Graphical Models

Marco Piastra

Graphical models *(Bayesian Networks)*

Structure and numbers, instead of just numbers

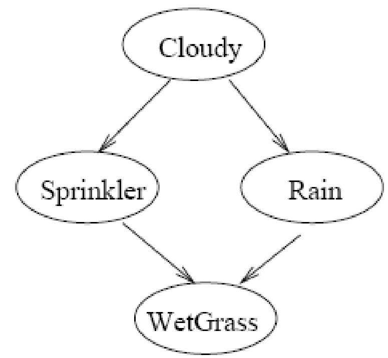
- A structured, pre-numerical representation of a joint probability

  Each model is an *oriented* graph

  The nodes are *random variables*

  The arcs represent *dependence*

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

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

The values in figure are just 9

From graphical models to joint probability

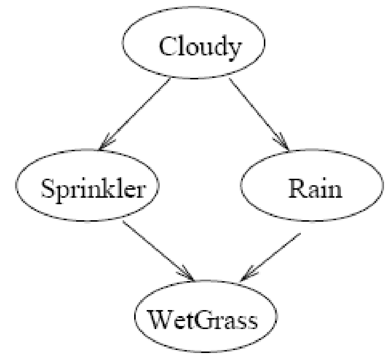
- Joint probability

  It can be expressed as a product of conditional probabilities

  (due to the extension of the *chain rule*)

  Example:

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

$$P(C, S, R, W) = P(C)P(S \mid C)P(R \mid S,C)P(W \mid R,S,C)$$

  In a graphical model, the joint distribution is

$$P(X_1, X_2, \ldots, X_n) = \prod_i P(X_i \mid parents(X_i))$$

  Where $parents(X_i)$ the nodes from which there is an entry arc to X_i

  In the example:

$$P(C, S, R, W) = P(C)P(S \mid C)P(R \mid C)P(W \mid R,S)$$

  Conditional independence assumptions: $\langle R \perp S \mid C \rangle$, $\langle W \perp C \mid R, S \rangle$

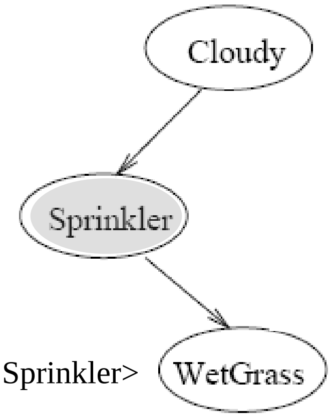
Graphical models and conditional independence

- *D-separation (Dependency-separation)*

  i.e. how to read a graphical model

  In a graphical model

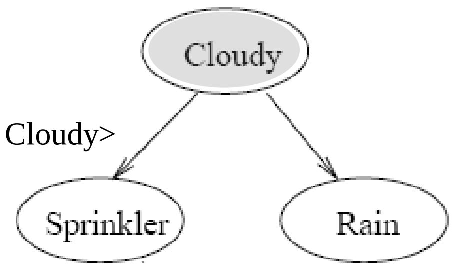
  Two nodes X and Y are conditional independent given a set of nodes $\{Z_k\}$ when **all** paths are blocked (see below)



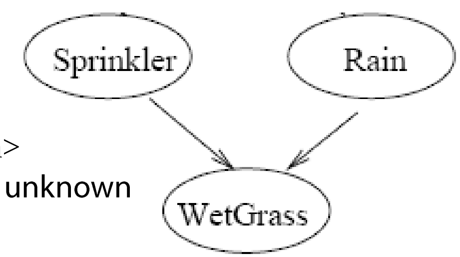
<WetGrass ⊥ Cloudy | Sprinkler>

A path between X e Y is blocked if:

1) It is either a sequence $X \rightarrow \ldots Z_i \ldots \rightarrow Y$ or a *fork* $X \leftarrow \ldots Z_i \ldots \rightarrow Y$ $(Z_i \in \{Z_k\})$
2) It is a *join* $X \rightarrow \ldots N \ldots \leftarrow Y$ where neither N nor all the *descendants* of N belong to $\{Z_k\}$



<Sprinkler ⊥ Rain | Cloudy>



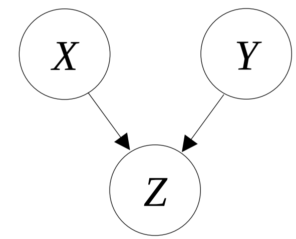
<Sprinkler ⊥ Rain>
when WetGrass is unknown

Explaining Away

A few more words on condition 2) of *D-separation*

Graphical model, with a *join*

Joint probability, from the graph:



$$P(X, Y, Z) = P(X)P(Y)P(Z|X,Y)$$

Marginal probability w.r.t X and Y (Z unknown):

$$P(X, Y) = P(X)P(Y)\sum_{Z} P(Z|X,Y) = P(X)P(Y)$$

Therefore X e Y are *marginally independent*

But when Z is known, then X and Y are <u>*dependent*</u>:

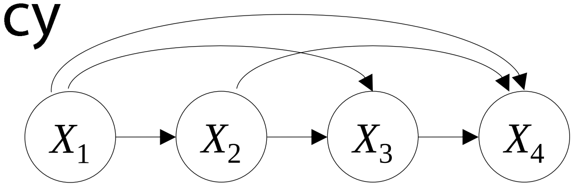
$$P(X, Y \mid Z=v) = \frac{P(X,Y,Z=v)}{P(Z=v)} = \frac{P(X)P(Y)P(Z=v|X,Y)}{\sum_{X,Y} P(X)P(Y)P(Z=v|X,Y)}$$

It is not a paradox.

Example:
X and Y are two tosses of the same coin, Z=1 if the result is the same, Z=0 otherwise.

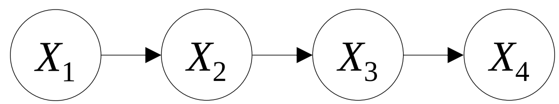
Example of graphical models

- Complete dependency



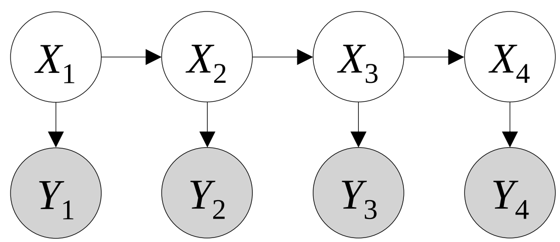
$$P(X_1, X_2, X_3, X_4) = P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1, X_2)P(X_4 \mid X_1, X_2, X_3)$$

- Markovian model



$$P(X_1, X_2, X_3, X_4) = P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_2)P(X_4 \mid X_3) = P(X_1)\prod_{i=2}^{n} P(X_i \mid X_{i-1})$$

- 'Hidden' Markovian model



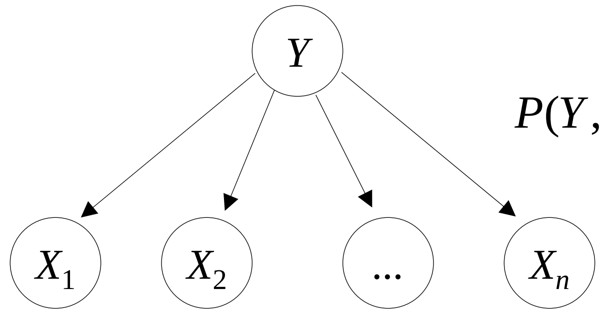
Typically, nodes X_i are *hidden*, in the sense of *non-observable* (see later, about *learning*)

$$P(X_1, X_2, X_3, X_4, Y_1, Y_2, Y_3, Y_4) = P(X_1)P(Y_1 \mid X_1)P(X_2 \mid X_1)P(Y_2 \mid X_2)P(X_3 \mid X_2)P(Y_3 \mid X_3)P(X_4 \mid X_3)P(Y_4 \mid X_4)$$

$$= P(X_1)P(Y_1 \mid X_1)\prod_{i=2}^{n} P(X_i \mid X_{i-1})P(Y_i \mid X_i)$$

Example: *anti-spam filter*

Typically (e.g. Mozilla Thunderbird): '*Naive* (*Discrete*) *Bayesian Classifier*'



$$P(Y, X_1, X_2, \ldots, X_n) = P(Y) \prod_{i=1}^{n} P(X_i \mid Y)$$

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 i word in the message

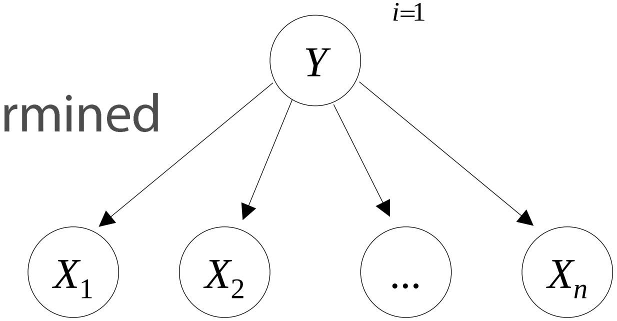
Assume (*for now*) that the probabilities are given

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

Inference in the *anti-spam filter*

$$P(Y, X_1, X_2, \ldots, X_n) = P(Y) \prod_{i=1}^{n} P(X_i \mid Y)$$

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



The message is *spam* if

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

Note that:

$$P(Y=1 \mid \{X_k\}) = \frac{P(\{X_k\} \mid Y=1)P(Y=1)}{\sum_Y P(\{X_k\} \mid Y)P(Y)} = \frac{P(Y=1)\prod_k P(X_k \mid Y=1)}{\sum_Y P(Y)\prod_k P(X_k \mid Y)}$$

Bayes' Theorem

Conditional independency

Therefore:

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

The logarithm is used
to simplify computations:

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

Building a graphical model

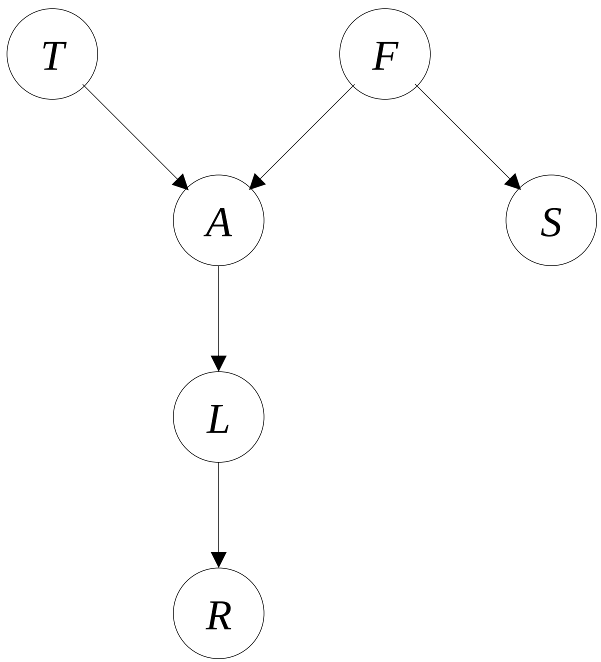
- Step 1

  Defining the nodes, i.e. the random variables

T : (*tampering*)
F : (*fire*)
A : (*alarm*)
S : (*smoke*)
L : (*leaving*)
R : (*report*)

- Step 2

  Defining the structure, i.e. the graph



We are thus saying that:

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

$<A \perp S \mid F>$

$<L \perp T \mid A>$

$<L \perp F \mid A>$

$<A \perp R \mid L>$

T : (*tampering*)

F : (*fire*)

A : (*alarm*)

S : (*smoke*)

L : (*leaving*)

R : (*report*)

Building a graphical model

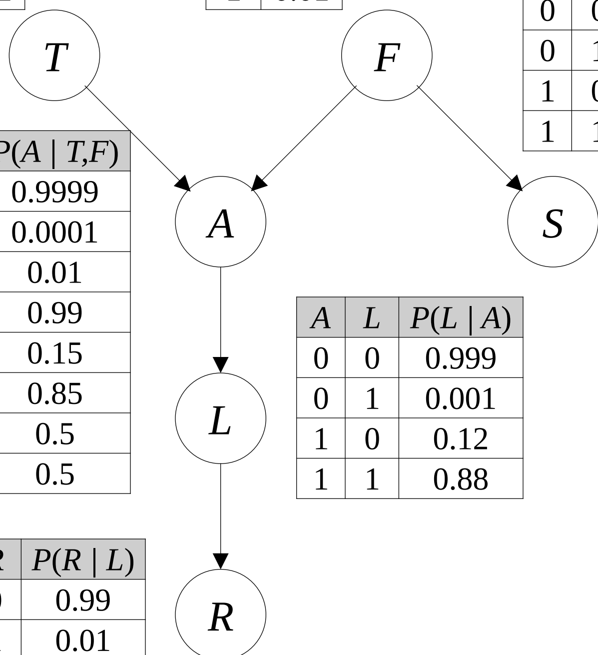
- Step 3

  Defining *conditional probability tables – CPTs*

T	$P(T)$
0	0.98
1	0.02

F	$P(F)$
0	0.99
1	0.01

F	S	$P(S \mid F)$
0	0	0.99
0	1	0.01
1	0	0.1
1	1	0.9



T	F	A	$P(A \mid T,F)$
0	0	0	0.9999
0	0	1	0.0001
0	1	0	0.01
0	1	1	0.99
1	0	0	0.15
1	0	1	0.85
1	1	0	0.5
1	1	1	0.5

A	L	$P(L \mid A)$
0	0	0.999
0	1	0.001
1	0	0.12
1	1	0.88

L	R	$P(R \mid L)$
0	0	0.99
0	1	0.01
1	0	0.25
1	1	0.75

T : (*tampering*)
F : (*fire*)
A : (*alarm*)
S : (*smoke*)
L : (*leaving*)
R : (*report*)

Probabilistic inference

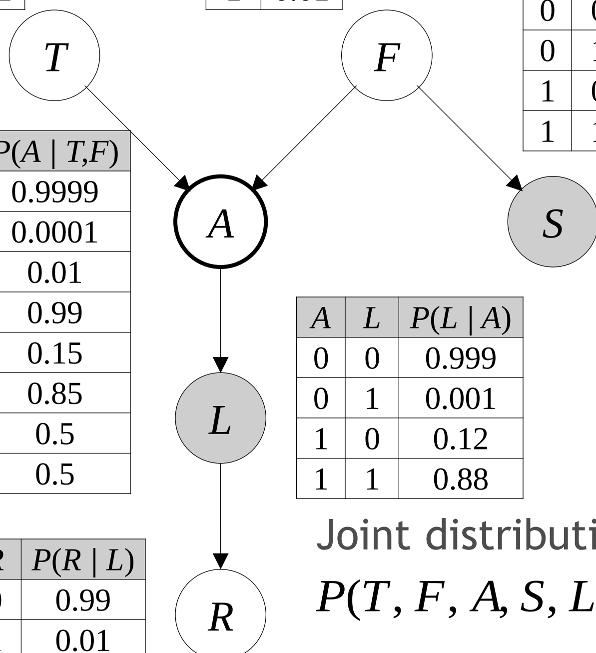
- Step 4

  Defining a specific problem

T	$P(T)$
0	0.98
1	0.02

F	$P(F)$
0	0.99
1	0.01

F	S	$P(S \mid F)$
0	0	0.99
0	1	0.01
1	0	0.1
1	1	0.9



T	F	A	$P(A \mid T,F)$
0	0	0	0.9999
0	0	1	0.0001
0	1	0	0.01
0	1	1	0.99
1	0	0	0.15
1	0	1	0.85
1	1	0	0.5
1	1	1	0.5

A	L	$P(L \mid A)$
0	0	0.999
0	1	0.001
1	0	0.12
1	1	0.88

L	R	$P(R \mid L)$
0	0	0.99
0	1	0.01
1	0	0.25
1	1	0.75

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:

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

hence, at the numerator:

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

Probabilistic inference

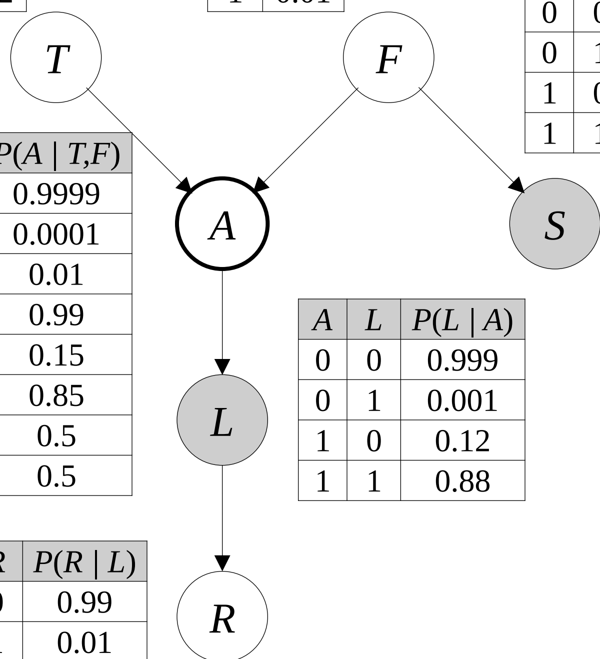
- Step 5

  Computing the answer

T	$P(T)$
0	0.98
1	0.02

F	$P(F)$
0	0.99
1	0.01

F	S	$P(S \mid F)$
0	0	0.99
0	1	0.01
1	0	0.1
1	1	0.9



T	F	A	$P(A \mid T,F)$
0	0	0	0.9999
0	0	1	0.0001
0	1	0	0.01
0	1	1	0.99
1	0	0	0.15
1	0	1	0.85
1	1	0	0.5
1	1	1	0.5

A	L	$P(L \mid A)$
0	0	0.999
0	1	0.001
1	0	0.12
1	1	0.88

L	R	$P(R \mid L)$
0	0	0.99
0	1	0.01
1	0	0.25
1	1	0.75

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

Probabilistic inference

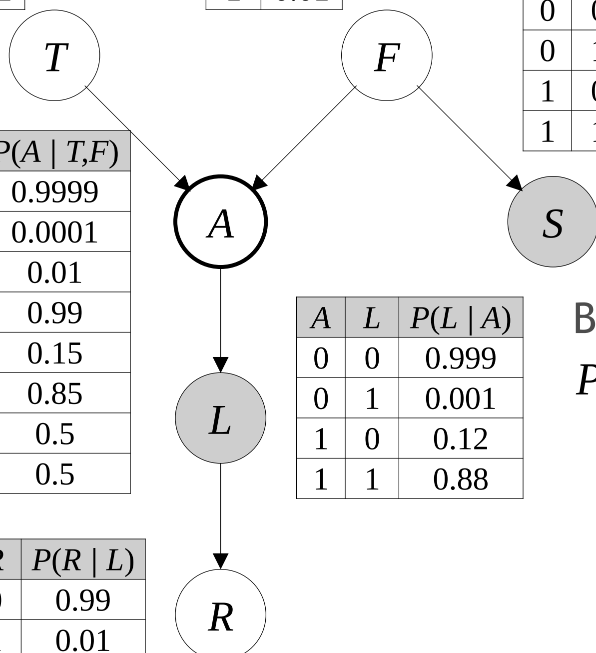
- Step 5

  Computing the answer

T	P(T)
0	0.98
1	0.02

F	P(F)
0	0.99
1	0.01

F	S	P(S | F)
0	0	0.99
0	1	0.01
1	0	0.1
1	1	0.9



T	F	A	P(A | T,F)
0	0	0	0.9999
0	0	1	0.0001
0	1	0	0.01
0	1	1	0.99
1	0	0	0.15
1	0	1	0.85
1	1	0	0.5
1	1	1	0.5

A	L	P(L | A)
0	0	0.999
0	1	0.001
1	0	0.12
1	1	0.88

L	R	P(R | L)
0	0	0.99
0	1	0.01
1	0	0.25
1	1	0.75

By rewriting:

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

This sum has value 1
This is not surprising
given that <$A \perp R \mid L$>

Probabilistic inference

- Step 5

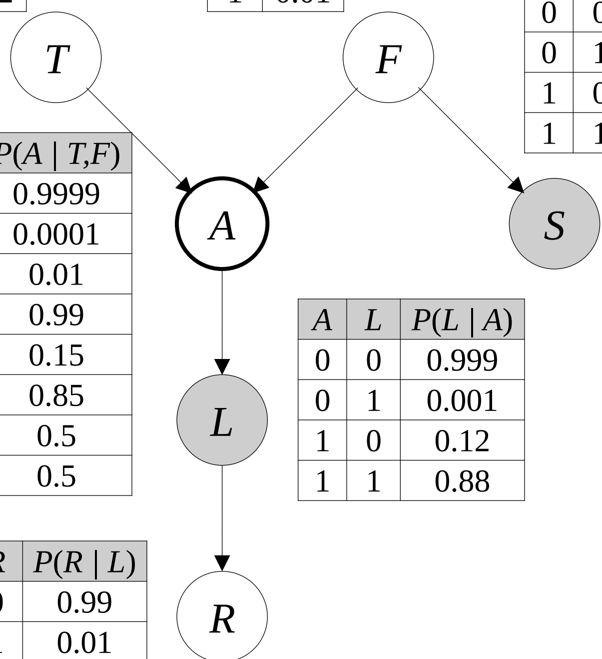
  Computing the answer

T	$P(T)$
0	0.98
1	0.02

F	$P(F)$
0	0.99
1	0.01

F	S	$P(S \mid F)$
0	0	0.99
0	1	0.01
1	0	0.1
1	1	0.9

T	F	A	$P(A \mid T,F)$
0	0	0	0.9999
0	0	1	0.0001
0	1	0	0.01
0	1	1	0.99
1	0	0	0.15
1	0	1	0.85
1	1	0	0.5
1	1	1	0.5



A	L	$P(L \mid A)$
0	0	0.999
0	1	0.001
1	0	0.12
1	1	0.88

L	R	$P(R \mid L)$
0	0	0.99
0	1	0.01
1	0	0.25
1	1	0.75

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*.

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

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

In general:
By summing w.r.t. a condition*ed* variable we obtain a marginal probability

$$P(A|S) = \sum_L P(A,L|S)$$

By summing w.r.t. a condition*ing* variable we just obtain a function

$$f_S(A,L) = \sum_S P(A,L|S)$$

Probabilistic inference

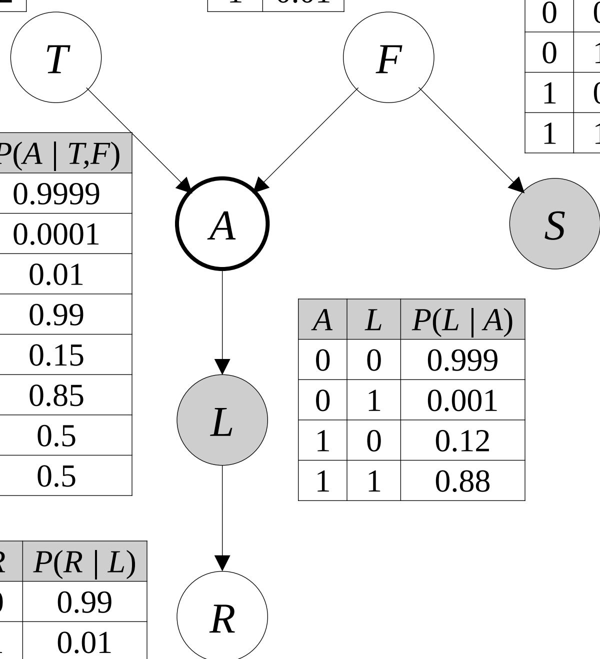
- Step 5

  Computing the answer

T	$P(T)$
0	0.98
1	0.02

F	$P(F)$
0	0.99
1	0.01

F	S	$P(S \mid F)$
0	0	0.99
0	1	0.01
1	0	0.1
1	1	0.9



T	F	A	$P(A \mid T,F)$
0	0	0	0.9999
0	0	1	0.0001
0	1	0	0.01
0	1	1	0.99
1	0	0	0.15
1	0	1	0.85
1	1	0	0.5
1	1	1	0.5

A	L	$P(L \mid A)$
0	0	0.999
0	1	0.001
1	0	0.12
1	1	0.88

L	R	$P(R \mid L)$
0	0	0.99
0	1	0.01
1	0	0.25
1	1	0.75

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 joint probability of the query in the form:

$$P(\{X_f\},\{X_e\}) = \sum_{\{X_r\}} \prod_{X_i} P(X_i \mid parents(X_i))$$

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 $P(\{X_f\},\{X_e\})$
5) Wrap it up and compute the response:

$$P(\{X_f\}|\{X_e\}) = \frac{P(\{X_f\},\{X_e\})}{\sum_{\{X_f\}} P(\{X_f\},\{X_e\})}$$

Remember: the method is NP-complete (anyway)

Graphical models as a probabilistic method

- *Advantages*

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

  Correctness (of representation) $\langle \{X\} \perp \{Y\} \mid \{Z\} \rangle_{GM} \Rightarrow \langle \{X\} \perp \{Y\} \mid \{Z\} \rangle_{JPD}$

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

  Not all JPDs can be faithfully represented by a graph model

  - *Counterexample: no DAG can represent*

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

    without introducing some further independence relation
    (*no closure under marginalization* - see also https://projecteuclid.org/download/pdf_1/euclid.aos/1031689015)