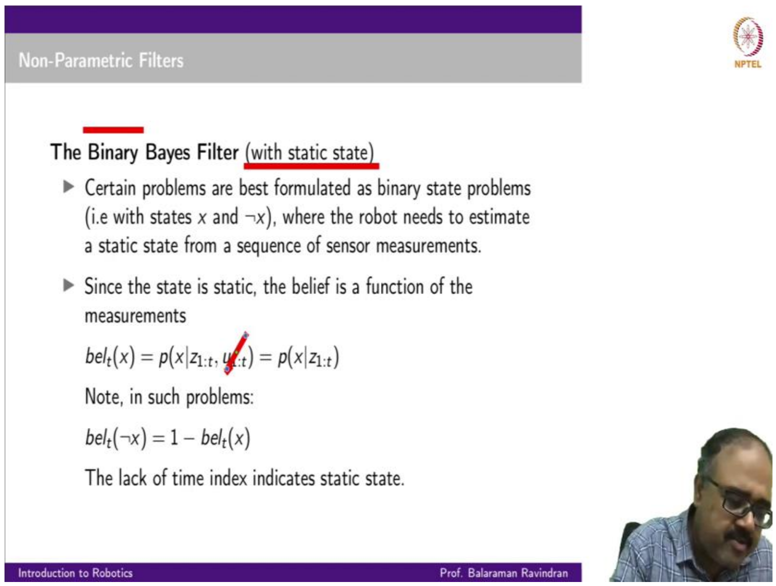
Introduction to Robotics
Professor Balaraman Ravindiran
Department of Computer Science
Indian Institute of Technology, Madras
Lecture - 36
Binary Bayes

(Refer Slide Time: 00:14)



Welcome back to the fourth lecture in Week 10, and we are going to continue looking at non-parametric filters. So if you remember, we started looking these filters as a way of doing recursive state estimation.

So what we are going to look at in this lecture is a very special case, where I really, really want to know what my state is, what my current state is. And I am trying to keep on making repeated measurements. And going to continue to refine my estimate of where I am. Such problem settings are called problems with static state.

So I am basically, I am going to assume that I am either going to, I have, I am not changing my state, my state is fixed and I am going to only repeatedly make measurements until I am sure about my state. And we are going to look at a very specific instance.

It is a non-parametric filter as you can see from the slide because it is going back to the original Bayes filter setup, not making any assumption about what the distribution is. I am just going to assume that it is a problem with static state. But I am also assuming it is a binary state problem.

So even though now I am presenting to you in the context of you know, recursive state estimation, we will see later that this binary Bayes filter with static state has other uses and it is more important in those context that we will see later on but I am just introducing it here for you so that it stays with all the other filters that we have studied.

So the goal here is to look at problems that are formulated as binary state problems. Basically, I have a state variable x that can either be true or false. So my states are either x or not x; x can be the true or it can be false.

And so, the robot just needs to estimate what is the value of x. Is x true or whether x is false. I mean, it could be something as simple as, okay, is there a door here or no door here, is our door open or not open? Or as we will see later, is there an obstacle in the cell that I am looking at, or is there no obstacle in the cell, is it clear? So what, the space in front of me is it clear or not clear.

It is basically, that is it. So it is a binary state estimator. Door open, door closed. You know, do I have fuel or no fuel. So is simply a single indicator variable which is either 0 or 1 and I am assuming that I am going to make multiple measurements until I am satisfied with my estimate of that.

Now, since the state is a static state, so the actions do not really matter. So actions do not play a role here. The actions do not play a role because the state is static. And so, static meaning actions do not affect the state. And you can see that by the fact that we do not put any time index on the actions but the observations still have a time index.

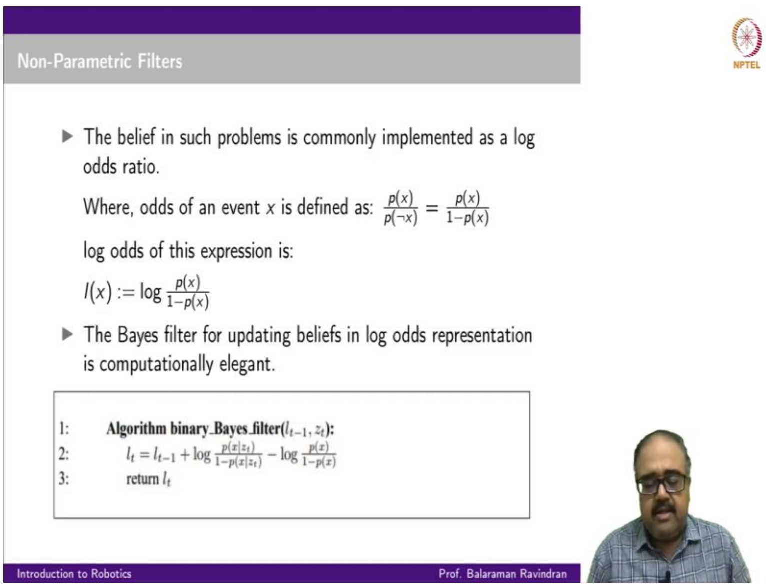
So my belief state at time t, so notice that it does not say bel xt anymore, it says bel t of x that is because my belief still keeps changing. So notice that my belief state is going to be only two numbers, whether x equal to 0 or whether x equal to 1.

So the probability of x equal to 0 is one number, probability of x equal to 1 is another number. And so, and that estimate is going to change with time so, therefore, I have a time index on the belief but x itself is static. Therefore, I removed the time index from x.

And so the belief, instead of being the probability of x, given all your observations and all your actions from the past is basically the probability of x given your observations alone. Given all the observations you have made up till from time 1 to time t what is the probability of x? In this case, what is the probability of x being true?

And so, in many, in these problems you should remember that, so 1 minus bel of x gives me bel of x bar because if x is not true, x has to be false. So 1 minus bel of x gives me bel of x bar and this is true for every time t. So that is basically the problem setup that I am currently looking at.

(Refer Slide Time: 04:40)



So one of the things that we do as you will see in a bit is the belief, instead of representing it as the probability distribution directly, I represent the belief as something known as the log odds ratio. So in, you know, in probability theory the odds of an event x is basically defined as the ratio of the probability that x happens divided by the probability that x does not happen.

So this is something that is, that is quite familiar with some people. Maybe if you look at the outcome of say, sporting events and things like that people say, what are the odds of that happening? So when the odds here actually refers to the ratio of p of x divided by p of not x. And in our case, we can say that it is p of x divided by 1 minus p of x. So this is the odds of x happening.

And log-odds obviously is going to be log of p of x divided by 1 minus p of x. Is that clear? So the log odds is essentially log of p of x divided by 1 minus p of x and we are going to denote that by the symbol l. So l of x equal to log of p of x divided by 1 minus p of x.

Now, I am just going to give you the Bayes filter algorithm for the log odds representation, just to tell, give you the motivation as to why we are looking at the log odds representation, and then we will actually go back and derive this. So remember what does my, the Bayes filter algorithm do? It takes my current belief which is lt minus 1 here, which is log odds at time t minus 1, and my current action and my current observation. The action at time t and observation at time t.

But I do not need my action at time t because I have static state, therefore, I ignore the action. I only take my observation at time t. And so, lt, obviously is going to be my belief at time t is basically this is additive expression. So I am going to look at lt minus 1, which is my belief at time t minus 1 times log of this expression, which is the probability of x given zt divided by 1 minus probability of x given zt minus log of p of x divided by 1 minus p of x.

So what is this p of x? So this p of x is essentially my prior probability of whether x is true or not. So p of x is probability that x is true when I have not seen any observation. So this is my initial belief as to whether x is true or not and I keep, keep basically adding or subtracting these odds, my initial odds on x every time I make the update. So we will see why that is the case as we go along the next few slides.

I am just going to read the expression again. So my belief at time t which is represented as log-odds, so lt is equal to the belief at time t minus 1, which is lt minus 1 plus log of the probability that x is true given zt has happened. So zt is observation at time t divided by 1 minus probability x is true given zt, which is basically the probability that x is not true given zt.

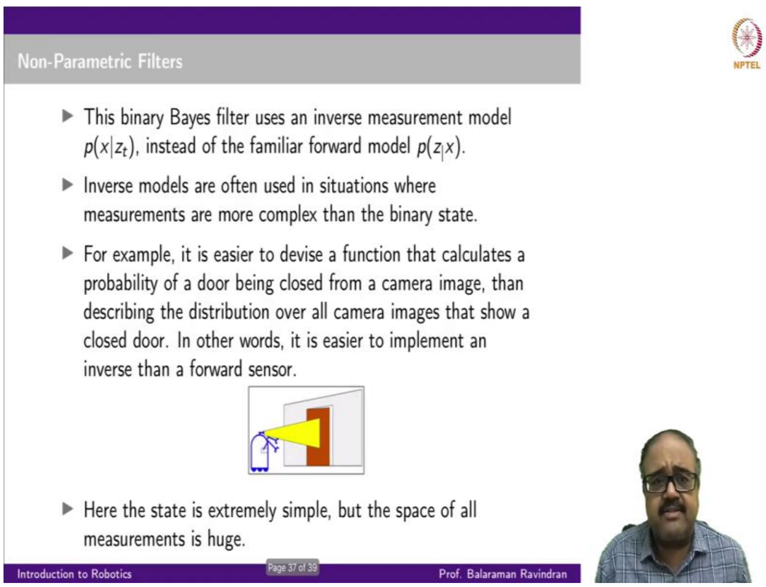
So this is basically the log odds of p of x given zt, minus log of p of x divided by 1 minus p of x, where p of x is the probability that x is true before I have seen any observations. You can think of this as the log odds, this whole expression can be thought of as l not, the belief at time 0. The log odds of the belief at time 0 is essentially what this expression is.

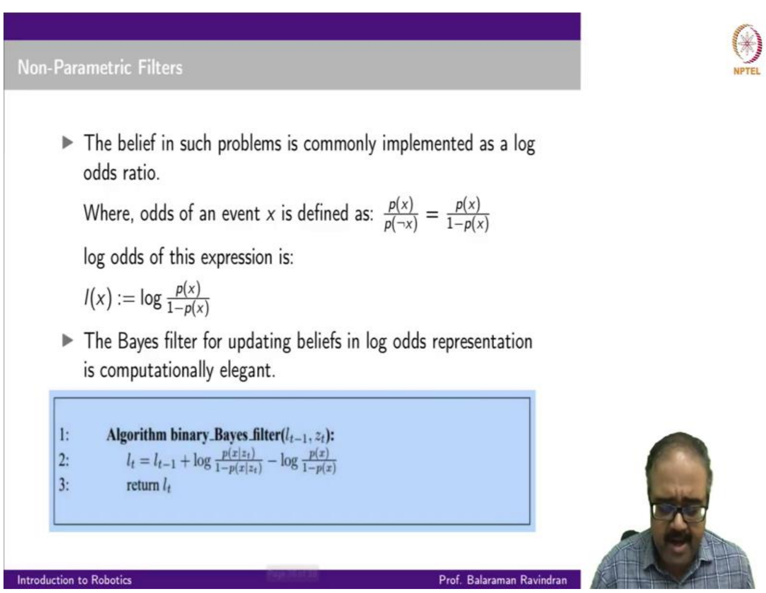
So you can think of this as lt equal to lt minus 1 plus log of this expression minus l not. And once I have done this computation, I just return lt. Note it is fairly straightforward. It is very simple additive expression and the nice thing about it is because I am working in this additive space and with these, the logs, I can handle more effectively numbers that are very, very, very small.

So if I am in a situation where the log odds of something being true or being false is very small, or if the probability of something being true or false is very small, using the log-odds expression, allows me to be more stable in my updates. Numerically more stable in my updates and allows me to handle this more elegantly. So that is the reason we go in for this log odds.

And later, when we look at where we use these, especially in places like map estimations, we will see that the probabilities do tend to be very small, and therefore, using this kind of a, the adaptation of this binary Bayes filter, it is very useful in such cases.

(Refer Slide Time: 09:38)





So one thing I want you to again notice is that in the updates here; I am sorry, so in the update here, we are using probability of x given zt. Normally, our measurement model is written as probability of zt given xt.

Normally, we will look at what is the probability of the measurement happening given x but in this update, you are using what is the probability of x given the measurement has happened. So and, this is called an inverse measurement model. And the reason we are using the inverse measurement model here is because our state is actually very simple.

In the normal case, our state would be very, it could potentially be a complex vector; very high dimensional state space but in this case, we are talking about binary state, so x is either true or false. So it is a fairly simple state but whereas, the observations could potentially be very complex.

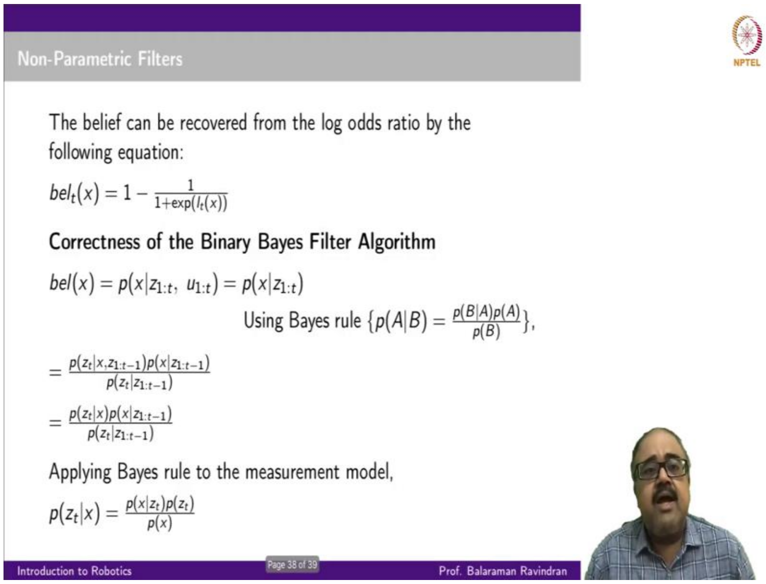
So let us go back to one of our old examples. The observation could be an image from a camera, and our state could be whether the door is open or closed. It is a binary state; open or closed. But the input, the z could be a full-blown image from a camera or like a small, small bit of a video from a camera, and therefore, the z could be very complex.

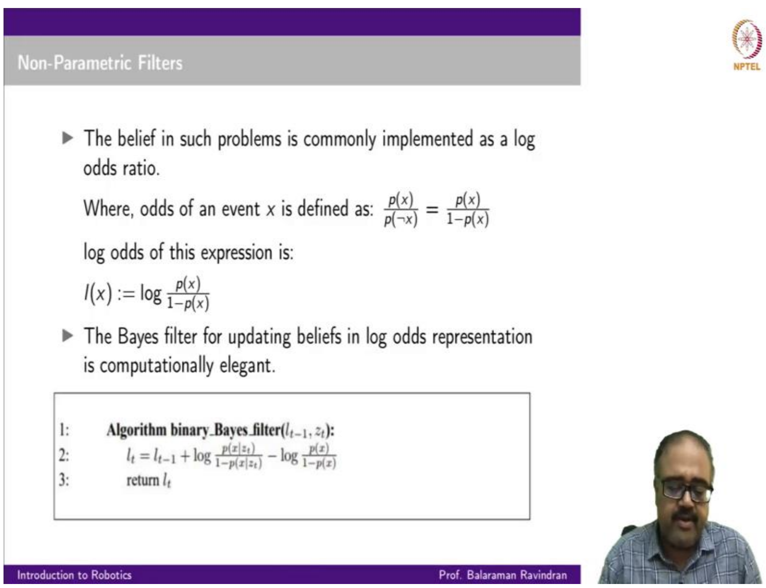
So if I am going to learn the forward model, if you are going to represent the forward model, then basically, we will have to represent the distribution over a very, very complex space, which is the space of the observation, which is space of all images that my camera could capture. That is a fairly complicated endeavor.

So what we do here is because the state is so simple, we try to see if we can work with this, this inverse measurement model. So we have to be, so one of the things that you will be finding out, as we keep going along is that we are learning a lot of tools. There is nothing like there is one single tool that is the best thing to use at every point and depending on the application, depending on the situation that you are actually using these tools, you will have to pick whatever is the best one for you

So in this case, because the state is simple, observations are complex, I would prefer to use a inverse model as opposed to the usual forward measurement model.

(Refer Slide Time: 12:06)





So now, so once I have the log odds ratio, so we will to come to that in a bit, the rest of the slide, just ignore that for the time being. Just look at this. Just edit everything I said after this slide came on.

So I kept saying that the log odds ratio is the belief, but if you want actual belief distribution, the belief distribution is the probability of x given z1 to t, so I can recover the belief of x at time t by just doing this. So 1 minus 1 by e power lt of x. Here, lt, if you remember is the log odds ratio at time t. So how did that come about?

I will just take you back to the original definition. So if you look at this definition, so l of x which is our log odds is equal to log of p of x by 1 minus p x. Now, if we take the log to that side so that becomes e power l of x. And then, I take the 1 minus p x to that side so that becomes e power l of x minus e power l of x times p x.

So I bring that back here. So I will get p of x into 1 plus e power l of x. I take that back that side. So I will get e of x; e, I am sorry. e power l of x divided by 1 plus e power l of x. I will simplify that to get this expression.

So if we take this up, we can see there is 1 plus e power l of x minus 1. So that will go away, so you will get e power l of x divided by 1 plus e power l of x. So that is, that is essentially the belief expression. Just little bit of algebra to recover the belief from the log odds.

Therefore, that is the reason I keep saying that the log odds ratio is essentially the belief because you can easily go back and forth between the one, between each other, and the reason we keep it this log odds is that our updates are nice and simple additive updates.

So remember that bel x is the probability of x given z1 to t. So bel x at time t is probability of x given z1 to t. So I am going to use the Bayes' rule, just if you remember that, so I am going to take just zt alone. So you can think of this as probability of x given z1 to t minus 1 comma zt.

So I am taking that as my p and I am moving things around. So I am going to rewrite this as probability of zt given x and z1 to t minus 1 times probability of x given z1 to t minus 1 divided by probability of zt given z1 to t minus 1.

Now, we have the Markov property. So the Markov property does not go away. So as soon as I have x, my zt is no longer dependent on my previous measurements. So I can remove this. I can go back to my usual measurement model, which is probability of zt given x. So given my Markov assumption, I can go back to my old measurement model which is probability of zt given x and the rest of it carries over.

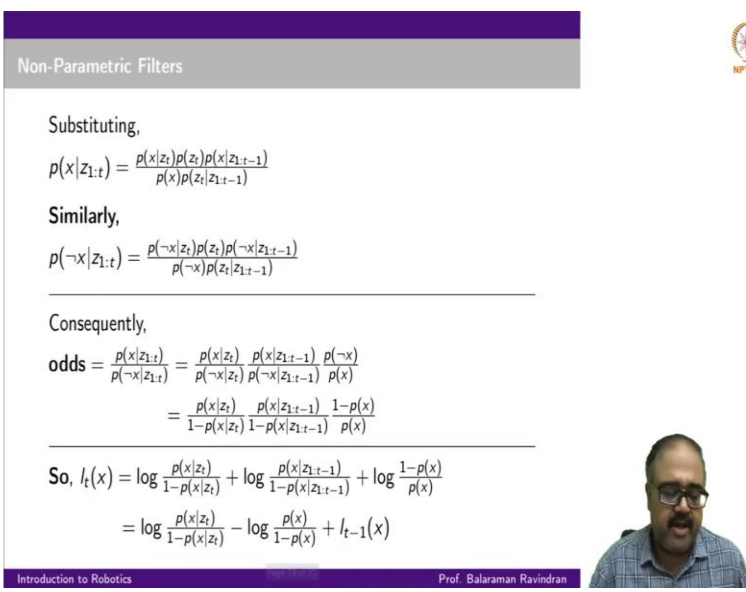
Now, let me try and simplify the measurement model also. I can apply the Bayes' rule again to the measurement model, which is probability of zt given x, and that gives me my inverse model which is probability of x given zt times probability of zt which is the kind of the unconditioned probability of making that measurement zt divided by probability of x, which is my prior probability before I have made any measurements.

Remember we already saw that in the Bayes filter algorithm. So we, this is where it gets introduced here. Now, there are few things which I do not want to compute. Remember I do not want anything where zt is a variable over which I am defining the distribution because z is a very complex space.

I would like to get rid of any dependence on or rather any place where I have to compute a probability over zt. That is the reason we wanted to use the inverse model. So I am happy with this but I do not like this, nor do I like this.

So probability of zt given z1 to t minus 1 is even more complex to determine than probability of zt. So I want to get rid of these somehow. So let me plug this back into the expression and see what we can do.

(Refer Slide Time: 16:33)



So now, substituting my Bayes expansion of probability of z given x, I get this. Probability of x given z1 to t which is my belief of x, belief of x at time t is equal to probability of x given zt. So probability of x given zt into probability of zt divided by probability of x into, so this part is done, into probability of x given z1 to t minus 1 divided by probability of zt given z1 to t minus 1. So those parts come in here.

So I have probability of x given zt times probability of zt divided by probability of x. Here again, the rest are remaining terms from the expansion earlier. Probability of x given z1 to t minus 1 divided by probability of zt given z1 to t minus 1.

This is for probability that x is true. I can write something similarly for the probability that x is false and I would get these quantities. The probability of not x given zt times probability of zt, probability of not x given z1 to t minus 1, the whole divided by probability of x, probability of zt given z 1 to t minus 1.

So here is where our thing comes in. So I am going to take odds. So which is probability of x divided by probability of not x given z1 to t. So this is basically the odds of our belief representation. Probability of x, this is probability of x is true under the belief at time t, this is probability of x being false under the belief time t. So I take that and that is my odds. Please edit that.

So now given that is odds I start dividing this. What is the nice thing when I start dividing? All these zt terms which are common to both x and not x will go away, so this zt goes away and this term also goes away. So what I am left with is probability of xt; sorry, probability of x given zt divided by probability of not x given zt. Probability of x given z 1 to t minus 1 divided by probability of not x given z1 to t minus 1.

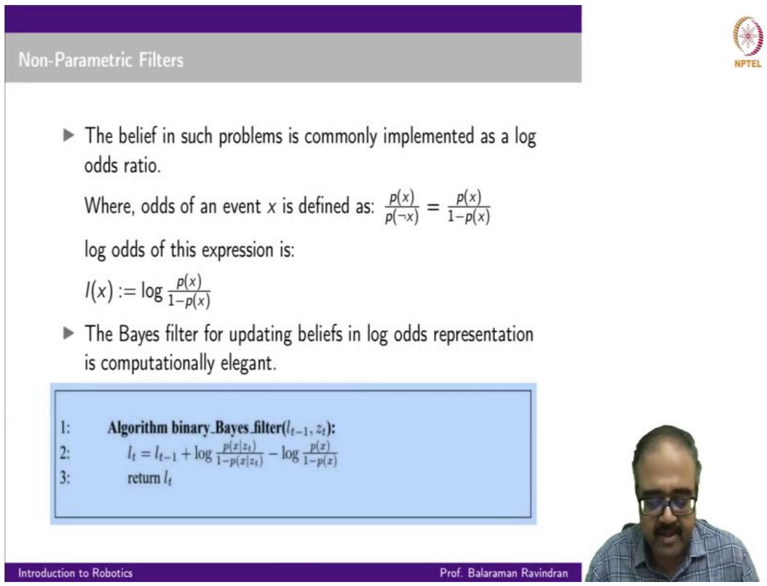
And this will go up and that will stay down. So I will get probability of not x divided by probability of x. And we all know that probability of not x can be written as 1 minus this so I basically get this expression. So all the not x parts get simplified as 1 minus x.

Now, what do I do? I take logs on both sides. So this is my log-odds for the belief at time t. So that is lt of x equal to log of px given zt divided by 1 minus px given zt, this is essentially the second term that we had in the Bayes filter expression, that is the log odds for the inverse measurement model for the current measurements at t.

And then, I get log of p of x given z1 to t minus 1 divided by 1 minus p of x given z1 to t minus 1, and what is that? That is lt minus 1 of x; that is my previous belief. So that is my previous belief and then I have this term which is log of 1 minus p x divided by p x and I can flip it around. I will take minus log p x divided by 1 minus p x which is my l not of x as we saw earlier.

So that is essentially my whole expression. So lt of x equal to lt minus 1 x plus log odds of the inverse measurement model minus l not which is the log odds of the initial belief before I make any measurements. So that gives us the expression for the, the binary Bayes filter update.

(Refer Slide Time: 20:32)



So we will go back here and so, that is the binary Bayes filter. And it might seem a little silly here because we are looking at, looking at a single state variable but later on, one of the applications that we will look at for this algorithm is where there are many, many, many such binary variables that we are trying to estimate in the world at the same time and having a more convenient additive updates are very useful. That is it for this week.