Maximum Likelihood Estimation

At the core of GraphSLAM is graph optimization - the process of minimizing the error present in all of the constraints in the graph. Let's take a look at what these constraints look like, and learn to apply a principle called *maximum likelihood estimation* (MLE) to structure and solve the optimization problem for the graph.

Likelihood

Likelihood is a complementary principle to probability. While probability tries to estimate the outcome given the parameters, likelihood tries to estimate the parameters that best explain the outcome. For example,

> **Probability:** What is the probability of rolling a 2 on a 6-sided die?

(Answer: 1/6)

> **Likelihood:** I've rolled a die 100 times, and a 2 was rolled 10% of the time, how many sides does my die have?

(Answer: 10 sides)

When applied to SLAM, likelihood tries to estimate the most likely configuration of state and feature locations given the motion and measurement observations.

Probability & Likelihood Quiz

QUIZ QUESTION

To solidify your understanding of the difference between probability and likelihood, label the following problems with their respective terms.

Submit to check your answer choices!

EXAMPLE	TYPE OF PROBLEM
The chance of being selected for a Udacity scholarship is 10% if there are 1000 applicants for 100 spots (assuming that everyone is equally qualified).	Probability
The weather forecast predicts a 30% chance of rain for this evening.	Probability
The grass is wet, and I didn't water it. It's likely that it rained earlier today.	Likelihood
A fair coin has a 50% chance of coming up heads, and 50% chance of coming up tails.	Probability
A coin was tossed 1,000 times, and came up heads 756 times. It is likely that the coin is rigged.	Likelihood

SUBMIT

Feature Measurement Example

Let's look at a very simple example - one where our robot is taking repeated measurements of a feature in the environment. This example will walk you through the steps required to solve it, which can then be applied to more complicated problems.

Let's look at a very simple example - one where our robot is taking repeated measurements of a feature in the environment. This example will walk you through the steps required to solve it, which can then be applied to more complicated problems.

The robot's initial pose has a variance of 0 - simply because this is its start location. Recall that wherever the start location may be - we call it location 0 in our relative map.

Every action pose and measurement hereafter will be uncertain. In GraphSLAM, we will continue to make the assumption that motion and measurement data has Gaussian noise.

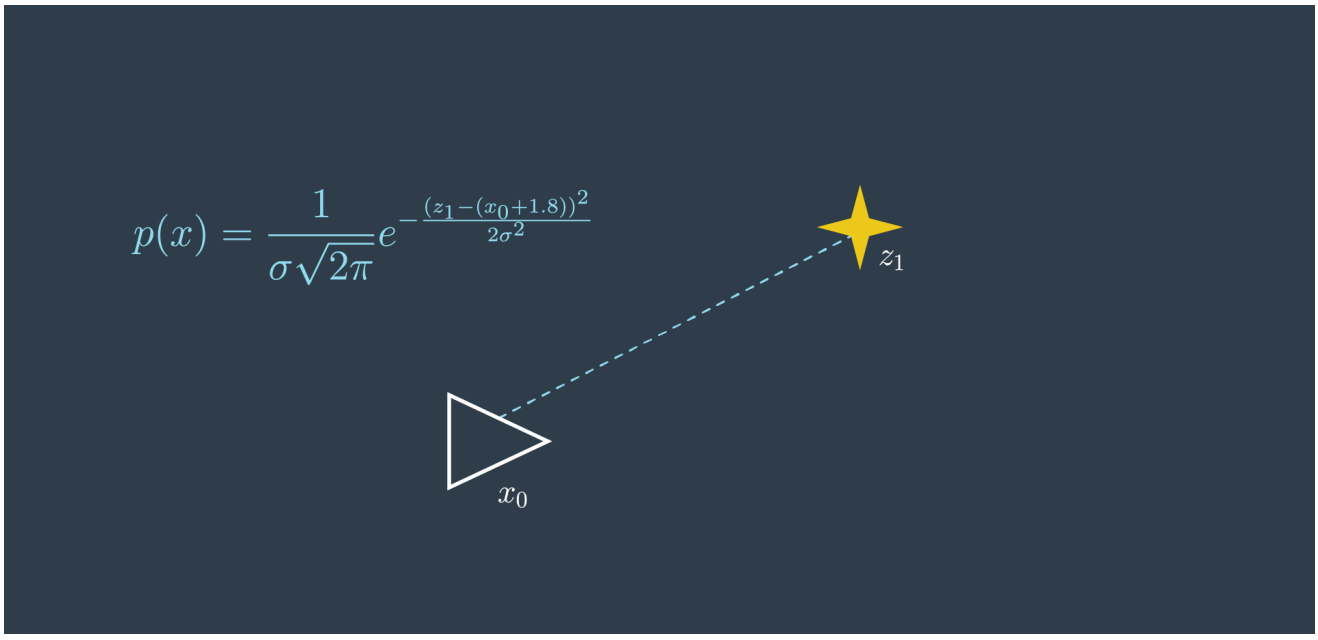
The robot takes a measurement of a feature, m_1, and it returns a distance of 1.8 metres.

If we return to our spring analogy - 1.8m is the spring's resting length. This is the spring's most desirable length; however, it is possible for the spring to be compressed or elongated to accommodate other forces (constraints) that are acting on the system.

This probability distribution for this measurement can be defined as so,

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(z_1-(x_0+1.8))^2}{\sigma^2}}$$

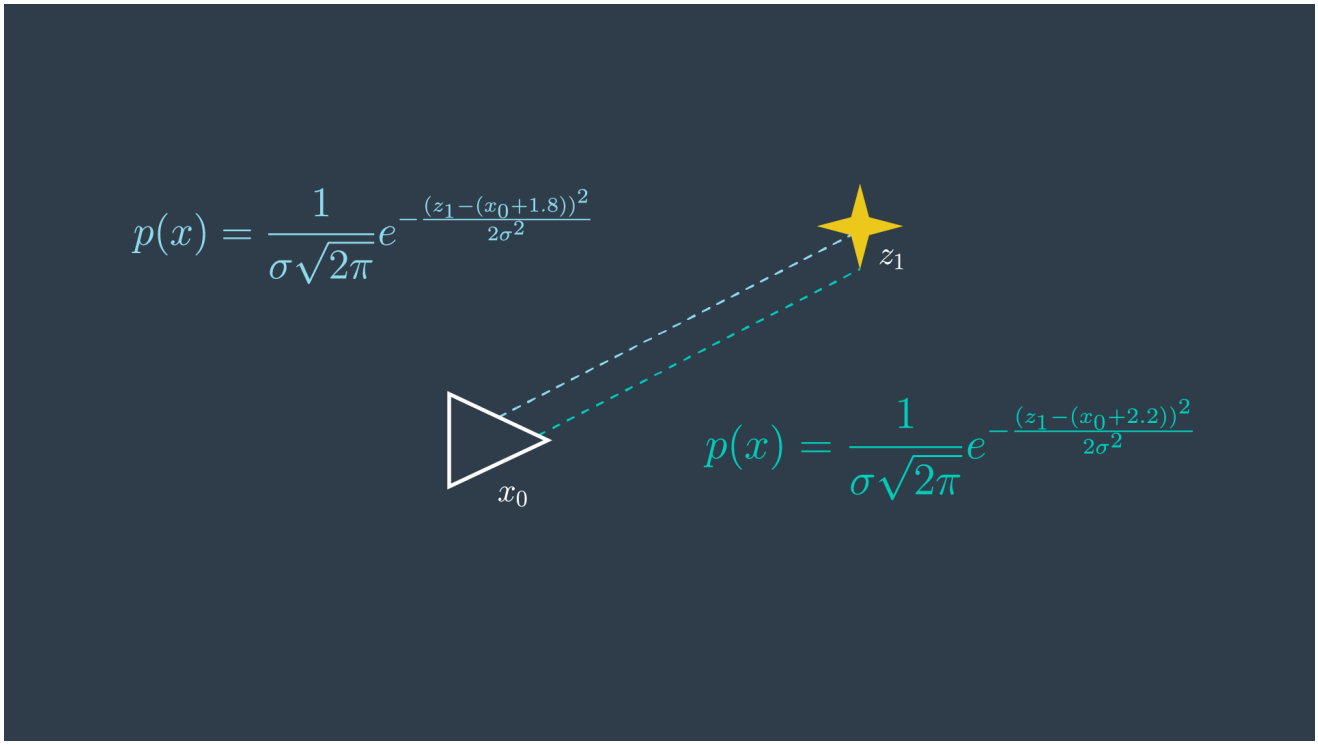
In simpler terms, the probability distribution is highest when z1 and x0 are 1.8 meters apart.



However, since the location of the first pose, x_0 is set to 0, this term can simply be removed from the equation.

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(z_1 - 1.8)^2}{\sigma^2}}$$

Next, the robot takes another measurement of the same feature in the environment. This time, the data reads 2.2m. With two conflicting measurements, this is now an overdetermined system - as there are more equations than unknowns!



With two measurements, the most probable location of the feature can be represented by the product of the two probabilities.

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(z_1 - 1.8)^2}{\sigma^2}} * \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(z_1 - 2.2)^2}{\sigma^2}}$$

In this trivial example, it is probably quite clear to you that the most likely location of the feature is at the 2.0 meter mark. However, it is valuable to go through the maximum

$$\sigma\sqrt{2\pi} \qquad \sigma\sqrt{2\pi}$$

In this trivial example, it is probably quite clear to you that the most likely location of the feature is at the 2.0 meter mark. However, it is valuable to go through the maximum likelihood estimation process to understand the steps entailed, to then be able to apply it to more complicated systems.

To solve this problem analytically, a few steps can be taken to reduce the equations into a simpler form.

Remove Scaling Factors

The value of m that maximizes the equation does not depend on the constants in front of each of the exponentials. These are scaling factors, however in SLAM we are not

usually interested in the absolute value of the probabilities, but finding the maximum likelihood estimate. For this reason, the factors can simply be removed.

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(z_1-1.8)^2}{\sigma^2}} * \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(z_1-2.2)^2}{\sigma^2}}$$

Log-Likelihood

The product of the probabilities has been simplified, but the equation is still rather complicated - with exponentials present. There exists a mathematical property that can be applied here to convert this product of exponentials into the sum of their exponents.

First, we must use the following property, $e^a e^b = e^{a+b}$, to combine the two exponentials into one.

$$e^{-\frac{1}{2}\frac{(z_1-1.8)^2}{\sigma^2}} * e^{-\frac{1}{2}\frac{(z_1-2.2)^2}{\sigma^2}}$$

$$= e^{-\frac{1}{2}\frac{(z_1-1.8)^2}{\sigma^2} - \frac{1}{2}\frac{(z_1-2.2)^2}{\sigma^2}}$$

Next, instead of working with the likelihood, we can take its natural logarithm and work with it instead.

$$\ln\left(e^{-\frac{1}{2}\frac{(z_1-1.8)^2}{\sigma^2} - \frac{1}{2}\frac{(z_1-2.2)^2}{\sigma^2}}\right)$$

$$e^{-\frac{1}{2}\frac{(z_1-1.8)^2}{\sigma^2}} * e^{-\frac{1}{2}\frac{(z_1-2.2)^2}{\sigma^2}}$$

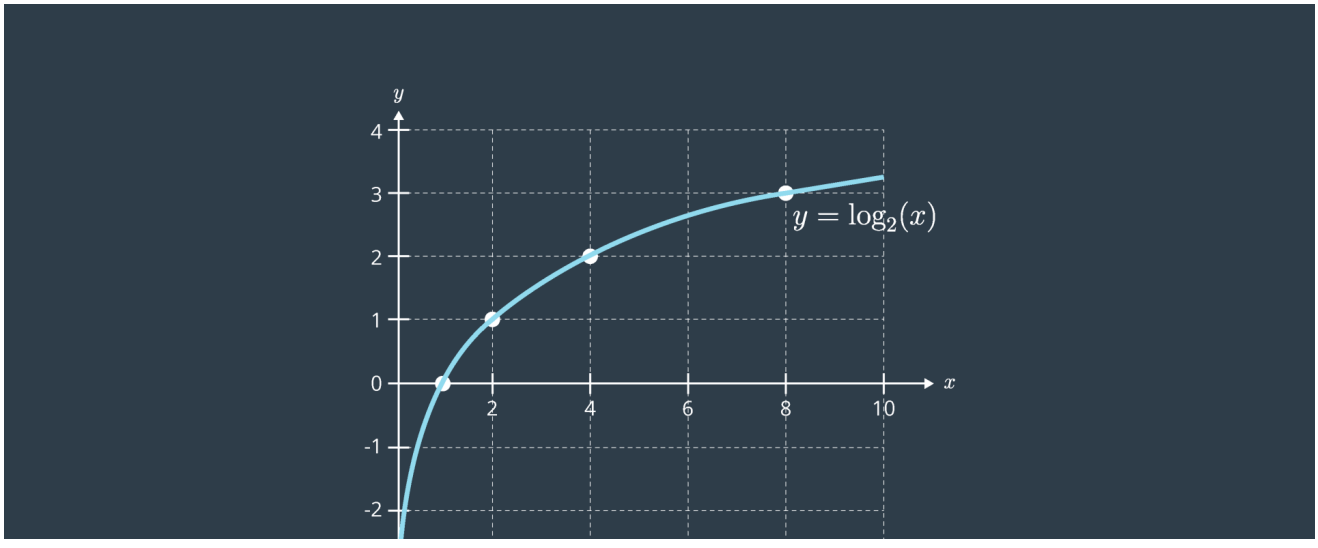
$$= e^{-\frac{1}{2}\frac{(z_1-1.8)^2}{\sigma^2} - \frac{1}{2}\frac{(z_1-2.2)^2}{\sigma^2}}$$

Next, instead of working with the likelihood, we can take its natural logarithm and work with it instead.

$$ln(e^{-\frac{1}{2}\frac{(z_1-1.8)^2}{\sigma^2} - \frac{1}{2}\frac{(z_1-2.2)^2}{\sigma^2}})$$

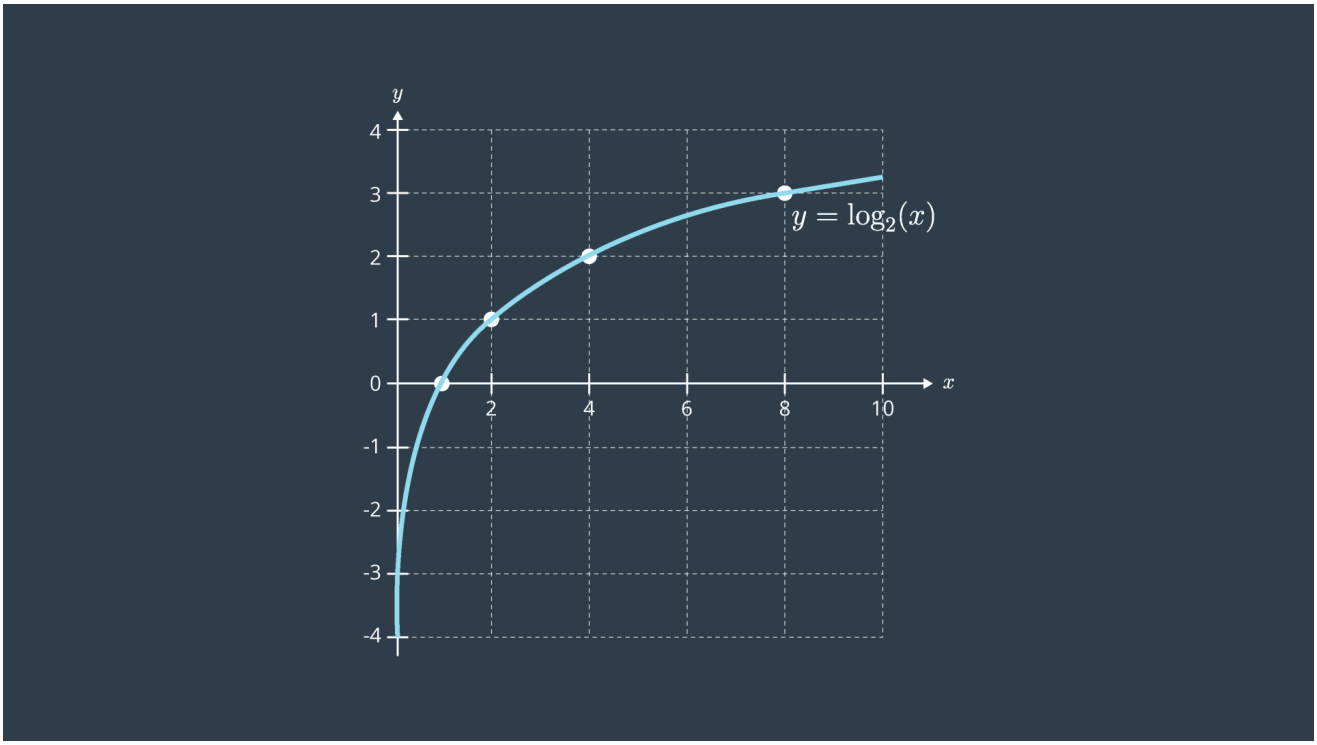
$$= -\frac{1}{2}\frac{(z_1-1.8)^2}{\sigma^2} - \frac{1}{2}\frac{(z_1-2.2)^2}{\sigma^2}$$

The natural logarithm is a **monotonic function** - in the log's case - it is always increasing - as can be seen in the graph below. There can only be a one-to-one mapping of its values. This means that optimizing the logarithm of the likelihood is no different from maximizing the likelihood itself.



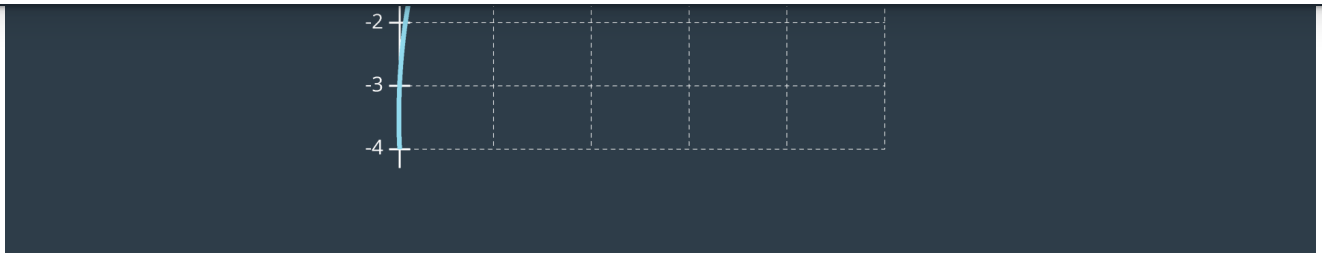
$2 \quad \sigma^2 \quad 2 \quad \sigma^2$

The natural logarithm is a **monotonic function** - in the log's case - it is always increasing - as can be seen in the graph below. There can only be a one-to-one mapping of its values. This means that optimizing the logarithm of the likelihood is no different from maximizing the likelihood itself.



One thing to note when working with logs of likelihoods, is that they are always negative. This is because probabilities assume values between 0 and 1, and the log of any value between 0 and 1 is negative. This can be seen in the graph above. For this reason, when working with log-likelihoods, optimization entails *minimizing* the *negative* log-likelihood; whereas in the past, we were trying to maximize the likelihood.

Lastly, as was done before, the constants in front of the equation can be removed without consequence. As well, for the purpose of this example, we will assume that the



One thing to note when working with logs of likelihoods, is that they are always negative. This is because probabilities assume values between 0 and 1, and the log of any value between 0 and 1 is negative. This can be seen in the graph above. For this reason, when working with log-likelihoods, optimization entails *minimizing* the *negative* log-likelihood; whereas in the past, we were trying to maximize the likelihood.

Lastly, as was done before, the constants in front of the equation can be removed without consequence. As well, for the purpose of this example, we will assume that the same sensor was used in obtaining both measurements - and will thus ignore the variance in the equation.

$$(z_1 - 1.8)^2 + (z_1 - 2.2)^2$$

Optimization

At this point, the equation has been reduced greatly. To get it to its simplest form, all that is left is to multiply out all of the terms.

$$2z_1^2 - 8z_1 + 8.08$$

To find the minimum of this equation, you can take the first derivative of the equation and set it to equal 0.

$$4z_1 - 8 = 0$$

$$4z_1 = 8$$

$$z_1 = 2$$

By doing this, you are finding the location on the curve where the slope (or *gradient*, in multi-dimensional equations) is equal to zero - the trough! This property can be

At this point, the equation has been reduced greatly. To get it to its simplest form, all that is left is to multiply out all of the terms.

$$2z_1^2 - 8z_1 + 8.08$$

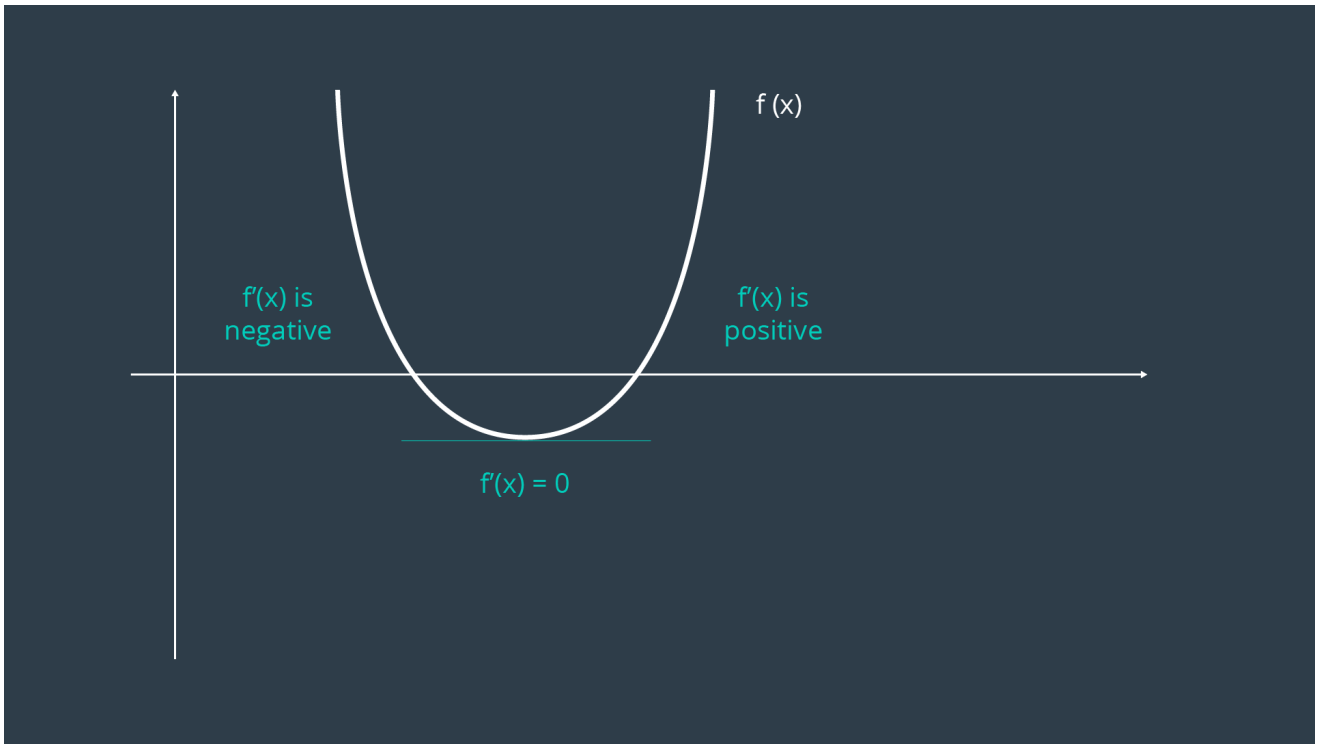
To find the minimum of this equation, you can take the first derivative of the equation and set it to equal 0.

$$4z_1 - 8 = 0$$

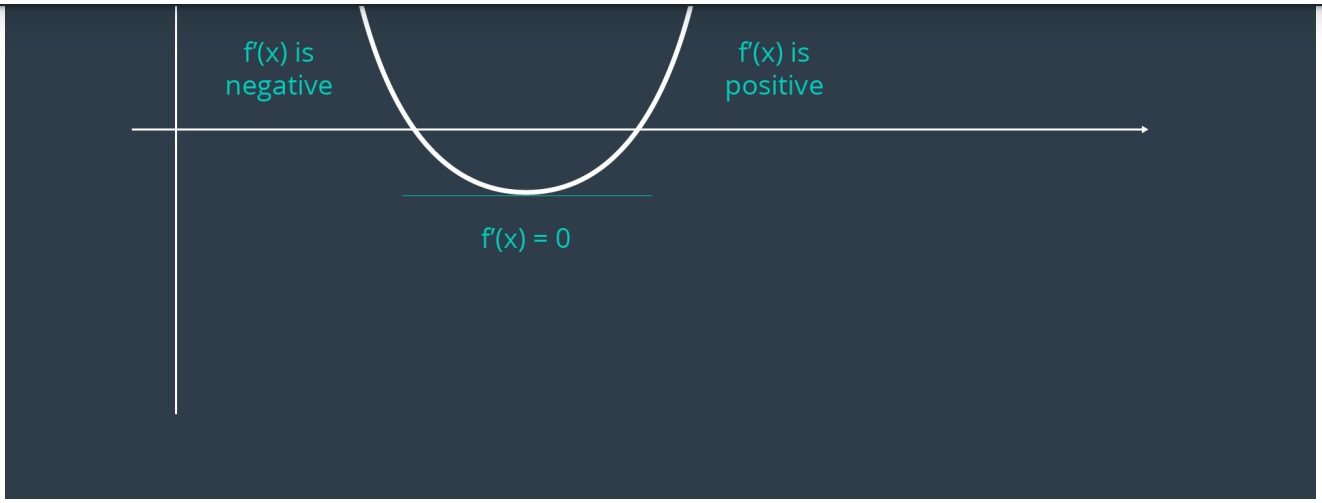
$$4z_1 = 8$$

$$z_1 = 2$$

By doing this, you are finding the location on the curve where the slope (or *gradient*, in multi-dimensional equations) is equal to zero - the trough! This property can be visualized easily by looking at a graph of the error function.



In more complex examples, the curve may be multimodal, or exist over a greater number of dimensions. If the curve is multimodal, it may be unclear whether the locations discovered by the first derivative are in fact troughs, or peaks. In such a case



In more complex examples, the curve may be multimodal, or exist over a greater number of dimensions. If the curve is multimodal, it may be unclear whether the locations discovered by the first derivative are in fact troughs, or peaks. In such a case, the second derivative of the function can be taken - which should clarify whether the local feature is a local minimum or maximum.

Overview

The procedure that you executed here is the *analytical* solution to an MLE problem. The steps included,

- Removing inconsequential constants,
- Converting the equation from one of *likelihood estimation* to one of *negative log-likelihood estimation*, and
- Calculating the first derivative of the function and setting it equal to zero to find the extrema.

In GraphSLAM, the first two steps can be applied to *every* constraint. Thus, any measurement or motion constraint can simply be labelled with its negative log-likelihood error. For a measurement constraint, this would resemble the following,

$$\frac{(z_t - (x_t + m_t))^2}{\sigma^2}$$

And for a motion constraint, the following,

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(z_1-1.8)^2}{\sigma^2}} * \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(z_1-2.2)^2}{\sigma^2}}$$

In this trivial example, it is probably quite clear to you that the most likely location of the feature is at the 2.0 meter mark. However, it is valuable to go through the maximum likelihood estimation process to understand the steps entailed, to then be able to apply it to more complicated systems.

To solve this problem analytically, a few steps can be taken to reduce the equations into a simpler form.

Remove Scaling Factors

The value of m that maximizes the equation does not depend on the constants in front of each of the exponentials. These are scaling factors, however in SLAM we are not usually interested in the absolute value of the probabilities, but finding the maximum likelihood estimate. For this reason, the factors can simply be removed.

$$\cancel{\frac{1}{\sigma\sqrt{2\pi}}} e^{-\frac{1}{2}\frac{(z_1-1.8)^2}{\sigma^2}} * \cancel{\frac{1}{\sigma\sqrt{2\pi}}} e^{-\frac{1}{2}\frac{(z_1-2.2)^2}{\sigma^2}}$$

Log-Likelihood

The product of the probabilities has been simplified, but the equation is still rather complicated - with exponentials present. There exists a mathematical property that can be applied here to convert this product of exponentials into the sum of their exponents.

First, we must use the following property, $e^a e^b = e^{a+b}$, to combine the two exponentials into one.

$$e^{-\frac{1}{2}\frac{(z_1-1.8)^2}{\sigma^2}} * e^{-\frac{1}{2}\frac{(z_1-2.2)^2}{\sigma^2}}$$

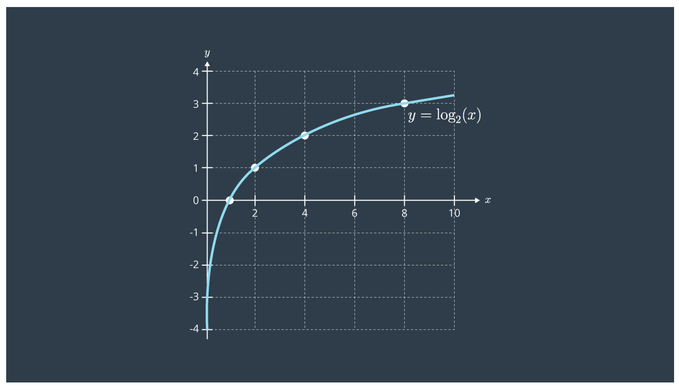
$$= e^{-\frac{1}{2}\frac{(z_1-1.8)^2}{\sigma^2} - \frac{1}{2}\frac{(z_1-2.2)^2}{\sigma^2}}$$

Next, instead of working with the likelihood, we can take its natural logarithm and work with it instead.

$$ln(e^{-\frac{1}{2}\frac{(z_1-1.8)^2}{\sigma^2} - \frac{1}{2}\frac{(z_1-2.2)^2}{\sigma^2}})$$

$$= -\frac{1}{2}\frac{(z_1-1.8)^2}{\sigma^2} - \frac{1}{2}\frac{(z_1-2.2)^2}{\sigma^2}$$

The natural logarithm is a **monotonic function** - in the log's case - it is always increasing - as can be seen in the graph below. There can only be a one-to-one mapping of its values. This means that optimizing the logarithm of the likelihood is no different from maximizing the likelihood itself.



One thing to note when working with logs of likelihoods, is that they are always negative. This is because probabilities assume values between 0 and 1, and the log of any value between 0 and 1 is negative. This can be seen in the graph above. For this reason, when working with log-likelihoods, optimization entails *minimizing* the *negative* log-likelihood; whereas in the past, we were trying to maximize the likelihood.

Lastly, as was done before, the constants in front of the equation can be removed without consequence. As well, for the purpose of this example, we will assume that the same sensor was used in obtaining both measurements - and will thus ignore the variance in the equation.

$$(z_1 - 1.8)^2 + (z_1 - 2.2)^2$$

Optimization

At this point, the equation has been reduced greatly. To get it to its simplest form, all that is left is to multiply out all of the terms.

$$2z_1^2 - 8z_1 + 8.08$$

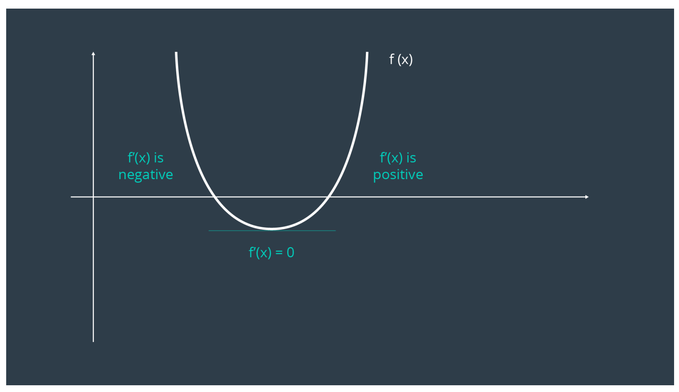
To find the minimum of this equation, you can take the first derivative of the equation and set it to equal 0.

$$4z_1 - 8 = 0$$

$$4z_1 = 8$$

$$z_1 = 2$$

By doing this, you are finding the location on the curve where the slope (or *gradient*, in multi-dimensional equations) is equal to zero - the trough! This property can be visualized easily by looking at a graph of the error function.



In more complex examples, the curve may be multimodal, or exist over a greater number of dimensions. If the curve is multimodal, it may be unclear whether the locations discovered by the first derivative are in fact troughs, or peaks. In such a case, the second derivative of the function can be taken - which should clarify whether the local feature is a local minimum or maximum.

Overview

The procedure that you executed here is the *analytical* solution to an MLE problem. The steps included,

- Removing inconsequential constants,
- Converting the equation from one of *likelihood estimation* to one of *negative log-likelihood estimation*, and
- Calculating the first derivative of the function and setting it equal to zero to find the extrema.

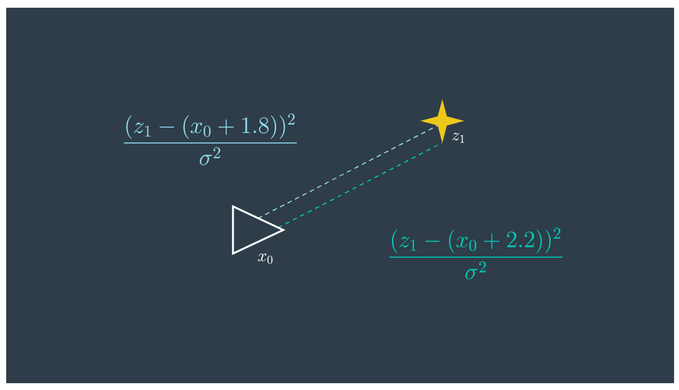
In GraphSLAM, the first two steps can be applied to *every* constraint. Thus, any measurement or motion constraint can simply be labelled with its negative log-likelihood error. For a measurement constraint, this would resemble the following,

$$\frac{(z_t - (x_t + m_t))^2}{\sigma^2}$$

And for a motion constraint, the following,

$$\frac{(x_t - (x_{t-1} + u_t))^2}{\sigma^2}$$

Thus, from now on, constraints will be labelled with their negative log-likelihood error,



with the estimation function trying to minimize the sum of all constraints,

$$J_{GraphSLAM} = \sum_t \frac{(x_t - (x_{t-1} + u_t))^2}{\sigma^2} + \sum_t \frac{(z_t - (x_t + m_t))^2}{\sigma^2}$$

In the next section, you will work through a more complicated estimation example to better understand maximum likelihood estimation, since it really is the basis of GraphSLAM.

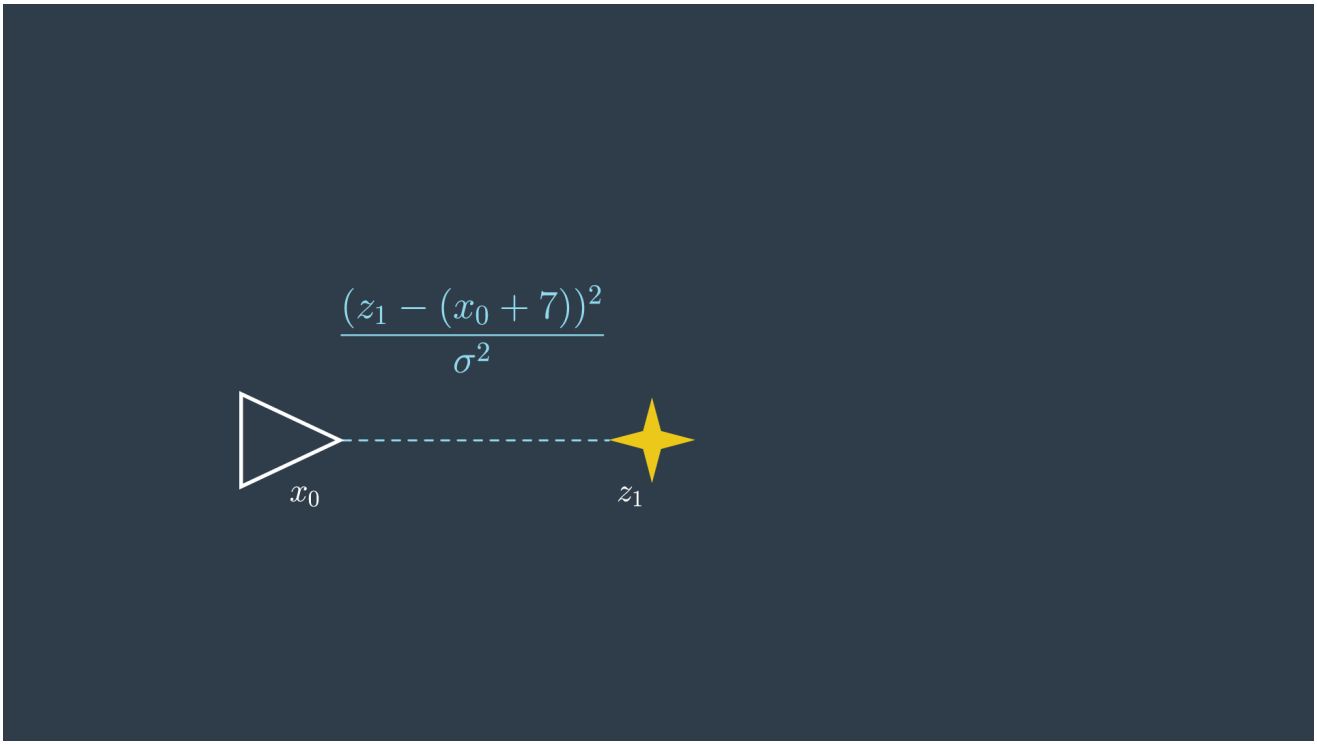
MLE Example

In the previous example you looked at a robot taking repeated measurements of the same feature in the environment. This example demonstrated the fundamentals of maximum likelihood estimation, but was very limited since it was only estimating one parameter - z_1.

In this example, you will have the opportunity to get hands-on with a more complicated 1-dimensional estimation problem.

Motion and Measurement Example

The robot starts at an arbitrary location that will be labeled 0, and then proceeds to measure a feature in front of it - the sensor reads that the feature is 7 meters way. The resultant graph is shown in the image below.



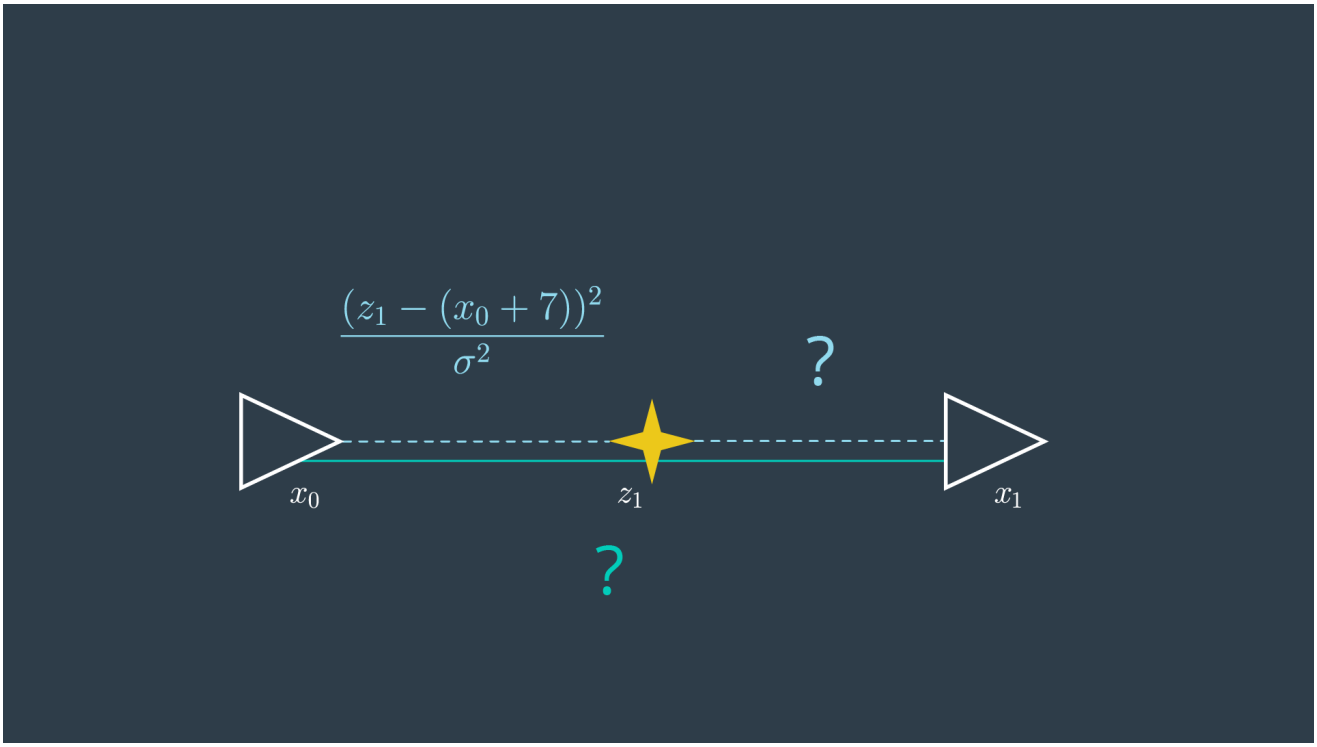
After taking its first measurement, the following Gaussian distribution describes the robot's most likely location. The distribution is highest when the two poses are 3 metres apart.

robot's most likely location. The distribution is highest when the two poses are 3 metres apart.

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z_1-(x_0+7))^2}{2\sigma^2}}$$

Recall that since we constrained the robot's initial location to 0, x_0 can actually be removed from the equation.

Next, the robot moves forward by what it records to be 10 meters, and takes another measurement of the same feature. This time, the feature is read to be 4 meters behind the robot. The resultant graph looks like so,



Now it's up to you to determine what the two new constraints look like!

Constraints Quizzes

QUESTION 1 OF 4

Which of the following is the correct constraint for the robot's motion from

MLE Example

QUESTION 1 OF 4

Which of the following is the correct constraint for the robot's motion from x_0 to x_1?

- [x] $\frac{(x_1-(x_0+10))^2}{\sigma^2}$
- [ ] $\frac{(x_1-(x_0-10))^2}{\sigma^2}$
- [ ] $\frac{(x_0-(x_1+3))^2}{\sigma^2}$
- [ ] $\frac{(x_1-(x_0-3))^2}{\sigma^2}$

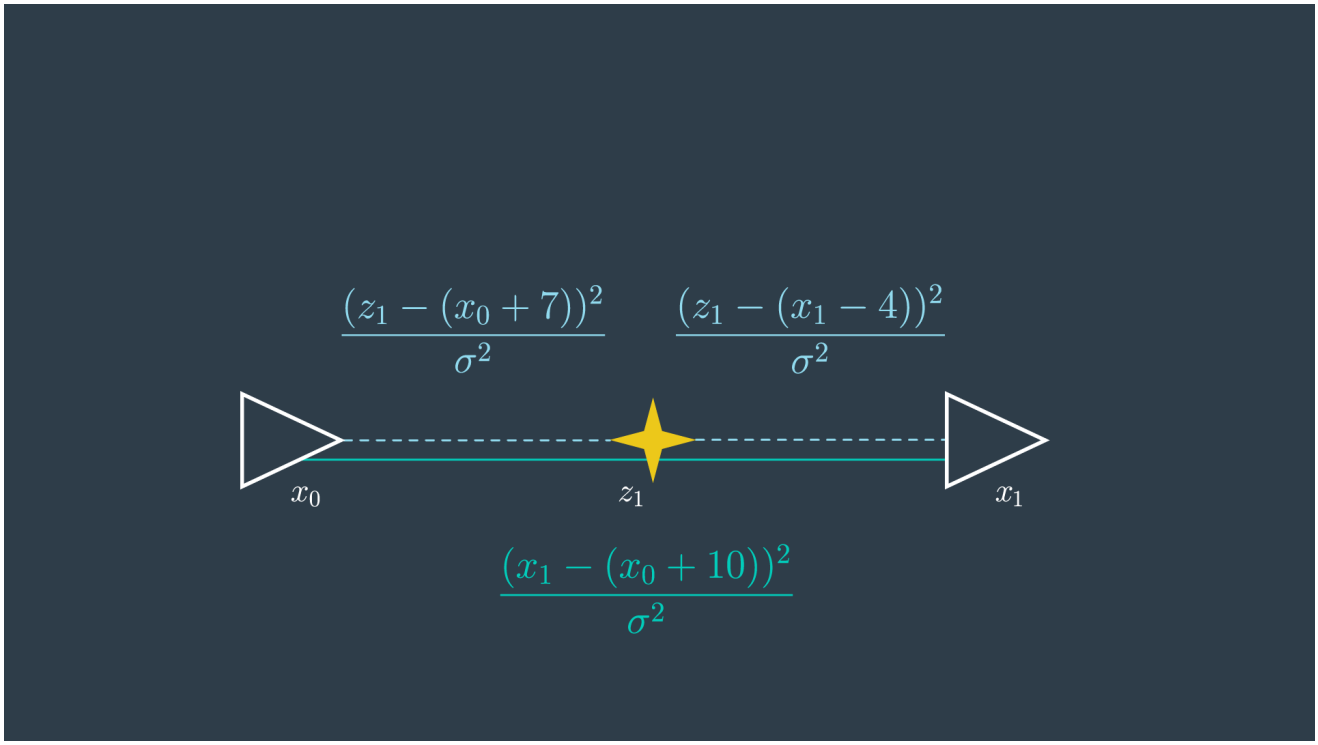
SUBMIT

QUESTION 2 OF 4

Which of the following is the correct constraint for the robot's measurement from x_1 to z_1?

- [ ] $\frac{(x_1-(z_1-3))^2}{\sigma^2}$
- [ ] $\frac{(z_1-(x_1+4))^2}{\sigma^2}$
- [ ] $\frac{(z_1-(x_1+3))^2}{\sigma^2}$
- [x] $\frac{(z_1-(x_1-4))^2}{\sigma^2}$

The completed graph, with all of its labelled constraints can be seen below.



Now, the task at hand is to minimize the sum of all constraints:

$$J_{GraphSLAM} = \frac{(z_1 - 7)^2}{\sigma^2} + \frac{(x_1 - (x_0 + 10))^2}{\sigma^2} + \frac{(z_1 - (x_1 - 4))^2}{\sigma^2}$$

To do this, you will need to take the first derivative of the function and set it to equal zero. Seems easy, but wait - there are two variables! You'll have to take the derivative with respect to each, and then solve the system of equations to calculate the values of the variables.

For this calculation, assume that the measurements and motion have equal variance.

See if you can work through this yourself to find the values of the variables, but if you're finding this task challenging and would like a hint, skip ahead to the solution to the quiz where I will step you through the process.

SUBMIT

If you've gotten this far, you've figured out that in the above example you needed to take the derivative of the error equation with respect to two different variables - z_1 and x_1 - and then perform variable elimination to calculate the most likely values for z_1 and x_1. This process will only get more complicated and tedious as the graph grows.

Optimization with Non-Trivial Variances

To make matters a little bit more complicated, let's actually take into consideration the variances of each measurement and motion. Turns out that our robot has the fanciest wheels on the market - they're solid rubber (they won't deflate at different rates) - with the most expensive encoders. But, it looks like the funds ran dry after the purchase of the wheels - the sensor is of very poor quality.

Redo your math with the following new information,

- Motion variance: 0.02,
- Measurement variance: 0.1.

Optimization Quiz 2

QUESTION 4 OF 4

What are the estimated locations of z_1 and x_1 based on your calculations when taking into account the variances?

- $z_1 = 6.61, x_1 = 10.12$
- $z_1 = 6.66, x_1 = 10.33$
- $z_1 = 6.54, x_1 = 10.09$

QUESTION 3 OF 4

What are the estimated locations of z_1 and x_1 based on your calculations?

- $z_1 = 6.5, x_1 = 10.5$
- $z_1 = 7, x_1 = 11$
- $z_1 = 6.5, x_1 = 10.75$
- $z_1 = 6.67, x_1 = 10.33$

SUBMIT

If you've gotten this far, you've figured out that in the above example you needed to take the derivative of the error equation with respect to two different variables - z_1 and x_1 - and then perform variable elimination to calculate the most likely values for z_1 and x_1. This process will only get more complicated and tedious as the graph grows.

Optimization with Non-Trivial Variances

To make matters a little bit more complicated, let's actually take into consideration the variances of each measurement and motion. Turns out that our robot has the fanciest wheels on the market - they're solid rubber (they won't deflate at different rates) - with the most expensive encoders. But, it looks like the funds ran dry after the purchase of the wheels - the sensor is of very poor quality.

Redo your math with the following new information,

- Motion variance: 0.02,
- Measurement variance: 0.1.

Optimization Quiz 2

QUESTION 4 OF 4

What are the estimated locations of z_1 and x_1 based on your calculations when taking into account the variances?

- $z_1 = 6.61, x_1 = 10.12$
- $z_1 = 6.66, x_1 = 10.33$
- $z_1 = 6.54, x_1 = 10.09$
- $z_1 = 6.6, x_1 = 10.25$

SUBMIT

That seemed to be a fair bit more work than the first example! At this point, we just have three constraints - imagine how difficult this process would be if we had collected measurement and motion data over a period of half-an hour, as may happen when mapping a real-life environment. The calculations would be tedious - even for a computer!

Solving the system analytically has the advantage of finding *the* correct answer. However, doing so can be very computationally intensive - especially as you move into multi-dimensional problems with complex probability distributions. In this example, the steps were easy to perform, but it only takes a short stretch of the imagination to think of how complicated these steps can become in complex multidimensional problems.

Well, *what is the alternative?* you may ask. Finding the maximum value can be done in two ways - *analytically* and *numerically*. Solving the problem numerically allows for a solution to be found rather quickly, however its accuracy may be sub-optimal. Next, you will look at how to solve complicated MLE problems numerically.

NEXT

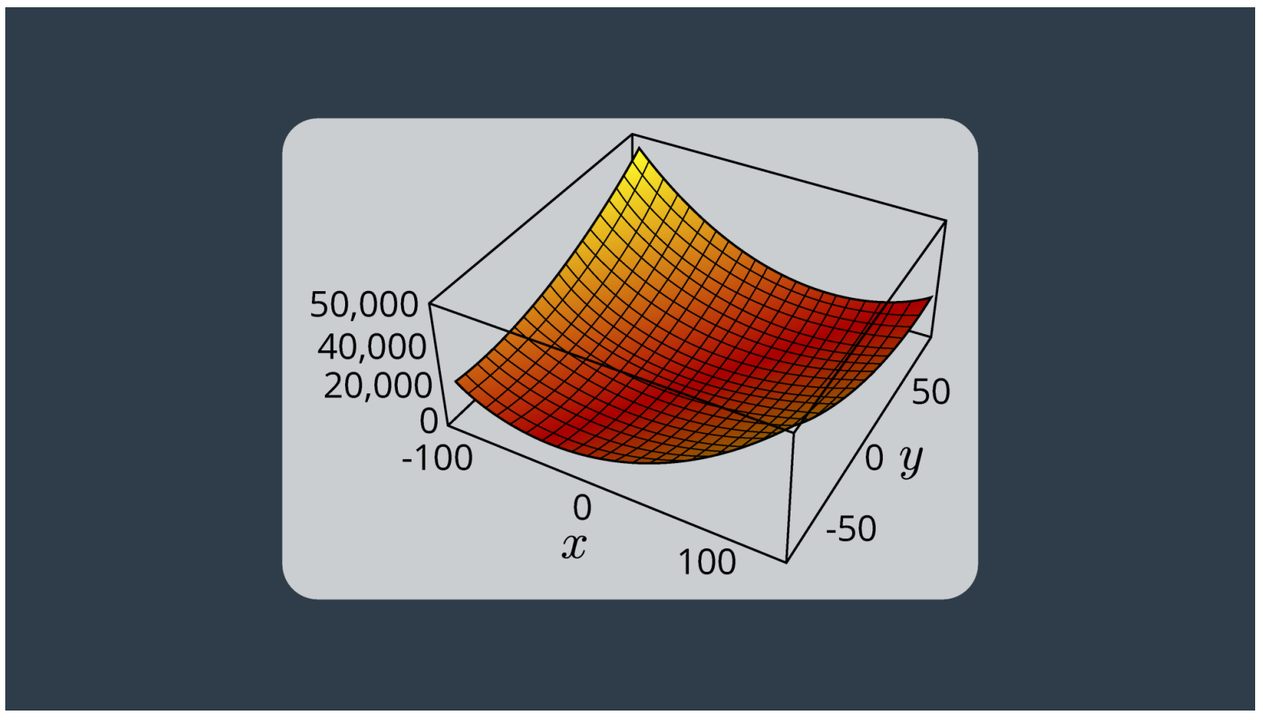
Numerical Solution to MLE

The method that you applied in the previous two examples was very effective at finding a solution quickly - but that is not always the case. In more complicated problems, finding the analytical solution may involve lengthy computations.

Luckily there is an alternative - numerical solutions to maximum likelihood problems can be found in a fraction of the time. We will explore what a numerical solution to the previous example would look like.

Numerical solution

The graph of the error function from the previous example is seen below. In this example, it is very easy to see where the global minimum is located. However, in more complicated examples with multiple dimensions this is not as trivial.



This MLE can be solved numerically by applying an optimization algorithm. The goal of an optimization algorithm is to *speedily* find the optimal solution - in this case, the local minimum. There are several different algorithms that can tackle this problem; in SLAM, the **gradient descent**, **Levenberg-Marquardt**, and **conjugate gradient** algorithms are quite common. A brief summary of gradient descent.

Quick Refresher on Gradient Descent

Recall that the gradient of a function is a vector that points in the direction of the greatest rate of change; or in the case of an extrema, is equal to zero.

In gradient descent - you make an initial guess, and then adjust it incrementally in the direction *opposite* the gradient. Eventually, you should reach a minimum of the function.

This algorithm does have a shortcoming - in complex distributions, the initial guess can change the end result significantly. Depending on the initial guess, the algorithm converges on two different local minima. The algorithm has no way to determine where the global minimum is - it very naively moves down the steepest slope, and when it reaches a local minima, it considers its task complete. One solution to this problem is to use stochastic gradient descent (SGD), an iterative method of gradient descent using subsamples of data.