Assignment 1 - Group 21

Abhas Ankit - 150008

Aditya Pratap Singh Rajawat - 15807053

Harsh Surana - 15917268

Mohit Yadav - 170399

Prashant Kumar - 15917514

Priyadarshini Agrawal - 15817528

1 Given formulation

We are given this formulation for a real-valued regression problem involving n labelled data points $(x^i, y^i)_{i=1,\ldots,n}$ where $x^i \in \mathbb{R}^d$ and $y^i \in \mathbb{R}$.

$$\arg\min_{\mathbf{w}\in\mathbb{R}^d} \|\mathbf{w}\|_1 + \sum_{i=1}^{n} \left(y^i - \left\langle \mathbf{w}, \mathbf{x}^i \right\rangle\right)^2 \quad (P1)$$

Taking $X = \left[\mathrm{x}^1, \ldots, \mathrm{x}^n\right]^\top \in \mathbb{R}^{n\times d}$ and $\mathrm{y} = \left[y^1, \ldots, y^n\right]^\top \in \mathbb{R}^n$, (P1) can be rewritten more compactly as

$$\arg\min_{\mathbf{w}\in\mathbb{R}^d} \|\mathbf{w}\|_1 + \|X\mathbf{w} - \mathbf{y}\|_2^2$$

We have used Primal SubGradient Descent, Primal Proximal Descent methods and Primal Co-Ordinate Descent but the results for Primal Subgradient Descent and Primal Proximal Subgradient Descent were better for this dataset as they converged faster.

1.1 Primal Subgradient Descent

We used this method on the primal problem P1.

The expression for the gradient is:

$$\nabla_w f(w) = 2X^T.(Xw - y) + \mathrm{sign}(w)$$

Then using the following algorithm :

(SUB) GRADIENT DESCENT (reference from class slides)

- Given: obj. func. $f : \mathbb{R}^d \to \mathbb{R}$ to minimize
- Initialize $\mathbf{w}^0 \in \mathbb{R}^d$

Preprint. Under review.

- For $t = 0, 1, \ldots$
  1. Obtain a (sub)gradient $\mathbf{g}^t \in \partial f\left(\mathbf{w}^t\right)$
  2. Choose a step length η_t
  3. Update $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t - \eta_t \cdot \mathbf{g}^t$
  4. Repeat until convergence

The step length used $\eta = 0.12/t^{1/2}$

1.2 Primal Proximal Gradient Descent

We used this method on the primal problem P1.

$$f(w) = L(w) + R(w)$$

where

$$L(w) = \|X\mathbf{w} - \mathbf{y}\|_2^2$$
$$R(w) = \|\mathbf{w}\|_1$$

The expression for the gradient of Loss function is:

$$\mathbf{g}^t = \nabla_w L(w) = 2X^T.(Xw - y)$$

Let us compute the proximity operator for $R(w)$. First we find an alternative characterization of the proximity operator $\text{prox}_R(x)$ as follows:

$$\begin{aligned} u = \text{prox}_{\eta_t R}(x) &\iff 0 \in \partial\left(R(u) + \frac{1}{2\eta_t}\|u - x\|_2^2\right) \\ &\iff 0 \in \partial R(u) + \frac{1}{\eta_t}(u - x) \\ &\iff \frac{1}{\eta_t}(x - u) \in \partial R(u) \end{aligned}$$

For $R(w) = \|w\|_1$ it is easy to compute $\partial R(w)$: the i th entry of $\partial R(w)$ is precisely

$$\partial |w_i| = \begin{cases} 1, & w_i > 0 \\ -1, & w_i < 0 \\ [-1, 1], & w_i = 0 \end{cases}$$

Using the above condition for optimality, for $R(w) = \|w\|_1$ and $\eta_t > 0$ we have that $\text{prox}_{\eta_t R}(x)$ is defined entrywise by

$$\left(\text{prox}_{\eta_t R}(x)\right)_i = \begin{cases} x_i - \eta_t, & x_i > \eta_t \\ 0, & |x_i| \leq \eta_t \\ x_i + \eta_t, & x_i < -\eta_t \end{cases}$$

which is known as the soft thresholding operator $S_{\eta_t}(x) = \text{prox}_{\eta_t|\cdot|_1}(x)$

We have used the following algorithm for the proximal gradient descent method.

PROXIMAL GRADIENT DESCENT (reference from class slides)

- Given: loss fn $\ell(\cdot)$ regularizer $r(\cdot)$
- Initialize $\mathbf{w}^0 \in \mathbb{R}^d$
- For $t = 0, 1, \ldots$
  a. Let $\mathbf{g}^t \in \partial\ell\left(\mathbf{w}^t\right)$ and choose η_t
  b. Let $\mathbf{u}^{t+1} \leftarrow \mathbf{w}^t - \eta_t \cdot \mathbf{g}^t$
  c. Let $\mathbf{w}^{t+1} \leftarrow \text{prox}_{\eta_t R}\left(\mathbf{u}^{t+1}\right)$
  d. Repeat until convergence

1.3 Primal Coordinate Descent

Formula for the gradient derivation ($\lambda = 1$):

$$f\left(\mathbf{w}^t\right) = \mathrm{RSS}(\mathrm{w}^{\mathrm{t}}) + \lambda\|\mathrm{w}^{\mathrm{t}}\|_1 = \sum_{i=1}^{N}\left(y_i - \sum_{j=1}^{D} w_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{D}|w_j|$$

$$\partial_{w_j} RSS(\mathrm{w}^{\mathrm{t}}) = -2\sum_{i=1}^{N}(x_{ij})\left(y_i - \sum_{k\neq j} w_k x_{ik})\right) + 2w_j\sum_{j=1}^{N} x_{ij}^2$$

where we denote $\rho_j = \sum_{i=1}^{N}(x_{ik})\left(y_i - \sum_{k\neq j} w_k x_{ik})\right)$ and $z_j = \sum_{j=1}^{N} x_{ij}^2$

Subgradient of the L1 term:

$$\partial_{w_j}|w_j| = \begin{cases} -1 & \text{when } w_j < 0 \\ [-1,1] & \text{when } w_j = 0 \\ 1 & \text{when } w_j > 0 \end{cases}$$

Subgradient of the function:

$$\partial_{w_j}[f(w^t)] = 2z_j w_j - 2\rho_j + \begin{cases} -1 & \text{when } w_j < 0 \\ [-1,1] & \text{when } w_j = 0 \\ 1 & \text{when } w_j > 0 \end{cases}$$

Optimal solution is obtained when $0 \in \partial_{w_j}[f(w^t)]$:

$$\partial_{w_j}[f(w^t)\,] = \begin{cases} 2z_j w_j - 2\rho_j - 1 & \text{when } w_j < 0 \\ 2\rho_j - 1, -2\rho_j + 1] & \text{when } w_j = 0 \\ 2z_j w_j - 2\rho_j + 1 & \text{when } w_j > 0 \end{cases}$$

$$\hat{w}_j = \begin{cases} (\rho_j + 1/2)/z_j & \text{if } \rho_j < -1/2 \\ 0 & \text{if } \rho_j \text{ in } [-1/2, 1/2] \\ (\rho_j - 1/2)/z_j & \text{if } \rho_j > 1/2 \end{cases}$$

We have used the following algorithm for the coordinate descent method (coordinate minimization as we are fully optimizing along one coordinate):

COORDINATE DESCENT/MINIMIZATION Algorithm used

- For $t = 0, 1, \ldots$
- Select a coordinate j_t cyclically and calculate corresponding value of ρ_j and $\mathrm{z}_j \in [d]$
- Let $\mathbf{u}_j^{t+1} \leftarrow \begin{cases} (\rho_j + 1/2)/z_j & \text{if } \rho_j < -1/2 \\ 0 & \text{if } \rho_j \text{ in } [-1/2, 1/2] \\ (\rho_j - 1/2)/z_j & \text{if } \rho_j > 1/2 \end{cases}$
- Let $\mathbf{u}_j^{t+1} \leftarrow \mathbf{x}_j^t$ for $j \neq j_t$
- Repeat until convergence

2 Determining hyper parameters

For the methods we have applied we have, $\eta(t)$ as hyperparameter for Primal Proximal Gradient Descent and Primal Subgradient Descent and type of coordinate selection for Coordinate descent.

2.1 Primal Proximal Gradient Descent

As mentioned in the code we checked for three functions of $\eta(t)$- "linear","quadratic","constant".

Determining optimal $\eta(t)$ has two parts to it- Deciding the form of function i.e linear/quadratic/constant and the corresponding optimal values for constant η.

We applied 10 fold cross validation for determining optimal form of $\eta(t)$ by using 100 values for η between [.01,0.5] and checked for which form of $\eta(t)$ produced minimum value of mean squared error on validation set. .

For Proximal Gradient Descent $\eta(t)$=.089 produced the minimum value of mean squared error on validation set and also produced good results when Proximal Gradient Descent is applied on the whole training data set.

2.2 Primal Sub Gradient Descent

We applied 10 fold cross validation for determining optimal form of $\eta(t)$ by using 100 values for η between [.01,0.5] which was selected through initial trials and checked for which form of $\eta(t)$ produced minimum value of mean squared error on validation set. .

For Primal Sub-Gradient Descent $\eta(t) = .12/t^{1/2}$ produced the minimum value of mean squared error on validation set and also produced good results when Sub Gradient Descent is applied on the whole training data set.

2.3 Primal Co-ordinate Descent

We used cyclic and random coordinate selection methods and found cyclic to work best using 10-fold cross validation.

3 Plots

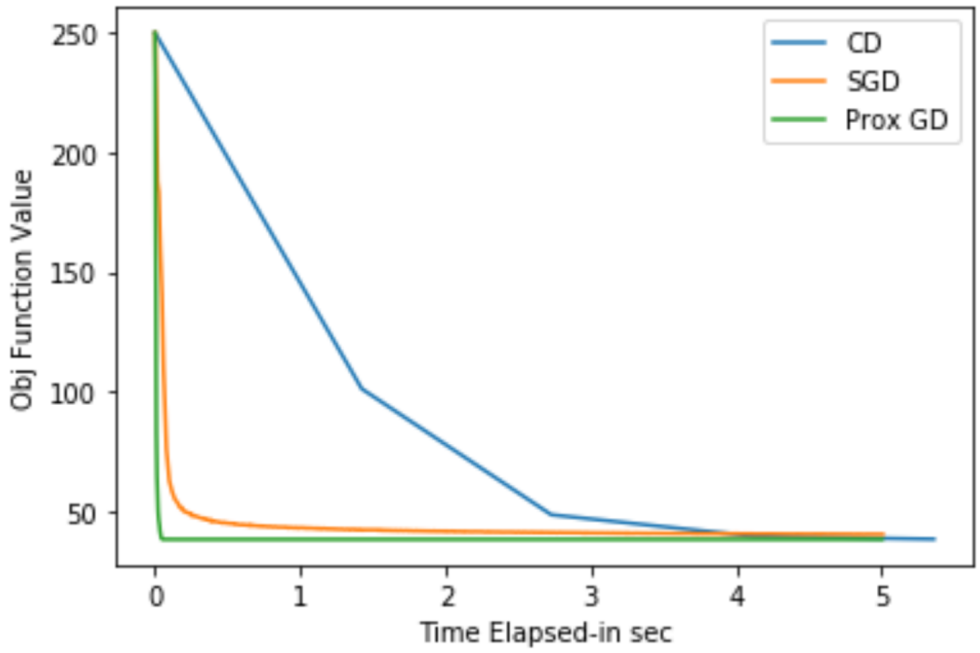

Figure 1: Proximal GD converges the fastest i.e < .1 sec while CD takes the most time to converge(>4 sec). Subgradient Descent converges faster than CD but slower than Proximal GD

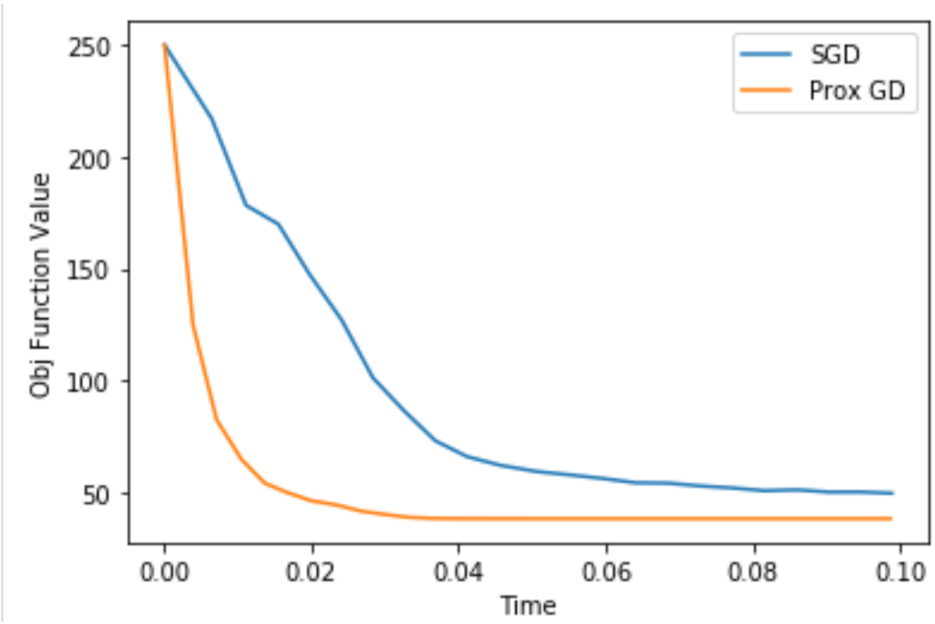

Figure 2: The graph shows the decay in objective value for SGD and Proximal GD at $timeout = 0.1sec$. Proximal GD works much faster and better than SGD

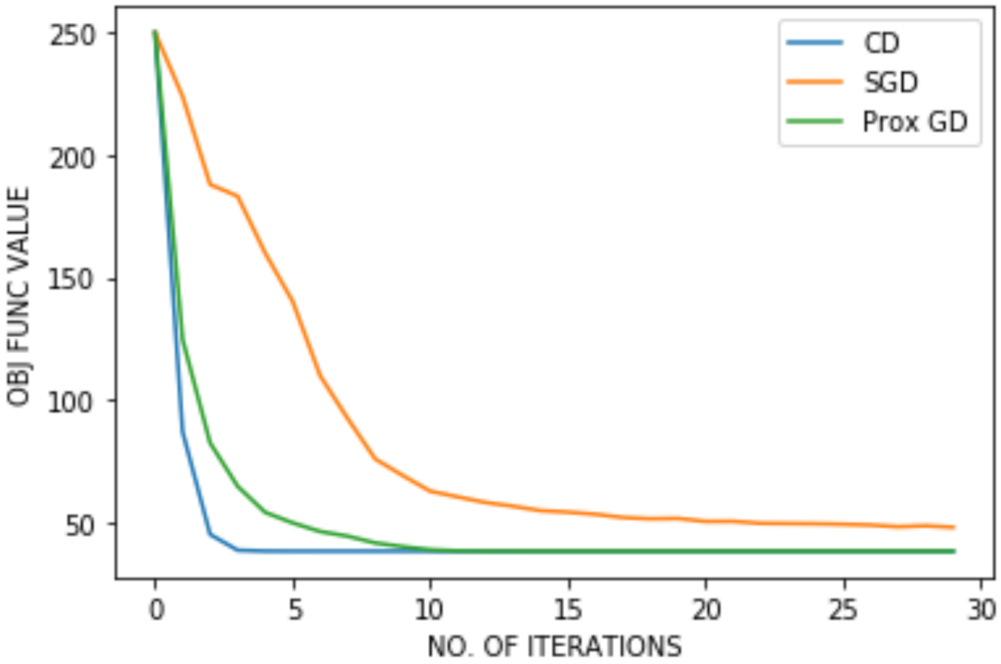

Figure 3: The graph shows the decay of Objective Value Funtion wrt no. of iterations. Interestingly Coordinate Descent which works slowest wrt time elapsed performs better than Prox GD and SGD by taking less no iterations to converge.
Note: One iteration for CD is counted when all the coordinates get updated once.

References

[1]https://www.stat.cmu.edu/~ryantibs/convexopt-S15/lectures/08-prox-grad.pdf

[2]https://www.stat.cmu.edu/~ryantibs/convexopt-S15/scribes/08-prox-grad-scribed.pdf

[3]http://www.cse.iitm.ac.in/~vplab/courses/SLT/PDF/Murphy_13.4Proximal%20Gradient.pdf

[4]http://niaohe.ise.illinois.edu/IE598_2016/pdf/IE598-lecture19-proximal%20gradient%20method%20and%20its%20acceleration.pdf

[5]https://people.eecs.berkeley.edu/~elghaoui/Teaching/EE227A/lecture18.pdf

[6]https://www.coursera.org/lecture/ml-regression/deriving-the-lasso-coordinate-descent-update-6OLyn

[7]https://github.com/purushottamkar/ml19-20w/blob/master/lecture_code/8_Optimization%20Refresher.ipynb

[8]https://en.wikipedia.org/wiki/Proximal_gradient_methods_for_learning

[9]https://www.coursera.org/lecture/ml-regression/deriving-the-lasso-coordinate-descent-update-6OLyn