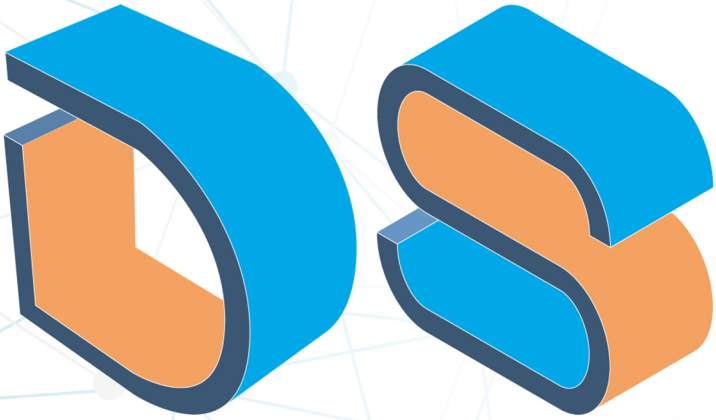


POLIMI
DATA SCIENTISTS

# Uncertainty in AI

## Course Notes

*Edited by:*
***Roberto Spatafora***
***Filippo Calió***


This course notes have been developed by **Polimi Data Scientist** staff.

Are you interested in **Data Science** initiatives at PoliMi?

follow
**Polimi Data Scientist**
on Facebook!

# Credits

The following notes have been written by Filippo Caliò in collaboration with the Polimi Data Scientists student association by combining Prof. Bonarini and Matteucci's lectures, the slides of the course and notes.

They are meant as a support for the students following the course and they should not be considered as a replacement for the professor's lectures or the book suggested in the course bibliography.

This document has been written by Roberto Spatafora

# Contents


**I Fuzzy Systems** 4

**1 Introduction** 5
- 1.1 Types of uncertainty . . . 5
- 1.2 Sources of uncertainty . . . 5

**2 Uncertainty models in AI** 6
- 2.1 Type of models . . . 6
- 2.2 Causes of ignorance . . . 6

**3 Fuzzy Sets** 8

**4 Fuzzy Logic** 11
- 4.1 Fuzzy modifiers . . . 11

**5 Fuzzy Rules** 12

**6 Fuzzy Design** 13

**7 Application of Fuzzy Systems** 14
- 7.1 Why use fuzzy control? . . . 14

**8 Fuzzy Measures** 15
- 8.1 Belief . . . 15
- 8.2 Plausibility . . . 16
- 8.3 Basic probability assignment . . . 16
- 8.4 Evidence Combination . . . 17
- 8.5 Possibility measures . . . 17
- 8.6 Possibility on fuzzy sets . . . 17
- 8.7 Necessity . . . 18
- 8.8 Confirmation Degree . . . 18
- 8.9 Fuzzyness measures . . . 18

**9 Fuzzy math** 19
- 9.1 Fuzzy numbers . . . 19
- 9.2 The four main operations . . . 19

**10 Evidence Theory** 21
- 10.1 Conflict . . . 21
- 10.2 Consonance . . . 21
- 10.3 Arbitrary . . . 22
- 10.4 Consistent . . . 22
- 10.5 Dempster rule . . . 22
- 10.6 Yager's combination rule . . . 23

**II Graphical Models** **24**

**11 Probabilistic Reasoning** **25**
11.1 Density Estimation . . . 26
11.1.1 Joint Distribution . . . 26
11.2 Learning a Naive Bayes Estimator . . . 28
11.3 Joint Density vs Niave Density . . . 28
11.4 Learning a Naive Bayes Estimator . . . 28

**12 Bayesian Networks** **29**
12.1 Independencies . . . 32
12.2 Conditional Independence in Bayesian Networks . . . 32
12.3 Bayesian Networks Wrap Up . . . 34
12.4 Naive Bayesian Classifier . . . 35

**13 Bayesian Network Inference** **36**
13.1 Variable Elimination . . . 36
13.2 Factor Graph . . . 38
13.3 The Chain: from directed to undirected graph . . . 40
13.4 Factor Graphs are not unique . . . 41
13.5 Polytree Exmaple . . . 41
13.5.1 Variable Elimination Algorithm . . . 42
13.6 Belief Propagation . . . 49
13.7 Belief Propagation in Trees . . . 50
13.8 Belief Propagation Update Equations . . . 58
13.9 Junction Tree . . . 59
13.10Sampling Based Methods . . . 63
13.11Rejection Sampling . . . 64
13.12Sampling . . . 65

**14 Learning Bayesan Networks** **67**

**15 Dynamic Bayesan Network** **68**
15.1 Probabilistic Reasoning for Time Series . . . 68
15.2 Markov Chains . . . 68
15.3 A-periodic Markov Chains . . . 71
15.4 Steady State Distribution . . . 71
15.5 Transitory Behaviour . . . 72
15.6 Dealing with Absoring States . . . 72
15.6.1 Inference in Absorbing Markov Chains . . . 72
15.7 Hidden Markov Models . . . 74
15.8 Forward Probability . . . 74
15.9 Viterbi Algorithm . . . 75

# Part I

# Fuzzy Systems

# Chapter 1

# Introduction

Uncertainty refers to epistemic situations involving imperfect or unknown information. It applies to predictions of future events, to physical measurements that are already made, or to the unknown.

Uncertainty is related to the *modelling: the need of describing a piece of reality.* Modelling is the way we may represent entities in a computer and making it reasoning on them

- A model is a representation of some entity, defined for a specific purpose
- A model captures only the aspects of the entity modelled that are relevant for the purpose

Examples of model: an image, a deep neural network, a fuzzy control system

## 1.1 Types of uncertainty

- Epimestic: due to things one could in principle know but does not in practice
- Aleatoric/Statistical: representative of unknowns that differ each time we run the same experiment

## 1.2 Sources of uncertainty

- Parameter uncertainty: from the **model parameters, whose exact values are unknown to experimentalists** and cannot be controlled in experiments, or whose values cannot be inferred by statistical methods.
- Parametric variability: from the **variability of input variables** of the model.
- Structural uncertainty *(model inadequacy, model bias, or model discrepancy)*: from the **lack of knowledge of the problem**.
- Algorithmic uncertainty *(numerical uncertainty, or discrete uncertainty)*: from **numerical errors and numerical approximations** in the implementation of the computer model.
- Experimental uncertainty *(observation error):* from the **variability of experimental measurements**
- Interpolation uncertainty: from a **lack of available data collected** from computer model simulations and/or experimental measurements

The type of uncertainty model depends on the **type** of uncertainty, its **sources** and the **information** we have on uncertainty and mostly has to do with some kind of **qualification** and **quantification** of uncertainty:

- Statistical models
- Logical models
- Cognitive models

# Chapter 2

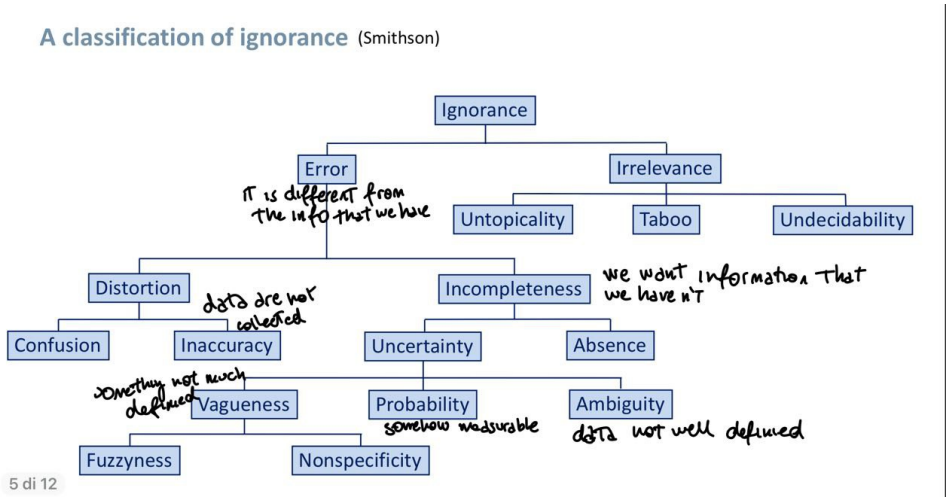
# Uncertainty models in AI

## 2.1 Type of models

- **Symbolic models**: elements of the models are expressed as terms related to entities to be modeled. The state of the world is represented by facts expressed in formal languages close to natural language.
  We assume a fact as true in a model if it is possible to collect enough evidence to support it
- **Sub-symbolic models**: elements of the models are expressed by codes
- **Black-box models**: the model can be computed, but it is only regarded as a computational way to map inputs to outputs

## 2.2 Causes of ignorance

- **Insufficient data**: I cannot access to a feature required by the model
- **Biased data**: data are collected by sensors affected by errors
- **Variable data**: data are collected by imprecise sensors
- **Reliability of data**, possibly coming from unreliable sources, data may be wrong
- **Fuzzyness**: we cannot have more than a qualitative estimation about a statement
- **Reliability of the model**: depends on the model design, building and parametrization
- **Incompleteness of the model**: the model does not include some relevant features



**Probability** is represented by **numbers** between 0 and 1, and a well-established set of rules and properties are associated to its management, among which, given a set of alternative hypotheses:

- the sum of their probabulities should be 1
- the probability a posteriori of a hypothesis hi given some evidence e is given by the Bayes theorem

$$P(h_i \mid e) = \frac{P(e \mid h_i) \cdot P(h_i))}{P(e)}$$

It is possible to write rules, most often represented as networks of probabilistic relationships among statements, known as **Bayesian networks** or **Graphic models**.

# Chapter 3

# Fuzzy Sets

They are a tool to model approximate concepts.

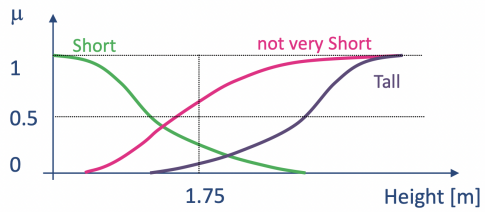
**Crisp set**: defined by a *boolean membership function* on some property of the considered elements. The membership function *returns true or false*.

**Fuzzy set**: a *set whose membership function ranges on the interval [0, 1] and the function does not return only true or false*. Fuzzy sets enable a smoother transition when labeling a value, instead of intervals that are rectangular. It is possible to define fuzzy sets also for variables with **discrete** values.

A *membership function (MF) defines a set, by defining the degree of membership of an element* of the universe of discourse to the set.

MFs define fuzzy sets. *Label*: a name is given to the set to make it possible to refer to it and it denotes fuzzy sets

A person 1.75m high belongs to:
- Short with membership 0.3
- Tall with membership 0.2
- not very Short with membership 0.6




A set of fuzzy sets fully covering the universe of discourse (the range of a variable), is called **frame of cognition**.
Properties of a frame of cognition:

- **Coverage**: each element of the universe of discourse is assigned to at least one granule with membership $> 0$

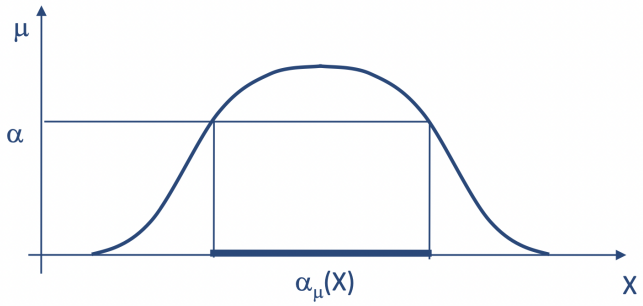
- **Unimodality of fuzzy sets**: there is a unique set of values for each granule with maximum membership

**Fuzzy partition**: frame of cognition for which the sum of the membership values of each value of the base variable is equal to 1.

## AlphaCut

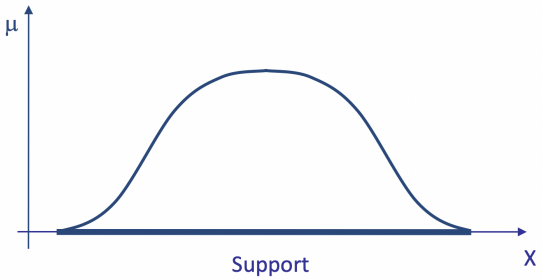
The $\alpha$-cut of a fuzzy set is the crisp set of the values of x such that $\mu(x) \geq \alpha$

$$\alpha_\mu(X) = \{x \mid \mu(x) \geq \alpha\}$$




## Support

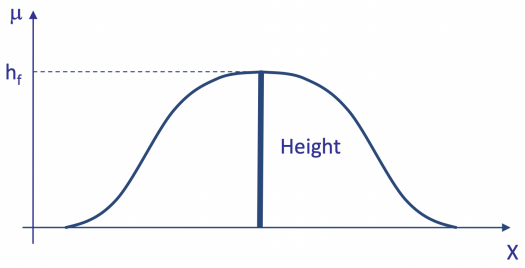
The crisp set of values x of X such that $\mu_f(x) > 0$ is the support of the fuzzy set f on the universe X




## Height

The **height** $h_f$ of a fuzzy set f on the universe X is the highest membership degree of an element of X to the fuzzy set

$h_f(X) = max_{x \in X}\ \mu_f(x)$



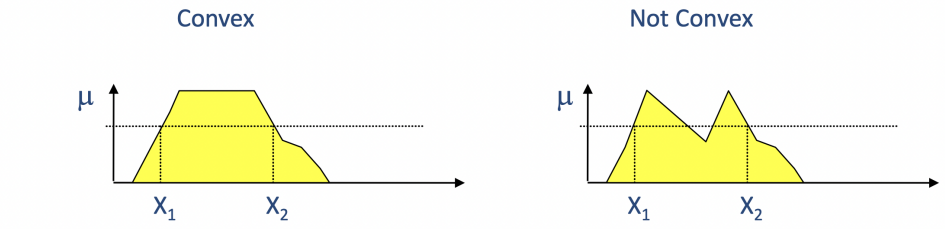

A fuzzy set f is **normal** iff $h_f(X)=1$

**Convex**

A fuzzy set is *convex* iff

$$\mu(\lambda x_1 + (1-\lambda)\, x_2) \geq \min[\mu(x_1), \mu(x_2)]$$

for any $x_1, x_2$ in $\Re$ and any $\lambda$ belonging to [0,1]




Singleton: fuzzy set with one member Interval: all of the members of the interval have the $MF = 1$

Complement

$$\mu_{\bar{f}}(x) = 1 - \mu_f(x)$$

Union

$$\mu_{f_1 \cup f_2}(x) = \max\left[\mu_{f_1}(x), \mu_{f_2}(x)\right]$$

Intersection

$$\mu_{f_1 \cap f_2}(x) = \min\left[\mu_{f_1}(x), \mu_{f_2}(x)\right]$$

The intersection operator is defined as a T-norm:

$$T_\alpha(a,b) = \frac{ab}{max[a,b,\alpha]}$$

for $\alpha$=1 we have *ab*

for $\alpha$=0 we have *min(a, b)*

Union and S-norms (or T-conorms):

$$\mu_{A \cup B}(x) = u[\mu_A(x), \mu_B(x)]$$

**Aggregation** is the *operator that aggregates the values of membership for the same fuzzy set, coming from different knowledge sources.* It is used in Fuzzy Rule Systems.

$$\mu_A(x) = h[\mu_{A1}(x), ..., \mu_{An}(x)]$$

# Chapter 4

# Fuzzy Logic

**Logic is a tool used to formally represent knowledge**. There are many types of logic:

- **Propositional logic**: truth values for propositions (e.g., The grass is green) is true.
- **First order predicate logic**: truth values for predicates. There are variables and quantifiers (e.g., $\exists X : Green\, X\, is\, true$)
- **Second order predicate logic**: predicates of predicates (e.g., $\exists X, Y : (Believe\, Y\, (Green\, X\, is\, true))\, is\, false$)

A logic *is truth functional* if the truth value of a compound sentence depends only on the truth values of the constituent atomic sentences, not on their meaning or structure. A **predicate** is a feature of language that can be used to make a statement about something, (to attribute a property to that thing). **Tautologies** are true by definition, and are used to prove theorems, so to prove the truth of an inferential chain.

Fuzzy logic is an infinite-valued logic, with truth values in [0..1] Propositions are expressed as A is L where:

- A is linguistic variable
- L is a label denoting a fuzzy set

A linguistic variable is defined by a 5-tuple: (X, T(X), U, G, M)
X = name of the variable
T(X) = set of terms for X (linguistic values), each corresponding to a fuzzy variable denoted by T(X) and ranging on U
U = universe of discourse defined on a base variable u
G = syntactic rule to generate the interpretation X of each value u
M = semantic rule to associate to X its meaning

Degrees of truth are often confused with probabilities. They are conceptually distinct:
fuzzy truth represents membership in vaguely defined sets, not likelihood of some event or condition.

## 4.1 Fuzzy modifiers

Fuzzy modifiers modify truth values

- **Strong modifiers** make the predicate stronger, so they reduce the truth of the proposition
- **Weak modifiers** make the predicate weaker, so they increase the truth of the proposition

# Chapter 5

# Fuzzy Rules

A fuzzy rule is a rule whose clauses have the shape (V is L), where V is a linguistic variable and L is a label, a value for V associated to a fuzzy set. This is a linguistic clause.
Linguistic rules: the consequent is a conjunction of linguistic clauses.
Model rules: bind a model (linear, non linear, NN, ...) to the linguistic interpretation of its applicability conditions

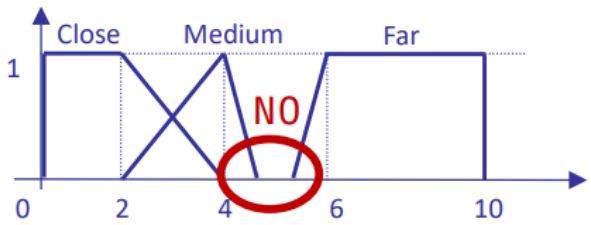
**Defuzification** is the process of producing a quantifiable result in crisp logic, given fuzzy sets and corresponding membership degrees. *It is the process that maps a fuzzy set to a crisp set.* It is typically needed in fuzzy control systems. These systems will have a number of rules that transform a number of variables into a fuzzy result, that is, the result is described in terms of membership in fuzzy sets.
We can also give weights to rules, to define the relative relevance of their contribution to the final result.
We use the operator **max** to aggregate weights given to the same output value.

# Chapter 6

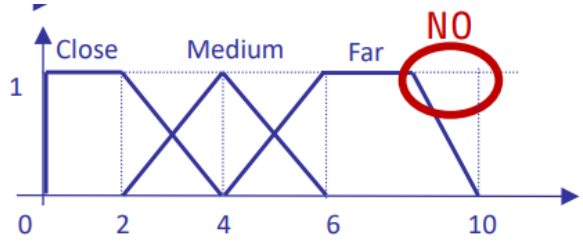
# Fuzzy Design

- Input variables: numerical or ordinal variables to define fuzzy sets on them The variables can come directly from sensors/users or computed from perceived variables (error/derivatives). Also non ordinal variables could be integrated in a fuzzy system (colors of traffic lights)
- Output variables: results of the model
- Goal of the fuzzy models: depend on the specification

MFs are defined by an objective evaluation, by interviews, by a probabilistic elaboration. In general we have from 3 to 7 membership functions for each variable.
Any point in the range of input variables has to be covered by at least one fuzzy set.




Boundary should be covered with maximum value



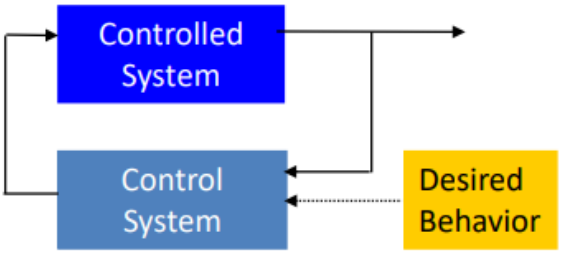

The MFs can be evenly distributed (max robustness to noise) or unevenly distributed (more precision where needed).

- AND of antecedent clauses
  - min: the worst degree of matching is the most relevant -> good for critical systems
  - product: all the degrees of matching are relevant -> good for evaluation systems, for instance
- Detachment: combination with the rule weight
  - min
  - product
- Aggregation of degrees of the same consequent
  - max: the best degree is the most relevant
  - probabilistic sum: all the collected knowledge is considered

# Chapter 7

# Application of Fuzzy Systems

A control system is a system able to control the behaviour of another system (a device, a community, ...)




In most cases it is a PID controller, where the output **u** depends on the difference ***e*** between the desired, and the observed behavior, its derivative (how fast ***e*** changes) and its integral (how large ***e*** has been in the past):

$$u = K_p e + K_D \frac{de}{dt} + K_I \frac{1}{T} \int_0^t e\, dt$$

A fuzzy control system has robustness with the respect to noise, has control rules, has smoothness of action and it is not linearity.

## 7.1 Why use fuzzy control?

- when there is need to adapt
- low quality of sensor data
- to simplify the interaction with the user
- hard to define and parametrize a mathematical model
- very noise interpretation of sensor data
- approximate and uncertain data

# Chapter 8

# Fuzzy Measures

It is a function that is monotene, continue or semicontinue.

A function g defined on a Borel field $\mathcal{B}$ of the universe of discourse X is a *fuzzy measure* if it has the following properties:

1. $g(\varnothing) = 0,\ g(X) = 1$
2. $if\ A, B \in \mathcal{B}\ and A \subseteq B,\ then\ g(A) \leq g(B)$
3. $if\ A_n \in \mathcal{B}, A_1 \subseteq A_2 \subseteq ...\ then\ \lim_{n\to\infty} g(A_n) = g(\lim_{n\to\infty} A_n)$

It is different from classical measures, since the additivity property is relaxed.

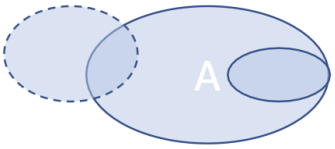
If a field has the property that, if sets $A_1$, ... $A_n$ belong to the field, then also union and intersection of the sets belong to the field, this is named as a **Borel field**.

We would like to define a measure of evidence for or against a proposition: **belief** and **plausibility**,

## 8.1 Belief

It is a function and an estimation of the **minimum** probability that can be assigned to an element, given the collected evidence.

$$Bel(A) = \Sigma_{B|B\subseteq A}\, m(B)$$




$Bel : \wp(X) \rightarrow [0, 1]$

$Bel(\emptyset) = 0,\ Bel(X) = 1$

$Bel(A_1 \cup A_2 \cup ... \cup A_n) \geq \Sigma_j\, Bel(A_j) - \Sigma_{j<k}\, Bel(A_j \cap A_k) + \quad + ... + (-1)^{n+1}\, Bel(A_1 \cap A_2 \cap ... \cap A_n)$

$Bel(A) + Bel(\neg A) \leq 1$

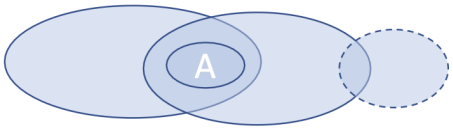
Here we have ≤ because, since A is a fuzzy set, in general A intersects ¬A, so the second term of the above reported equation is non null. Since $A \cup \neg A = X$ and Bel (X) = 1 we have the mentioned relationship.

If they do not intersect (crisp set) we only have =

## 8.2 Plausibility

It is a function and an estimation of the **maximum** probability that can be assigned to an element, given the collected evidence. It is bigger/larger than the belief.

$Pl(A) = \Sigma_{B \mid B \cap A \neq \emptyset} m(B)$




$Pl : \wp(X) \rightarrow [0, 1]$

$Pl(\emptyset) = 0, Pl(X) = 1$

$Pl(A_1 \cap A_2 \cap \ldots \cap A_n) \geq \Sigma_j Pl(A_j) - \Sigma_{j<k} Pl(A_j \cup A_k) + \ldots + (-1)^{n+1} Pl(A_1 \cup A_2 \cup \ldots \cup A_n)$

$Pl(A) + Pl(\neg A) \geq 1$

In the last equations we have ≥ and not only =, since A is a fuzzy set, so A may intersect ¬A. Even when A has maximum plausibility, Pl(A) = 1, the part of ¬A that intersects A may have anyway non null plausibility , so Pl(¬A) > 0.

## 8.3 Basic probability assignment

It is a function. When we consider the empty set, the function returns zero

$m: \wp(X) \rightarrow [0, 1]$

$m(\emptyset) = 0$

$\Sigma_{A \in \wp(X)} m(A)=1$

m gives, for any set A belonging to the power set of X ($\wp(X)$), how much the available and relevant evidence supports the fact that a given element belongs to A

Note that:

1. It is not needed that m(X) = 1
2. It is not needed that $m(A) \leq m(B)$ when $A \subseteq B$
3. No relationship holds between m(A) e m(¬A)

## Relationship between m, Bel, PI

$$Bel(A) = \Sigma_{B|B\subseteq A}\, m(B)$$

$$Pl(A) = \Sigma_{B|B\cap A\neq \emptyset}\, m(B)$$

$$m(A) = \Sigma_{B|B\subseteq A}\ (-1)^{|A-B|}\ Bel(B)$$

$$Pl(A) \geq Bel(A)$$

# 8.4 Evidence Combination

Evidence provided by different sources can be combined by:

$$m_{1,2}(A) = \frac{\sum_{B\cap C=A} m_1(B)\cdot m_2(C)}{1-K}$$

$$K = \sum_{B\cap C=O} m_1(B)\cdot m_2(C)$$

for any $A \neq \emptyset$ and $m_{1,2}(\emptyset) = 0$

These measures can be used to model uncertainty when it is possible to assign intervals of probabilities (or fuzzy models) to different subsets of a set of hypothesis, possibly coming from different sources, either humans or other AI systems.

# 8.5 Possibility measures

$\wp(X)$ is the power set of the set X

A possibility measure is given by the equation $\Pi$: $\wp(X) \rightarrow [0,1]$, for which the following properties hold:

1. $\Pi(\emptyset)=0$, $\Pi(X)=1$
2. $A \subseteq B \Rightarrow \Pi(A) \leq \Pi(B)$
3. $\Pi(A) = \sup_{x\in A} f(x), A \subset X$

It can be univoquely defined by a possibility relationship f: X → [0,1] so that

$$\Pi(\bigcup_{i\in I} A_i) = \sup_{i\in I} \Pi(A_i)$$

Therefore, f is defined as $\Pi(\{x\})$, $\forall\ x \in X$

# 8.6 Possibility on fuzzy sets

If A is a fuzzy set defined on the universe of discourse U, and $\Pi_x(A)$ is a possibility distribution associated to a variable x ranging in U then the measure of possibility of A is defined as:

$$poss\{x\ is\ A\} \equiv \prod_x(A) \equiv sup_{u\in U} min\{\mu_A(u),\ \Pi_x(u)\}$$

## 8.7 Necessity

Necessity is the dual of possibility
$\Pi(A) = 1 - N(\neg A)$
It satisfies the condition $min(N(A), N(\neg A)) = 0$.
Moreover: $\Pi(A) \geq N(A)$
$N(A) > 0 \Rightarrow \Pi(A) = 1$
$\Pi(A) < 1 \Rightarrow N(A) = 0$

## 8.8 Confirmation Degree

It is possible to put together possibility and necessity in a unique *confirmation degree C*, given by:
$C(A) = N(A) + \Pi(A) - 1$
Negative values of C(A) correspond to a disconfirmation degree.

## 8.9 Fuzzyness measures

They provide the fuzzyness degree of a fuzzy set. A fuzzyness measure is the entropy of a fuzzy set.
Given a fuzzy set $A = \{x, \mu_A(x)\}$
$d(A) = K \sum_{i=1}^{n} S(\mu_A(x_i))$
Where S(x) is the Shannon's function $S(x) = -x \ln x - (1 - x) \ln(1 - x)$

# Chapter 9

# Fuzzy math

## 9.1 Fuzzy numbers

- Used to represent approximation
- they are fuzzy sets defined over the set of real numbers, which model our concept of approximate value

They are based on two properties:

1. Each fuzzy number can be completely represented by its alfa-cuts, in a unique way
2. The alfa-cuts of fuzzy numbers are closed intervals of real numbers

**Constraints**

1. Normal fuzzy sets → captures the concept of approximate value corresponding to a number
2. Convex fuzzy sets (all alpha-cut intervals should be closed)
3. The support of A should be bounded

Note that constraints 2 and 3 are needed to define arithmetic.

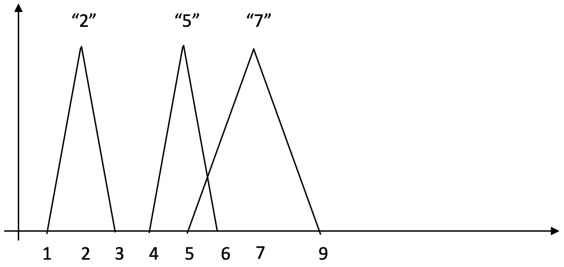
## 9.2 The four main operations

$$[a,\, b] + [d,\, e] = [a + d,\, b + e]$$
$$[a,\, b] - [d,\, e] = [a - d,\, b - e]$$
$$[a,\, b] * [d,\, e] = [\min(ad,\, ae,\, bd,\, be),\, \max(ad,\, ae,\, bd,\, be)]$$

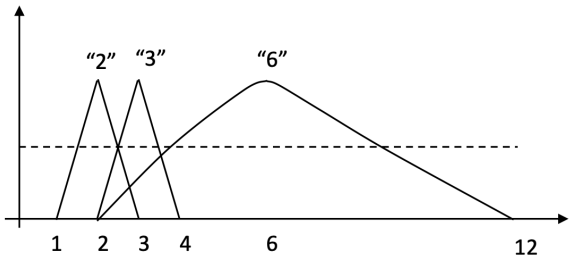
and $if\, 0 \notin [d,\, e]$

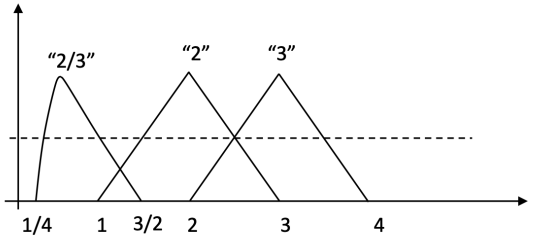
$$[a,\, b]/[d,\, e] = [a,\, b] * [\frac{1}{d},\, \frac{1}{e}] = [\min(a/d,\, a/e,\, b/d,\, b/e),\, \max(a/d,\, a/e,\, b/d,\, b/e)]$$

**Sum**



**Product**



# Chapter 10

# Evidence Theory

Evidence they should be used when we have multiple sources of knowledge (experts, sensors, AI systems,...) and when the basic probability assignment is distributed on different sets of statements, or intervals.

For examples, different sensors assign a target in different locations, or different targets in the same location.

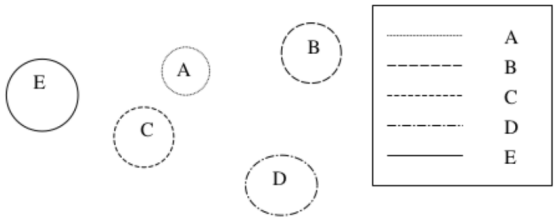
## Implications

1. It is **not necessary to explicit a precise measurement** from a knowledge source or an experiment if it is not realistic or feasible to do so.
2. The **Principle of Insufficient Reason** (e.g., distribute the probability among the elements of a set if it is not possible to distinguish among them) **is not imposed**. Statements can be made about the likelihood of multiple events together without having to resort to assumptions about the probabilities of the individual events under ignorance.
3. The axiom of **additivity is not imposed**. The measures do not have to add to 1. When they do, it corresponds to a traditional probabilistic representation. When the sum is less than 1, called the subadditive case, this implies an incompatibility between multiple sources of information providing conflicting information. When the sum is greater than 1, the superadditive case, this implies a cooperative effect between multiple sources of information, e.g., multiple sensors providing the same information.

A, B, C, D, E are different sources of information

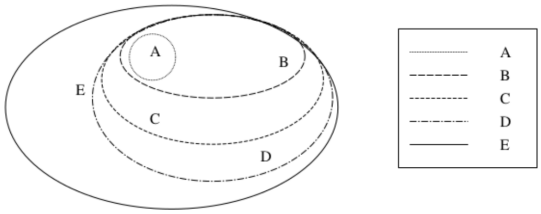
## 10.1 Conflict

Each source provides evidence for disjoint sets



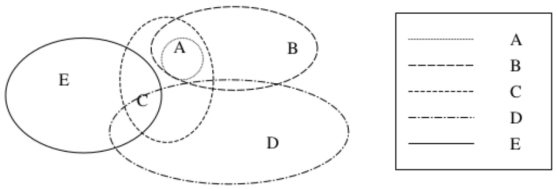

## 10.2 Consonance

Sources provide some evidence on nested sets converging on the target (here A)



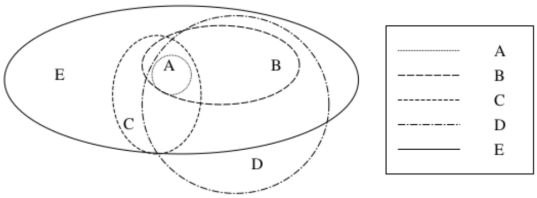

## 10.3 Arbitrary

Each source provides evidence for sets, only some of which include the target




## 10.4 Consistent

All sources provide some evidence for sets that include the same hypothesis (here A)




## 10.5 Dempster rule

$$m_{1,2}(A) = \frac{\sum_{B \cap C = A} m_1(B) \cdot m_2(C)}{1 - K} \qquad K = \sum_{B \cap C = O} m_1(B) \cdot m_2(C)$$

K is the basic probability mass associated with conflict. Its role in the denominator has the effect of completely ignoring conflict and attributing any probability mass associated with conflict to the null set.

**Example**

For instance, let's assume that two experts give this diagnosis of a detected failure.

Expert 1:

m 1 (A) = 0.99 (failure due to Component A)

m 1 (B) = 0.01 (failure due to Component B)

Expert 2:

m 2 (B) = 0.01 (failure due to Component B)

m 2 (C) = 0.99 (failure due to Component C)

We would like to decide which component is faulty

| | | | **Expert 1** | | | |
|---|---|---|---|---|---|---|
| | | | **A** | **B** | **C** | Failure Cause |
| | | | 0.99 | 0.01 | 0 | $m_1$ |
| **Expert 2** | Failure Cause | $m_2$ | | | | |
| | **A** | 0 | $m_1(A)\ m_2(A) = 0$ | $m_1(B)\ m_2(A) = 0$ | $m_1(C)\ m_2(A) = 0$ | |
| | **B** | 0.01 | $m_1(A)\ m_2(B) = 0.0099$ | $m_1(B)\ m_2(B) = 0.0001$ | $m_1(C)\ m_2(B) = 0$ | |
| | **C** | 0.99 | $m_1(A)\ m_2(C) = 0.9801$ | $m_1(B)\ m_2(C) = 0.0099$ | $m_1(C)\ m_2(C) = 0$ | |

From the table, we notice that the bpm of conflicting situations is high.

By computing the composite bpm of the only non conflicting solution we obtain, with the Dempster combination rule, that

$$K = (0.99)(0.01) + (0.99)(0.01) + (0.99)(0.99) = 0.9999$$

$$m_1(B) \cdot m_2(B) = (0.01)(0.01)/(1 - 0.9999) = 1$$

So, having associated all the conflict to the null set, it turns out that the solution with the worst bpm for each of the two experts is used to say that the two agree and the final, selected hypothesis is B, with bpm=1. This is counter intuitive, and some solutions have been proposed to amend this issue.

## 10.6 Yager's combination rule

Yager defines a ground probability mass assignment (q) as:

$$q(A) = \sum_{B \cap C = A} m_1(B) \cdot m_2(C)$$

$q(\varnothing) \geq 0$, where $q(\varnothing)$ is a measure of conflict.

The basic probability assignment of the universal set $m_Y(X)$ is defined as: $m(X) = q(X) + q(\varnothing)$

Conflict is assigned to the bpa of the universal set. The interpretation of the mass of the universal set X is the **degree of ignorance**.

In the example, we have:
$q_{12}(B) = m_{12}(B) = (0.01)(0.01) = 0.0001 = Bel(B)$ not including conflict (left to X), but much lower than each expert's estimations for B.

# Part II

# Graphical Models

# Chapter 11

# Probabilistic Reasoning

A is a *boolean-valued random variable* if A denotes an event and there is some degree of uncertainty as to whetever A occurs.

**Multivalued random variable**: random variable of arity k if it can take on exactly one values out of v1, v2, ... , vk

$$P(A = v_i \wedge A = v_j) = 0 \quad if \quad i \neq j$$
$$P(A = v_i \vee A = v_2 \vee A = v_3 \vee ... \vee A = v_k) = 1$$

Conditional Probability = Bayes Theorem

Chain rule: $P(A \wedge B) = P(A \mid B) * P(B)$
Bayes theorem: $P(A \mid B) = \frac{P(A \wedge B)}{P(B)} = \frac{P(B|A) * P(A)}{P(B)}$
Sum rule = $\sum_b P(A \wedge B = b) = P(A) \rightarrow P(A) = \sum_b P(A \wedge B = b)$

$$P(A \mid B) = \frac{P(B \mid A) * P(A)}{P(B|A) * P(A) + P(B|\bar{A}) * P(\bar{A})}$$

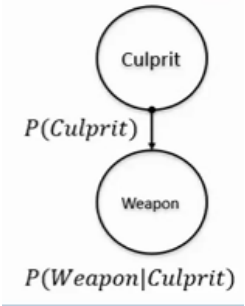
$$P(A \mid B \wedge X) = \frac{P(B \mid A \wedge X) * P(A \wedge X)}{P(B \wedge X)}$$

$$P(A = v_i \mid B) = \frac{P(B \mid A = v_i) * P(A = v_i)}{P(B)}$$

**Independent variables**

A and B are boolean random variables → they are independent iff $P(A \mid B) = P(A)$

The following image is a Bayesian Network:




Joint Distribution base on the chain rule:

| | Pistol | Knife | Poker |
|---|---|---|---|
| Cook | 4% | 52% | 24% |
| Butler | 16% | 2% | 2% |

Three types of graphs:

- directed graphs: for designing models → Bayesian Networks
- undirected graphs: computer vision
- factor graphs: for inferecne and learning

## 11.1 Density Estimation

### 11.1.1 Joint Distribution

All the variables depend on the other variables.
Given two random variables X and Y, the joint distribution of X and Y is the distribution of X and Y together: P(X,Y)

- - make a truth table with all the combinations
- for each combination compute the probability
- check that all probabilities sum up to 1



| A | B | C | P(A,B,C) |
|---|---|---|---|
| 0 | 0 | 0 | 0.30 |
| 0 | 0 | 1 | 0.05 |
| 0 | 1 | 0 | 0.10 |
| 0 | 1 | 1 | 0.05 |
| 1 | 0 | 0 | 0.05 |
| 1 | 0 | 1 | 0.10 |
| 1 | 1 | 0 | 0.25 |
| 1 | 1 | 1 | 0.10 |

Figure 11.1: P(A): sum of probabilities where A=1

Make inference = probabilistic reasoning → the process of reaching a conclusion about something from known facts or evidence

A Density Estimator is a function that predicts the probability of a certain combination and it is used to create Joint Distribution.

- build a joint distribution table for the attributes in which the probabilities are unspecified
- the fill in each row with $\hat{P}(row) = \frac{\#records\ matching\ row}{total\ number\ of\ records}$

If we do not have enough data to evaluate the table created, we use likelihood for evaluating density estimation.

- Given a record $\boldsymbol{x}$, a density estimator $\boldsymbol{M}$ tells you how likely it is

$$\hat{P}(x|M)$$

- Given a dataset with $\boldsymbol{N}$ records, a density estimator can tell how likely data is under the assumption that all records were independently generated from it

$$\hat{P}(dataset) = \hat{P}(x_1, x_2, \ldots, x_N) = \prod_{n=1}^{N} \hat{P}(x|M)$$

- Since likelihood can get too small we usually use log-likelihood:

$$log\hat{P}(dataset) = log \prod_{n=1}^{N} \hat{P}(x_n|M) = \sum_{n=1}^{N} log\hat{P}(x_n|M)$$

Density estimators can sort the records by probability, can do inference $P(E1 \mid E2)$, can be used for Bayes Classifiers. But they can badly overfit ($\rightarrow$ performing very bad in new data).

The Naive Bayes Estimator assumes that each attribute is distributed independently of any of the other attributes. All the variables are independent.

- Let $\boldsymbol{x[i]}$ denote the $\boldsymbol{i^{th}}$ field of record $\boldsymbol{x}$
- The Naïve Density Estimator says that:

$$x[i] \perp \{x[1], x[2], .., x[i-1], x[i+1], \ldots, x[I]\}$$

Suppose $A, B, C, D$ independently distributed, what is $P(A, \bar{B}, C, \bar{D})$ ?

$$\begin{aligned} P(A, \bar{B}, C, \bar{D}) &= P(A|\bar{B}, C, \bar{D}) * P(\bar{B}, C, \bar{D}) \\ &= P(A) * P(\bar{B}, C, \bar{D}) \\ &= P(A) * P(\bar{B}|C, \bar{D}) * P(C, \bar{D}) \\ &= P(A) * P(\bar{B}) * P(C, \bar{D}) \\ &= P(A) * P(\bar{B}) * P(C|\bar{D}) * P(\bar{D}) \\ &= P(A) * P(\bar{B}) * P(C) * P(\bar{D}) \end{aligned}$$

If the variables are independent, we obtain a chain rule.

## 11.2 Learning a Naive Bayes Estimator

Suppose $x[1], x[2], \ldots, x[I]$ are independently distributed:

- We can construct any row of the implied Joint Distribution on demand

$$\hat{P}(x[1] = u_1, x[2] = u_2, \ldots, x[I] = u_I) = \prod_{k=1}^{I} \hat{P}(x[k] = u_k)$$

- We can do any inference!

$$P(E_1|E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{row \sim E_1 \wedge E_2} P(row)}{\sum_{row \sim E_2} P(row)}$$

How do we learn a Naïve Density Estimator?

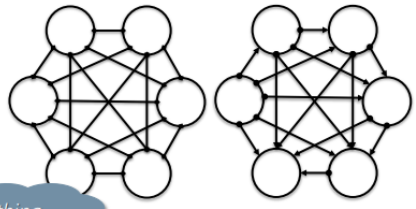
$$\hat{P}(x[i] = u) = \frac{\#\ records\ for\ which\ x[i] = u}{total\ number\ of\ records}$$

## 11.3 Joint Density vs Niave Density

## 11.4 Learning a Naive Bayes Estimator

Joint Distribution Estimator

- Can model anything
- Given 100 records and more than 6 Boolean attributes will perform poorly
- Can easily overfit the data



Is there anything in between?

Naïve Distribution Estimator

- Can model only very boring distributions
- Given 100 records and 10,000 multivalued attributes will be fine
- Quite robust to overfitting

Naive Distribution Estimator used as Naive Bayes Classifier.

# Chapter 12

# Bayesian Networks

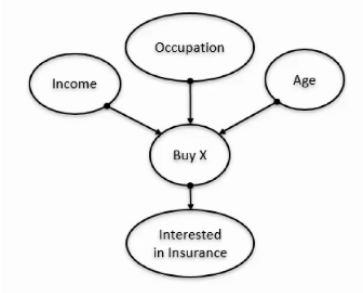
A *Bayesian Belief Networks*, or *Bayesian Network*, is a method to describe the joint probability distribution of a set of variables. There are no loops and it is composed by:

- - **DAG** (*Directed Acyclic Graph*): nodes represent random variables and edges represent direct influence
- **CPD** (*Conditional Probability Distributions*): influenced variables

A *Bayesian Network* is a representation of the joint probability distribution via explicit indication of conditional independences.

$x_1, ..., x_n$: set of variables of the Bayesian Network.
The full joint distribution require: $2^n - 1$ parameters.

**Example**




$P(Income)$, $P(Occupation)$, $P(Age)$, $P(Buy\,X \mid Income, Occupation, Age)$, $P(Interested\,in\,insurance \mid Buy\,X)$ Age, Occupation and Income determine if customer will buy the products.

The indipendences that there are in this graph are called *Conditionally Indipendent*.

We say $X_1$ is *Conditionally Independent* from $X_2$ given $X_3$ if the probability of $X_1$ is independent of $X_2$ given some knowledge about $X_3$:

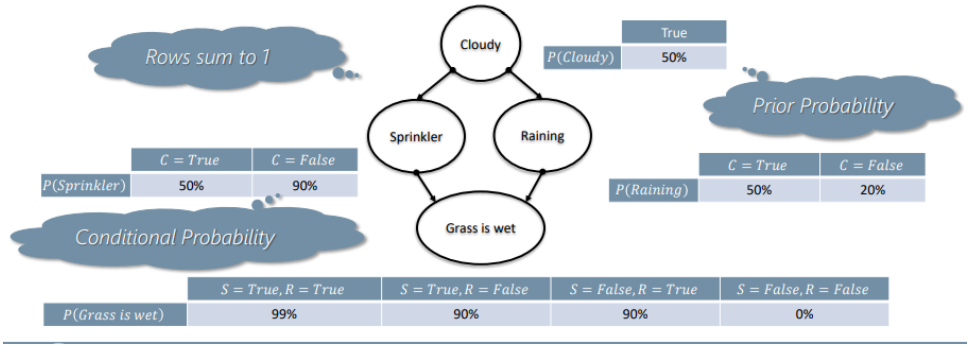
$$P(X_1 \mid X_2, X_3) = P(X_1 \mid X_3)$$

The same can be said for a set of variables: $X_1$, $X_2$, $X_3$ is independent from $Y_1$, $Y_2$, $Y_3$ given $Z_1$, $Z_2$, $Z_3$:

$$P(X_1, X_2, X_3 \mid Y_1, Y_2, Y_3, Z_1, Z_2, Z_3) = P(X_1, X_2, X_3 \mid Z_1, Z_2, Z_3)$$

## The Sprinkler Example: Modeling

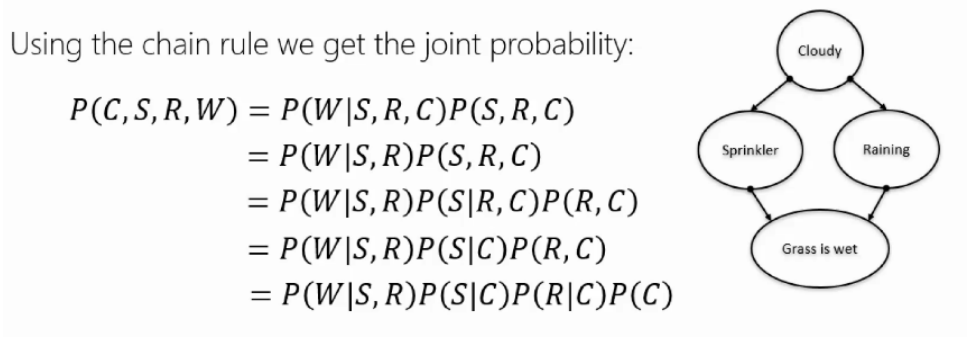
The event "Grass is wet" (W=true) has two possibility causes: either the water Sprinkler is on (S = true) or it is Raining (R = true)




With N binary nodes and k the maximum node fan-in, the full joint distribution requires $O(2^N)$ parameters while the factored one $O(N * Q^k)$, with Q: arity (2 for a binary variable)

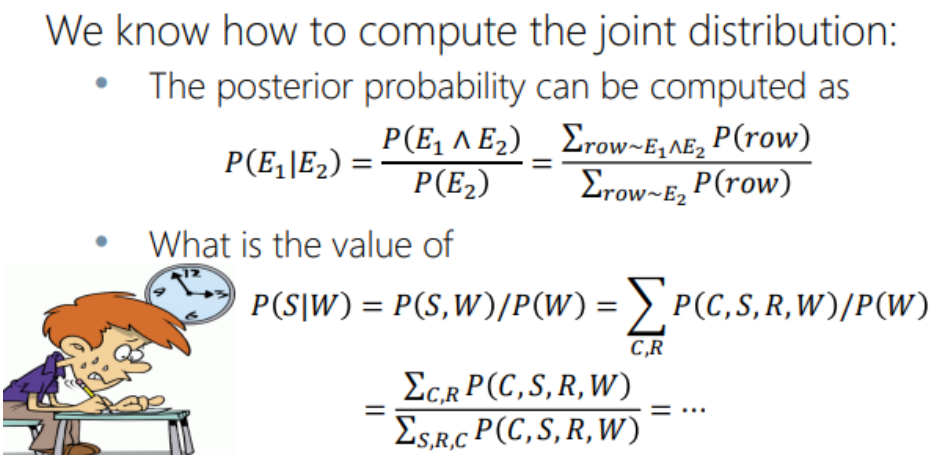
## The Sprinkler Example: Joint Probability

The simplest conditional independence: "a node is independent of its ancestors given its parents"




## The Sprinkler Example: Making Inference

We observe that the grass is wet. There are two possible causes for this: (a) sprinkler is on or (b) it is raining. Which is more likely?



| C | S | R | W | P(C,S,R,W) | P(C,S,R,W) |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0.5*0.1*0.8*1 | 0.04 |
| 0 | 0 | 0 | 1 | 0.5*0.1*0.8*0 | 0 |
| 0 | 0 | 1 | 0 | 0.5*0.1*0.2*0.1 | 0.001 |
| 0 | 0 | 1 | 1 | 0.5*0.1*0.2*0.9 | 0.009 |
| 0 | 1 | 0 | 0 | 0.5*0.9*0.8*0.1 | 0.036 |
| 0 | 1 | 0 | 1 | 0.5*0.9*0.8*0.9 | 0.324 |
| 0 | 1 | 1 | 0 | 0.5*0.9*0.2*0.01 | 0.0009 |
| 0 | 1 | 1 | 1 | 0.5*0.9*0.2*0.99 | 0.0891 |
| 1 | 0 | 0 | 0 | 0.5*0.5*0.5*1 | 0.125 |
| 1 | 0 | 0 | 1 | 0.5*0.5*0.5*0 | 0 |
| 1 | 0 | 1 | 0 | 0.5*0.5*0.5*0.1 | 0.0125 |
| 1 | 0 | 1 | 1 | 0.5*0.5*0.5*0.9 | 0.1125 |
| 1 | 1 | 0 | 0 | 0.5*0.5*0.5*0.1 | 0.0125 |
| 1 | 1 | 0 | 1 | 0.5*0.5*0.5*0.9 | 0.1125 |
| 1 | 1 | 1 | 0 | 0.5*0.5*0.5*0.01 | 0.00125 |
| 1 | 1 | 1 | 1 | 0.5*0.5*0.5*0.99 | 0.12375 |

Should sum up to 1!

$$P(S|W) = \frac{\sum_{C,R} P(C,S,R,W)}{\sum_{S,R,C} P(C,S,R,W)}$$

To compute P(S—W), we start from the numerator and we summarize the value of the rows considering:
C R 0 0 0 1 1 0 1 1 Same thing for the denominator, but now we have three terms → S R C (000, 001, 010,...)

## The Sprinkler Example: Explaining Away

In Sprinkler Example the two causes "compete" to "explain" the observed data. Hence $S$ and $R$ become conditionally dependent given that their common child, $W$, is observed.

Example: Suppose the grass is wet, but we know that it is raining. Then the posterior probability of sprinkler being on becomes:

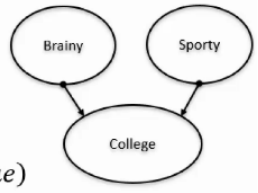
$$P(S \mid W, R) = \frac{P(S, W \mid R)}{P(W \mid R)} = \frac{\sum_C P(C, S, W \mid R)}{\sum_{S,C} P(C, S, W \mid R)}$$

numerator = probability of S, R and W equals to 1
denominator = probability of R and W equals to 1

The probability of Sprinkler goes down: $P(S \mid W, R) \leq P(S \mid W)$

This phenomenon is called Explaining Away: known as Berkson's Paradox or Selection Bias and it describes two variables which become dependent because you observe a third one.

*Example*: Consider a college which admits students who are either *Brainy* or *Sporty* (or both!). Let $C$ denote the event "admitted to College", which is **True** if a student is either Brainy ($B$) or Sporty ($S$).
Suppose in population, $B$ and $S$ are independent.
In *College*, being *Brainy* makes you less likely to be *Sporty*, because either are sufficient to explain $C$

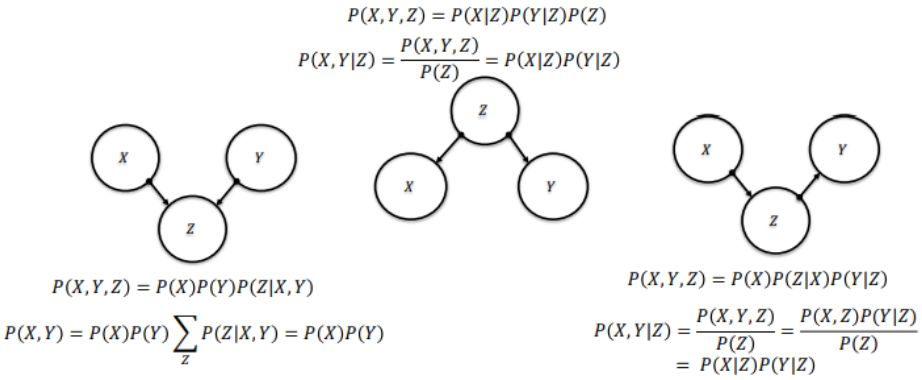



$$P(S = True|C = True, B = True) \leq P(S = True|C = True)$$

## Joint Distribution Factorization

$$P(X) = P(X_1, X_2, ..., X_N) = \prod_{k=1}^{N} P(X_k \mid parents(X_k)) = \prod_{k=1}^{N} P(X_k \mid pa_k))$$

## 12.1 Independencies



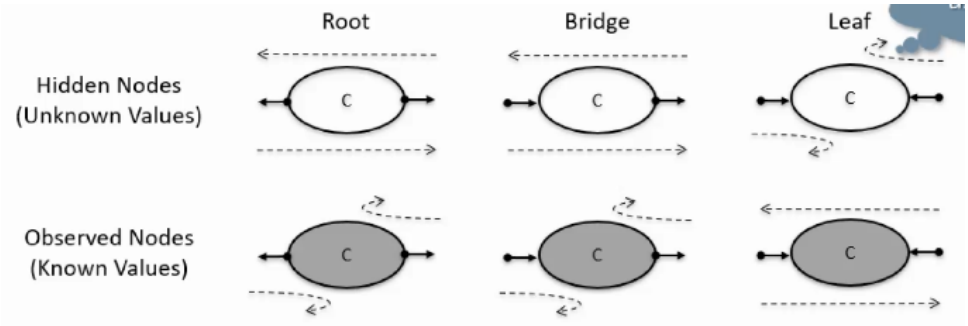

1. If you don't know z, x and y are independent (z is the leaf)

2. z is the root, x and y are not independent unless you know z → z separates x and y

3. z is the bridge, x and y are conditionally independent if you know z

## 12.2 Conditional Independence in Bayesian Networks

Two (sets of) nodes $A$ and $B$ are conditionally independent (d-separated) given $C$ if and only if all the paths from $A$ to $B$ are blocked by $C$.



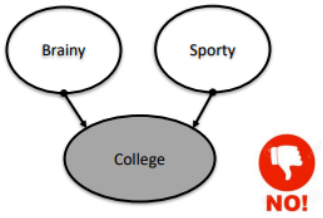

The dotted arcs indicate direction of flow in the path, if they do not traverse then the path is shielded.

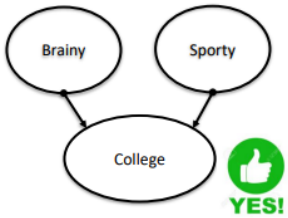
- - C is a "root": if C is hidden, children are dependent due to a hidden common cause. If C is observed, they are conditionally independent;

- C is a "leaf": if C is hidden, its parents are marginally independent, but if C, or any descendant, is observed parents become dependent (Explaining Away);

- C is a "bridge": nodes upstream and downstream of C are dependent if and only if C is hidden, because conditioning breaks the graph at that point

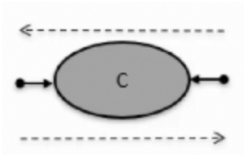
**Examples on d-separation**



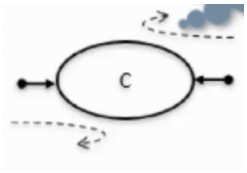




$$P(B,S|C) = P(B|C) * P(S|C) \qquad P(B,S) = P(B) * P(S)$$

Are Brainy and Sporty independent given the College information? NO, so we can't write that probability

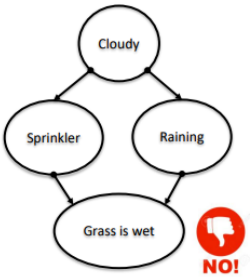


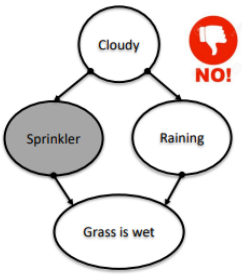

Are Brainy and Sporty independent? YES, the path is blocked

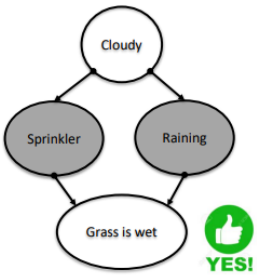



Two variables are independent if all the paths among them are blocked

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




$$P(W,C) = P(W) * P(C) \qquad P(W,C|S,R) = P(W|S,R) * P(C|S,R)$$

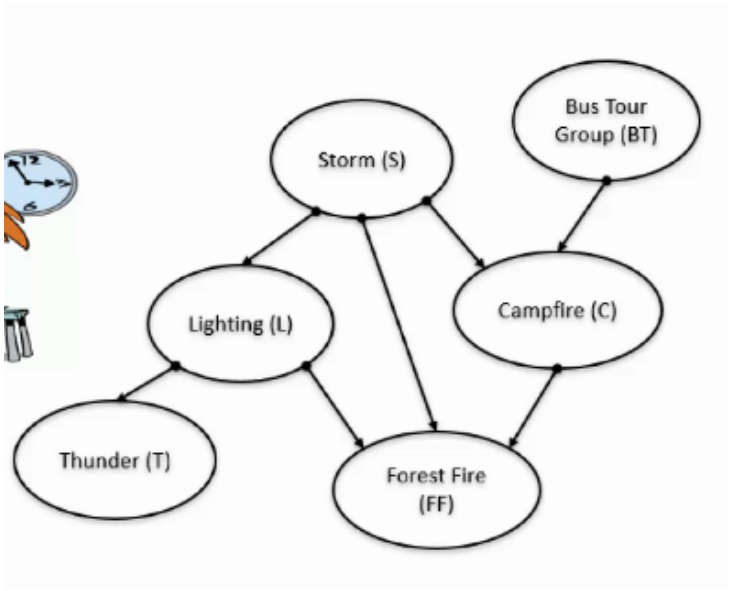
1. Are W and C independent? We do not know S and R, so the bridge is not blocked and they are not independent!

2. In the middle graph:
   C - S - G is a bridge with S that is know
   C - R - G is a bridge and it is not lock

3. We know S and R. when the bridge is known, the both paths are locked → they are independent

The arrows are used only to know if the nodes are roots, leaves or bridges.

**Example**

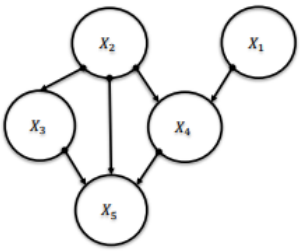



1. P(T, FF) = P(FF) P(T)? Are they independent? NO let's look all the paths from T to FF. let's check if they are locked or not? NO, because we don't know L
2. P(T, FF — L) = P(T — L) P(FF — L) Are T and FF independent, given L? YES the paths between T and FF are all locked (the root is known), so they are unconditionally independent
3. P(BT, FF) = P(BT) P(FF) ? NO check all the paths between BT and F. BT - C - FF is a bridge and the information passes, because it is not locked → NO independent
4. P(BT, FF — S) = P(BT — C) P(FF — C) ? NO when there are two arrows coming to a node, that node is a leaf, like C a node can be act as a leaf, root, bridge depending on the path when you have a root that you don't know, information passes → NO independent

## 12.3 Bayesian Networks Wrap Up

*Bayesian Network*

- A Directed Acyclic Graph (DAG) where Nodes represent random variables and Edges represent direct influence
- Conditional Probability Distributions (CPD) for priors and "influenced" variables




*Joint Distribution Factorization*

$$P(X) = P(X_1, X_2, \dots, X_N) = \prod_{k=1}^{N} P\big(X_k | parents(X_k)\big) = \prod_{k=1}^{N} P(X_k | pa_k)$$

Inference: concept of estimating the probability of a variable knowing some other variables

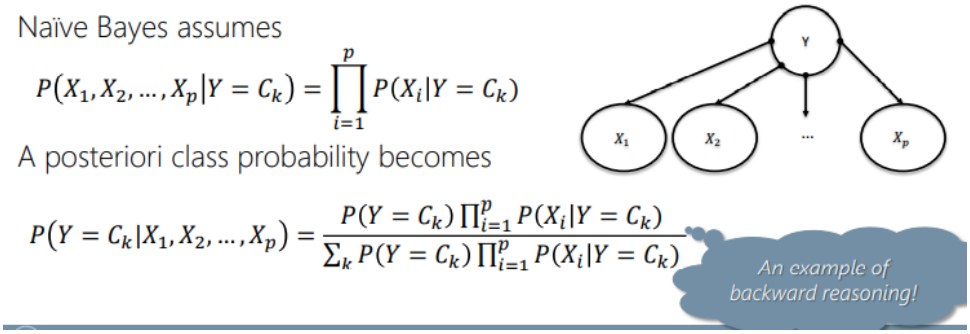
- **Bottom Up**: *grass is wet* → you make inference to understand the most likely cause. It goes from effects to causes
- **Top Down**: what happens to the grass if I switch on the sprinkler? we compute the probability that the grass will be wet given that it is cloudy

## 12.4 Naive Bayesian Classifier

You want to predict the most likely class ($C_k$) given a set of observation.

$$P(Y = C_k \mid X_1, X_2, ..., X_p) = \frac{P(X_1, X_2, ..., X_p \mid Y = C_K)P(Y = C_k)}{P(X_1, X_2, ..., X_p)}$$

You can make some assumptions saying that variables are indipendent unless you know the class.




Y is the apriori probability. Given the class, X2 is indipendent from X1.
Bayesian Networks can have:

- Real nodes: Gaussian distribution
- Discrete (that are only true or false) nodes → logistic/softmax distribution

# Chapter 13

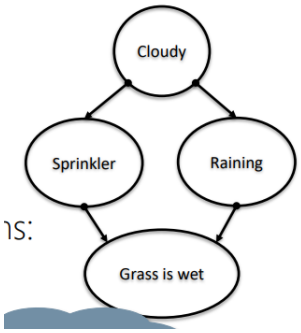
# Bayesian Network Inference

Bayesian networks exist to make inference, made via *marginalization* (summing out irrelevant variables)
Brute force enumeration has $O(|X_i|^N)$ with N=number of variables.

## 13.1 Variable Elimination

Exponential time to compute for example the probability of wet (with N=3, that are the variables).

$$\begin{aligned} P(W) &= \sum_C \sum_S \sum_R P(C,S,R,W) \\ &= \sum_C \sum_S \sum_R P(W \mid S,R)P(S \mid C)P(R \mid C)P(C) \\ &= \sum_C P(C) \sum_S P(S \mid C) \sum_R P(W \mid S,R)P(R \mid C) \end{aligned} \tag{13.1}$$

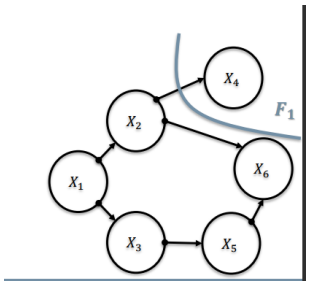



We save in temporary variables some sums to have some advantages (only if we remove the variables with the right order): we move from exponential number of sums ($2\hat{3}$) to a linear number of sums (2*3).

$$\begin{aligned} \mu_1(C,W,S) &= \sum_R P(W \mid S,R)P(R \mid C) \\ \mu_2(C,W) &= \sum_S P(S \mid C)\mu_1(C,W,S) \\ P(W) &= \sum_C P(C)\mu_2(C,W) \end{aligned} \tag{13.2}$$

**Example**

For instance, I want to compute the probability of $X_5$ → we have to marginalize out all the other variables.

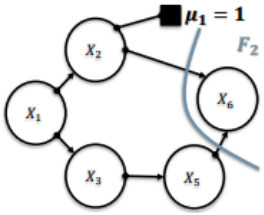



$$P(X_5) = \sum_{X_1}\sum_{X_2}\sum_{X_3}\sum_{X_4}\sum_{X_6} P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1)P(X_4 \mid X_2)P(X_5 \mid X_3)P(X_6 \mid X_5, X_2)$$

Starting from the leaves... (X4) (heuristic decision)
$\mu_1$ has two values

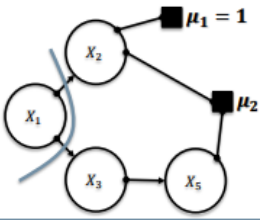
$$P(X_5) = \sum_{X_1}\sum_{X_2}\sum_{X_3}\sum_{X_6} P(X_1)P(X_2|X_1)\, P(X_3|X_1)P(X_5|X_3)P(X_6|X_5,X_2) \underbrace{\sum_{X_4} P(X_4|X_2)}_{F_1(X_2,X_4)}$$

$$P(X_5) = \sum_{X_1}\sum_{X_2}\sum_{X_3}\sum_{X_6} P(X_1)P(X_2|X_1)\, P(X_3|X_1)P(X_5|X_3)P(X_6|X_5,X_2)\mu_1(X_2)$$




$$P(X_5) = \sum_{X_1}\sum_{X_2}\sum_{X_3} P(X_1)P(X_2|X_1)\, P(X_3|X_1)P(X_5|X_3)\mu_1(X_2) \underbrace{\sum_{X_6} P(X_6|X_5,X_2)}_{F_2(X_6,X_2,X_5)}$$

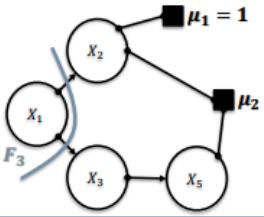
$$P(X_5) = \sum_{X_1}\sum_{X_2}\sum_{X_3} P(X_1)P(X_2|X_1)\, P(X_3|X_1)P(X_5|X_3)\mu_1(X_2)\mu_2(X_2,X_5)$$




$\mu_2$ is function of $X_5$ and $X_2$

$$P(X_5) = \sum_{X_1}\sum_{X_2}\sum_{X_3} P(X_1)P(X_2|X_1)\,P(X_3|X_1)P(X_5|X_3)\mu_1(X_2)\sum_{X_6}\underbrace{P(X_6|X_5,X_2)}_{F_2(X_6,X_2,X_5)}$$

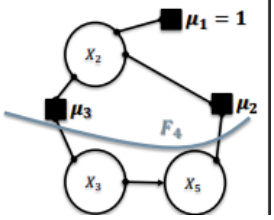
$$P(X_5) = \sum_{X_2}\sum_{X_3} P(X_5|X_3)\mu_1(X_2)\mu_2(X_2,X_5)\sum_{X_1}\underbrace{P(X_1)P(X_2|X_1)\,P(X_3|X_1)}_{F_3(X_1,X_2,X_3)}$$




Now we marginalize out a new factor ($\mu_3$) that depend on $X_2$ and $X_3$.

$$P(X_5) = \sum_{X_2}\sum_{X_3} P(X_5|X_3)\mu_1(X_2)\mu_2(X_2,X_5)\sum_{X_1}\underbrace{P(X_1)P(X_2|X_1)\,P(X_3|X_1)}_{F_3(X_1,X_2,X_3)}$$

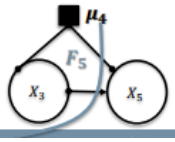
$$P(X_5) = \sum_{X_2}\sum_{X_3} P(X_5|X_3)\mu_1(X_2)\mu_2(X_2,X_5)\mu_3(X_2,X_3)$$




$$P(X_5) = \sum_{X_3} P(X_5 \mid X_3)\sum_{X_2}\underbrace{\mu_1(X_2)\mu_2(X_2,X_5)\mu_3(X_2,X_3)}_{F_4(X_2,X_5,X_3)}$$

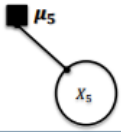
$$P(X_5) = \sum_{X_3} P(X_5|X_3)\sum_{X_2}\underbrace{\mu_1(X_2)\mu_2(X_2,X_5)\mu_3(X_2,X_3)}_{F_4(X_2,X_5,X_3)}$$

$$P(X_5) = \sum_{X_3}\underbrace{P(X_5|X_3)\mu_4(X_3,X_5)}_{F_5(X_5,X_3)}$$




$$P(X_5) = \sum_{X_3} P(X_5|X_3)\sum_{X_2}\underbrace{\mu_1(X_2)\mu_2(X_2,X_5)\mu_3(X_2,X_3)}_{F_4(X_2,X_5,X_3)}$$

$$P(X_5) = \sum_{X_3}\underbrace{P(X_5|X_3)\mu_4(X_3,X_5)}_{F_5(X_5,X_3)} = \mu_5(X_5)$$

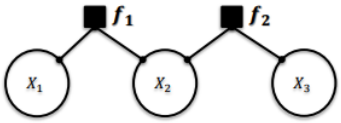



Variable Elimination is based on Dynamic Programming. The factorization of the Joint Distribution determines in which order Variable Elimination is efficient and determines what the terms F(...) and mu(...) depend on.

## 13.2 Factor Graph

It's an alternative representation of a Joint distribution factorization.

Form of Graphical Model where the box notation indicates terms that depend on some variables.




A Factor graph is a

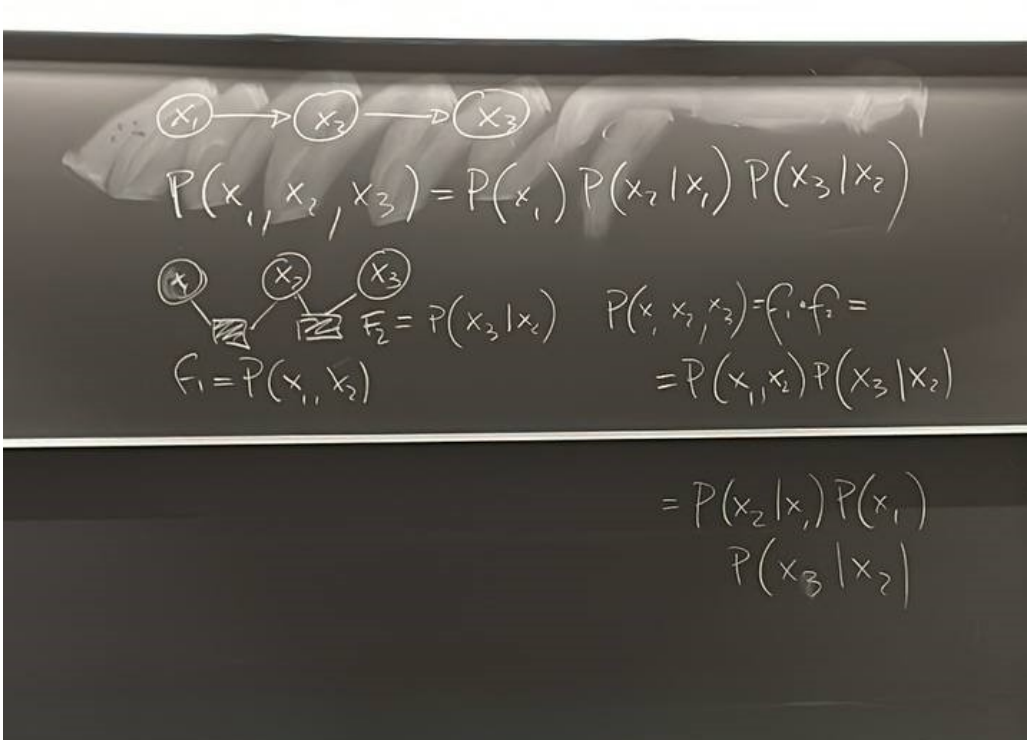
- Bipartite graph
- Each circle node represents a random variable $X_i$
- Each box node represents a factor $f_k$, which is a function $f_k(X_{C_k})$
- The joint probability distribution is given as

$$P(X_1, X_2, \ldots, X_N) = \prod_{k=1}^{K} f_k(X_{C_k})$$

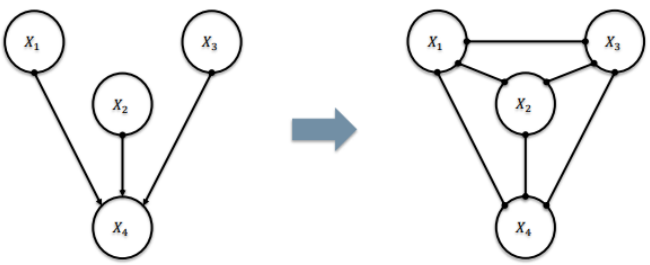
$C_k$ is the set of neighbors of k also named Clique

Note: Factors can be more general than conditional probabilities

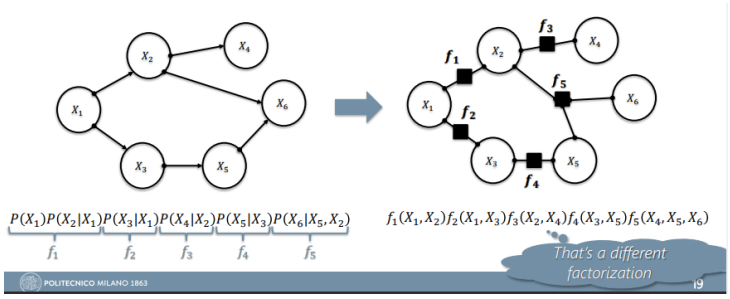
**Example**



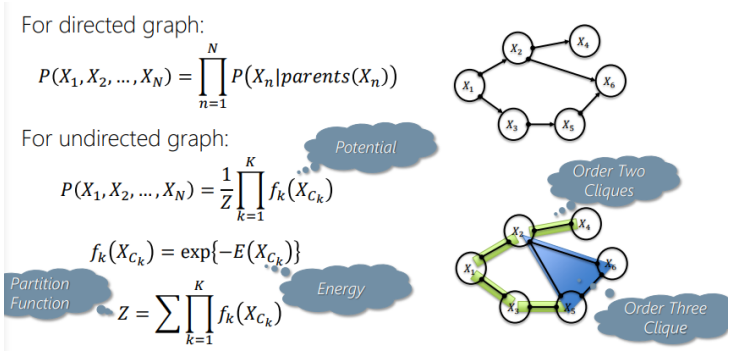

Transforming Bayesian Network into Factors Graph required an operation before: directed graph to undirected graph using the method moralization: marrying the parents → if we have two parents for a node and they are not connected, we need to connect them if they are not "married".



Moralization: we join unmarried parents into a single factor.

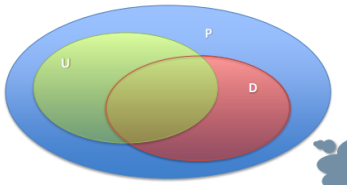



How do we get to this last graph?



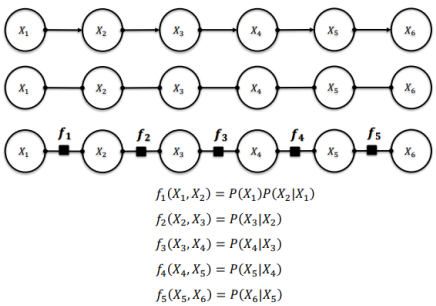

First graph: the joint is the product
Second graph: joint distribution is written as potentials for factors over Clique = subgraph of fully connected variables.
x1 is fully connected to x2 and x3. x4 is only fully connected with x2. Subgraph x2, x5, x6 is fully connected.

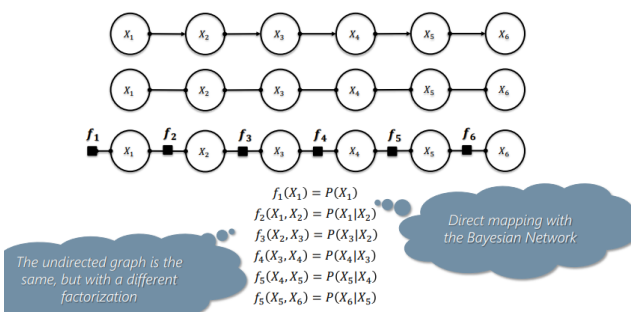
Not all the distribution can be described as Bayesan Network (directed/undirected graph). Not all directed graph can be represented as undirected graph. Not all undirected graph can be represented as directed graph



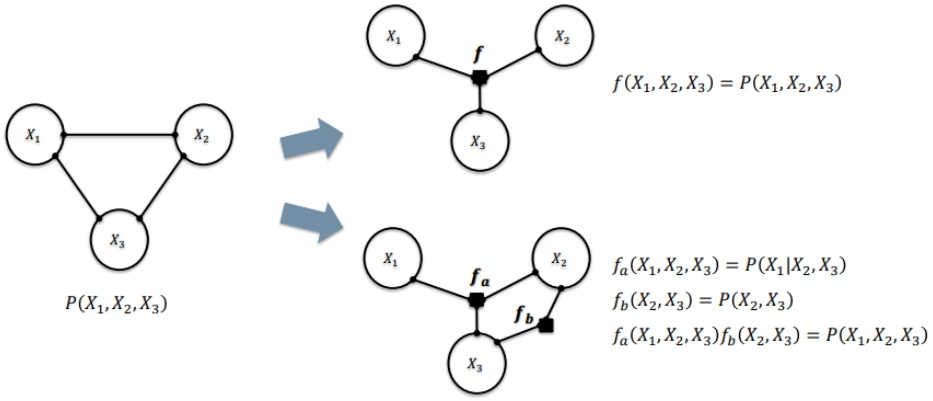

## 13.3 The Chain: from directed to undirected graph

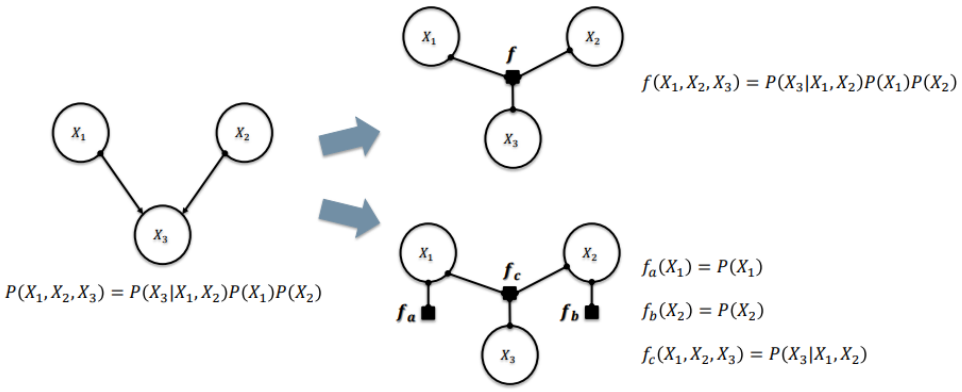
The most simple graph: it is enough if we remove the arrows



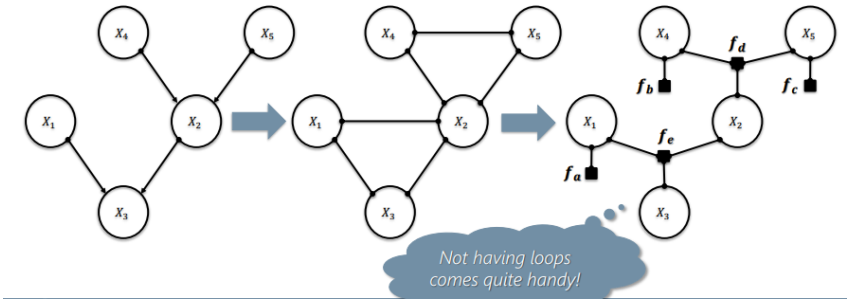
## 13.4 Factor Graphs are not unique



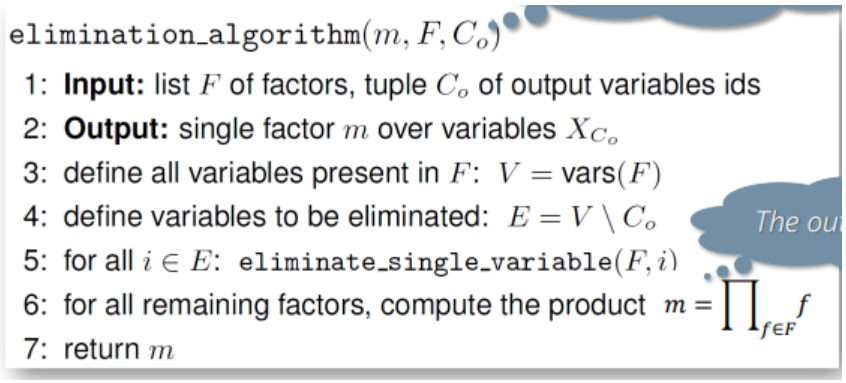

## 13.5 Polytree Exmaple

Polytree can be converted in a tree shaped factor graph → no loops.



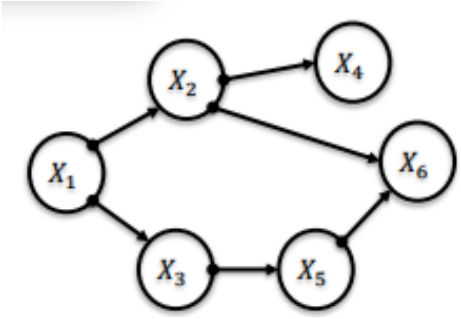

First step: marginalization
Second step: finding cliques
Third step: substituting cliques with factors.

### 13.5.1 Variable Elimination Algorithm




Example: compute the following marginal

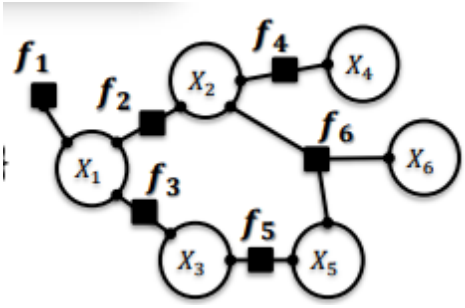
$$P(X_1, X_6) = \mu(X_1, X_6)$$




$$P(X_1, X_6) = \mu(X_1, X_6) \quad F = \{f1, f2, f3, f4, f5, f6\} \quad C_0 = \{X_1, X_6\}$$

$$V = \{X_1, X_2, X_3, X_4, X_5, X_6\} \quad E = \{X_2, X_3, X_4, X_5\}$$

V: all the variables
E: variables to remove, in this case there are 4 variables to be eliminated → the algorithm should be applied 4 times
$C_0$: variables to keep.



`eliminate_single_variable`$(F, i)$

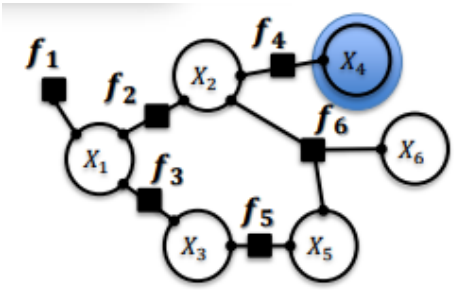
1: **Input:** list $F$ of factors, variable id $i$ $E = \{X_2, X_3, X_4, X_5\}$
2: **Output:** list $F$ of factors
3: find relevant subset $f \subset F$ of factors over $i$: $f = \{C : i \in C\}$
4: define remaining clique $C_t$ = all variables in $f$ except $i$ $C_t = \text{vars}(f) \setminus \{i\}$
5: compute temporary factor $\mu(X_{C_t}) = \sum_{X_i} \prod_{f \in F} f$
6: remove old factors $f$ and append new temporary factor $t$ to $F$
7: return $F$

We decide to start eliminating X4: variable who has less factors

$$P(X_1, X_6) = \mu(X_1, X_6) \quad F = \{f1, f2, f3, f4, f5, f6\} \quad i = X_4$$

$$f = \{f_4\} \quad C_t = \{X_2\} \quad \mu_1(X_2) = \sum_{X_4} P(X_4 \mid X_2)$$

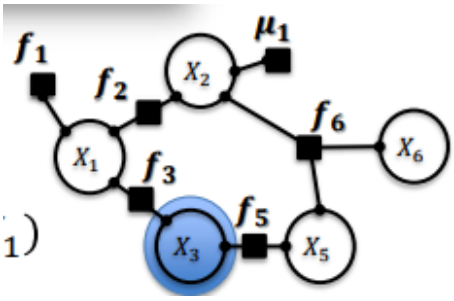
$C_t$ is the set of factors connected to the variable




Eliminating $X_4$ e replacing with $\mu_1$

$$P(X_1, X_6) = \mu(X_1, X_6) \quad F = \{f1, f2, f3, f5, f6, \mu_1\} \quad i = X_3$$

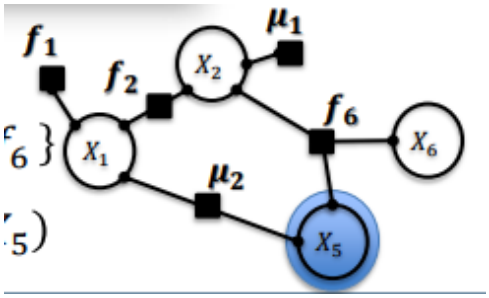
$$f = \{f_3, f_5\} \quad C_t = \{X_1, X_5\} \quad \mu_2(X_1, X_5) = \sum_{X_3} P(X_5 \mid X_3) P(X_3 \mid X_1)$$




Eliminating $X_3$ because it is connected to only two variables

$$P(X_1, X_6) = \mu(X_1, X_6) \quad F = \{f1, f2, f6, \mu_1, \mu_2\} \quad i = X_5$$

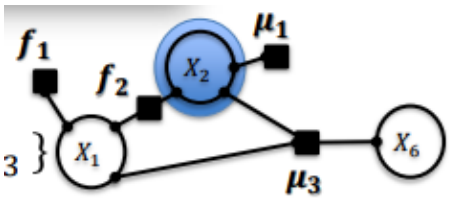
$$f = \{\mu_2, f_6\} \quad C_t = \{X_1, X_2, X_6\} \quad \mu_3(X_1, X_2, X_6) = \sum_{X_5} \mu_2(X_1, X_5) P(X_6 \mid X_2, X_5)$$




Eliminating $X_5$ because it has less factors

$$P(X_1, X_6) = \mu(X_1, X_6) \quad F = \{f1, f2, \mu_1, \mu_3\} \quad i = X_2$$

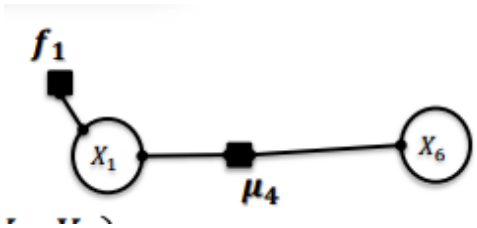
$$f = \{\mu_1, f_2, \mu_3\} \quad C_t = \{X_1, X_6\} \quad \mu_4(X_1, X_6) = \sum_{X_2} \mu_1(X_2) P(X_2 \mid X_1) \mu_3(X_1, X_2, X_6)$$




Eliminating $X_2$

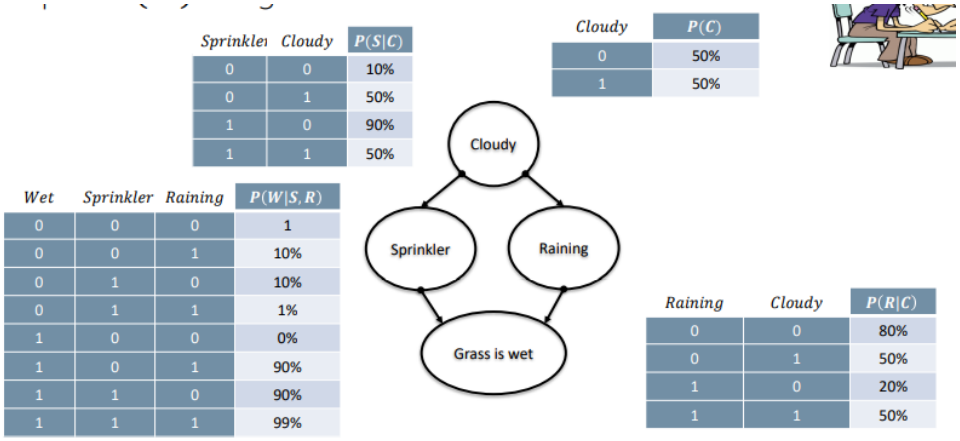
$$P(X_1, X_6) = \mu(X_1, X_6) \quad F = \{f1, \mu_4\} \quad C_0 = \{X_1, X_6\}$$

$$E = \{\cancel{X_2}, \cancel{X_3}, \cancel{X_4}, \cancel{X_5}\} \quad P(X_1, X_6) = \mu(X_1, X_6) = P(X_1)\mu_4(X_1, X_6)$$




We have eliminated all the variables we had to.

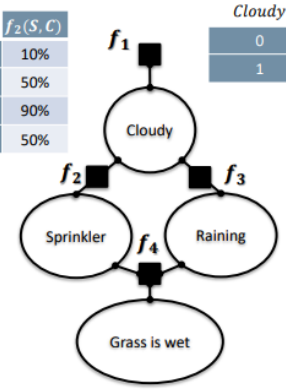
## The Sprinkler Example



We want to transform it in a factor graph, adding to the root a factor.

| Sprinkler | Cloudy | $f_2(S,C)$ |
|---|---|---|
| 0 | 0 | 10% |
| 0 | 1 | 50% |
| 1 | 0 | 90% |
| 1 | 1 | 50% |

| Cloudy | $f_1(C)$ |
|---|---|
| 0 | 50% |
| 1 | 50% |




| Wet | Sprinkler | Raining | $f_4(W,S,R)$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 10% |
| 0 | 1 | 0 | 10% |
| 0 | 1 | 1 | 1% |
| 1 | 0 | 0 | 0% |
| 1 | 0 | 1 | 90% |
| 1 | 1 | 0 | 90% |
| 1 | 1 | 1 | 99% |

| Raining | Cloudy | $f_3(R,C)$ |
|---|---|---|
| 0 | 0 | 80% |
| 0 | 1 | 50% |
| 1 | 0 | 20% |
| 1 | 1 | 50% |

Start to eliminate some variables (for example Cloudy)

## The Sprinkler Example

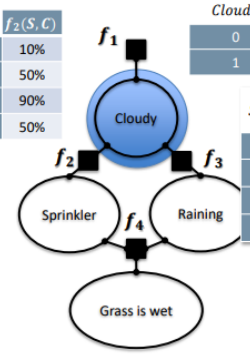
$$C_t = \{S,R\} \qquad f = \{f_1, f_2, f_3\}$$

$$\mu_1(S,R) = \sum_C f_1 \cdot f_2 \cdot f_3$$

Compute $P(W)$ using Variable Elimination

| Sprinkler | Cloudy | $f_2(S,C)$ |
|---|---|---|
| 0 | 0 | 10% |
| 0 | 1 | 50% |
| 1 | 0 | 90% |
| 1 | 1 | 50% |

| Cloudy | $f_1(C)$ |
|---|---|
| 0 | 50% |
| 1 | 50% |




| Sprinkler | Raining | $\sum_C f_1 \cdot f_2 \cdot f_3$ | $\mu_1(S,R)$ |
|---|---|---|---|
| 0 | 0 | 0.5*0.1*0.8+0.5*0.5*0.5 | 0.165 |
| 0 | 1 | 0.5*0.1*0.2+0.5*0.5*0.5 | 0.135 |
| 1 | 0 | 0.5*0.9*0.8+0.5*0.5*0.5 | 0.485 |
| 1 | 1 | 0.5*0.9*0.2+0.5*0.5*0.5 | 0.215 |

| Wet | Sprinkler | Raining | $f_4(W,S,R)$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 10% |
| 0 | 1 | 0 | 10% |
| 0 | 1 | 1 | 1% |
| 1 | 0 | 0 | 0% |
| 1 | 0 | 1 | 90% |
| 1 | 1 | 0 | 90% |
| 1 | 1 | 1 | 99% |

| Raining | Cloudy | $f_3(R,C)$ |
|---|---|---|
| 0 | 0 | 80% |
| 0 | 1 | 50% |
| 1 | 0 | 20% |
| 1 | 1 | 50% |

Replacing the three factors and the variable with the $\mu_1 \rightarrow$ creating a table of probabilities with the variables involved.
Eliminating sprinkler or raining, it's the same thing.

Compute $P(W)$ using Variable Elimination

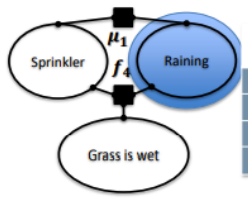
| Sprinkler | Raining | $\sum_C f_1 \cdot f_2 \cdot f_3$ | $\mu_1(S,R)$ |
|---|---|---|---|
| 0 | 0 | 0.5*0.1*0.8+0.5*0.5*0.5 | 0.165 |
| 0 | 1 | 0.5*0.1*0.2+0.5*0.5*0.5 | 0.135 |
| 1 | 0 | 0.5*0.9*0.8+0.5*0.5*0.5 | 0.485 |
| 1 | 1 | 0.5*0.9*0.2+0.5*0.5*0.5 | 0.215 |

$$C_t = \{S,W\}$$

$$f = \{\mu_1, f_4\}$$

$$\mu_2(S,W) = \sum_R \mu_1 \cdot f_4$$

| Wet | Sprinkler | Raining | $f_4(W,S,R)$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 10% |
| 0 | 1 | 0 | 10% |
| 0 | 1 | 1 | 1% |
| 1 | 0 | 0 | 0% |
| 1 | 0 | 1 | 90% |
| 1 | 1 | 0 | 90% |
| 1 | 1 | 1 | 99% |



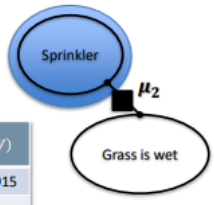

| Sprinkler | Wet | $\sum_R \mu_1 \cdot f_4$ | $\mu_2(S,W)$ |
|---|---|---|---|
| 0 | 0 | 0.165*1+0.135*0.1 | 0.1785 |
| 0 | 1 | 0.165*0+0.135*0.9 | 0.1215 |
| 1 | 0 | 0.485*0.1+0.215*0.01 | 0.05065 |
| 1 | 1 | 0.485*0.9+0.215*.99 | 0.64935 |

Without Raining and the two factors, we obtain this.

$C_t = \{W\}$

$f = \{\mu_2\}$

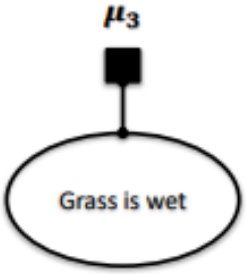
$\mu_3(W) = \sum_S \mu_2$




| Wet | $\sum_S \mu_2$ | $\mu_3(W)$ |
|---|---|---|
| 0 | 0.1785+0.05065 | 0.22915 |
| 1 | 0.1215+0.64935 | 0.77085 |

| Sprinkler | Wet | $\sum_R \mu_1 \cdot f_4$ | $\mu_2(S,W)$ |
|---|---|---|---|
| 0 | 0 | 0.165*1+0.135*0.1 | 0.1785 |
| 0 | 1 | 0.165*0+0.135*0.9 | 0.1215 |
| 1 | 0 | 0.485*0.1+0.215*0.01 | 0.05065 |
| 1 | 1 | 0.485*0.9+0.215*.99 | 0.64935 |

We want to obtain only the probability of W.

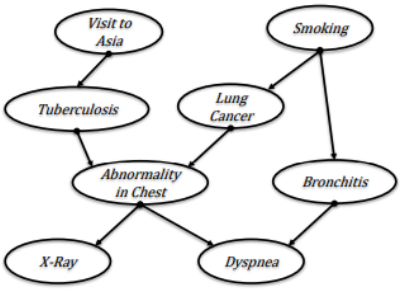



| Wet | $\sum_S \mu_2$ | $\mu_3(W)$ |
|---|---|---|
| 0 | 0.1785+0.05065 | 0.22915 |
| 1 | 0.1215+0.64935 | 0.77085 |

## The Asia Network Example

Suppose we are interested in $P(d)$

We need to eliminate: $v, s, x, t, l, a, b$



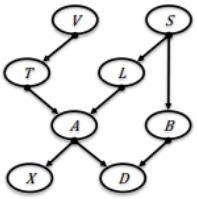

If we apply the Brute force approach we know is $O(2^N)$

$$P(d) = \sum_v \sum_s \sum_x \sum_t \sum_l \sum_a \sum_b P(v)\,P(s)\,P(t\,|\,v)\,P(l\,|\,s)\,P(b\,|\,s)\,P(a\,|\,t,l)\,P(x\,|\,a)\,P(d\,|\,a,b)$$

Removing variables one after one (Variable Elimination): it's better to remove the variables that are external

Eliminate variables in order

$v \rightarrow s \rightarrow x \rightarrow t \rightarrow l \rightarrow a \rightarrow b$



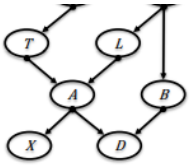

Initial factors

$P(v)\,P(s)\,P(t \mid v)\,P(l \mid s)\,P(b \mid s)\,P(a \mid t,l)\,P(x \mid a)\,P(d \mid a,b)$

$$f_v(t) = \sum_v P(v)P(t \mid v) \quad \Rightarrow f_v(t)P(s)P(l \mid s)P(b \mid s)P(a \mid t,l)P(x \mid a)P(d \mid a,b)$$

Eliminate variables in order

$v \rightarrow s \rightarrow x \rightarrow t \rightarrow l \rightarrow a \rightarrow b$




Initial factors

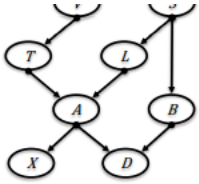
$P(v)\,P(s)\,P(t \mid v)\,P(l \mid s)\,P(b \mid s)\,P(a \mid t,l)\,P(x \mid a)\,P(d \mid a,b)$

$\Rightarrow f_v(t)P(s)P(l \mid s)P(b \mid s)P(a \mid t,l)P(x \mid a)P(d \mid a,b)$

$$f_s(b,l) = \sum_s P(s)P(b \mid s)P(l \mid s) \quad \Rightarrow f_v(t)f_s(b,l)P(a \mid t,l)P(x \mid a)P(d \mid a,b)$$

Eliminate variables in order

$v \rightarrow s \rightarrow x \rightarrow t \rightarrow l \rightarrow a \rightarrow b$




Initial factors

$P(v)\,P(s)\,P(t \mid v)\,P(l \mid s)\,P(b \mid s)\,P(a \mid t,l)\,P(x \mid a)\,P(d \mid a,b)$

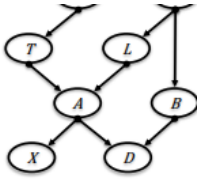
$\Rightarrow f_v(t)P(s)P(l \mid s)P(b \mid s)P(a \mid t,l)P(x \mid a)P(d \mid a,b)$

$\Rightarrow f_v(t)f_s(b,l)P(a \mid t,l)P(x \mid a)P(d \mid a,b)$

$$f_x(a) = \sum_x P(x \mid a) \quad \Rightarrow f_v(t)f_s(b,l)f_x(a)P(a \mid t,l)P(d \mid a,b)$$

Eliminate variables in order

$v \rightarrow s \rightarrow x \rightarrow t \rightarrow l \rightarrow a \rightarrow b$




Initial factors

$P(v)\,P(s)\,P(t \mid v)\,P(l \mid s)\,P(b \mid s)\,P(a \mid t,l)\,P(x \mid a)\,P(d \mid a,b)$

$\Rightarrow f_v(t)P(s)P(l \mid s)P(b \mid s)P(a \mid t,l)P(x \mid a)P(d \mid a,b)$

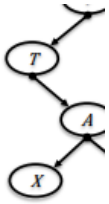
$\Rightarrow f_v(t)f_s(b,l)P(a \mid t,l)P(x \mid a)P(d \mid a,b)$

$\Rightarrow f_v(t)f_s(b,l)f_x(a)P(a \mid t,l)P(d \mid a,b)$

$$f_t(a,l) = \sum_t f_v(t)P(a \mid t,l) \quad \Rightarrow f_s(b,l)f_x(a)f_t(a,l)P(d \mid a,b)$$

Eliminate variables in order

$$v \rightarrow s \rightarrow x \rightarrow t \rightarrow l \rightarrow a \rightarrow b$$



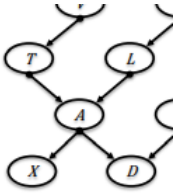

Initial factors

$$P(v)\,P(s)\,P(t|v)\,P(l|s)\,P(b|s)\,P(a|t,l)\,P(x|a)\,P(d|a,b)$$
$$\Rightarrow f_v(t)P(s)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$
$$\Rightarrow f_v(t)f_s(b,l)P(a|t,l)P(x|a)P(d|a,b)$$
$$\Rightarrow f_v(t)f_s(b,l)f_x(a)P(a|t,l)P(d|a,b)$$
$$\Rightarrow f_s(b,l)f_x(a)f_t(a,l)P(d|a,b)$$

$$f_l(a,b)=\sum_l f_s(b,l)f_t(a,l) \qquad \Rightarrow f_l(a,b)f_x(a)P(d|a,b)$$

Eliminate variables in order

$$v \rightarrow s \rightarrow x \rightarrow t \rightarrow l \rightarrow a \rightarrow b$$




Initial factors

$$P(v)\,P(s)\,P(t|v)\,P(l|s)\,P(b|s)\,P(a|t,l)\,P(x|a)\,P(d|a,b)$$
$$\Rightarrow f_v(t)P(s)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$
$$\Rightarrow f_v(t)f_s(b,l)P(a|t,l)P(x|a)P(d|a,b)$$
$$\Rightarrow f_v(t)f_s(b,l)f_x(a)P(a|t,l)P(d|a,b)$$
$$\Rightarrow f_s(b,l)f_x(a)f_t(a,l)P(d|a,b)$$
$$\Rightarrow f_l(a,b)f_x(a)P(d|a,b)$$

$$f_a(b,d)=\sum_a f_l(a,b)f_x(a)p(d|a,b) \qquad \Rightarrow f_a(b,d)$$

Eliminate variables in order

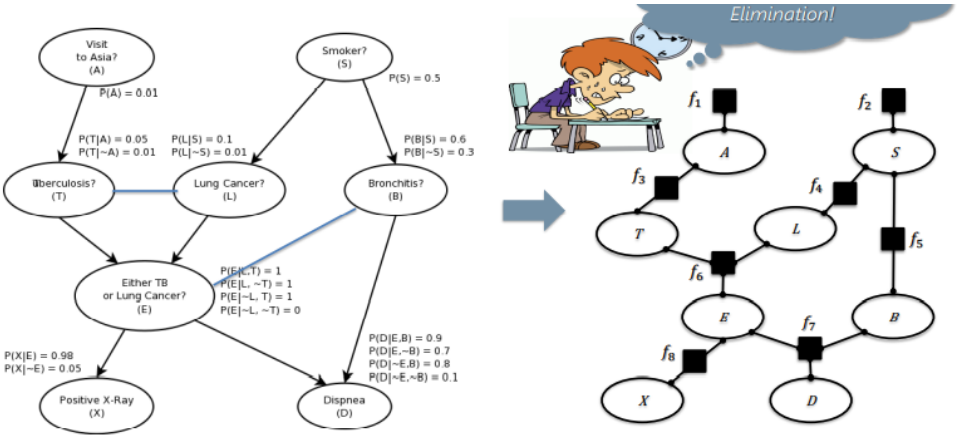
$$v \rightarrow s \rightarrow x \rightarrow t \rightarrow l \rightarrow a \rightarrow b$$

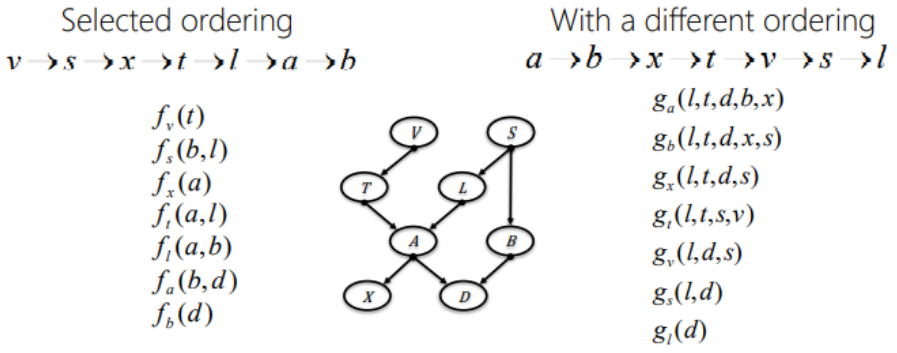

Initial factors

$$P(v)\,P(s)\,P(t|v)\,P(l|s)\,P(b|s)\,P(a|t,l)\,P(x|a)\,P(d|a,b)$$
$$\Rightarrow f_v(t)P(s)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$
$$\Rightarrow f_v(t)f_s(b,l)P(a|t,l)P(x|a)P(d|a,b)$$
$$\Rightarrow f_v(t)f_s(b,l)f_x(a)P(a|t,l)P(d|a,b)$$
$$\Rightarrow f_s(b,l)f_x(a)f_t(a,l)P(d|a,b)$$
$$\Rightarrow f_l(a,b)f_x(a)P(d|a,b)$$
$$\Rightarrow f_a(b,d)$$

$$f_b(d)=\sum_b f_a(b,d) \Rightarrow f_b(d)$$

Now substitute triangles with factors




complexity is exponential in the size of these factors → we should find the right ordering, keeping k (= maximal number of variables in the factors) as mininum as possible.

PROS VARIABLE ELIMINATION:

- very simple to implement
- does what you would do on paper
- with optimal ordering complexity is $O(N * 2^K)$

CONS VARIABLE ELIMINATION:

- finding the optimal ordering is an NP hard problem
- it computes only one marginal at the time
- requires N executions to compute all marginals

To improve on this we can use Belief Propagation (→ Sum Product Algorithms) which uses reusable local factors and message passing.

## 13.6 Belief Propagation

Very efficient if we have a polytree structure → no loops.
It's similar to Variable Elimination, sometime known as message passing. It's an alternative way to compute variable elimination.

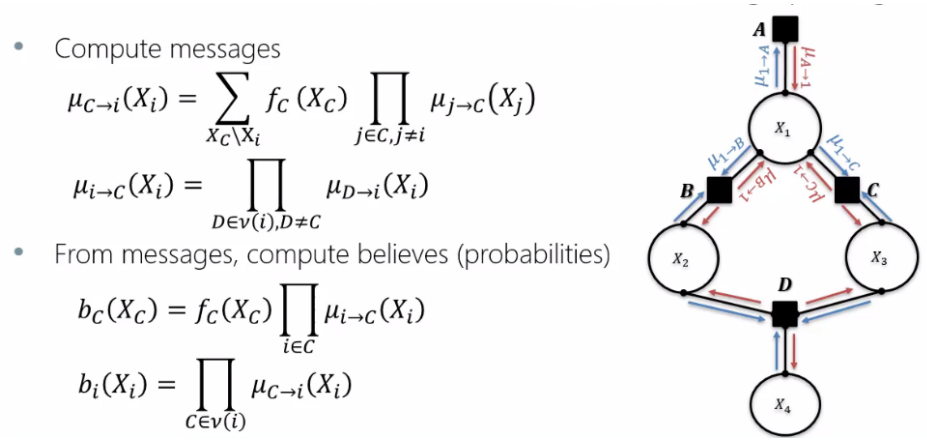
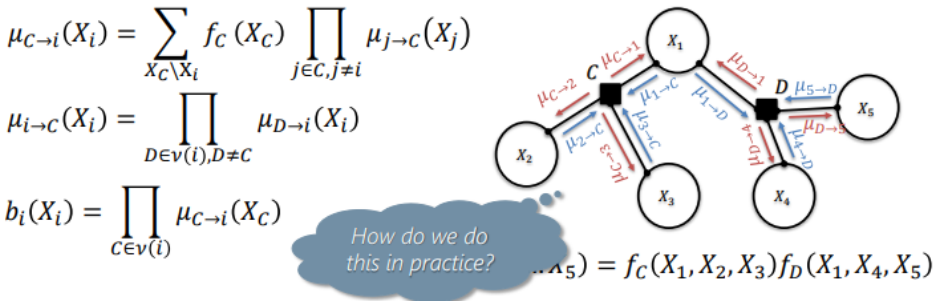



Figure 13.2: Red ones: factors to variables messages. Blue ones: variables to factor messages

We want to avoid cycles.
Circular dependences can be resolved if the graph is a tree! → you don't have cycles in a tree

## 13.7 Belief Propagation in Trees




Two factors: C and D. In Variable Elimination (one variable at the time):

$$P(X_1) = \left[\sum_{X_2,X_3} f_C(X_1, X_2, X_3)\right]\left[\sum_{X_4,X_5} f_D(X_1, X_4, X_5)\right] \qquad P(X_1,$$

$$\mu_C(X_1) = \left[\sum_{X_2,X_3} f_C(X_1, X_2, X_3)\right] \quad \mu_D(X_1) = \left[\sum_{X_4,X_5} f_D(X_1, X_4, X_5)\right]$$

In Belief Propagation you can compute variable elimination in parallel (this is the main difference with the Variable Elimination):

$$b(X_1) = \mu_{C\to 1}(X_1)\mu_{D\to 1}(X_1) \qquad P(X_1$$

$$\mu_{C\to 1}(X_1) = \sum_{X_2,X_3} f_C(X_1, X_2, X_3) \quad \mu_{D\to 1}(X_1) = \sum_{X_4,X_5} f_D(X_1, X_4, X_5)$$

The branches of the tree contain independent information, and we can fuse branches information via

simple multiplication.
Once propagated all messages, we can return all marginals.

We can write Belief Propagation as a Parallel Procedure:

1. Initialize all messages from factor to variable as 1. $\mu_{f_s \to X_i} = 1$

2. Update messages

   - Allows for parallel update of messages
   - Iterate until convergence

$$\mu^{new}_{f_s \to X_i}(X_i) = \sum_{X_S \setminus X_i} f_S(X_i, X_S) \prod_{X_j \in ne(X_i), j \neq i} \mu^{old}_{X_j \to f_S}(X_j)$$

$$\mu^{new}_{X_i \to f_s}(X_i) = \prod_{\substack{f_l \in ne(X_i) \\ f_l \neq f_s}} \mu^{old}_{f_l \to X_i}(X_i)$$

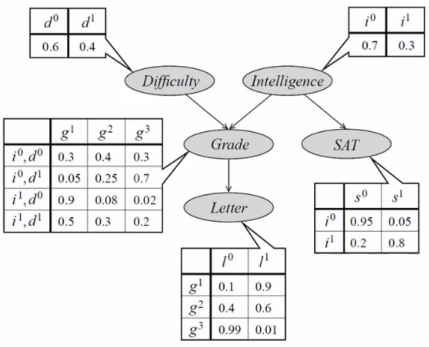
Iterate unt

3. Update believes (at the end of the converge). Believes over clicks and variables.

$$b_{f_S}(X_S) = f_S(X_S) \prod_{j \in f_S} \mu_{X_j \to f_S}(X_j)$$

$$b_{X_i}(X_i) = \prod_{f_S \in ne(i)} \mu_{f_S \to X_i}(X_i)$$

## Belief Propagation Example




| $d^0$ | $d^1$ |
|---|---|
| 0.6 | 0.4 |

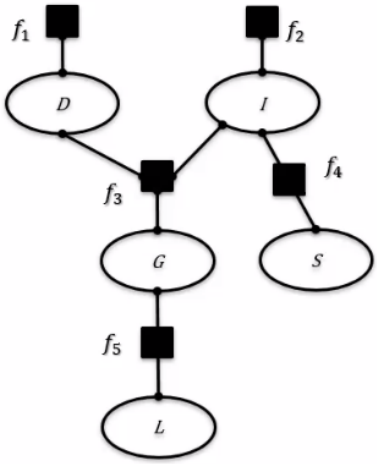
| $i^0$ | $i^1$ |
|---|---|
| 0.7 | 0.3 |

| | $g^1$ | $g^2$ | $g^3$ |
|---|---|---|---|
| $i^0, d^0$ | 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 |

#sat: 2 to 1
#parameters for grade: two inputs ( = 4 rows), you need two columns (three variables - 1)
Transforming the network in a factor graph, remembering to add factors to root: f1 = apriori probability of D, f3 = conditional probability of G given D and I




| G | I | D | $f_3(G,I,D)$ |
|---|---|---|---|
| 1 | 0 | 0 | 30% |
| 1 | 0 | 1 | 5% |
| 1 | 1 | 0 | 90% |
| 1 | 1 | 1 | 50% |
| 2 | 0 | 0 | 40% |
| 2 | 0 | 1 | 25% |
| 2 | 1 | 0 | 8% |
| 2 | 1 | 1 | 30% |
| 3 | 0 | 0 | 30% |
| 3 | 0 | 1 | 70% |
| 3 | 1 | 0 | 2% |
| 3 | 1 | 1 | 20% |

| D | $f_1(D)$ |
|---|---|
| 0 | 60% |
| 1 | 40% |

| I | $f_2(I)$ |
|---|---|
| 0 | 70% |
| 1 | 30% |

| L | G | $f_5(L,G)$ |
|---|---|---|
| 0 | 1 | 10% |
| 0 | 2 | 40% |
| 0 | 3 | 99% |
| 1 | 1 | 90% |
| 1 | 2 | 60% |
| 1 | 3 | 1% |

| S | I | $f_4(S,I)$ |
|---|---|---|
| 0 | 0 | 95% |
| 0 | 1 | 20% |
| 1 | 0 | 5% |
| 1 | 1 | 80% |

This is a Polytree → we can apply Belief Propagation

1. Let initialize the variables

$$\mu_{D\to 1} = [1,1] \quad \mu_{1\to D} = [1,1]$$
$$\mu_{D\to 3} = [1,1] \quad \mu_{2\to I} = [1,1]$$
$$\mu_{G\to 3} = [1,1] \quad \mu_{3\to D} = [1,1]$$
$$\mu_{G\to 5} = [1,1] \quad \mu_{3\to I} = [1,1]$$
$$\mu_{I\to 3} = [1,1] \quad \mu_{3\to G} = [1,1,1]$$
$$\mu_{I\to 2} = [1,1] \quad \mu_{4\to I} = [1,1]$$
$$\mu_{I\to 4} = [1,1] \quad \mu_{4\to S} = [1,1]$$
$$\mu_{L\to 5} = [1,1] \quad \mu_{5\to G} = [1,1,1]$$
$$\mu_{S\to 4} = [1,1] \quad \mu_{5\to L} = [1,1]$$

2. Update messages from factors to variables.

$$\mu^{new}_{f_S\to X_i}(X_i) = \sum_{X_S\setminus X_i} f_S(X_i, X_S) \prod_{X_j\in ne(X_i), j\neq i} \mu^{old}_{X_j\to f_S}(X_j)$$

$\mu_{1\to D}(D) =$

| D | $\mu_{1\to D}(D)$ |
|---|---|
| 0 | 0.6 |
| 1 | 0.4 |

$\mu_{2\to I}(I) =$

| I | $\mu_{2\to I}(I)$ |
|---|---|
| 0 | 0.7 = 0.7 |
| 1 | 0.3 = 0.3 |

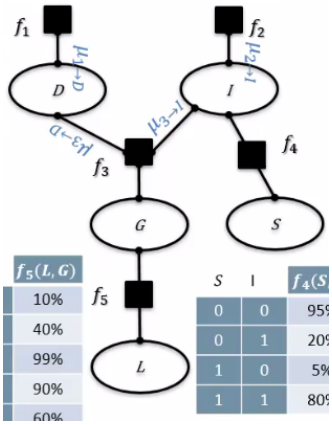
$\mu_{3\to D}(D) =$

| D | $\mu_{3\to D}(D)$ |
|---|---|
| 0 | $0.3\cdot 1\cdot 1+0.9\cdot 1\cdot 1+0.4\cdot 1\cdot 1+0.08\cdot 1\cdot 1+0.3\cdot 1\cdot 1+0.02\cdot 1\cdot 1=2$ |
| 1 | $0.05\cdot 1\cdot 1+0.5\cdot 1\cdot 1+0.25\cdot 1\cdot 1+0.3\cdot 1\cdot 1+0.7\cdot 1\cdot 1+0.2\cdot 1\cdot 1=2$ |

$\mu_{3\to I}(I) =$

| I | $\mu_{3\to I}(I)$ |
|---|---|
| 0 | $0.3\cdot 1\cdot 1+0.05\cdot 1\cdot 1+0.4\cdot 1\cdot 1+0.25\cdot 1\cdot 1+0.3\cdot 1\cdot 1+0.7\cdot 1\cdot 1=2$ |
| 1 | $0.9\cdot 1\cdot 1+0.5\cdot 1\cdot 1+0.08\cdot 1\cdot 1+0.3\cdot 1\cdot 1+0.02\cdot 1\cdot 1+0.2\cdot 1\cdot 1=2$ |

| | | |
|---|---|---|
| 2 | 0 | 0 |
| 2 | 0 | 1 |
| 2 | 1 | 0 |
| 2 | 1 | 1 |
| 3 | 0 | 0 |
| 3 | 0 | 1 |
| 3 | 1 | 0 |
| 3 | 1 | 1 |




$\mu_1$: all the factors involved in the factor 1. Messages to below

$\mu_{4\to S}(S) =$

| S | $\mu_{4\to S}(S)$ |
|---|---|
| 0 | $0.95\cdot 1+0.2\cdot 1=1.15$ |
| 1 | $0.05\cdot 1+0.8\cdot 1=0.85$ |

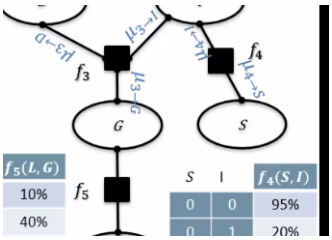
$\mu_{4\to I}(I) =$

| I | $\mu_{2\to I}(I)$ |
|---|---|
| 0 | $0.95\cdot 1+0.05\cdot 1=1$ |
| 1 | $0.2\cdot 1+0.8\cdot 1=1$ |

$\mu_{3\to G}(G) =$

| G | $\mu_{3\to G}(G)$ |
|---|---|
| 1 | $0.3\cdot 1\cdot 1+0.05\cdot 1\cdot 1+0.9\cdot 1\cdot 1+0.5\cdot 1\cdot 1=1.75$ |
| 2 | $0.4\cdot 1\cdot 1+0.25\cdot 1\cdot 1+0.08\cdot 1\cdot 1+0.3\cdot 1\cdot 1=1.03$ |
| 3 | $0.3\cdot 1\cdot 1+0.7\cdot 1\cdot 1+0.02\cdot 1\cdot 1+0.2\cdot 1\cdot 1=1.22$ |

| | | |
|---|---|---|
| 2 | 0 | 0 |
| 2 | 0 | 1 |
| 2 | 1 | 0 |
| 2 | 1 | 1 |
| 3 | 0 | 0 |
| 3 | 0 | 1 |
| 3 | 1 | 0 |
| 3 | 1 | 1 |



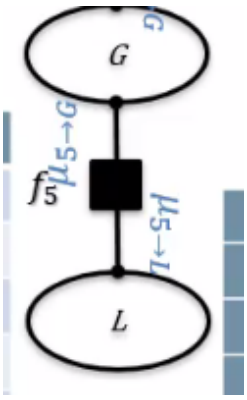

Last two messages

$\mu_{5\to G}(G) =$

| G | $\mu_{5\to G}(G)$ |
|---|---|
| 1 | $0.1\cdot 1+0.9\cdot 1=1$ |
| 2 | $0.4\cdot 1+0.6\cdot 1=1$ |
| 3 | $0.99\cdot 1+0.01\cdot 1=1$ |

$\mu_{5\to L}(L) =$

| L | $\mu_{5\to L}(L)$ |
|---|---|
| 0 | $0.1\cdot 1+0.4\cdot 1+0.99\cdot 1=1.49$ |
| 1 | $0.9\cdot 1+0.6\cdot 1+0.01\cdot 1=1.51$ |




3. Updating the variables and now we consider messages from variables to factors

$\mu_{D\to1} = [1,1]$ $\mu_{1\to D} = [0.6, 0.4]$
$\mu_{D\to3} = [1,1]$ $\mu_{2\to I} = [0.7, 0.3]$
$\mu_{G\to3} = [1,1]$ $\mu_{3\to D} = [2,2]$
$\mu_{G\to5} = [1,1]$ $\mu_{3\to I} = [2,2]$
$\mu_{I\to3} = [1,1]$ $\mu_{3\to G} = [1.75, 1.03, 1.22]$
$\mu_{I\to2} = [1,1]$ $\mu_{4\to I} = [1,1]$
$\mu_{I\to4} = [1,1]$ $\mu_{4\to S} = [1.15, 0.85]$
$\mu_{L\to5} = [1,1]$ $\mu_{5\to G} = [1,1,1]$
$\mu_{S\to4} = [1,1]$ $\mu_{5\to L} = [1.49, 1.51]$

$$\mu^{new}_{X_i\to f_s}(X_i) = \prod_{\substack{f_l\in ne(X_i)\\ f_l\neq f_s}} \mu^{old}_{f_l\to X_i}(X_i)$$

$\mu_{D\to1}(D) =$

| D | $\mu_{D\to1}(D)$ |
|---|---|
| 0 | 2 |
| 1 | 2 |

$\mu_{D\to3}(D) =$

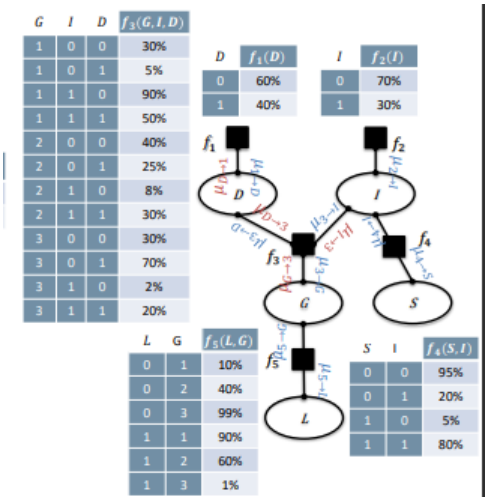
| D | $\mu_{D\to3}(D)$ |
|---|---|
| 0 | 0.6 |
| 1 | 0.4 |

$\mu_{G\to3}(G) =$

| G | $\mu_{G\to3}(G)$ |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |

$\mu_{I\to3}(I) =$

| I | $\mu_{I\to3}(I)$ |
|---|---|
| 0 | $1 \cdot 0.7 = 0.7$ |
| 1 | $1 \cdot 0.3 = 0.3$ |




message D→1 = we look the message 3→D that is [2,2]
message D→3 = we look the message 1→D;
message I→3 = product of f2 (0,7 0,3) and f4 (1, 1).

$\mu_{I\to2}(I) =$

| I | $\mu_{I\to2}(I)$ |
|---|---|
| 0 | $2 \cdot 1 = 2$ |
| 1 | $2 \cdot 1 = 2$ |

$\mu_{S\to4}(S) =$

| S | $\mu_{S\to4}(S)$ |
|---|---|
| 0 | 1 |
| 1 | 1 |

$\mu_{G\to5}(G) =$

| G | $\mu_{G\to5}(G)$ |
|---|---|
| 1 | 1.75 |
| 2 | 1.03 |
| 3 | 1.22 |

$\mu_{L\to5}(L) =$

| L | $\mu_{L\to5}(L)$ |
|---|---|
| 0 | 1 |
| 1 | 1 |

$\mu_{I\to4}(I) =$

| I | $\mu_{I\to4}(I)$ |
|---|---|
| 0 | $0.7 \cdot 2 = 1.4$ |
| 1 | $0.3 \cdot 2 = 0.6$ |

We have updated this message, but we have to iterate until convergence

$\mu_{D\to 1} = [2,2]$ $\mu_{1\to D} = [0.6, 0.4]$

$\mu_{D\to 3} = [0.6, 0.4]$ $\mu_{2\to I} = [0.7, 0.3]$

$\mu_{G\to 3} = [1,1,1]$ $\mu_{3\to D} = [2,2]$

$\mu_{G\to 5} = [1.75, 1.03, 1.22]$ $\mu_{3\to I} = [2,2]$

$\mu_{I\to 3} = [0.7, 0.3]$ $\mu_{3\to G} = [1.75, 1.03, 1.22]$

$\mu_{I\to 2} = [2,2]$ $\mu_{4\to I} = [1,1]$

$\mu_{I\to 4} = [1.4\ 0.6]$ $\mu_{4\to S} = [1.15, 0.85]$

$\mu_{L\to 5} = [1,1]$ $\mu_{5\to G} = [1,1,1]$

$\mu_{S\to 4} = [1,1]$ $\mu_{5\to L} = [1.49, 1.51]$

$$\mu^{new}_{f_S\to X_i}(X_i) = \sum_{X_S\setminus X_i} f_S(X_i, X_S) \prod_{X_j\in ne(X_i), j\neq i} \mu^{old}_{X_j\to f_S}(X_j)$$

$\mu_{1\to D}(D) =$

| D | $\mu_{1\to D}(D)$ |
|---|---|
| 0 | 0.6 |
| 1 | 0.4 |

$\mu_{2\to I}(I) =$

| I | $\mu_{2\to I}(I)$ |
|---|---|
| 0 | 0.7 |
| 1 | 0.3 |

$\mu_{3\to D}(D) =$

| D | $\mu_{3\to D}(D)$ |
|---|---|
| 0 | $0.3\cdot 0.7\cdot 1 + 0.9\cdot 0.3\cdot 1 + 0.4\cdot 0.7\cdot 1 + 0.08\cdot 0.3\cdot 1 + 0.3\cdot 0.7\cdot 1 + 0.02\cdot 0.3\cdot 1 = 1$ |
| 1 | $0.05\cdot 0.7\cdot 1 + 0.5\cdot 0.3\cdot 1 + 0.25\cdot 0.7\cdot 1 + 0.3\cdot 0.3\cdot 1 + 0.7\cdot 0.7\cdot 1 + 0.2\cdot 0.3\cdot 1 = 1$ |

$\mu_{3\to I}(I) =$

| I | $\mu_{3\to I}(I)$ |
|---|---|
| 0 | $0.3\cdot 0.6\cdot 1 + 0.05\cdot 0.4\cdot 1 + 0.4\cdot 0.6\cdot 1 + 0.25\cdot 0.4\cdot 1 + 0.3\cdot 0.6\cdot 1 + 0.7\cdot 0.4\cdot 1 = 1$ |
| 1 | $0.9\cdot 0.6\cdot 1 + 0.5\cdot 0.4\cdot 1 + 0.08\cdot 0.6\cdot 1 + 0.3\cdot 0.4\cdot 1 + 0.02\cdot 0.6\cdot 1 + 0.2\cdot 0.4\cdot 1 = 1$ |

| | | |
|---|---|---|
| 1 | 0 | |
| 1 | 1 | |
| 1 | 1 | |
| 2 | 0 | |
| 2 | 0 | |
| 2 | 1 | |
| 2 | 1 | |
| 3 | 0 | |
| 3 | 0 | |
| 3 | 1 | |
| 3 | 1 | |

$\mu_{4\to S}(S) =$

| S | $\mu_{4\to S}(S)$ |
|---|---|
| 0 | $0.95\cdot 1.4 + 0.2\cdot 0.6 = 1.45$ |
| 1 | $0.05\cdot 1.4 + 0.8\cdot 0.6 = 0.55$ |

$\mu_{4\to I}(I) =$

| I | $\mu_{2\to I}(I)$ |
|---|---|
| 0 | $0.95\cdot 1 + 0.05\cdot 1 = 1$ |
| 1 | $0.2\cdot 1 + 0.8\cdot 1 = 1$ |

| | | |
|---|---|---|
| 2 | 0 | 1 |
| 2 | 1 | 0 |
| 2 | 1 | 1 |
| 3 | 0 | 0 |
| 3 | 0 | 1 |
| 3 | 1 | 0 |
| 3 | 1 | 1 |

$\mu_{3\to G}(G) =$

| G | $\mu_{3\to G}(G)$ |
|---|---|
| 1 | $0.3\cdot 0.6\cdot 0.7 + 0.05\cdot 0.4\cdot 0.7 + 0.9\cdot 0.6\cdot 0.3 + 0.5\cdot 0.4\cdot 0.3 = 0.362$ |
| 2 | $0.4\cdot 0.6\cdot 0.7 + 0.25\cdot 0.4\cdot 0.7 + 0.08\cdot 0.6\cdot 0.3 + 0.3\cdot 0.4\cdot 0.3 = 0.2884$ |
| 3 | $0.3\cdot 0.6\cdot 0.7 + 0.7\cdot 0.4\cdot 0.7 + 0.02\cdot 0.6\cdot 0.3 + 0.2\cdot 0.4\cdot 0.3 = 0.3496$ |

Message 3→D = 0) 0.3 (probability of D=0) * 0,7 (probability of I=0) * 1 (probability of G=1). To compute these last long messages, we need to look into the probability table that contains G, I, D.

$\mu_{5\to G}(G) =$

| G | $\mu_{5\to G}(G)$ |
|---|---|
| 1 | $0.1\cdot 1 + 0.9\cdot 1 = 1$ |
| 2 | $0.4\cdot 1 + 0.6\cdot 1 = 1$ |
| 3 | $0.99\cdot 1 + 0.01\cdot 1 = 1$ |

$\mu_{5\to L}(L) =$

| L | $\mu_{5\to L}(L)$ |
|---|---|
| 0 | $0.1\cdot 1.75 + 0.4\cdot 1.03 + 0.99\cdot 1.22 = 1.7948$ |
| 1 | $0.9\cdot 1.75 + 0.6\cdot 1.03 + 0.01\cdot 1.22 = 2.2052$ |

$\mu_{D\to 1} = [2,2]$ $\mu_{1\to D} = [0.6, 0.4]$

$\mu_{D\to 3} = [0.6, 0.4]$ $\mu_{2\to I} = [0.7, 0.3]$

$\mu_{G\to 3} = [1,1,1]$ $\mu_{3\to D} = [1,1]$

$\mu_{G\to 5} = [1.75, 1.03, 1.22]$ $\mu_{3\to I} = [1,1]$

$\mu_{I\to 3} = [0.7, 0.3]$ $\mu_{3\to G} = [0.362, 0.288, 0.349]$

$\mu_{I\to 2} = [2,2]$ $\mu_{4\to I} = [1,1]$

$\mu_{I\to 4} = [1.4\ 0.6]$ $\mu_{4\to S} = [1.45, 0.55]$

$\mu_{L\to 5} = [1,1]$ $\mu_{5\to G} = [1,1,1]$

$\mu_{S\to 4} = [1,1]$ $\mu_{5\to L} = [1.7948, 2.2052]$

$$\mu^{new}_{X_i \to f_s}(X_i) = \prod_{\substack{f_l \in ne(X_i) \\ f_l \neq f_s}} \mu^{old}_{f_l \to X_i}(X_i)$$

$\mu_{D\to1}(D) =$

| D | $\mu_{D\to1}(D)$ |
|---|---|
| 0 | 1 |
| 1 | 1 |

$\mu_{D\to3}(D) =$

| D | $\mu_{D\to3}(D)$ |
|---|---|
| 0 | 0.6 |
| 1 | 0.4 |

$\mu_{I\to2}(I) =$

| I | $\mu_{I\to2}(I)$ |
|---|---|
| 0 | $1 \cdot 1 = 1$ |
| 1 | $1 \cdot 1 = 1$ |

$\mu_{S\to4}(S) =$

| S | $\mu_{S\to4}(S)$ |
|---|---|
| 0 | 1 |
| 1 | 1 |

$\mu_{G\to3}(G) =$

| G | $\mu_{G\to3}(G)$ |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |

$\mu_{G\to5}(G) =$

| G | $\mu_{G\to5}(G)$ |
|---|---|
| 1 | 0.362 |
| 2 | 0.2884 |
| 3 | 0.3494 |

$\mu_{L\to5}(L) =$

| L | $\mu_{L\to5}(L)$ |
|---|---|
| 0 | 1 |
| 1 | 1 |

$\mu_{I\to3}(I) =$

| I | $\mu_{I\to3}(I)$ |
|---|---|
| 0 | $1 \cdot 0.7 = 1.4$ |
| 1 | $1 \cdot 0.3 = 0.6$ |

$\mu_{I\to4}(I) =$

| I | $\mu_{I\to4}(I)$ |
|---|---|
| 0 | $0.7 \cdot 1 = 0.7$ |
| 1 | $0.3 \cdot 1 = 0.3$ |

$$\mu_{D\to1} = [1, 1] \quad \mu_{1\to D} = [0.6, 0.4]$$
$$\mu_{D\to3} = [0.6, 0.4] \quad \mu_{2\to I} = [0.7, 0.3]$$
$$\mu_{G\to3} = [1, 1, 1] \quad \mu_{3\to D} = [1, 1]$$
$$\mu_{G\to5} = [1.75, 1.03, 1.22] \quad \mu_{3\to I} = [1, 1]$$
$$\mu_{I\to3} = [1.4, 0.6] \quad \mu_{3\to G} = [0.362, 0.288, 0.349]$$
$$\mu_{I\to2} = [2, 2] \quad \mu_{4\to I} = [1, 1]$$
$$\mu_{I\to4} = [0.7, 0.3] \quad \mu_{4\to S} = [1.45, 0.55]$$
$$\mu_{L\to5} = [1, 1] \quad \mu_{5\to G} = [1, 1, 1]$$
$$\mu_{S\to4} = [1, 1] \quad \mu_{5\to L} = [1.7948, 2.2052]$$

They don't necessary have to sum up to 1.
Computing the belief

$$b_{X_i}(X_i) = \prod_{f_S \in ne(i)} \mu_{f_S \to X_i}(X_i)$$

| | | |
|---|---|---|
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 1 |
| 2 | 0 | 0 |
| 2 | 0 | 1 |
| 2 | 1 | 0 |
| 2 | 1 | 1 |
| 3 | 0 | 0 |
| 3 | 0 | 1 |
| 3 | 1 | 0 |
| 3 | 1 | 1 |

$$b(D) = [0.6 \cdot 1 = 0.6, 0.4 \cdot 1 = 0.4]$$
$$b(I) = [0.7 \cdot 1 \cdot 1 = 0.7, 0.3 \cdot 1 \cdot 1 = 0.3]$$
$$b(S) = [\frac{1.45}{1.45 + 0.55} = 0.72, \frac{0.55}{1.45 + 0.55} = 0.28]$$
$$b(L) = [\frac{1.79}{1.79 + 2.20} = 0.45, \frac{2.20}{1.79 + 2.20} = 0.55]$$
$$b(G) = [0.362 \cdot 1 = 0.362, 0.288 \cdot 1 = 0.288, 0.349 \cdot 1 = 0.349]$$

we compute belief in leaves with that formula (*see b(S) and b(L)*).

## Adding Evidence

Let's consider $P(X) = P(X_1)P(X_2|X_1)P(X_3|X_2)P(X_4|X_3)$ the marginal probability of $X_2$ and observations $X_1 = x_{e_1}$ and $X_3 = x_{e_3}$ is:

$$P(X_2|X_1 = x_{e_1}, X_3 = x_{e_3}) = P(X_1 = x_{e_1})P(X_2|X_1 = x_{e_1})P(X_3 = x_{e_3}|X_2)\sum_{X_4} P(X_4|X_3)$$

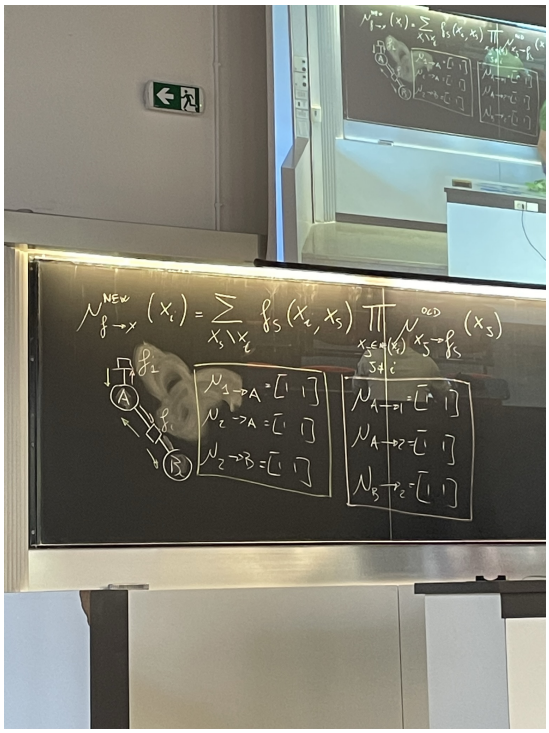
while the marginal probability of $X_2$ having no observation is:

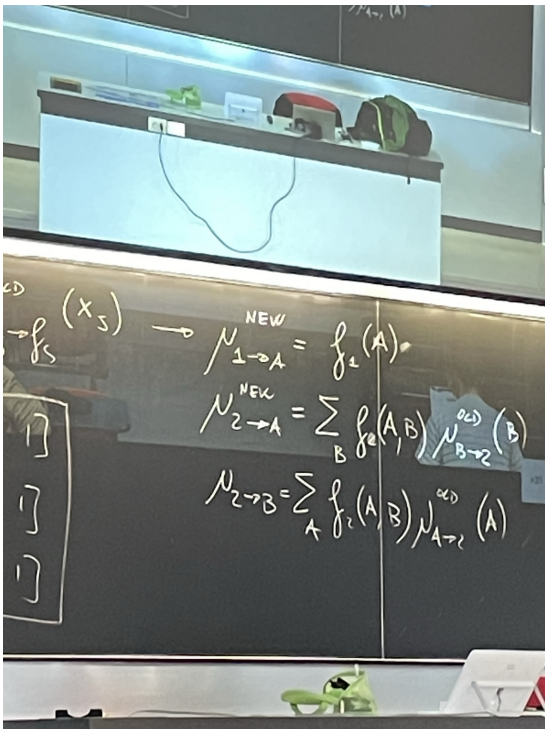
$$P(X_2) = \sum_{X_1} P(X_1)\,P(X_2|X_1)\sum_{X_3} P(X_3|X_2)\sum_{X_4} P(X_4|X_3)$$

We can compute it on a tree easily via Belief Propagation:

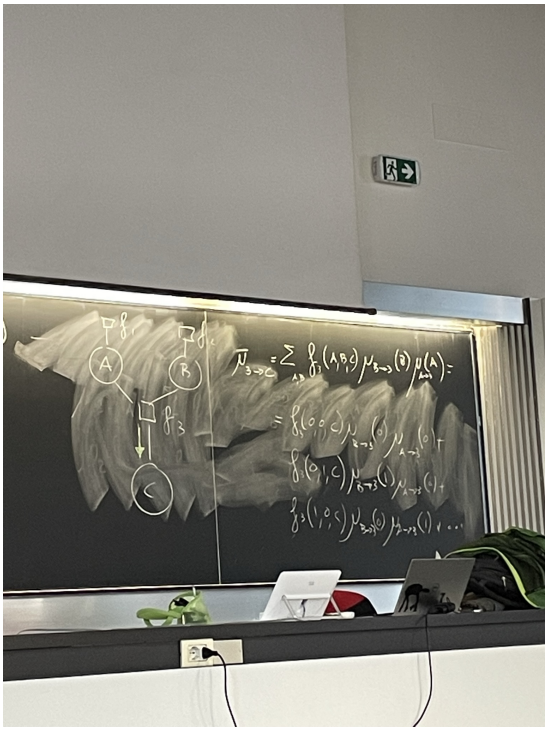
- If some nodes $X_e$ are observed, we use probability of observed values instead of summing over all possible values when computing their messages
- After normalization, this gives the conditional probability given the evidence

NOTATION: letters for variables and numbers for factors





It's uncommon to have factor graphs with cliques bigger than 3 values.



## 13.8 Belief Propagation Update Equations

To understand loops, we can rewrite Belief Propagation updates in an alternative way, reversing the order of steps.

- Initialize all messages as one $\mu_{f_s \to X_i} = 1$, $\mu_{X_i \to f_S} = 1$
- Update believes

$$b_{f_S}^{new}(X_S) = f_S(X_S) \prod_{X_j \in f_S} \mu_{X_j \to f_S}^{new}(X_j)$$

$$b_{X_i}^{new}(X_i) = \prod_{f_S \in ne(X_i)} \mu_{f_S \to X_i}^{new}(X_i)$$

Iterate until convergence

- Update messages

$$\mu_{f_s \to X_i}^{new}(X_i) = \frac{1}{\mu_{X_i \to f_s}^{old}(X_j)} \sum_{X \setminus X_i} b_{f_s}^{old}(X_S)$$

$$\mu_{X_i \to f_s}^{new}(X_i) = \frac{1}{\mu_{f_s \to X_i}^{old}(X_i)} b_i^{old}(X_i)$$

Converges to Marginal Consistency!

First update the believes and then update the messages. Iterating the formulas we converge at the end to the marginal consistency *Marginal Consistency*: if the network is a tree, the formula converge to a less strong equilibrium.
It does not mean that the algorithm converges in the right solution, it depends on the structure of the network.

Given two cliques (= complete subgraph of a network) C and D sharing a variable $X_i$, then their marginal beliefs should coincide:

$$b(X_i) = \sum_{X_c \setminus X_i} b(X_c) = \sum_{X_D \setminus X_i} b(X_D)$$

This property is called **Marginal Consistency**. It also implies

$$b(X_i) = \mu_{C \to i}(X_i) \mu_{i \to C}(X_i)$$

It is a fixed point of the Belief Propagation updates → Belief Update does not change the messages.

## Belief Propagation Wrap Up

Belief Propagation, a.k.a, Sum-Product Algorithm with recursive computation of messages, leads to:

- Exact inference on tree (equivalent to Variable Elimination Algorithm)
- Marginal consistency on any graph (including non-trees)

Marginal consistency means:

- On trees, the parallel update of messages will converge to the true messages
- On polytrees, it will converge to the true messages as well
- On non-trees, it just reaches a state of marginal consistency

We use it for non-tree graphs, but ...

Marginal Consistency → if the network is a tree or a polytree: we converge to an exact solution if the network has loops, it won't converge to an exact solution, because there will be multiple solutions. We can apply Belief Update equation to loopy graphs but having loops, branches of nodes to not represent independent information.

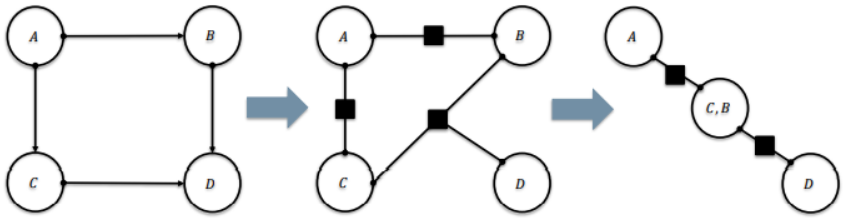
## 13.9 Junction Tree

Algorithm composed by two steps:

1. Transformation of Bayesian Network with loops into a tree (performing a Variable Elimination, keeping track of the messages)

2. Belief Propagation

If the model has loops, to convert it into a tree you have to define variable groups (*separator*) on which messages are defined.



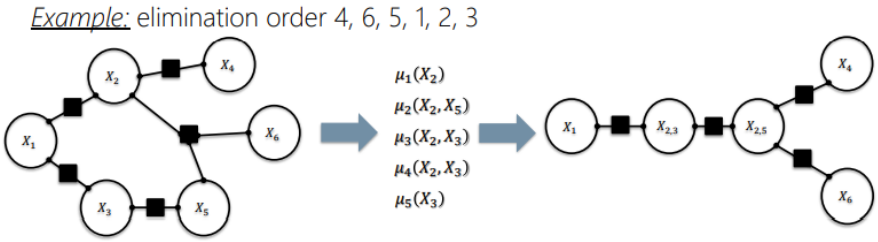

A variable substitution, where we rename $(B, C) = E$. A random variable may be part of multiple separators, but only along a running intersection.

We join variables to obtain a tree to to inference to apply belief propagation

**How we do this transformation?**

**Clique**: fully connected subset of nodes in a graph. We start with a factor graph, we choose an order of variable elimination and we keep track of the mu terms, variables they depend on identify the separators.

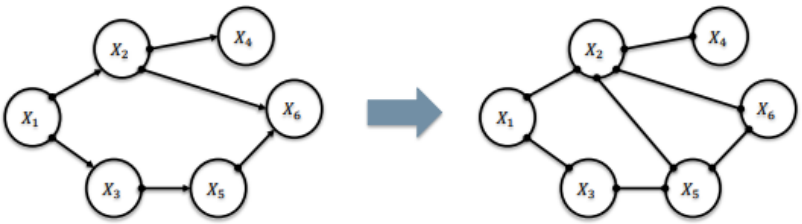
**Example**



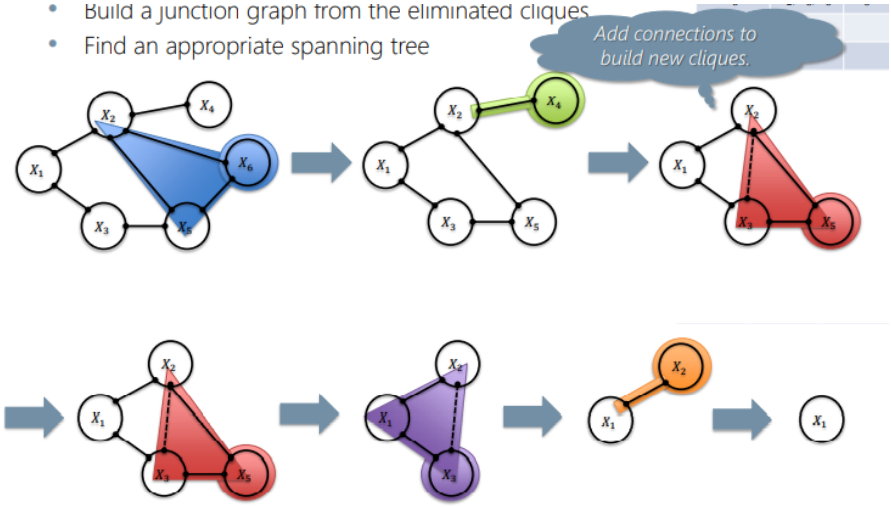

This graph has a loop, we start eliminating variables that have few factors ($X_4$ has only one variable connected). Now the graph is a tree.
Same variable can appear in two different messages.
If you select the wrong order, you will end up in an exponential complexity problem.

We are going to study the *Junction Tree via Moralization and Triangulation*

1. Moralize the graph if directed



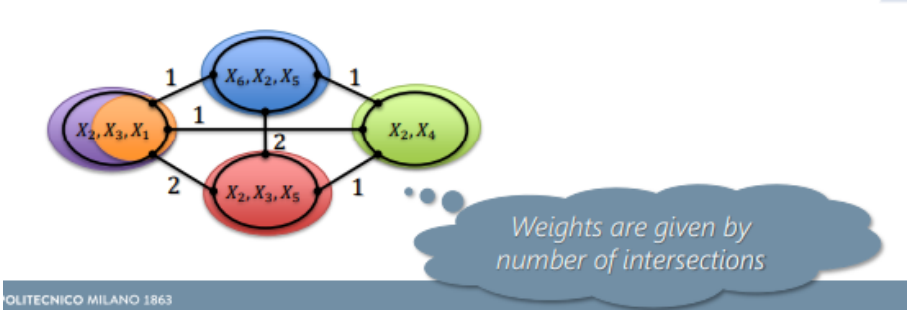
2. Choose a node ordering and find the cliques generated by variable elimination. This gives a triangulation of the graph.
   **Trick**: starting from the biggest clique




removing variables we have created cliques that we can see in the following table:

| Removed | Clique | Added |
|---|---|---|
| $X_6$ | $X_6, X_2, X_5$ | - |
| $X_4$ | $X_2, X_4$ | - |
| $X_5$ | $X_2, X_3, X_5$ | $X_3 - X_2$ |
| $X_3$ | $X_2, X_3, X_1$ | - |
| $X_2$ | $X_2, X_1$ | - |

3. Build a junction graph from the eliminated cliques



4. Find an appropriate spanning tree


Figure 13.11: On the arches we put the number of variables that two nodes share

There could be different junction trees.

To turn the last graph into a junction tree:

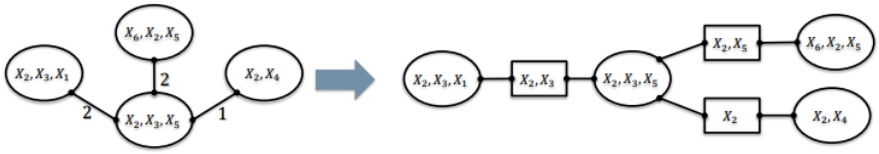
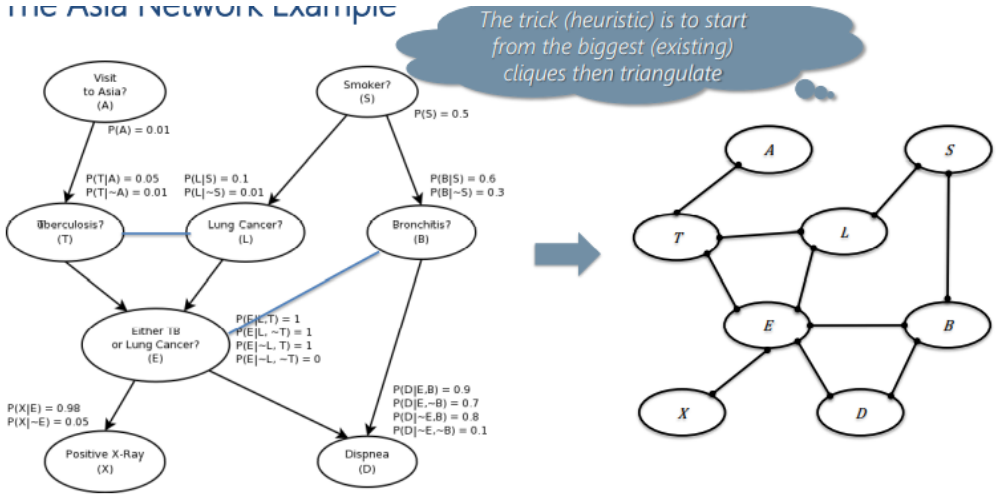


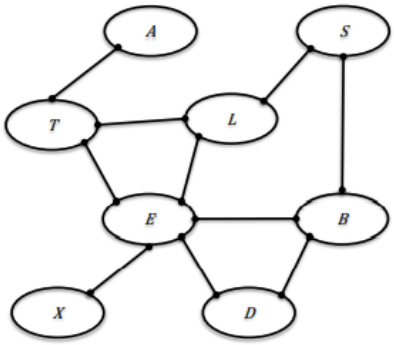

Figure 13.12: Intermediate nodes: those nodes that have variables in common.

Each node can have at most three variables.
Complexity is exponential in the number of variables in the nodes and linear in the number of nodes.

## Asia Network Example



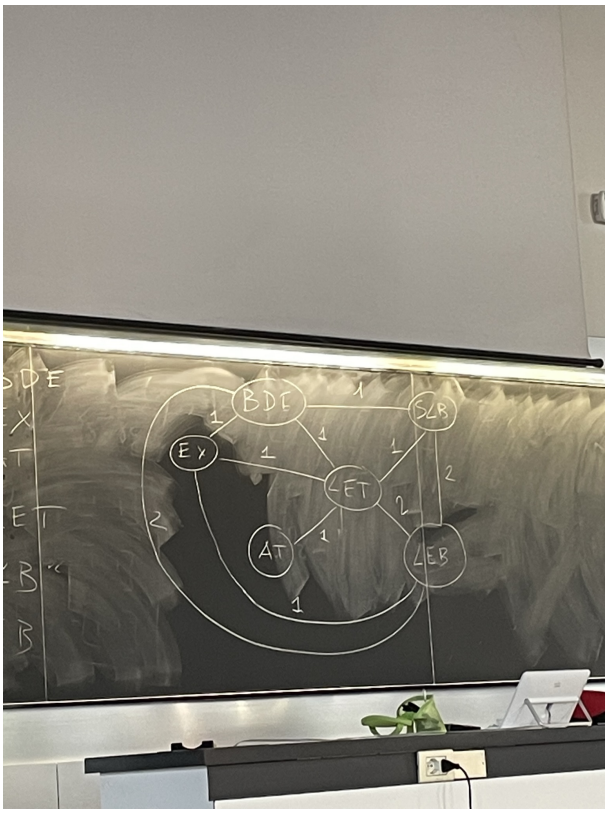

1. Variables that could be eliminated: D (clique of three nodes), X (clique of two nodes) and A (clique of two nodes)




| Removed | Clique | Added |
|---|---|---|
| $D$ | $D, E, B$ | - |
| $X$ | $X, E$ | - |
| $A$ | $A, T$ | - |
| $S$ | $S, L, B$ | $L - B$ |
| $B$ | $B, L, E$ | - |
| $L$ | $L$,E,T | - |
| $T$ | $T, E$ | - |
| $E$ | $E$ | - |

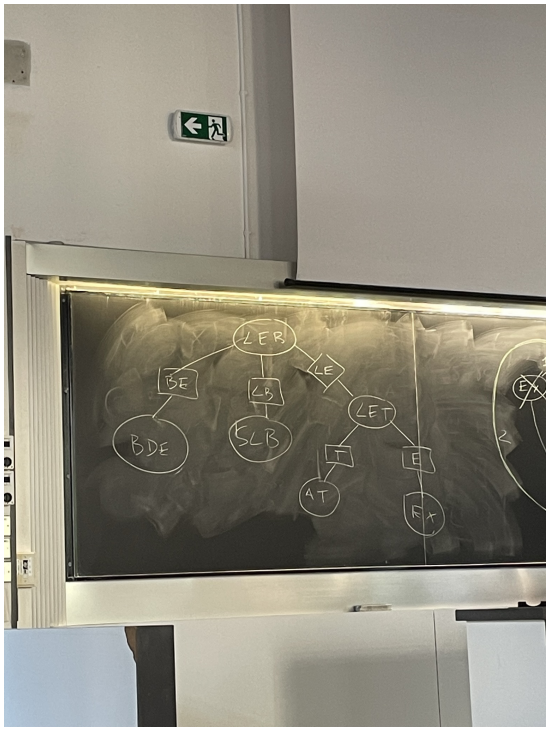
*Starting from the biggest clique*: EBD, we eliminate the node D: D → BDE After eliminating D, we eliminate the node X: X → EX Then I eliminate A: A→ AT (AT was the clique before the deletion). The clique available is ELT, I eliminate T: T → LET; we do not have any more cliques, so we create connections Then we remove S: S→ SLB We can remove any of the remaining L, E or B We decide to remove B: B → LEB

2. Building a graph with all of the cliques, writing the number of variables in common on the arches

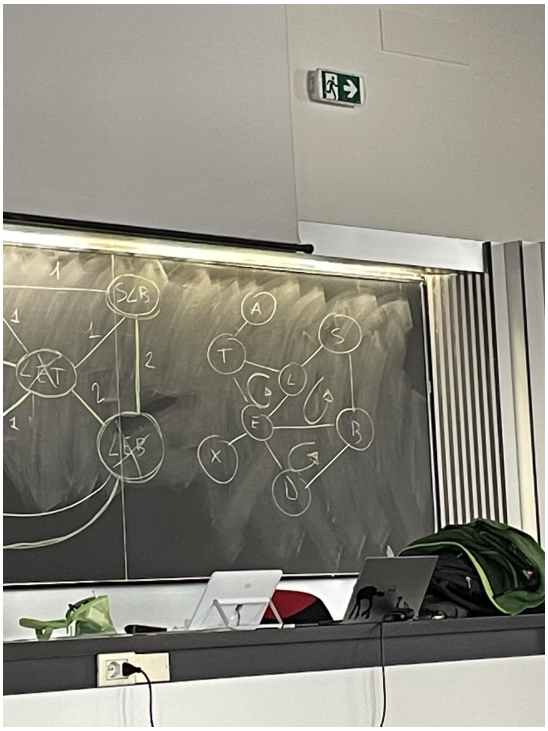



3. Maximun Spanning Tree start taking the links with 2 variables in common.

4. Put everything in a tree starting for example from LEB, then connect LET, they have in common two variables.




(a) the square nodes are the nodes containing common variables




(b) very loopy network

## 13.10 Sampling Based Methods

*To sample*: to get examples from a distribution

Let a Bayesian Network with random variables $X_1, \dots, X_N$, some of which are observed: $X_{obs} = y_{obs}$, $obs \subset \{1,2,\dots,N\}$; our goal is to compute marginal posteriors $P(X_i|X_{obs} = y_{obs})$ conditioned on the observations.

We can generate a set of $K$ (joint) samples $S = \{(x_1, x_2, \dots, x_N)\}_{k=1}^{K}$ being each sample $(x_1, x_2, \dots, x_N)$ a list of instantiations of all $X_1, \dots, X_N$.

Having these samples we can compute

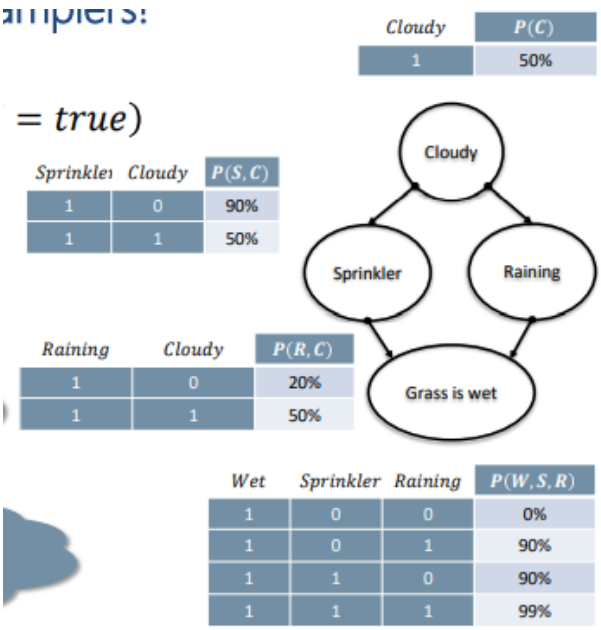
$$P(X_i = x|X_{obs} = y_{obs}) \approx \frac{count_S(x_i^k = x \wedge x_{obs}^k = y_{obs})}{count_S(x_{obs}^k = y_{obs})}$$

Several strategies exists for Sampling

If I have a set of examples, I can perform *approximate inference*: I can compute a posteriori probability by counting how many times the variable happens in my sample fract the number of times the variable happens in all the samples.

Generating samples from Bayesan Network, you could easily perform any inference.

### The Sprinker Example




Start sampling from the root (Cloudy) and going down in depth to compute the probability $P(C|W = True)$

We have the four variables C, S, R, W and we want to generate a sample:
the probability of Cloudy is 50%, so we generate a random number that has a range between 0 and 1 with the same probability (if it is ¡0.5, it is equal to 1 otherwise is equal to 0).
After that, we go below and we see there is the probability of the Sprinkler, that has probability to be 1 of 90% (so a random number generated from 0 to 0,9 is equal to 1, otherwise is 0).
Same actions with Raining with the range 0 and 0,2 for the value 1. And so on.
**Wrap up**: you get a random number and according to the probability given by the initial table of probability, you assign the right value to that variable.

For example, we could have the following sample: C=0, S=1, R=0, W=1 (values obtained by generating random numbers with the threshold of the probability).

## 13.11 Rejection Sampling

Generating how many samples you want until you reach a certain number, starting from the root of the network

To generate a single sample $x_{1:N}^k$:

- Sort all random variables in topological order (i.e., from root to leaves)
- Start with $i = 1$
- Sample a value $x_i^k \sim P\left(X_i | x_{Parents(X_i)}^k\right)$ conditional on $x_{1:i-1}^k$
- If $i \in obs$ compare the sampled value $x_i^k$ with the observation $y_i$; reject and repeat from the previous steps if the sample is not equal to the observation
- Repeat with $i = i + 1$

Having $K$ samples we can compute

*Computers are fast, but for rare events this might waste lot of samples*

$$P(X_i = x | X_{obs} = y_{obs}) \approx \frac{count_S\left(x_i^k = x\right)}{K}$$

Complexity is linear in variables and in the number of the variables; complexity in operation is nothing impor

## 13.12 Sampling

Let a Bayesian Network with random variables $X_1, \dots, X_N$, some of which are observed: $X_{obs} = y_{obs}$, $obs \subset \{1,2,\dots,N\}$; our goal is to compute marginal posteriors $P(X_i|X_{obs} = y_{obs})$ conditioned on the observations.

We can generate a set of $K$ (joint) samples $S = \{(x_1, x_2, \dots, x_N)\}_{k=1}^{K}$ being each sample $(x_1, x_2, \dots, x_N)$ a list of instantiations of all $X_1, \dots, X_N$.

Having these samples we can compute

$$P(X_i = x|X_{obs} = y_{obs}) \approx \frac{count_S(x_i^k = x \wedge x_{obs}^k = y_{obs})}{count_S(x_{obs}^k = y_{obs})}$$

*Several strategies exists for Sampling*

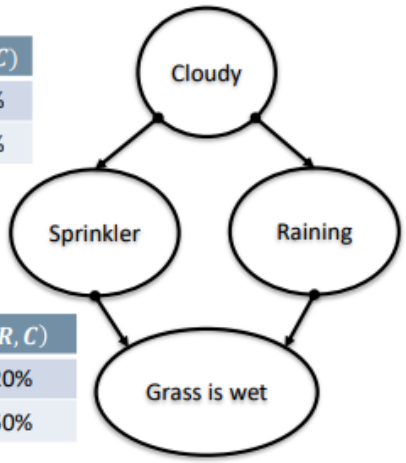
Advantage: it is linear in the number of samples.

### Example with different algorithms (Sprinkler Example)

Let's start sampling to compute $P(C \mid W = true)$.
To compute it, we need "grass is wet" marginalization

| Cloudy | P(C) |
|---|---|
| 1 | 50% |

| Sprinkler | Cloudy | P(S,C) |
|---|---|---|
| 1 | 0 | 90% |
| 1 | 1 | 50% |




| Raining | Cloudy | P(R,C) |
|---|---|---|
| 1 | 0 | 20% |
| 1 | 1 | 50% |

| Wet | Sprinkler | Raining | P(W,S,R) |
|---|---|---|---|
| 1 | 0 | 0 | 0% |
| 1 | 0 | 1 | 90% |
| 1 | 1 | 0 | 90% |
| 1 | 1 | 1 | 99% |

Sampling given the network above with C,S,R,W:

- Generate a number $s \sim U(0,1)$
  - If $s \leq P(C) \triangleright x_C^k = 1$ else $x_C^k = 0$
- Generate a number $s \sim U(0,1)$
  - If $s \leq P(S|x_C^k) \triangleright x_S^k = 1$ else $x_S^k = 0$
- Generate a number $s \sim U(0,1)$
  - If $s \leq P(R|x_C^k) \triangleright x_R^k = 1$ else $x_R^k = 0$
- Generate a number $s \sim U(0,1)$
  - If $s \leq P(W|x_S^k, x_R^k) \triangleright x_W^k = 1$ else $x_W^k = 0$

It might waste a lot of samples because we throw away samples if they contain only 0

# Chapter 14

# Learning Bayesan Networks

To define a Bayesan Network we need to specify:

- Structure: the graph topology
- The parameters of each Conditional Probability Density

We can specify both with the help of experts or learn from data:

- Learning the structure is much harder than learning parameters
- Learning when some of the nodes are hidden or we have missing data is much harder than when everything is observed

There are four cases:

- Known Structure & Full Observability
- Known Structure & Partial Observability
- Unknown Structure & Full Observability
- Unknown Structure & Partial Observability

# Chapter 15

# Dynamic Bayesan Network

## 15.1 Probabilistic Reasoning for Time Series

To describe an ever changing world, we can use a series of random variables describing the world state at any time instant.

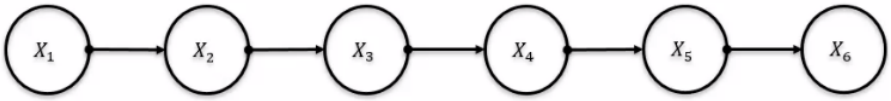



Figure 15.1: 6 timestamps, not 6 different variables

It represents a sequence of states X1, X..., where the number represents the position in the sequence. We assume the transition from $X_{t-1} = x_i$ to $X_t = x_j$ depends only on $X_{t-1}$

**Markov Property**

$$P(X_t \mid X_{t-1}, X_{t-2}, ..., X_0) = P(X_t \mid X_{t-1}$$

## 15.2 Markov Chains

$X_t$: binary random variable at time $t$
In Markov Chain each state transition depends on previous states.

- **Discrete Stochastic Process** describes the relationship between the stochastic description of a system $(X_0, X_1, ..)$ at some discrete time steps.
  It is a **first order Markov Chain** when we have that:

$$\forall t = 1,2,3, \ldots \text{ and for all } N \text{ states it holds:}$$
$$P(X_t|X_{t-1}, X_{t-2}, .., X_0) = P(X_t|X_{t-1})$$

  Whenever the probability of an event is independent from time the Markov Chain is *Stationary*: $P(X_{t+1} = j \mid X_t = i) = p_{ij}$ The probability depend only on the previous and the next value

  In a Stationary Process transition probabilities are the same at any $t$

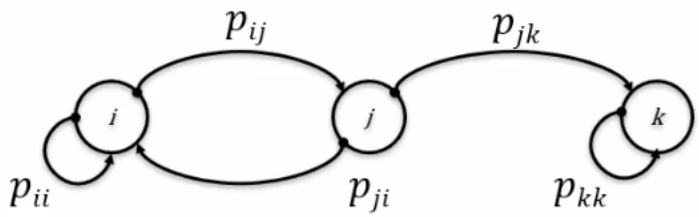
- **Continuous Stochastic Process** is a stochastic process (if we have fixed steps) where the state can be observed at any time.

Markov Chain can be described using a Transition Matrix where $p_{ij}$ describes the probability of getting into state $j$ starting from state $i$

$$P = \begin{bmatrix} p_{11} & \cdots & p_{1N} \\ \vdots & \ddots & \vdots \\ p_{N1} & \cdots & p_{NN} \end{bmatrix}, \qquad \sum_{j=1}^{N} p_{ij} = 1$$

N = number of possible values of my state; not number of variables of my bayesan network

This transition matrix can be described also using a directed graph.
The graph describes how a random variable changes during time




Probability from i to k = 0 different values of the same random variable

## Computing probabilities

Given a Markov Chain in state $i$ at time $m$, states probability after $n$ steps:

$$P(X_{m+n} = j \mid X_m = i) = P(X_n = j \mid X_0 = i) = P_{ij}(n)$$

Probability from state i to state j after n steps

If we take n=2, we have:

$$P_{ij}(n) = \sum_k p_{ik} \cdot_{kj}$$

That is the probability of going to a state j starting from a state i.

In general $P_{ij}(n) = ij^{th}$ element of $P^n$

Probability of being in a given state $j$ at time $n$ without knowing the exact state of Markov Chain at time $0$ is:

$$\sum_i q_i \cdot P_{ij}(n) = q \cdot (column\ j\ of\ P^n)$$

### The Cola example

There are two brands of Cola on the market: Cola1 and Cola2. A person buying Cola1 will buy Cola1 (probability 0.9). A person buying Cola2 will buy Cola2 (probability 0.8).

$$P = \begin{array}{c} \\ Cola_1 \\ Cola_2 \end{array} \begin{array}{c} \begin{array}{cc} Cola_1 & Cola_2 \end{array} \\ \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} \end{array}$$

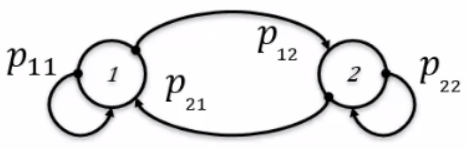



Figure 15.2: Transistion Matrix with the Transition Graph

1. Someone has bought Cola2, how likely she will buy Cola1 after 2 times?

$$P(X_2 = 1 | X_0 = 2) = P_{21}\ (2)$$

$$P(2) = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} = \begin{bmatrix} 0.83 & 0.17 \\ 0.34 & 0.66 \end{bmatrix}$$

2. Someone has bought Cola1, how likely she will buy Cola1 after 3 times?

$$P(X_3 = 1 | X_0 = 1) = P_{11}\ (3)$$

$$P(3) = \begin{bmatrix} 0.83 & 0.17 \\ 0.34 & 0.66 \end{bmatrix} \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} = \begin{bmatrix} 0.781 & 0.219 \\ 0.438 & 0.562 \end{bmatrix}$$

3. Suppose at some time 60% of clients bought Cola1 and 40% Cola2. After 3 purchases what is the percentage of people buying Cola1?

$$\sum_i q_i \cdot P_{ij}(3) = q \cdot (column\ 1\ of\ P^3)$$

$$p = \begin{bmatrix} 0.60 & 0.40 \end{bmatrix} \begin{bmatrix} 0.781 \\ 0.438 \end{bmatrix} = 0.6438$$

## Definitions

Given a Markov Chain we define:

- State $j$ is reachable from $i$ if it exist a path from $i$ to $j$
- States $i$ and $j$ communicate if $i$ is reachable from $j$ and viceversa
- A set of states $S$ is closed if no state outside $S$ is reachable from a state in $S$
- A state $i$ is an absorbing state if $p_{ii} = 1$
- A state $i$ is transient if exists $j$ reachable from $i$, but $i$ is not reachable from $j$
- A state that is not transient is defined as recurrent
- A state $i$ is periodic with period $k > 1$ if $k$ is the biggest number that divides the length of all path from $i$ to $i$, a state that is not periodic is said a-periodic

If all states in a Markov Chain are recurrent, a-periodic, and communicate with each other, it is said to be *Ergodic*

*ERGODIC* = all states of the Markov Chain are recurrent, a-periodic and communicate with each other

## 15.3 A-periodic Markov Chains

A-periodic = if they are not a multiple of some k

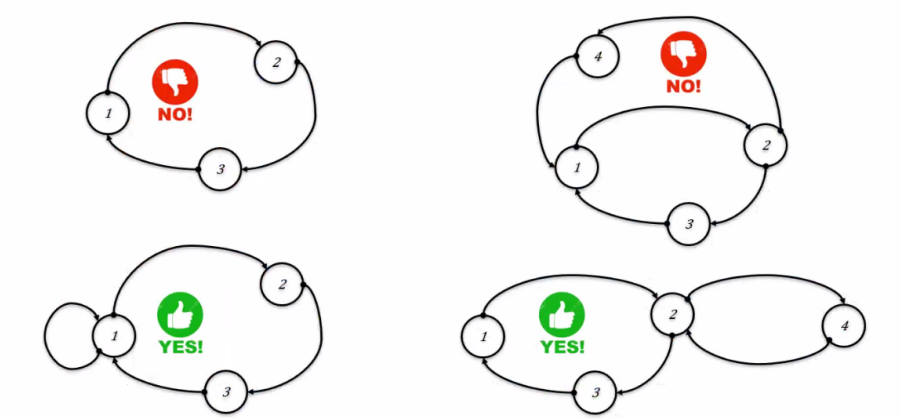
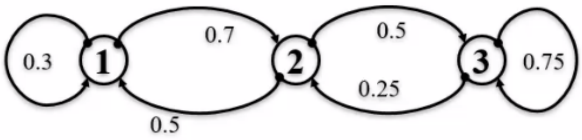



Figure 15.3: top left: it's periodic: all the path have the same length (multiple of three); bottom left: it's a-periodic; top right: all the parts are multiple of 3

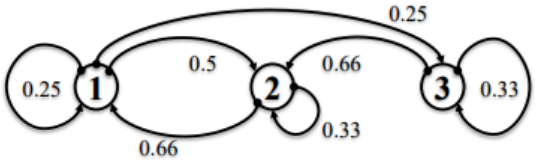
### Examples of Ergodic Markov Chains

They are Ergodic Markov Chains because each variable is connected to the others, for example in the second example, 3 is connected to 1 indirectly but it is connected

$$P = \begin{bmatrix} 0.3 & 0.7 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 0.25 & 0.75 \end{bmatrix}$$




$$P = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 2/3 & 1/3 & 0 \\ 0 & 2/3 & 1/3 \end{bmatrix}$$




## 15.4 Steady State Distribution

Being P the transition matrix of an Ergodic Markov Chain with N states:

$$\lim_{x \to \infty} P_{ij}(n) = \pi_j$$

with $\pi = [\pi_1, \pi_2, ..., \pi_N]$ being the *Steady State Distribution*

*The Cola Example:*

$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$$

$$\lim_{n \to \infty} P(n) = \pi = \begin{bmatrix} 0.67 & 0.33 \\ 0.67 & 0.33 \end{bmatrix}$$

| $n$ | $p_{11}(n)$ | $p_{12}(n)$ | $p_{21}(n)$ | $p_{22}(n)$ |
|---|---|---|---|---|
| 1 | .90 | .10 | .20 | .80 |
| 2 | .83 | .17 | .34 | .66 |
| 3 | .78 | .22 | .44 | .56 |
| 5 | .72 | .28 | .56 | .44 |
| 10 | .68 | .32 | .65 | .35 |
| 20 | .67 | .33 | .67 | .33 |
| 30 | .67 | .33 | .67 | .33 |
| 40 | .67 | .33 | .67 | .33 |

Figure 15.4: the green box is the limit of probabiity, over that n the probabilities remain the same forever

The Steady State is the probability to end up in a page.

## 15.5 Transitory Behaviour

The behaviour of a Markov Chain before getting to the Steady State is defined transitory: we can compute the expected number of transitions to reach state j being in state i for an Ergodic Markov Chain as:

$$m_{ij} = p_{ij}(1) + \sum_{k \neq j} p_{ik} \cdot (1 + m_{kj}) = 1 + \sum_{k \neq j} p_{ik} m_{kj}$$

Number of transitions to go from i to j = probability of going to j in one step + probability of going from i to k + probability of going from k to j

**Example**

How many bottles on average Cola1 buyer will have before switching to Cola2?

$$m_{12} = 1 + \sum_{k \neq j} p_{1k} m_{k2} = 1 + p_{11} m_{12} = 1 + 0.9 * m_{12} = \frac{1}{1 - 0.9} = 10$$

How long do I stay in state 1 before change state?

Viceversa

$$m_{21} = 1 + \sum_{k \neq j} p_{2k} m_{k1} = 1 + p_{22} m_{21} = 1 + 0.8 * m_{21} = \frac{1}{1 - 0.8} = 5$$

No matter in which page you start, you will end up in a certain page.
The Steady State is the probability to end up in that page. (*PageRank*)
PageRank of a page = fraction of steps the surfer spends on it in the limit.
It summarizes the web opinion about the page importance.

## 15.6 Dealing with Absoring States

**Absoring state**: state where you enter and you cannot go out.
For example, the ending of a game, the termination of a game.

Q: transition matrix for transient states
R: transition matrix from transient to absoring states
0 matrix: no arches from absoring states to transient states
1 matrix: from absoring states to absoring states

$$P = \left[\begin{array}{c|c} Q & R \\ \hline 0 & 1 \end{array}\right]$$

### 15.6.1 Inference in Absorbing Markov Chains

How long do I remain in a transient state starting from a transient one?

- Being in a transient state $i$ the average time spent in a transient state $j$ is the $ij^{th}$ element of $(I - Q)^{-1}$

Starting from a transient state, how long does it takes to get to an absorbing one?

- Being in transient state $i$ the probability to get into an absorbing state $j$ is the $ij^{th}$ element of $(I - Q)^{-1} \cdot R$

**Cola example**

$Cola_1$ $0.9 \mid 0.1$
$Cola_2$ $0 \mid 1$
$Q = 0.9 \quad R = 0.1$

First question → formula of transity behaviour $m_{11} = (I - Q)^{-1}$
Second question → formula of probability $m_{11} = (I - Q)^{-1}$

*Example*: in a company there are 3 levels (J, S, P):

- How long does a junior remains in the company?
- What's the probability for a junior to leave the company as partner?

$$P = \begin{array}{c} \begin{matrix} J & S & P & LN & LP \end{matrix} \\ \begin{bmatrix} 0.80 & 0.15 & 0 & 0.05 & 0 \\ 0 & 0.70 & 0.20 & 0.10 & 0 \\ 0 & 0 & 0.95 & 0 & 0.05 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$(I - Q)^{-1} = \begin{bmatrix} 5 & 2.5 & 10 \\ 0 & 3.3 & 13.3 \\ 0 & 0 & 20 \end{bmatrix} \qquad (I - Q)^{-1} \cdot R = \begin{bmatrix} 0.5 & 0.5 \\ 0.3 & 0.7 \\ 0 & 1 \end{bmatrix}$$

How long does a junior remains in the company?

- He/she will stay as Junior: $m_{11} = 5$
- He/she will stay as Senior: $m_{12} = 2.5$
- He/she will stay as Partner: $m_{13} = 10$

} 17.5 *years*

What's the probability for a junior to leave the company as partner?

- He/She will end up in state LP: $m_{12} = 0.5$

## Exercise: Gambler's Ruin

Suppose you start from a 3$ capital. With probability $p = 1/3$ you can win 1$ and with $1 - p = 2/3$ you loose 1$. You succeed if capital gets 5.


- Possible states: 0, 1, 2, 3, 4, 5
- Transition probability: p(Xt+1=Xt+1)=1/3, p(Xt+1=Xt-1)=2/3

What kind of reasoning can we apply to this model?

- What's the probability of sequence 3, 4, 3, 2, 3, 2, 1, 0?
- What's the probability of success for the gambler?
- What's the average number of bets the gambler will make?

Figure 15.5: two absorbing states: 0 and 5; $p(X(t+1) = X(t+1)$

The transition matrix will have 6 columns and one row, it is not ergodic

## 15.7 Hidden Markov Models

It's a dynamic bayesan network.

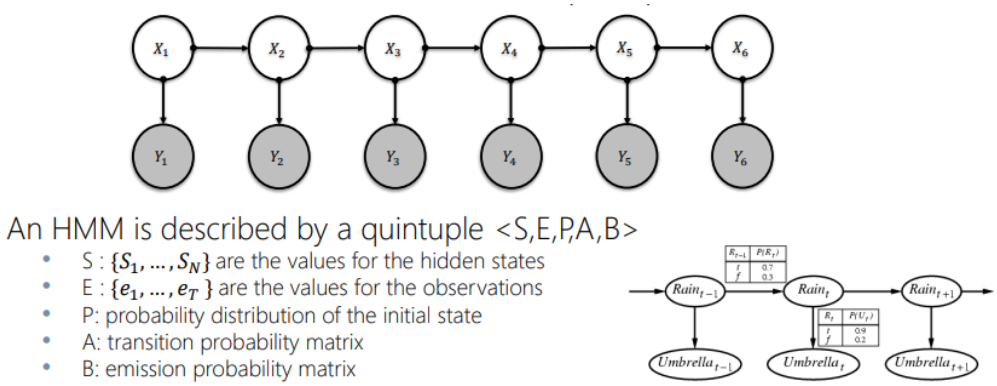
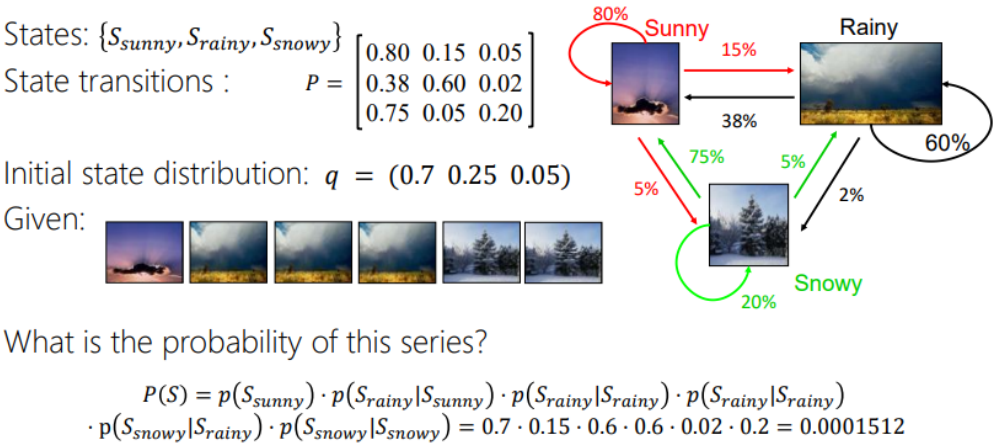



Figure 15.6: Some variables are known, other variables are unknown

HMM is a tree, $x_1$ is the root $\rightarrow$ everything in trees is linear.

**Example**



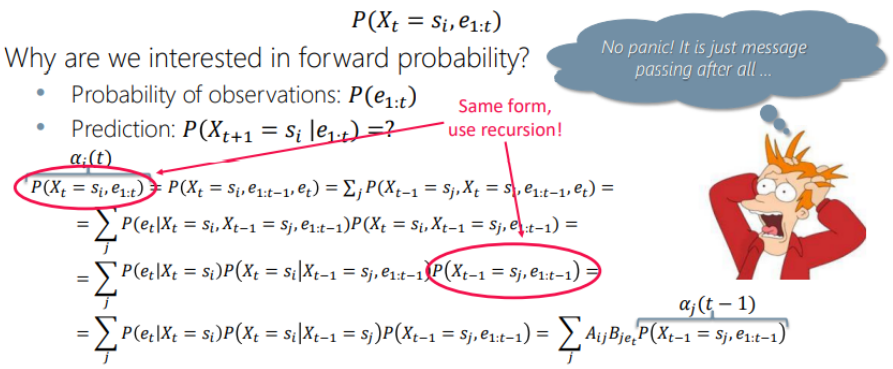

What is the probability of this series?

$$P(S) = p(S_{sunny}) \cdot p(S_{rainy}|S_{sunny}) \cdot p(S_{rainy}|S_{rainy}) \cdot p(S_{rainy}|S_{rainy})$$
$$\cdot p(S_{snowy}|S_{rainy}) \cdot p(S_{snowy}|S_{snowy}) = 0.7 \cdot 0.15 \cdot 0.6 \cdot 0.6 \cdot 0.02 \cdot 0.2 = 0.0001512$$

Probability of series (likelihood of the series) is perfect for anomaly detection.

## 15.8 Forward Probability

It is the join probability of actual state and sequence of the observations. It's not the probability of the sequence.
With this probability, we can predict what it will be the most likely state.




$e_t$ = specific observation at a certain time

## 15.9 Viterbi Algorithm

To show the most likely sequence given a set of observation.

From observations, compute the most likely hidden state sequence:

$$argmax\ P(X_{1:t}|e_{1:t}) = argmax\ P(X_{1:t}, e_{1:t})/P(e_{1:t}) = argmax\ P(X_{1:t}, e_{1:t})$$

By applying the Bayesian Network factorization

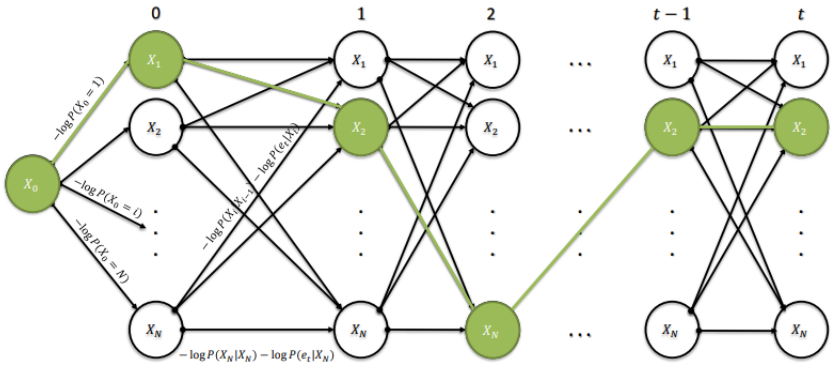
$$P(X_{1:t}, e_{1:t}) = P(X_0) \prod_{i=1:t} P(X_i|X_{i-1}) P(e_i|X_i)$$

The solution we are looking for is the one that minimizes

$$-\log P(X_{1:t}, e_{1:t}) = -\log P(X_0) + \sum_{i=1:t} (-\log P(X_i|X_{i-1}) - \log P(e_i|X_i))$$

Construct a graph that consists $1 + t \cdot N$ nodes, one initial node and $N$ node at time $i$ where $j^{th}$ represents $X_i = s_j$.

Goal: want to minimize the length of the path from the initial node to the last node → shortest path.




Most likely sequence of states → find the shortest path in the graph, using for example Dijkstra