ARO lec. 2, localization, factorgraphs

Lukáš Nejezchleb

May 20, 2024

1 Outline, Goals

When we have a robot in the world, we cannot directly accurately measure all the states required for decisions on which actions to perform. This means that if we want to decide correctly, we have to estimate our state as best as we can based on measurements we have. Our only option is usually to guess the current state as some state in which there is a maximum probability of the measurements we get.

2 Notation

In order to be consistent with lecture slides, the following notation will be used

- x_i $\mathbf{x}_i$ are robot i-th states (could be viewed as states at time i), first one is just one dimensional state (1D robot on line measuring its distance from origin), second one is vector (which could consist of 3D position, orientation, temperature, battery voltage, ...)
- x_i^* $\mathbf{x}_i^*$ is the optimal state. In most cases we do not know the ground truth (actual robot state). Because of that we suppose that the optimal state (the state, in which there is the highest probability of the sensors giving us values we obtained) is somewhat close to the ground truth.
- z_i $\mathbf{z}_i$ are measuremtents taken with a goal to estimate the state, in which a robot is (or was, or will be). This could be a gps measurement, lidar scan, odometry measurement, temperature or voltage measurement.
- σ Σ are variance (or multidimensional covariance matrix). This shows us, how big noise and drifts of our measurements are. The bigger the value is, the less we can trust the measured values. When we use a covariance matrix in the multidimensional case, we can put a different weight on each vector element. This allows us to edit our "trust" element-wise (for example put lower values of variance on elements corresponding to x and y accelerometer axis and slightly higher variance on z axis (which tends to be noisier because of manufacturing technology).

3 Short intro to joint probability (only basics, if you know it, feel free to skip it)

3.1 Independent events

When two events are independent, the knowledge of the results of one event cannot help us in any way in guessing the outcome of the other one.

If we get two (or more independent events) and want to obtain the probability that both events occured at the same time, we just multiply the individual probabilities.

3.1.1 Independent event vs not independent event, what is Conditional probability

We can use the struggle of some teachers with the controls of blackboard raising mechanism to our advantage and demonstrate the possible differences in independent and not independent events (this example might have or might have not happened in real life). Let's suppose we want to raise the blackboard. But on the side there are 3 switches each controlling different blackboard. Imagine our goal is to press the correct switch the most times in 3 tries.

If we assume independent events, we try one of those three switches. There is 1/3 chance of choosing the correct one. We immediately forget if the switch was correct (most probably out of nervousness caused by the whole room looking at us). On our second try, we know nothing and we have 1/3 chance of choosing the correct switch again. This repeats for the third time. Our probability of choosing the correct switch in the third try is still 1/3. If we try to calculate the probability of us not getting the correct switch at all, we just raise 2/3 (probability of missing the switch in one try) to the third power (because of three tries) and we get 8/27≈30 %.

Now we can look what would happen if we get rid of our nervousness and we can suddenly remember. We try some switch. If we find out the switch was a correct one, there is no need to try other ones. We just flick the same switch two more times. If we chose the wrong one, we are just down to two switches increasing our chance to 50 %. So how to write this down?

The probability of selecting the correct switch in the first try is $P(\text{first correct}) = 1/3$. Now we use something called conditional probability. This tells us, what is the probability of some event given that we know the outcomes of some other events. We can write the probabilities for the example above. $P(\text{second correct}|\text{first correct}) = 1$ (if we know the first one was correct, we just flick the same one). $P(\text{second correct}|\text{first wrong}) = 1/2$. Now we might ask, what is the probability of selecting the correct switch on the second try. If we managed to select the correct one on the first try, we are guaranteed to choose correct one in the second as well. This happens in 1/3 of the cases. In 2/3 of all possible cases we don't manage to select the correct one on the first time. In half of those cases we manage to choose the correct switch in the second try. In mathematics, it could be written as

$$\begin{aligned} P(\text{second correct}) &= P(\text{first correct}) \cdot P(\text{second correct}|\text{first correct}) + P(\text{first wrong}) \cdot P(\text{second correct}|\text{first wrong}) \\ &= \frac{1}{3} \cdot 1 + \frac{2}{3} \cdot \frac{1}{2} = \frac{2}{3} \end{aligned}$$

We can see that we can manage to flick the correct switch in the second try with 2/3 chance (and therefore the probability of us not getting the correct switch in the second try is 1/3). Now if we managed to flick the correct switch, we just flick the same one. If we did not, there is only one left, so in this case

$$\begin{aligned} P(\text{third correct}) &= P(\text{second correct}) \cdot P(\text{third correct}|\text{second correct}) + P(\text{second wrong}) \cdot P(\text{third correct}|\text{second wrong}) \\ &= \frac{2}{3} \cdot 1 + \frac{1}{3} \cdot 1 = 1 \end{aligned}$$

We can see that the probability of us not managing to find the correct switch is zero. You can compare this with the 30 % result when we assumed the events to be independent.

3.1.2 Independent events ex: balls from pouch

If we decide to draw a ball from a pouch containing a red ball and blue ball, the probability of us drawing a red ball is 50 %. If we put the drawn ball back and draw again, the probability is 50 % as well. This means that there are 4 possible situations that could have happened.

- We got blue ball in both draws.
- We got red ball in both draws.
- We got blue ball in the first draw and red ball .
- We drew red ball in the first try and blue ball in the second try.

We can see that out of these scenarios there is just one in which we draw two blue balls. The probability is therefore 1/4 = 25 %. We could have gotten the same result by multiplying the probability, we get blue ball in the first draw (50 %) and probability of getting the blue ball in the second draw (50 %).

4 MAP (Maximum aposteriory estimate)

Now that we know something about probability, let's put it to a good use. When we are given results of multiple measurements (eg. on our position: GPS, camera, lidar, odometry and previous position), our goal is to estimate our current state (position). In this part, we are considering the robot as one point, where we do not have to consider rotation. Position is represented as only x,y(,z)

Every sensor has it's variance (how much the results are spread out - noisy). If we get a measurement with really low variance, we can trust it much more compared to a measurement with really high variance.

Let's start with some assumptions (some of them are frequently done to make the task easier)

- **The measurements are independent** (If there was a bug in lidar which caused the readout value to be completely wrong, there is no reason to beleve the GPS reading is wrong as well)
- **Uniform prior** (There is no reason to believe robot has bigger chance of being at some particular place, it could be anywhere. We estimate the current state just based on our observation.)
- **Gaussian noise** (The probability of sensor measurement resulting in certain value is given as Gaussian distribution with mean of the actual state and variance from the specific sensor. The most probable measurements are the ones which are closest to the actual state)

Given these assumptions we can write the probability of some state being the real one given the measurements we got.

$$P(\mathbf{x}|\mathbf{z}) = \frac{P(\mathbf{x}\&\mathbf{z})}{P(\mathbf{x})} = \frac{P(\mathbf{z}|\mathbf{x}) \cdot P(\mathbf{x})}{P(\mathbf{z})}$$

We want to find some state which maximizes this probability (There is the biggest chance of us guessing correctly).

We can make our task easier using our assumption about uniform prior. Because we don't have any assumptions and we just want to maximize the probability, we can write a formula for state with maximum probability (using $P(\mathbf{x}) = const.$ and $P(\mathbf{z}) = const.$)

$$\mathbf{x}* = \arg\max_{\mathbf{x}} \left(P(\mathbf{x}|\mathbf{z})\right) = \arg\max_{\mathbf{x}} \left(P(\mathbf{z}|\mathbf{x})\right)$$

$$\mathbf{x}* = \arg\max_{\mathbf{x}} \left(P(\mathbf{z}|\mathbf{x})\right) = \arg\max_{x} \left[\Pi_{i=0}^{N} P\left(\mathbf{z}_i|\mathbf{x}\right)\right]$$

The last assumption is the Gaussian noise. This, again, does not need to be correct, but is quite useful and allows us to convert this problem into least squares problem, which has closed form solution. Now we assume that the probability is (for simplicity in one variable)

$$P\left(z|x\right) = \mathcal{N}\left(z|x, \sigma^2\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left[-\frac{1}{2}\left(\frac{z-x}{\sigma}\right)^2\right]}$$

With that we can rewrite

$$x* = \arg\max_{x} \left[\Pi_{i=0}^{N} P\left(z_i|x\right)\right] = \arg\max_{x} \left[\Pi_{i=0}^{N} \mathcal{N}\left(z_i|x, \sigma_i^2\right)\right] = \arg\max_{x} \ \Pi_{i=0}^{N} \left[\frac{1}{\sigma_i\sqrt{2\pi}} e^{\left[-\frac{1}{2}\left(\frac{z_i-x}{\sigma_i}\right)^2\right]}\right]$$

Finding maximum of this function just as is would be quite inconvenient. Firstly we get rid of constants. Than we can simplify all of this by taking a negative logarithm of this function (which converts the Π multiplication into summation). Because logarithm is monotonically increasing (higher argument always leads to higher result), we can say that if some state is the argmax of logarithm, it is the argmax of the original function as well. Then we add the minus sign just to make our lives easier and get rid of the -1/2. This means we have reversed the order and we have to find argmin instead of argmax.

$$x* = \arg\max_{x} \ \Pi_{i=0}^{N} \left[\frac{1}{\sigma_i\sqrt{2\pi}} e^{\left[-\frac{1}{2}\left(\frac{z_i-x}{\sigma_i}\right)^2\right]}\right] = \arg\max_{x} \ \Pi_{i=0}^{N} \left[e^{\left[-\frac{1}{2}\left(\frac{z_i-x}{\sigma_i}\right)^2\right]}\right] =$$

$$= \arg\max_{x} \ \Sigma_{i=0}^{N} \left[-\frac{1}{2}\left(\frac{z_i-x}{\sigma_i}\right)^2\right] = \arg\min_{x} \ \Sigma_{i=0}^{N} \left(\frac{z_i-x}{\sigma_i}\right)^2$$

When we look at the result, it is quite familiar. Our only task is just to minimize sum of squares: This means we have converted this huge task into least square problem.

On the following figure, we can see the transformation to negative logarithm space. We can see, that the extrema position stayed at the same place.

We can also see one advantage in the conversion to negative logarithm space. In case the measurement was too far from the actual state, the probability calculations could quickly turn problematic (as multiplication of near zero numbers could quickly get to zero). When we move into negative logarithm, the quadratic function does not rise that quickly and summation won't get to extreme values that quickly.

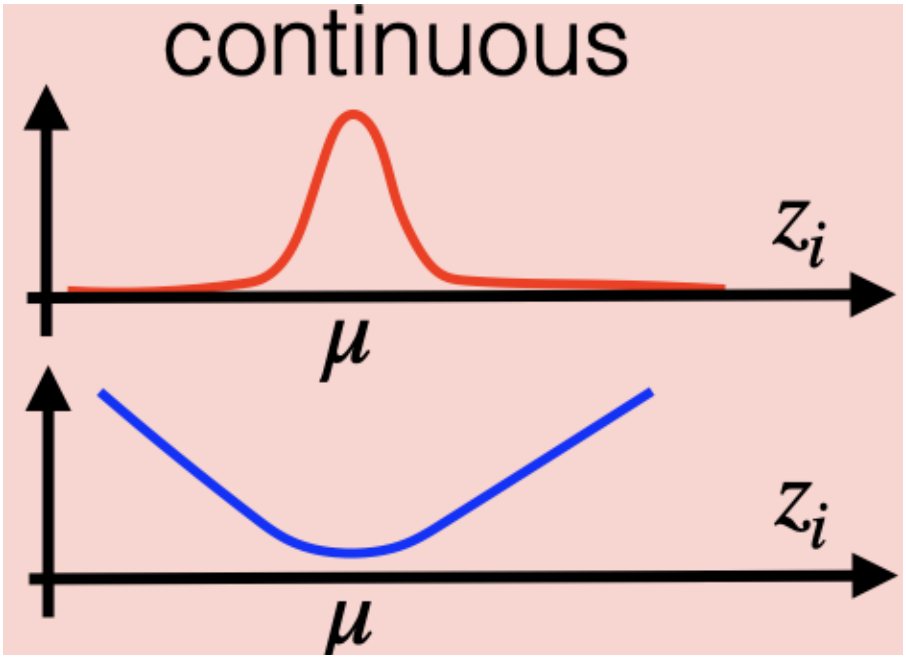

Figure 1: Probability transformed to neglog space

4.1 Multidimensional probability

When we assume multidimensional gaussian probability instead of one dimensional one (as in previous part), we can get to a quite similar result.

$$P\left(\mathbf{z}|\mathbf{x}\right) = \mathcal{N}\left(\mathbf{z}|\mathbf{x},\Sigma\right) = \frac{1}{\sqrt{(2\pi)^{\dim(\mathbf{x})}\det(\Sigma)}} e^{\left[-\frac{1}{2}(\mathbf{z}-\mathbf{x})^T\Sigma^{-1}(\mathbf{z}-\mathbf{x})\right]}$$

With that we can rewrite

$$\mathbf{x}* = \arg\max_{\mathbf{x}} \left[\Pi_{i=0}^{N} P\left(\mathbf{z}_i|x\right)\right] = \arg\max_{\mathbf{x}} \left[\Pi_{i=0}^{N} \mathcal{N}\left(\mathbf{z}_i|\mathbf{x},\Sigma_i\right)\right] = \arg\max_{\mathbf{x}} \Pi_{i=0}^{N} \left[\frac{1}{\sqrt{(2\pi)^{\dim(\mathbf{x})}\det(\Sigma_i)}} e^{\left[-\frac{1}{2}(\mathbf{z}_i-\mathbf{x})^T\Sigma_i^{-1}(\mathbf{z}_i-\mathbf{x})\right]}\right]$$

As previously, we are just searching for maximum of that function. We can therefor get rid of constants and take a negative logarithm of the function.

$$\begin{aligned}\mathbf{x}* &= \arg\max_{\mathbf{x}} \Pi_{i=0}^{N} \left[e^{\left[-\frac{1}{2}(\mathbf{z}_i-\mathbf{x})^T\Sigma_i^{-1}(\mathbf{z}_i-\mathbf{x})\right]}\right] = \arg\max_{\mathbf{x}} \Sigma_{i=0}^{N} \left[-\frac{1}{2}\left(\mathbf{z}_i-\mathbf{x}\right)^T\Sigma_i^{-1}\left(\mathbf{z}_i-\mathbf{x}\right)\right] \\ &= \arg\min_{\mathbf{x}} \Sigma_{i=0}^{N} \left[\left(\mathbf{z}_i-\mathbf{x}\right)^T\Sigma_i^{-1}\left(\mathbf{z}_i-\mathbf{x}\right)\right]\end{aligned}$$

Again, we obtained least squares problem. The Σ^{-1} could be considered as a weight matrix. In dimensions, where Σ has low values (low noise), Σ^{-1} has high values increasing the weight we put on errors in that particular dimension.

4.2 Intuitive view

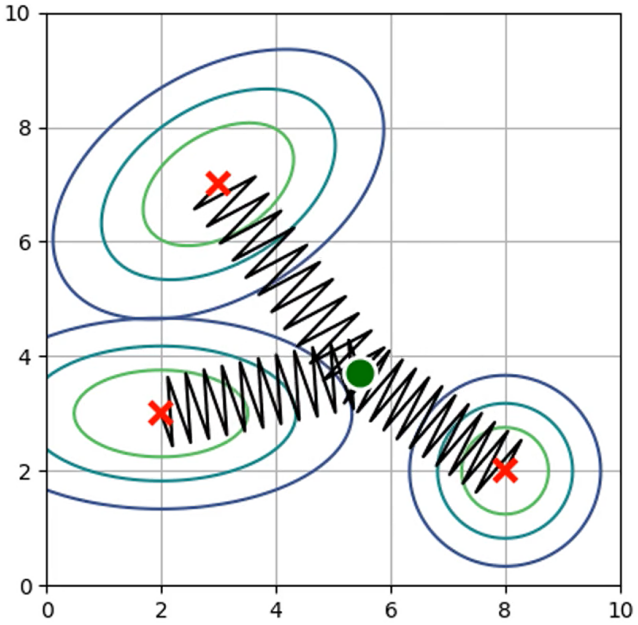

Figure 2: Springs from GPS measurements pulling the state towards themselves

Now that we crunched through all this boring math, we might ask ourselves if there is some intuitive view. If we look at the results, it might be slightly familiar. If we look at equations for potential energy of spring ($E = cx^2$), where c is stiffness constant and x is spring displacement. All springs are "the happiest" when they are relaxed and their displacement is zero. If we connect multiple springs together, they contract and expand in order for the system to have as low energy as possible. The stiffer the spring, the less it will want to expand, because the higher thickness means higher weight when adding all energies together.

Last thing we should find out is what is the spring stiffness in our problem. If we compare the equations, we can see that c corresponds to the term $\frac{1}{\sigma^2}$. σ or Σ (variance for one variable, covariance matrix for multiple) tells us, how much we can trust the measurements. The higher values it contains, the more spread out the measurements are (more noisy). This makes sense if we think about it. If the measurement variance is huge, we cannot be sure about anything. The assigned spring to this measurement is really weak, so that if the measurement is wrong, it does not influence our state choice that much.

4.3 What if measurements do not have Gaussian distribution

There are two possible options. We usually say that the actual distribution is somewhat close to the gaussian distribution and behave as everything had a gaussian noise. Other option is to calculate using that known distribution. This could be quite problematic, because it could convert our task into much harder one without closed form solution.

5 What could be considered as a measurement

Most measurements could be divided into two groups, unary and pairwise. Differences are based on the number of states it links and whether the links are relative to some state in different time or absolute. In reality there are more than just these two (ternary terms linking three states, loop closers), but the following two are quite useful and relatively simple to understand.

5.1 Unary

These give us some intel about robots current state (e.g. GPS). It links the current state to some coordinate frame.

5.2 Pairwise

These measurements give us some intel about changes robot state, they usually link previous state to the current one. Typical examples are

- odometry (measuring the changes in robot position, eg: robot moved 3 cm in x direction and 2cm in y direction)
- motion model (because we know the control inputs, we can calculate (based on physics) the expected movement.

6 Factorgraph

The localization is usually done by maximalizing the product of all probabilities (unary, pairwise, others), or, as we have shown, the weighted distances from obtained measurements (gaussian probabilities converted to negative logarithm space).

To keep track of all the probability links, factorgraph can be used. It has two types of nodes:

- **variables** which correspond to robot state.
- **factors** which show us the probability links between the states.

and edges which link the probabilities (factors) with the states (variables). We can imagine factors as springs pulling the variables towards themselves (unary factors) or pulling and pushing the variables to have some particular separation (pairwise, ternary, ...). Example of factorgraph can be seen in the following figure

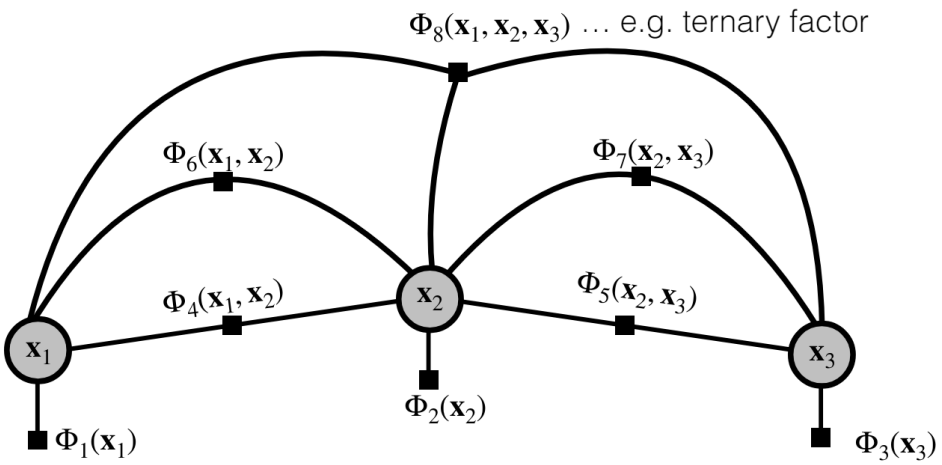

Figure 3: factorgraph

We can see that in each state, we performed a measurement Φ_1,Φ_2,Φ_3 to find out robot state (e.g. GPS measurement). We have could have looked at odometry (measuring the expected change in robots position) (Φ_4,Φ_5), factors Φ_6,Φ_7 could correspond to motion model (us knowing the robot inputs and physics behind it). Φ_8 could depict us detecting a relative marker in all three states. This would link the positions in all three states.

Our task is to maximalize probability of all measurements combined. Depending on the factors, there could be a closed form solution (if all factors are linear). If not, the solution could be obtained by nonlinear optimalization methtods (gradiend descend, ...)