Causality
Backgrounds II

Maciej Liśkiewicz

University of Lübeck

December, 2022

Last Meeting

- Motivation
- Backgrounds
  - Probabilities and Independencies
  - Graphs and Probabilities
  - Bayesian Networks
  - d-separation
  - An Algorithm for d-Separation (presented on the next slides)

An Algorithm for d-Separation

- Assume P is a distribution that factorizes over a DAG G, i.e. $P(x_1, \ldots, x_n) = \prod_j P(x_j \mid pa_j)$
- Recall, d-separation in such G allows to infer independences of P simply by examining the d-separation in G
- We have given a definition for d-separation

$$(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})_G$$

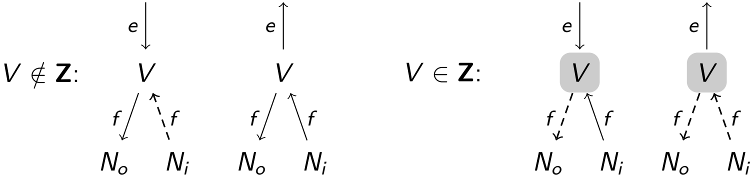
  in a non-constructive way: every path between a node $X \in \mathbf{X}$ and $Y \in \mathbf{Y}$ should be blocked by $\mathbf{Z}$
- Remark: by $\mathbf{X}, \mathbf{Y}$, etc. we will denote sets
- In this lecture we present very elegant and efficient algorithm for d-Separation, called Bayes-Ball, that requires only linear time in the size of the graph
- Bayes-Ball was proposed by Shachter in 1998

An Algorithm for d-Separation

- More precisely,
  - for a given $G = (\mathbf{V} = \{X_1, \ldots, X_n\}, \mathbf{E})$ and two disjoint subsets $\mathbf{X} \subseteq \mathbf{V}$, and $\mathbf{Z} \subseteq \mathbf{V}$
  - the algorithm finds nodes reachable from $\mathbf{X}$ given $\mathbf{Z}$ via open d-paths

An Algorithm for d-Separation

- More precisely,
  - for a given $G = (\mathbf{V} = \{X_1, \ldots, X_n\}, \mathbf{E})$ and two disjoint subsets $\mathbf{X} \subseteq \mathbf{V}$, and $\mathbf{Z} \subseteq \mathbf{V}$
  - the algorithm finds nodes reachable from $\mathbf{X}$ given $\mathbf{Z}$ via open d-paths
- The algorithm runs BFS from $\mathbf{X}$ using the following rules:
  - the Bayes ball goes through the entering top edge e and passes through the node V to nodes N_o (out-node), resp. N_i (in-node)
  - Forbidden passes are marked as dashed arrows
  - The figure shows all possible combinations of types of entering e and leaving edges f and considers two cases: $V \notin \mathbf{Z}$ and $V \in \mathbf{Z}$ (gray)
  - The leaving edge f can correspond to the entering edge e in which case the ball might return to the start node of the entering edge, which is called a *bouncing* ball in the original Bayes-Ball algorithm



function BAYES-BALL($G = (\mathbf{V} = \{X_1, \ldots, X_n\}, \mathbf{E}), \mathbf{X} \subseteq \mathbf{V}, \mathbf{Z} \subseteq \mathbf{V}$)
  function VISIT(f, j) ▷ Visit node X_j from direction f
    if $Mark(f, j) = 0 \wedge X_j \notin \mathbf{X}$ **then** ▷ If no such visit is scheduled
      push (f, j) to queue *toVisit* ▷ Schedule visit
      $Mark(f, j) \leftarrow 1$ ▷ Mark as scheduled

▷ Initial values

  for all $j \in \{1, \ldots, n\}$ **do**
    for all $f \in \{parent, child\}$ **do**
      $Mark(f, j) \leftarrow 0$ ▷ Mark all nodes as unreachable
  $toVisit = ()$ ▷ Empty queue of nodes to visit

▷ Start visiting at the neighbours of **X**

  for all $X_i \in \mathbf{X}$ **do**
    for all $X_j \in Pa(X_i)$ **do** VISIT($child, j$)
    for all $X_j \in Ch(X_i)$ **do** VISIT($parent, j$)

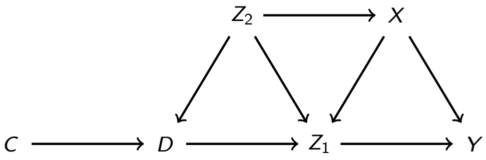
▷ Visit all reachable nodes

  while *toVisit* not empty **do**
    pop (f, k) from queue *toVisit* ▷ Visit the next (*from*, *node*)-tuple
    if $X_k \in \mathbf{Z} \wedge f = parent$ **then** ▷ Node in **Z** bounces back balls from the parent
      for all $X_j \in Pa(X_k)$ **do** VISIT($child, j$)
    if $X_k \notin \mathbf{Z} \wedge f = parent$ **then** ▷ Node not in **Z** passes balls from the parent
      for all $X_j \in Ch(X_k)$ **do** VISIT($parent, j$)
    if $X_k \notin \mathbf{Z} \wedge f = child$ **then** ▷ Node not in **Z** passes and bounces balls from the child
      for all $X_j \in Pa(X_k)$ **do** VISIT($child, j$)
      for all $X_j \in Ch(X_k)$ **do** VISIT($parent, j$)
  return $\{X_j : Mark(f, j) = 1 \text{ for some } f\}$

An Algorithm for d-Separation

Theorem (Bayes-Ball Algorytm)

The algorithm Bayes-Ball($G = (\mathbf{V}, \mathbf{E}), \mathbf{X}, \mathbf{Z}$) returns the set of all nodes reachable from $\mathbf{X}$ via d-paths that are active in G given $\mathbf{Z}$. It runs in linear time in the size of the graph: $|\mathbf{V}| + |\mathbf{E}|$.



Independencies in Bayesian Networks

Basic Independencies

BNs combine two related concepts:

- Independencies in distributions and
- Independencies induced by graphs

Independencies in Bayesian Networks

Basic Independencies

- Let $X_1, X_2, \ldots, X_n$ be random variables
- Let $G = (\mathbf{V}, \mathbf{E})$ be a DAG with $\mathbf{V} = \{X_1, X_2, \ldots, X_n\}$
- We denote parents of X_i in G as Pa_i or $Pa(X_i)$
- Recall, if P admits the factorization

$$P(x_1, \ldots, x_n) = \prod_j P(x_j \mid pa_j)$$

  relative to DAG G, we say
  - that G represents P,
  - that G and P are compatible,
  - that P is Markov relative to G

Independencies in Bayesian Networks

Basic Independencies

- Let $X_1, X_2, \ldots, X_n$ be random variables
- Let $G = (\mathbf{V}, \mathbf{E})$ be a DAG with $\mathbf{V} = \{X_1, X_2, \ldots, X_n\}$
- We denote parents of X_i in G as Pa_i or $Pa(X_i)$
- Recall, if P admits the factorization

$$P(x_1, \ldots, x_n) = \prod_j P(x_j \mid pa_j)$$

  relative to DAG G, we say
    - that G represents P,
    - that G and P are compatible,
    - that P is Markov relative to G
- BN: a DAG G which represents a probability distribution P
- BN = "structure G" + "conditional probability distributions (CPDs)"
- Formally: A BN $\mathcal{B}$ is a pair $\mathcal{B} = (G, P)$ where P factorizes over G, and where P is specified as a set of CPDs associated with G's nodes

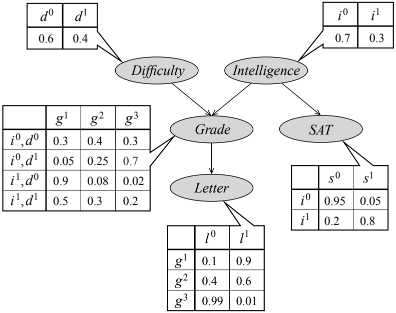
BN Example Used in this Lecture

Koller, Friedman (2009)

Consider the problem faced by a company trying to hire a recent college graduate:

- I – student's intelligence: *low, high*
- D – difficulty of the course: *easy, hard*
- G – student's grade in some course: $1, 2, 3$
- L – the quality of the recommendation letter : *strong, weak*
- S – the student's SAT score: *low, high*

The joint distribution has 48 entries. The corresponding example Bayesian network:



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

Reasoning Pattern in BNs: $P(H = h \mid E = e)$

Independencies in Bayesian Networks

Basic Independencies

- Question: which independencies induces (encodes) a DAG?
- E.g.:
  - $D \rightarrow G \rightarrow L$
  - $R \rightarrow H \leftarrow S$

Independencies in Bayesian Networks

Basic Independencies

- *Alternatively*, the formal semantics of a BN graph G can be defined as a set of *independence* assertions as follows
- Let $De(X_i)$ denote the set of descendants of X_i in G
- Note that
$$\mathbf{V} \setminus De(X_i)$$
are the variables in G that are non descendants of X_i

Independencies in Bayesian Networks

Basic Independencies

- *Alternatively*, the formal semantics of a BN graph G can be defined as a set of *independence* assertions as follows
- Let $De(X_i)$ denote the set of descendants of X_i in G
- Note that
$$\mathbf{V} \setminus De(X_i)$$
are the variables in G that are non descendants of X_i
- Then G in a BN encodes the following set of conditional independence assumptions, called the local independencies, and denoted by $\mathcal{I}_{local}(G)$:
$$\forall X_i \quad (X_i \perp\!\!\!\perp \mathbf{V} \setminus (De(X_i) \cup Pa(X_i)) \mid Pa(X_i))$$

Independencies in Bayesian Networks

Basic Independencies

- *Alternatively*, the formal semantics of a BN graph G can be defined as a set of *independence* assertions as follows
- Let $De(X_i)$ denote the set of descendants of X_i in G
- Note that
$$\mathbf{V} \setminus De(X_i)$$
are the variables in G that are non descendants of X_i
- Then G in a BN encodes the following set of conditional independence assumptions, called the local independencies, and denoted by $\mathcal{I}_{local}(G)$:
$$\forall X_i \quad (X_i \perp\!\!\!\perp \mathbf{V} \setminus (De(X_i) \cup Pa(X_i)) \mid Pa(X_i))$$
- In other words, the local independencies state that each variable X_i is conditionally independent of its non-descendants given its parents

Independencies in Bayesian Networks

Basic Independencies

- *Alternatively*, the formal semantics of a BN graph G can be defined as a set of *independence* assertions as follows
- Let $De(X_i)$ denote the set of descendants of X_i in G
- Note that
$$\mathbf{V} \setminus De(X_i)$$
are the variables in G that are non descendants of X_i
- Then G in a BN encodes the following set of conditional independence assumptions, called the local independencies, and denoted by $\mathcal{I}_{local}(G)$:
$$\forall X_i \quad (X_i \perp\!\!\!\perp \mathbf{V} \setminus (De(X_i) \cup Pa(X_i)) \mid Pa(X_i))$$
- In other words, the local independencies state that each variable X_i is conditionally independent of its non-descendants given its parents
- We show that this definition is, in fact, equivalent with our first definition of a BN as a DAG annotated with CPDs, which define a joint distribution P via the chain rule

Independencies in Bayesian Networks

I-map

- Let P be a distribution over $\mathbf{V} = \{X_1, X_2, \ldots, X_n\}$
- We define $\mathcal{I}(P)$ to be the set of independence assertions of the form $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})$ that hold in P

Example

- Consider a joint probability P over two independent random variables X and Y

X	Y	$P(X, Y)$
0	0	0.08
0	1	0.32
1	0	0.12
1	1	0.48

- It is easy to see that $(X \perp\!\!\!\perp Y)$ in P. E.g. we have
- $P(X = 1) = 0.6, P(Y = 1) = 0.8$, $P(X = 1) \cdot P(Y = 1) = 0.48$ and
- $P(X = 1, Y = 1) = 0.48$
- Thus $\mathcal{I}(P) = \{(X \perp\!\!\!\perp Y)\}$

Independencies in Bayesian Networks

I-map

Example

- Consider a joint probability P over two independent random variables X and Y

X	Y	$P(X,Y)$
0	0	0.08
0	1	0.32
1	0	0.12
1	1	0.48

- It is easy to see that $(X \perp\!\!\!\perp Y)$ in P. E.g. we have
- $P(X=1) = 0.6, P(Y=1) = 0.8,\ P(X=1) \cdot P(Y=1) = 0.48$ and
- $P(X=1, Y=1) = 0.48$
- Thus $\mathcal{I}(P) = \{(X \perp\!\!\!\perp Y)\}$
- For the distribution P':

X	Y	$P'(X,Y)$
0	0	0.10
0	1	0.16
1	0	0.64
1	1	0.10

we have $(X \perp\!\!\!\perp Y) \notin \mathcal{I}(P')$. In fact, $\mathcal{I}(P') = \emptyset$

Independencies in Bayesian Networks

I-map

- Let P be a distribution over $\mathbf{V} = \{X_1, X_2, \ldots, X_n\}$
- Let $\mathcal{I}(P) = \{(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) : \ \mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{V}\}$
- We can now express the statement that "P satisfies the local independencies associated with G" as

$$\mathcal{I}_{local}(G) \subseteq \mathcal{I}(P)$$

- In this case, we say that G is an independency map, I-map for short, for P

Independencies in Bayesian Networks

- However, it is useful to define the concept I-map more broadly
- Let G be a DAG with a set of independencies $\mathcal{I}(G)$
- We say that

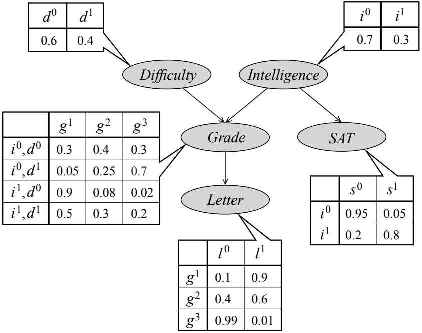
$$G \text{ is an I-map for } P \text{ if } \mathcal{I}(G) \subseteq \mathcal{I}(P)$$

- Intuitively, a DAG G is an I-map of a distribution P if all Markov assumptions implied by G are satisfied by P
- From direction of inclusion $\mathcal{I}(G) \subseteq \mathcal{I}(P)$:
  - Distribution can have more CIs than the graph
  - Graph does not mislead in independencies existing in P: any CI that G asserts must also hold in P

Independencies in Bayesian Networks

I-map

Can we read off all independencies $\mathcal{I}(G)$ from a BN defined as a DAG annotated with CPDs?



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^1, d^0	0.9	0.08	0.02
i^1, d^1	0.5	0.3	0.2

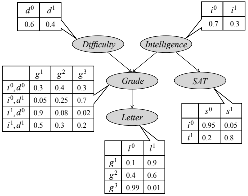
	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

Independencies in Bayesian Networks

I-map

- Independencies in a DAG G with CPDs



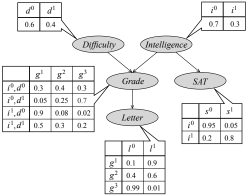
- G encodes factorization: $P(d,i,g,s,l) = P(d)P(i)P(g|i,d)P(s|i)P(l|g)$
- Local (conditional) independencies $\mathcal{I}_{local}(G)$ are the following

$(L \perp\!\!\!\perp I, D, S \mid G)$	L is cond. indep. on all other variables given parent G
$(S \perp\!\!\!\perp D, G, L \mid I)$	L is cond. indep. on all other variables given parent I
$(G \perp\!\!\!\perp S \mid D, I)$	G is cond. indep. on S given parents
	but even given parents, G is *not* cond. indep. on descendent L
$(I \perp\!\!\!\perp D)$	variables with no parents are marginally independent
$(D \perp\!\!\!\perp I, S)$	D is marginally indep. on non-descendants I and S

Independencies in Bayesian Networks

I-map

- Independencies in a DAG G with CPDs



- G encodes factorization: $P(d,i,g,s,l) = P(d)P(i)P(g|i,d)P(s|i)P(l|g)$
- Local (conditional) independencies $\mathcal{I}_{local}(G)$ are the following

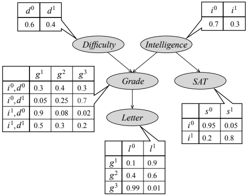
$(L \perp\!\!\!\perp I, D, S \mid G)$	L is cond. indep. on all other variables given parent G
$(S \perp\!\!\!\perp D, G, L \mid I)$	L is cond. indep. on all other variables given parent I
$(G \perp\!\!\!\perp S \mid D, I)$	G is cond. indep. on S given parents
	but even given parents, G is *not* cond. indep. on descendent L
$(I \perp\!\!\!\perp D)$	variables with no parents are marginally independent
$(D \perp\!\!\!\perp I, S)$	D is marginally indep. on non-descendants I and S

- Parents of a variable X shield it from a "probabilistic influence": *if values of parents are known, we do not learn more about X when we know additionally the values of non-descendants (excluding parrents)*

Independencies in Bayesian Networks

I-map

- Independencies in a DAG G with CPDs



- G encodes factorization: $P(d,i,g,s,l) = P(d)P(i)P(g|i,d)P(s|i)P(l|g)$
- Local (conditional) independencies $\mathcal{I}_{local}(G)$ are the following

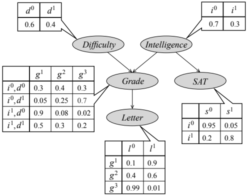
$(L \perp\!\!\!\perp I, D, S \mid G)$	L is cond. indep. on all other variables given parent G
$(S \perp\!\!\!\perp D, G, L \mid I)$	L is cond. indep. on all other variables given parent I
$(G \perp\!\!\!\perp S \mid D, I)$	G is cond. indep. on S given parents
	but even given parents, G is *not* cond. indep. on descendent L
$(I \perp\!\!\!\perp D)$	variables with no parents are marginally independent
$(D \perp\!\!\!\perp I, S)$	D is marginally indep. on non-descendants I and S

- Parents of a variable X shield it from a "probabilistic influence": *if values of parents are known, we do not learn more about X when we know additionally the values of non-descendants (excluding parrents)*
- Information about descendants can change beliefs about a node

Independencies in Bayesian Networks

I-map

- Independencies in a DAG G with CPDs



- G encodes factorization: $P(d,i,g,s,l) = P(d)P(i)P(g|i,d)P(s|i)P(l|g)$
- Local (conditional) independencies $\mathcal{I}_{local}(G)$ are the following

$(L \perp\!\!\!\perp I, D, S \mid G)$	L is cond. indep. on all other variables given parent G
$(S \perp\!\!\!\perp D, G, L \mid I)$	L is cond. indep. on all other variables given parent I
$(G \perp\!\!\!\perp S \mid D, I)$	G is cond. indep. on S given parents
	but even given parents, G is *not* cond. indep. on descendent L
$(I \perp\!\!\!\perp D)$	variables with no parents are marginally independent
$(D \perp\!\!\!\perp I, S)$	D is marginally indep. on non-descendants I and S

- Using properties satisfied by the above CI relations we also get, for example:
- $(L \perp\!\!\!\perp I, D \mid G), (L \perp\!\!\!\perp I, S \mid G), (L \perp\!\!\!\perp I \mid G)$, etc.
- In general: $\mathcal{I}_{local}(G) \subseteq \mathcal{I}(G)$ and, typically, the inclusion is proper

Independencies in Bayesian Networks

I-map

G is an I-map for P if $\mathcal{I}(G) \subseteq \mathcal{I}(P)$

- **Example** Consider the following DAGs

DAG		$\mathcal{I}(G)$
G_0 :	$X \quad\quad Y$	$\mathcal{I}(G_0) = \{(X \perp\!\!\!\perp Y)\}$
G_1 :	$X \rightarrow Y$	$\mathcal{I}(G_1) = \emptyset$
G_2 :	$X \leftarrow Y$	$\mathcal{I}(G_2) = \emptyset$

- For the probability:

X	Y	$P(X, Y)$
0	0	0.08
0	1	0.32
1	0	0.12
1	1	0.48

we have $\mathcal{I}(P) = \{(X \perp\!\!\!\perp Y)\}$

- Thus:
  - G_0 is an I-map of P, since $\{(X \perp\!\!\!\perp Y)\} \subseteq \mathcal{I}(P)$
  - G_1 is an I-map of P, since $\emptyset \subseteq \mathcal{I}(P)$
  - G_2 is an I-map of P, since $\emptyset \subseteq \mathcal{I}(P)$

Independencies in Bayesian Networks

I-map

G is an I-map for P if $\mathcal{I}(G) \subseteq \mathcal{I}(P)$

- **Example** Consider the following DAGs

DAG	$\mathcal{I}(G)$
$G_0: \quad X \qquad Y$	$\mathcal{I}(G_0) = \{(X \perp\!\!\!\perp Y)\}$
$G_1: \quad X \rightarrow Y$	$\mathcal{I}(G_1) = \emptyset$
$G_2: \quad X \leftarrow Y$	$\mathcal{I}(G_2) = \emptyset$

- For the probability:

X	Y	$P(X, Y)$
0	0	0.08
0	1	0.32
1	0	0.12
1	1	0.48

  we have $\mathcal{I}(P) = \{(X \perp\!\!\!\perp Y)\}$

- Thus:
  - G_0 is an I-map of P, since $\{(X \perp\!\!\!\perp Y)\} \subseteq \mathcal{I}(P)$
  - G_1 is an I-map of P, since $\emptyset \subseteq \mathcal{I}(P)$
  - G_2 is an I-map of P, since $\emptyset \subseteq \mathcal{I}(P)$

- If G is an I-map of P then it captures some of the independences, but not necessarily all of them

Independencies in Bayesian Networks

I-map

G is an I-map for P if $\mathcal{I}(G) \subseteq \mathcal{I}(P)$

- **Example** Consider the following DAGs

DAG	$\mathcal{I}(G)$
$G_0 : \quad X \qquad Y$	$\mathcal{I}(G_0) = \{(X \perp\!\!\!\perp Y)\}$
$G_1 : \quad X \rightarrow Y$	$\mathcal{I}(G_1) = \emptyset$
$G_2 : \quad X \leftarrow Y$	$\mathcal{I}(G_2) = \emptyset$

- For the probability:

X	Y	$P'(X, Y)$
0	0	0.10
0	1	0.16
1	0	0.64
1	1	0.10

  we have $\mathcal{I}(P') = \emptyset$

- Thus:
  - G_0 is an not an I-map of P, since $\{(X \perp\!\!\!\perp Y)\} \not\subseteq \mathcal{I}(P')$
  - G_1 is an I-map of P, since $\emptyset \subseteq \mathcal{I}(P')$
  - G_2 is an I-map of P, since $\emptyset \subseteq \mathcal{I}(P')$

Independencies in Bayesian Networks

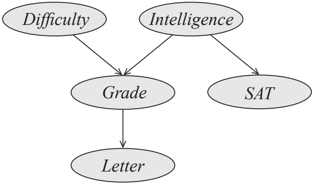
I-map and Factorization

- A DAG G of a BN encodes a factorization of a distribution P
- Every distribution P for which G is an I-map should satisfy the CIs assumptions encoded by$G$
- We show the fundamental connection between the CIs encoded by the structure G and the factorization of the distribution P
- We discuss two directions:
  - I-map to Factorization
  - Factorization to I-map

Independencies in Bayesian Networks

I-map to Factorization

- A DAG representation of our example BN



encodes the factorization of the joint distribution:

$$P(i, d, g, l, s) = P(d)P(i)P(g|i, d)P(s|i)P(l|g)$$

- However we can also factorize P as follows

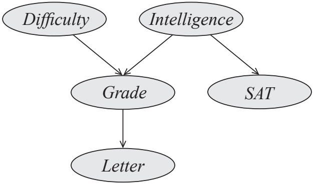
$$P(i, d, g, l, s) = P(i)P(d|i)P(g|i, d)P(l|i, d, g)P(s|i, d, g, l)$$

- This factorization relies on no assumptions and it holds for *any* joint distribution P. Why?
- A drawback: It provides an inefficient representation for CPDs

Independencies in Bayesian Networks

I-map to Factorization

- A DAG representation of our example BN



encodes the factorization of the joint distribution:

$$P(i, d, g, l, s) = P(d)P(i)P(g|i, d)P(s|i)P(l|g)$$

- However we can also factorize P as follows

$$P(i, d, g, l, s) = P(i)P(d|i)P(g|i, d)P(l|i, d, g)P(s|i, d, g, l)$$

- This factorization relies on no assumptions and it holds for *any* joint distribution P. Why?
- A drawback: It provides an inefficient representation for CPDs
- The key observation which allows the compact factorized representation: take into consideration only CIs of distributions for which G should be an I-map

Independencies in Bayesian Networks

I-map to Factorization

Example: From CIs $\mathcal{I}(P)$ to factorization of P

- Consider our example, with $\mathbf{V} = \{I, D, G, L, S\}$
- Due to the chain rule, we get, e.g., the following factorization

$$P(i, d, g, l, s) \;=\; P(i) \cdot P(d|i) \cdot P(g|i, d) \cdot P(l|i, d, g) \cdot P(s|i, d, g, l)$$

- Let us assume that the resulting DAG is an I-map for the distribution P for our example "student"

Independencies in Bayesian Networks

I-map to Factorization

Example: From CIs $\mathcal{I}(P)$ to factorization of P

- Consider our example, with $\mathbf{V} = \{I, D, G, L, S\}$
- Due to the chain rule, we get, e.g., the following factorization

$$P(i, d, g, l, s) \;=\; P(i) \cdot P(d|i) \cdot P(g|i, d) \cdot P(l|i, d, g) \cdot P(s|i, d, g, l)$$

- Let us assume that the resulting DAG is an I-map for the distribution P for our example "student"
- In particular, we assume implicitly that Intelligence (of a student) and Difficulty (of the course) are independent, i.e. we have that $(D \perp\!\!\!\perp I) \in \mathcal{I}(P)$
- This means: $P(d|i) = P(d)$

Independencies in Bayesian Networks

I-map to Factorization

Example: From CIs $\mathcal{I}(P)$ to factorization of P

- Consider our example, with $\mathbf{V} = \{I, D, G, L, S\}$
- Due to the chain rule, we get, e.g., the following factorization

$$P(i,d,g,l,s) \;=\; P(i) \cdot P(d|i) \cdot P(g|i,d) \cdot P(l|i,d,g) \cdot P(s|i,d,g,l)$$

- Let us assume that the resulting DAG is an I-map for the distribution P for our example "student"
- In particular, we assume implicitly that Intelligence (of a student) and Difficulty (of the course) are independent, i.e. we have that $(D \perp\!\!\!\perp I) \in \mathcal{I}(P)$
- This means: $P(d|i) = P(d)$
- Similarly, we take assertion: "the professor's recommendation letter depends only on the student's grade in the class", i.e. that we have $(L \perp\!\!\!\perp I, D, S \mid G) \in \mathcal{I}(P)$
- Hence $P(l|i,d,g) = P(l|g)$

Independencies in Bayesian Networks

I-map to Factorization

Example: From CIs $\mathcal{I}(P)$ to factorization of P

- Consider our example, with $\mathbf{V} = \{I, D, G, L, S\}$
- Due to the chain rule, we get, e.g., the following factorization

$$P(i,d,g,l,s) \; = \; P(i) \cdot P(d|i) \cdot P(g|i,d) \cdot P(l|i,d,g) \cdot P(s|i,d,g,l)$$

- Let us assume that the resulting DAG is an I-map for the distribution P for our example "student"
- In particular, we assume implicitly that Intelligence (of a student) and Difficulty (of the course) are independent, i.e. we have that $(D \perp\!\!\!\perp I) \in \mathcal{I}(P)$
- This means: $P(d|i) = P(d)$
- Similarly, we take assertion: "the professor's recommendation letter depends only on the student's grade in the class", i.e. that we have $(L \perp\!\!\!\perp I, D, S \mid G) \in \mathcal{I}(P)$
- Hence $P(l|i,d,g) = P(l|g)$
- Finally, we assert $(S \perp\!\!\!\perp D, G, L \mid I) \in \mathcal{I}(P)$ that implies: $P(s|i,d,g,l) = P(s|i)$
- This leads to the following factorization

$$P(i,d,g,l,s) \; = \; P(i) \cdot P(d) \cdot P(g|i,d) \cdot P(l|g) \cdot P(s|i)$$

- Now we are ready to show the first direction of the fundamental connection between the CIs encoded by G and the factorization of P

Theorem (I-map to Factorization)

Let G be a BN structure over a set of random variables $\mathbf{V}$, and let P be a joint distribution over the same space. If G is an I-map for P, then P factorizes according to G.

- Assume G is an I-map for P, i.e. $\mathcal{I}_{local}(G) \subseteq \mathcal{I}(P)$
- To prove the theorem, we need to show that P factorizes according to G
- To this end, we generalise our example analysis

Independencies in Bayesian Networks

Factorization to I-map

- The opposite direction says the following:

Theorem (Factorization to I-map)

Let G be a BN structure over a set of random variables $\mathbf{V}$, and let P be a joint distribution over the same space. If P factorizes according to G, then G is an I-map for P.

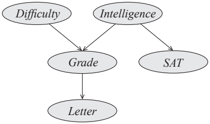
- Let P be some distribution that factorizes according to G
- To prove the theorem, we need to show that $\mathcal{I}_{local}(G) \subseteq \mathcal{I}(P)$

Independencies in Bayesian Networks

Factorization to I-map

Example: Illustration of the theorem

- Assume the DAG representation:



that encodes the factorization: $P(i, d, g, l, s) = P(i) \cdot P(d) \cdot P(g \mid i, d) \cdot P(s \mid i) \cdot P(l \mid g)$

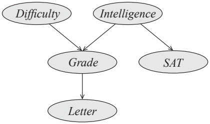
- Consider e.g. variable S; The analysis for other variables is analogous

Independencies in Bayesian Networks

Factorization to I-map

Example: Illustration of the theorem

- Assume the DAG representation:



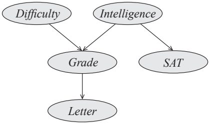
  that encodes the factorization: $P(i, d, g, l, s) = P(i) \cdot P(d) \cdot P(g \mid i, d) \cdot P(s \mid i) \cdot P(l \mid g)$
- Consider e.g. variable S; The analysis for other variables is analogous
- We have that $(S \perp\!\!\!\perp D, G, L \mid I)$ belongs to local independencies $\mathcal{I}_{local}(G)$
- The task is to prove that $(S \perp\!\!\!\perp D, G, L \mid I)_P$, i.e. that $P(s \mid i, d, g, l) = P(s \mid i)$

Independencies in Bayesian Networks

Factorization to I-map

Example: Illustration of the theorem

- Assume the DAG representation:



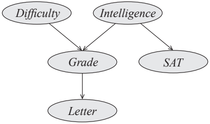
  that encodes the factorization: $P(i, d, g, l, s) = P(i) \cdot P(d) \cdot P(g \mid i, d) \cdot P(s \mid i) \cdot P(l \mid g)$
- Consider e.g. variable S; The analysis for other variables is analogous
- We have that $(S \perp\!\!\!\perp D, G, L \mid I)$ belongs to local independencies $\mathcal{I}_{local}(G)$
- The task is to prove that $(S \perp\!\!\!\perp D, G, L \mid I)_P$, i.e. that $P(s \mid i, d, g, l) = P(s \mid i)$
- By definition: $P(s \mid i, d, g, l) = \frac{P(s,i,d,g,l)}{P(i,d,g,l)}$

Independencies in Bayesian Networks

Factorization to I-map

Example: Illustration of the theorem

- Assume the DAG representation:



that encodes the factorization: $P(i, d, g, l, s) = P(i) \cdot P(d) \cdot P(g \mid i, d) \cdot P(s \mid i) \cdot P(l \mid g)$

- Consider e.g. variable S; The analysis for other variables is analogous
- We have that $(S \perp\!\!\!\perp D, G, L \mid I)$ belongs to local independencies $\mathcal{I}_{local}(G)$
- The task is to prove that $(S \perp\!\!\!\perp D, G, L \mid I)_P$, i.e. that $P(s \mid i, d, g, l) = P(s \mid i)$
- By definition: $P(s \mid i, d, g, l) = \frac{P(s,i,d,g,l)}{P(i,d,g,l)}$
- From the marginalizing over a joint distribution and factorization of P we get

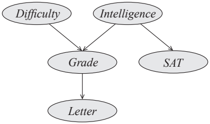
$$P(i, d, g, l) = P(i) \cdot P(d) \cdot P(g \mid i, d) \cdot P(l \mid g) \cdot \sum_{s} P(s \mid i) = P(i) \cdot P(d) \cdot P(g \mid i, d) \cdot P(l \mid g)$$

Independencies in Bayesian Networks

Factorization to I-map

Example: Illustration of the theorem

- Assume the DAG representation:



that encodes the factorization: $P(i,d,g,l,s) = P(i) \cdot P(d) \cdot P(g \mid i,d) \cdot P(s \mid i) \cdot P(l \mid g)$

- Consider e.g. variable S; The analysis for other variables is analogous
- We have that $(S \perp\!\!\!\perp D,G,L \mid I)$ belongs to local independencies $\mathcal{I}_{local}(G)$
- The task is to prove that $(S \perp\!\!\!\perp D,G,L \mid I)_P$, i.e. that $P(s \mid i,d,g,l) = P(s \mid i)$
- By definition: $P(s \mid i,d,g,l) = \frac{P(s,i,d,g,l)}{P(i,d,g,l)}$
- From the marginalizing over a joint distribution and factorization of P we get

$$P(i,d,g,l) = P(i) \cdot P(d) \cdot P(g \mid i,d) \cdot P(l \mid g) \cdot \sum_s P(s \mid i) = P(i) \cdot P(d) \cdot P(g \mid i,d) \cdot P(l \mid g)$$

- Then we can conclude

$$P(s \mid i,d,g,l) = \frac{P(s,i,d,g,l)}{P(i,d,g,l)} = \frac{P(i) \cdot P(d) \cdot P(g \mid i,d) \cdot P(s \mid i) \cdot P(l \mid g)}{P(i) \cdot P(d) \cdot P(g \mid i,d) \cdot P(l \mid g)} = P(s \mid i)$$

Independencies $\mathcal{I}(G)$

- As we have seen a graph structure G encodes a certain set of conditional independence assumptions:

$$\mathcal{I}_{local}(G) \;=\; \{(X_i \perp\!\!\!\perp \mathbf{V} \setminus (De(X_i) \cup Pa(X_i)) \mid Pa(X_i)) \;:\; \forall X_i \in \mathbf{V}\}$$

- Question:
  - which independencies $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})$ hold in a distribution associated with a BN with the structure G or, equivalently,
  - which independencies follow from $\mathcal{I}_{local}(G)$?
- We will denote all CIs which follows from $\mathcal{I}_{local}(G)$ as $\mathcal{I}(G)$

Independencies $\mathcal{I}(G)$

We analyse the problem as follows

- W start with the case of single variables X and Y in G
- Assume X and Y are not directly connected in G, but they are connected via Z as

$$X \sim Z \sim Y$$

- When "influence" can flow from X to Y via Z, we say that the path $X \sim Z \sim Y$ is active
- By case analysis for active two-edge paths we get

  Causal path $X \to Z \to Y$: active iff Z is not observed

  Evidential path $X \leftarrow Z \leftarrow Y$: active iff Z is not observed

  Common cause $X \leftarrow Z \to Y$: active iff Z is not observed

  Common effect $X \to Z \leftarrow Y$: active iff either Z or one of Z's descendants is observed

Definition

A structure $X \to Z \leftarrow Y$, where X and Y are not directly connected is called v-structure

Independencies $\mathcal{I}(G)$

We can generalize this analysis to longer paths $X_1 \sim X_2 \sim \ldots \sim X_n$ in G

- Let G be a BN structure, and $X_1 \sim X_2 \sim \ldots \sim X_n$ be a path in G.
- Let $\mathbf{Z}$ be a subset of observed variables
- The path $X_1 \sim X_2 \sim \ldots \sim X_n$ is active given $\mathbf{Z}$ if
  - Whenever we have a v-structure $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, then X_i or one of its descendants are in $\mathbf{Z}$
  - no other node along the path is in $\mathbf{Z}$

Independencies $\mathcal{I}(G)$

We can generalize this analysis to longer paths $X_1 \sim X_2 \sim \ldots \sim X_n$ in G

- Let G be a BN structure, and $X_1 \sim X_2 \sim \ldots \sim X_n$ be a path in G.
- Let $\mathbf{Z}$ be a subset of observed variables
- The path $X_1 \sim X_2 \sim \ldots \sim X_n$ is active given $\mathbf{Z}$ if
  - Whenever we have a v-structure $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, then X_i or one of its descendants are in $\mathbf{Z}$
  - no other node along the path is in $\mathbf{Z}$
- Putting these together, we get justification for the notion of *d*-separation and the following definition

$$\mathcal{I}(G) = \{(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) : \text{ for all } \mathbf{X}, \mathbf{Y}\ , \mathbf{Z} \text{ sets of nodes in } G \text{ with } (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})_G\}$$

Independencies $\mathcal{I}(G)$

We can generalize this analysis to longer paths $X_1 \sim X_2 \sim \ldots \sim X_n$ in G

- Let G be a BN structure, and $X_1 \sim X_2 \sim \ldots \sim X_n$ be a path in G.
- Let $\mathbf{Z}$ be a subset of observed variables
- The path $X_1 \sim X_2 \sim \ldots \sim X_n$ is active given $\mathbf{Z}$ if
  - Whenever we have a v-structure $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, then X_i or one of its descendants are in $\mathbf{Z}$
  - no other node along the path is in $\mathbf{Z}$
- Putting these together, we get justification for the notion of d-separation and the following definition

$$\mathcal{I}(G) = \{(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}) : \text{ for all } \mathbf{X}, \mathbf{Y}, \mathbf{Z} \text{ sets of nodes in } G \text{ with } (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})_G\}$$

- The important result is that (local) basis set of d-separation statements

$$\{(X_i \perp\!\!\!\perp \mathbf{V} \setminus (De(X_i) \cup Pa(X_i)) \mid Pa(X_i))_G \ : \ \forall X_i \in \mathbf{V}\}$$

  entails all other statements $\mathcal{I}(G)$ when combining them using the axioms of conditional independence

Independencies $\mathcal{I}(G)$

Markov equivalence

- Let us consider the sets $\mathcal{I}(G)$ of the following four DAGs

$$G_1: X \rightarrow Z \rightarrow Y \quad G_2: X \leftarrow Z \leftarrow Y \quad G_3: X \leftarrow Z \rightarrow Y \quad \text{and} \quad G_4: X \rightarrow Z \leftarrow Y$$

- Interestingly, we get that

$$\mathcal{I}(G_1) = \mathcal{I}(G_2) = \mathcal{I}(G_3) = \{(X \perp\!\!\!\perp Y \mid Z)\}$$

and

$$\mathcal{I}(G_4) = \{(X \perp\!\!\!\perp Y)\}$$

- Thus, G_1, G_2, G_3 encode the same CIs, while G_4 not
- This leads to the following

Definition Two DAGs G and G' over $\mathbf{V}$ are Markov equivalent (called also I-equivalent) if $\mathcal{I}(G) = \mathcal{I}(G')$

Independencies $\mathcal{I}(G)$

Markov equivalence

- Question: how can we verify that two DAGs G and G' are Markov equivalent?
- The skeleton of a graph G over $\mathbf{V}$ is an undirected graph over $\mathbf{V}$ that contains an edge $X - Y$ for every edge $X \to Y$ or $X \leftarrow Y$ in G

Theorem (Verma, Pearl)

Let G and G' be two DAGs over $\mathbf{V}$. The graphs are Markov equivalent if and only if G and G' have the same skeleton and the same set of v-structures.

Independencies $\mathcal{I}(G)$

Markov equivalence

- Question: how can we verify that two DAGs G and G' are Markov equivalent?
- The skeleton of a graph G over $\mathbf{V}$ is an undirected graph over $\mathbf{V}$ that contains an edge $X - Y$ for every edge $X \rightarrow Y$ or $X \leftarrow Y$ in G

Theorem (Verma, Pearl)

Let G and G' be two DAGs over $\mathbf{V}$. The graphs are Markov equivalent if and only if G and G' have the same skeleton and the same set of v-structures.

- For example

  $G_1 : X \rightarrow Z \rightarrow Y \quad G_2 : X \leftarrow Z \leftarrow Y \quad G_3 : X \leftarrow Z \rightarrow Y$ and $G_4 : X \rightarrow Z \leftarrow Y$

  all graphs have he same skeleton $X - Z - Y$
- The set of v-structures for G_1, G_2, G_3 is empty, thus they are Markov equvalent
- G_4 has a v-structure $X \rightarrow Z \leftarrow Y$, thus it is not Markov equvalent with G_i, $i = 1, 2, 3$

Independencies $\mathcal{I}(G)$

Markov equivalence

Fact

The set of all DAGs over $\mathbf{V}$ is partitioned into a set of mutually exclusive and exhaustive Markov equivalent classes, which are the set of equivalence classes induced by the Markov equivalence relation

Independencies $\mathcal{I}(G)$

Markov equivalence

Fact

The set of all DAGs over $\mathbf{V}$ is partitioned into a set of mutually exclusive and exhaustive Markov equivalent classes, which are the set of equivalence classes induced by the Markov equivalence relation

For example, for $\mathbf{V} = \{X, Y, Z\}$ the equivalence classes induced by the Markov equivalence are the following

- $\{G = (\{X, Y, Z\}, \mathbf{E}) : \ |\mathbf{E}| = 3 \text{ and } G \text{ is no cycle}\}$
- For DAGs with $|\mathbf{E}| = 2$ we show only the case, when X and Y are not incident
  - $X \to Z \to Y, \quad X \leftarrow Z \leftarrow Y, \quad X \leftarrow Z \to Y$
  - $X \to Z \leftarrow Y$
- DAGs with $|\mathbf{E}| = 1$
  - $X \to Z \quad Y, \quad X \leftarrow Z \quad Y$
  - $X \to Y \quad Z, \quad X \leftarrow Y \quad Z$
  - $Y \to Z \quad X, \quad Y \leftarrow Z \quad X$
- DAGs with $|\mathbf{E}| = 0$
  - $X \quad Z \quad Y$

Literatur

- D. Koller and N. Friedman (2009), Ch.3
- J. Pearl (2009), Ch.1
- J. Pearl, M. Glymour, and N.P. Jewell (2016), Ch. 1,2