10-708: Probabilistic Graphical Models 10-708, Spring 2014

1 : Introduction

Lecturer: Eric P. Xing *Scribes: Daniel Silva and Calvin McCarter*

1 Course Overview

In this lecture we introduce the concept of Bayesian Networks and how it is able to represent a probabilistic model with intrinsic independency relationships. We also discuss the different possible structures in a Bayesian network and how they enconde different independency information. Finally, we analyse the equivalency of Bayesian networks and probabilistic models through the concepts of soundness and complenetess.

2 Notation

The notation used throughout the course will be the following:

- X, Y, Z: random variables. E.g. $X \sim \mathcal{N}(\mu, \Sigma)$
- x, y, z: possible values for a random variable. E.g. $P(X = x|Y = y, Z = z) = 0.35$
- $\vec{X}, \vec{Y}, \vec{Z}$: random vectors
- X_1, X_2, X_3: set of related random variables. E.g: $\vec{X} = (X_1, X_2, X_3)$
- $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$: random matrices. E.g. $\boldsymbol{X} = \{\vec{X_1}, \vec{X_2}, \vec{X_3}, \vec{X_4}\}$

3 Representing Multivariate Distributions

The main purpose of the probabilistic graphical modelling is to provide more intuitive tools for dealing with multivariate probabilistic models. Such models are represented by the joint probability of its variables:

$$P(X_1, X_2, ...X_n) \tag{1}$$

PGM's approach is to represent such joint probabilities in terms of conditional probabilities. In a very generalized way, it can be rewritten as:

$$P(X_1, X_2, ...X_n) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2) \dots P(X_n|X_1 \dots X_{n-1}) \tag{2}$$

Equation (2) assumes no prior independency information on the data. The other extreme case is assuming complete independency of the model's random variables. In this case, the joint probability could be rewritten as:

$$P(X_1, X_2, ...X_n) = P(X_1)P(X_2)P(X_3) \dots P(X_n) \tag{3}$$

On the one hand (Equation 2), we have a very general representation of the model, with no restrictions and able to correctly represent any probabilistic model. Nevertheless, it is clearly inefficient, both computationally and intuitively, since it requires the allocation of a huge probability table (if the variables are discrete and with k possible values, the number of entries in that table is of order $O(k^n)$). On the other hand (Equation 3), we have a model that, despite being very efficient computationally (requiring $O(kn)$ memory, instead of an exponential amount), is extremely restrictive in terms of modelling hypotheses. It assumes that all variables are independent, which is far from being the case in most of real-world problems. Probabilistic graphical models will usually target a middle-ground between these two extremes: it tries to find a compromise between computational efficiency and reasonable modelling capacity by assuming an intermediate degree of dependency among the model variables.

4 Graphical Model Illustration: The Dishonest Casino

Imagine a casino game where you bet against the casino. Each one bets $1 and rolls a die and the player with largest number wins the $2. Now imagine you are playing this game for quite a while and that you realize the casino is rolling the number 6 with an unexpectedly high frequency, to the point where you start getting suspicious that their die is loaded, or at least that they are switching between a fair die and a loaded die.
Such a situation would make you ask three questions:

1. EVALUATION: How statistically likely is that sequence of results, considering a “fair die model” ?

2. DECODING: What portions of the sequence were generated by a loaded die ?

3. LEARNING: How loaded is the loaded die (how skewed are the outcome probabilities) and how often is the casino switching between fair and loaded dice?

There are three steps we must follow in order to model this problem and answer the questions above: selecting the variables for the model, selecting the structure of the model, and selecting the associated probabilities. Let’s do it for this example:

1. Selecting the variables: We select X_n as being the observed outcome of the $n-th$ die and Y_n the binary hidden variable that represents if the die used for the $n-th$ outcome was loaded or not.

2. Selecting the structure: This problem is a clear example of a generative model and we will choose a structure that properly represents it (see Figure 1). We assume there is a strategy for picking the dice (that is the reason we have $Y_t \rightarrow Y_{t+1}$) and that the choice of the die at the time t will be responsible for the rolling outcome at time t (which is why we have the relationships $Y_t \rightarrow X_t$)

3. Selecting the probabilities: We’ll select the probabilities associated with the variables and structure described above. Probabilities such as the probabilities of rolling specific numbers for the fair and loaded dices or the probability of switching between loaded and fair dice. We also consider that the same strategy for the die picking is used at any given time, which means that for all t $P(Y_{t+1}|Y_t)$ and $P(X_t|Y_t)$ follow the same distribution.

The final outcome of such a modelling is shown in Figure 1. That particular structure is known as a Hidden Markov Model (HMM) and is highly used in pattern recognition tasks such as POS tagging or speech recognition.

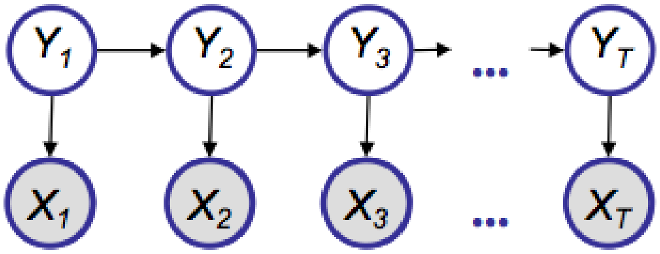

Figure 1: Bayesian network model for the dishonest casino illustration

5 Bayesian Networks

5.1 Introduction

A Bayesian Network (BN) is a probabilistic graphical model that represents a probability distribution through a directed acyclic graph (DAG) that encodes conditional dependency and independency relationships among variables in the model. In this specific graph structure, nodes represent random variables, and each directed edge represents a dependency relationship between the two variables it connects.

5.2 Factorization Theorem

The Factorization Theorem provides the most general form of the joint probability distribution of the model that is consistent with the graph factors according to "node given its parents". Intuitively, this theorem states the "Bayesian Network's version" of the joint probability, which is the model's compromise between Equations (2) and (3). The theorem states that the joint probability is the product of the probability of each node conditioned to its parents:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1:n} P(X_i | Parents(X_i)) \tag{4}$$

5.3 Local Structures & Independencies

In order to extract all the conditional probability information encoded in a Bayesian network we must be able to properly interpret its structure. The understanding of the three local structures represented in Figure 2 will allow us to interpret such models.

- **Cascade**: this local structure is represented in Figure 2(a). The information it encodes is $(X \perp Z | Y)$, i.e. the observation of Y decouples X and Z.
- **Common Cause**: this local structure is represented in Figure 2(b). The information it encodes is $(X \perp Z | Y)$, i.e. the observation of Y decouples X and Z.
- **Common Effect**: this local structure, also known as v-structure, is represented in Figure 2(c). The information it encodes is that X and Z are independent if no prior information is know, but that they are both dependent if Y is observed (the observation of Y couples X and Z). This coupling effect is also known by the name of "explaining away". The main intuition here is that if an effect has two

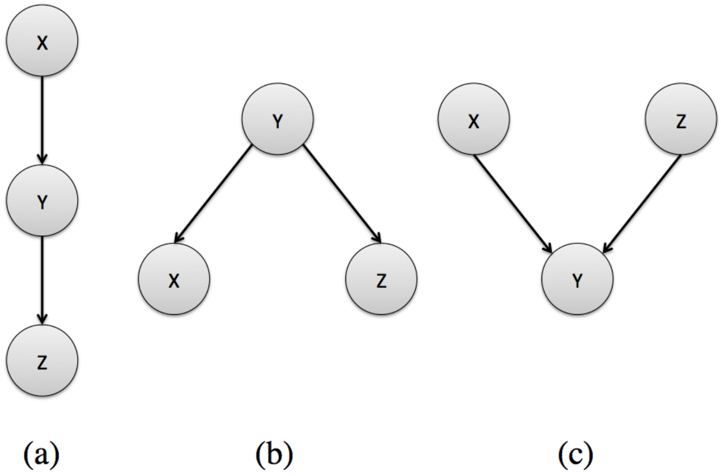

Figure 2: Local structures in Bayesian network

possible causes, observing one of the causes would "explain away" the second cause, making it less likely to be observed.

5.4 Introduction to I-maps

The basic idea of I-maps is that we want independence between variables in the joint distribution to correspond to some notion of separation in the graph (and vice versa). Intuitively, a graph G is an I-map of a distribution P if every independence assertion implied by G is also implied by P. Thus, for G to be an I-map of P, it is necessary that G does not mislead us regarding independencies in P. However, P may have additional independence assertion that are not reflected in G.

More formally, let P be a distribution over $\boldsymbol{X}$. We define $I(P)$ to be the set of independence assertions of the form $(X \perp Y \mid Z)$ that hold in P, and similarly $I(K)$ to be the set of independence assertions associated with graph object K. We say that K is an I-map for a set of independencies I if $I(K) \subseteq I$. Based on these definitions, we now say that G is an I-map for P if G is an I-map for $I(P)$, where $I(G)$ is the set of independencies associated with G.

5.5 Markovian Independence Assumptions

The structure of G asserts independence relations that reflect both local Markov assumptions and global Markov assumptions. Let us consider G, a directed acyclic graph whose nodes represent random variables $X_1, ..., X_n$. The local Markov assumption is that each node X_i is independent of its nondescendants given its parents.

More formally, Let Pa_{X_i} denote the parents of X_i in G, and $\text{NonDescendants}_{X_i}$ denote the variables in the graph that are not descendants of X_i. Then G encodes the following set of local conditional independence assumptions $I_l(G)$: $X_i \perp \text{NonDescendants}_{X_i} \mid Pa_{X_i} : \forall i$.

The global Markov assumptions are related to the notion of D-separation. Two nodes X and Y are D-separated given node Z if they are conditional independent given Z. There are two equivalent ways to establish D-separation in a graph. The first involves derivation of the "moralized" ancestral graph, while the other requires classifying trails as either "active" or "inactive".

To get the moralized ancestral graph, we first construct the ancestral graph which includes only variables X, Y, Z, and all the ancestors of any of these variables (their parents, their parents' parents, etc.). We then "moralize" the ancestral graph by "marrying the parents", inserting an undirected edge between any two variables in the ancestral graph that have a common child.

A trail is a path between three variables, for simplicity X, Y, and Z. There are exactly four ways this trail can be active:

- Causal Trail $X \rightarrow Z \rightarrow Y$: active iff Z is not observed.
- Evidential Trail $X \leftarrow Z \leftarrow Y$: active iff Z is not observed.
- Common Cause $X \leftarrow Z \rightarrow Y$: active iff Z is not observed.
- Common Effect $X \rightarrow Z \leftarrow Y$: active iff Z (or any of its descendents) is observed.

Notice that common effect bahaves differently than the other kinds of active trails. The others are activated by having the "middle" node unobserved, but common effect is activated by observations.

With either criterion for establishing D-separation, we can finally define $I_g(G)$ to be all the independence properties that correspond to D-separation: $I(G) = \{X \perp Y \mid Z : dsep_G(X \perp Y \mid Z)\}$.

5.6 Quantitative Specification of Probability Distributions

Separation properties in the graph imply independence properties about the associated variables. The equivalence theorem states that:

- For a graph G, let D_1 denote the family of all distributions that satisfy $I(G)$. Let D_2 denote the family of all distributions that factor according to G. Then $D_1 \equiv D_2$.

For the graph to be useful, any conditional independence properties we can derive from the graph should hold for the probability distribution that the graph represents. Because of the Equivalence Theorem, we can specify distributions P as a set of conditional probability density (CPD) functions, one for each variable conditioned on its parents. A CPD function for a variable allows us to compute its probability distribution given the values of its parents. For discrete-valued distributions, these can be represented as the familiar conditional probability tables (CPTs).

5.7 Soundness & Completeness

D-separation is sound with respect to the Bayesian Network Factorization Theorem. Soundness is a desirable property because it means that if a distribution P factorizes according to G, then $I(G) \subseteq I(P)$. We would also like the "completeness" property: for any distribution P that factorizes over G, if $(X \perp Y \mid Z) \in I(P)$, then $dsep_G(X \perp Y \mid Z)$. Equivalently, we may ask: if X and Y are not d-separated given Z in G, then are X and Y are dependent in all distributions P that factorize over G? This is false, because a distribution that factorizes over G may yet contain additional "accidental" independencies not asserted by the graph structure. For example, a distribution for "accidentally" independent A and B will still factorize over a graph where $A \rightarrow B$.

More formally we have the two following theorems:

- Let G be a graph corresponding to a Bayes Network. If X and Y are not d-separated given Z in G, then X and Y are dependent in some distribution P that factorizes over G.
- For "almost all" distributions P that factorize over G, $I(P) = I(G)$. By "almost all" we mean all distributions except for a set of measure zero in the space of CPD parameterizations.