Lectures 15-16

Gradient Descent and Convexity

DSC 40A, Fall 2025

Agenda

- Minimizing functions using gradient descent.
- Convexity.
- More examples.
    - Huber loss.
    - Gradient descent with multiple variables.

Question 🤔

Answer at q.dsc40a.com

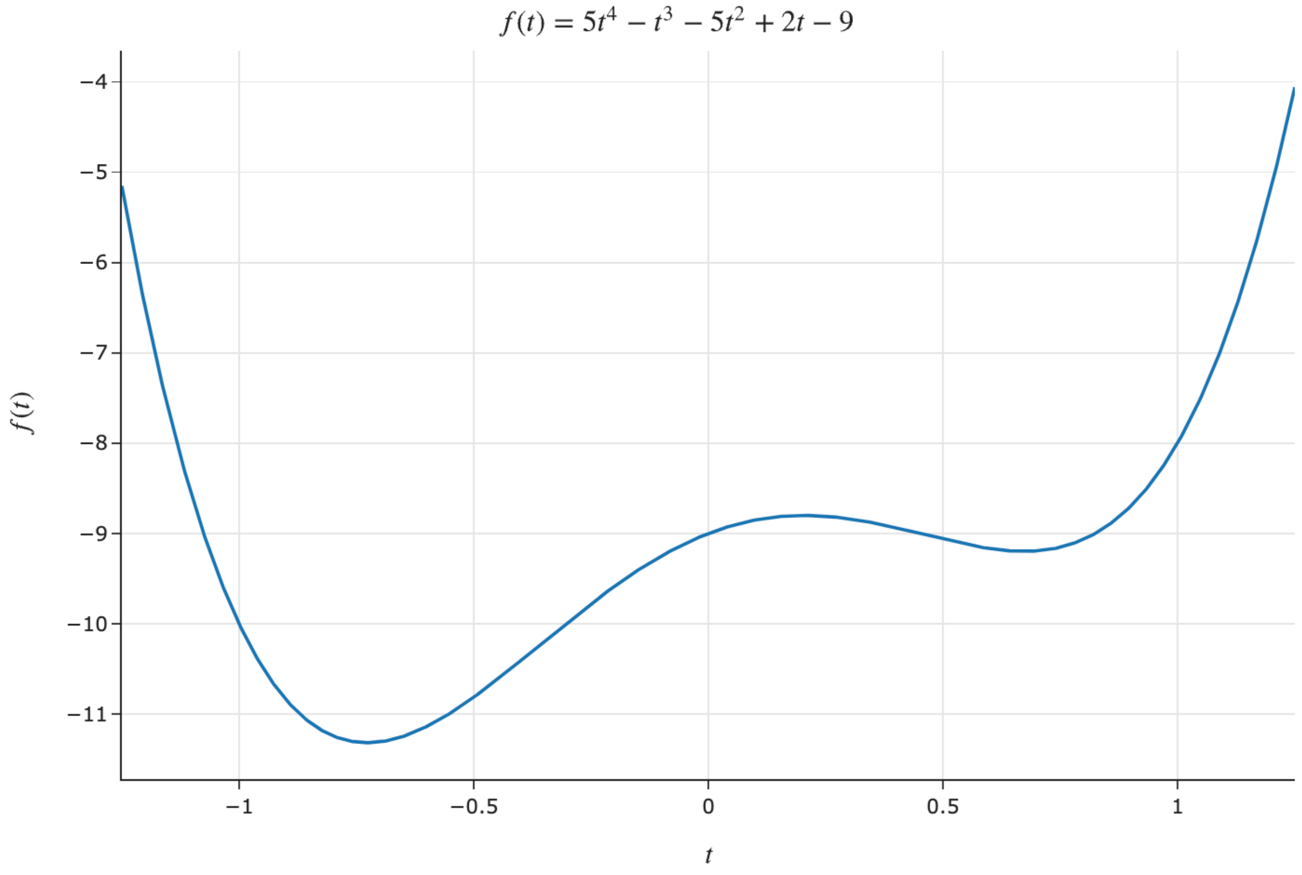
Remember, you can always ask questions at q.dsc40a.com!

If the direct link doesn't work, click the "🤔 Lecture Questions" link in the top right corner of dsc40a.com.

Minimizing functions using gradient descent

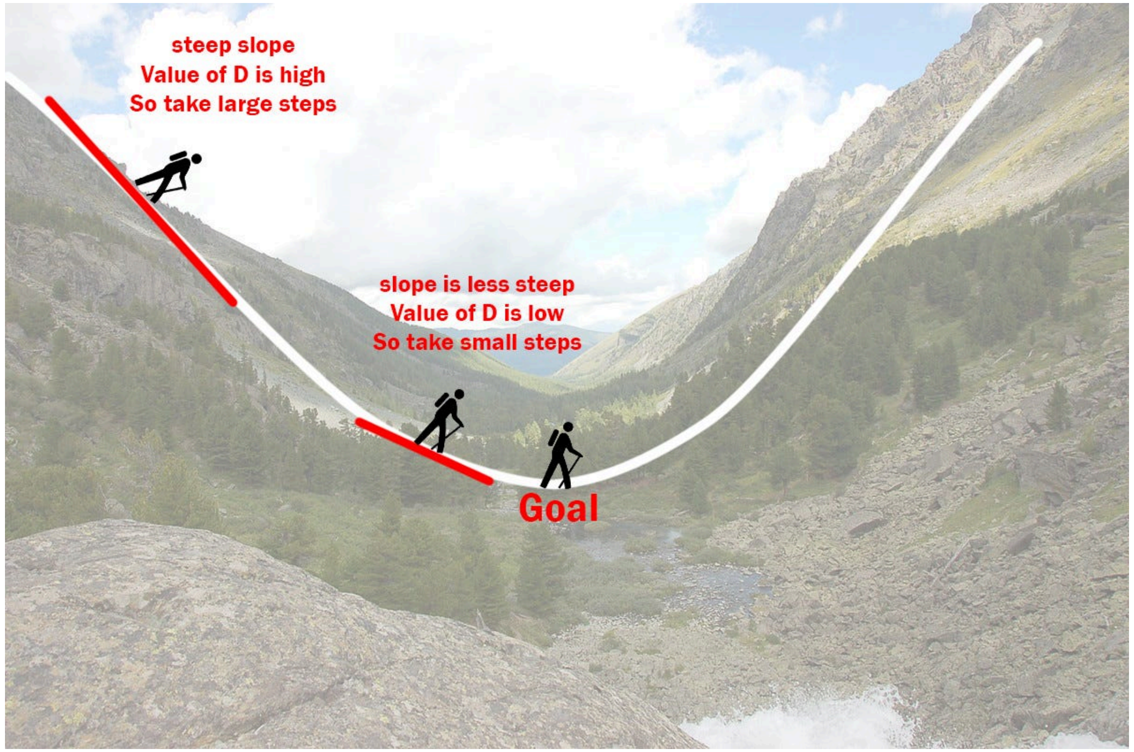
What does the derivative of a function tell us?

- **Goal**: Given a **differentiable** function $f(t)$, find the input t^* that minimizes $f(t)$.
- What does $\frac{d}{dt} f(t)$ mean?

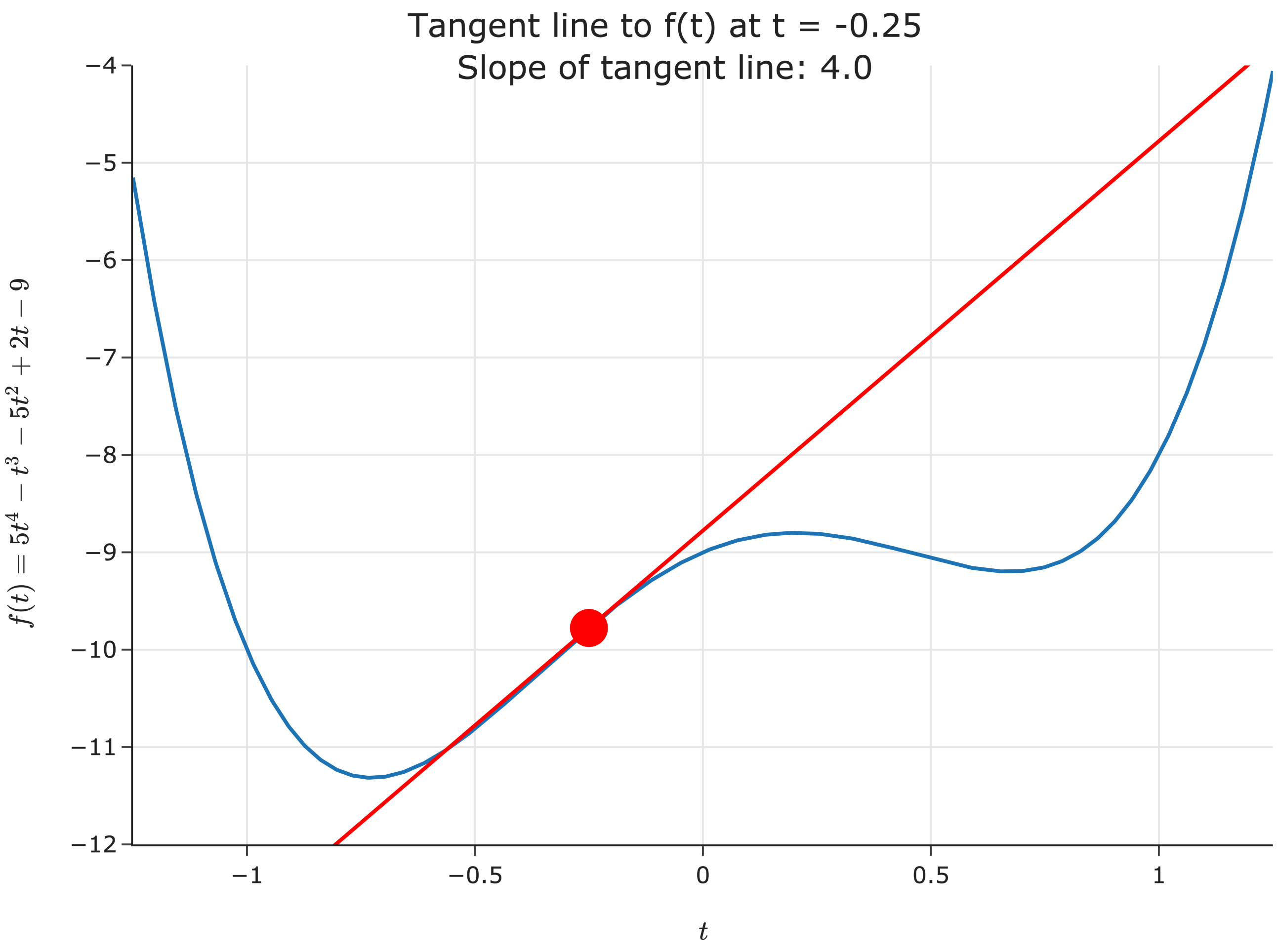


Let's go hiking!

- Suppose you're at the top of a mountain 🏔️ and need to get **to the bottom**.
- Further, suppose it's really cloudy ☁️, meaning you can only see a few feet around you.
- **How** would you get to the bottom?



Searching for the minimum

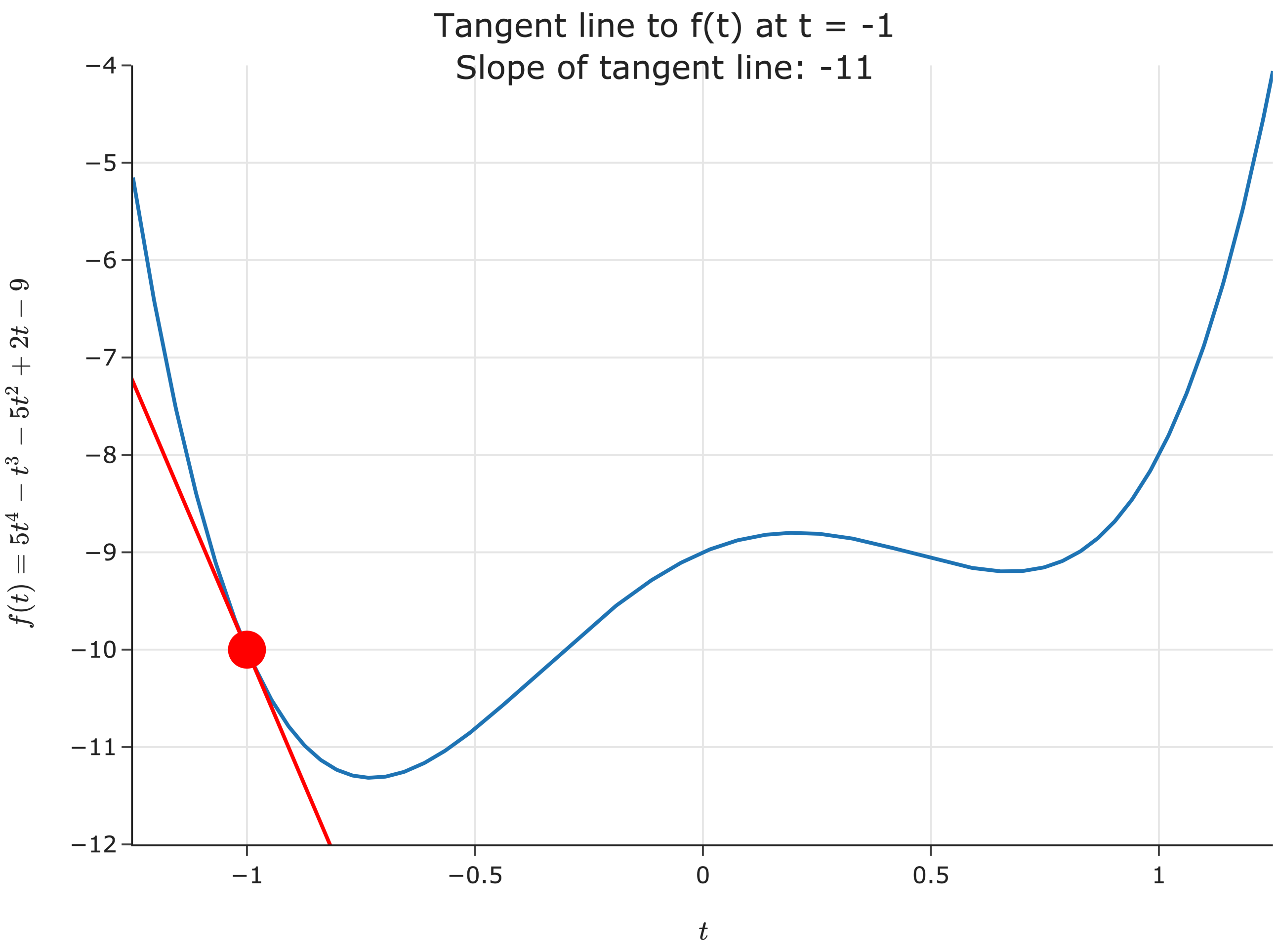


Suppose we're given an initial *guess* for a value of t that minimizes $f(t)$.

If the **slope of the tangent line at $f(t)$** is **positive** 📈:

- Increasing t **increases** f.
- This means the minimum must be to the **left** of the point $(t, f(t))$.
- Solution: **Decrease** t ⬇️.

Searching for the minimum



Suppose we're given an initial *guess* for a value of t that minimizes $f(t)$.

If the **slope of the tangent line at $f(t)$** is **negative** 📉:

- Increasing t **decreases** f.
- This means the minimum must be to the **right** of the point $(t, f(t))$.
- Solution: **Increase t** ⬆️.

Intuition

- To minimize $f(t)$, start with an initial guess t_0.
- Where do we go next?
  - If $\frac{df}{dt}(t_0) > 0$, **decrease** t_0.
  - If $\frac{df}{dt}(t_0) < 0$, **increase** t_0.
- One way to accomplish this:

$$t_1 = t_0 - \frac{df}{dt}(t_0)$$

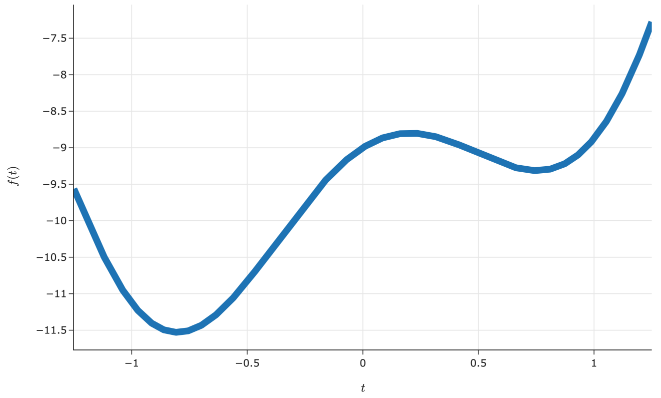
Gradient descent

To minimize a **differentiable** function f:

- Pick a positive number, α. This number is called the **learning rate**, or **step size**.
- Pick an **initial guess**, t_0.
- Then, repeatedly update your guess using the **update rule**:

$$t_{i+1} = t_i - \alpha \frac{df}{dt}(t_i)$$

- Repeat this process until **convergence** – that is, when t doesn't change much.



What is gradient descent?

- Gradient descent is a numerical method for finding the input to a function f that minimizes the function.
- Why is it called **gradient** descent?
  - The gradient is the extension of the derivative to functions of multiple variables.
  - We will see how to use gradient descent with multivariate functions next class.
- What is a **numerical** method?
  - A numerical method is a technique for approximating the solution to a mathematical problem, often by using the computer.

Gradient descent

```python
def gradient_descent(derivative, h, alpha, tol=1e-12):
    """Minimize using gradient descent."""
    while True:
        h_next = h - alpha * derivative(h)
        if abs(h_next - h) < tol:
            break
        h = h_next
    return h
```

See this notebook for a demo!

Gradient descent and empirical risk minimization

- While gradient descent can minimize other kinds of differentiable functions, its most common use case is in **minimizing empirical risk**.
- Gradient descent is **widely used** in machine learning, to train models from linear regression to neural networks and transformers (includng ChatGPT)!

Question 🤔

Answer at q.dsc40a.com

- For example, consider:
  - The constant model, $H(x) = h$.
  - The dataset $-4, -2, 2, 4$.
  - The initial guess $h_0 = 4$ and the learning rate $\alpha = \frac{1}{4}$.
- **Exercise**: Find h_1 and h_2.

Empirical Minimization with Gradient Descent

$$R_{\text{sq}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - h)^2 \qquad \frac{dR_{\text{sq}}}{dh} = \frac{2}{n}\sum_{i=1}^{n}(h - y_i)$$

- The dataset $-4, -2, 2, 4$.
- The initial guess $h_0 = 4$ and the learning rate $\alpha = \frac{1}{4}$.

$h_1 =$

Lingering questions

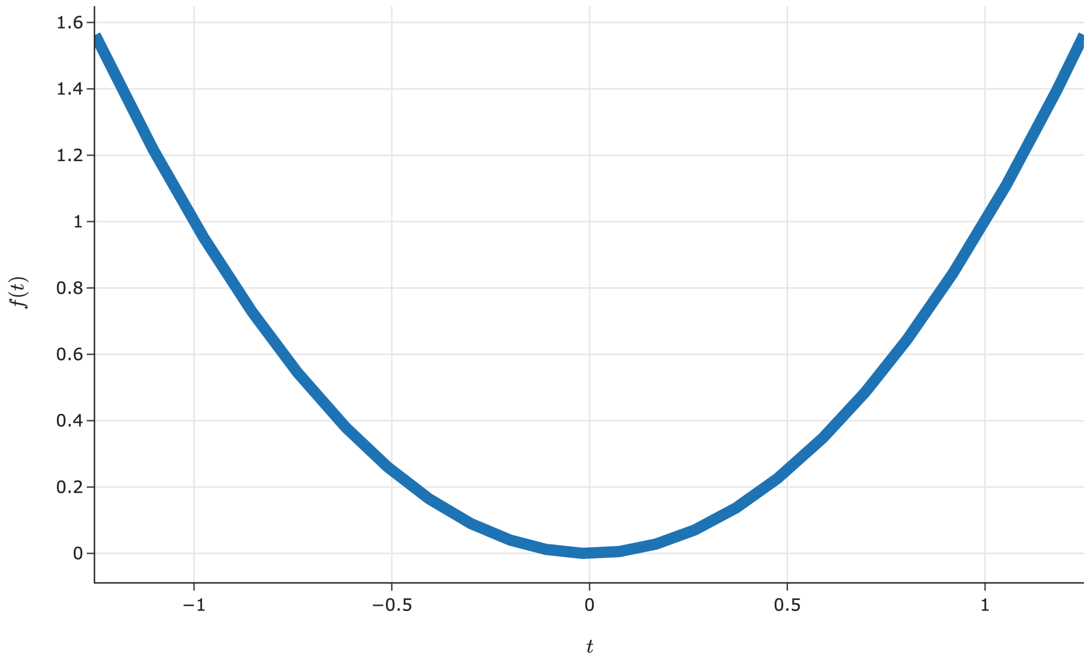
Now, we'll explore the following ideas:

- When is gradient descent *guaranteed* to converge to a global minimum?
    - What kinds of functions work well with gradient descent?
- How do I choose a step size?
- How do I use gradient descent to minimize functions of multiple variables, e.g.:

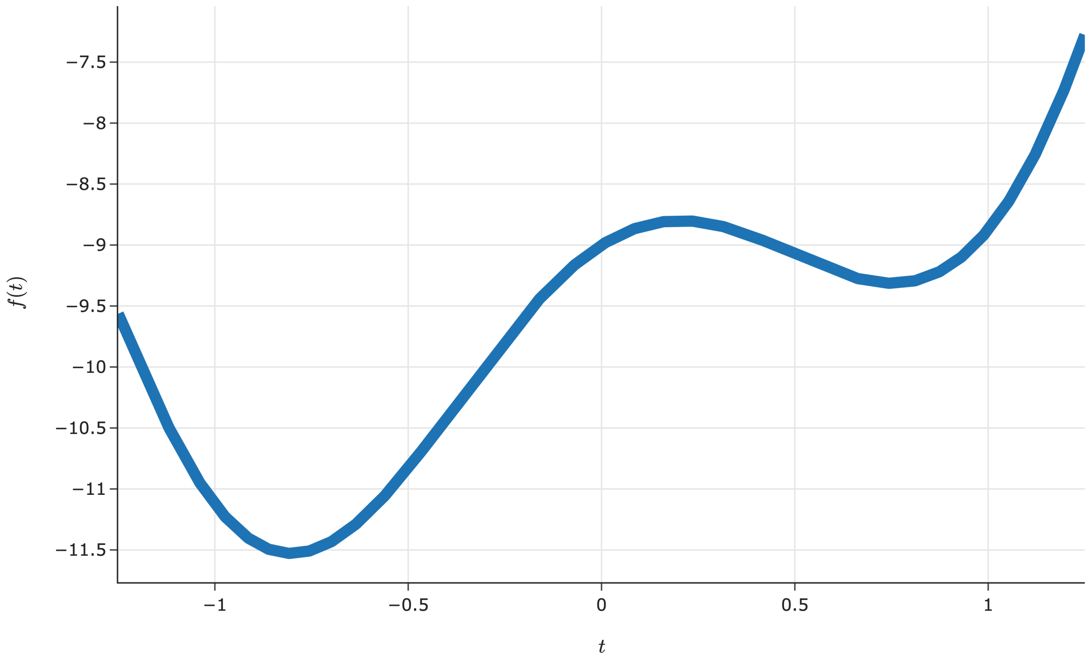
$$R_{\text{sq}}(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - (w_0 + w_1 x_i))^2$$

When is gradient descent guaranteed to work?

Convex functions



A **convex** function ✅



A **non-convex** function ❌