COL775 - Deep Learning | Homework-02 | Rohit Agarwal

Part-A

Assume least squares objective,

$f(w) = \frac{1}{2n} \|Xw - y\|_2^2$, gradient $g(w) = \nabla f(w) = \frac{1}{n} X^T(Xw-y)$, Hessian $H = \frac{1}{n} X^T X$

→ As we iterate w_k, take a gradient descent p_k (for vanilla GD, $p_k = -g_k$), stepsize $t_k > 0$

(i) Exact Line Search (closed form for least squares)

for quadratic f, with PSD H, the exact minimizer along the line $w_k + t p_k$ is:

$$\boxed{t_k^* = \arg\min_{t \geq 0} \phi(t) = f(w_k + t p_k) = \frac{-g_k^T p_k}{p_k^T H p_k}}$$

Special Case : (steepest descent)

$p_k = -g_k$

$$t_k^* = \frac{g_k^T g_k}{g_k^T H g_k}$$

Algorithm:- (1) Compute $g_k = \nabla f(w_k)$ → (2) Set $p_k = -g_k$ → (3) Compute $t_k = \frac{-g_k^T p_k}{p_k^T H p_k}$ → (4) Update $w_{k+1} = w_k + t_k p_k$

(ii) α-β Backtracking Line Search (Armijo rule)

Parameters : choose $\alpha \in (0, 0.5)$, $\beta \in (0,1)$ (typical $\alpha = 10^{-4}$, $\beta = 0.5$)

Goal : find smallest $m \in \{0,1,2,\dots\}$ s.t Armijo decrease holds.

Algorithm:- (1) Compute $g_k = \nabla f(w_k)$ → (2) Initialise $t \leftarrow 1$ → (3) while $f(w_k + t p_k) > f(w_k) + \alpha t g_k^T p_k$, set $t \leftarrow \beta t$. → (4) set $t_k \leftarrow t$ and update $w_{k+1} = w_k + t_k p_k$

(iii) Ternary search along a line (for unimodal $\phi(t) = f(w_k + t p_k)$)

Use when $\phi(t)$ is unimodal on an interval $[0, T]$, if T is unknown first bracket it.

Bracketing (doubling)

(1) Set $t_0 \leftarrow 0$, $t_1 \leftarrow \tau > 0$ → (2) While $\phi(t_1) < \phi(t_0)$: set $t_0 \leftarrow t_1$, $t_1 \leftarrow 2t_1$ → (3) Now $[0, t_1]$ brackets the min^n.

Ternary Search on $[L,R]$ (with $L=0$, $R=T$):

(1) Repeat until $R - L < \varepsilon$,

- $m_1 = L + (R-L)/3$, $m_2 = R - \frac{R-L}{3}$
- If $\phi(m_1) < \phi(m_2)$, set $R \leftarrow m_2$ else $L \leftarrow m_1$

→ (2) Return $t_k \in [L, R]$ → (3) update $w_{k+1} = w_k + t_k p_k$

Part-B

We use the Kaggle House prices - Advanced Regression techniques train set (train.csv). To keep a pure least-squares objective, only numerical predictors are used. Missing values are imputed with column means, features are standardized and a bias column is added.

We fit a linear regression model via gradient descent ($p = -\nabla f$) on

$f(w) = \frac{1}{2n}\|Xw - y\|_2^2$ $\quad \nabla f(w) = \frac{1}{n} X^T(Xw - y)$.

1. α-β backtracking ($\alpha = 10^{-4}$, $\beta = .1$, $t_{init} = 1$).
2. Ternary search along the line $w + tp$ with bracketing by doubling (start $T = 1$, growth $= 2$) and $\epsilon = 10^{-7}$ (tolerance).

Stopping rule :→ $|f_k - f_{k-1}| < 10^{-10}$ or 5000 iterations.

Results :→
- Backtracking 462 iterations, 0.2154s, final $f = 5.7550 \times 10^8$
- Ternary 486 iterations, 1.3594s, final $f = 5.7550 \times 10^8$

Graphs:

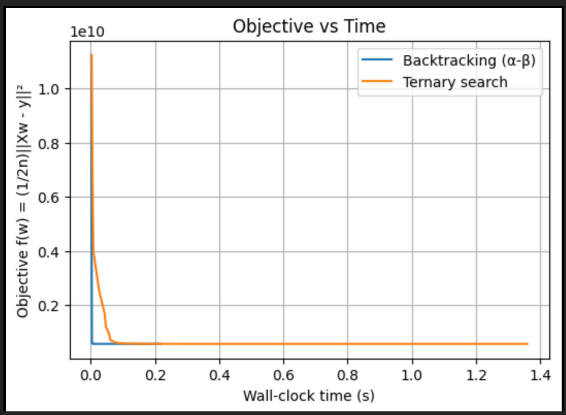

Fig 1. Objective f(w) vs Time (s)

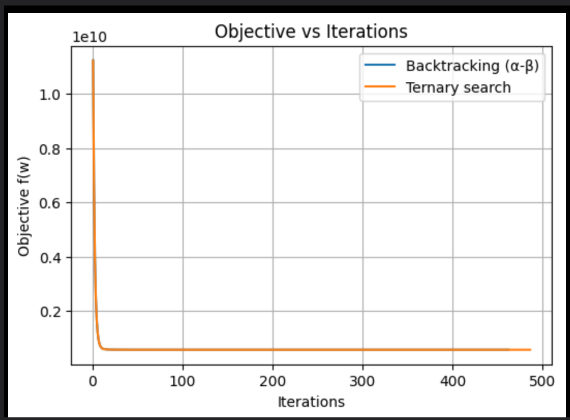

Fig 2: Objective f(w) vs iterations