Gradient Method: Part 1

Lecture 4-1 - CMSE 382

Prof. Elizabeth Munch

Michigan State University
::
Dept of Computational Mathematics, Science & Engineering

Mon, Feb 2, 2026

This Lecture

Topics:

- Descent direction
- Gradient descent algorithm

Announcements:

- Quiz 2 on Wednesday, Feb 4
- Office hours posted on course webpage

Section 1

Descent Direction

Descent direction

Analytical approach

- **Goal:** Solve $\min\{f(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^n\}$
- Set $\nabla f(\mathbf{x}) = 0$ and find the stationary points $\{\mathbf{x}^*\}_i$
- Test $\nabla^2 f$ at each $\mathbf{x}^*$ to identify local optima
- Find any other potential global optima (e.g., boundary points)

What if that's not an option?

For example, where is $\nabla f(x,y,z) = \mathbf{0}$ for

$$\nabla f(x,y,z) = \begin{bmatrix} 3x - \cos(yz) - \frac{3}{2} \\ 4x^2 - 625y^2 + 2y - 1 \\ e^{-xy} + 20z + 10\pi \end{bmatrix}$$

(I don't wanna....)

Idea: The foggy mountain analogy

Photo by Ricardo Gomez Angel on Unsplash.

Descent Direction

Given $f : \mathbb{R}^n \to \mathbb{R}$ which is continuously differentiable.

Definition

The **directional derivative** of f at $\mathbf{x}$ along the direction $\mathbf{d}$ is defined as $f'(\mathbf{x}; \mathbf{d}) = \nabla f(\mathbf{x})^\top \mathbf{d}$.

- Gives the instantaneous rate of change of f along direction $\mathbf{d}$ through point $\mathbf{x}$.

desmos.com/3d/ojt8rjazr7

Definition

A nonzero vector $\mathbf{d} \in \mathbb{R}^n$ is a **descent direction** of f at $\mathbf{x}$ if the **directional derivative** $f'(\mathbf{x}; \mathbf{d}) = \nabla f(\mathbf{x})^\top \mathbf{d}$ is negative.

Descent direction

Lemma

Let f be a continuously differentiable function over an open set U, and let $\mathbf{x} \in U$. Suppose that $\mathbf{d}$ is a descent direction of f at $\mathbf{x}$.

Then there exists $\varepsilon > 0$ such that

$$f(\mathbf{x} + t\mathbf{d}) < f(\mathbf{x})$$

for any $t \in (0, \varepsilon]$.

Translation

There is a t such that if you start at $\mathbf{x}$ and move along $\mathbf{d}$ for a distance t, then you will reach a lower function value.

Gradient descent algorithm

The foggy mountain: Using local information to navigate the global landscape

Decisions needed:

- Starting point?
- Descent direction?
  - In the gradient method $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$.
- Stepsize?
- Stopping criteria?

Choosing the Stepsize

- **Fixed step size** keeps the step size constant.
  - How do we find the constant? Often heuristics, or trial and error.
- **Adaptive step size** via *exact line search* for t_k that minimizes $\min_{t \in \mathbb{R}} f(x_k + t\mathbf{d}_k)$.
  - Not always possible to find the exact minimizer.
- **Adaptive step size** via *backtracking line search*: pick three parameters: an initial guess $s > 0$, and $\alpha \in (0,1), \beta \in (0,1)$. Then the stepsize is $t_k = s\beta^{i_k}$, where i_k is the smallest non-negative integer such that $f(\mathbf{x}_k) - f(\mathbf{x}_k + s\beta^{i_k}\mathbf{d}_k) \geq -\alpha s \beta^{i_k} \nabla f(\mathbf{x}_k)^\top \mathbf{d}_k$.
  - Compromise that finds a "good enough" stepsize.
  - A theorem guarantees the existence of i_k.
- **Annealing step size**: Larger initial step size that is gradually decreased every step.
  - Steps are often exponentially decayed
  - Allows smaller steps as the algorithm approaches the minimum

The Gradient Method

Input: tolerance parameter $\varepsilon > 0$.

Initialization: Pick $x_0 \in \mathbb{R}^n$ arbitrarily.
For any $k = 0, 1, 2, \ldots$ **do**:

1. Set descent direction to $\mathbf{d} = -\nabla f(\mathbf{x}_k)$
2. Pick a stepsize t_k
   - For example, using exact line search on the function $g(t) = f(\mathbf{x}_k - t\nabla f(\mathbf{x}_k))$.
3. Set $\mathbf{x}_{k+1} = \mathbf{x}_k - t_k \nabla f(\mathbf{x}_k)$.
4. If $\|\nabla f(\mathbf{x}_{k+1})\| \leq \varepsilon$, then STOP and $\mathbf{x}_{k+1}$ is the output.

Groups - Round 2

Group 1
Abigail, Shanze, Jack, Quang Minh,

Group 2
Igor, Atticus, K M Tausif, Long,

Group 3
Yousif, Zheng, Jake, Purvi,

Group 4
Maye, Alice, Arjun, Kyle,

Group 5
Monirul Amin, Jay, Brandon, Luis,

Group 6
Scott, Ha, Lora, Tianjian,

Group 7
Braedon, Sai, Joseph, Noah,

Group 8
Michal, Aidan, Jonid, Dev,

Group 9
Vinod, Saitej, Anthony, Breena,

Group 10
Karen, Dori, Lowell, Aaron,

Group 11
Jamie, Sanskaar, Dominic, Lauryn,

Group 12
Andrew, Arya, Daniel, Morgan,