Python Data Products

Course 2: Design thinking and predictive pipelines

Lecture: gradient descent

Learning objectives

In this lecture we will...

- Introduce the notion of gradient descent, a general-purpose approach for model fitting

Closed form solutions

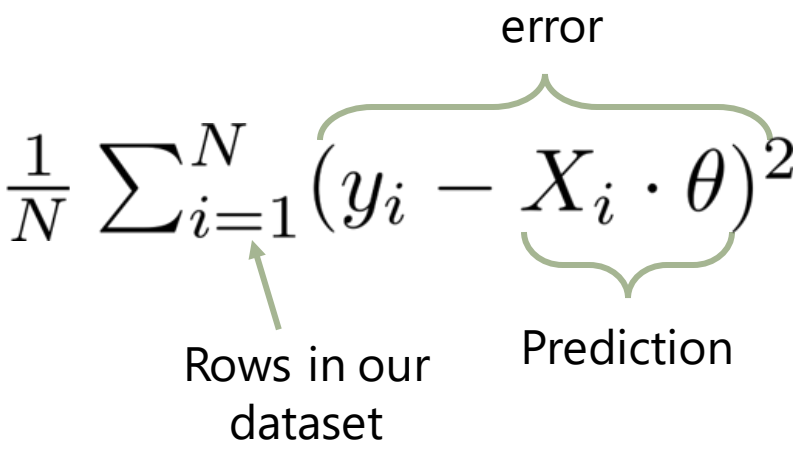
When we introduced regression, we optimized its parameters by solving a system of matrix equations:

$$X\theta = y \longrightarrow \theta = (X^T X)^{-1} X^T y$$

- But in general we don't have nice **closed-form** solutions to choosing our models
- What can we do in cases where a closed-form isn't available?
  - E.g. how would we solve *this* problem if we couldn't find a closed form?

Optimization with gradient descent

Precisely, the problem we're trying to solve looks like



- How do we **solve this for theta**?
  - i.e., how do we choose

$$\arg\min_\theta \sum_i (x_i \cdot \theta - y_i)^2$$

Concept: Gradient Descent

Gradient Descent is a general-purpose optimization approach to solve **continuous minimization** problems that don't have a closed form

- Normally, to solve a continuous minimization problem, we would
  1. Compute the gradient w.r.t. theta
  2. Find the points where the gradient is equal to zero (i.e., the *minima* of the function)
- If we can't do (2) above, gradient descent helps us to find *local minima*

Concept: Gradient Descent

With gradient descent, rather than "solving" the problem by finding its zeros, we instead start with an initial guess, and iteratively update our solution *in the direction of the gradient*

- In this way, we gradually find solutions that come progressively closer to being zeros of the gradient equation – even though we couldn't solve it in closed-form
- The points we find are called *local minima* of the original function

The gradient descent algorithm

In essence gradient descent (to minimize a function $f(\theta)$) works as follows:

1. Initialize θ at random
2. While (not converged) do
   $\theta := \theta - \alpha f'(\theta)$

All sorts of annoying issues:

- How to initialize theta?
- How to determine when the process has converged?
- How to set the step size alpha

(these aren't really the point of this course though)

The gradient descent algorithm

So what exactly is going on here?

The gradient descent algorithm

And how would we use it to solve our regularization objective?

$$f(\theta) = \frac{1}{N}\|y - X\theta\|_2^2$$

$$\frac{\partial f}{\partial \theta_k} \ ?$$

Summary of concepts

- Introduced the **gradient descent** algorithm
- Showed how this algorithm can be applied to solve the types of regression problems we've seen so far