with L2 regularizer
\[ L' = L + C \sum_c w_c^T w_c \]
if the derivative is over the same index (drag)
This adds 2C, making it + 2C

Newton's Method (1-D)

Taylor Series \rightarrow\text{approx w/ quadratic}

\[ f(x) + f'(x)(x-x^*) + \frac{f''(x)}{2}(x-x^*)^2 \]

\[ y = ax^2 + bx + c \]

\[ f'(x_i) = 2ax_i + b \]
\[ f''(x_i) = 2a \quad \Rightarrow \quad a = \frac{f''(x_i)}{2} \]

\[ f'(x_i) = f''(x_i)x_i + b \quad \Rightarrow \quad b = f'(x_i) - f''(x_i)x_i \]

\[ c \Rightarrow \text{who cares?} \]
\[
\begin{align*}
\text{minimum } & \quad 2ax_i + b = 0 \\
 f''(x_i)x_i + f'(x_i) - f''(x_i)x_i = 0 \\
 x_i &= x_i - \frac{f'(x_i)}{f''(x_i)}
\end{align*}
\]

**DEMO**

**III** Newton/Quasi-Newton

\[
X_{n+1} = X_n - H\nabla
\]

**BFGS** - set \( H_0 = I \) \( x_0 = ? \)

\[
\begin{align*}
\gamma &= \nabla(x_{k+1}) + \nabla(x_k) \\
H_{k+1} &= f(H_k, \gamma_k) \\
\text{if } H_k \text{ is+Def } \Rightarrow H_{k+1} \text{ is +Def}
\end{align*}
\]

**L-BFGS**

Compute \( H \) on the fly each time from the last \( m \) gradients.
Line Search

Brackets

\[
\begin{array}{cccc}
1 & 1 & 1 & 1 \\
\uparrow & \uparrow & \uparrow & \uparrow \\
x_1 & x_3 & x_4 & x_2 \\
\end{array}
\]

Where is the min?

Repeat until Wolfe conditions are satisfied

1. Objective has decreased "enough"
2. Slope has decreased "enough"

Defined by constants

In Practice

Matlab

Slow example file

Java - Mallet

Slow example file