Machine Learning for Data Science (CS 4786)

Lecture 18-19: Graphical Models

The text in black outlines main ideas to retain from the lecture. The text in blue give a deeper understanding of how we "derive" or get to the algorithm or method. The text in red are mathematical details for those who are interested. But is not crucial for understanding the basic workings of the method.

1 Representing Probabilistic Models as Graphs

Consider the mixture models we covered in class. We looked at Gaussian mixture models, mixture of multinomials etc. In all of this, the generative story was the same. On every round, we first drew cluster assignment as a random variable from distribution π as $c_t \sim \pi$. Next, give cluster identity to be one of the K clusters, we picked x_t as a a point whose distribution was specific to cluster c_t. In gaussian mixture models for example, we drew $x_t \sim N(\mu_{c_t}, \Sigma_{c_t})$ where $\mu_1, \ldots, \mu_K$ and $\Sigma_1, \ldots, \Sigma_K$ are the means and covariances of the K clusters and cluster c_t is a gaussian centered at μ_{c_t} with covariance Σ_{c_t}. In the mixture of multinomials, x_t was drawn from multinomial with parameter p_{c_t}. Notice a pattern here?

In all these models, only parameterization varies, the relation between variables are same. Graphical models are a graph based representation of probabilistic models that abstract away parameterization and help capture relation between the variables in the model. Here is the grammar for a special kind of graphical models called Bayesian Networks that are catered towards capturing generative models:

- Observed nodes are drawn as nodes in the graph with unshaded circles (with variable names in them)

- Unobserved or latent variables are drawn as nodes in the graph with shaded circles (with variable names in them)

- A directed edge from node A to node B id drawn if A in part generates B. For instance, say we have variable A and A_0 both being gaussian random variable. Now let B be a variable defined as $B = A + A_0 +$ noise where noise is some small random variable with mean 0. In this example, we have directed edges from A and A_0 going out to B.

- Finally, we apply plate notation by drawing a rectangle around a bunch of variables to say we repeat these variables by drawing this set of variables multiple times using the same process. For instance, in GMM example, c_t's and x_t's given c_t are drawn for every t identically from same process n times. Hence the graphical model for a mixture model is given as:

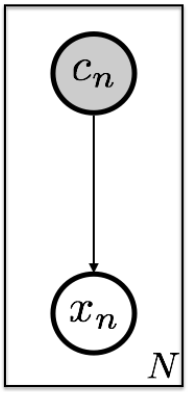

Figure 1: Mixture model

So overall, a Bayesian network is represented as a directed graph where directions indicate which variables generate which ones. It is obvious that we shouldn't have directed cycles in a BN as we can't have a variable generating itself. Such graphs are called directed acyclic graphs (DAG). What kind of relationship between variables can a BN capture?

2 Conditional and Marginal Independence

To understand the powers and limitations of a BN we need to first review concepts of conditional and marginal independences.

Definition 1. *We say that a variable X_i is conditionally independent of X_j given a set of variables A if*

$$P(X_i, X_j|A) = P(X_i|A) \times P(X_j|A)$$

Notice that the above definition is identical to the definition of independence except for the conditioning on A. If A is taken to be the null set, we call this marginal independence. In the above definition using the identity $P(X_i, X_j|A) = P(X_i|X_j, A)P(X_j|A)$ we can also conclude equivalently the definition of conditional independence as:

$$P(X_i, X_j|A) = P(X_i|X_j, A)P(X_j|A) = P(X_i|A) \times P(X_j|A)$$

and so

$$P(X_i|X_j, A) = P(X_i|A)$$

Or in other words, if X_i is conditionally independent of X_j given A, then $P(X_i|X_j, A) = P(X_i|A)$. This is intuitively simple to understand, it says that given A, no more information is revealed about X_i by knowing X_j. Perhaps the best example to have in mind is that of genetic information. Given that you already have your parents (complete) genetic information, knowing your grandparents genetics reveals nothing more than you already know. So you are conditionally independent of your grandparents given your parents.

- Two nodes can be marginally independent but can become conditionally dependent given the third. Example: Say X and Y are independent coin flips (0 for heads and 1 for tails). Hence by definition they are marginally independent. But if $Z = X + Y$ a third variable is revealed, then knowing Z, if we know value of X, then we can determine value of Y. Hence knowing Z, X and Y become dependent.

- Two nodes can be conditionally independent given third yet marginally dependent. Think of genetic information from you and your sibling. Clearly there will be high correlation between biological siblings. However, given complete genetic information of parents, knowing genetic information of one sibling does not provide new insight into genetic information about the other sibling.

3 Local Markov Property and Factorizing over the Graph

Now given the definition of conditional independence, we are ready to understand the power of Bayesian networks. A simple property of a Bayesian network called local markov is that "each variable is conditionally independent of all its non-descendants give parents". This property is intuitive, it says that given parents you gain no more information about yourself knowing anything about your ancestors and other non-descendants (like siblings and their descendants etc.). Thinking of the genetic information case, this is obviously true.

Now the fascinating fact is that, if we are given DAG and the local markov property holds, then we can conclude that the joint probability of the variables factorizes over the graph as follows:

$$P(X_1, \ldots, X_N) = \prod_{t=1}^{N} P(X_t | \text{Parents}(X_t))$$

Proof. To show this fact, we will assume this following factoid about DAGs. "We can always find a topological sort of nodes of a graph". Meaning, we can always find an ordering of the nodes $X_1, \ldots, X_N$ in such a way that for any i, j, if there is a directed edge from X_i to X_j, then X_i has to appear before X_j (or $i < j$). Now we assume that the nodes are ordered according to some topological sort, the key here is that in a topological sort, if we consider a node X_i, all nodes before it are either parents or non-descendants (since they clearly cant be descendants). Using this we show the fact claimed above. To this end note that,

$$\begin{aligned} P(X_1, \ldots, X_N) &= \prod_{t=1}^{N} P(X_t | X_{t-1}, \ldots, X_1) \\ &= \prod_{t=1}^{N} P(X_t | \text{Parents}(X_t)) \end{aligned}$$

where the last step is because of local markov property by which given parents we can remove all other non-descendants which under topological sort is all other nodes other than parents. □

Hence from now on, a Bayesian networks is completely specified by giving a DAG and a bunch of conditional probability tables for each variable given its parents. We will see how to do inference which is answering probability questions about nodes in this BN given other nodes values and learning which is estimating these conditional probabilities given observations.