Reasoning Under Uncertainty: Independence and Inference

CPSC 322 – Uncertainty 5

Textbook §6.3.1 (and 6.5.2 for HMMs)

March 25, 2011

Lecture Overview

- Recap: Bayesian Networks and Markov Chains

- Inference in a Special Type of Bayesian Network
  - Hidden Markov Models (HMMs)
  - Rainbow Robot example

- Inference in General Bayesian Networks
  - Observations and Inference
  - Time-permitting: Entailed independencies
  - Next lecture: Variable Elimination

Recap: Conditional Independence

Definition (Conditional independence)
Random variable X is (conditionally) independent of random variable Y given random variable Z, written X ⫫ Y | Z if, for all x ∈ dom(X), y_j ∈ dom(Y), y_k ∈ dom(Y) and z ∈ dom(Z) the following equation holds:

$$P(X = x | Y = yj, Z = z)$$
$$= P(X = x | Y = yk, Z = z)$$
$$= P(X = x | Z = z)$$

- Definition of X ⫫ Y | Z in distribution form: $P(X|Y, Z) = P(X|Z)$

Recap: Bayesian Networks, Definition

Definition (Bayesian Network)
A Bayesian network consists of
- A directed acyclic graph (V, E) whose nodes are labeled with random variables
- A domain for each random variable
- A conditional probability distribution for each variable X
  - Specifies $P(X|Parents(X))$
  - $Parents(X)$ is the set of variables X' with $(X', \mathrm{X}) \in E$
    - For nodes X without predecessors, $Parents(X) = \{\ \}$

- Chain rule: $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|X_1,\ldots,X_{i-1})$
- Bayesian Network semantics:
  - A variable is conditionally independent of its non-descendants given its parents
  - $X_i \perp\!\!\!\perp \{X_1,\ldots,X_{i-1}\} \setminus Pa(X_i) \mid Pa(V)$
  - I.e., $P(X_i|X_1,\ldots,X_{i-1}) = P(X_i|pa(X_i))$

Recap: Construction of Bayesian Networks

Encoding the joint over $\{X_1, \ldots, X_n\}$ as a Bayesian network:

- Totally order the variables: e.g., $X_1, \ldots, X_n$
- For every variable X_i, find the smallest set of parents $Pa(X_i) \subseteq \{X_1, \ldots, X_{i-1}\}$ such that $X_i \perp\!\!\!\perp \{X_1, \ldots, X_{i-1}\} \setminus Pa(X_i) \mid Pa(X_i)$
  - X_i is conditionally independent from its other ancestors given its parents
- For every variable X_i, construct its conditional probability table
  - $P(X_i \mid Pa(X_i))$
  - This has to specify a conditional probability distribution $P(X_i \mid Pa(X_i) = pa(X_i))$ for every instantiation $pa(X_i)$ of X_i's parents
  - If a variable has 3 parents each of which has a domain with 4 values, how many instantiations of its parents are there?

3^4 4^3 $3*4$ 4^3-1

Recap: Construction of Bayesian Networks

Encoding the joint over $\{X_1, \ldots, X_n\}$ as a Bayesian network:

- Totally order the variables: e.g., $X_1, \ldots, X_n$
- For every variable X_i, find the smallest set of parents $Pa(X_i) \subseteq \{X_1, \ldots, X_{i-1}\}$ such that $X_i \perp\!\!\!\perp \{X_1, \ldots, X_{i-1}\} \setminus Pa(X_i) \mid Pa(X_i)$
  - X_i is conditionally independent from its other ancestors given its parents
- For every variable X_i, construct its conditional probability table
  - $P(X_i \mid Pa(X_i))$
  - This has to specify a conditional probability distribution $P(X_i \mid Pa(X_i) = pa(X_i))$ for every instantiation $pa(X_i)$ of X_i's parents
  - If a variable has 3 parents each of which has a domain with 4 values, how many instantiations of its parents are there?
    - $4 * 4 * 4 = 4^3$
    - For each of these 4^3 values we need one probability distribution defined over the values of X_i

Recap of BN construction with a small example

- Two Boolean variables: Disease and Symptom

1. The causal ordering: Disease, Symptom
2. Chain rule:
   P(Disease, Symptom) = P(Disease) × P(Symptom |Disease)
3. Is Disease ⫫ Symptom | {} ?
   - I.e., are they marginally independent (conditioned on nothing)?

   Yes No

Disease *D*	Symptom S	*P(D,S)*
t	t	0.0099
t	f	0.0001
f	t	0.0990
f	f	0.8910

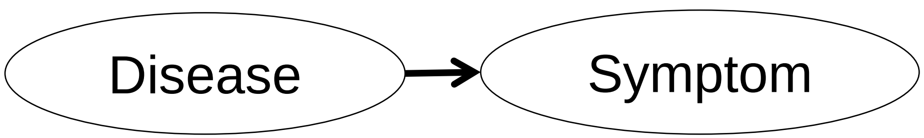
Disease *D*	*P(D)*
t	0.01
f	0.99

Symptom S	*P(S)*
t	0.1089
f	0.8911

Recap of BN construction with a small example

- Two Boolean variables: Disease and Symptom

1. The causal ordering: Disease, Symptom
2. Chain rule:
   P(Disease, Symptom) = P(Disease) $\times$ P(Symptom |Disease)
3. Is Disease $\perp\!\!\!\perp$ Symptom | {} ?
   - I.e., are they marginally independent (conditioned on nothing)?
   - No! That would mean P(D,S) = P(D) $\times$ P(S), which is not true
   - We have to put an edge from the parent (Disease) to the child (Symptom)



Disease *D*	Symptom S	*P(D,S)*
t	t	0.0099
t	f	0.0001
f	t	0.0990
f	f	0.8910

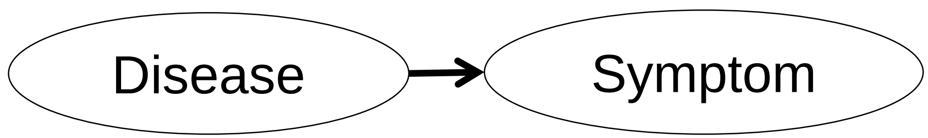
Disease *D*	*P(D)*
t	0.01
f	0.99

Symptom S	*P(S)*
t	0.1089
f	0.8911

Recap of BN construction with a small example

- Which conditional probability tables do we need?

P(D) P(D|S) P(S|D) P(D,S)



Disease *D*	Symptom S	*P(D,S)*
t	t	0.0099
t	f	0.0001
f	t	0.0990
f	f	0.8910

Disease *D*	*P(D)*
t	0.01
f	0.99

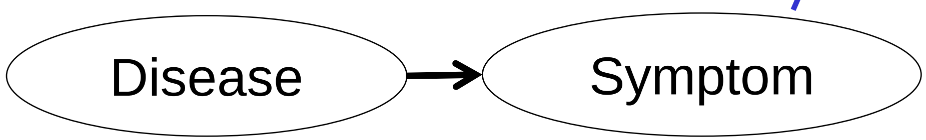
Symptom S	*P(S)*
t	0.1089
f	0.8911

Recap of BN construction with a small example

- Which conditional probability tables do we need?
  - P(D) and P(S|D)
  - In general: for each variable X in the network: P(X|Pa(X))

P(D=t)
0.01

Disease *D*	*P(S=t|D)*
t	0.0099/(0.0099+0.0001)=0.99
f	0.099/(0.099+0.891)=0.1



Disease *D*	Symptom S	*P(D,S)*
t	t	0.0099
t	f	0.0001
f	t	0.0990
f	f	0.8910

Disease *D*	*P(D)*
t	0.01
f	0.99

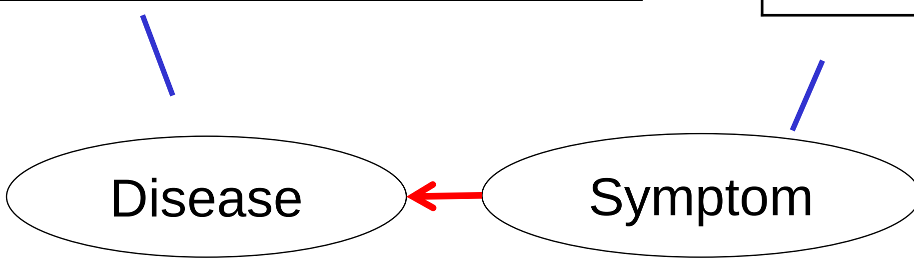
Symptom S	*P(S)*
t	0.1089
f	0.8911

Recap of BN construction with a small example

- How about a different ordering? Symptom, Disease
  - We need distributions P(S) and P(D|S)
  - In general: for each variable X in the network: P(X|Pa(X))

Symptom S	*P(D=t|S)*
t	0.0099/(0.0099+0.099)=0.00909090
f	0.0001/(0.0001+0.891)=0.00011122

P(S=t)
0.1089



Disease *D*	Symptom S	*P(D,S)*
t	t	0.0099
t	f	0.0001
f	t	0.0990
f	f	0.8910

Disease *D*	*P(D)*
t	0.01
f	0.99

Symptom S	*P(S)*
t	0.1089
f	0.8911

Remark: where do the conditional probabilities come from?

- The joint distribution is not normally the starting point
  - We would have to define exponentially many numbers

- First define the Bayesian network structure
  - Either by domain knowledge
  - Or by machine learning algorithms (see CPSC 540)
    - Typically based on local search

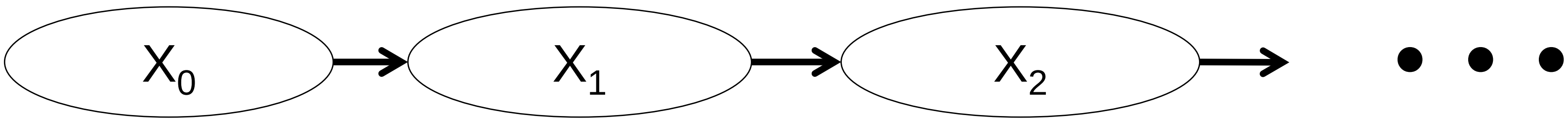
- Then fill in the conditional probability tables
  - Either by domain knowledge
  - Or by machine learning algorithms (see CPSC 340, CPSC 422)
    - Based on statistics over the observed data

Lecture Overview

- Recap: Bayesian Networks and Markov Chains
- Inference in a Special Type of Bayesian Network
  - Hidden Markov Models (HMMs)
  - Rainbow Robot example
- Inference in General Bayesian Networks
  - Observations and Inference
  - Time-permitting: Entailed independencies
  - Next lecture: Variable Elimination

Markov Chains

- A Markov chain is a special kind of Bayesian network:



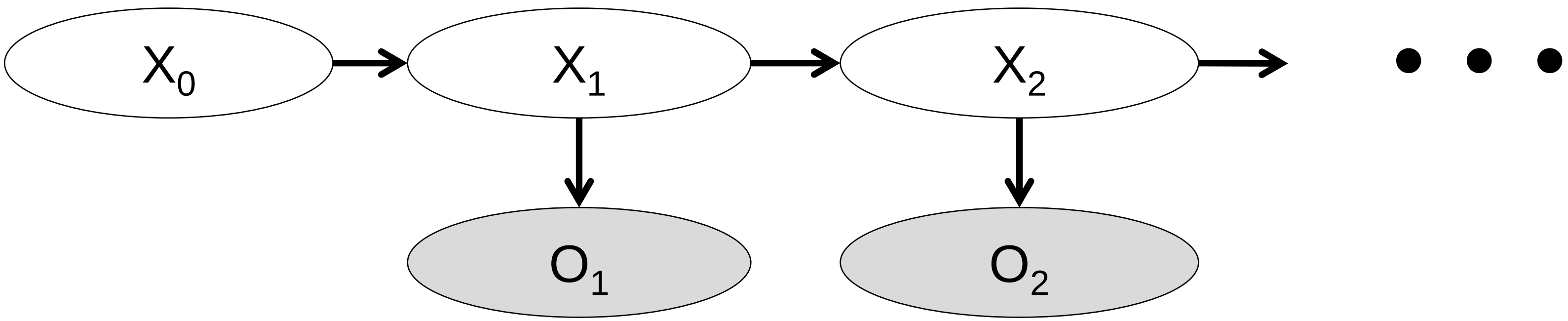
  - Intuitively X_t conveys all of the information about the history that can affect the future states:
    "The past is independent of the future given the present."

- JPD of a Markov Chain: $P(X_0,\ldots,X_T) = P(X_0) \times \prod_{t=1}^{T} P(X_t|X_{t-1})$

- A Markov chain is stationary iff
  - All state transition probability tables are the same
  - I.e., for all t > 0, t' > 0: $P(X_t|X_{t-1}) = P(X_{t'}|X_{t'-1})$
    - Thus, we only need to specify $P(X_0)$ and $P(X_t|X_{t-1})$

Hidden Markov Models (HMMs)

- A Hidden Markov Model (HMM) is a stationary Markov chain plus a noisy observation about the state at each time step:

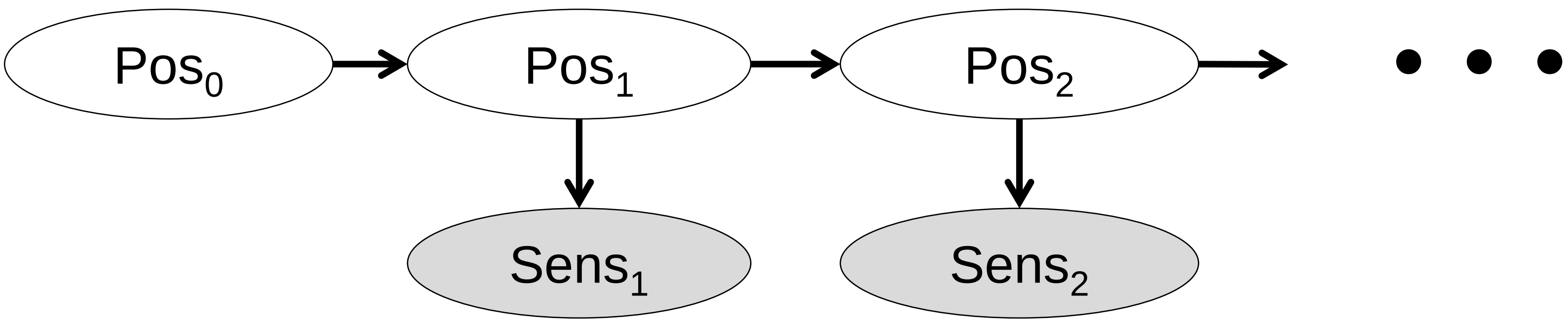


- Same conditional probability tables at each time step
  - The state transition probability $P(X_t|X_{t-1})$
    - also called the system dynamics
  - The observation probability $P(O_t|X_t)$
    - also called the sensor model

- JPD of an HMM: $P(X_0, \ldots, X_T, O_1, \ldots, O_T)$
$= P(X_0) \times \prod_{t=1}^{T} P(X_t|X_{t-1}) \times \prod_{t=1}^{T} P(O_t|X_t)$

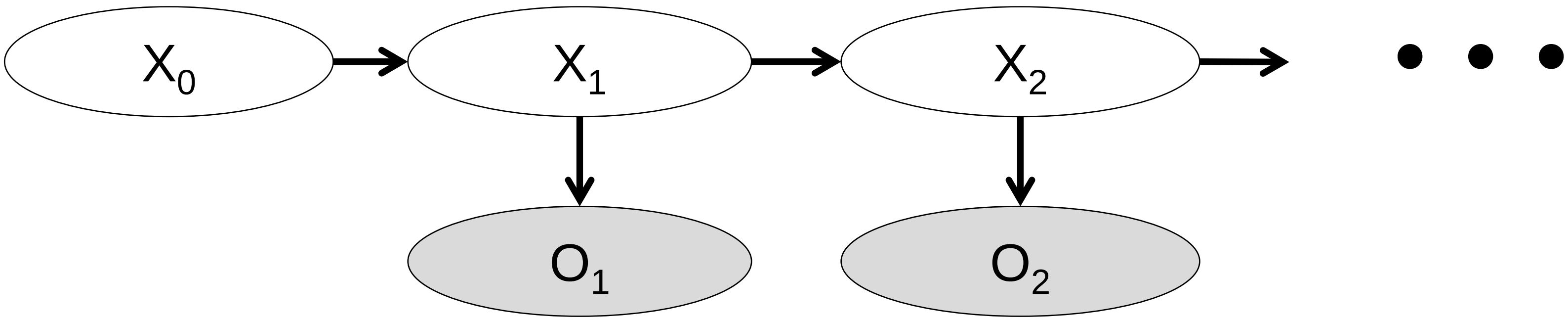
Example HMM: Robot Tracking

- Robot tracking as an HMM:



- Robot is moving at random: $P(Pos_t|Pos_{t-1})$
- Sensor observations of the current state $P(Sens_t|Pos_t)$

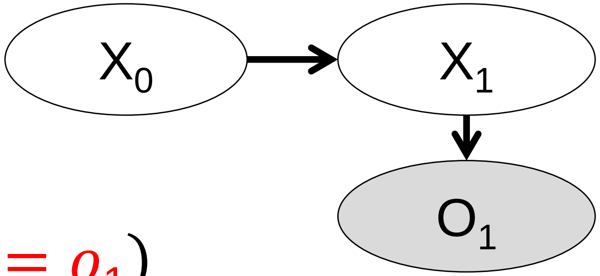
Filtering in Hidden Markov Models (HMMs)



- Filtering problem in HMMs:
  at time step t, we would like to know $P(X_t|o_1, \ldots, o_t)$

- We will derive simple update equations for this belief state:
  - We are given $P(X_0)$ (i.e., $P(X_0 \mid \{\})$)
  - We can compute $P(X_t|O_1, \ldots, O_t)$ if we know $P(X_{t-1}|o_1, \ldots, o_{t-1})$
  - A simple example of dynamic programming

HMM Filtering: first time step



$$P(X_1|O_1 = o_1)$$

By applying marginalization over X_0 "backwards":

$$= \sum_{x \in dom(X_0)} P(X_1, X_0 = x | O_1 = o_1)$$

Direct application of Bayes rule

$$= \sum_{x \in dom(X_0)} \frac{P(O_1 = o_1 | X_1, X_0 = x) \times P(X_1, X_0 = x)}{P(O_1 = o_1)}$$

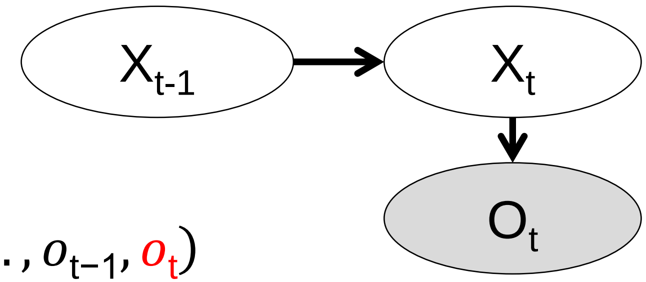
$O_1 \perp\!\!\!\perp X_0 \mid X_1$ and product rule

$$= \sum_{x \in dom(X_0)} \frac{P(O_1 = o_1 | X_1) \times P(X_1 | X_0 = x) \times P(X_0 = x)}{P(O_1 = o_1)}$$

Normalize to make the probability sum to 1. $P(O_1 = o_1)$ is just a number.

$$\propto \sum_{x \in dom(X_0)} P(O_1 = o_1 | X_1) \times P(X_1 | X_0 = x) \times P(X_0 = x)$$

HMM Filtering: general time step t



$$P(X_t|o_1,\dots,o_t)$$

By applying marginalization over X_{t-1} "backwards":

$$= \sum_{x\,\in dom(X_{t-1})} P(X_t, X_{t-1}=x|o_1,\dots,o_{t-1},o_t)$$

Direct application of Bayes rule

$$= \sum_{x\,\in dom(X_{t-1})} \frac{P(o_t|X_t,X_{t-1}=x,o_1,\dots,o_{t-1}) \times P(X_t,X_{t-1}=x|o_1,\dots,o_{t-1})}{P(o_t|o_1,\dots,o_{t-1})}$$

$O_t \perp\!\!\!\perp \{X_{t-1}, O_1,\dots,O_{t-1}\} \mid X_t$ and $X_t \perp\!\!\!\perp \{O_1,\dots,O_{t-1}\} \mid X_{t-1}$

$$= \sum_{x\,\in dom(X_{t-1})} \frac{P(o_t|X_t) \times P(X_t|X_{t-1}=x) \times P(X_{t-1}=x|o_1,\dots,o_{t-1})}{P(o_t|o_1,\dots,o_{t-1})}$$

Normalize to make the probability sum to 1.
$P(o_t|o_1,\dots,o_{t-1})$ is just a number.

$$\propto \sum_{x\,\in dom(X_{t-1})} P(o_t|X_t) \times P(X_t|X_{t-1}=x) \times P(X_{t-1}=x|o_1,\dots,o_{t-1})$$

HMM Filtering Summary

- Initialize belief state at time 0: $P(X_0)$
  - In Rainbow Robots, we initialize this for you: $P(Pos_t)$
  - Belief over where the other robot is: 6x6 matrix summing to one.

- At each time step, update belief state given new observation:

Observation probability

We already know this from the previous step

Transition probability

$$P(X_t = x_t | o_1, \ldots, o_t)$$

$$\propto \sum_{x \in dom(X_{t-1})} P(o_t | x_t) \times P(x_t | X_{t-1} = x) \times P(X_{t-1} = x | o_1, \ldots, o_{t-1})$$

- In Rainbow robots, how many probabilities do you have to store for the updated belief state $P(X_t | o_1, \ldots, o_t)$ at time t?

36t 36^t t^{36} 36

HMM Filtering Summary

- Initialize belief state at time 0: $P(X_0)$
  - In Rainbow Robots, we initialize this for you: $P(Pos_t)$
  - Belief over where the other robot is: 6x6 matrix summing to one.

- At each time step, update belief state given new observation:

Observation probability

Transition probability

We already know this from the previous step

$$P(X_t = x_t | o_1, \ldots, o_t)$$

$$\propto \sum_{x \in dom(X_{t-1})} P(o_t | x_t) \times P(x_t | X_{t-1} = x) \times P(X_{t-1} = x | o_1, \ldots, o_{t-1})$$

- In Rainbow robots, how many probabilities do you have to store for the updated belief state $P(X_t | o_1, \ldots, o_t)$ at time t?
  - It's just 36.
  - That's the beauty of HMMs: the belief state does not grow.
  - You simply update the previous belief state by applying the same equation every time, with a different observation o_t

HMM Filtering: Rainbow Robot Summary

- You will need to implement the belief state updates
  - Not hard if you understand HMM Filtering
  - About 40 lines of Java Code to compute the
    - transition probability $P(Pos_t = x_{\mathrm{t}} | Pos_{\mathrm{t-1}} = x)$ and the
    - observation probability $P(Sens_t = sens_{\mathrm{t}} | Pos_{\mathrm{t}} = x_{\mathrm{t}})$
- Then it's just the equation from the previous slide
  - You have the previous belief state $P(Pos_{\mathrm{t-1}} = x | sens_1, \ldots, \mathrm{sens}_{\mathrm{t-1}})$
    - (6x6 matrix summing to one)
  - Probability for entry x_{t} of the new belief state is computed as above:
    - Sum $P(Pos_t = x_{\mathrm{t}} | Pos_{\mathrm{t-1}} = x) \times P(Pos_{\mathrm{t-1}} = x | sens_1, \ldots, \mathrm{sens}_{\mathrm{t-1}})$ over all values for x
    - Multiply with the observation probability $P(Sens_t = sens_{\mathrm{t}} | Pos_{\mathrm{t}} = x_{\mathrm{t}})$
  - Normalize to make the new belief state sum to one

Lecture Overview

- Recap: Bayesian Networks and Markov Chains
- Inference in a Special Type of Bayesian Network
  - Hidden Markov Models (HMMs)
  - Rainbow Robot example
- Inference in General Bayesian Networks
  - Observations and Inference
  - Time-permitting: Entailed independencies
  - Next lecture: Variable Elimination

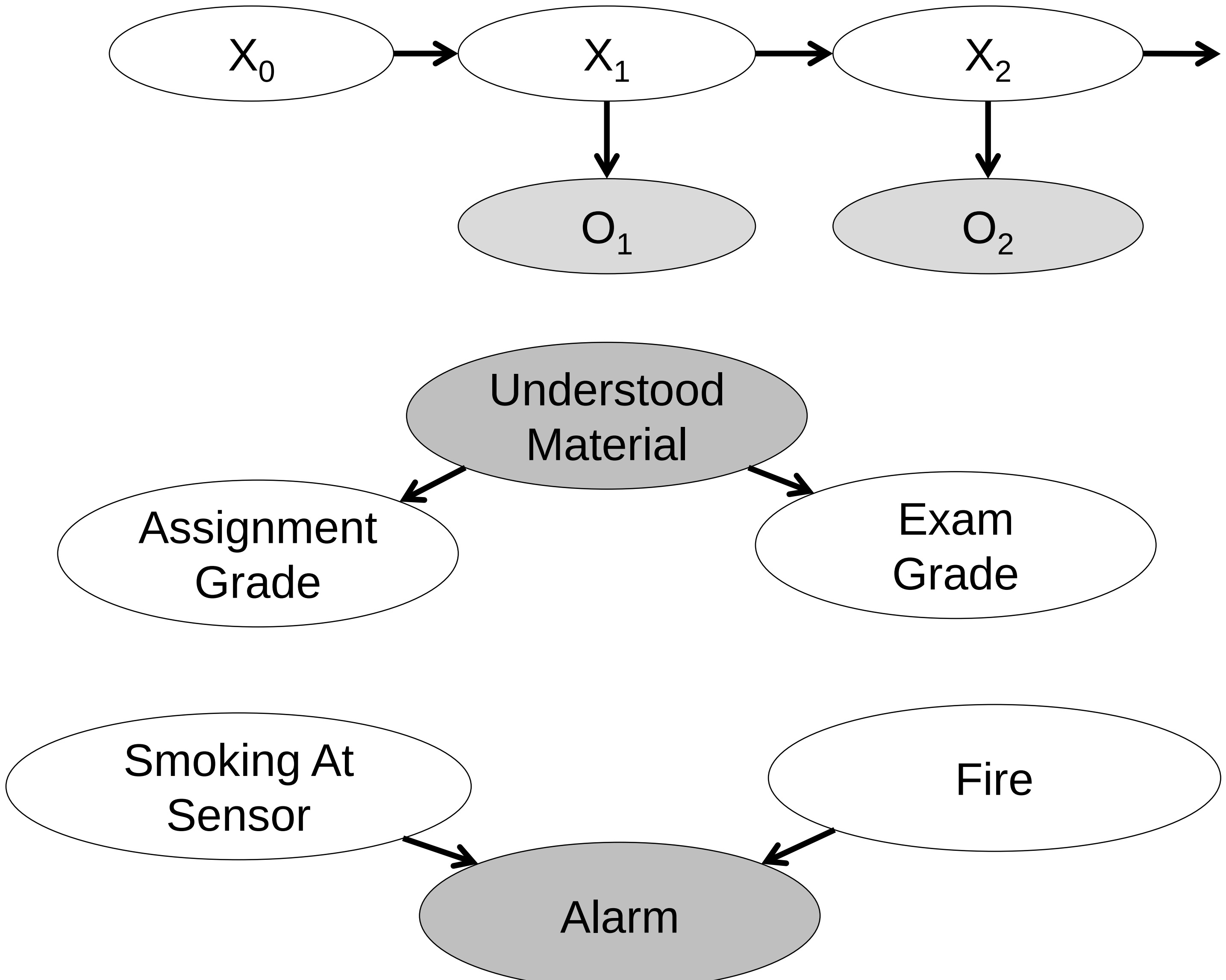
Bayesian Networks: Incorporating Observations

- In the special case of Hidden Markov Models (HMMs):
  - we could easily incorporate observations
  - and do efficient inference (in particular: filtering)

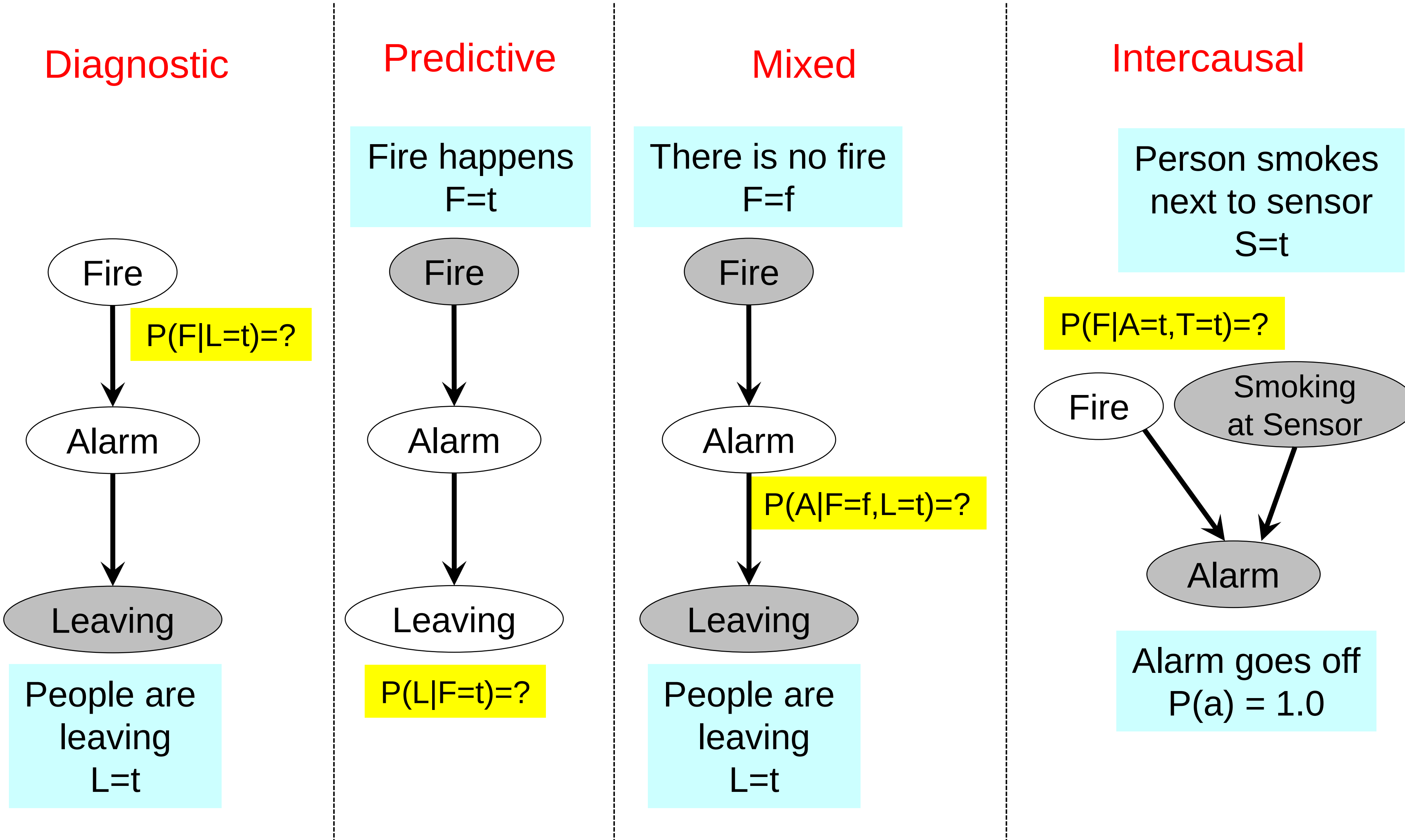
- Back to general Bayesian Networks
  - We can still incorporate observations
  - And we can still do (fairly) efficient inference

Bayesian Networks: Incorporating Observations

We denote observed variables as shaded. Examples:



Bayesian Networks: Types of Inference



We will use the same reasoning procedure for all of these types

Lecture Overview

- Recap: Bayesian Networks and Markov Chains
- Inference in a Special Type of Bayesian Network
  - Hidden Markov Models (HMMs)
  - Rainbow Robot example
- Inference in General Bayesian Networks
  - Observations and Inference
  - Time-permitting: Entailed independencies
  - Next lecture: Variable Elimination

Inference in Bayesian Networks

Given

- A Bayesian Network BN, and
- Observations of a subset of its variables E: E=e
- A subset of its variables Y that is queried

Compute the conditional probability P(Y|E=e)

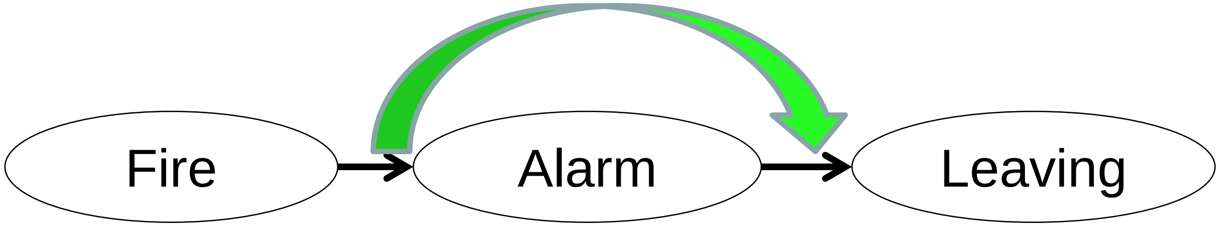
- Step 1: Drop all variables X of the Bayesian network that are conditionally independent of Y given E=e
  - By definition of $Y \perp\!\!\!\perp X \mid E=e$, we know P(Y|E=e) = P(Y|X=x,E=e)
  - We can thus drop X
- Step 2: run variable elimination algorithm (next lecture)

Information flow in Bayesian Networks

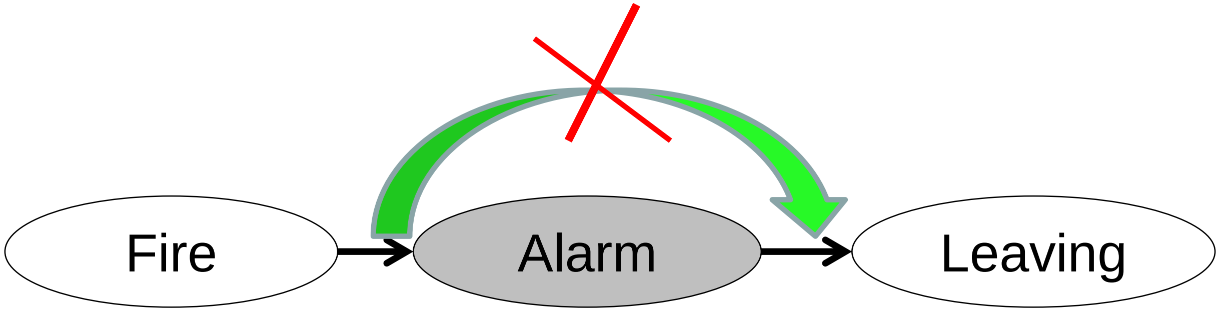
- A Bayesian network structure implies a number of conditional independencies and dependencies
  - We can determine these directly from the graph structure
    - I.e., we don't need to look at the conditional probability tables
    - Conditional independencies we derive for a graph structure will hold for all possible conditional probability tables
- Information we acquire about one variable flows through the network and changes our beliefs about other variables
  - This is also called probability propagation
  - But information does not flow everywhere: Some nodes block information

Information flow through chain structure

- Unobserved node in a chain lets information pass
  - E.g. learning the value of Fire will change your belief about Alarm
  - Which will in turn change your belief about Leaving

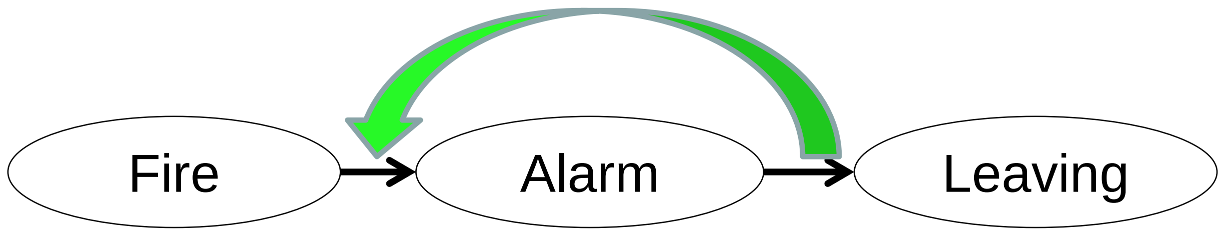


- Observed node in a chain blocks information
  - If you know the value of Alarm to start with:
    learning the value of Fire yields no extra information about Leaving

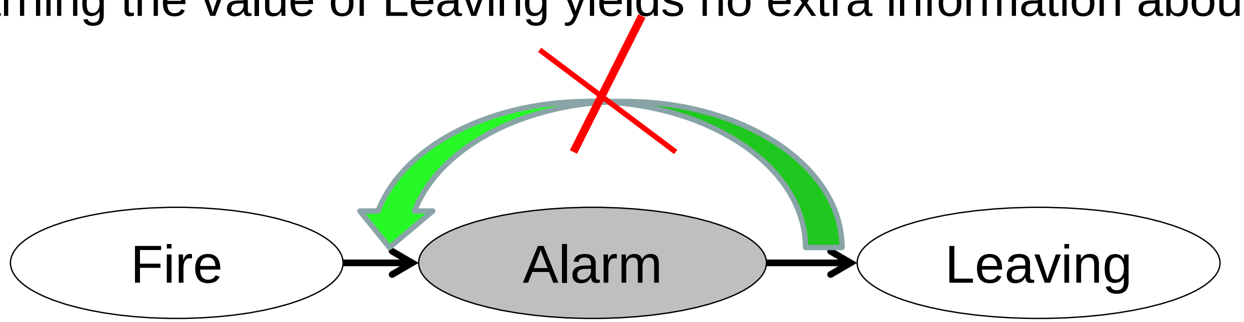


Information flow through chain structure

- Information flow is symmetric ($X \perp\!\!\!\perp Y \mid Z$ and $Y \perp\!\!\!\perp X \mid Z$ are identical)
  - Unobserved node in a chain lets information pass (both ways)
    - E.g. learning the value of Leaving will change your belief about Alarm
    - Which will in turn change your belief about Fire

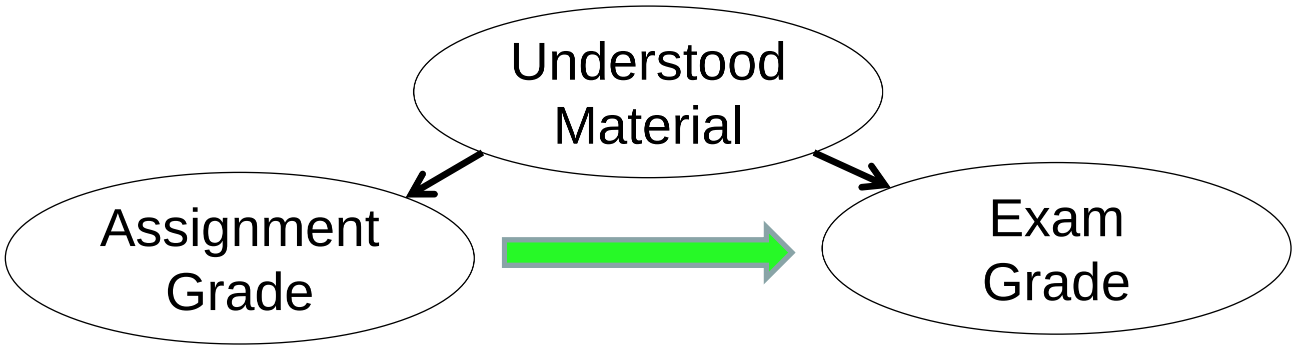


  - Observed node in a chain blocks information (both ways)
    - If you know the value of Alarm to start with:
      learning the value of Leaving yields no extra information about Fire

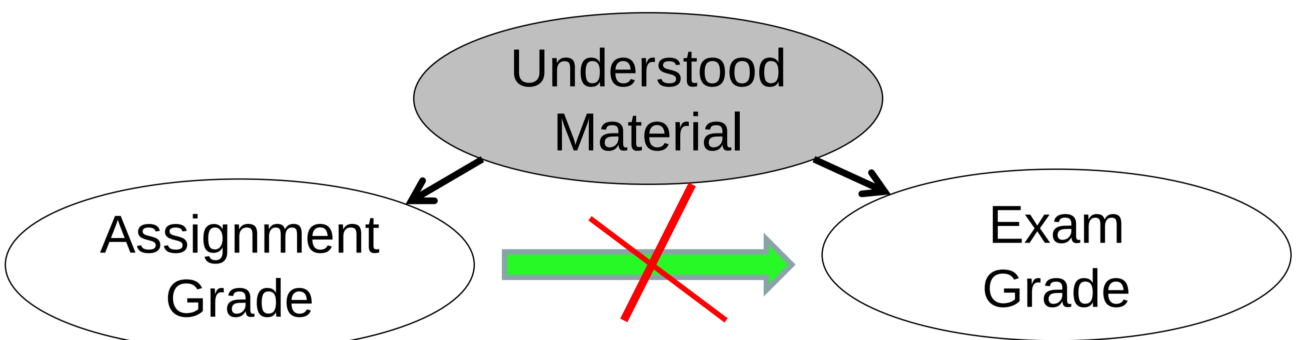


Information flow through common parent

- Unobserved common parent lets information pass
  - E.g. learning the value of AssignmentGrade changes your belief about UnderstoodMaterial
  - Which will in turn change your belief about ExamGrade

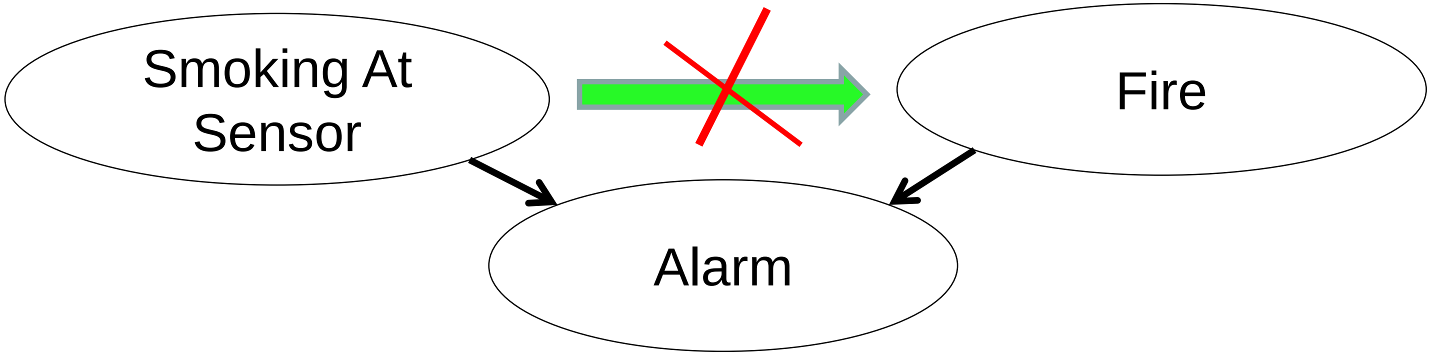


- Observed common parent blocks information
  - If you know the value of UnderstoodMaterial to start with
    - Learning AssignmentGrade yields no extra information about ExamGrade

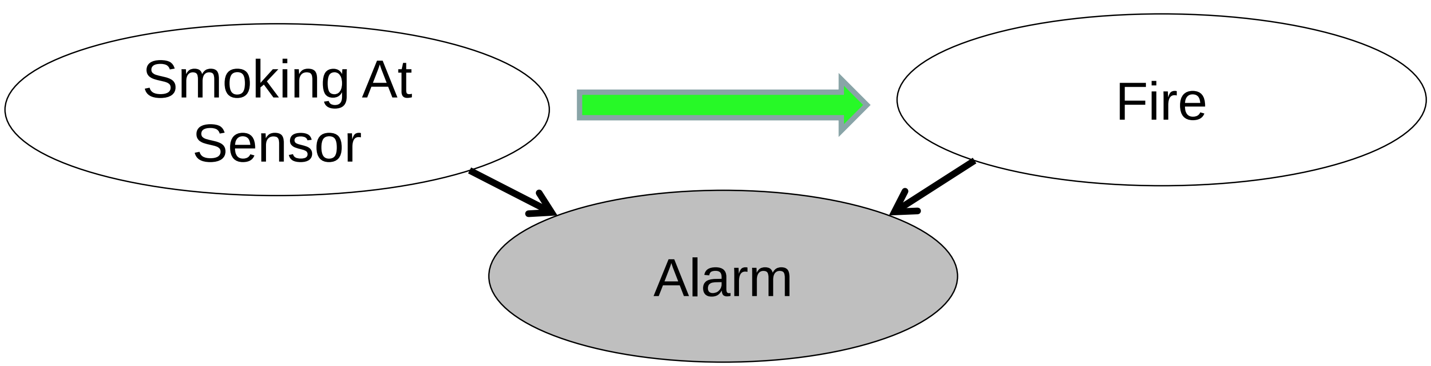


Information flow through common child

- Unobserved common child blocks information
  - E.g. learning the value of Fire will not change your belief about Smoking: the two are marginally independent

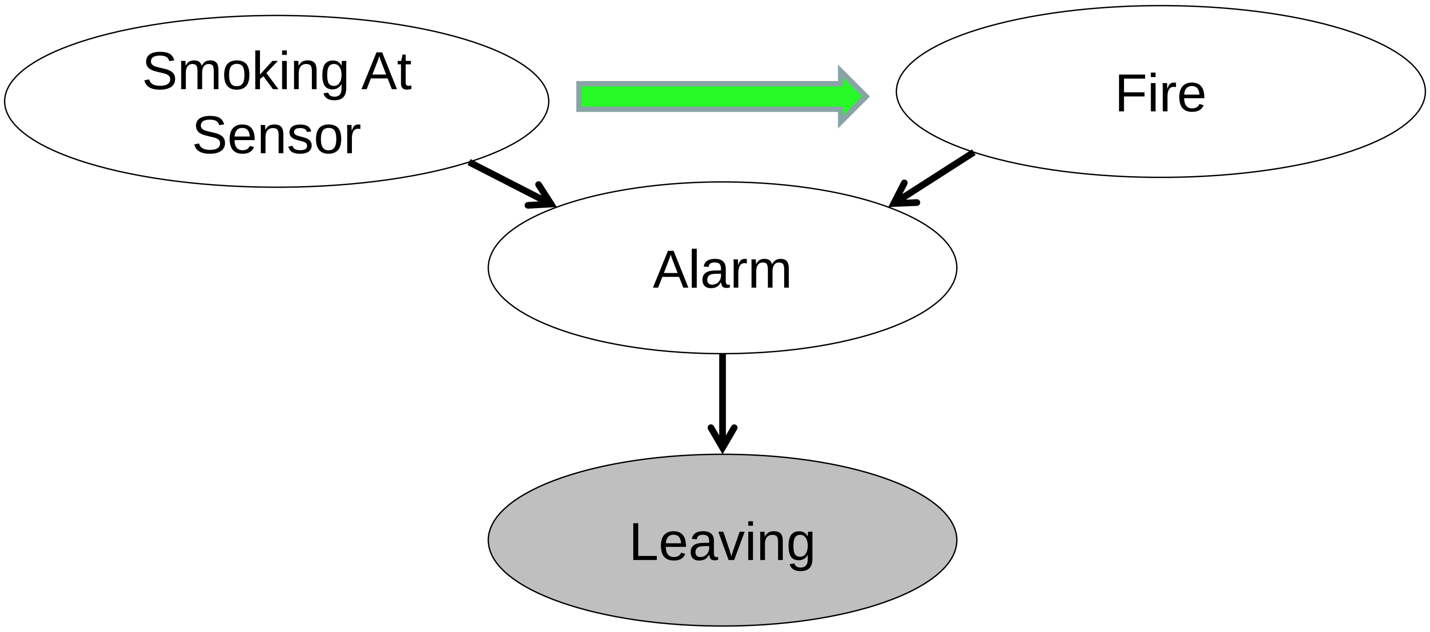


- Observed common child lets information pass: explaining away
  - E.g., when you know the alarm is going:
    - then learning that a person smoked next to the fire sensor "explains away the evidence" and thereby changes your beliefs about fire.



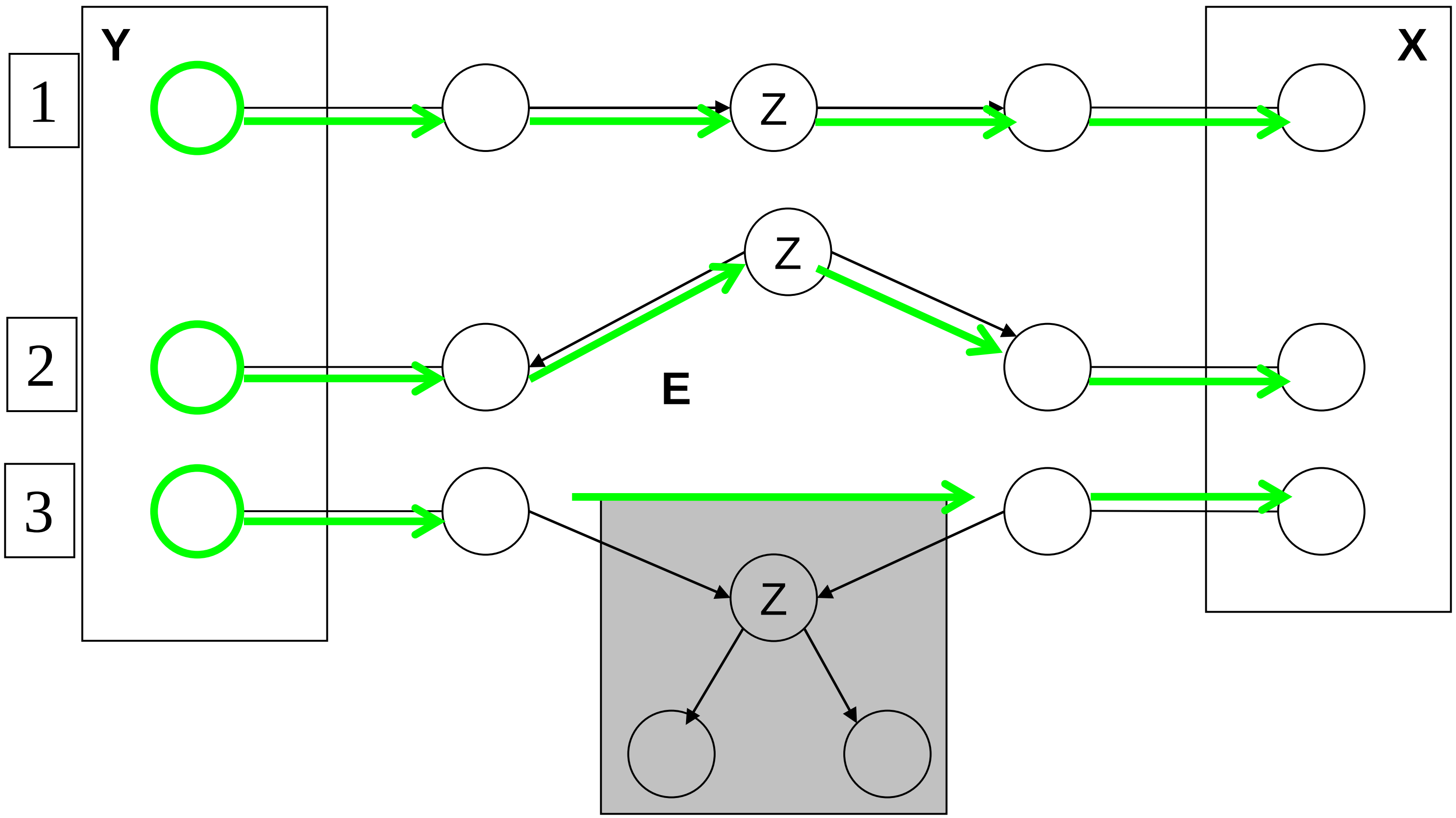
Information flow through common child

- Exception: unobserved common child lets information pass if one of its descendants is observed
  - This is just as if the child itself was observed
    - E.g., Leaving could be a deterministic function of Alarm, so observing Leaving means you know Alarm as well
    - Thus, a person smoking next to the fire alarm can still “explain away” the evidence of people leaving the building



Summary: (Conditional) Dependencies

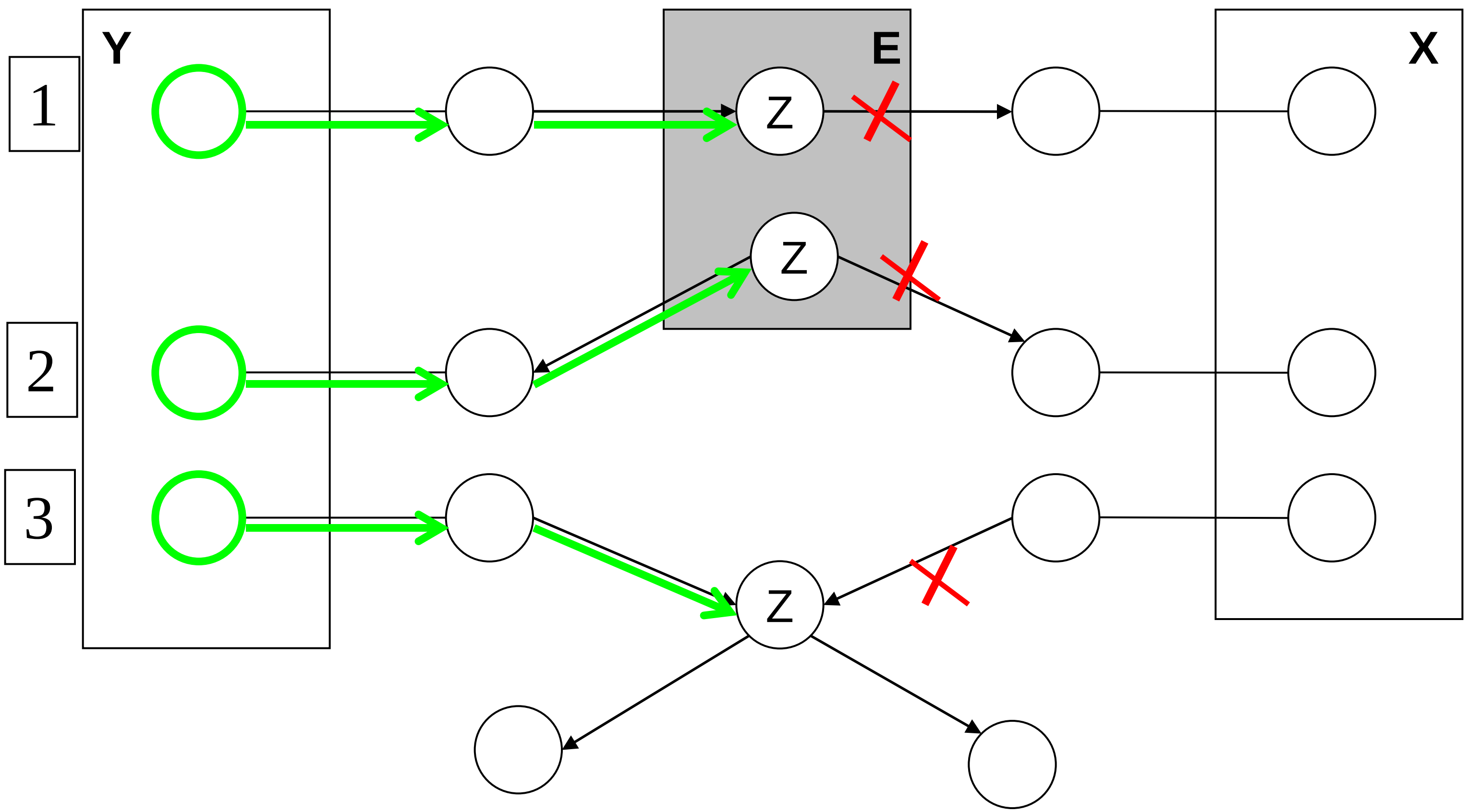
- In these cases, X and Y are (conditionally) dependent



- In 3, X and Y become dependent as soon as there is evidence on Z or on *any of its descendants.*

Summary: (Conditional) Independencies

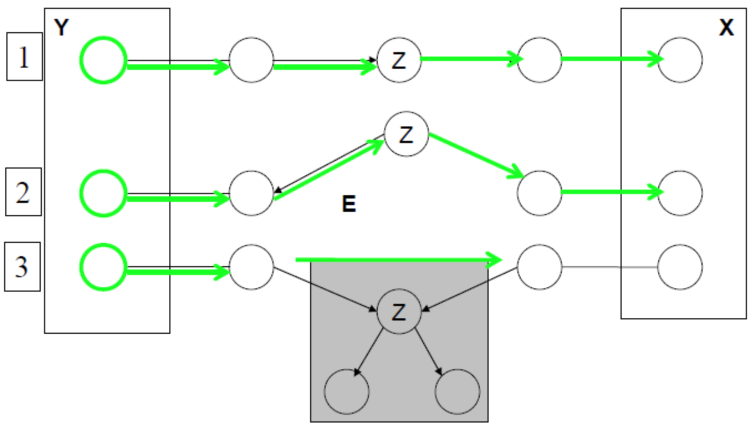
- Blocking paths for probability propagation. Three ways in which a path between Y to X (or vice versa) can be blocked, given evidence E

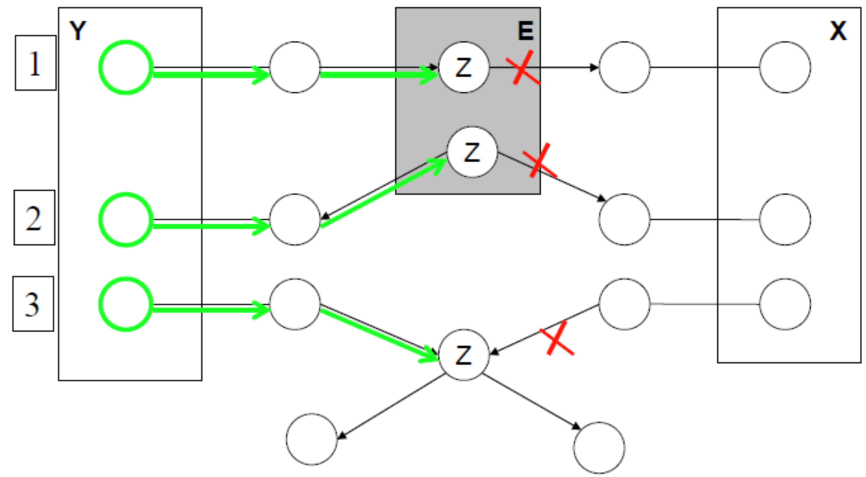


Training your understanding of conditional independencies in AIspace

- These concepts take practice to get used to
- Use the AIspace applet for Belief and Decision networks (http://aispace.org/bayes/)
  - Load the “conditional independence quiz” network (or any other one)
  - Go in “Solve” mode and select “Independence Quiz”
- You can take an unbounded number of quizzes:
  - It generates questions, you answer, and then get the right answer
  - It also allows you to ask arbitrary queries

Conditional Independencies in a BN



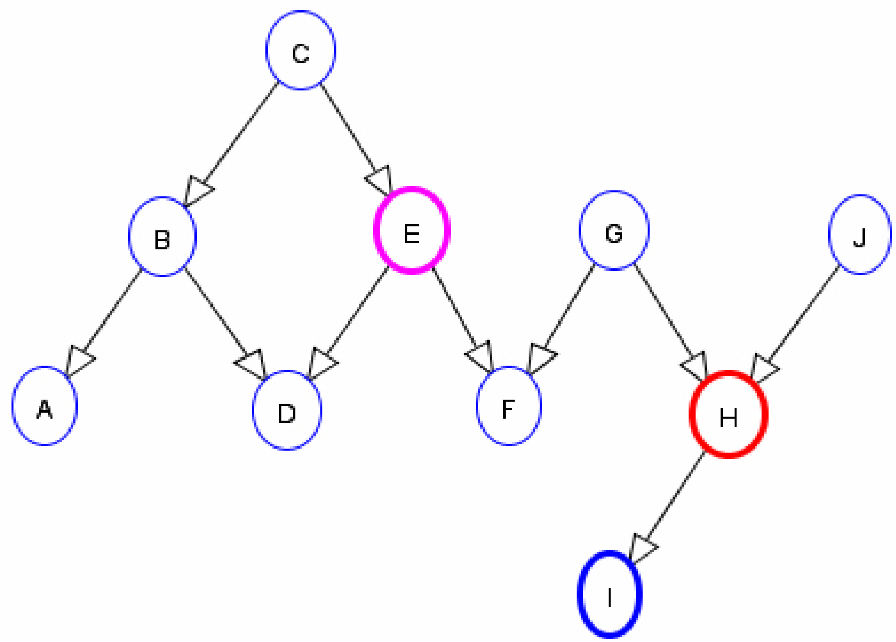


Is **H** conditionally independent of **E** given **I**?

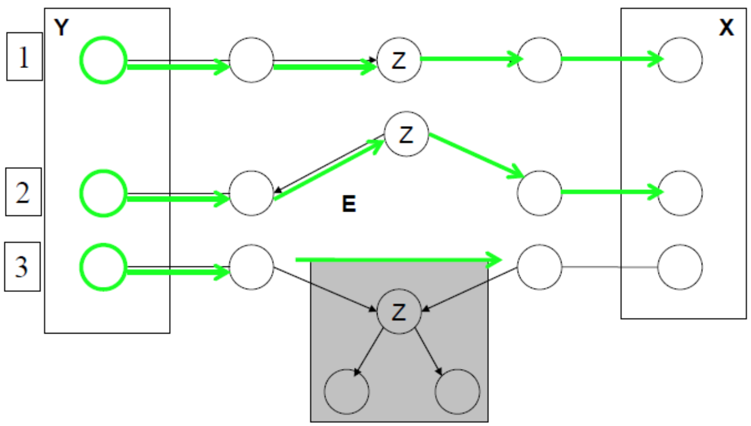
I.e., $H \perp\!\!\!\perp E \mid I$?

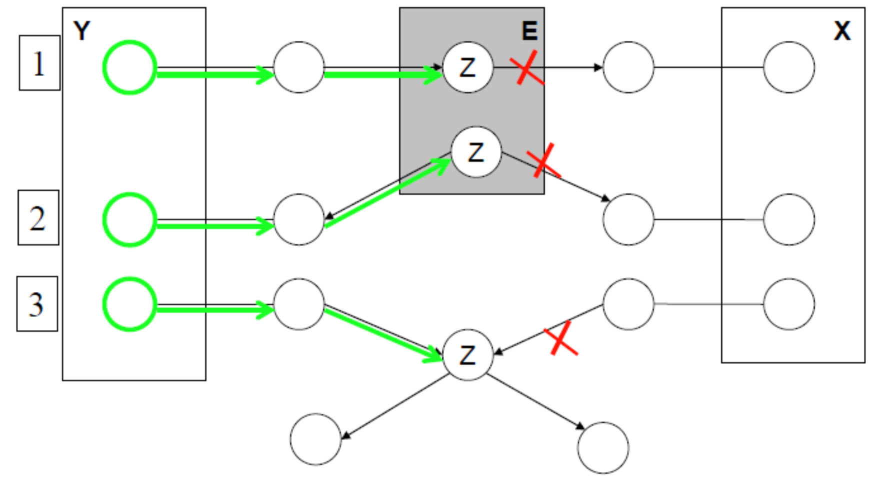
Yes

No



Conditional Independencies in a BN

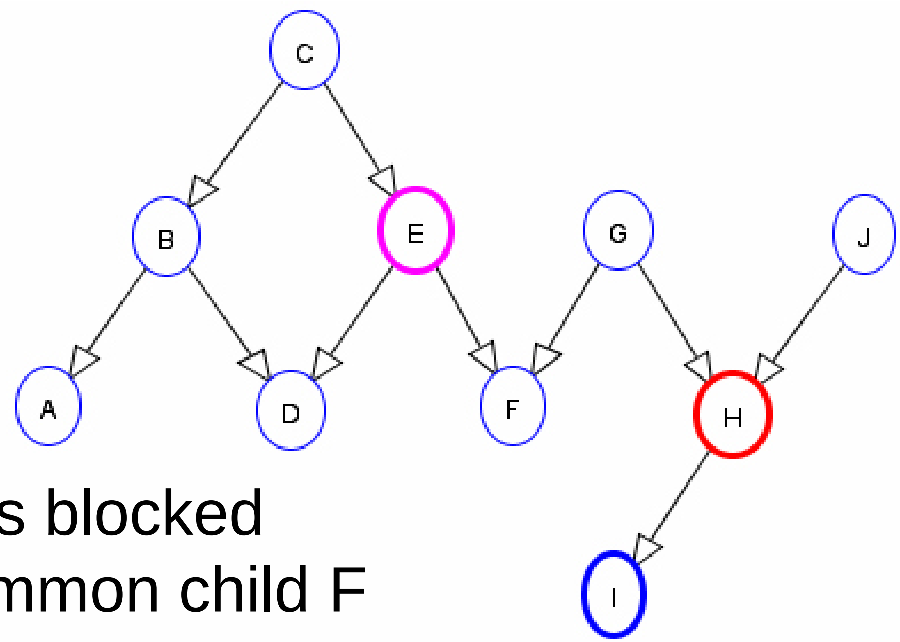




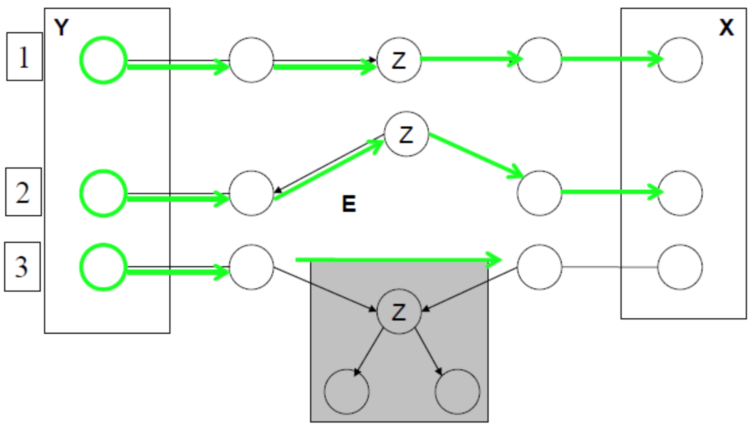
Is **H** conditionally independent of **E** given **I**?

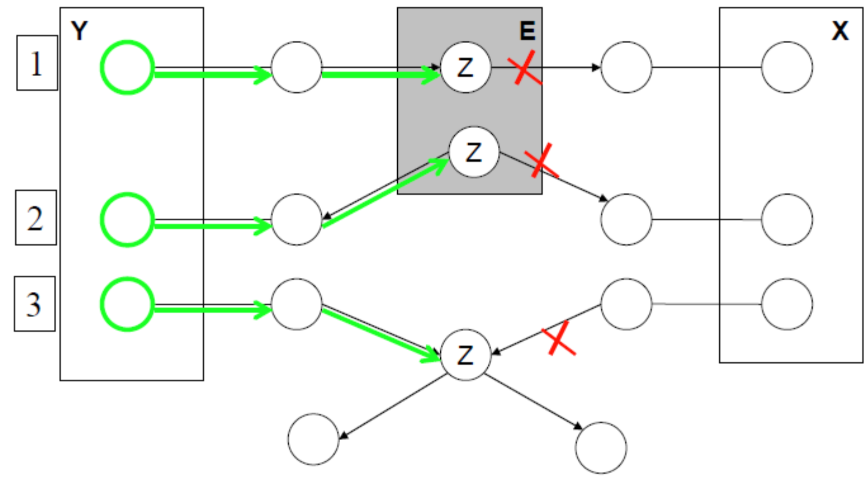
I.e., **H** $\perp\!\!\!\perp$ **E** | **I** ?

Yes! Information flow is blocked by the unobserved common child F



Conditional Independencies in a BN



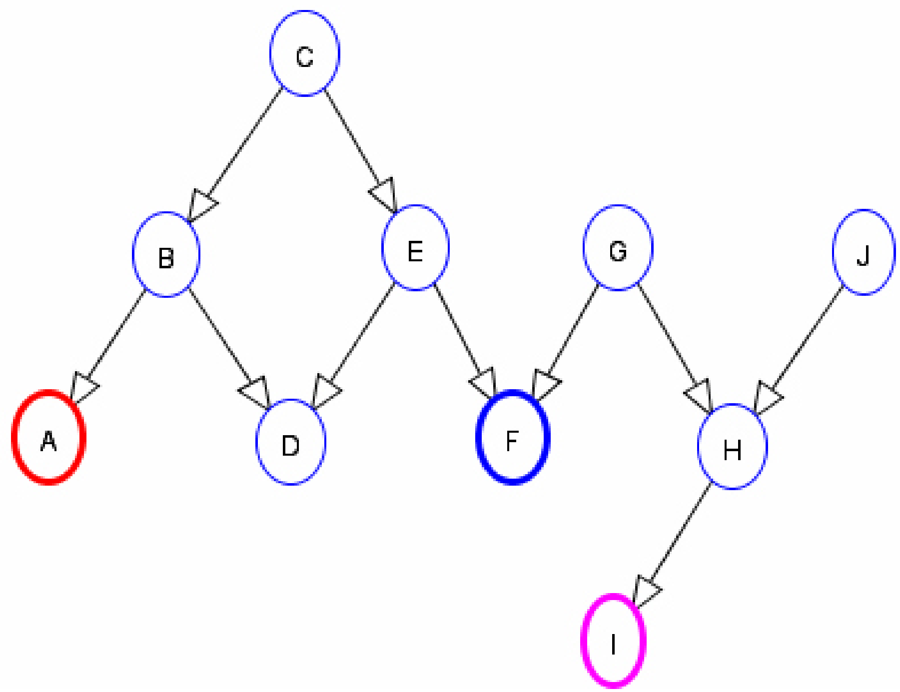


Is **A** conditionally independent of **I** given **F**?

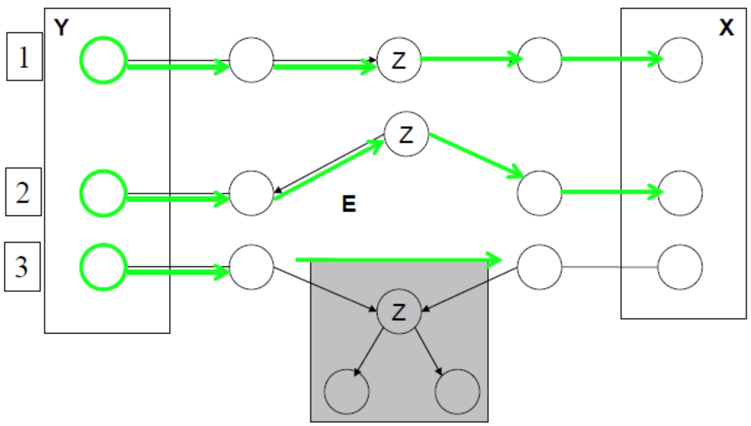
I.e., $A \perp\!\!\!\perp I \mid F$?

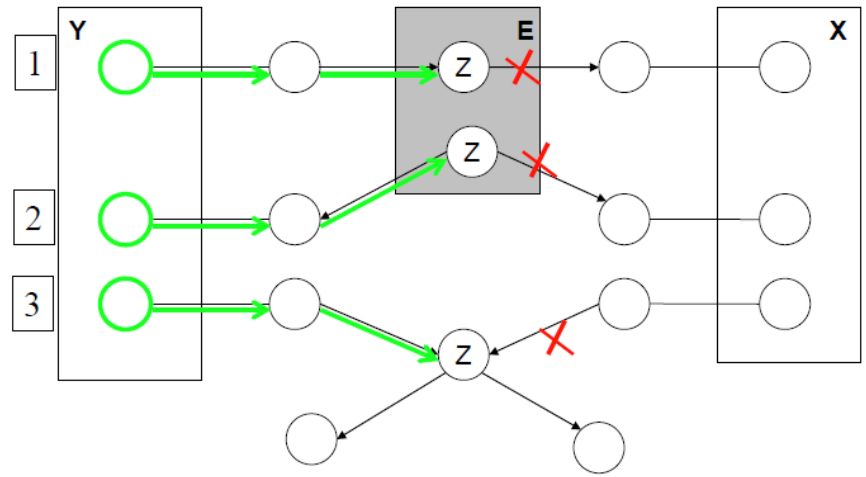
Yes

No



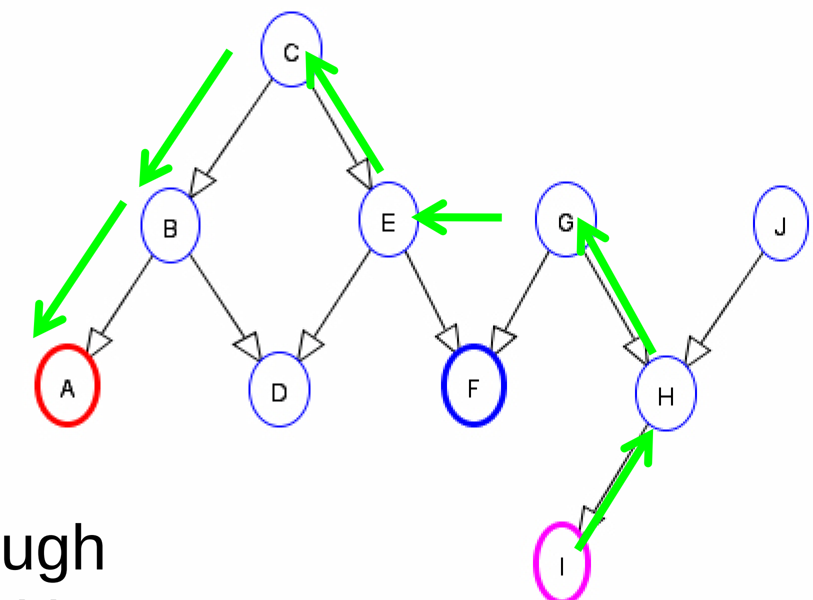
Conditional Independencies in a BN





Is **A** conditionally independent of **I** given **F**?

I.e., **A** ⫫ **I** | **F** ?



No. Information can pass through (all it takes is one possible path)

Learning Goals For Today's Class

- Build a Bayesian Network for a given domain
- Classify the types of inference:
  - Diagnostic, Predictive, Mixed, Intercausal
- Identify implied (in)dependencies in the network

- Assignment 4 available on WebCT
  - Due Monday, April 4
    - Can only use 2 late days
    - So we can give out solutions to study for the final exam
  - You should now be able to solve questions 1, 2, and 5
  - Questions 3 & 4: mechanical once you know the method
    - Method for Question 3: next Monday
    - Method for Question 4: next Wednesday/Friday

- Final exam: Monday, April 11
  - Less than 3 weeks from now