THE UNIVERSITY OF TEXAS AT AUSTIN
Department of Statistics and Data Sciences
College of Natural Sciences

SDS 385: Stat Models for Big Data

Lecture 5: Proximal methods

Purnamrita Sarkar
Department of Statistics and Data Science
The University of Texas at Austin

https://psarkar.github.io/teaching

Proximal methods

- You want to minimize functions of the form

$$f(x) = \underbrace{g(x)}_{convex, differentiable} + \underbrace{h(x)}_{convex, nonsmooth}$$

- If h was differentiable, we would use

$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

- Here we would use:

$$x_{k+1} = \arg\min_z \frac{1}{2\alpha} \underbrace{\|z - (x_t - \alpha \nabla g(x_t))\|^2}_{\text{Stay close to the gradient}} + \underbrace{h(z)}_{\text{minimize h}}$$

Proximal mapping

- Define:

$$\text{prox}_{\alpha}(x) = \arg\min_{z} \frac{1}{2\alpha}\|x - z\|^2 + h(z)$$

- Proximal GD:
  - Choose initial $x^{(0)}$
  - Repeat, for $k = 1, 2, 3$

$$x_{k+1} = \text{prox}_{\alpha_k}(x_k - \alpha_k \nabla g(x_k))$$

- But, we just turned one minimization into another. And both has h which is the troublesome part.

Example: Lasso

$$f(\beta) = \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1$$

- The proximal map is:

$$\begin{aligned}
\text{prox}_\alpha(\beta) &= \arg\min_z \left(\frac{1}{2\alpha}\|\beta - z\|^2 + \lambda\|z\|_1 \right) \\
&= S_{\lambda\alpha}(\beta) \\
[\text{prox}_\alpha(\beta)]_i &= \begin{cases} \beta_i - \lambda\alpha & \text{If } \beta_i > \lambda\alpha \\ 0 & \text{If } |\beta_i| \leq \lambda\alpha \\ \beta_i + \lambda\alpha & \text{If } \beta_i < -\lambda\alpha \end{cases}
\end{aligned}$$

Lasso

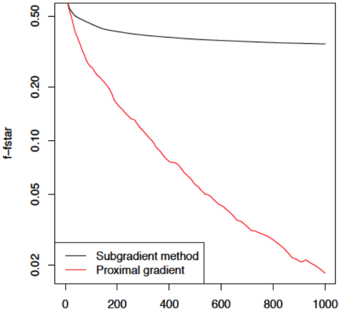
- In this case, the gradient is

$$\nabla g(\beta) = -X^T(y - X\beta)$$

- So the update step for Lasso becomes:

$$\beta_{k+1} = S_{\lambda\alpha}\left(\beta_k + \alpha X^T(y - X\beta)\right)$$

- This is the Iterative Soft Thresholding Algorithm.



Example: matrix completion

Given a matrix $Y \in \mathbb{R}^{m \times n}$ and observed entries $(i, j) \in \Omega$, you want to fill missing entries by solving:

$$\min_{B \in \mathbb{R}^{m \times n}} \frac{1}{2} \sum_{ij \in \Omega} (Y_{ij} - B_{ij})^2 + \lambda \|B\|_*$$

- $\|B\|_*$ is the nuclear norm of B, defined as:

  $$\|B\|_* = \sum_{i=1}^{k} \sigma_i(B),$$

  where k is the rank of B and $\sigma_1(B) \geq \sigma_2(B) \dots$ are the singular values.
- Nuclear norm is a convex approximation of rank, think how you cannot easily minimize ℓ_0 norm, aka the number of nonzero entries, an instead minimize the ℓ_1 norm to induce sparsity in regression problems.

Proximal gradient

- $$[P_\Omega(B)]_{ij} = B_{ij} 1((ij) \in \Omega)$$
- So the optimization can also be written as:

$$\min \frac{1}{2} \|P_\Omega(Y) - P_\Omega(B)\|_F^2 + \lambda \|B\|_*$$

- Gradient of smooth first part: $-(P_\Omega(Y) - P_\Omega(B))$
- Prox function:

$$\text{prox}_\alpha(B) = \arg \min_{Z \in \mathbb{R}^{m \times n}} \frac{1}{2\alpha} \|B - Z\|_F^2 + \lambda \|Z\|_*$$

Proximal GD

- We will show that $\text{prox}_\alpha(B) = S_\alpha(B)$, where
- $S_\alpha(B)$ is $U\Sigma_\alpha V^T$, where $B = U\Sigma V^T$ and

$$\Sigma_\alpha(i,i) = \max(\Sigma_{ii} - \alpha, 0)$$

- First, it is known that the subdifferential of the nuclear norm is given by: $\partial\|Z\|_* = \{UV^T + W : \|W\| \leq 1, U^T W = 0, WV = 0\}$, where $U \in \mathbb{R}^{m\times r}, V \in \mathbb{R}^{n\times r}$ where $Z = U\Sigma V^T$ where Σ contains the nonzero singular values of Z.
- Now we will show that $0 \in S_{\alpha\lambda}(B) - B + \lambda\alpha\partial\|S_{\alpha\lambda}(B)\|_*$

Proximal GD

- Take U_0, V_0 as the singular vectors corresponding to $\sigma_i(B) > \lambda\alpha =: t$.
- Take the remaining singular vectors as $U_\perp, V_\perp$ and the corresponding singular value matrix as $\Sigma_\perp$
- $S_t(B) - B = -tU_0V_0^T - U_\perp\Sigma_\perp V_\perp^T$
- $S_t(B) - B + t(U_0V_0^T + W) = tW - U_\perp\Sigma_\perp V_\perp^T$
- Taking $W = U_\perp\Sigma_\perp V_\perp^T/t$, we see that
  - $U^TW = 0$
  - $WV = 0$
  - $\|W\| \leq 1$

Proximal GD

- $B_{k+1} = S_{\lambda\alpha}(B + t(P_\Omega(Y) - P_\Omega(B))$
- This is called the Soft Impute algorithm.
  - Cai et al, "A Singular Value Thresholding Algorithm for Matrix Completion", 2010.
  - Mazumdar et al 2011, "Spectral regularization algorithms for learning large incomplete matrices"

Acknowledgment

Cai et al, ”A Singular Value Thresholding Algorithm for Matrix Completion”, 2010.