

# *Artificial Intelligence*

## Graphical Models

Marco Piastra

# Chain Factorization

## ■ Univariate factorization of a JPD

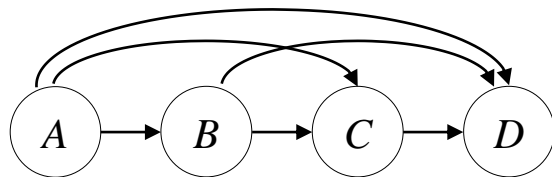
From the definition of conditional probability

$$P(A, B, C, D) = P(A)P(B|A)P(C|A, B)P(D|A, B, C)$$

Any joint probability distribution can be factorized in a way such that each factor is *univariate* (i.e. one random variable as independent) conditional distribution.

- Each factorization depends on an arbitrary *sequence* of the *random variables*
- Hence factorizations are not *unique*: any sequence produces a legitimate factorization of the same kind

Graphical equivalent



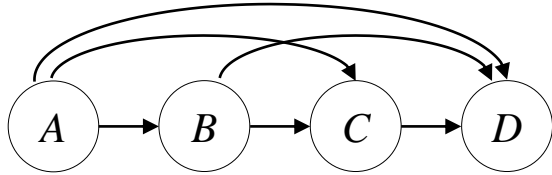
In this oriented graph:

- each node represents a random variable (and the corresponding *univariate* factor)
- each arc represents a conditioning of a random variable over another one (i.e. *dependence*)

# Chain Factorization

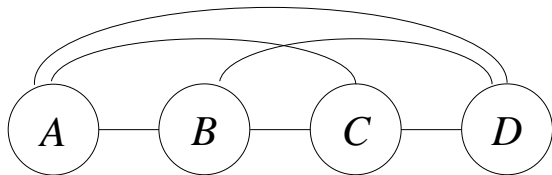
## ■ Graphical model

$$P(A, B, C, D) = P(A)P(B|A)P(C|A, B)P(D|A, B, C)$$



This graph:

- is *acyclic*: if you follow the arrows, you will never return to the same node
- is *completely connected*: if you ignore arc orientations, every node is connected to any other node



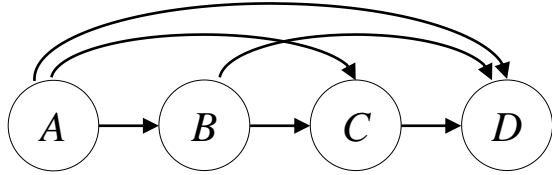
Any *univariate factorization* can be represented by a *graphical model*

Every *completely connected, acyclic and oriented graph* represents a *univariate factorization*

# Chain Factorization and Independence Assumptions

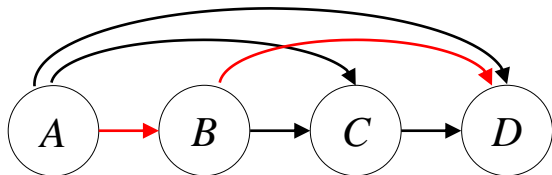
- **Graphical model**

$$P(A, B, C, D) = P(A)P(B|A)P(C|A, B)P(D|A, B, C)$$



- **Independence**

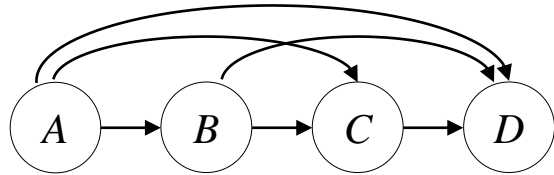
Let's remove a few arcs from the graph and rewrite the factorization accordingly



# Chain Factorization and Independence Assumptions

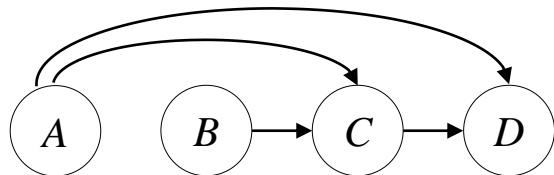
## ■ Graphical model

$$P(A, B, C, D) = P(A)P(B|A)P(C|A, B)P(D|A, B, C)$$



## ■ Independence

Let's remove a few arcs from the graph and rewrite the factorization accordingly



$$P(A, B, C, D) = P(A)P(B)P(C|A, B)P(D|A, C)$$

The latter holds true only if

$$P(B|A) = P(B)$$

$$P(D|A, B, C) = P(D|A, C)$$

$\langle A \perp B \rangle$  — Independence  
 $\langle B \perp D | A, C \rangle$  — Conditional Independence

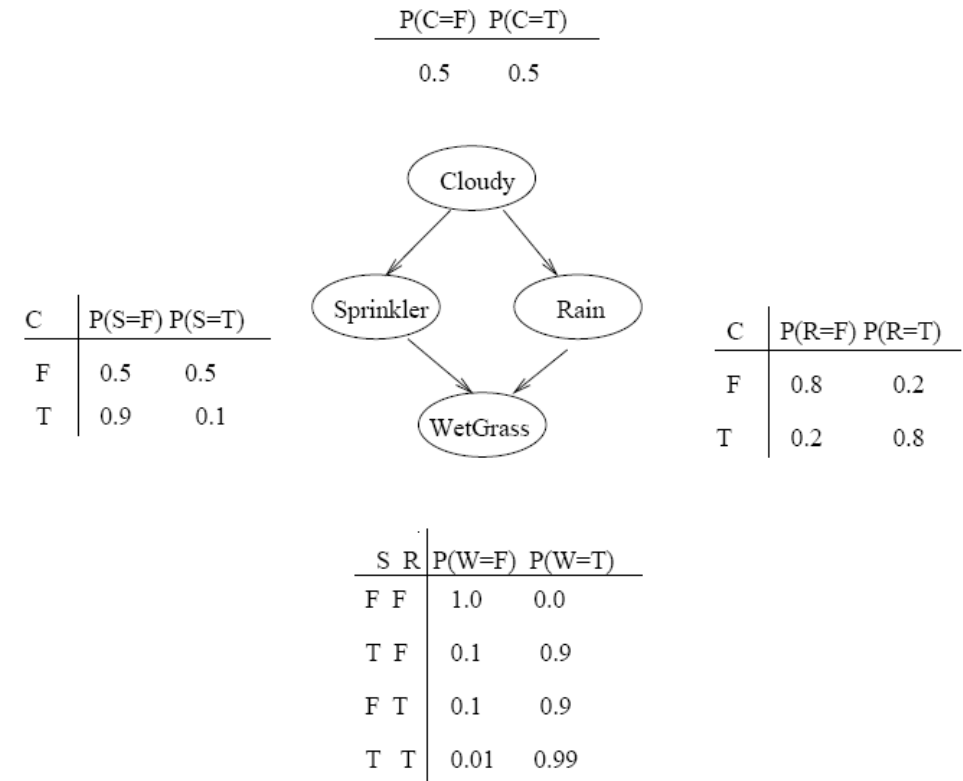
# Graphical models (a.k.a. Bayesian Networks)

Structure and numbers, instead of just numbers

- A structured, pre-numerical representation of a joint probability

Each graphical model is an *oriented* graph

- nodes are *random variables*
- arcs represent *dependence*



# From graphical models to joint probability

## Joint probability factorization

A chain factorization like the following is always allowed

$$P(C, S, R, W) = P(C)P(S|C)P(R|C, S)P(W|C, S, R)$$

Hint: apply the definition of conditional probability repeatedly  
(such factorization is not unique)

A complete specification  
of a joint probability would require  
 $2^4 = 16$  values

The values in figure are just 9

## Factorization for a graphical model

$$P(X_1, X_2, \dots, X_n) = \prod_i P(X_i | \text{parents}(X_i))$$

where  $\text{parents}(X_i)$  are the nodes from which  
there is an entry arc to  $X_i$

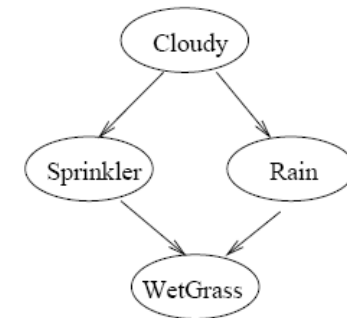
For this example, the above rule produces:

$$P(C, S, R, W) = P(C)P(S|C)P(R|C)P(W|S, R)$$

Note the difference from above

Independence assumptions:  $\langle R \perp S | C \rangle, \langle W \perp C | R, S \rangle$

	P(C=F)	P(C=T)
	0.5	0.5



C	P(S=F)	P(S=T)
F	0.5	0.5
T	0.9	0.1

C	P(R=F)	P(R=T)
F	0.8	0.2
T	0.2	0.8

S	R	P(W=F)	P(W=T)
F	F	1.0	0.0
T	F	0.1	0.9
F	T	0.1	0.9
T	T	0.01	0.99

# Patterns in Graphical Models

## ■ **Sequence or Chain**

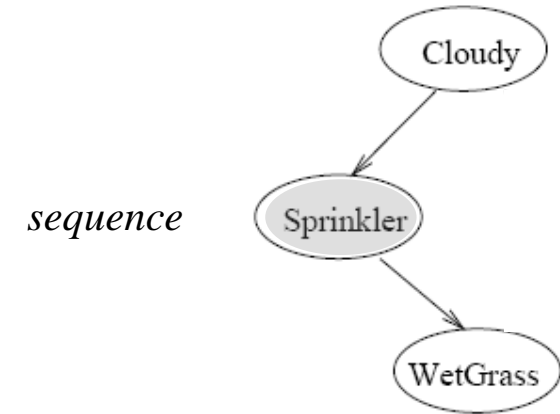
Consider the graph on the right

$$P(C, S, W) = P(C)P(S|C)P(W|S)$$

Now suppose you observe  $S$

$$\begin{aligned} P(C, W|S) &= \frac{P(C, S, W)}{P(S)} \\ &= \frac{P(C)P(S|C)P(W|S)}{P(S)} \\ &= \frac{P(C, S)}{P(S)} P(W|S) \\ &= P(C|S)P(W|S) \end{aligned}$$

This implies  $\langle C \perp W | S \rangle$





# Patterns in Graphical Models

- **Fork**

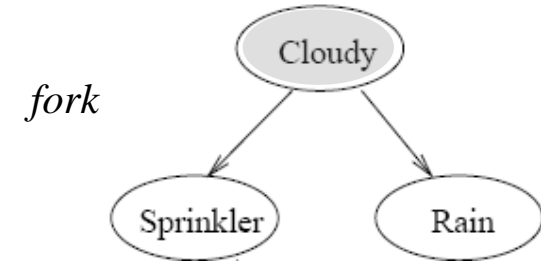
Consider the graph on the right

$$P(C, S, R) = P(C)P(S|C)P(R|C)$$

Now suppose you observe  $C$

$$\begin{aligned} P(R, S|C) &= \frac{P(C, S, R)}{P(C)} \\ &= \frac{P(C)P(S|C)P(R|C)}{P(C)} \\ &= \frac{P(C, S)}{P(C)}P(R|C) \\ &= P(S|C)P(R|C) \end{aligned}$$

This implies  $\langle R \perp S | C \rangle$



# Patterns in Graphical Models

## ■ **Join or Collider**

*CAUTION: this case is different from the previous two*

Consider the graph on the right

$$P(R, S, W) = P(S)P(R)P(W|S, R)$$

which is true only if  $\langle S \perp R \rangle$

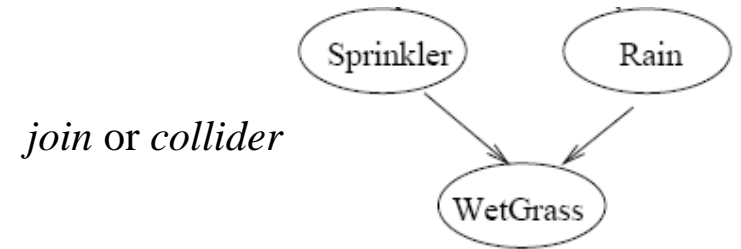
*Independence (also  
'Marginal Independence')*

Now suppose you observe  $W$

$$\begin{aligned} P(R, S|W) &= \frac{P(R, S, W)}{P(W)} \\ &= \frac{P(S)P(R)P(W|S, R)}{P(W)} \\ &\neq P(S|W)P(R|W) \end{aligned}$$

*No further simplification  
possible*

This implies  $\langle S \not\perp R | W \rangle$



# Patterns in Graphical Models

## ■ **Join or Collider**

*The same loss of independence occurs if you observe any of the descendants...*

Consider the graph on the right

$$P(R, S, W, D) = P(S)P(R)P(W|S, R)P(D|W)$$

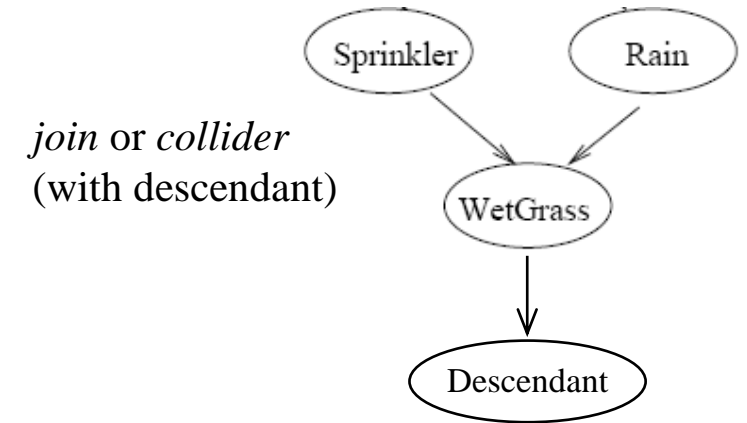
which is true only if  $\langle S \perp R \rangle$  *Independence (also 'Marginal Independence')*

Now suppose you observe  $D$

$$\begin{aligned} P(R, S, W|D) &= \frac{P(R, S, W, D)}{P(D)} \\ &= \frac{P(S)P(R)P(W|S, R)P(D|W)}{P(D)} \quad \text{No further simplification possible} \\ &\neq P(S|D)P(R|D) \end{aligned}$$

This implies  $\langle S \not\perp R | D \rangle$

*... at any subsequent level of descendance (try yourself)*



# Paths in Graphical Models

In a graphical model

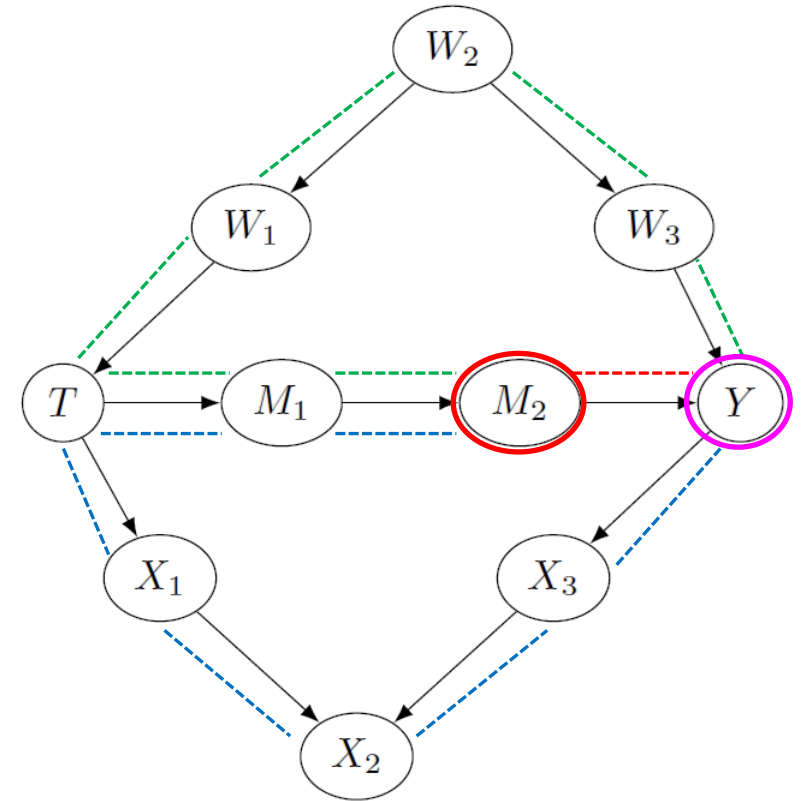
Consider any two nodes  $A$  and  $B$

A *path* between  $A$  and  $B$

is a path in the graph ignoring orientation (i.e. arrows)

*Example:*

In the graph on the right,  
consider all paths between  $M_2$  and  $Y$



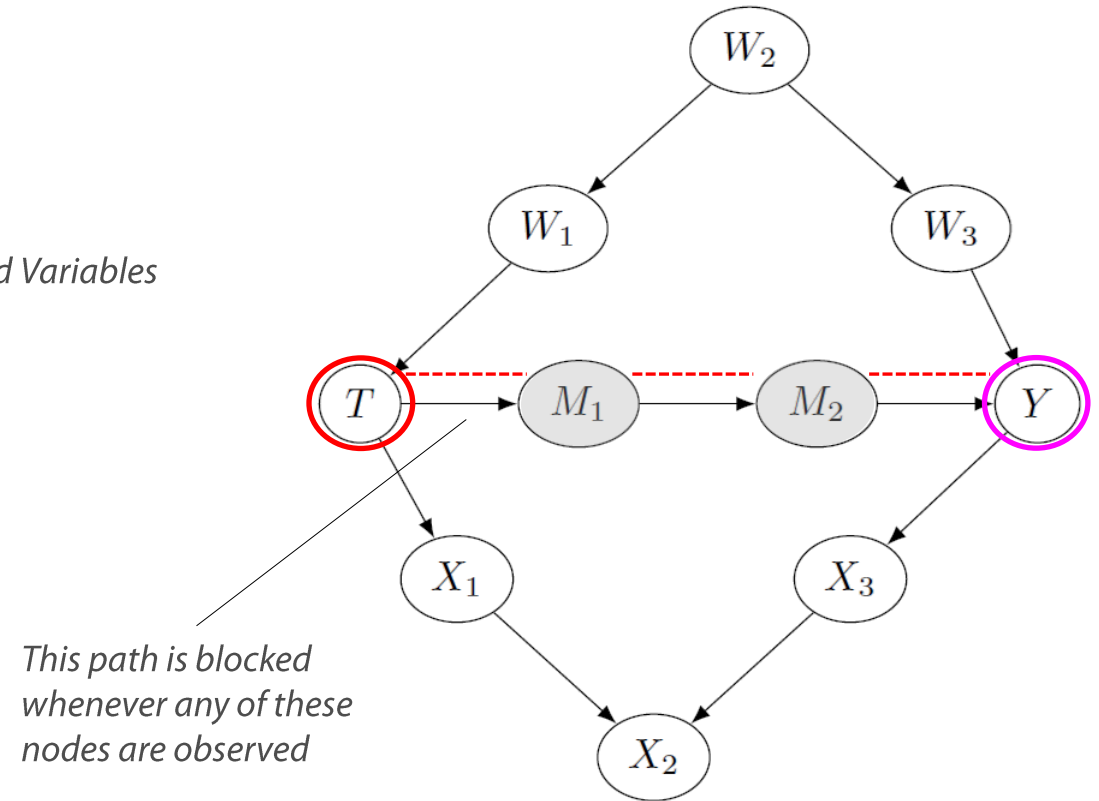
# Blocked Paths in Graphical Models

In a graphical model

A *path* between any two nodes  $A$  and  $B$  is **blocked** whenever the observations  $\{X_o\}$  are such that the path contains either:

Observed Variables

- 1) a *sequence* or a *fork* for which one observation  $X \in \{X_o\}$  creates a condition of independence
- 2) a *collider* for which  $\{X_o\}$  does not contain the observation of the join node nor of any of its descendants



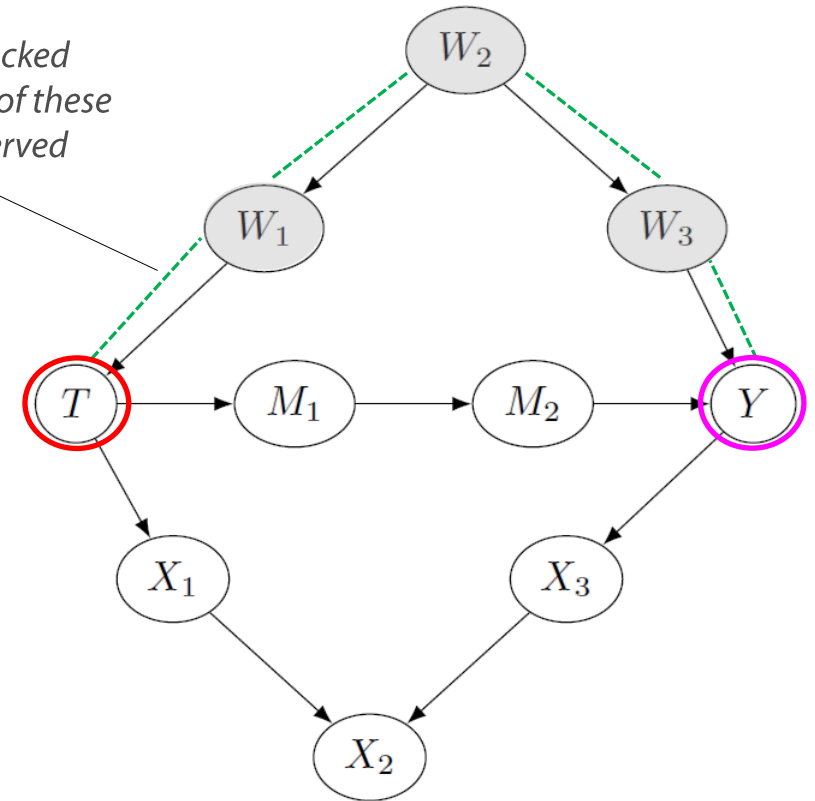
# Blocked Paths in Graphical Models

In a graphical model

A path between any two nodes  $A$  and  $B$  is **blocked** whenever the observations  $\{X_o\}$  are such that the path contains either:

- 1) a *sequence* or a *fork* for which one observation  $X \in \{X_o\}$  creates a condition of independence
- 2) a *collider* for which  $\{X_o\}$  does not contain the observation of the join node nor of any of its descendants

This path is blocked whenever any of these nodes are observed

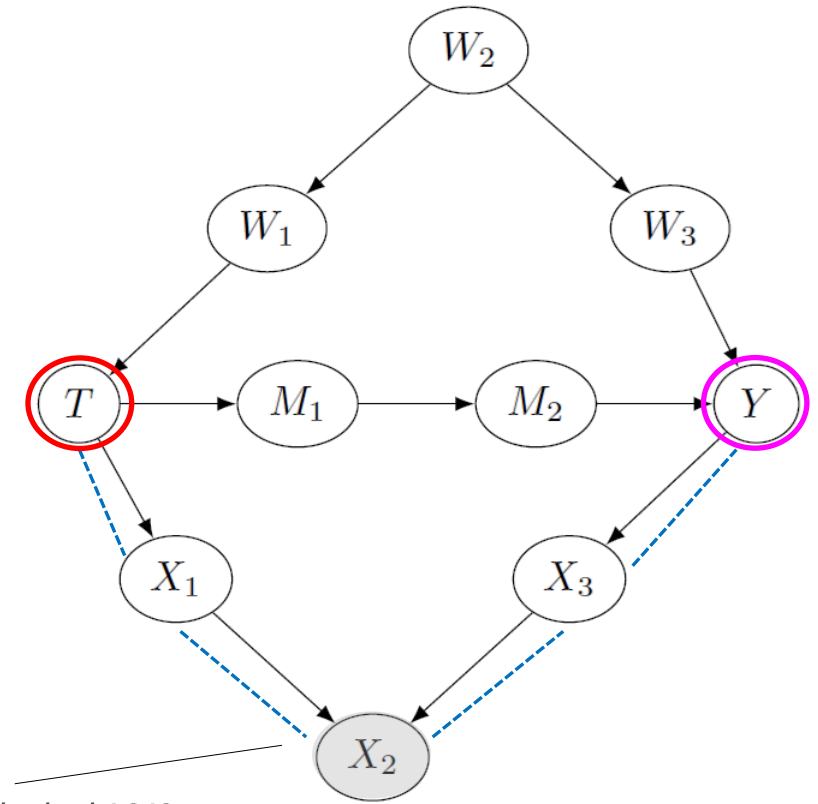


# Blocked Paths in Graphical Models

In a graphical model

A *path* between any two nodes  $A$  and  $B$  is **blocked** whenever the observations  $\{X_o\}$  are such that the path contains either:

- 1) a *sequence* or a *fork* for which one observation  $X \in \{X_o\}$  creates a condition of independence
- 2) a *collider* for which  $\{X_o\}$  does not contain the observation of the join node nor of any of its descendants



This path is blocked AS IS:  
the collider blocks it

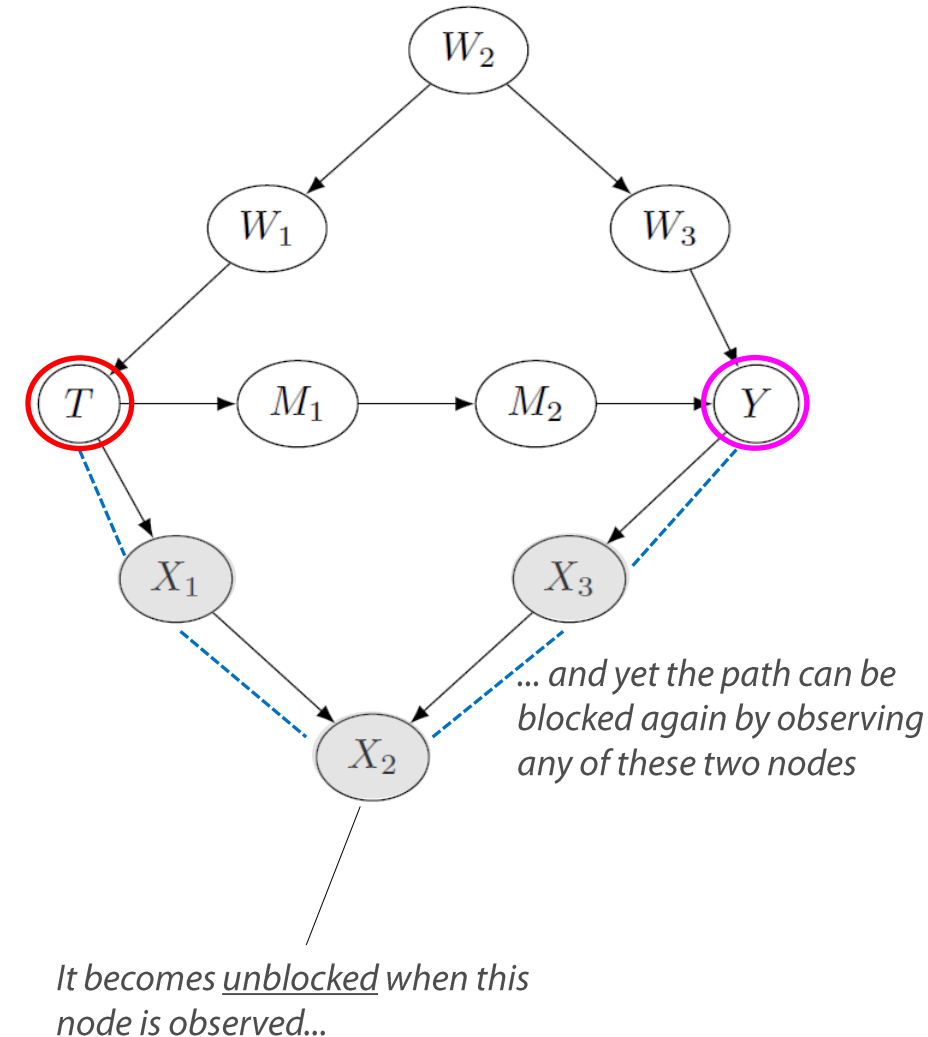
It becomes unblocked when this  
node is observed...

# Blocked Paths in Graphical Models

In a graphical model

A *path* between any two nodes  $A$  and  $B$  is **blocked** whenever the observations  $\{X_o\}$  are such that the path contains either:

- 1) a *sequence* or a *fork* for which one observation  $X \in \{X_o\}$  creates a condition of independence
- 2) a *collider* for which  $\{X_o\}$  does not contain the observation of the join node nor of any of its descendants





# D-Separation in Graphical Models

## ▪ Dependency Separation (d-separation)

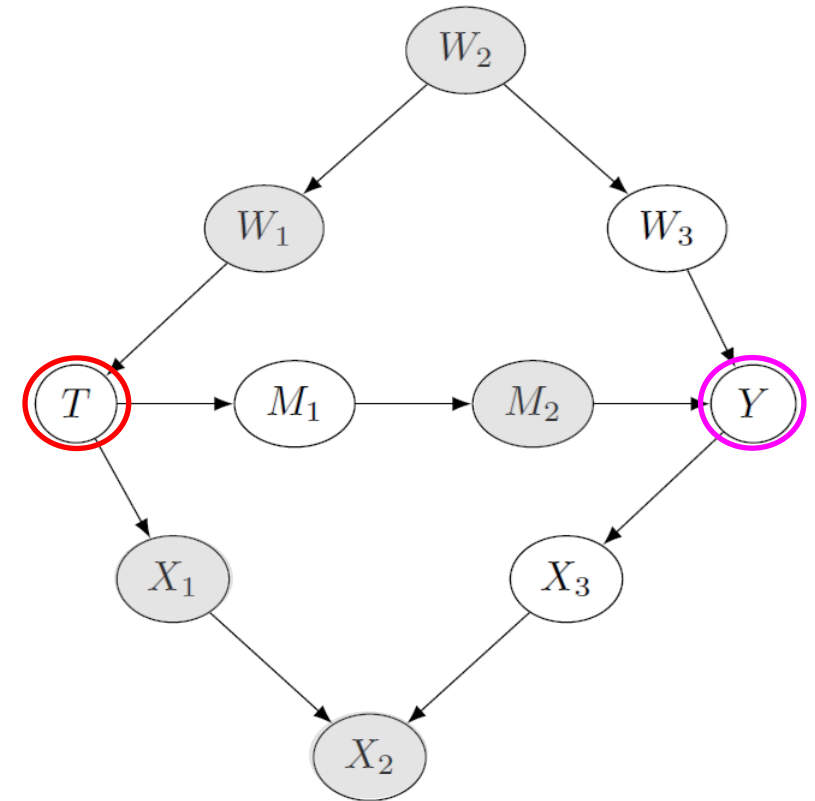
Any two nodes  $A$  and  $B$  in a graphical model are ***d-separated*** whenever the observations  $\{X_o\}$  are such that all paths between  $A$  and  $B$  are blocked

*Observed Variables*

In that case we have

$$\langle A \perp B \mid \{X_o\} \rangle$$

*But only when all paths are blocked*



*These observations make the two nodes d-separated*

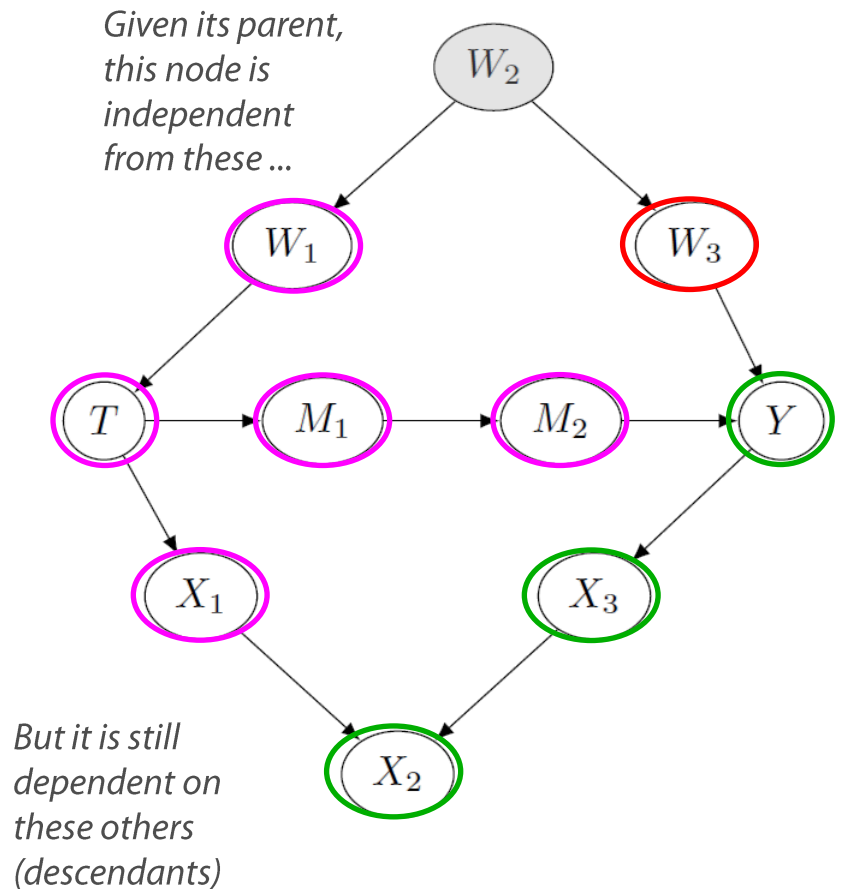
# Graphical models: fundamental assumptions

- **Minimality**

Adjacent nodes in the graph are dependent.

- **Local Markov Assumption**

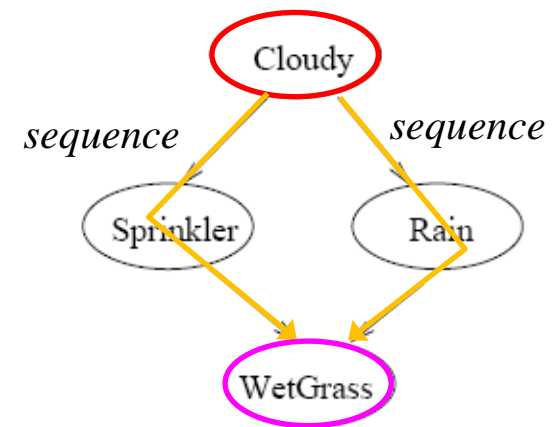
Given its parents in the graph, a node  $A$  is independent of all its non-descendants



# D-Separation in Graphical Models

Example:

Cloudy and WetGrass are independent when both paths in color are blocked



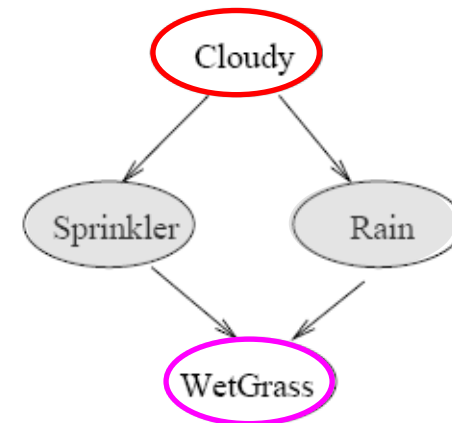
# D-Separation in Graphical Models

Example:

Cloudy and WetGrass are independent when both paths in color are blocked

These are two *sequences*:  
Sprinkler and Rain must be known

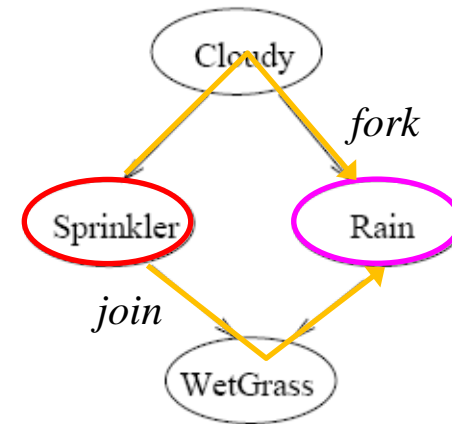
$\langle \text{Cloudy} \perp \text{WetGrass} \mid \text{Sprinkler, Rain} \rangle$



# D-Separation in Graphical Models

Example:

Sprinkler and Rain are independent when both paths in color are blocked



# D-Separation in Graphical Models

Example:

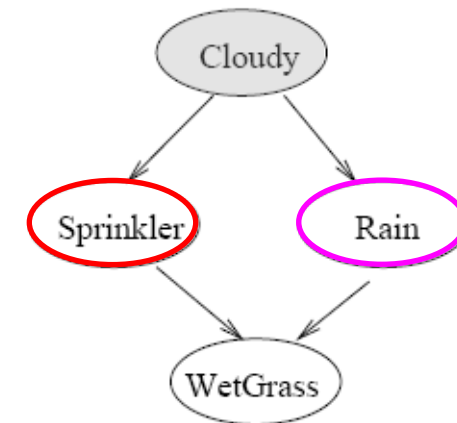
Sprinkler and Rain are independent when both paths in color are blocked

One *fork* and one *collider*:

Cloudy must be known whereas WetGrass must be unknown

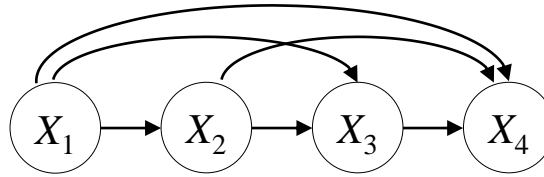
< Sprinkler  $\perp$  Rain | Cloudy >

*Check more examples and quiz with Bayes program (see course webpage)!*



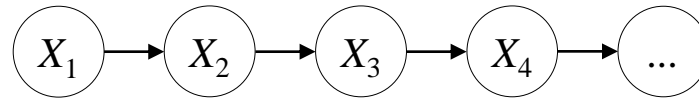
# Example of graphical models

- Complete dependency



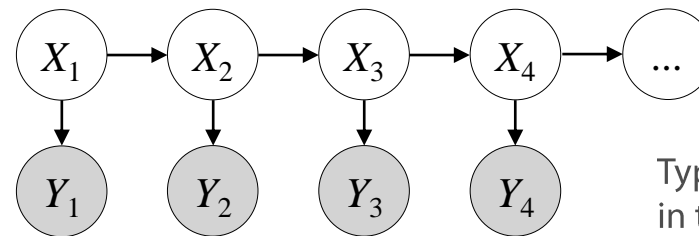
$$P(X_1, X_2, X_3, X_4) = P(X_1) P(X_2 | X_1) P(X_3 | X_1, X_2) P(X_4 | X_1, X_2, X_3)$$

- Markovian model



$$P(X_1, \dots, X_n) = P(X_1) \prod_{i=2}^n P(X_i | X_{i-1})$$

- 'Hidden' Markovian model

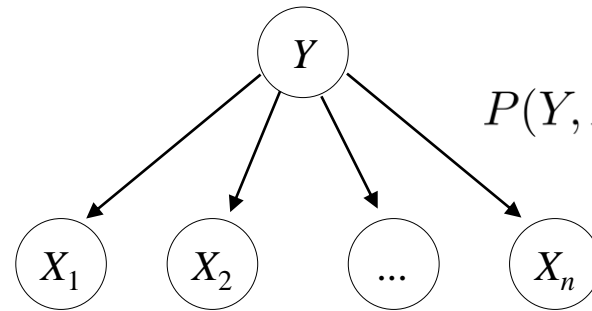


Typically, nodes  $X_i$  are *hidden*, in the sense of *non-observable* (see later, about *learning*)

$$P(X_1, \dots, X_n Y_1, \dots, Y_n) = P(X_1) P(Y_1 | X_1) \prod_{i=2}^n P(X_i | X_{i-1}) P(Y_i | X_i)$$

# Example: *anti-spam filter*

a.k.a. 'Naïve (Discrete) Bayesian Classifier'



$$P(Y, X_1, \dots, X_n) = P(Y) \prod_{i=1}^n P(X_i | Y)$$

*Anti-spam filter:*

- All random variables are *binomial* (value: either 0 or 1)
- $Y$  represents the class of the message: 1 *spam*, 0 *not-spam*
- Each  $X_i$  represents the occurrence of the word  $i$  in the message

Assume (*for now*) that the probabilities are given

As we will see, finding the 'right' numbers is a *learning* problem (see after)



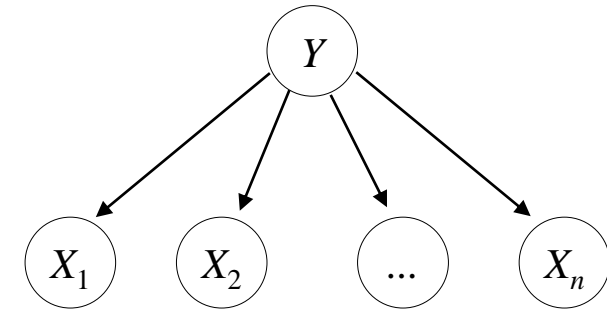
# Inference in the *anti-spam filter*

$$P(Y, X_1, \dots, X_n) = P(Y) \prod_{i=1}^n P(X_i | Y)$$

Given a message with occurrence values  $\{X_k\}$ ,  
the class with the highest conditional probability is determined

The message is  
*spam* if

$$\frac{P(Y = 1 | \{X_k\})}{P(Y = 0 | \{X_k\})} > \lambda$$



Note that:

$$P(Y = 1 | \{X_k\}) \stackrel{\text{Bayes' Theorem}}{=} \frac{P(\{X_k\} | Y = 1)P(Y = 1)}{\sum_Y P(\{X_k\} | Y)P(Y)} \stackrel{\text{Conditional independency}}{=} \frac{P(Y = 1) \prod_k P(X_k | Y = 1)}{\sum_Y P(Y) \prod_k P(X_k | Y)}$$

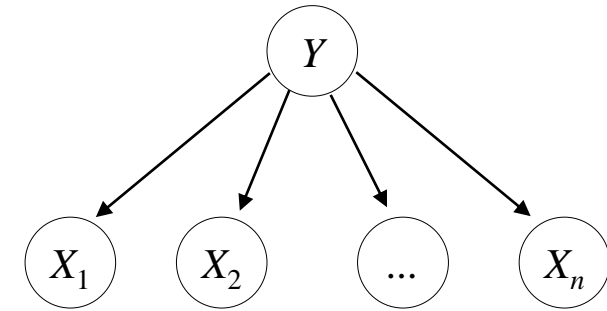
# Inference in the *anti-spam filter*

$$P(Y, X_1, \dots, X_n) = P(Y) \prod_{i=1}^n P(X_i | Y)$$

Given a message with occurrence values  $\{X_k\}$ ,  
the class with the highest conditional probability is determined

The message is  
*spam* if

$$\frac{P(Y = 1 | \{X_k\})}{P(Y = 0 | \{X_k\})} > \lambda$$



Note that:

$$P(Y = 1 | \{X_k\}) \stackrel{\text{Bayes' Theorem}}{=} \frac{P(\{X_k\} | Y = 1)P(Y = 1)}{\sum_Y P(\{X_k\} | Y)P(Y)} = \frac{P(Y = 1) \prod_k P(X_k | Y = 1)}{\sum_Y P(Y) \prod_k P(X_k | Y)} \stackrel{\text{Conditional independency}}{=}$$

Therefore:

$$\frac{P(Y = 1 | \{X_k\})}{P(Y = 0 | \{X_k\})} = \frac{P(Y = 1)}{P(Y = 0)} \prod_k \frac{P(X_k | Y = 1)}{P(X_k | Y = 0)}$$

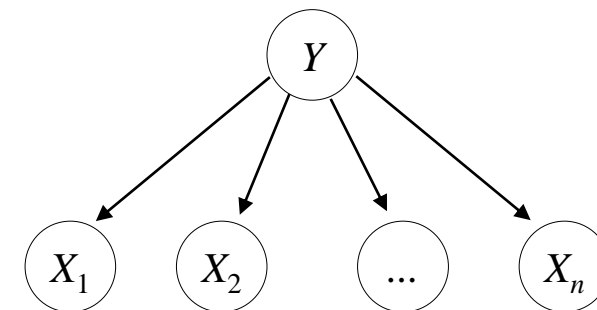
# Inference in the *anti-spam filter*

$$P(Y, X_1, \dots, X_n) = P(Y) \prod_{i=1}^n P(X_i | Y)$$

Given a message with occurrence values  $\{X_k\}$ ,  
the class with the highest conditional probability is determined

The message is  
*spam* if

$$\frac{P(Y = 1 | \{X_k\})}{P(Y = 0 | \{X_k\})} > \lambda$$



Note that:

$$P(Y = 1 | \{X_k\}) \stackrel{\text{Bayes' Theorem}}{=} \frac{P(\{X_k\} | Y = 1)P(Y = 1)}{\sum_Y P(\{X_k\} | Y)P(Y)} = \frac{P(Y = 1) \prod_k P(X_k | Y = 1)}{\sum_Y P(Y) \prod_k P(X_k | Y)} \stackrel{\text{Conditional independency}}{=}$$

Therefore:

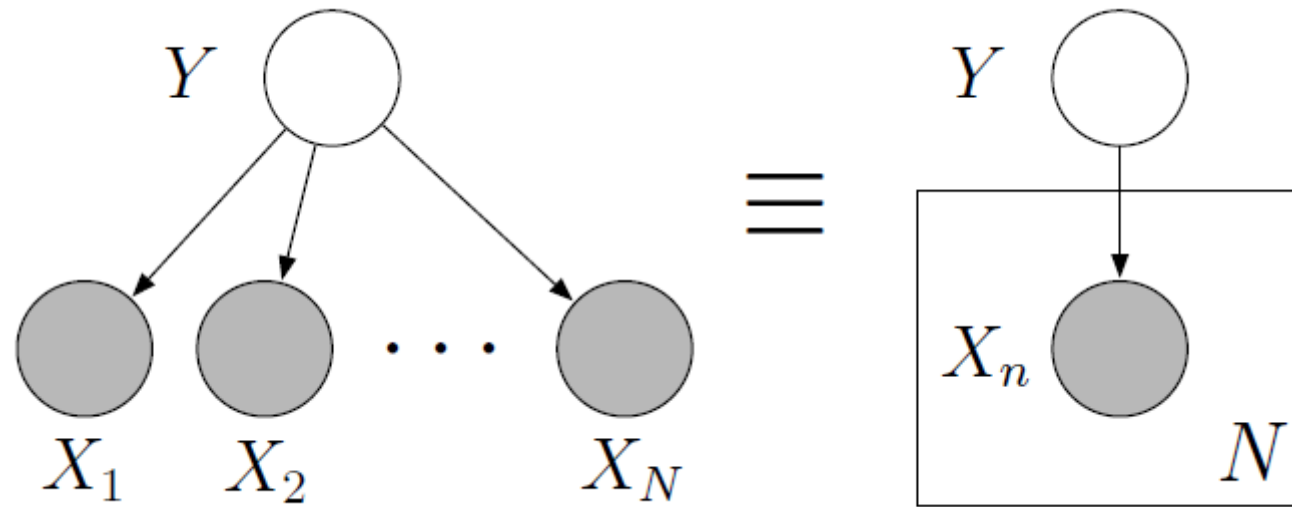
$$\frac{P(Y = 1 | \{X_k\})}{P(Y = 0 | \{X_k\})} = \frac{P(Y = 1)}{P(Y = 0)} \prod_k \frac{P(X_k | Y = 1)}{P(X_k | Y = 0)}$$

The logarithm is used  
to simplify computations:

$$\log \frac{P(Y = 1 | \{X_k\})}{P(Y = 0 | \{X_k\})} = \log \frac{P(Y = 1)}{P(Y = 0)} + \sum_k \log \frac{P(X_k | Y = 1)}{P(X_k | Y = 0)}$$

# An aside: plate notation

A shorthand notation for graphical models



# Building a graphical model

- Step 1

Defining the nodes, i.e. the random variables

*T* : (tampering)

*F* : (fire)

*A* : (alarm)

*S* : (smoke)

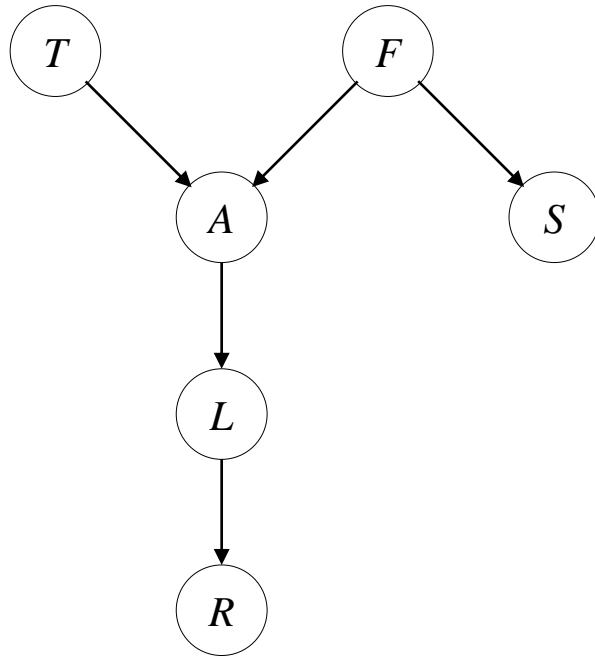
*L* : (leaving)

*R* : (report)

# Building a graphical model

## ■ Step 2

Defining the structure, i.e. the graph



We are thus saying that:

$\langle T \perp F \rangle$  (but they become dependent when any of  $A$ ,  $L$  or  $R$  are known)

$\langle A \perp S \mid F \rangle$

$\langle L \perp T \mid A \rangle$

$\langle L \perp F \mid A \rangle$

$\langle A \perp R \mid L \rangle$

$T$ : (*tampering*)

$F$ : (*fire*)

$A$ : (*alarm*)

$S$ : (*smoke*)

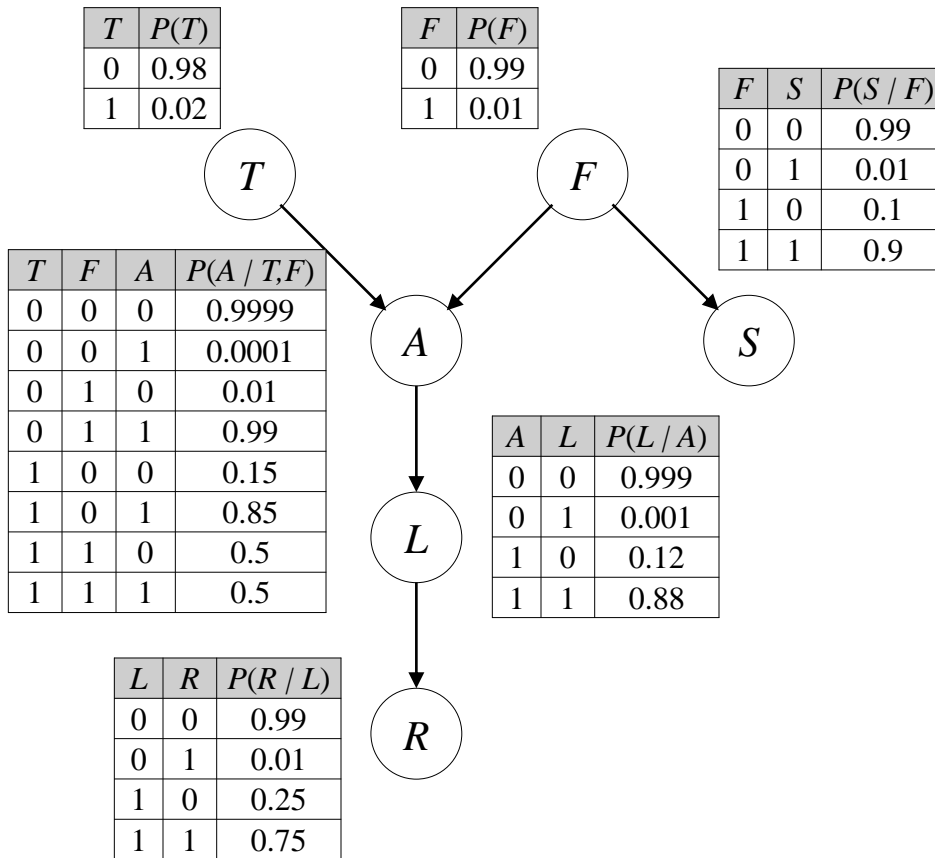
$L$ : (*leaving*)

$R$ : (*report*)

# Building a graphical model

## ■ Step 3

Defining *conditional probability tables – CPTs*



$T$  : (tampering)

$F$  : (fire)

$A$  : (alarm)

$S$  : (smoke)

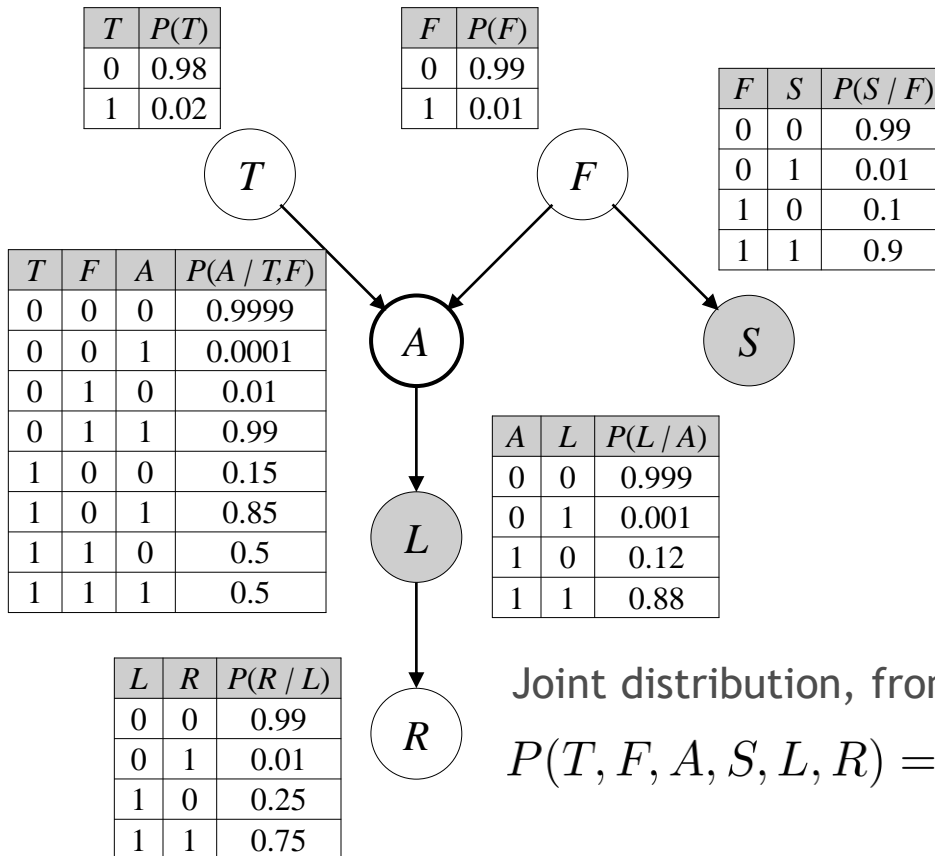
$L$  : (leaving)

$R$  : (report)

# Probabilistic inference

## Step 4

Consider a specific problem



Example: finding  $A$  given  $L=1$  e  $S=0$

$$P(A|L = 1, S = 0) = \frac{P(A, L = 1, S = 0)}{P(L = 1, S = 0)}$$

Joint distribution, from the graph:

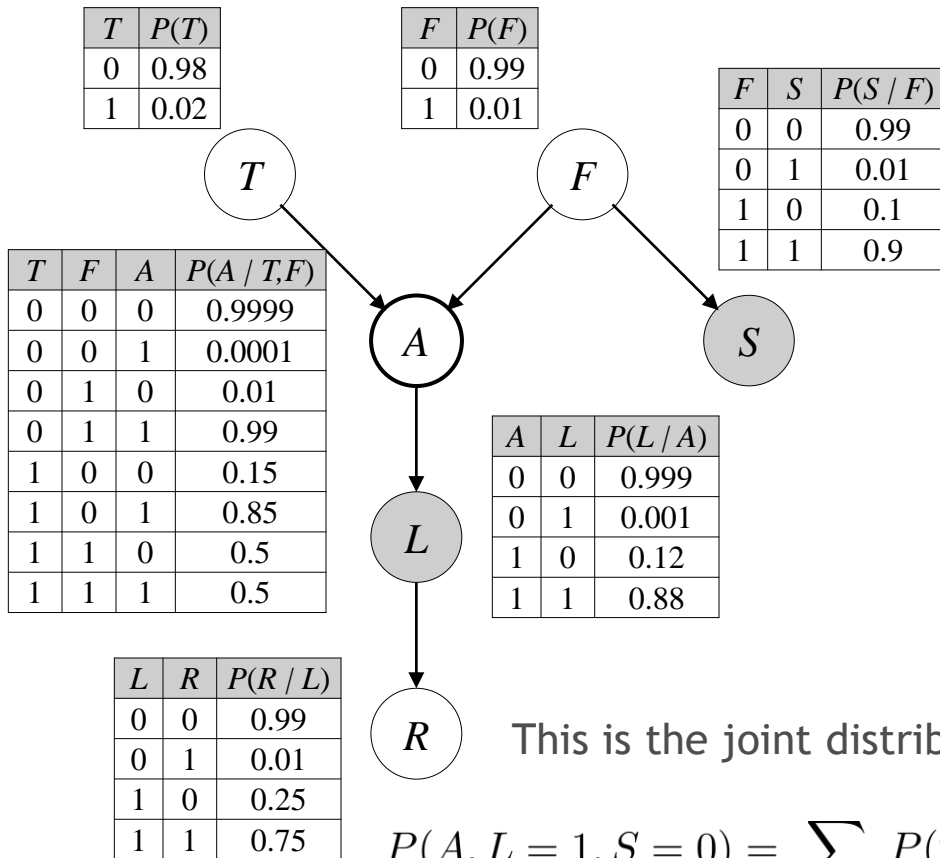
$$P(T, F, A, S, L, R) = P(T)P(F)P(A|T, F)P(S|F)P(L|A)P(R|L)$$



# Probabilistic inference

## Step 5

### Computing the answer



Note that:

$$P(A|L = 1, S = 0) = \frac{P(A, L = 1, S = 0)}{P(L = 1, S = 0)}$$

This is a normalizing term:  
it can be computed from  
 $P(A, L = 1, S = 0)$

In fact:

$$P(L = 1, S = 0) = \sum_A P(A, L = 1, S = 0)$$

Typically, the most time-consuming computations  
in an inference problem are marginalizations

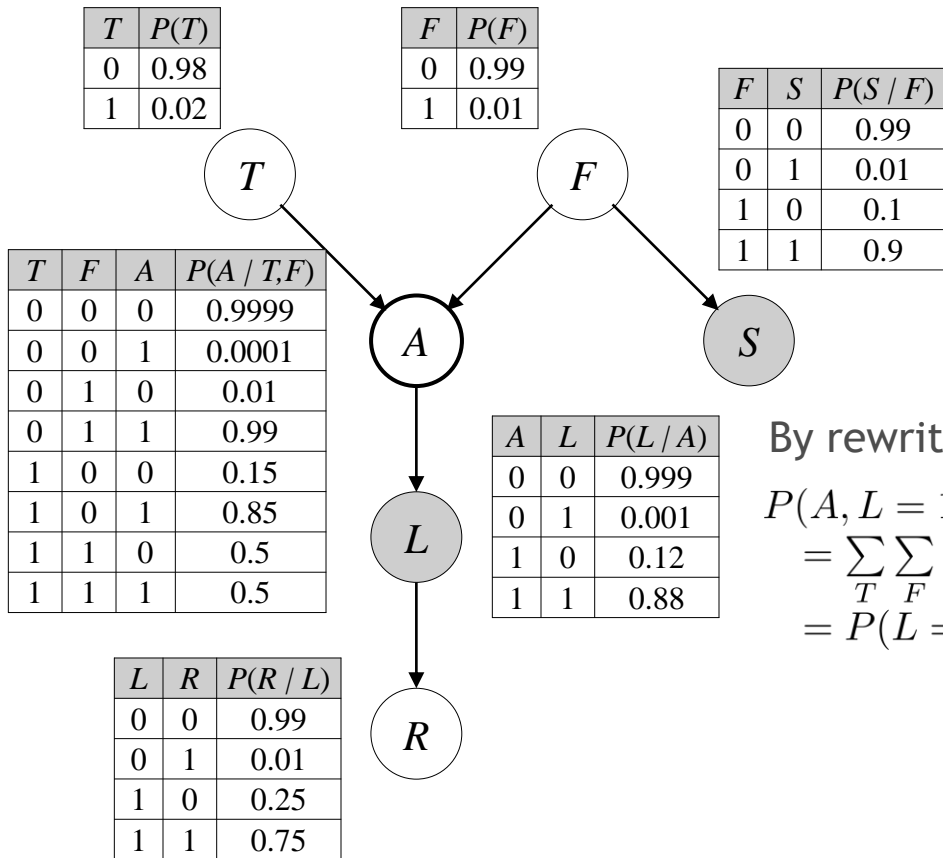
This is the joint distribution to be computed:

$$P(A, L = 1, S = 0) = \sum_{T, F, R} P(T)P(F)P(A|T, F)P(S = 0|F)P(L = 1|A)P(R|L = 1)$$

# Probabilistic inference


## Step 5

Computing the answer



By rewriting the joint distribution:

$$\begin{aligned}
 P(A, L = 1, S = 0) &= \sum_T \sum_F \sum_R P(L = 1|A)P(A|T, F)P(T)P(F)P(S = 0|F)P(R|L = 1) \\
 &= P(L = 1|A) \sum_T \sum_F P(A|T, F)P(T)P(F)P(S = 0|F) \sum_R P(R|L = 1)
 \end{aligned}$$

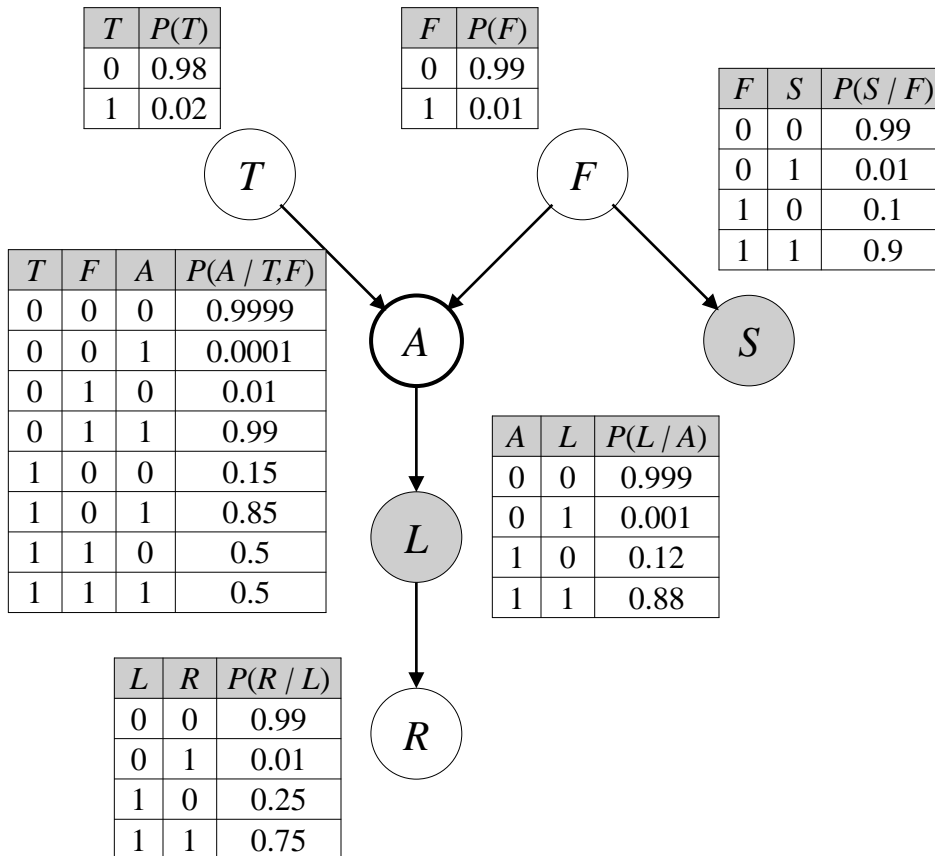

 This sum has value 1  
 This is not surprising  
 given that  $\langle A \perp R | L \rangle$

# Probabilistic inference

## Step 5

Computing the answer

$$P(A, L = 1, S = 0) = P(L = 1|A) \sum_T \sum_F P(A|T, F)P(T)P(F)P(S = 0|F)$$



By convention, we write:

$$P(A, L = 1, S = 0) = f_{T,F,S=0}(A) f_{L=1}(A)$$

where the  $f$  are the *factors* of the method also known as *elimination of variables*:

$$f_{T,F,S=0}(A) := \sum_T \sum_F P(A|T, F)P(T)P(F)P(S = 0|F)$$

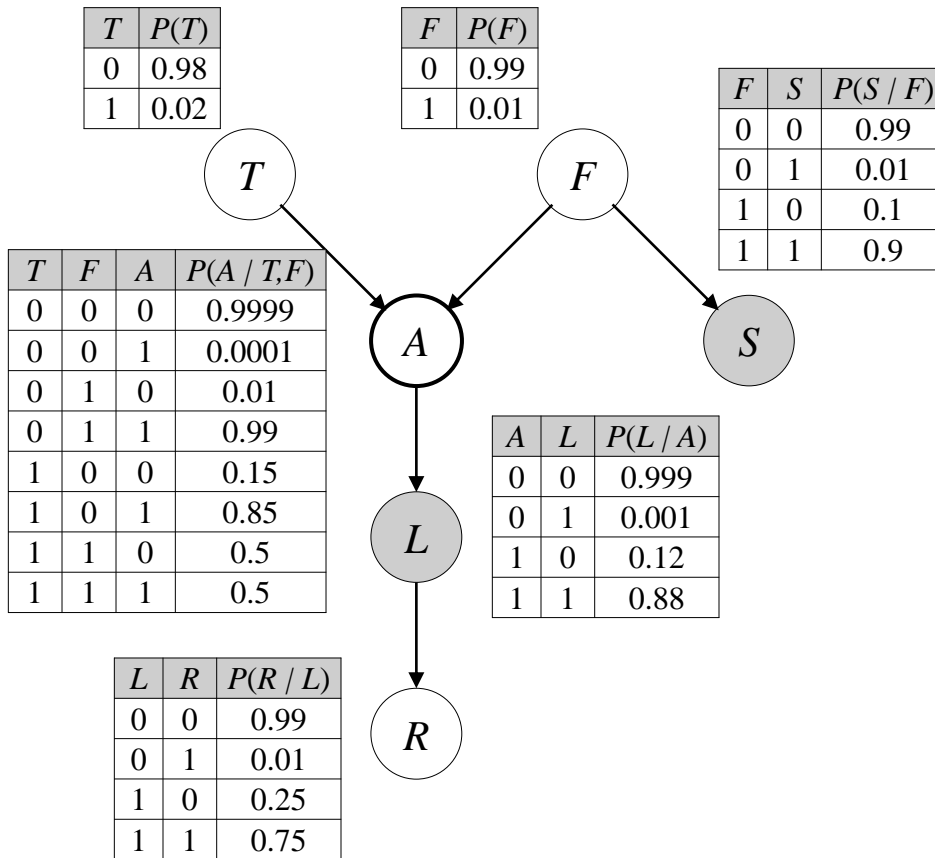
$$f_{L=1}(A) := P(L = 1|A)$$

Note in passing that *factors*  $f$  are not probabilities (i.e. they do not sum to 1).

# Probabilistic inference

## Step 5

Computing the answer



Note that:

$$P(A, L = 1, S = 0) = f_{T,F,S=0}(A) f_{L=1}(A)$$

This factor comes from  
the *parents* of  $A$

This factor comes from  
the *descendants* of  $A$

This is true  
for any node  $A$  that *d-separates* the graph

# Variable elimination for graphical models

## ■ General idea

Write the marginal joint probability from the query in the form:

$$P(\{X_r\}, \{X_o\}) = \sum_{\{X_i\}} \prod_X P(X \mid \text{parents}(X))$$

- 1) Find the best ordering of terms for the marginalization of irrelevant variables:
- 2) Move summations 'inside' the product as much as possible (i.e. find *factors*  $f$ )
- 3) Compute factors (i.e. by sum of products) and obtain numbers (i.e. *terms*)
- 4) Plug these *terms* into the product and obtain a simpler form for  $P(\{X_r\}, \{X_o\})$
- 5) Wrap it up and compute the response:

$$P(\{X_r\}|\{X_o\}) = \frac{P(\{X_r\}, \{X_o\})}{\sum_{\{X_r\}} P(\{X_r\}, \{X_o\})}$$

*Remember: the method is NP-complete (anyway)*

# Graphical models as a probabilistic method

## ■ Advantages

Correctness (of representation)

$$\langle \{X\} \perp \{Y\} \mid \{Z\} \rangle_{GM} \Rightarrow \langle \{X\} \perp \{Y\} \mid \{Z\} \rangle_{JPD}$$

Independence in the graph model  
↓  
implies independence in the joint probability distribution

In a *finitary setting*, they are always computable

Graph models are easy to read (compared to JPDs)

## ■ Limitations

No *abstraction* over multiplicity

(i.e. no First-order Logic equivalent – see also <http://www.pr-owl.org/basics/bn.php#reasoning>)

- Consider you receive multiple reports (random variable  $R$ ) of fire: do they support each other? Which ones are reliable?
- Time sequences or specific patterns of variable size

No *completeness*

$$\langle \{X\} \perp \{Y\} \mid \{Z\} \rangle_{JPD} \not\Rightarrow \langle \{X\} \perp \{Y\} \mid \{Z\} \rangle_{GM}$$

- Counterexample: no DAG can represent

$$\langle X_1 \perp \{X_2, Y_2\} \rangle, \langle X_2 \perp \{X_1, Y_1\} \rangle$$

Not all JPDs can be faithfully represented  
by a graph model

without introducing some further independence relation

(no closure under marginalization - see also [https://projecteuclid.org/download/pdf\\_1/euclid.aos/1031689015](https://projecteuclid.org/download/pdf_1/euclid.aos/1031689015))