

# Statistical Machine Learning 2018

## Assignment 3

Deadline: 25th of November 2018

Christoph Schmidl

s4226887

c.schmidl@student.ru.nl

Mark Beijer

s4354834

mbeijer@science.ru.nl

November 15, 2018

### Exercise 1 - The faulty lighthouse (weight 5)

A lighthouse is somewhere off a piece of straight coastline at a position  $\alpha$  along the shore and a distance  $\beta$  out to sea. Due to a technical fault, as it rotates the light source only occasionally and briefly flickers on and off. As a result it emits short, highly focused beams of light at random intervals. These pulses are intercepted on the coast by photo-detectors that record only the fact that a flash has occurred, but not the angle from which it came. So far,  $N$  flashes have been recorded at positions  $\mathcal{D} = x_1, \dots, x_N$ . Where is the lighthouse?

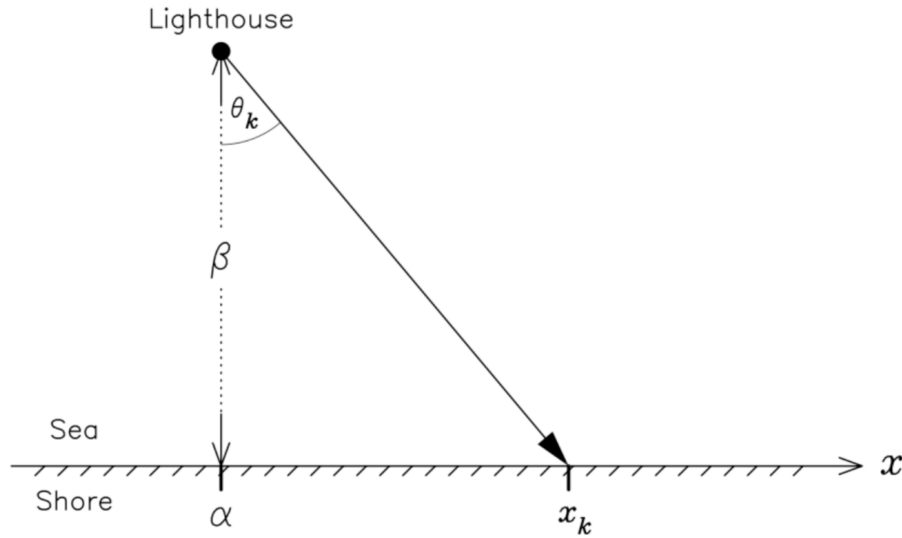


Figure 1: Geometry of the lighthouse problem.

#### Part 1 - Constructing the model

##### 1.1.1

Let  $\theta_k$  be the (unknown) angle for the k-th recorded flash, see fig.1. Argue why

$$p(\theta_k | \alpha, \beta) = \frac{1}{\pi} \quad (1)$$

would be a reasonable distribution over  $\theta_k$  between  $\pm \frac{\pi}{2}$  (zero otherwise).

**Answer:**

### 1.1.2

We only have the position  $x_k$  of the detector that recorded flash  $k$ , but we can relate this to the unknown  $\theta_k$  via elementary geometry as

$$\beta \tan(\theta_k) = x_k - \alpha \quad (2)$$

Show that the expected distribution over  $x$  given  $\alpha$  and  $\beta$  can be written as

$$p(x_k|\alpha, \beta) = \frac{\beta}{\pi[\beta^2 + (x_k - \alpha)^2]} \quad (3)$$

by using (2) to substitute variable  $x_k$  for  $\theta_k$  in the distribution (1). Plot the distribution for  $\beta = 1$  and a particular value of  $\alpha$ .

Hint: use the Jacobian  $|\frac{d\theta}{dx}|$  (Bishop, p.18) and the fact that  $(\tan^{-1}x)' = \frac{1}{1+x^2}$ .

**Answer:**

### 1.1.3

Inferring the position of the lighthouse corresponds to estimating  $\alpha$  and  $\beta$  from the data  $\mathcal{D}$ . This is still quite difficult, but if we assume that  $\beta$  is known, then from Bayes theorem we know that  $p(\alpha|\mathcal{D}, \beta) \propto p(\mathcal{D}|\alpha, \beta)p(\alpha|\beta)$ . We have no a priori knowledge about the position  $\alpha$  along the coast other than that it should not depend on the distance out at sea.

Show that with these assumptions the log of the posterior density can be written as

$$L = \ln(p(\alpha|\mathcal{D}, \beta)) = \text{constant} - \sum_{k=1}^N \ln[\beta^2 + (x_k - \alpha)^2] \quad (4)$$

and give an expression for the value  $\hat{\alpha}$  that maximizes this posterior density.

**Answer:**

### 1.1.4

Suppose we have a data set (in km) of  $\mathcal{D} = \{3.6, 7.7, -2.6, 4.9, -2.3, 0.2, -7.3, 4.4, 7.3, -5.7\}$ . We also assume that the distance  $\beta$  from the shore is known to be 2 km. As it is difficult to find a simple expression for the value of  $\hat{\alpha}$  that maximizes (4), we try an alternative approach instead.

Plot  $p(\alpha|\mathcal{D}, \beta = 2)$  as a function of  $\alpha$  over the interval  $[-10, 10]$ . What is your most likely estimate for  $\hat{\alpha}$  based on this graph? Compare with the mean estimate of the dataset. Can you explain the difference?

**Answer:**

## Part 2 - Generate the lighthouse data

We will try to solve the original problem by letting matlab/Python find the lighthouse for us. For that we first need a data set.

### 1.2.1

Sample a random position  $(\alpha_t, \beta_t)$  from a uniform distribution over an interval of 10 km along the coast and between 2 and 4 km out to sea.

**Answer:**

### 1.2.2

From this position generate a data set  $\mathcal{D} = \{x_1, \dots, x_N\}$  of 500 flashes in random directions that have been registered by a detector at point  $x_i$  along the coast. Assume that the flashes are i.i.d. according to (1).

**Answer:**

### 1.2.3

Make a plot of the mean of the data set as a function of the number of points. Compare with the true position of the lighthouse  $\alpha_t$ . How many points do you expect to need to obtain a reasonable estimate of  $\alpha_t$  from the mean? Explain.

**Answer:**

## Part 3 - Find the lighthouse

From the analysis in the first part we know that trying to find a maximum likelihood estimate in the usual way is possible (compute gradient, set equal to zero and solve), but that this does not result in a nice closed-form expression for the solution, even when one of the parameters is assumed to be known. As we want to find estimates of both  $\alpha$  and  $\beta$  from the data, we will try a different approach instead.

### 1.3.1

Use (3) to get an expression for the loglikelihood of the data  $\mathcal{D}$  as a function of  $\alpha$  and  $\beta$ .

**Answer:**

### 1.3.2

We can see how this likelihood (as a function of  $\alpha$  and  $\beta$ ) changes, as data points come in.

Process your data set  $\mathcal{D}$  one by one and make a plot of the (log)likelihood after one, two, three, and 20 points have arrived, respectively. Explain what happens.

Hint: Create a function that calculates the (log)likelihood at a specific point  $(\alpha, \beta)$  after the first  $k$  data points  $\{x_1, \dots, x_k\}$  have come in. Use this with the matlab `meshgrid` and `surf` functions to make plots over the interval  $[-10 \leq \alpha \leq +10] \times [0 \leq \beta \leq 5]$ . Decide if/when it makes more sense to use the likelihood directly or the log of the likelihood.

**Answer:**

### 1.3.3

We can make a reasonable (visual) estimate of the most probable position of the lighthouse from the graph, after a few data points have been observed. However, as we are working with a computer, we will let matlab do the dirty work for us.

Create a function that uses matlab function `fminsearch` to compute the values of  $\alpha$  and  $\beta$  that maximize the likelihood for a data set of  $k$  points, and plot these as a function of the number of points. Use  $[0, 1]$  as the initial starting value for `fminsearch` (see examples in matlab-help). Compare your final estimate with the true values  $(\alpha_t, \beta_t)$ .

**Answer:**

## Exercise 2 - Bayesian linear regression (weight 2)

This exercise builds on exercise 2, week 7, Fitting a straight line to data. For a detailed description (and explanation) see Exercises and Answers, Week 7 in Brightspace. The final part of that exercise computed the predictive distribution after a single data point was observed. Here we consider a new data set, consisting of no less than two points:  $x_1, t_1 = (0.4, 0.1)$  and  $x_2, t_2 = (0.6, -0.4)$ .

### 2.1

Assume  $\alpha = 1$  and  $\beta = 15$ . Compute the predictive distribution  $p(t|x, \mathbf{t}, \mathbf{x}, \alpha, \beta)$  after these two points are observed.

**Answer:**

### 2.2

Plot the mean of the predictive Gaussian distribution and one standard deviation on both sides as a function of  $x$  over the interval  $[0, 1]$ . Plot the data in the same figure. See `a009plotideas.m` in Brightspace for some plotting hints. Compare your plot with Figure 3.8b (Bishop, p.157) and explain the difference.

**Answer:**

### 2.3

Sample five functions  $y(x, \mathbf{w})$  from the posterior distribution over  $\mathbf{w}$  for this data set and plot them in the same graph (i.e. with the predictive distribution). You may use the Matlab function `mvnrnd`. See again `a009plotideas.m` for some plotting hints.

**Answer:**

## Exercise 3 - Gradient descent revisited (weight 3)

In this exercise, we will have a closer look at the gradient descent algorithm for function minimization. When the function to be minimized is  $E(x)$ , the gradient descent iteration is

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \eta \nabla E(\mathbf{x}_n) \quad (5)$$

where  $\eta > 0$  is the so-called learning-rate.

### 3.1

Consider the function  $f(x) = \frac{\lambda}{2}(x - a)^2$  with parameters  $\lambda > 0$ , and  $a$  arbitrary.

#### 3.1a

Write down the gradient descent iteration rule. Verify that the minimum of  $f$  is a fixed point<sup>1</sup> of the gradient descent iteration rule.

**Answer:**

The gradient of  $f$  is:

$$\nabla f(x) = \lambda(x - a) \quad (6)$$

And therefore the gradient descent iteration rule is:

$$x_{n+1} = x_n - \eta \lambda(x_n - a) \quad (7)$$

---

<sup>1</sup>A fixed point  $x^*$  of an iteration  $x_{n+1} = F(x_n)$  satisfies  $x^* = F(x^*)$ .

The minimum of  $f$  can be found by setting its gradient to 0:

$$\lambda(x - a) = 0 \quad (8)$$

$$x - a = 0 \quad (9)$$

$$x = a \quad (10)$$

To show it's a minimum we can take the second derivative at point  $a$ :

$$f''(x) = \lambda \rightarrow f''(a) = \lambda \quad (11)$$

Since  $\lambda$  is strictly greater than 0, the second derivative at  $a$  is strictly greater than 0 and therefore  $a$  is the minimum.

$a$  is a fixed point since:

$$x_{n+1} = a - \eta \nabla f(a) \quad (12)$$

$$= a - \eta \lambda(a - a) = a \quad (13)$$

### 3.1b

Find  $\eta$  for which convergence is the fastest (actually in one step).

**Answer:**

Since it is given that the fastest is in one step, we can write :

$$a = x - \eta \lambda f(x) \quad (14)$$

$$= x - \eta \lambda(x - a) \quad (15)$$

$$\eta \lambda(x - a) = x - a \quad (16)$$

$$\eta \lambda = 1 \quad (17)$$

$$\eta = \frac{1}{\lambda} \quad (18)$$

You can also see this by looking at the iteration rule. Since when  $\eta = \lambda^{-1}$ , in the next iteration  $x_n$  will cancel and you're left with  $(-(-a))=a$ , thus arriving at the minimum in one step, independent of  $x$ .

### 3.1c

We will investigate the convergence properties for different values of  $\eta$ . For this we look at the ratio of the distance to the fixed point after the step and the distance before the step:

$$r_n = \frac{|x_{n+1} - x^*|}{|x_n - x^*|} \quad (19)$$

#### 3.1c.i

What does it mean if all  $r_n < c < 1$  (i.e. all ratios are smaller than a certain number below 1). Give for this case the (optimal) upper bound of the distance  $|x_n - x^*|$  in terms of  $|x_0 - x^*|$ ,  $c$  and  $n$ .

**Answer:**

If for all  $r_n < c < 1$  then it means that always, independent of the iteration step, beginning starting point  $x_0$ , the ratios of the next iteration and the current are always lower than some number strictly smaller than 1. This means that the distance  $r_n$  to the fixed point always decreases with a rate bounded by  $c$ . It also means that every new step brings you closer to  $a$ , it's not like there is some other fixed point you might end up at.

At every step the distance is multiplied by at least  $c$ , since the difference between the distance of the current and next step is  $c$  or lower. So the upper bound on the distance would be:

$$|x_n - x^*| = c^n |x_0 - x^*| \quad (20)$$

Note that  $c^n < 1$  since  $0 < c < 1$ .

### 3.1c.ii

What is the consequence if all  $r_n > c > 1$ ? Give for this case the optimal lower bound of the distance  $|x_n - x^*|$  in terms of  $|x_0 - x^*|$ ,  $c$  and  $n$ .

**Answer:**

Then with every step you get further and further away from the optimal point  $x^*$ . Therefore you'll diverge from the point  $x^*$ . The constant  $c$  is larger than 1, so it is bounded by 1. While  $c$  can't be 1, it is the lower bound. Taking this we get that the best case scenario the distance won't change, so we get:

$$|x_n - x^*| = |x_0 - x^*| + \epsilon \quad (21)$$

With  $0 < \epsilon$ .

### 3.1d

Show that in our case,  $r_n = |1 - \eta\lambda| \equiv r$ , independent of  $n$  (we refer to  $r$  as the convergence rate). For which  $\eta$  is the algorithm convergent?

**Answer:**

$$r_n = \frac{|x_{n+1} - x^*|}{|x_n - x^*|} \quad (22)$$

$$= \frac{|x_n - \eta\lambda(x_n - a) - a|}{|x_n - a|} \quad (23)$$

$$= \frac{|(1 - \eta\lambda)(x_n - a)|}{|x_n - a|} \quad (24)$$

$$= |1 - \eta\lambda| \quad (25)$$

This converges for when  $r_n$  is strictly smaller than 1, so when  $0 < \eta\lambda < 2$ . So:

$$0 < \eta < \frac{2}{\lambda} \quad (26)$$

## 3.2

Consider the function  $g(x, y) = \frac{\lambda_1}{2}(x - a_1)^2 + \frac{\lambda_2}{2}(y - a_2)^2$  with parameters  $0 < \lambda_1 \leq \lambda_2$ , and  $a_i$  arbitrary.

### 3.2a

Write down the gradient descent iteration rule. Verify that the minimum of  $f$  is a fixed point.

**Answer:**

The gradient of the function  $g$  is:

$$\frac{\partial g}{\partial x} = \lambda_1(x - a_1) \quad (27)$$

$$\frac{\partial g}{\partial y} = \lambda_2(y - a_2) \quad (28)$$

$$\nabla g = \begin{bmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{bmatrix} = \begin{bmatrix} \lambda_1(x - a_1) \\ \lambda_2(y - a_2) \end{bmatrix} \quad (29)$$

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} - \eta \nabla g(x_n) \quad (30)$$

$$= \begin{bmatrix} x_n \\ y_n \end{bmatrix} - \eta \begin{bmatrix} \lambda_1(x_n - a_1) \\ \lambda_2(y_n - a_2) \end{bmatrix} \quad (31)$$

The minimum of  $g$  is: (I think the  $f$  was a typing error)

$$\nabla g = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (32)$$

$$= \begin{bmatrix} \lambda_1(x - a_1) \\ \lambda_2(y - a_2) \end{bmatrix} \quad (33)$$

$$\lambda_1(x - a_1) = 0 \rightarrow x = a_1 \quad (34)$$

$$\lambda_2(y - a_2) = 0 \rightarrow y = a_2 \quad (35)$$

$$(36)$$

Now to show it is a fixed point:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} - \eta \begin{bmatrix} \lambda_1(a_1 - a_1) \\ \lambda_2(a_2 - a_2) \end{bmatrix} \quad (37)$$

$$= \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \quad (38)$$

### 3.2b

Show that  $\eta = \frac{2}{\lambda_2 + \lambda_1}$  minimizes the larger ratio between  $r_{n,x} = \frac{|x_{n+1} - x^*|}{|x_n - x^*|}$  and  $r_{n,y} = \frac{|y_{n+1} - y^*|}{|y_n - y^*|}$ , i.e.,  $\arg \min_{\eta} \{ \max\{r_{n,x}, r_{n,y}\} \} = \frac{2}{\lambda_2 + \lambda_1}$ . What happens if  $\eta$  is smaller than this optimal value? What happens if it is larger?

**Answer:**

For the first ratio:

$$r_{n,x} = \frac{|x_{n+1} - x^*|}{|x_n - x^*|} \quad (39)$$

$$= \frac{|x_n - \eta \lambda_1(x_n - a_1) - a_1|}{|x_n - a_1|} \quad (40)$$

$$= \frac{|(1 - \eta \lambda_1)(x_n - a_1)|}{|x_n - a_1|} \quad (41)$$

$$= |1 - \eta \lambda_1| \quad (42)$$

For the second ratio:

$$r_{n,y} = \frac{|y_{n+1} - y^*|}{|y_n - y^*|} \quad (43)$$

$$= \frac{|y_n - \eta \lambda_2(y_n - a_2) - a_2|}{|y_n - a_2|} \quad (44)$$

$$= \frac{|(1 - \eta \lambda_2)(y_n - a_2)|}{|y_n - a_2|} \quad (45)$$

$$= |1 - \eta \lambda_2| \quad (46)$$

When they are at the minimum, the values are equal. For if one is not equal but higher you can always go in one direction to lower the first until it lowers so much it equals the second. This is easier to understand by looking at an example plot, here  $\lambda_1 = 1$  and  $\lambda_2 = 2$ .

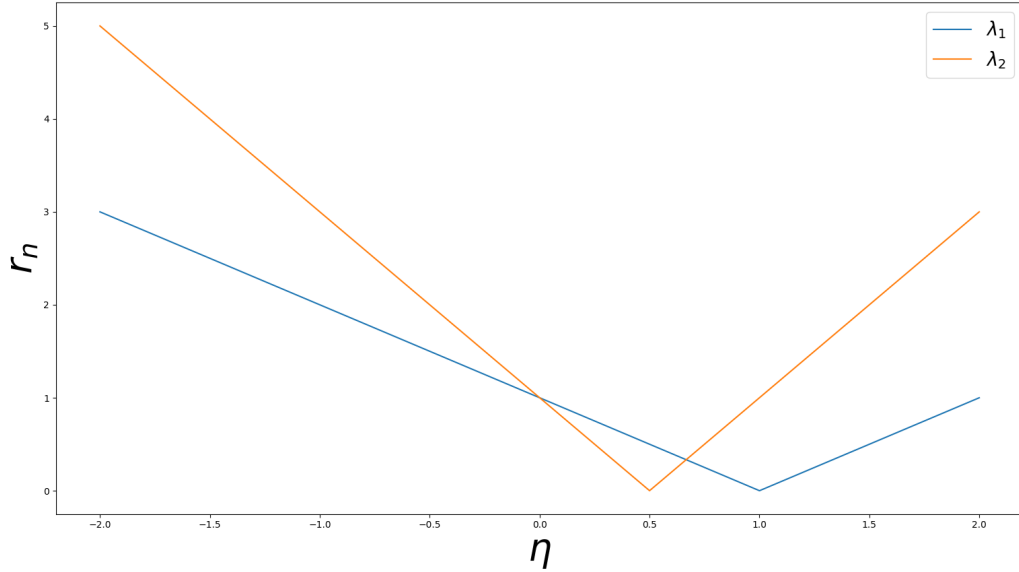


Figure 2: Two ratios for  $\lambda_1 = 1$  and  $\lambda_2 = 2$ . Notice that the optimal value is found at the second intersection, where  $\eta \neq 0$ , for the highest value of both of the functions is the lowest, moving left or right of that point either one is higher then at the intersection.

$$|1 - \eta\lambda_1| = |1 - \eta\lambda_2| \quad (47)$$

$$(1 - \eta\lambda_1)^2 = (1 - \eta\lambda_2)^2 \quad (48)$$

$$1 - \eta\lambda_1 = \pm(1 - \eta\lambda_2) \quad (49)$$

$$1 - \eta\lambda_1 = 1 - \eta\lambda_2 \quad (50)$$

$$\eta\lambda_2 - \eta\lambda_1 = 0 \quad (51)$$

$$\eta = 0 \quad \text{Not the lowest solution, let's try the -} \quad (52)$$

$$1 - \eta\lambda_1 = \eta\lambda_2 - 1 \quad (53)$$

$$2 = \eta(\lambda_1 + \lambda_2) \quad (54)$$

$$\eta = \frac{2}{\lambda_1 + \lambda_2} \quad (55)$$

We can say without loss of generality that  $\lambda_1 < \lambda_2$ . If that's not the case we can either swap labels. If they are the same then the ideal rate is optimal in both dimensions, and using an  $\eta$  different then the optimal will decrease the converging rate for both directions x and y.

As we can see in figure 2, a value of  $\eta$  which is lower then the optimal will make y converge faster, though if it's negative it would make x diverge not as fast. If  $\eta$  is greater then the optimal value, x will converge faster and after some point diverge slower then y.



### 3.2c

What is the convergence rate for this  $\eta$ ? What does this say about the applicability of gradient descent to functions with steep hills and flat valleys (i.e., if  $\lambda_2 \gg \lambda_1$ )?

**Answer:**

The convergence rate for x and y will be:

$$r_{n,x} = \left| 1 - \frac{2\lambda_1}{\lambda_1 + \lambda_2} \right| \quad (56)$$

$$r_{n,y} = \left| 1 - \frac{2\lambda_2}{\lambda_1 + \lambda_2} \right| \quad (57)$$

In the case that  $\lambda_2 \gg \lambda_1$ , we can say that  $\lambda_1 = \epsilon\lambda_2$ . Then we can take the limit for  $\epsilon \rightarrow 0$ .

$$\lim_{\epsilon \rightarrow 0} (1 + \epsilon) = 1 \quad (58)$$

$$r_{n,x} = \left| 1 - \frac{2\epsilon\lambda_2}{(1 + \epsilon)\lambda_2} \right| \quad (59)$$

$$\lim_{\epsilon \rightarrow 0} r_{n,x} = \lim_{\epsilon \rightarrow 0} \left| 1 - \frac{2\epsilon\lambda_2}{(1 + \epsilon)\lambda_2} \right| \quad (60)$$

$$= \lim_{\epsilon \rightarrow 0} \left| 1 - \frac{2\epsilon\lambda_2}{\lambda_2} \right| \quad (61)$$

$$= 1 \quad (62)$$

Now for y:

$$r_{n,y} = \left| 1 - \frac{2\lambda_2}{(1 + \epsilon)\lambda_2} \right| \quad (63)$$

$$\lim_{\epsilon \rightarrow 0} r_{n,y} = \lim_{\epsilon \rightarrow 0} \left| 1 - \frac{2\lambda_2}{(1 + \epsilon)\lambda_2} \right| \quad (64)$$

$$= |1 - 2| = 1 \quad (65)$$

So it won't converge in either direction.

### 3.2d

Implement the gradient descent algorithm for g in matlab.

**Answer:**

### 3.2e

Make some plots of the trajectories  $\{(x_n, y_n)\}_{n=0}^N$  for different values of  $\lambda_i$  and  $\eta$  ( $\eta$  optimal, larger than optimal, and smaller than optimal) to illustrate what is going on. Plot these trajectories on top of a contour plot of g. Monitor the convergence rates. Explain what happens.

**Answer:**

Let's first take an example from what we saw in 3.2c. We've chosen  $\lambda_1 = 10^7$  and  $\lambda_2 = 1$ . We've taken  $\eta$  to be the optimal value.  $x_0 = 3$  and  $y_0 = 4$  for all the situations.

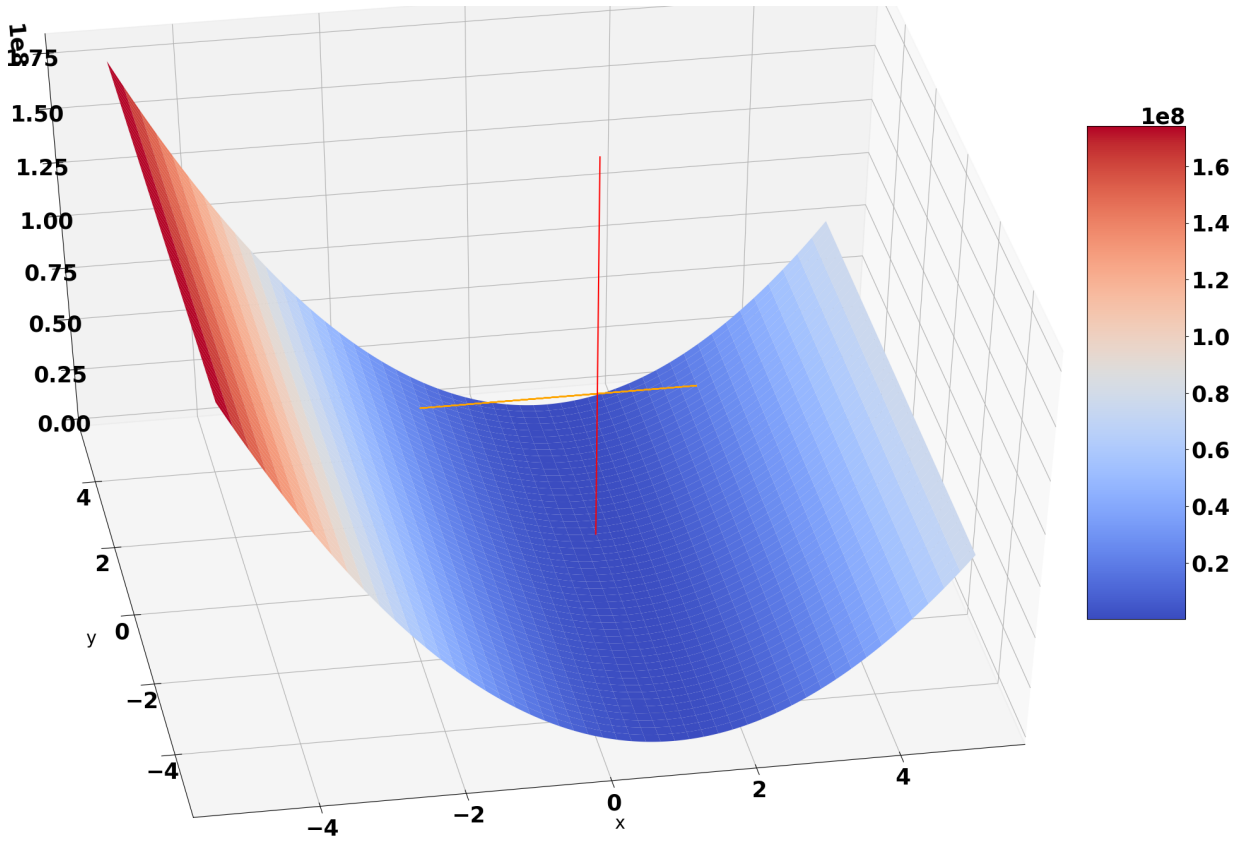


Figure 3: Gradient descent for high  $\lambda_1$ . Notice how high the hill is ( $10^8$ ). In the x direction it hops between  $\approx 3$  and  $\approx -1$ . y stays at  $\approx 4$ .

What happens at 3 is that in x,  $\eta\lambda$  equals 2. So it would need to go to  $x=1$ , but it overshoots by going exactly twice the required amount. So it will land on  $3 - (2 \cdot -2) = -1$ . From there it will overshoot going back and it will land again on  $x \approx 2$ . For y  $\eta\lambda$  almost equals 0, so it doesn't go anywhere and stays at  $y=4$ . I use approximate because it doesn't go exactly to -1, it first for examples goes to -0.9999996000000402. This is because it will have this behaviour in the event of  $\lambda_1 \rightarrow \infty$  and we're not quite there yet for  $\lambda_1 = 10^7$ , but we can get arbitrary close to it if we make  $\lambda_1$  larger. (This all assumes  $\lambda_2$  stays the same)

Another way to make it stuck forever (this time truly forever) is by choosing  $\eta$  in such a way that it will always overshoot. If we take  $\lambda_1 = \lambda_2 = \lambda$ , then  $\eta = \frac{2}{\lambda}$  will give the desired result, since then  $r_n = |1 - \eta\lambda| = |1 - 2| = 1$