Math 345 - Notes
Gradient Descent Algorithm
October 24, 2018

Gradient Descent Algorithm

In this section we will find another way of finding the argument that minimizes an error function. For example, in linear regression, we can find the weight vector $\mathbf{w}$ that gives the minimum for $E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\sum_{k=1}^{N}(\mathbf{w}^T\mathbf{x}_k - y_k)^2$ by considering the gradient

$$\nabla E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\nabla\left(\sum_{k=1}^{N}(\mathbf{w}^T\mathbf{x}_k - y_k)^2\right) = \frac{2}{N}\left(X^TX\mathbf{w} - X^T\mathbf{y}\right).$$

The Gradient Descent Algorithm can be used to minimize any twice-differentiable function. However, the algorithm finds local minima (depending on the starting configuration), and may not find the absolute minimum.

The gradient and directional derivatives

Suppose we have a real valued function f with $d-$dimensional input vectors $\mathbf{x} = (x_1, x_2, \ldots, x_d)$. The gradient at $P(a_1, \cdots, a_d)$ is defined as

$$\nabla f(a_1, \cdots, a_d) = (f_{x_1}(P), \cdots, f_{x_d}(P)) = \left(\frac{\partial f}{\partial x_1}(P), \cdots, \frac{\partial f}{\partial x_d}(P)\right).$$

The gradient of f at P is a measure of how f changes. The i coordinate of $\nabla f(P)$ represents the rate at which f changes in the direction of the i coordinate. More generally, if $\mathbf{u}$ is a unit vector (vector of length 1), we can measure the change of f in the direction of $\mathbf{u}$ at P by computing the directional derivative

$$D_{\mathbf{u}}f(P) = \nabla f(P)\cdot\mathbf{u} = ||\nabla f(P)||\,||\mathbf{u}||\,cos(\theta) = ||\nabla f(P)||\,cos(\theta)$$

where θ is the angle between $\nabla f(P)$ and $\mathbf{u}$. That is, the rate of change of f in the direction of $\mathbf{u}$ is found by taking the dot product of the gradient with the vector $\mathbf{u}$. Note that the maximum increase of f is obtained when $\theta = 0$, that is, when $\mathbf{u}$ and $\nabla f(p)$ have the same direction. Similarly, f decreases fastest in the direction of $-\nabla f(P)$.

The algorithm

The idea here is simple. We want to reach a local minimum by iteratively taking small steps toward the minimum. To decide in which direction to go, choose the direction with the fastest descent, that is, the direction opposite to the direction of the gradient.

Let η denote the "size" of the step toward the minimum we take during an iteration. To approximate the local minimum $\mathbf{x}_{\text{min}}$ of $f(\mathbf{x})$, the algorithm proceeds as follows:

Step 1. Initialize $\mathbf{x}_0$

Step 2. For $t \in \mathbb{N}$, move in the opposite direction if the gradient $\mathbf{g}_t = \nabla f(\mathbf{x}_t)$ with increment η. The update rule is

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t).$$

Iterate until the stopping condition is met.

Step 3. Output final $\mathbf{x}_t$

Remarks:

1. For linear regression, the most common ways to initialize are
   (i) $\mathbf{x}_0 = \mathbf{0}$
   (ii) Each coordinate of $\mathbf{x}_0$ is independently initialized using a normal random variable with mean zero and small variance.
2. For a general function $f(\mathbf{x})$, the initial $\mathbf{x}_0$ influences which local minimum is found by the algorithm.
3. The step size η may be not constant. In fact, taking variable steps η_t that get smaller as t increases might lead to a better approximation of the local minimum.
4. The stopping condition can be any combination of the following:
   (i) Run algorithm for a large number of steps (thousands). This may lead to a bad approximation of the minimum.
   (ii) Run algorithm until the gradient is very small: $||\mathbf{g}_t|| < \delta$, for some small δ.
   (iii) Run algorithm until the change in the function is small: $||f(\mathbf{x}_{t+1} - f(\mathbf{x}_t)|| < \delta$, for some small δ.

Examples: Let us see a few examples.

1. Suppose $f(x) = x^4 - 4x^2$.

   Then we can analytically solve for the extrema: $f'(x) = 4x^3 - 8x = 0$ if $x \in \{-\sqrt{2}, 0, \sqrt{2}\}$, and since $f''(x) = 12x^2 - 8$, using the Second Derivative Test, we conclude $x = 0$ gives a local maximum and $x = \pm\sqrt{2}$ lead to local minima.

   Using the gradient descent algorithm instead, with step $\eta = 0.1$, and initial value x_0

   (a)

$$\begin{aligned} x_0 &= 0 \\ x_1 &= x_0 - \eta f'(x_0) = 0 - (0.1)(0) = 0 \end{aligned}$$

   The algorithm does not find a minimum!

(b)

$$\begin{aligned}
x_0 &= 1\\
x_1 &= x_0 - \eta\, f'(x_0) = 1 - (0.1)f'(1) = 1.4\\
x_2 &= x_1 - \eta\, f'(x_1) = 1.4 - (0.1)f'(1.4) = 1.4224\\
x_3 &= x_2 - \eta\, f'(x_2) = 1.4224 - (0.1)f'(1.4224) = 1.409188\\
x_4 &= x_3 - \eta\, f'(x_3) = 1.409188 - (0.1)f'(1.409188) = 1.417186
\end{aligned}$$

The algorithm converges to $\sqrt{2}$.

(c) We can try other starting values:

$$\begin{aligned}
x_0 = 2 &\rightarrow \text{converges to } \sqrt{2}\\
x_0 = 2.5 &\rightarrow \text{converges to } -\sqrt{2}\\
x_0 = 3 &\rightarrow \text{no convergence, blows up}\\
x_0 = -1 &\rightarrow \text{converges to } -\sqrt{2}\\
x_0 = -2 &\rightarrow \text{converges to } -\sqrt{2}\\
x_0 = -2.5 &\rightarrow \text{converges to } \sqrt{2}\\
x_0 = 3 &\rightarrow \text{no convergence, blows up}
\end{aligned}$$

2. If $f(x, y) = x^2 + y^2 - 4y + 4$, we compute the gradient $\nabla f(x, y) = (2x, 2y - 4)$.

Solving analytically, the function has critical points when $\nabla f = (0, 0)$, hence at $(0, 2)$. Note that $f(x, y) = x^2 + (y - 2)^2 \geq 0$ and it equals zero only at $(0, 2)$, so the critical point must be a local (and absolute) minimum.

Using the gradient descent algorithm instead, with step $\eta = 0.1$, and initial value $\mathbf{x}_0 = \mathbf{0}$:

$$\begin{aligned}
\mathbf{x}_0 &= (0, 0)\\
\mathbf{x}_1 &= \mathbf{x}_0 - \eta\, \nabla f(\mathbf{x}_0) = (0, 0) - (0.1)(0, -4) = (0, 0.4)\\
\mathbf{x}_2 &= \mathbf{x}_1 - \eta\, \nabla f(\mathbf{x}_1) = (0, 0.4) - (0.1)(0, -3.2) = (0, 0.72)\\
\mathbf{x}_3 &= \mathbf{x}_2 - \eta\, \nabla f(\mathbf{x}_2) = (0, 0.72) - (0.1)(0, -2.56) = (0, 0.976)\\
\mathbf{x}_4 &= \mathbf{x}_3 - \eta\, \nabla f(\mathbf{x}_3) = (0, 0.976) - (0.1)(0, -2.048) = (0, 1.1808)\\
\mathbf{x}_5 &= \mathbf{x}_4 - \eta\, \nabla f(\mathbf{x}_4) = (0, 1.1808) - (0.1)(0, -1.6384) = (0, 1.34464)
\end{aligned}$$

The algorithm slowly converges to $(0, 2)$.

Using any other initial configuration, the algorithm will converge to $(0, 2)$.