

Lecture Notes

Probabilistic Graphical Models

This lecture notes covers the 2nd module of the course Probabilistic Graphical Models.

Directed Graphical Models

Directed graphs consist of two things:

- **Nodes:** The nodes represent the random variables.
- **Arrows:** The arrows represent the direction of influence between nodes. The arrows only have a single head.

A variation of directed graphs is called **directed acyclic graphs (DAG)**. The word 'acyclic' means there should be no cycles in the graph.

The **Hammersley-Clifford theorem** provides a way to factor a **joint probability distribution (JPD)**.

$$p(x_1, x_2, \dots, x_n) = \prod_{i=1}^n p(x_i | \text{Parents}(x_i))$$

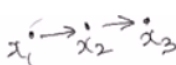
A node might or might not have a parent node. In case there is no parent, the conditional set will be empty for that node.

Working with large joint distributions is **computationally very expensive**. By using the Hammersley-Clifford theorem, one doesn't directly deal with the joint distribution but instead deals with the **conditional probability distributions (CPDs)**, which are far less computationally expensive.

Note that when we say **joint probability distribution (JPD)**, it is by default that we're referring to the model where each node effects every other node. So, the number of parameters in a joint probability distribution will be the combination of the total number of values that each variable can take. Calculating these high number of parameters is expensive as you already know.

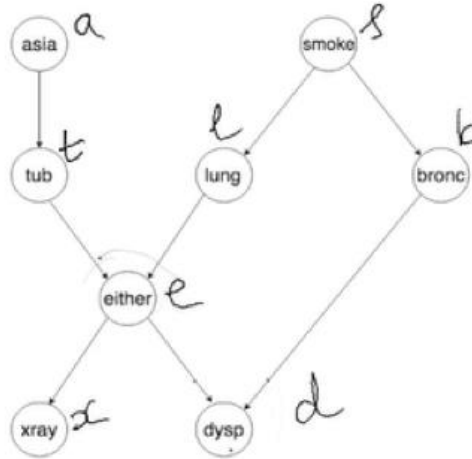
On the other hand, by using a directed graph and the Hammersley-Clifford theorem, you can reduce the total number of parameters significantly. We'll refer to these parameters as the parameters of the **conditional probability distribution (CPD)**.

You also saw in the lectures that HMM is a directed probabilistic graphical model.

In directed graphical model , two variables are said to be independent of each other if they do not have a path connecting them. In probability terms, this is equivalent to the fact that the two variables are independent conditioned on the intermediate variable, i.e. $p(x_3 | x_1, x_2) = p(x_3 | x_2)$ implies that x_3 and x_1 are independent conditioned on x_2 . In other words, conditioning x_3 on x_2 , x_3 becomes

independent of x_1 .

With this theory, for the structure shown below



we can write, $p(a, t, s, l, b, e, x, d) = p(a) \cdot p(t|a) \cdot p(s) \cdot p(l|s) \cdot p(b|s) \cdot p(e|t, l) \cdot p(x|e) \cdot p(d|e, b)$

Hence, a directed probabilistic graphical model has three parts:

1. The **variables** which are the nodes in the graph.
2. The **structure** which is defined by the placement of the nodes and the directed arrows. You'll study how the structure of a graph can be learnt a bit later in the module.
3. The **parameters** which are values of the different entries of the CPD.

You already know that the **CPDs are the model parameters**. In other words, if you need to use a directed model, you need to have the variables, the graph structure and the parameters of the model. If these three things are there then we can ask any kind of query from a directed model.

You have seen that finding out the value $p(x=1|s=1)$ is called as an inference problem.

$p(x=1|s=1) = p(x=1, s=1)/p(s=1)$ which upon expanding becomes

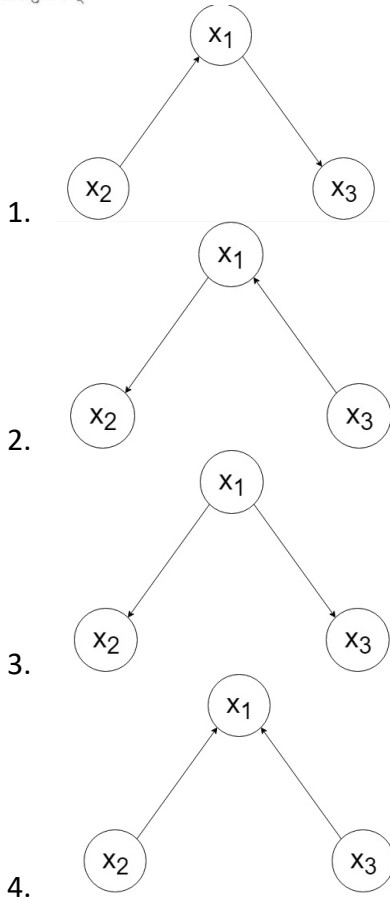
$$p(x=1|s=1) = \sum_{l,t,e,b,d,a} p(a) \times p(s=1) \times p(t|a) \times p(l|s=1) \times p(b|s=1) \times p(e|l,t) \times p(x=1|e) \times p(d|b,e) / p(s=1)$$

and performing variable elimination reduces to

$$p(x=1|s=1) = \sum_{l,t,e,a} p(a) \times p(t|a) \times p(l|s=1) \times p(e|l,t) \times p(x=1|e)$$

Notice that variables b and d have been eliminated and the expression is a simpler one which can be solved using the CPDs.

Now, let's understand the different kinds of structures that are possible inside a directed graph. After all, to come up with a graph, you need to understand all the interdependencies between different variables in the graph. There are four different types of structures that are possible in a directed graph. Note that in the following graphs, the direction of arrows are different.



The structures 1, 2 and 3 are referred to as the non-collider structure while the structure 4 is referred to as the collider structure as the arrows from x_2 and x_3 collide at x_1 . Now, there are some interesting properties.

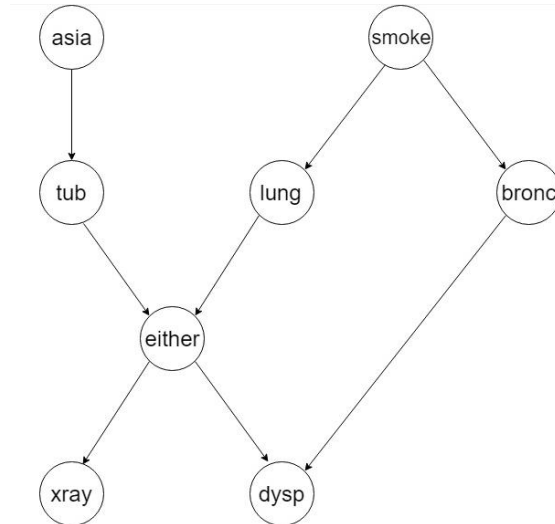
In the **non-collider structure**, x_2 and x_3 are dependent on each other. But once we fix the value of x_1 , they both become independent.

While in the **collider structure**, it's the opposite. x_2 and x_3 are independent of each other. But once we condition on x_2 , that is once we provide the value of x_1 , nodes x_2 and x_3 become dependent on each other.

The probability expression for the graphs 1 and 2 are the same but the direction of causality is different in the two. This is important while deciding the graph structure as the set of variables may satisfy the same equations, but the causality will decide the correct graph structure.

You also looked at the concept of d-separation which states that 2 variables/set of variables are d-separated if there is no path connecting them.

In the Asia example,



1. When an intermediate node is observed: We say that a node is observed when we want to check the conditional independence between two variables A and B given the variable C. The variable C is called the observed variable. Now, suppose you want to check whether the variables 'tub' and 'bronc' are independent or not when 'either' is observed. Now, there are two paths to go from 'tub' to 'bronc'. Let's consider them separately:
 1. **tub-either-dysp-bronc**: In this path, 'either' is observed. Therefore, 'tub' and 'dysp' are independent. If 'tub' and 'dysp' are independent then 'tub' and 'bronc' are also independent and therefore d-separated in this path.
 2. **tub-either-lung-smoke-bronc**: Since 'either' is observed, 'tub' and 'lung' are not independent. Now, 'lung' and 'smoke' are also not independent, and neither are 'smoke' and 'bronc' independent. Therefore, in this path, 'tub' can influence 'bronc' and they're not d-separated in this path.
2. When no intermediate node is observed: This case is a simple one. There is no observed node. Suppose you want to check the dependence between 'either' and 'bronc'. There are only two paths:
 1. **either-dysp-bronc**: This path is a colliding structure. So 'either' and 'bronc' are independent variables. Looking only at this path, the two variables are d-separated.
 2. **either-lung-smoke-bronc**: In this path, consider the microstructure smoke-lung-either. Since it's a non-colliding structure, 'either' is not independent of 'smoke'. And 'smoke' and 'bronc' are also not independent because of the direct edge between them. Therefore, the two variables are not d-separated from this path.

You already know how to make use of a model and do different kinds of inferences using the model. However, the question that comes to the mind is how do you arrive at the model?

Recall that the model consists of three elements:

1. Variables
2. Parameters
3. Structure

When you start working on a problem, you'll only have the dataset and the different variables as the columns in the dataset. To arrive at the model, you need to find out the structure first and then parameters of the model.

Structure learning is, however, still an open problem and even the theory is not mature at the moment to give robust enough results.

In a directed graph, the joint probability can be written as the conditional probabilities of each of the individual nodes given their parents.

$$p(x_1, x_2, \dots, x_n) = \prod_{i=1}^n p(x_i | \text{Parents}(x_i))$$

Now, let's see how does one estimate or find the parameters, that is, all the conditional probabilities in a directed graph. It is recommended that you go through the material on maximum likelihood estimation (MLE). Understanding MLE enables you to better understand the parameter estimation in case of directed models. The likelihood of the dataset can be written as

$$\prod_{j=1}^m p(y_j) = \prod_{j=1}^m \prod_{i=1}^n p(y_{ji} | \text{Parents}(i))$$

which we need to maximize. y_{ji} represents the i th feature of the j th datapoint.

The likelihood can be understood as the probability of the occurrence of the values of the variables present in the dataset for all the datapoints.

The above optimization is quite complex but the results come out to be simple:

$p(x_i = c | \text{parents})$ = number of datapoints where x_i takes the value c and the parents occur / number of datapoints where the parents occur.

Structure learning is the hardest part in graphical models as of today. In most of the problems, the structure is arrived at by talking to domain experts. Take, for instance, the healthcare industry. The doctor has a great deal of knowledge about the causal effects of different symptoms on different diseases and conditions. So, when it comes to making a directed model for disease diagnosis, the graph is arrived at by consulting the doctors.

Similar is the case with other domains. Ideally, we should have an algorithm that creates a graph for us. That's because we expect machine learning and AI to look at the hidden patterns in the data that even domain experts with all their experience can't see. A human is only capable of looking at a small sample out of all the possible cases. That's where ML and AI can play a significant role. You just feed all the data

that was probably witnessed by thousands of doctors, and then you let the machine and algorithms decide which is the best graph, that is, what are the different causal relationships between different variables.

Today, however, we do not have such robust algorithms which can provide us results better than the industry experts. Hence, in general you start with a graph and incrementally change the graph which is better able to capture the conditional independencies represented by the data. After knowing the structure by this iterative process, we perform parameter estimation and then the inference which we wish to make.