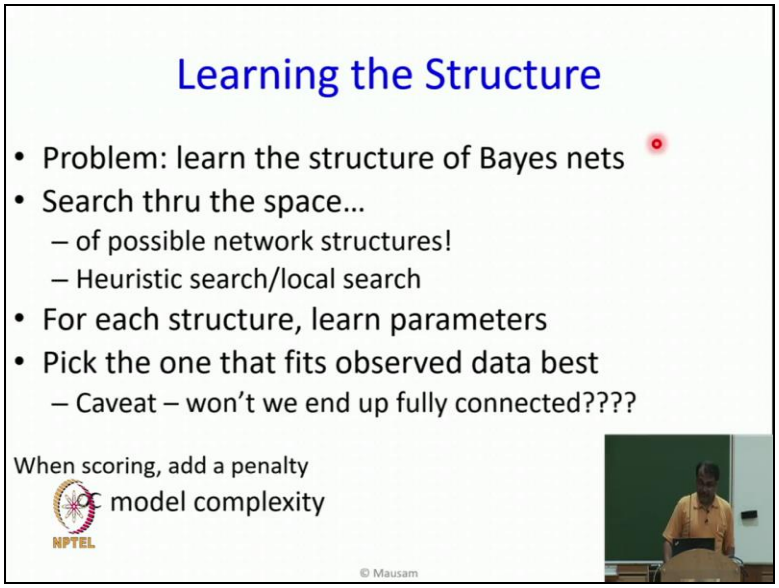
Artificial Intelligence
Prof. Mausam
Department of Computer Science and Engineering
Indian Institute of Technology - Delhi

Lecture-64
Bayesian Networks: Structure Learning and Expectation Maximization

(Refer Slide Time: 00:29)

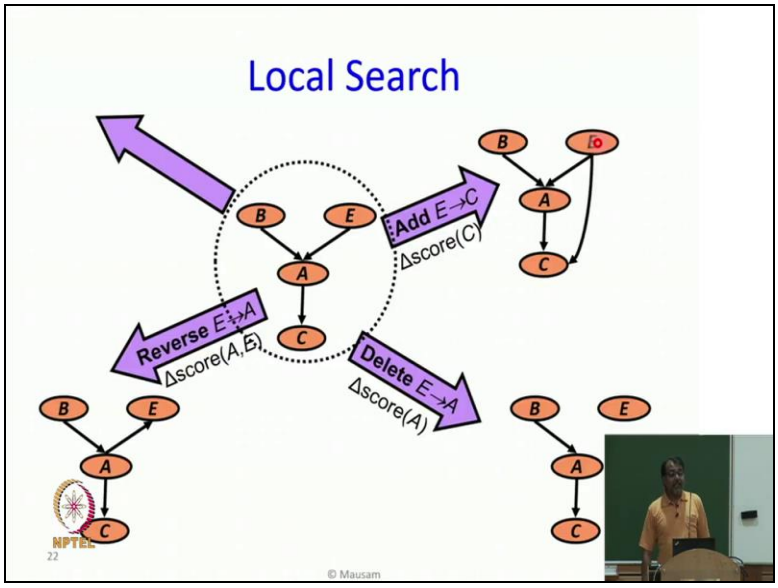


Now I have to do 2 more times of learning. One is learning the structure, up until now we have been saying the domain designer gives us the structure and we use this structure and estimate the probabilities using data. But the domain designer could say I do not know the structure or I only know approximately the structure. Why do not you use the data to figure out the structure also that is for structure learning.

There are lot of papers that were written in structure learning in 2000 but it never became very successful but I will give you the general idea. And the general idea is because structure is a discrete subject let us do search over this structure space. So, how do you do searches over the structure space? So, I give you a structure based on that structure you learn the parameters based on those parameters you see how well it fits the data and then you go back and change make some local changes in the structure right.

And what do you mean by making changes in the structure? Suppose I give you one structure how can you make changes in the structure? What operations could I do? Delete or add an edge very good.

(Refer Slide Time: 01:37)



So, suppose I give you this middle structure like B, E goes to a goes to C then I can define some operations that will change the structure. One operation could be add an edge like I add an edge from E to C. One operation could be reverse an edge like I reverse the edge from E to A. One operation could be delete an edge like I completely remove the edge from E to A. For each set of structure I can learn the probability parameters by maximum likelihood or MAP or whatever.

Then I can compute some score of what is the probability of generating the data and so on. And now I can keep doing this using local search. Can somebody see a problem in this? It is hard to see but if you can see that will be awesome. Can somebody see a problem in this approach? Viswajith you have to be louder, very good. So, Viswajith says I put keep on adding edges by continuously adding edges am I increasing my parameters or reducing my parameters?

Increasing my parameters, if I have more parameters then will my optimization usually lead to a better fit or worse fit? See I have more degrees of freedom. So, it will typically lead to a better fit. So, what would what would I do I will add an edge my fit will increase. Then I add another

edge my fit will increase further I keep adding edges eventually I will have a highly connected Bayesian network with a human less amounts of parameters.

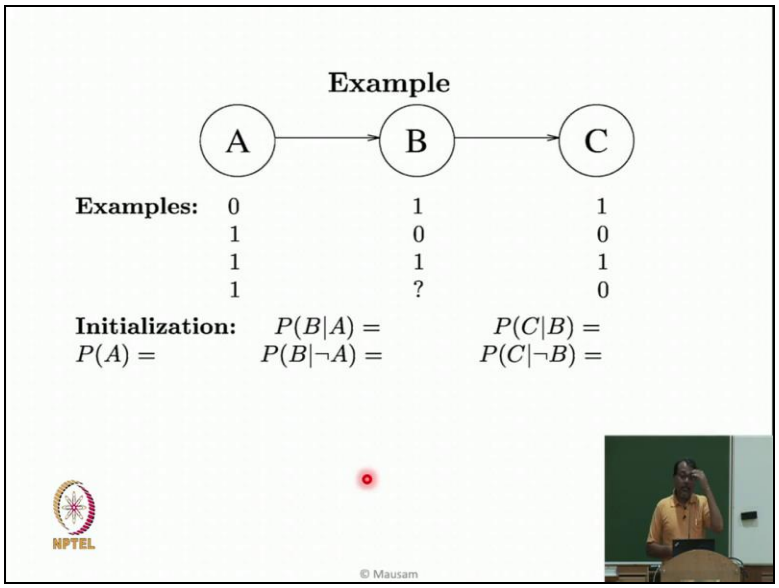
And I have what is called over fit on the training data but it will not generalize very well. So, in order to fix this I use what is called the Occam's razor or the KISS principle. Do you know the KISS principle? So, the KISS principle is keep it simple. What we are saying in the KISS principle or it is also called the Occam's razor in the language of machine learning. Occam's razor states that a more complicated model must pay a heavy penalty for being complicated.

So, the degree of fit should become so much better that you are allowed to add another edge. And we accomplish this by adding a score which is for model complexity. So, maybe number of edges exponential of number of edges or something like that you add to your degree of fit. So, now they trade off they fight with each other if adding an edge increases the probability it reduces the score for model complexity.

But if it really increases the fit much more then it reduces then it is a you should add this edge otherwise you should not add this edge. And how do you trade these off all of that you will learn in machine learning course. So, this is one way of what is called a regularization of the model this is a technical term which says that I want the model to be less complex more regularized. Now the last algorithm I am going over this a little fast I am sure you can see this.

But you know these are important nuggets of insight and you know these are the first time we are introducing this concept we do not have to go into the depth.

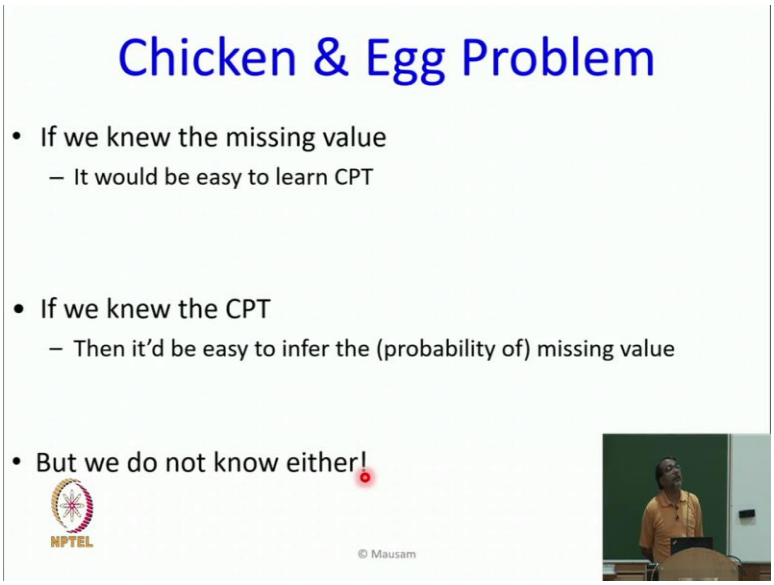
(Refer Slide Time: 05:07)



So, the last algorithm and one of my favorite ones in this class, so, let us say I do not give you a full data let us say I tell you that you know the data is 0 1 1 1 0 0 1 1 1 and then for the fourth data point I forgot to check the value of B. Now this would often happen in the real world. Now you are trying to learn a Bayesian network you send somebody into village you ask people ok go and check the vital statistics of every newborn in the village.

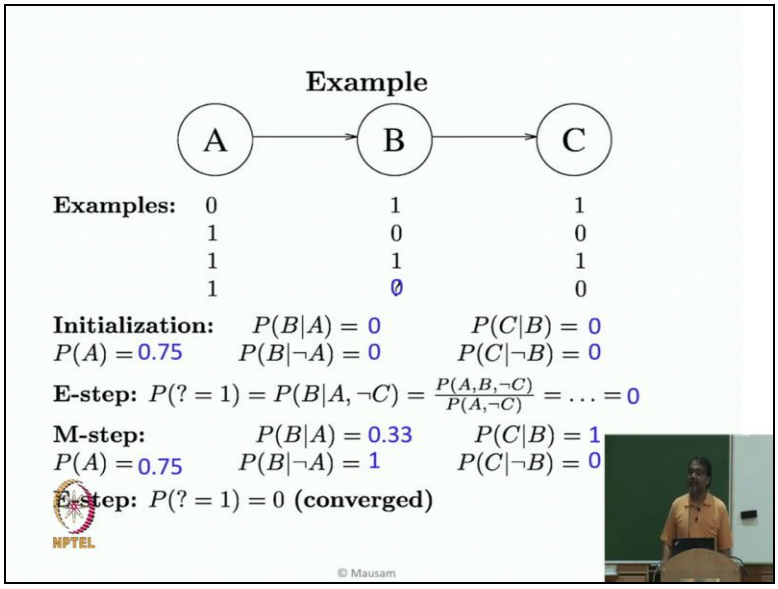
You check their height, weight, gender etcetera and you forgot to check their temperature or whatever. The thermometer was broken that day. You wrote down the temperature but there was the rain and the raindrop is removed the temperature. All of these things happen in the real world. In the real world data is highly noisy which we are not going to talk about but data is also often incomplete. What do I do when I have to learn from this kind of data.

(Refer Slide Time: 06:16)



Now this is what I describe as a chicken-and-egg problem. Why is this a chicken-and-egg problem because if we had the chicken you would have eggs and if we had the eggs we can get chickens but the problem is we neither have chicken or the egg. In other words if we knew the missing value if I knew the value that is question mark?

(Refer Slide Time: 06:40)



Can I do learning that is what we have been doing this whole class maximum-likelihood I may be whatever your favorite let us say I mean you can do learning. However if I knew the parameters of this model if I knew probability of A probability of B given A, probability of C given B if I knew the parameters of this model and then I was asked to compute the missing value it is called imputing the missing value. If I was asked to impute the missing value I would

have to ask the query tell me the probability of B given A is true C is false and the model will use its probability distribution to give me some inference probability.

So, if I knew the model I can do inference to figure out the question mark. If I knew the question mark I can use learning to figure out the model. The problem is I know neither of the 2. Now this would happen a lot in your life I am telling you I have seen this, felt it. You will always have two 2 sets of problems one problem determines the second the second problem determines the first you do not know the answer either of the 2.

In all such situations do what is obvious what is obvious, what is obvious? Assume some initial value let us say of the parameters then you use those parameters to compute the missing value. Use the missing value to compute the parameters. Use the parameters to compute the missing value you keep going back and forth mutual recursion and that is it. What is your name? Roshith oh I am telling you often the obvious thing is the right thing to do and the hard thing to prove anything about.

But in the case of this Bayesian network setting this algorithm is called the EM algorithm and you can do many interesting theorems about it. You can also do that it converges and under certain conditions the estimates are really good. And the reason it is called the EM algorithm is because and when we are doing estimations when we are computing the question mark that is called the expectation step because you are doing inference you are doing the expectation of B given A comma C.

And then we are using its value to estimate probabilities then we are doing some maximization because you know Arg max of theta and so that's called the maximization step. And this now the slides show you the example let us quickly run over the slides. So, let us say I initialize the values like this and then I in the east step I estimated what is the probability of the question mark equal to one I use Bayes rule and that gives me the probability 0.

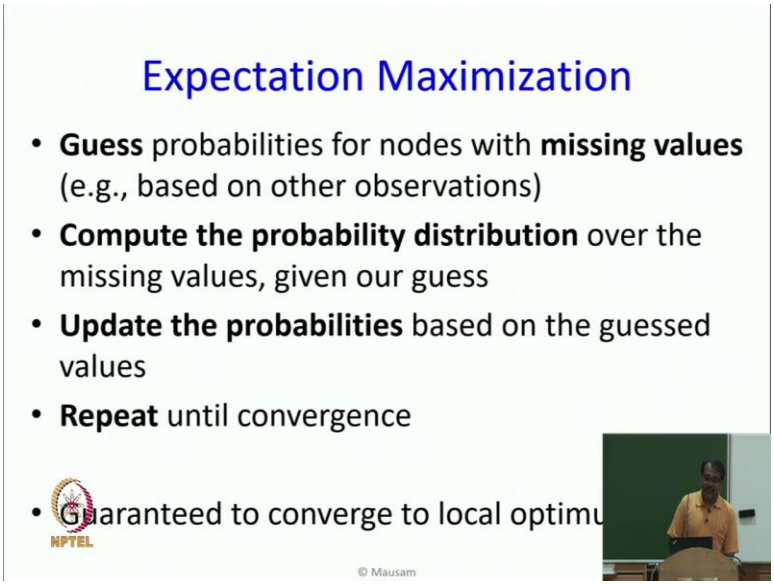
If that value is 0 so then I impute the missing value to 0. If I impute the missing value to 0 and then we estimated the probabilities I will get these probability estimates. Then I reuse these

probabilities to re-estimate the probability the question mark is equal to 1. If I get the same value I have converged and I stop. Now of course this example was written so that it can fit in a slide in practice you do not do it like this.

First of all you do not start with probabilities 0, we do not like zeroes right. Secondly when you do the E step you will not get a value between exactly 0 or 1 you will get a value close somewhere in the middle let us say you get a 0.8. Then you do sampling so either your sample and you say a 0.8 means I will say this value was 1. All you say that 0.8 means that I will make 0.8 of a data point which has value 1 and 0.2th it of a data point which has value 0 and this is what is called soft TM. The first case is called hard TM.

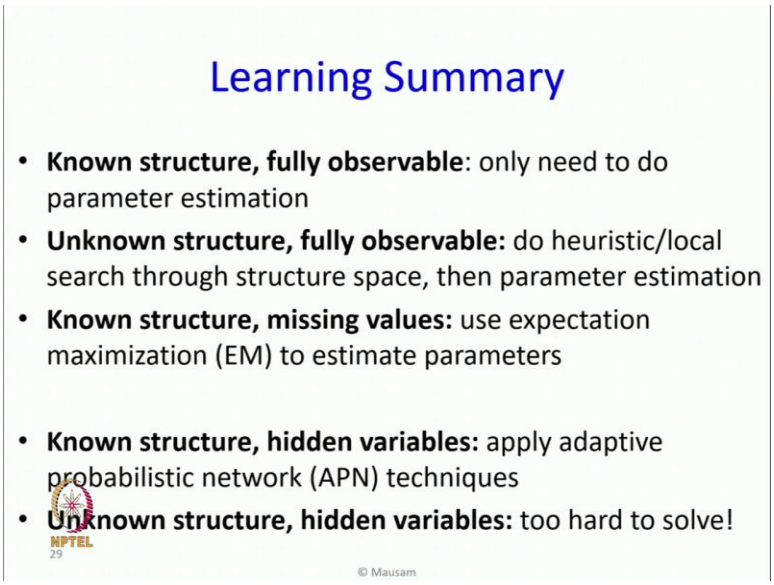
So, pretty much everybody uses soft TM in practice because that converges better and then you convergence condition is that your probability estimates do not move by much. This algorithm is a very important algorithm and has a lot of really nice properties as I said ok.

(Refer Slide Time: 11:16)



So, this is called expectation maximization you guess the probabilities of nodes ones with missing values you use those to compute the probability distribution over the missing values. Then you update the probabilities and then you repeat until convergence. And this is guaranteed to converge to a local optimum.

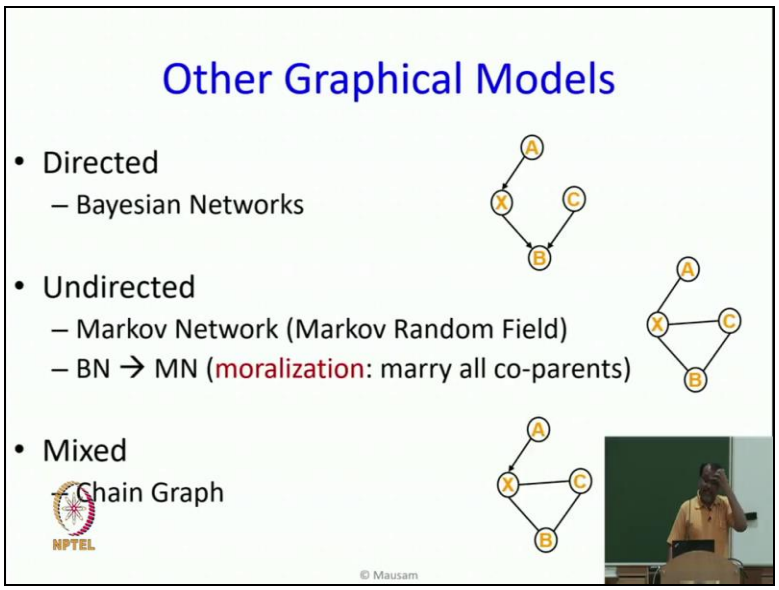
(Refer Slide Time: 11:35)



So, this is the summary of what we learned today if I know the structure and every data point is observed we just need to do parameter estimation we can do ML or MAP or what not if I do not give you the structure but I give you full data then you can do structure learning for example you can do local search through the structure space if I give you known structure but missing values you can do expectation maximization. If I give you unknown structure and missing values you will have to first estimate structure then estimate do EM.

Then re-estimate structure, then do EM that is called structural EM and then they are more harder they are even harder problems that I am not going to talk about.

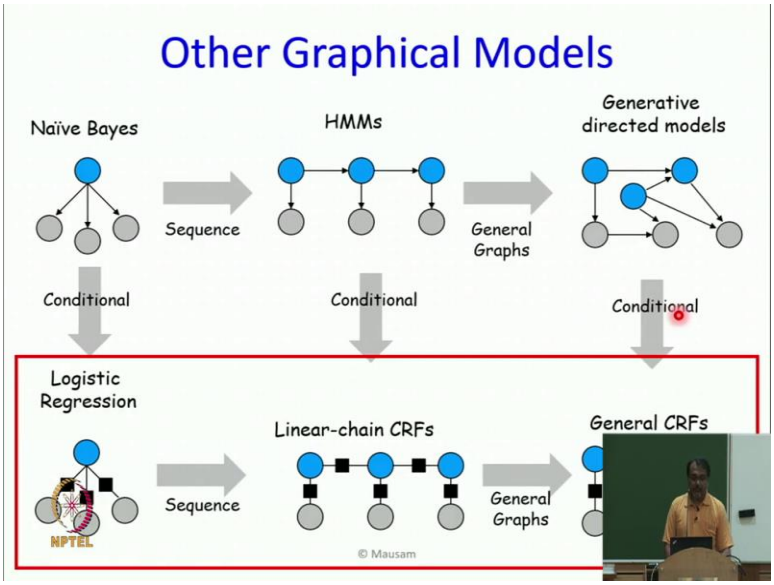
(Refer Slide Time: 12:15)



The last thing I want to say because we are now coming to the close of this whole mix series on Bayesian networks. Is that Bayesian network is one of the many models. It is the first model that we learn it has a lot of intuitive properties. It is a directed model but then there are also undirected models when it is hard to figure out who is going to be the parent we often use undirected models. An example of an undirected model is a Markov network.

And in fact Bayesian network and Markov network makes some similar conditional independence assumptions and in fact you can convert a Bayesian network into Markov network by just marrying all the parents and that process is called modelization. So, if you start with the first Bayesian network. And then you marry all the parents like for example parents of B or X and C you marry them and add an edge and remove the arrows that creates an equivalent Markov network and that process is called modelization.

(Refer Slide Time: 13:17)



There are also models which have both directed and undirected edges for example chain graphs I also want to point out that there is a full theory about the kinds of models people find in different applications. For example we have learned the most difficult or most general form of Bayesian networks the directed generative models. But specific structures where let us say I have one set of parents which transition between each other and a set of observations which depend only on the current parent is what is called a hidden Markov model.

Very famous model so right from the 80's a lot of machine translation speech recognition a lot of NLP applications everything used to be done using hidden Markov models. The very simplest form of Bayesian network is called the naive Bayes algorithm where there is one parent and lots of children and it is a very important algorithm that you will study in your machine learning course.

So after we have discussed what we have discussed you will find many, many frustrations of these models and you will find them to have much simpler characteristics than what we have studied in our class. And of course equivalently there are non generative models like conditional random fields CRF's. There is a general CRF the linear chain CRF and logistic regression which is the most simple form of CRF.

These are heavily used in a lot of machine learning applications and this is where we will stop talking about this and we will start talking about something else from the next class, so thank you.