Unconstrained Optimization

- Optimization problem

  Given $f : \mathbb{R}^n \longrightarrow \mathbb{R}$
  find $x_* \in \mathbb{R}^n$*, such that* $x_* = \operatorname*{argmin}_x f(x)$

- Global minimum and local minimum
- Optimality
  Necessary condition: $\nabla f(x_*) = 0$
  Sufficient condition: $H_f(x_*) = \nabla^2 f(x_*)$ is positive definite

Newton's method

- Taylor series approximation of f at k-th iterate x_k:

$$f(x) \approx f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2}(x - x_k)^T H_f(x_k)(x - x_k)$$

- Differentiating with respect to x and setting the result equal to zero yields the $k+1$-th iterate, namely Newton's method:

$$x_{k+1} = x_k - [H_f(x_k)]^{-1} \nabla f(x_k).$$

- Newton's method converges quadratically when x_0 is near a minimum.

Gradient descent optimization

- Directional derivative of f at x in the direction u:

$$\mathcal{D}_u f(x) = \lim_{n \to \infty} \frac{1}{h} \left[f(x + hu) - f(x) \right] = u^T \nabla f(x)$$

  It measures the change in the value of f relative to the change in the variable in the direction of u.

- To min $f(x)$, we would like to find the direction in which f decreases the fastest. Using the directional derivative,

$$\begin{aligned} \min_u u^T \nabla f(x) &= \min_u \|u\|_2 \|\nabla f(x)\|_2 \cos\theta \\ &= -\|\nabla f(x)\|_2^2 \end{aligned}$$

  when

$$u = -\nabla f(x).$$

- $u = -\nabla f(x)$ is call the steepest descent direction.

Gradient descent optimization

- The steepest descent algorithm:

$$x_{k+1} = x_k - \tau \cdot \nabla f(x_k),$$

where τ is called *stepsize* or "*learning rate*", which can be chosen as follows:

1. $\min_{\tau} f(x_k - \tau \cdot \nabla f(x_k))$, or
2. $\tau =$ small const. or
3. evaluate $f(x - \tau \nabla f(x))$ for several different values of τ and choose the one that results in the smallest objective function value.

Solving LS by gradient-descent

- Let $A \in \mathbb{R}^{m \times n}$ and $b = (b_i) \in \mathbb{R}^m$
- The least squares problem

$$\begin{aligned}\min_x f(x) &= \min_x \frac{1}{2}\|Ax - b\|_2^2 \\ &= \min_x \frac{1}{2}\sum_{i=1}^{m} f_i^2(x)\end{aligned}$$

  where

$$f_i(x) = A(i,:)^T x - b_i$$

- Gradient: $\nabla f(x) = A^T A x - A^T b$
- The method of gradient descent:
  - set the stepsize τ and tolerance δ to small positive numbers.
  - while $\|A^T A x - A^T b\|_2 > \delta$ do

$$x \leftarrow x - \tau \cdot (A^T A x - A^T b)$$

Solving LS by gradient-descent

MATLAB demo code: `lsbygd.m`

```
>> ...
>> r = A'*(A*x - b);
>> xp = x - tau*r;
>> res(k) = norm(r);
>> if res(k) <= tol, ... end
>> ...
>> x = xp;
>> ...
```

Connection with root finding

Solving nonlinear system of equations:

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0 \end{aligned}$$

is equivalent to solve the optimization problem

$$\min_x g(x) = g(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} (f_i(x_1, x_2, \ldots, x_n))^2$$