Pattern Recognition
Introduction to Gradient Descent

Ad Feelders

Universiteit Utrecht

Optimization (single variable)

Suppose we want to find the value of x for which the function

$$y = f(x)$$

is minimized (or maximized).

From calculus we know that a necessary condition for a minimum is:

$$\frac{df}{dx} = 0 \tag{1}$$

This condition is not sufficient, since maxima and points of inflection also satisfy equation (1). Together with the second-order condition:

$$\frac{d^2f}{dx^2} > 0, \tag{2}$$

we have a sufficient condition for a local minimum.

Optimization (single variable)

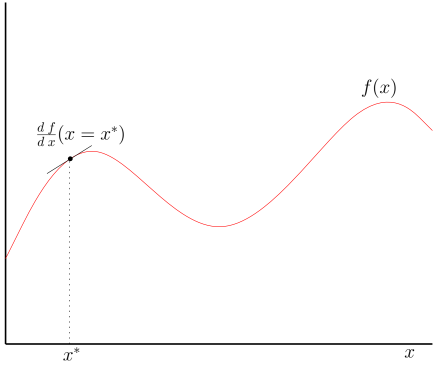
The equation

$$\frac{df}{dx} = 0$$

may not have a closed form solution however.

In such cases we have to resort to iterative numerical procedures such as gradient descent.

Optimization (single variable)



The derivative at $x = x^*$ is positive, so to increase the function value we should increase the value of x, i.e. make a step in the direction of the derivative.

Gradient Descent Algorithm (single variable)

The basic *gradient-descent algorithm* is:

1. Set $t = 0$, and *choose* an initial value $x^{(0)}$
2. determine the derivative
$$\frac{df}{dx}(x = x^{(t)})$$
of $f(x)$ at $x^{(t)}$ and *update*
$$x^{(t+1)} = x^{(t)} - \eta \frac{df}{dx}(x = x^{(t)})$$
Set $t = t + 1$.
3. Repeat the previous step until
$$\frac{df}{dx} = 0$$
and *check* if a *(local) minimum* has been reached.

$\eta > 0$ is the *step size* (or *learning rate*).

Optimization (multiple variables)

Suppose we want to find the values of $x_1, \ldots, x_p$ for which the function

$$y = f(x_1, \ldots, x_p)$$

is minimized (or maximized).

Analogous to the single-variable case a necessary condition for a minimum is:

$$\frac{\partial f}{\partial x_j} = 0 \qquad j = 1, \ldots, p \tag{3}$$

Again this condition is not sufficient, since maxima and saddle points also satisfy (3). For the second order condition, define the Hessian matrix H, with

$$H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

Together with the second-order condition that H is positive definite, i.e.

$$z^\top H z > 0, \qquad \text{for all } z \neq 0 \tag{4}$$

we have a sufficient condition for a local minimum.

Linear Functions

Consider a linear function

$$f(x) = a + \sum_{i=1}^{p} b_i x_i = a + b^\top x$$

The contour lines of f are given by

$$f(x) = a + b^\top x = c,$$

for different values of the constant c.

For linear functions the contours are parallel straight lines.

The Gradient

The gradient of

$$f(x_1, x_2, \ldots, x_p),$$

is the vector of partial derivatives

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_p} \end{bmatrix}$$

Gradient of a Linear Function

The gradient of a linear function

$$f(x) = a + b^\top x$$

is given by

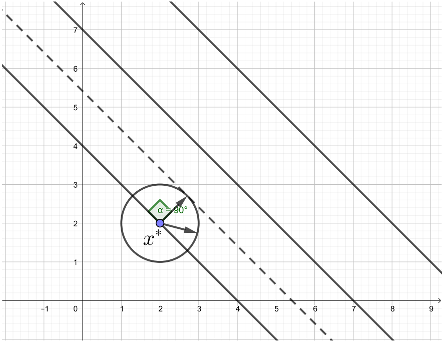
$$\nabla f = b$$

Furthermore, for linear functions we have:

$$\Delta f = b^\top \Delta x = \nabla f^\top \Delta x$$

In which direction should we move to maximize Δf?

The direction of steepest ascent (descent) is perpendicular to the contour line



The direction of steepest ascent (descent) is an increasing (decreasing) direction perpendicular to the contour line. The direction of steepest ascent (descent) from the point x^* is where the contour line is tangent to a circle of radius one around x^*.

The gradient is also perpendicular to the contour line

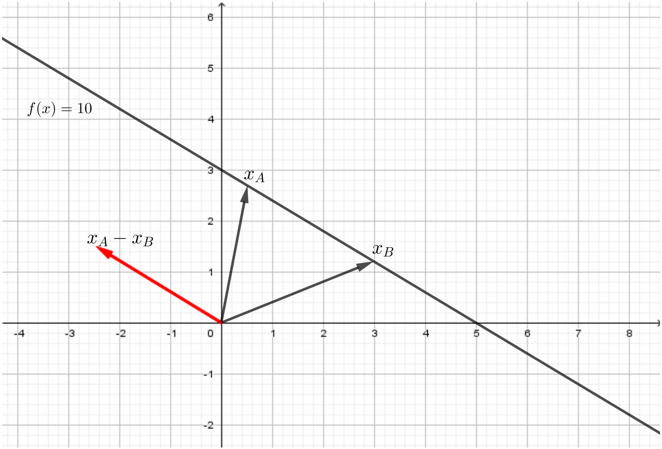
Consider two points x_A and x_B both of which lie on the same contour line. Because $f(x_A) = f(x_B) = c$, we have

$$f(x_A) - f(x_B) = (a + b^\top x_A) - (a + b^\top x_B) = b^\top (x_A - x_B) = 0$$

and so the gradient is perpendicular to the contour line, because

1. The vector $x_A - x_B$ runs parallel to the contour line.
2. Vectors are perpendicular if their dot product is zero.

The gradient is also perpendicular to the contour line



The gradient is perpendicular to the contour line

For linear functions the direction of steepest increase is perpendicular to the contour line, as is the gradient.

From

$$\Delta f = b^\top \Delta x = \nabla f^\top \Delta x$$

we conclude that the gradient points in an increasing direction, since filling in ∇f for Δx gives

$$\Delta f = \nabla f^\top \nabla f = \|\nabla f\|^2$$

Therefore:

1. The gradient points in the direction of fastest increase of f.
2. Minus the gradient points in the direction of fastest decrease of f.

Linear Approximation

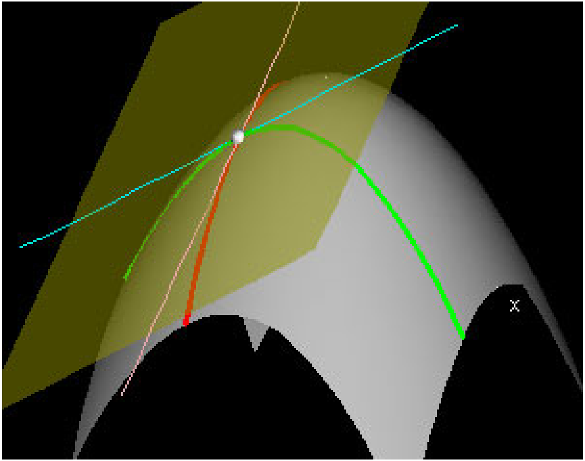
This reasoning works for arbitrary functions by considering a *local linear approximation* to the function at x^* by the tangent plane:

$$(y - y^*) = \frac{\partial f}{\partial x_1}(x_1 - x_1^*) + \frac{\partial f}{\partial x_2}(x_2 - x_2^*),$$

and using the linear approximation

$$\Delta f \approx \frac{\partial f}{\partial x_1}\Delta x_1 + \frac{\partial f}{\partial x_2}\Delta x_2 = \nabla f^\top \Delta x.$$

Local Linear Approximation by Tangent Plane



Gradient Descent Algorithm (multivariable)

The basic *gradient-descent algorithm* is:

1. Set $t = 0$, and *choose* an initial value $x^{(0)}$
2. determine the gradient $\nabla f(x^{(t)})$ of $f(x)$ at $x^{(t)}$ and *update*

   $$x^{(t+1)} = x^{(t)} - \eta \nabla f(x^{(t)})$$

   Set $t = t + 1$.
3. Repeat the previous step until

   $$\nabla f(x^{(t)}) = 0$$

   and *check* if a *(local) minimum* has been reached.

$\eta > 0$ is the *step size* (or *learning rate*).

Example of gradient descent

Note: w_0 and w_1 are the variables here!

n	x	t	$y = w_0 + w_1 x$	$e = t - y$
1	0	1	w_0	$1 - w_0$
2	1	3	$w_0 + w_1$	$3 - w_0 - w_1$
3	2	4	$w_0 + 2w_1$	$4 - w_0 - 2w_1$
4	3	3	$w_0 + 3w_1$	$3 - w_0 - 3w_1$
5	4	5	$w_0 + 4w_1$	$5 - w_0 - 4w_1$

$$\begin{aligned}\text{SSE}(w_0, w_1) &= (1 - w_0)^2 + (3 - w_0 - w_1)^2 \\ &\quad + (4 - w_0 - 2w_1)^2 + (3 - w_0 - 3w_1)^2 \\ &\quad + (5 - w_0 - 4w_1)^2\end{aligned}$$

Example of gradient descent

$$
\begin{aligned}
\frac{\partial \mathrm{SSE}}{\partial w_0} &= [2(1-w_0)(-1)] + [2(3-w_0-w_1)(-1)] \\
&+ [2(4-w_0-2w_1)(-1)] + [2(3-w_0-3w_1)(-1)] \\
&+ [2(5-w_0-4w_1)(-1)] = -32 + 10w_0 + 20w_1
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \mathrm{SSE}}{\partial w_1} &= 0 + [2(3-w_0-w_1)(-1)] \\
&+ [2(4-w_0-2w_1)(-2)] + [2(3-w_0-3w_1)(-3)] \\
&+ [2(5-w_0-4w_1)(-4)] = -80 + 20w_0 + 60w_1
\end{aligned}
$$

Example of gradient descent

So the gradient is:

$$\nabla \text{SSE} = \begin{bmatrix} \frac{\partial \text{SSE}}{\partial w_0} \\ \frac{\partial \text{SSE}}{\partial w_1} \end{bmatrix} = \begin{bmatrix} -32 + 10w_0 + 20w_1 \\ -80 + 20w_0 + 60w_1 \end{bmatrix}$$

Let $w^{(0)} = (0, 0)$. Then the gradient evaluated in the point $w^{(0)}$ is:

$$\nabla \text{SSE}(w^{(0)}) = \begin{bmatrix} -32 + 10 \times 0 + 20 \times 0 \\ -80 + 20 \times 0 + 60 \times 0 \end{bmatrix} = \begin{bmatrix} -32 \\ -80 \end{bmatrix}$$

Example of gradient descent

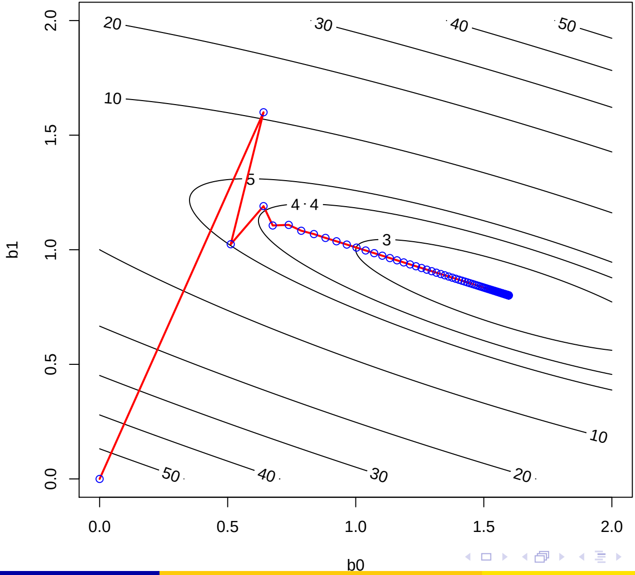
Let $\eta = \frac{1}{50}$. Then we get the following update:

$$w_0^{(1)} = w_0^{(0)} - \eta \frac{\partial \text{SSE}}{\partial w_0} = 0 - \frac{1}{50} \times -32 = 0.64$$
$$w_1^{(1)} = w_1^{(0)} - \eta \frac{\partial \text{SSE}}{\partial w_1} = 0 - \frac{1}{50} \times -80 = 1.6$$

Or both at once:

$$w^{(1)} = w^{(0)} - \eta \frac{\partial \text{SSE}}{\partial w} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \frac{1}{50} \begin{bmatrix} -32 \\ -80 \end{bmatrix} = \begin{bmatrix} 0.64 \\ 1.6 \end{bmatrix}$$

Gradient Descent with step size $\eta = 0.02$



b1
b0
0.0
0.5
1.0
1.5
2.0
0.0
0.5
1.0
1.5
2.0
20
30
40
50
10
5
4
4
3
10
50
40
30
20