximal gradient iteration is a descent step:

$$f(x_{k+1}) < f(x_k), \quad \|x_k - x^*\|_2^2 \leq c^k \|x_0 - x^*\|_2^2,$$

   where $c = 1 - \frac{m}{L}$. If $m > 0$, $c < 1 \Rightarrow x_k \xrightarrow[k\to\infty]{} x^*$

2. Convergence rate of proximal gradient method is $\mathcal{O}(1/k)$:

$$f(x_k) - f^* \leq \frac{L}{2k} \|x_0 - x^*\|_2^2.$$

[1]

[1] http://www.seas.ucla.edu/~vandenbe/236C/lectures/proxgrad.pdf

Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)

- Consider

$\min \frac{1}{2}\|Ax-b\|^2 + \gamma \|x\|_1$

$$\min_{x \in \mathbb{R}^n} g(x) + h(x),$$

where

  - $g(x)$ is convex, differentiable
  - $h(x)$ is convex, (possibly non-differentiable), with an inexpensive proximal mapping.

- FISTA[2]: an accelerated proximal gradient method.

$$y_1 = x_0, \ \alpha_1 = 1$$

$$x_k = \mathrm{prox}_{t_k h}(y_k - t_k \nabla g(y_k)), \quad k \geq 1$$

$$\alpha_{k+1} = \frac{1}{2}\left(1 + \sqrt{4\alpha_k^2 + 1}\right)$$

$$y_{k+1} = x_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(x_k - x_{k-1})$$

$\frac{k-1}{k+p}$, $p \geq 2$

- Convergent rate of FISTA: Under the same assumptions as Theorem 1,

$$f(x_k) - f^* \leq \frac{2L}{(k+1)^2}\|x_0 - x^*\|_2^2, \quad \forall k \geq 1.$$

$O\left(\frac{1}{k^2}\right)$

[2] "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems", by Beck & Teboulle

Another ℓ_1-Optimization Package: FASTA

Remark:

- The optimal convergent rate we can get from first-order gradient methods (using only the gradient, under the same assumptions as Theorem 1) is $\mathcal{O}(1/k^2)$.
- A review of other variants of (accelerated) proximal gradient methods: "A Field Guide to Forward-Backward Splitting with a FASTA Implementation", by Goldstein, Studer, & Baraniuk.
- Matlab package: FASTA (http://www.cs.umd.edu/~tomg/projects/fasta/) / SPGL1
  - Contains implementations of many variants of proximal gradient methods (such as FISTA, SpaRSA)
  - Automatically handle stepsize selection, acceleration, and stopping conditions.

From Constrained to Unconstrained Optimization Problem.

- Method of Multipliers

① Consider : $\boxed{\min_{x \in \mathbb{R}^n,\, y \in \mathbb{R}^m} f(x) + g(y) \quad \text{s.t } Ax + By + c = 0}$

where $A \in \mathbb{R}^{d \times n}$, $B \in \mathbb{R}^{d \times m}$, $c \in \mathbb{R}^d$.

② Examples:

2.1 $\min_{x \in \mathbb{R}^{n}} \frac{1}{2} \|Ax - b\|_2^2 + \gamma \|\underset{\substack{\uparrow \\ y}}{\tilde{x}}\|_1$

$\Leftrightarrow \min_{x, y \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2 + \gamma \|y\|_1 \quad \text{s.t. } x - y = 0$

$$\boxed{\min_{u\in\mathbb{R}^{n\times n}} \frac{1}{2}\|Au-f\|_2^2 + \gamma\left(|\nabla_x u| + |\nabla_y u|\right)}$$

(2.2)

$\|\nabla u\|$

TV Denoising / Deblurring

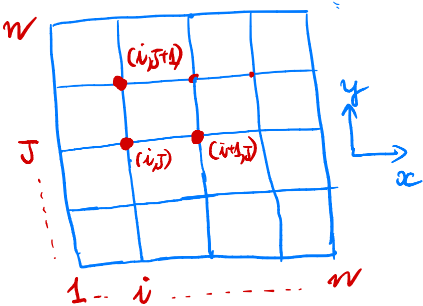
$$\Longleftrightarrow \min_{u, d_x, d_y} \frac{1}{2}\|Au-f\|^2 + \gamma\left(|d_x| + |d_y|\right)$$

subject to

$$\begin{cases} \nabla_x u = d_x \\ \nabla_y u = d_y \end{cases}$$

where $\nabla_x u(i,j) := u(i+1,j) - u(i,j)$

$\nabla_y u(i,j) := u(i,j+1) - u(i,j)$



2.3
$$\min_{u \in \mathbb{R}^{n \times n}} \frac{1}{2}\|Au - f\|_2^2 + \gamma \sum_{i,j} \underbrace{\sqrt{|\nabla_x u(i,j)|^2 + |\nabla_y u(i,j)|^2}}_{\|\nabla u\|_1}$$

$$\Longleftrightarrow \min \frac{1}{2}\|Au - f\|_2^2 + \lambda \, \|(d_x, d_y)\|_2$$

such that
$$d_x(i,j) = \nabla_x u(i,j)$$
$$d_y(i,j) = \nabla_y u(i,j)$$

Here
$$\|(d_x, d_y)\|_2 = \sum_{i,j} \sqrt{d_x^2(i,j) + d_y^2(i,j)}$$

③ Method of multipliers = Backward Gradient

Problem
$$\min_{x\in\mathbb{R}^n,\, y\in\mathbb{R}^m} f(x) + g(y) \quad \text{s.t} \quad Ax + By + c = 0.$$

Step 1: The corresponding augmented Lagrangian form is

no requirement about λ

$$L_\tau(x,y,\lambda) := f(x) + g(y) + \langle \lambda, Ax + By + c\rangle + \frac{\tau}{2}\|Ax + By + c\|_2^2$$

Step 2:
$$\begin{cases} x_{k+1}, y_{k+1} = \operatorname{argmin} L_\tau(x, y, \lambda_k) \\ \lambda_{k+1} = \lambda_k + \tau(Ax_{k+1} + By_{k+1} + c) \end{cases}$$

To solve step 2, we use alternating direction method of multipliers (ADMM):

$$x_{k+1} = \arg\min_{x} f(x) + \langle \lambda_k, Ax \rangle + \frac{\tau}{2} \| Ax + By_k + c \|_2^2$$

$$y_{k+1} = \arg\min_{y} g(y) + \langle \lambda_k, By \rangle + \frac{\tau}{2} \| Ax_{k+1} + By + c \|_2^2$$

$$\lambda_{k+1} = \lambda_k + Ax_{k+1} + By_{k+1} + c$$ ← add error back

④ Go back to Example 2.1.

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Ax-b\|_2^2 + \gamma \underbrace{\|x\|_1}_{=y}$$

$$\Leftrightarrow \min_{x,y\in\mathbb{R}^n} \frac{1}{2}\|Ax-b\|_2^2 + \gamma\|y\|_1 \quad \text{s.t} \quad x-y=0$$

Step 1 The augmented Lagrangian form is

$$L_\tau(x,y,\lambda) := \frac{1}{2}\|Ax-b\|_2^2 + \gamma\|y\|_1 + \langle\lambda, x-y\rangle + \frac{\tau}{2}\|x-y\|_2^2$$

Step2 : ADMM :

$$x_{k+1} = \underset{x}{\operatorname{argmin}} \frac{1}{2}\|Ax-b\|_2^2 + \langle \lambda_k , x\rangle + \frac{\tau}{2}\|x-y_k\|_2^2$$

$$A^T(Ax-b) + \lambda_k + \tau(x-y_k) = 0$$

$$(A^TA + \underline{\tau\, Id})\, x = A^Tb - \lambda_k + \tau y_k$$

$$y_{k+1} = \operatorname{argmin} \underline{\gamma\|y\|_1} - \underbrace{\langle \lambda_k , y\rangle + \frac{\tau}{2}\|x_{k+1} - y\|_2^2}$$

$$\operatorname{prox}_{\gamma\|.\|_1}(\cdot) = \operatorname{argmin} \gamma\|y\|_1 + \frac{\tau}{2}\left\| y - x_{k+1} - \frac{\lambda_k}{\tau}\right\|_2^2$$

$$\lambda_{k+1} = \lambda_k + \tau(x_{k+1} - y_{k+1})$$

⑤ Fast ADMM := ADMM + FISTA. $O\left(\frac{1}{(k+2)^2}\right)$

- $\min_{x,y} f(x) + g(y)$ s.t $Ax + By + c = 0$
- $L_\tau(x,y,\lambda) = f(x) + g(y) + \langle \lambda, Ax + By + c\rangle + \frac{\tau}{2}\|Ax + By + c\|_2^2$
- $x_k = \underset{x}{\operatorname{argmin}} f(x) + \langle \hat{\lambda}_k, Ax\rangle + \frac{\tau}{2}\|Ax + B\hat{y}_k + c\|_2^2$

$y_k = \underset{y}{\operatorname{argmin}} g(y) + \langle \hat{\lambda}_k, By\rangle + \frac{\tau}{2}\|A\hat{x}_k + By + c\|_2^2$

$\lambda_k = \hat{\lambda}_k + \tau(Ax_k + By_k + c)$

$\alpha_{k+1} = \frac{1 + \sqrt{1 + 4\alpha_k^2}}{2}$; $\hat{y}_{k+1} = y_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(y_k - y_{k-1})$

$\hat{\lambda}_{k+1} = \lambda_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(\lambda_k - \lambda_{k-1})$

Summary Compressive Sensing & Sparse Optimization.

① Compressive Sensing : Solve $\min_{z \in \mathbb{C}^n} \|z\|_0$ s.t. $y = Az$. where $A \in \mathbb{C}^{m \times n}$.

<u>Essential</u>

① $w \in \mathbb{C}^n$ is sparse or compressible

↑ solution

($\sigma_s(w)_1$ is small)

② Randomness in A → Gaussian | Random Bernoulli | Matrix

→ from bounded orthonormal system

③ Limited Data : $m \ll n$, $m = O(s \log(\frac{N}{s}))$

② Greedy Algorithms: OMP, IHT, HTP

③ ℓ_1-minimization : FISTA, Nesterov's, ADMM, SPGL1

fast algorithm $O(\frac{1}{k^2})$

④ Reconstruction Guarantees using

coherence

null space property

$\Uparrow$

RIP

Error Estimation

⑤ Applications

$$\min_z \|z\|_1 \quad \text{s.t} \quad y = Az$$

See chapter 1

Foucart & Rauhut.

$$\min_z \|z\|_1 \quad \text{s.t} \quad \|y - Az\|_2 \le \eta$$

$$\min_z \lambda \|z\|_1 + \frac{1}{2} \|Az - y\|_2^2$$

Sparse Optimization & PDE

$$(\sqrt{u})' = \frac{1}{2\sqrt{u}} \qquad \frac{1}{2\sqrt{u+\varepsilon}}$$

Obstacle problem

$$\begin{cases} \min\limits_{u} \frac{1}{2}\int_{\Omega} |\nabla u|^2 \, dx \\ u = g \text{ on } \partial\Omega \\ u \geq \varphi \text{, where } \varphi : \Omega \to \mathbb{R} \text{ is a given smooth function} \end{cases}$$

when μ is large enough

$$\underbrace{\min \frac{1}{2}\int |\nabla u|^2 + \mu(\underbrace{\varphi - u}_{= v})_+ \, dx}_{\text{solve by ADMM}}$$

See the attached slides.