Lecture 10, 9/26/24

ECE 364
Fall 2024

Mathematical Models of Optimization

S: Constraint set – available decisions

f: cost function – maps elements of S into $\mathbb{R}$

$f(x)$: scalar measure of undesirability of choosing decision x.

Goal: find an optimal decision, i.e.,

an $x^* \in S$ s.t.

$$f(x^*) \leq f(x), \quad \forall x \in S$$

In ECE 364: x is n-dim - n-tuple of real numbers $(x_1, x_2, \cdots, x_n)$

$\Longrightarrow S \subset \mathbb{R}^n$

- Continuous: S is infinite & continuous

  e.g. $S = \mathbb{R}^n$ (unconstrained)

  Else specified by some eqns. & inequal.

- Discrete: usually S is finite

  subclass: integer programming

Nonliear programming:

- either f is non-linear or S is specified by non-lin. equs. & inequal.

- Linear programming

  f is linear and S is a polyhedral set specified by linear inequal. constraints.

Gradient Methods for Unconstrained Optim.

Iterative Descent

start at some point x_0, and succesively generate $x_1, x_2, \dots,$ s.t.

$$f(x_{k+1}) < f(x_k) \quad , \quad k = 0, 1, \dots$$

Steepest Descent

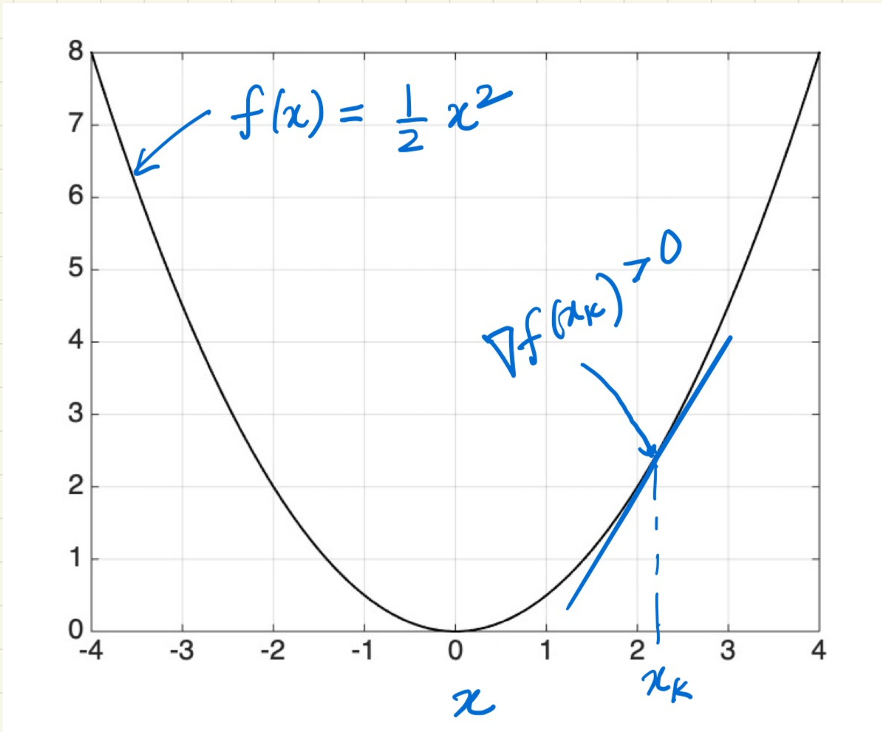
Move x_k in direction that decreases function most

Steepest Descent Algorithm

Assuming that $\nabla f(x_k) \neq 0$

to go from x_k to x_{k+1}

move $-\nabla f(x_k)$:



$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

α_k controls movement along gradient

- If α_k is too large function value may increase

General Gradient Descent Algorithm

Assume that $\nabla f(x_k) \neq 0$, then

$$x_{k+1} = x_{k+} + \alpha_k d_k$$

where d_k is s.t.

$$\nabla f(x_k)^T d_k < 0 \equiv -\nabla f(x_k)^T d_k > 0$$

d_k has a positive projection along $-\nabla f(x_k)$

- If $d_k = -\nabla f(x_k)$ we get steepest desc

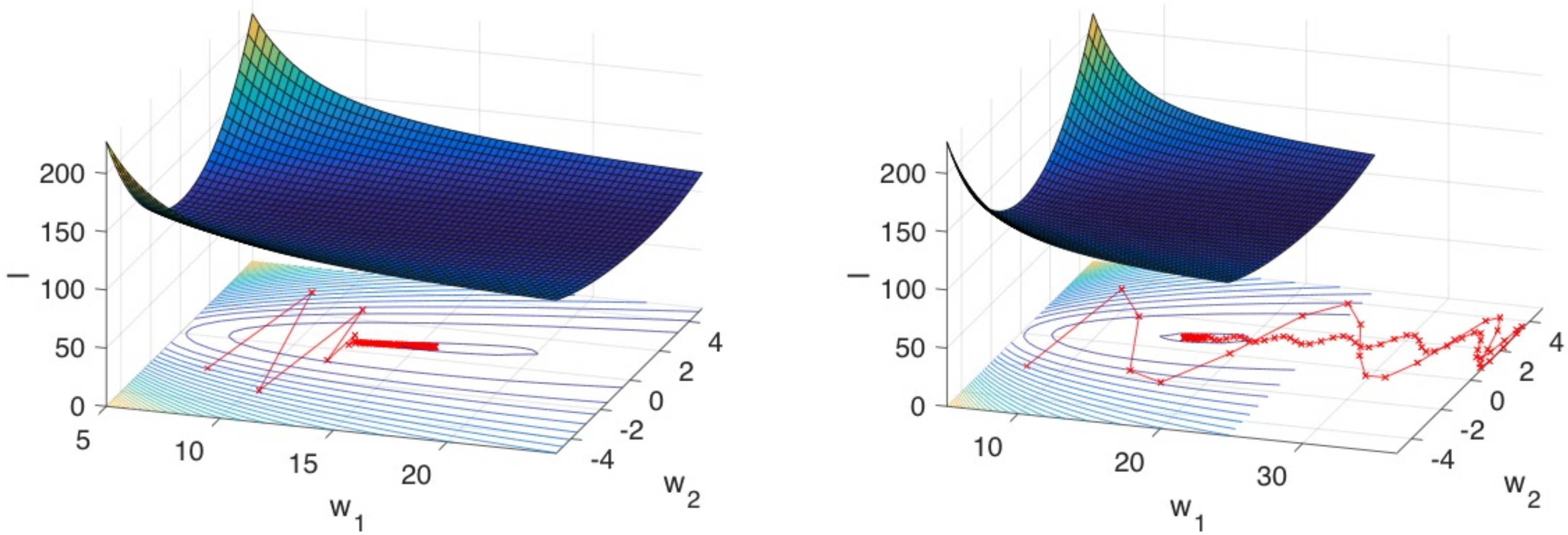
Typical construction: matrix $D_k \succ 0$

$$d_k = -D_k \nabla f(x_k)$$

then $$d_k^T \nabla f(x_k) = -\nabla f(x_k)^T D_k \nabla f(x_k) < 0$$

Gradient with momentum

Intuition:



- Polyak's method (aka heavy-ball)

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k) + \beta_k(\boldsymbol{w}_k - \boldsymbol{w}_{k-1})$$

- Momentum method in deep learning

$$\begin{aligned} \boldsymbol{v}_{k+1} &= \beta \boldsymbol{v}_k + \nabla f(\boldsymbol{w}_k) \\ \boldsymbol{w}_{k+1} &= \boldsymbol{w}_k - \alpha \boldsymbol{v}_{k+1} \end{aligned}$$