CS262a Bayesian Networks

Tingfeng Xia @ UCLA

Winter Quarter, 2022

Preface

This document is intentionally kept short, and is not a complete summary of what is covered in this course. Excerpt from online:

> The objective of this class is to provide an in-depth exposition of knowledge representation, reasoning, and machine learning under uncertainty using the framework of Bayesian networks. Both theoretical underpinnings and practical considerations will be covered, with a special emphasis on constructing and learning graphical models, and on various exact and approximate inference algorithms. Additional topics include logical approaches to probabilistic inference, compilation techniques, sensitivity analysis, undirected graphical models, and statistical relational learning.

Instructor: Professor Adnan Darwiche

Book: Adnan Darwiche. Modeling and Reasoning with Bayesian Networks. Cambridge University Press 2009.

Contents

1 Propositional Logic

1.1 Principle Logical Forms

Inconsistent. Something that never holds; $\text{Mods}(\cdot) =$; $\Pr(\alpha) = 0$

Valid. Something that always holds; $\text{Mods}(\cdot) = \Omega$; $\Pr(\alpha) = 1$

Equivalent. $\text{Mods}(\alpha) = \text{Mods}(\beta)$

Mutual Exclusive. $\text{Mods}(\alpha) \cap \text{Mods}(\beta) = \emptyset$

Exhaustive. $\text{Mods}(\alpha) \cup \text{Mods}(\beta) = \Omega$

Entailment / Implication. $\alpha \models \beta \triangleq \text{Mods}(\alpha) \subseteq \text{Mods}(\beta)$

1.2 Equivalent Forms

- $\text{Mods}(\alpha \wedge \beta) = \text{Mods}(\alpha) \cap \text{Mods}(\beta)$
- $\text{Mods}(\alpha \vee \beta) = \text{Mods}(\alpha) \cup \text{Mods}(\beta)$
- $\text{Mods}(\neg\alpha) = \overline{\text{Mods}(\alpha)}$

1.3 Instantiation Agreement

Two instantiation, each of which can cover a subset of different varaibles, are said to be compatible with each other if they argree on all common variables. Denoted as $\mathbf{x} \sim \mathbf{y}$.

1.4 Information Theory

Entropy.

$$\text{ENT}(X) = -\sum_x \Pr(x) \log \Pr(x) \tag{1.1}$$

where $0 \log 0 = 0$ by convention. With a higher entropy, we say that it is more chaotic.

Conditional Entropy.

$$\text{ENT}(X|Y) = \sum_y \Pr(y)\text{ENT}(X|y) \quad \text{where} \quad \text{ENT}(X|y) = -\sum_x \Pr(x|y) \log \Pr(x|y) \tag{1.2}$$

Conditioning never increases the entropy, i.e.

$$\text{ENT}(X|Y) \leq \text{ENT}(X) \tag{1.3}$$

Mutual Information

$$\begin{aligned}\mathrm{MI}(X;Y) &= \sum_{x,y} \Pr(x,y) \log \frac{\Pr(x,y)}{\Pr(x)\Pr(y)} &(1.4)\\ &= \mathrm{ENT}(X) - \mathrm{ENT}(X|Y) &(1.5)\\ &= \mathrm{ENT}(Y) - \mathrm{ENT}(Y|X) &(1.6)\end{aligned}$$

Conditional Mutual Information

$$\begin{aligned}\mathrm{MI}(X;Y|Z) &= \sum_{x,y,z} \Pr(x,y,z) \log \frac{\Pr(x,y|z)}{\Pr(x|z)\Pr(y|z)} &(1.7)\\ &= \mathrm{ENT}(X|Z) - \mathrm{ENT}(X|Y,Z) &(1.8)\\ &= \mathrm{ENT}(Y|Z) - \mathrm{ENT}(Y|X,Z) &(1.9)\end{aligned}$$

2 Probability Calculus

2.1 Bayesian Conditioning

Bayesian Condition is specified by the formula

$$\Pr(\alpha|\beta) = \frac{\Pr(\alpha \wedge \beta)}{\Pr(\beta)} \tag{2.1}$$

In particular, not to be confused with Bayesian inference (to be added later).

2.2 Independence and Notations

Independence.

$$\begin{aligned}\alpha \perp\!\!\!\perp \beta &\iff \Pr(\alpha|\beta) = \Pr(\alpha) \vee \Pr(\beta) = 0 &(2.2)\\ &\iff \Pr(\alpha \wedge \beta) = \Pr(\alpha)\Pr(\beta) &(2.3)\end{aligned}$$

Conditional Independence.

$$\begin{aligned}(\alpha \perp\!\!\!\perp \beta)|\gamma &\iff \Pr(\alpha|\beta \wedge \gamma) = \Pr(\alpha|\gamma) \vee \Pr(\beta \wedge \gamma) = 0 &(2.4)\\ &\iff \Pr(\alpha \wedge \beta|\gamma) = \Pr(\alpha|\gamma)\Pr(\beta|\gamma) \vee \Pr(\gamma) = 1 &(2.5)\end{aligned}$$

Set Independence

$$I_{\Pr}(X,Z,Y) \iff (x \perp\!\!\!\perp y)|z, \quad \forall x,y,z \in X,Y,Z \tag{2.6}$$

3 Bayesian Networks

3.1 Soft Evidence

3.1.1 All Things Considered Method

We normalize / rescale w according to new evidence.

$$\Pr'(w) = \begin{cases} \frac{\Pr'(\beta)}{\Pr(\beta)}\Pr(w) & \text{if} w \models \beta \\ \frac{\Pr'(\neg\beta)}{\Pr(\neg\beta)}\Pr(w) & \text{if} w \models \neg\beta \end{cases} \tag{3.1}$$

The closed form is called the Jefferey's Rule.

Jeffery's Rule

$$\Pr'(\alpha) = q\Pr(\alpha|\beta) + (1-q)\Pr(\alpha|\neg\beta) \tag{3.2}$$

Jeffery's Rule - General Case

$$\Pr'(\alpha) = \sum_{i=1}^{n} \Pr'(\beta_i)\Pr(\alpha|\beta_i) \tag{3.3}$$

3.1.2 Nothing-else Considered Method

Odds.

$$O(\beta) = \frac{\Pr(\beta)}{\Pr(\neg\beta)} \tag{3.4}$$

Bayes Factor

$$k = \frac{O'(\beta)}{O(\beta)} = \frac{Pr'(\beta)/\Pr'(\neg\beta)}{...} \tag{3.5}$$

from where we can expand and organize

$$\Pr'(\beta) = \frac{k\Pr(\beta)}{k\Pr(\beta) + \Pr(\neg\beta)} \tag{3.6}$$

Closed Form Solution.

$$\Pr'(\alpha) = \frac{k\Pr(\alpha \wedge \beta) + \Pr(\alpha \wedge \neg\beta)}{k\Pr(\beta) + \Pr(\neg\beta)} \tag{3.7}$$

3.2 Noisy Sensors

$$O'(\beta) = \underbrace{\frac{1-f_n}{f_p}}_{k^+} O(\beta) \qquad O'(\beta) = \underbrace{\frac{f_n}{1-f_p}}_{k^-} O(\beta) \tag{3.8}$$

3.3 Markov Assumptions

$$\text{Markov}(G) = \{I_{\text{Pr}}(V, \text{Parents}(V), \text{ND}(V)\}_V \tag{3.9}$$

where ND means non-descendants, and includes all nodes except for V, $\text{Parents}(V)$ and $\text{Descendants}(V)$ (all the way till leaf)

3.4 Graphoid Axioms

Symmetry.

$$I_{\text{Pr}}(X, Z, Y) \iff I_{\text{Pr}}(Y, Z, X) \tag{3.10}$$

Decomposition.

$$I_{\text{Pr}}(X, Z, Y \cup W) \implies I_{\text{Pr}}(X, Z, Y) \wedge I_{\text{Pr}}(X, Z, W) \tag{3.11}$$

Weak Union.

$$I_{\text{Pr}}(X, Z, Y \cup W) \implies I_{\text{Pr}}(X, Z \cup Y, W) \tag{3.12}$$

Contraction.

$$I_{\text{Pr}}(X, Z, Y) \wedge I_{\text{Pr}}(X, Z \cup Y, W) \implies I_{\text{Pr}}(X, Z, Y \cup W) \tag{3.13}$$

Triviality.

$$I_{\text{Pr}}(X, Z, \emptyset) \tag{3.14}$$

3.5 Positive Graphoid Axioms

... includes everything from Graphoid Axioms (Section 3.4) and in addition has

Intersection.

$$I_{\text{Pr}}(X, Z \cup W, Y) \wedge I_{\text{Pr}}(X, Z \cup Y, W) \implies I_{\text{Pr}}(X, Z, Y \cup W) \tag{3.15}$$

3.6 D-seperation Linear Prune Theorem

3.7 D-seperation Properties

Soundness.

$$\mathrm{dsep}_G(X, Z, Y) \implies I_{\mathrm{Pr}}(X, Z, Y) \tag{3.16}$$

(Weak) Completeness. There exists a parametrization Θ that for every DAG G such that

$$I_{\mathrm{Pr}}(X, Z, Y) \iff \mathrm{dsep}_G(X, Z, Y) \tag{3.17}$$

4 Inference by Factor Elimination

4.1 Elimination Trees

Variables. $vars(i)$ denotes the variables mentioned at node i. $vars(i, j)$ denotes all variables mentioned in nodes to the i-side of the graph (inclusive). Hence, it holds that $vars(i) \subseteq vars(i, j)$.

Separators.

$$S_{ij} \triangleq vars(i, j) \cap vars(j, i) \tag{4.1}$$

Clusters.

$$C_i \triangleq vars(i) \cup \bigcup_j S_{ij} \tag{4.2}$$

5 Inference by Conditioning

5.1 Run time Comparison

5.1.1 Variable Elimination (VE)

Let w be the width of the tree, n denote the number of variables, and $|Q|$ as query variable size.

Time Complexity. $O(n \exp(w))$

Space Complexity.

$$O(n \exp(w) + n \exp(|Q|)) \equiv O(n \exp(w)), \quad \text{if } |Q| < \infty \tag{5.1}$$

5.1.2 Massage Passing

One Specific Message. The cost to pass one specific message is

$$\mathcal{O}(\exp(|C_i|)) \tag{5.2}$$

where $|C_i|$ is the cluster size.

Every Message. Since cluster sizes are bounded above by tree width, we can say that passing of every message will have an upper-bound runtime of

$$\mathcal{O}(\exp(w)) \tag{5.3}$$

where w is the width of the elimination tree.

Amount of Messages. The total amount of messages is

$$\mathcal{O}(2(m-1)) \qquad \text{where } m = |V| \tag{5.4}$$

since we have a tree structure, meaning we have exactly $(m-1)$ edges and each edge can have forward backward each once.

All Messages - All Cluster Marginals. The total time to pass all messages and compute all cluster marginals is

$$\mathcal{O}(m\exp(w)) \qquad \text{or} \qquad \mathcal{O}(n\exp(w)), \text{ for } \mathcal{O}(n) \text{ edges.} \tag{5.5}$$

5.1.3 Polytree / Belief Propagation

Runtime. Define k as the max number of parents in the poly tree, then k is the same as the width of elimination tree. Let, also, n denote the number of nodes in the polytree. The algorithm has runtime

$$\mathcal{O}(n\exp(k)) \tag{5.6}$$

5.1.4 Cut Set Conditioning

Time and Space (Total) Complexity.

$$\mathcal{O}(n\exp(k)) \qquad \text{where } n = |N| \text{ and } k \text{ is width} \tag{5.7}$$

5.1.5 Any Space Recursive Cut Set

	no cache	all cache	Δ *no* → *all*
space	$O(wn)$	$O(n \exp(w))$	↑
time	$O(n \exp(w \log n)))$	$O(n \exp(w))$	↓

6 Compiling Bayesian Networks

6.1 Network Polynomials

The network polynomial is a summation over all instantiations of a network,

$$f \triangleq \sum_{z} \prod_{\theta_{x|u} \sim z} \theta_{x|u} \prod_{\lambda_x \sim z} \lambda_x \tag{6.1}$$

6.2 AC Properties

AC Size. of an AC is defined as the number of edges in the circuit.

AC Complexity. is the size of smallest AC that represents the network polynomial.

Decomposable. At each $\star$ node, we need

$$vars(AC_A) \cap vars(AC_B) = \emptyset \tag{6.2}$$

Deterministic. At each $+$ node, we require at most one positive input is non-zero for all *complete instantiation*.

Smooth. At each $+$ node, we require

$$vars(AC_A) = vars(AC_B) \tag{6.3}$$

AC for Marginals. requires decomposable and smooth. This guarantees that sub-circuits are of complete variable instantiations.

AC for Marginals and MPE. requires all three above: decomposable, deterministic, and smooth. The additional determinism guarantees a 1-to-1 mapping between sub-circuits and complete variable instantiations.

6.3 AC Derivative Probabilistic Implications

$$\frac{\partial f}{\partial \lambda_{\mathbf{x}}}(\mathbf{e}) = \Pr(\mathbf{x}, \mathbf{e} - X) \tag{6.4}$$

and

$$\theta_{\mathbf{x}|\mathbf{u}} \frac{\partial f}{\partial \theta_{\mathbf{x}|\mathbf{u}}}(\mathbf{e}) = \Pr(\mathbf{x}, \mathbf{u}, \mathbf{e}) \tag{6.5}$$

6.4 Compilation via Variable Elimination

Circuit Factors. "In a circuit factor, each variable instantiation is mapped to a circuit node instead of a number."

Operations. We use $+(n_1, n_2)$ to denote an addition node that has n_1 and n_2 as its children. Similarly, $\star(n_1, n_2)$ denotes a multiplication node. An operation (multiplication or addition) of two circuit factors $f(X)$ and $f(Y)$ is a factor over variables $Z = X \cup Y$,

$$f(z) = [\star \text{ or } +](f(x), f(y)), \quad \text{where } x \sim z \quad \text{and} \quad y \sim z \tag{6.6}$$

Procedure.

1. **Made nodes for each CPT.** For each family $X|U$, construct nodes $\star(\lambda_x, \theta_{x|u})$ for each instantiation xu of XU.
2. **Eliminate Everything.** We apply VE to eliminate all variables in the network to reach trivial instantiation $\top$ (corresponds to root).

7 Causality - Interventions

7.1 Notations

Causal Effect (CE). of $X = x$ on $Y = y$ can be written as

$$\Pr(Y = y | do(X = x)) \equiv \Pr(y | do(x)) \equiv \Pr(y_x) \tag{7.1}$$

Interventional Distribution. For $\Pr(X, Y, Z)$, the interventional distribution for $do(X = x)$ is denoted as

$$\Pr_{X=x}(Y, Z) \tag{7.2}$$

7.2 Types of Causal Graphs

[7.1]

[7.1] Hidden variables are roots.

7.2.1 Markovian Model

Each hidden variable in a Markovian Model has at most one child. It has an alternative name of "no hidden confounders". In this case, causal effects are always identifiable.

7.2.2 Semi-Markovian Model

Some hidden variable has more than one child. In this case, causal effects are not always identifiable.

7.3 Identifiability Criterion

7.3.1 Causal Effect Rule

The Causal Effect Rule links together association and intervention. It states the following: if **Z** are the parents of X, then

$$\Pr(y|do(x)) = \Pr(y_x) = \sum_{\mathbf{z}} \Pr(y|x, \mathbf{z})\Pr(\mathbf{z}) \tag{7.3}$$

The catch to this formulation is one have to know the parents - meaning that we need to have a correct causal structure prior to using this formula. This is a strong assumption. Often, the structure is exactly what we are after.[7.2]

7.3.2 Backdoor Criteria

A path between X and Y is *blocked* by Z iff

- some collider is not in Z, or
- some non-collider is in Z.

where a collider node is simply a convergent valve defined earlier ($\rightarrow W \leftarrow$). Here we distinguish only between colliders and non-colliders. The Backdoor Criteria states the following: Consider a causal graph G and causal effect $\Pr(y_x)$. A set of variables **Z** satisfis the backdoor criteria iff

- no node in **Z** is a descendant of X,
- **Z** *blocks* every path between X and Y that contains an arrow into X.

Then, if **Z** is a backdoor, then

$$\Pr(y_x) = \sum_{\mathbf{z}} \Pr(y|x, \mathbf{z})\Pr(\mathbf{z}) \tag{7.4}$$

[7.2] Recall that different causal structures can generate the same distribution, and data alone is not enough.

Incompleteness. The backdoor criteria is incomplete. When it identifies that a causal effect has no backdoor, the causal effect can be either identifiable or not identifiable (inconclusive).

7.3.3 Frontdoor Criteria

Consider a causal graph G and causal effect $\Pr(y_x)$. A set of variables $\mathbf{Z}$ satisfies the frontdoor criteria iff .... Then if $\mathbf{Z}$ is a frontdoor, then,

$$\Pr(y_x) = \sum_{\mathbf{z}} \Pr(\mathbf{z}|x) \sum_{x'} \Pr(y|x', \mathbf{z}) \Pr(x') \tag{7.5}$$

7.4 The Do-Calculus

The key idea is to apply a series of rules until we get a formula that is comprised of solely associational quantities. There are three re-write rules in total. $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W}$ are disjoint sets of variables,

Rule 1. Ignoring Observations.

$$\Pr(\mathbf{y}|do(\mathbf{x}), \mathbf{z}, \mathbf{w}) = \Pr(\mathbf{y}|do(\mathbf{x}), \mathbf{w}) \quad \text{if} \quad \text{dsep}_{G_{\overline{\mathbf{X}}}}(\mathbf{Y}, \mathbf{XW}, \mathbf{Z}) \tag{7.6}$$

where we perform dsep test on an altered graph $G_{\overline{\mathbf{X}}}$, rather than the original causal graph G. (Detailed in Section 7.4.1).

Rule 2. Action / Observation Exchange.

$$\Pr(\mathbf{y}|do(\mathbf{x}), (\mathbf{z}), \mathbf{w}) = \Pr(\mathbf{y}|do(\mathbf{x}), \mathbf{z}, \mathbf{w}) \quad \text{if} \quad \text{dsep}_{G_{\overline{\mathbf{X}}\underline{\mathbf{Z}}}}(\mathbf{Y}, \mathbf{XW}, \mathbf{Z}) \tag{7.7}$$

Rule 3. Ignoring Actions.

$$\Pr(\mathbf{y}|do(\mathbf{x}), (\mathbf{z}), \mathbf{w}) = \Pr(\mathbf{y}|do(\mathbf{x}), \mathbf{w}) \quad \text{if} \quad \text{dsep}_{G_{\overline{\mathbf{X}}\overline{Z(W)}}}(\mathbf{Y}, \mathbf{XW}, \mathbf{Z}) \tag{7.8}$$

where encounter a special notation $\overline{Z(W)}$ that means "not all variables in $\mathbf{Z}$, but only those variables in $\mathbf{Z}$ that do not have ancestor in $\mathbf{W}$".

7.4.1 Graph Alterations

The calculus rules we specified earlier performs dsep tests on altered graphs, where

- $G_{\overline{\mathbf{X}}}$ is obtained via removing edges pointing into variables $\mathbf{X}$ from G.
- $G_{\underline{\mathbf{X}}}$ is obtained via removing edges pointing away from variables $\mathbf{X}$ from G.