

1 Line Search

L'algoritmo di line search calcola la grandezza del passo di ottimizzazione che si può fare nella direzione discendente $p_k = -B_k^{-1} \nabla f_k$, dove B è una matrice simmetrica, non-singolare, *positive semi-definite* che approssima la matrice Hessiana, e ∇f è il gradiente della funzione ¹. La direzione discendente p_k garantisce che il valore della funzione f può essere diminuito, ed è tale che $p_k^T \nabla f = -\nabla f_k^T B_k^{-1} \nabla f_k < 0$, essendo B *positive semi-definite*.

L'obiettivo della line search è trovare un minimizzatore locale della seguente funzione:

$$\phi(\alpha) = f(x_k + \alpha p_k) \quad , \alpha > 0$$

dove α è lo step length (passo di ottimizzazione), ed output dell'algoritmo di line search. Essendo una ricerca locale, anziché globale, la line search è detta inesatta. Ad ogni iterazione k si calcola uno step length α_k e si aggiorna la soluzione candidata x_k in $x_{k+1} = x_k + \alpha_k p_k$.

Per garantire il successo della line search, α_k deve soddisfare una o più delle seguenti condizioni: condizione di Armijo, condizione di Wolfe e condizione forte di Wolfe.

Condizione di Armijo. La condizione di Armijo, chiamata anche condizione di decremento sufficiente, è definita dalla seguente disuguaglianza:

$$\begin{aligned} \phi(\alpha) &\leq l(\alpha), \\ l(\alpha) &= f(x_k) + c_1 \alpha \nabla f_k^T p_k \end{aligned} \tag{1}$$

dove $l(\alpha)$ è una funzione lineare in α con coefficiente angolare $c_1 \nabla f_k^T p_k < 0$, il quale garantisce che il valore della funzione f diminuisca, e $c_1 \in (0, 1)$ è una costante. La condizione di Armijo verrebbe soddisfatta da ogni valore α sufficientemente piccolo, con lo svantaggio che si avrebbe una lenta convergenza dell'algoritmo. D'altra parte un valore di α troppo grande potrebbe andare oltre il punto di minimo cercato. Per questi motivi è opportuno che α soddisfi la seguente condizione.

Condizione di Wolfe. La condizione di Wolfe, conosciuta anche come condizione di curvatura, è definita dalla seguente disuguaglianza:

$$\phi'(\alpha) \geq c_2 \phi'(0) \tag{2}$$

dove $c_2 \in (c_1, 1)$ è una costante, $\phi'(\alpha_k) = \nabla f(x_k + \alpha_k p_k)^T p_k$, e $\phi'(\alpha) = \nabla f_k^T p_k$. (spiegare per bene, ancora non ho ben capito come funziona) Per evitare che $\phi'(\alpha) \gg 0$ sia troppo positivo, andrebbero esclusi i punti che vanno troppo oltre un punto stazionario di ϕ . Per soddisfare questo criterio si può modificare la condizione di curvatura, ottenendo quella che è chiamata la condizione forte di Wolfe, di seguito definita:

$$|\phi'(\alpha)| \leq c_2 |\phi'(0)| \tag{3}$$

¹Il pedice k indica l'indice di iterazione dell'algoritmo.

1.1 Algoritmo di Line Search

L'algoritmo di line search implementato è conosciuto sotto il nome di Backtracking Line Search. L'algoritmo parte da uno step size $\alpha = \alpha_0$ e lo decrementa finché non trova uno step size α che soddisfa la condizione di Armijo (criterio di stop della line search). Sebbene le condizioni di curvatura assicurano una convergenza più rapida dell'algoritmo di apprendimento, queste possono essere dispensate scegliendo un adeguato valore di α_0 (impostato ad 1 per quasi-Newton). Algoritmo 1 mostra lo pseudo-codice dell'algoritmo, in cui $c_1 = 10^{-4}$ è costante e $\rho \in (0, 1)$ è il fattore di contrazione di α . Il valore di ρ sarà deciso empiricamente e potrà variare da iterazione a iterazione.

Algorithm 1 Backtracking Line Search. La condizione di decremento sufficiente (definita in (1)) è stata riscritta completamente in termini di ϕ .

```
1:  $\alpha = 1$ 
2:  $c_1 = 10^{-4}$ 
3: while not  $\phi(\alpha) \leq \phi(0) + c_1 \alpha \phi'(0)$  do
4:    $\alpha \leftarrow \rho \alpha$ 
5: return  $\alpha$ 
```

Convergenza della Line Search. L'algoritmo convergerà in un numero finito di passi, dipendenti dal valore di ρ , perché α diventerà sufficientemente piccolo da soddisfare la condizione di Armijo.