COMP9418 – Advanced Topics in Statistical Machine Learning

W2 – Exact Inference In Graphical Models

Instructor: Edwin V. Bonilla

School of Computer Science and Engineering

August 2nd, 2017

(Last Update: August 2nd at 11:40, 2017)

Admin

Please post all your questions on the course forum. I will not reply individual emails.

Acknowledgements

Material derived from:

- [Murphy, MLaPP, 2012] Machine Learning: A Probabilistic Perspective, Kevin P. Murphy, 2012
- [Barber, BRML, 2012] Bayesian Reasoning and Machine Learning, David Barber, 2012
- [Bishop, PRML, 2006] Pattern Recognition and Machine Learning, Christopher Bishop, 2006
- [Domke, MLSS, 2015] Slides from Justin Domke, Machine Learning Summer School, Sydney, 2015
- [Bonilla, PML, 2015] Slides from course to industry, Edwin V. Bonilla, Practical Machine Learning © 2010 – 2017

Aims

This lecture will allow you to understand probabilistic directed graphical models. Following it you should be able to:

- Define statistical (marginal) independence and conditional independence
- Differentiate between directed graphical models and undirected graphical models
- Apply the concept of d-separation to establish conditional independences in a directed graphical model
- Carry out marginal and conditional marginal calculations using variable elimination
- Apply the junction tree algorithm for exact probabilistic inference

Outline

I. Statistical Independence

II. Probabilistic Graphical Models
   1. Directed and undirected graphical models
   2. Main tasks in graphical models

III. Conditional Independence in Directed Graphical Models
   1. The Bayes' ball algorithm

IV. Exact Inference Via Variable Elimination

V. The Junction Tree Algorithm

I. Statistical Independence

Statistical Independence

In our fruit-box example, suppose that both boxes (red and blue) contain the same proportion of apples and oranges, say:

$$P(F = a|B = r) = P(F = a|B = b) = 0.2$$
$$P(F = o|B = r) = P(F = o|B = b) = 0.8$$

The probability of selecting an apple (or an orange) is independent of the box that is chosen *show that if the above is true: P(F,B) = P(F)P(B)*

Independent Variables

Two variables X and Y are statistically independent iff their joint distribution factorises into the product of their marginals:

$$X \perp\!\!\!\perp Y \leftrightarrow P(X, Y) = P(X)P(Y)$$

This definition generalises to more than two variables

Statistical Independence

Homework (for fun, submission not required)

1. Consider two binary variables X , Y $\in$ {0, 1} which always happen to take on the value 1 jointly:

$$P(X=0,Y=0)=0 \quad P(X=0,Y=1)=0$$
$$P(X=1,Y=0)=0 \quad P(X=1,Y=1)=1$$

Are these variables statistically independent?

2. Is statistical independence transitive?

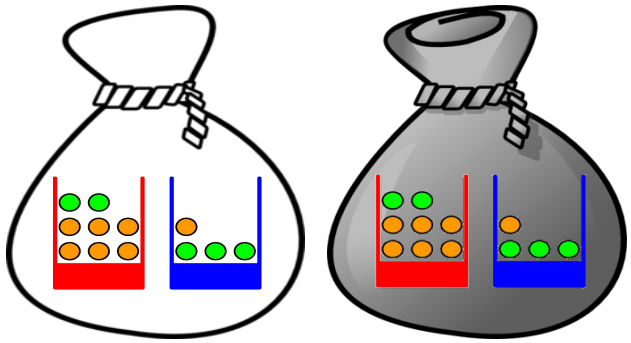
If $X \perp\!\!\!\perp Y$ and $Y \perp\!\!\!\perp Z$, what can you say about $X \overset{?}{\perp\!\!\!\perp} Z$

HINT: Consider the following distribution:

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

Conditional Independence

- In our fruit box example, imagine that we have to pick a box from a sack (S) randomly:
  1. Pick a sack ($S \in \{w, g\}$)
  2. Pick a box ($B \in \{b, r\}$)
  3. Pick a fruit ($F \in \{o, a\}$)



- If we know the identity of the box, knowing the identity of the sack does not help us predict the fruit being picked:
  - $P(S, F \mid B) = P(S \mid B)\, P(F \mid B)$

Conditionally Independent Variables

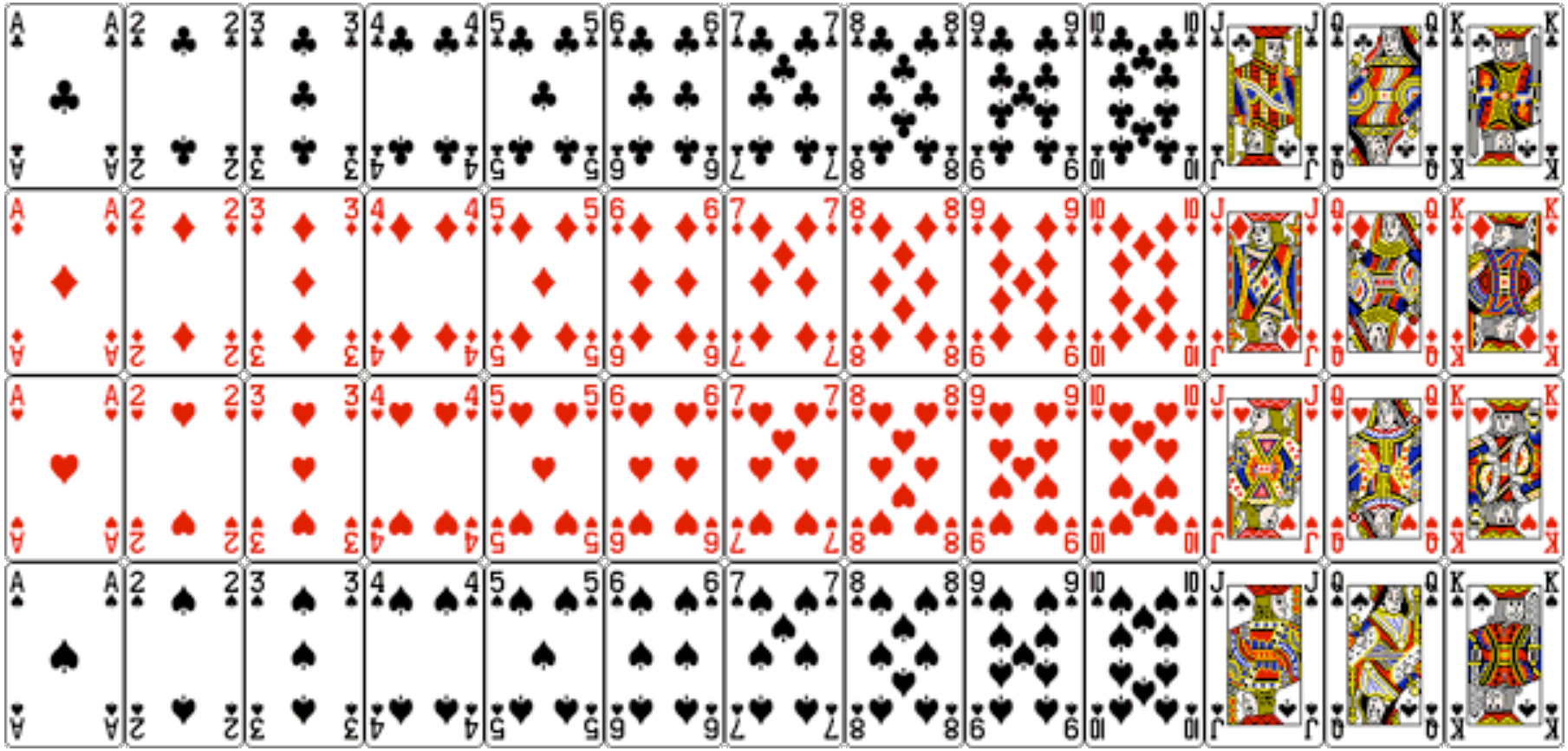
X and Y are conditionally independent given Z iff their conditional distribution factorises:

$$X \perp\!\!\!\perp Y|Z \Leftrightarrow P(X, Y|Z) = P(X|Z)P(Y|Z)$$

Conditional Independence

Example 1 of 3 (Domke, MLSS, 2015)

Is Value independent of Colour?

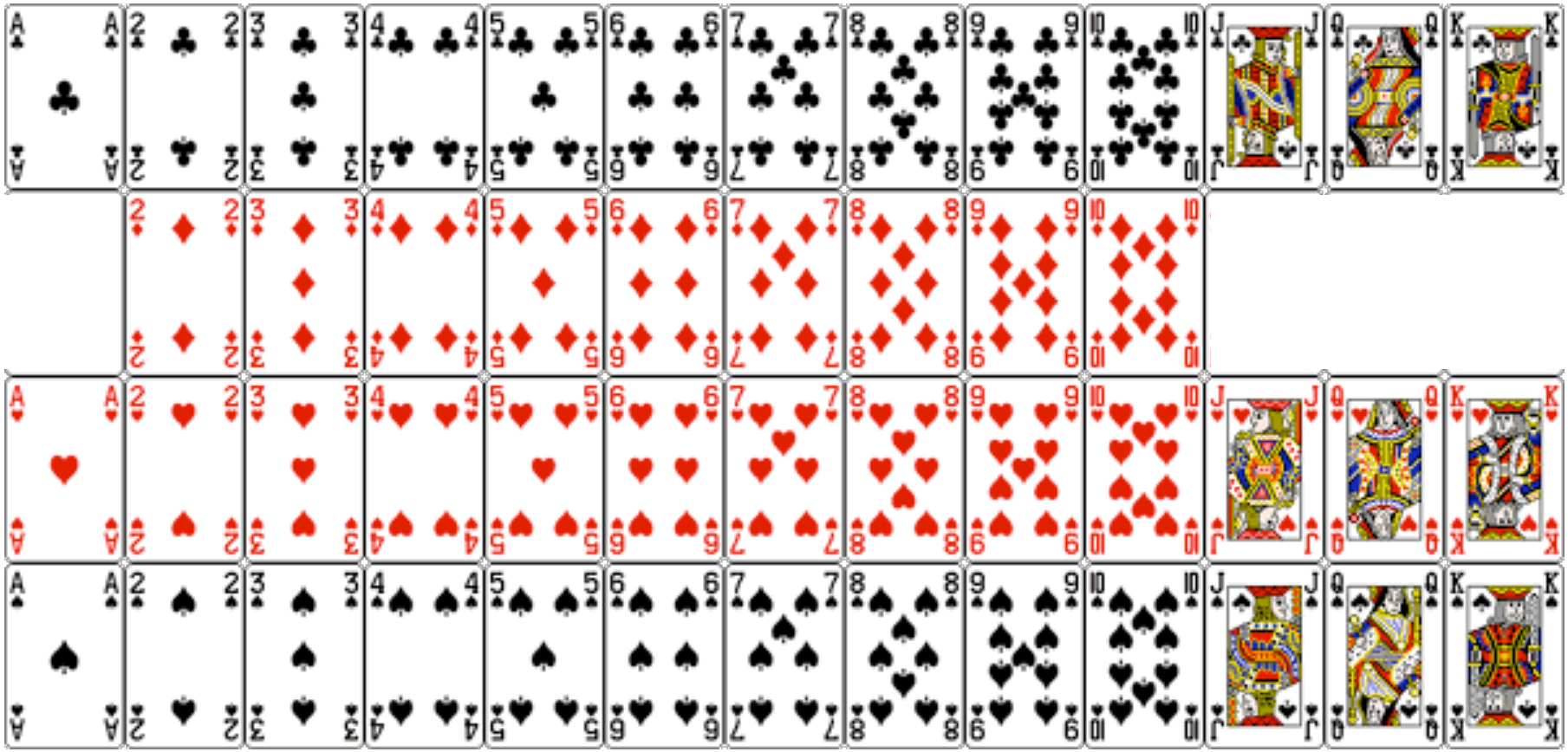


$$Value \perp\!\!\!\perp Colour$$

Conditional Independence

Example 2 of 3 (Justin Domke, MLSS 2015)

Is Value independent of Colour?

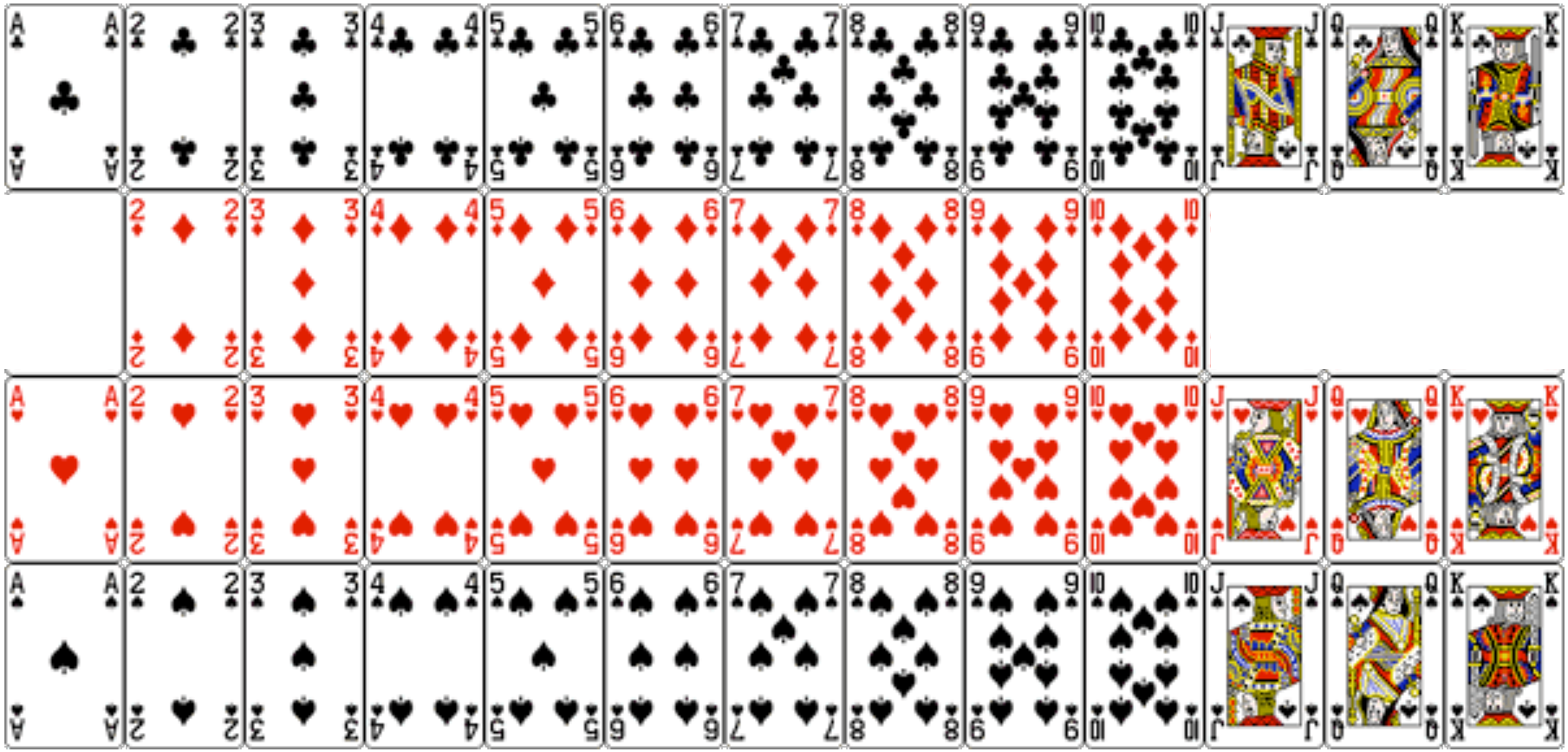


$$Value \not\perp Colour$$

Conditional Independence

Example 3 of 3 (Domke, MLSS, 2015)

Is Value independent of Colour given Face card?



$$Value \perp\!\!\!\perp Colour | Facecard$$

II. Probabilistic Graphical Models

The Rules of Probability and Terminology

- Sum Rule: $P(X) = \sum_{Y} P(X, Y)$
- Product Rule: $P(X, Y) = P(Y|X)P(X)$
- By symmetry: $P(Y, X) = P(X, Y)$

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

- Normalisation
- Marginalisation

Joint, Marginal and Conditional Probabilities
A General Formulation

Given D random variables $X_1, \ldots, X_D$

- Marginal: $P(X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, x_D) = \sum_{X_i} P(X_1, \ldots, X_D)$

- Chain rule:

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

What are we using here?

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

$$P(X_1, \ldots, X_D) = P(X_1) \prod_{d=2}^{D} P(X_d|X_{1:d-1})$$

  - Note order above is somewhat arbitrary.

Probabilistic Models

The Curse of Dimensionality

By the chain rule we can always represent a joint distribution over D variables as:

$$P(X_1, \ldots, X_D) = P(X_1) \prod_{d=2}^{D} P(X_d | X_{1:d-1})$$

Assuming $X_d \in \{1, \ldots, K\}$:

- How may parameters (i.e. probabilities) do we need to represent this distribution? $\mathcal{O}(K^D)$
- Problems in inference
  - Marginal, conditional distribution computation (many summations!)

Problems in learning

- Parameter estimation (need a lot of data to learn so many parameters)

Key Idea: Represent distributions efficiently by making conditional independence assumptions

Probabilistic Models

The Curse of Dimensionality – Example

Consider a distribution over D=100 binary variables: $X_d \in \{0, 1\}$

- $P(X_1, \ldots, X_{100})$ has 1267650600228229401496703205375 parameters!
- Naive Computation of $P(X_{100}=1|X_{99}=1)$:

  → 316912650057057350374175801344 summations

If these variables follow a first-order Markov chain:

$$P(X_1, \ldots, X_{100}) = P(X_1) \prod_{d=2}^{100} P(X_d|X_{d-1})$$

- Main assumption: The future is independent of the past given the present
- How may parameters? 199
- Computation of $P(X_{100} = 1 \mid X_{99}=1)$ is straightforward (table lookup)

Probabilistic graphical models exploit this type of structure in distributions

Probabilistic Graphical Models

Conditional independences yield factorised distributions

Graphical Model (GM)

A graphical model is a way to represent a joint distribution by making conditional independence assumptions.

- More efficient parameterisation (e.g. more compact CPTs)
- More efficient inference

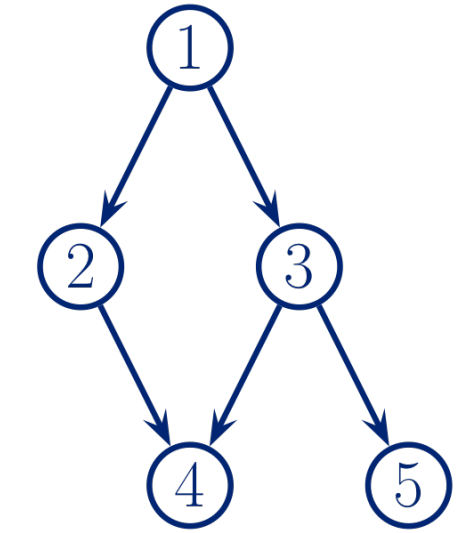
$$P(X_1, X_2, X_3, X_4) = P(X_1)P(X_2|X_1)P(X_3|X_2)P(X_4|X_3)$$

$$P(X_1, X_2, X_3, X_4) \propto \psi(X_1, X_2)\psi(X_2, X_3)\psi(X_3, X_4)$$

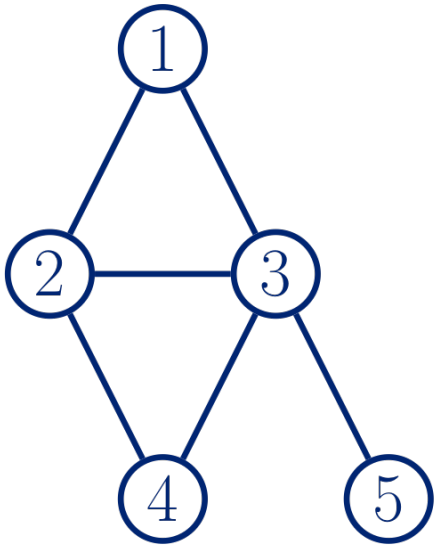
- Intuition: To predict the value of one variable it is sufficient to know the value of some neighbouring variables
- Instead of using all other variables

Graph Terminology

- Graph: a set of nodes and edges
  - Directed or undirected
- Parents and children of a node
  - Feed into and out of it
- Root node
- Neighbours of a node
  - All immediately connected nodes



Directed acyclic graph (DAG)



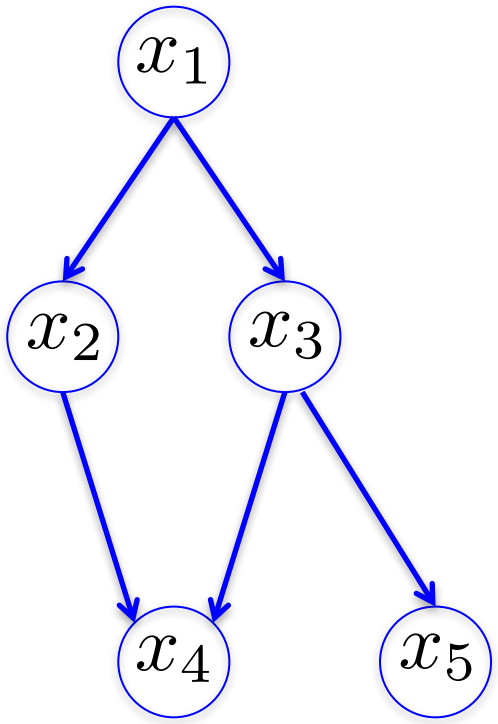
Undirected graph

Fig. from Murphy (MLaPP, 2012)

- Cycle or loop
- Directed acyclic graph: A DG with no directed cycles
- Clique: A set of nodes that are all neighbours of each other
- Maximal clique: Set of nodes that cannot be made any larger without loosing the clique property

Directed Graphical Models (DGMs)

- Represent joint distributions as the product of local conditional distributions
- Directed acyclic graph (DAG)



- Each of these conditionals is, by definition, already normalised

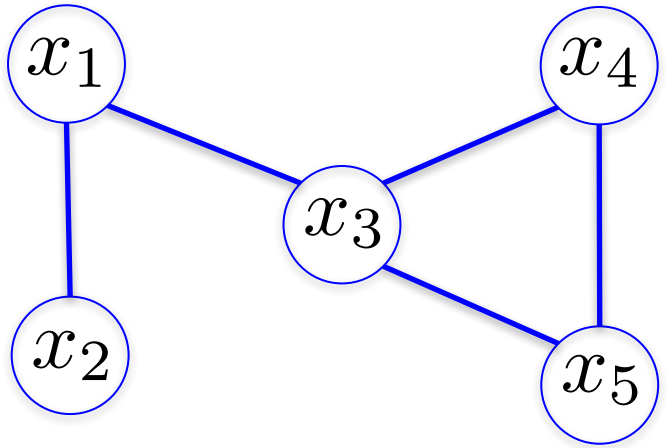
Hence the joint is readily normalised

$$P(X_1, X_2, X_3, X_4, X_5) = P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|X_2, X_3)P(X_5|X_3)$$

- Also known as Bayesian networks (Bayes / belief nets)
  - Examples in medical diagnosis, speech processing, tracking, etc

Undirected Graphical Models (UGMs)

- Represent joint distributions as product of potentials on cliques
- Undirected graphs



- Local potentials have no probabilistic interpretation
- Unnormalised

- Need to ensure global normalisation

$$P(X_1, X_2, X_3, X_4, X_5) = \frac{1}{Z}\psi(X_1, X_2)\psi(X_1, X_3)\psi(X_3, X_4, X_5)$$

$$\text{with } Z = \sum_{X_1,\ldots,X_5} \psi(X_1, X_2)\psi(X_1, X_3)\psi(X_3, X_4, X_5)$$

- Also known as Markov Random Fields (MRFs)
  - Applications in computer vision, natural language processing, etc

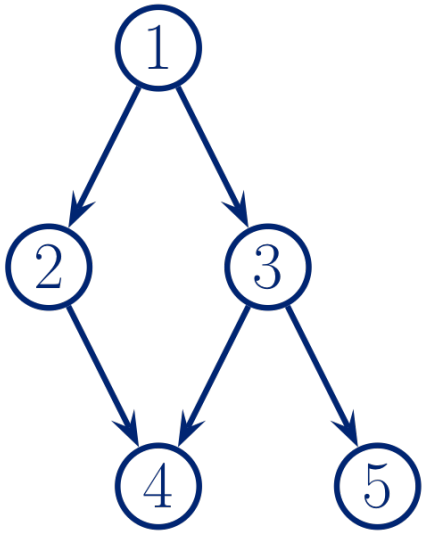
Key Property of Directed Acyclic Graphs (DAGs)

Nodes can be ordered such that parents come before the children

- $\pi(i)$: Set of parents nodes for each node *i*
- Then

$$\forall j \in \pi(i), j < i$$

$$\pi(1) = \{\}$$
$$\pi(2) = \{1\}$$
$$\pi(3) = \{1\}$$
$$\pi(4) = \{2, 3\}$$
$$\pi(5) = \{3\}$$

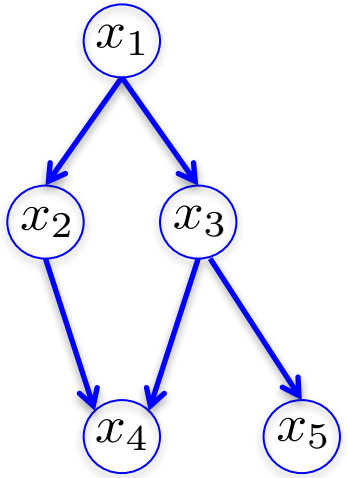


- This is called a topological ordering and can be constructed from any DAG

Key Property of Directed Graphical Models

- Ordered Markov property: A node only depends on its immediate parents (not all its predecessors in the ordering)

$$X_i \perp\!\!\!\perp X_{\text{pred}(i)\setminus\pi(i)} \,|\, X_{\pi(i)}$$



- Where $\text{pred}(i)$: Predecessors of node *i* in the ordering
  $\pi(i)$: Parents of node *i*
  \ : Excluding operator

$$P(\mathbf{X}_{1:5}) = P(X_1)P(X_2|X_1)P(X_3|X_1,\cancel{X_2})P(X_4|\cancel{X_1},X_2,X_3)P(X_5|\cancel{X_1},\cancel{X_2},X_3,\cancel{X_4})$$
$$= P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|X_2,X_3)P(X_5|X_3)$$

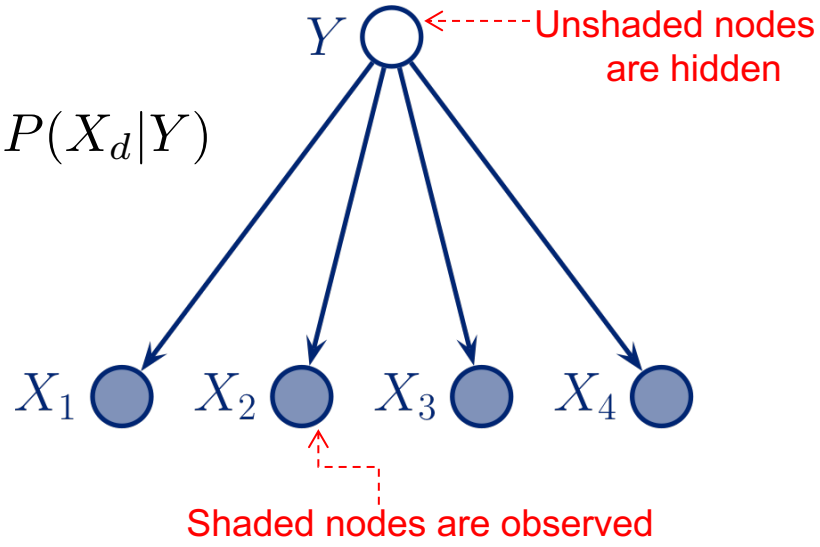
In general for a DGM: $P(\mathbf{X}_{1:D}) = \prod_{d=1}^{D} P(X_d|X_{\pi(d)})$

- So conditional independence yields this explicit factorisation
- If each node has O(F) parents and K states → $O(DK^F)$ parameters

Examples of Directed Graphical Models (DGMs)

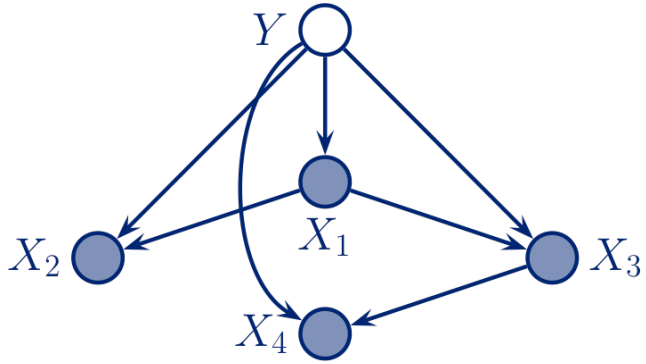
Naïve Bayes

- Naïve Bayes classifier
  - Generative model $P(Y, \mathbf{X}) = P(Y) \prod_{d=1}^{D} P(X_d|Y)$
  - Model P(Y) and P(**X**|Y)
  - Predictions using Bayes' rule
  - Features conditionally independent given the class label



- We can capture dependencies between features by a using a more "flexible" graphical model

- Tree-augmented Naïve Bayes
  - P(X|Y) has a tree structure
  - Trees are nice
  - Optimal structure (Chow-Liu algorithm)



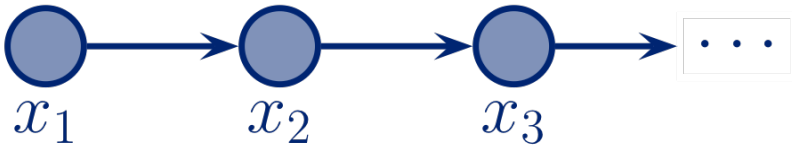
Examples of Directed Graphical models (DGMs)

Markov models and Hidden Markov Models (HMMs)

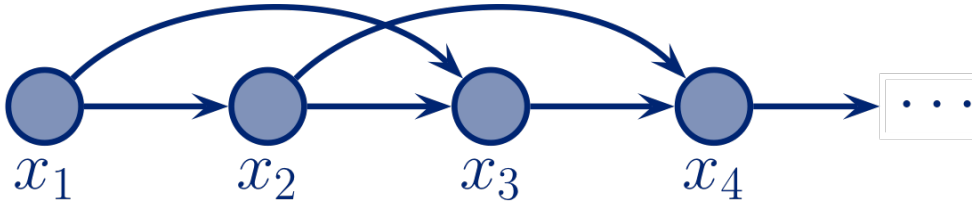
Modelling sequential data

- Temporal: Financial forecasting, speech, object tracking
- Non-temporal: Predictive texting, DNA analysis

- First-order Markov chain
  - Only immediate past matters

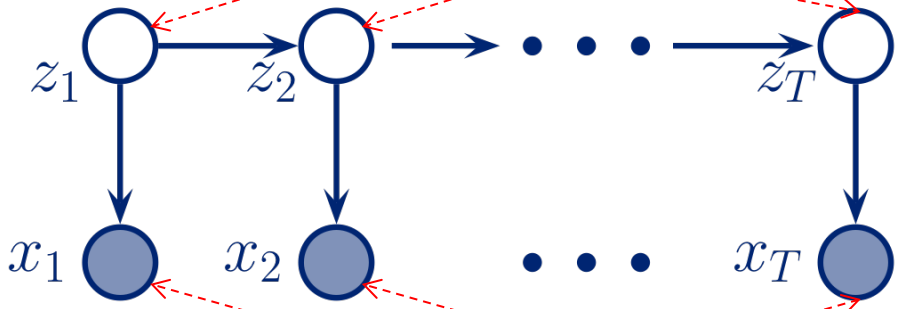


- Second-order Markov chain
  - Accounts for longer dependencies
  - # Parameters blows up in higher-order models



Quantities of interest: e.g. identity of the word

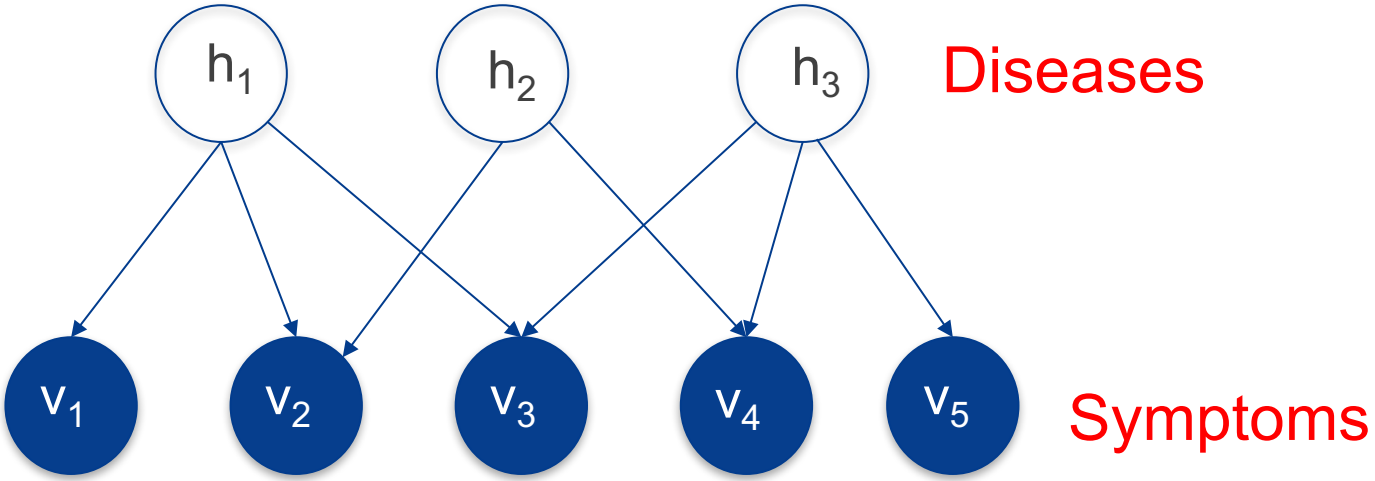
- Hidden Markov models
  - Modelling long-range dependencies across observations



What we measure: e.g. Acoustic waveform

Examples of Directed Graphical models (DGMs)
Medical Diagnosis

Quick medical reference network (Shwe et al, 1991)



$$P(\mathbf{v}, \mathbf{h}) = \prod_s P(h_s) \prod_t P(v_t | \mathbf{h}_{\pi(t)})$$

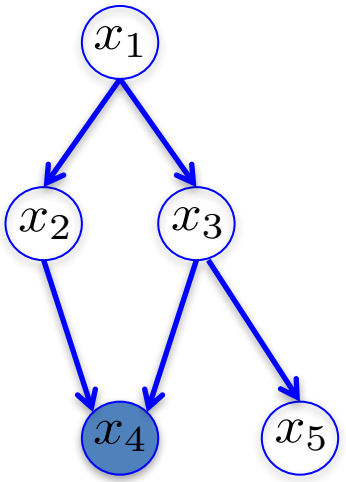
- Model infectious diseases
- Binary nodes
- Full network contains 570 hidden nodes and 4075 visible modes

Main Tasks in Graphical Models

Inference

- Probabilistic inference: Estimating unknown quantities from known quantities:
- Given X_4, what is the the distribution over the *latent* variables?

$$P(X_1, X_2, X_3, X_5 | X_4 = x_4) = \frac{P(X_1, X_2, X_3, X_4 = x_4, X_5)}{P(X_4 = x_4)}$$



- Given X_4, what is the distribution over X_2?

$$P(X_2 | X_4 = x_4) = \frac{P(X_2, X_4 = x_4)}{P(X_4 = x_4)}$$

- In general, we are interested in:

query visible uninteresting

$$P(\mathbf{X}_q | \mathbf{X}_v) = \sum_{\mathbf{X}_u} P(\mathbf{X}_q, \mathbf{X}_u | \mathbf{X}_v) = \frac{\sum_{\mathbf{X}_u} P(\mathbf{X}_q, \mathbf{X}_u, \mathbf{X}_v)}{\sum_{\mathbf{X}_{u'}, \mathbf{X}_{q'}} P(\mathbf{X}_{q'}, \mathbf{X}_{u'}, \mathbf{X}_v)}$$

hidden

  - Naively, $O(K^D)$ in time → We can do much better by exploiting structure

Main Tasks In Graphical Models

Learning

- Where do all those probability tables come from?

- In fact, our distributions are all conditioned on those tables (parameters): $P(X_1, \ldots, X_D|\boldsymbol{\theta})$

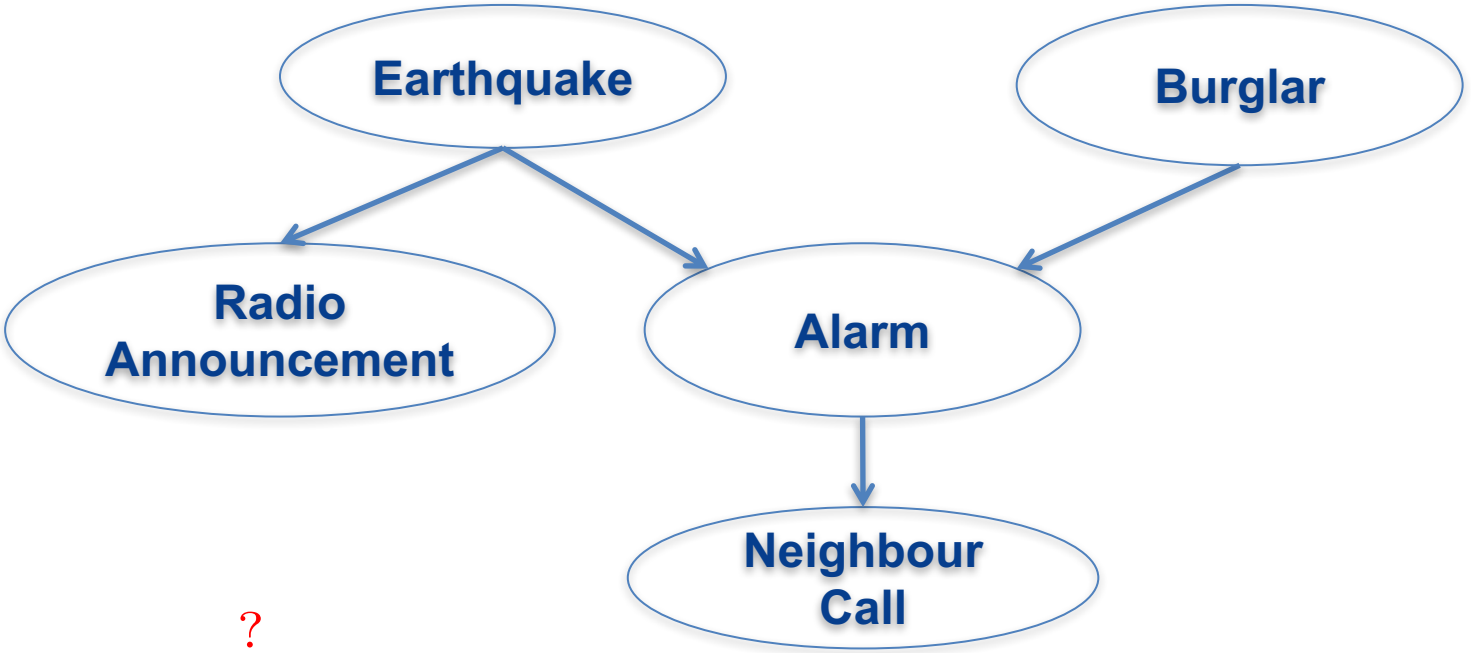
- Learning is the estimation of these parameters $\boldsymbol{\theta}$
  - Usually done through maximum a posteriori (MAP) estimation
  - For a uniform prior $P(\boldsymbol{\theta})$, it reduces to maximum likelihood estimation (MLE)

- For a Bayesian, there is no difference between inference and learning
  - $\boldsymbol{\theta}$ are simply random (latent) variables to be *inferred*
  - As usually there are only fewer parameters than other latent variables, we can get away with point estimation for $\boldsymbol{\theta}$

III. Conditional Independence in Directed Graphical Models

Conditional Independence Properties in DGMs

Example: Burglar Alarm Network



$$\text{Alarm} \overset{?}{\perp\!\!\!\perp} \text{Radio} \quad \text{NO}$$

$$\text{Alarm} \overset{?}{\perp\!\!\!\perp} \text{Radio} \mid \text{Earthquake} \quad \text{YES}$$

$$\text{Earthquake} \overset{?}{\perp\!\!\!\perp} \text{Burglar} \quad \text{YES}$$

$$\text{Earthquake} \overset{?}{\perp\!\!\!\perp} \text{Burglar} \mid \text{Alarm} \quad \text{NO}$$

Conditional Independence and D-separation

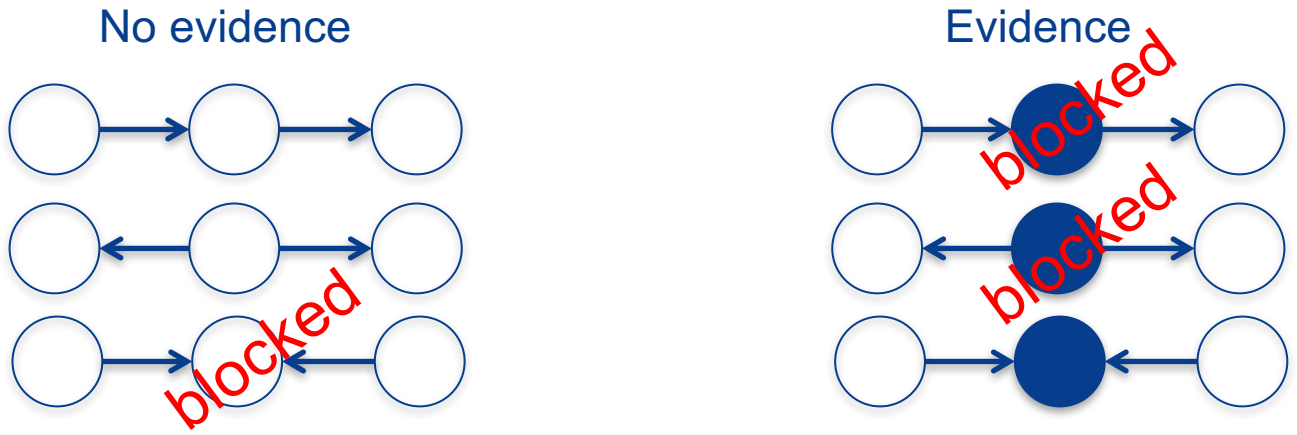
Slide from Domke (MLSS, 2015)

Given a directed model, how to tell if some variable X_i is conditionally independent of some variable X_j given X_A ?

The Bayes ball algorithm:

Take the graph, color all nodes A, start a ball at *i*

Bounce the ball around the graph with the following rules:

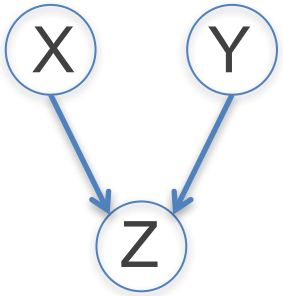


- If you <u>can</u> hit j, they are not conditionally independent.
- If it is <u>impossible</u> to hit j, they are conditionally independent.
  - We say that X_i is d-separated from X_j given X_A if **all** the paths are blocked

Conditional Independence and D-separation

Explaining away

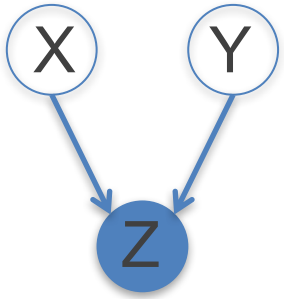
Marginally independent



$$X \perp\!\!\!\perp Y$$

Collider or v-structure

Conditionally dependent

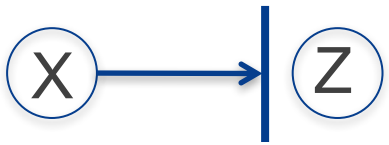


$$X \not\perp\!\!\!\perp Y \mid Z$$

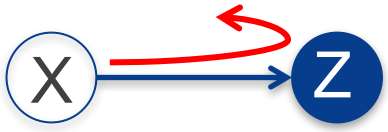
- Conditioning on a common child at the bottom of the v-structure makes its parent dependent
  - Knowing Z, X is useful in guessing Y (or vice versa)
- E.g. toss two coins representing {0,1}
  - A priori the coins are independent
  - If we observe their sum they become coupled

Conditional Independence and D-separation

The Bayes Ball Algorithm: Boundary conditions

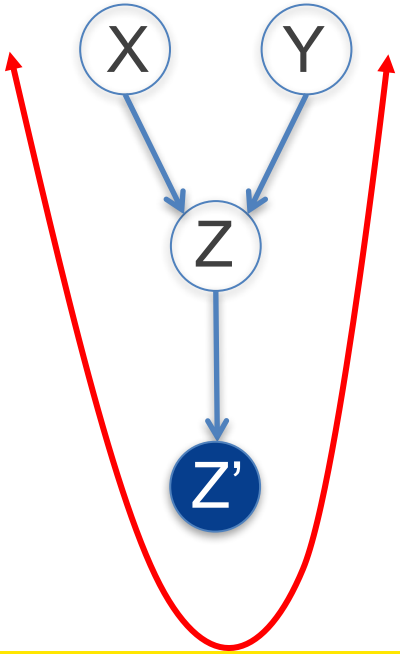


The ball does not bounce



The ball bounces back

Why do we need boundary conditions?

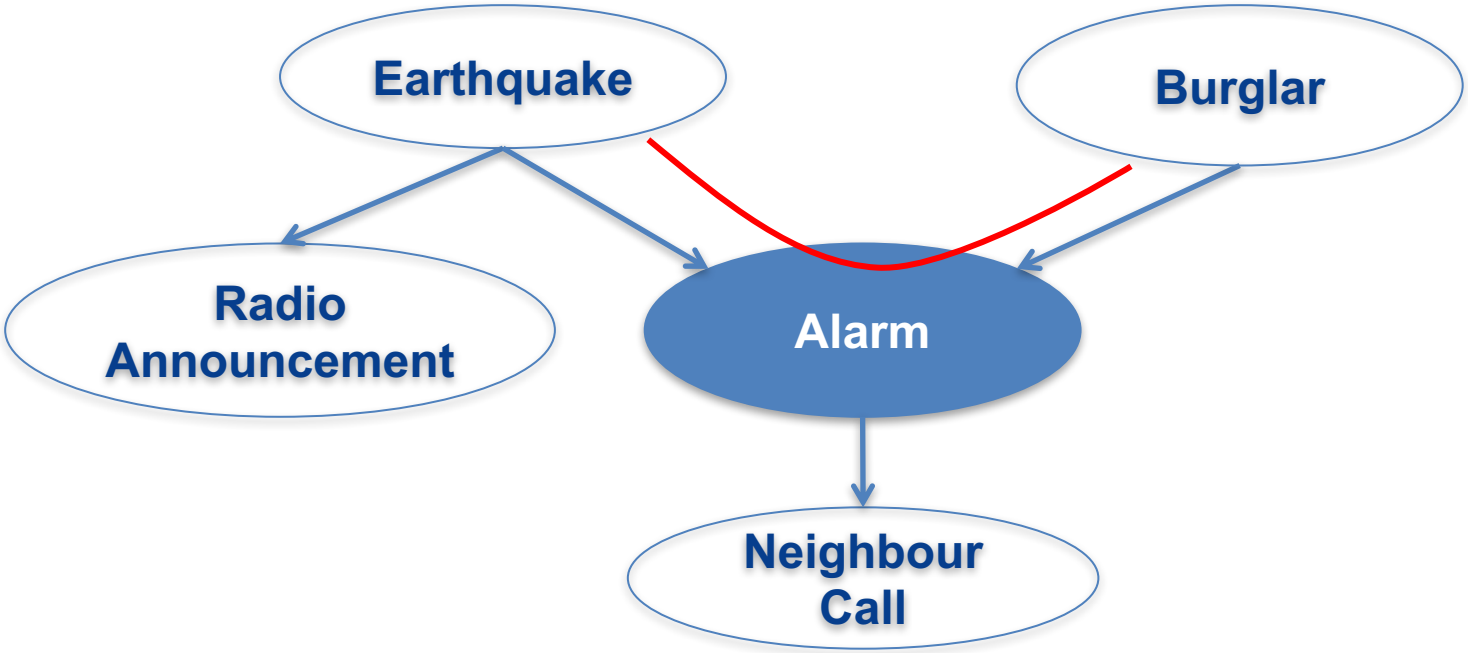


- Suppose Z' is a copy of Z
- We effectively observe Z
  - Observed collider structure
- The ball needs to bounce back along

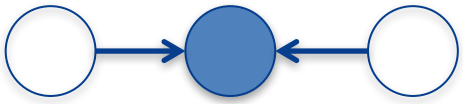
$$X \not\perp\!\!\!\perp Y \mid Z'$$

Conditional Independence and D-separation

Example: The Burglar Alarm Network (1)

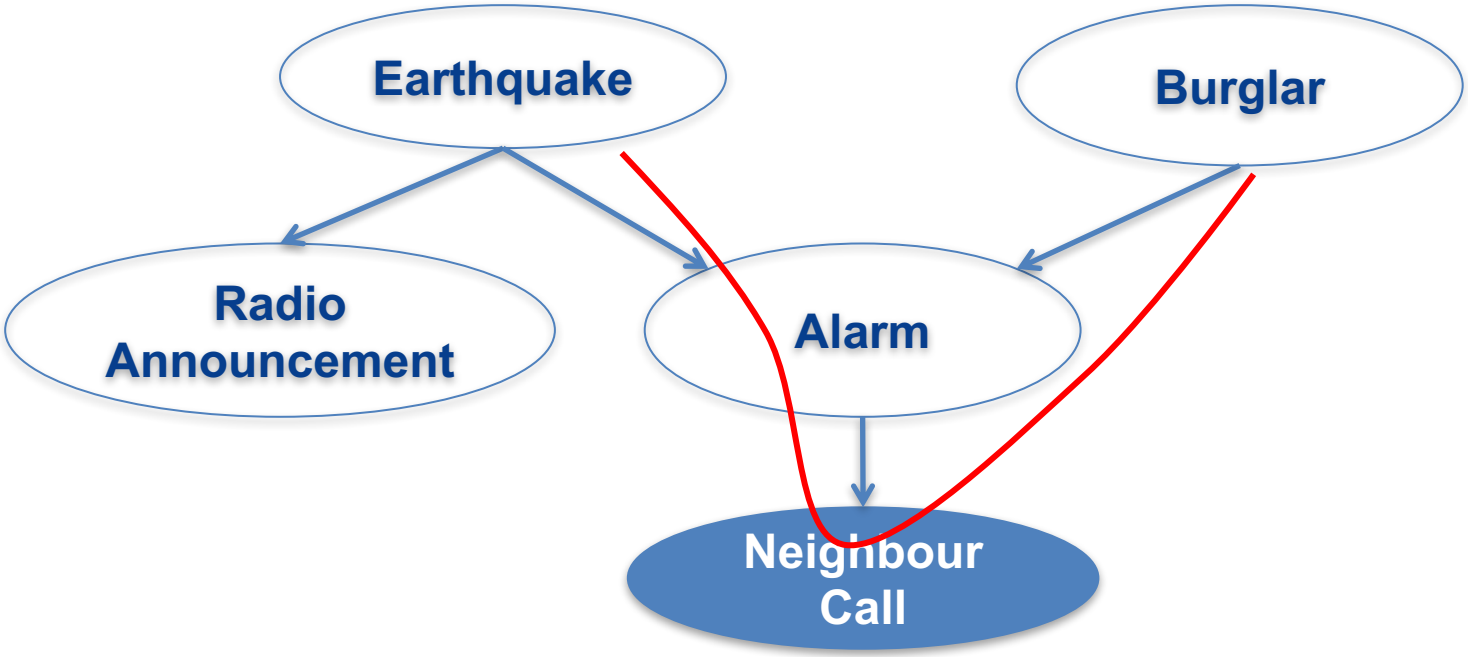


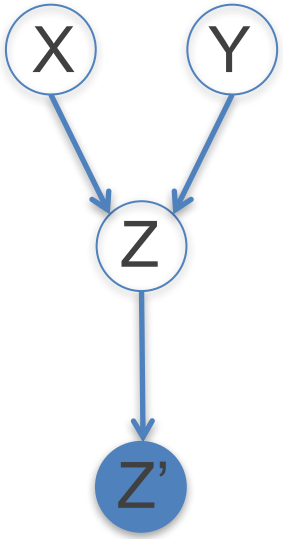
$$\text{Earthquake} \overset{?}{\perp\!\!\!\perp} \text{Burglar} \mid \text{Alarm}$$ NO



Conditional Independence and D-separation

Example: The Burglar Alarm Network (2)

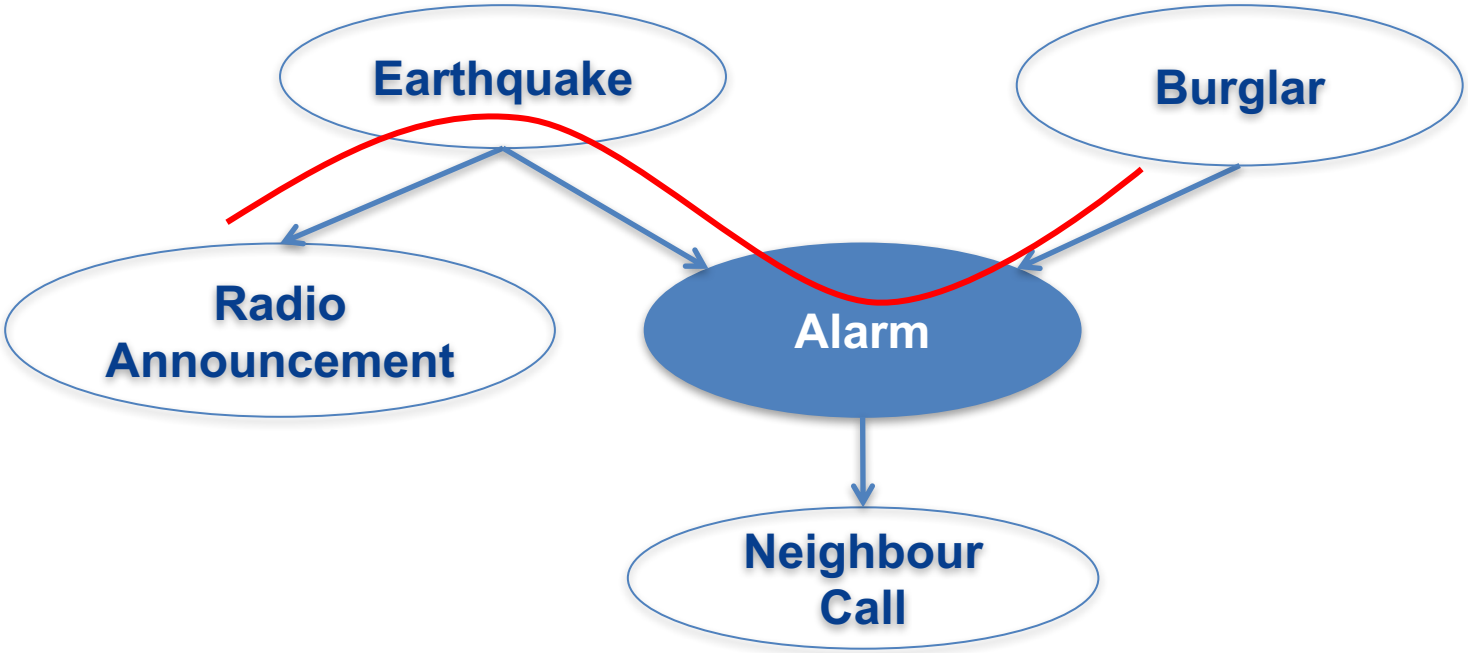




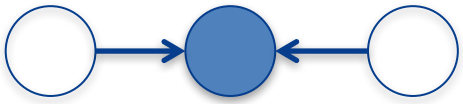
$$\text{Earthquake} \overset{?}{\perp\!\!\!\perp} \text{Burglar} \mid \text{Radio}, \text{Neighbour}$$ NO

Conditional Independence and D-separation

Example: The Burglar Alarm Network (3)

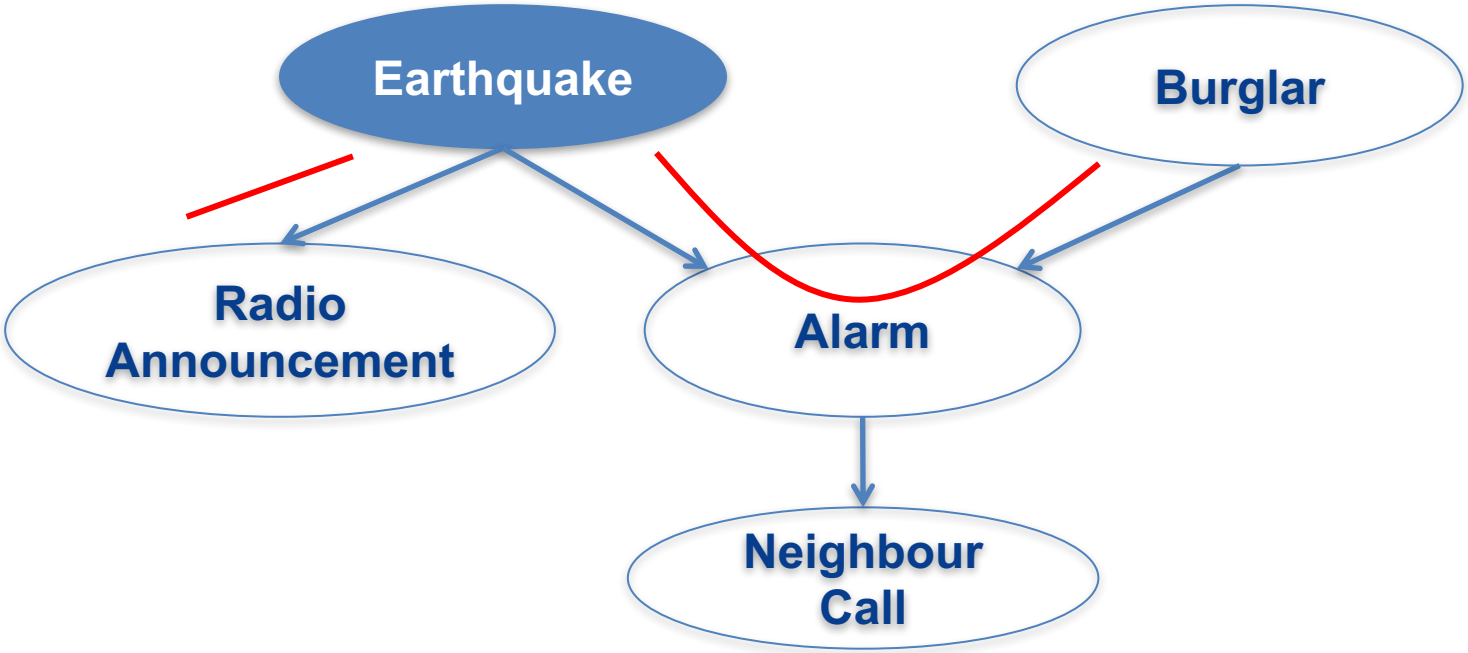


$$\text{Radio} \overset{?}{\perp\!\!\!\perp} \text{Burglar} \mid \text{Alarm}$$ NO

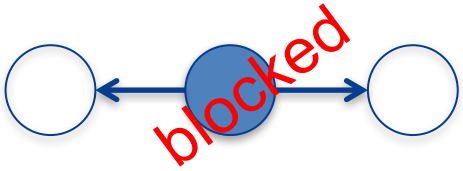


Conditional Independence and D-separation

Example: The Burglar Alarm Network (4)



$$\text{Radio} \overset{?}{\perp\!\!\!\perp} \text{Burglar} \mid \text{Earthquake}$$ Yes



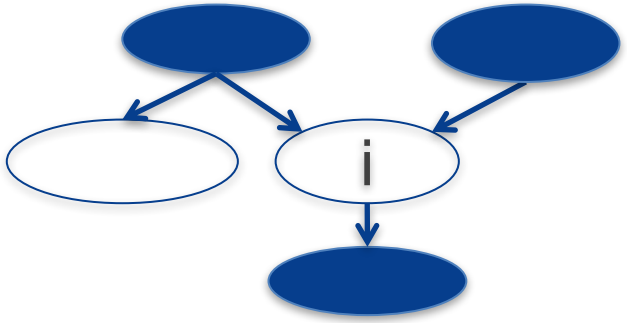
Markov Blanket

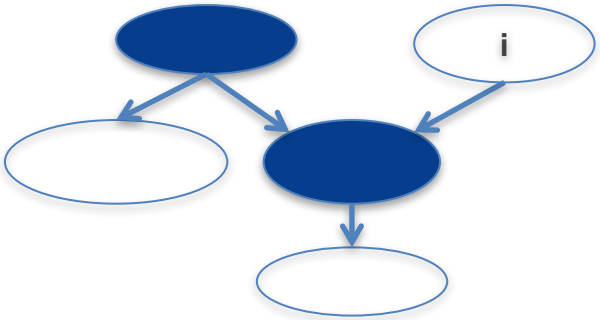
Markov Blanket of a Node

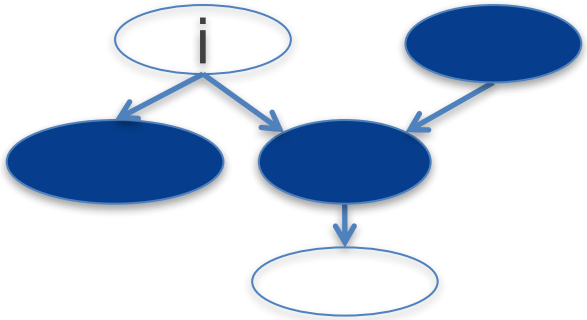
The smallest set of nodes that renders it conditionally independent of all other nodes

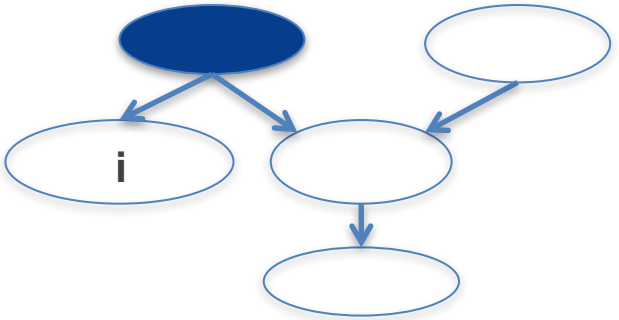
All what wee need to guess node i

- Parents of node i
- Children of node i
- Parents of children of node i







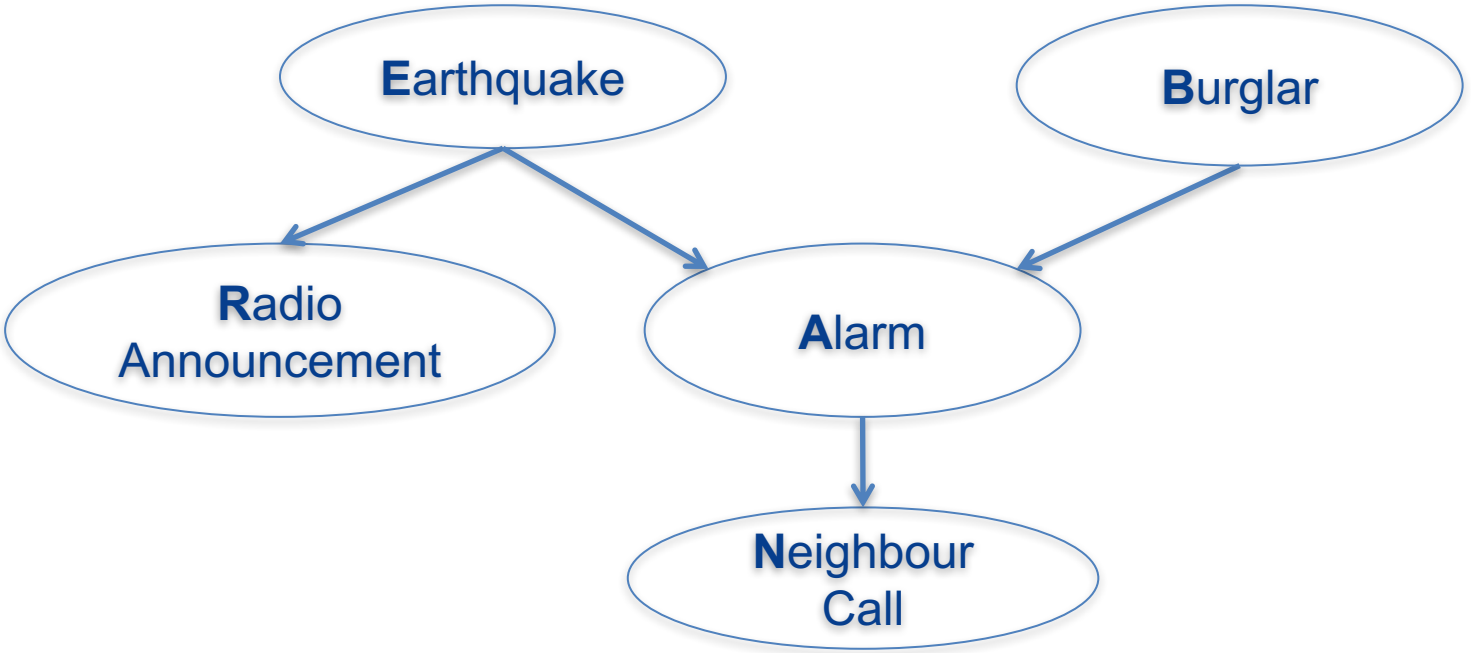


IV. Exact Inference via Variable Elimination

Inference In Graphical Models

Exact Inference by Enumeration (1)

Computing a marginal distribution

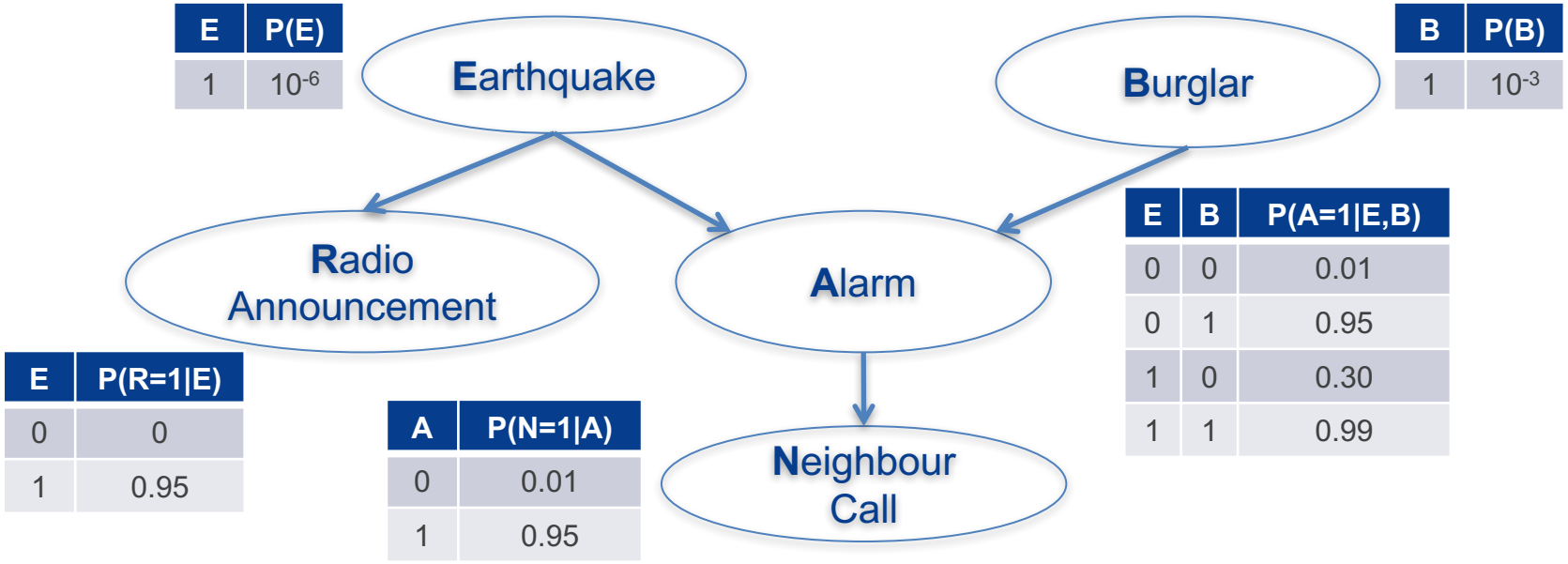


How to compute P(R)?

- Enumerate all possible states for all other variables and sum them out from the joint distribution (marginalisation)

Inference In Graphical Models

Exact Inference by Enumeration (2)



E	P(E)
1	10^{-6}

B	P(B)
1	10^{-3}

E	B	P(A=1|E,B)
0	0	0.01
0	1	0.95
1	0	0.30
1	1	0.99

E	P(R=1|E)
0	0
1	0.95

A	P(N=1|A)
0	0.01
1	0.95

Joint: $$P(E, B, R, A, N) = P(E)P(B)P(R|E)P(A|E, B)P(N|A)$$

Marginal: $$P(R) = \sum_{e,b,a,n} P(e, b, R, a, n) = \sum_{e}\sum_{b}\sum_{a}\sum_{n} P(e, b, R, a, n)$$

- Naively, $O(K^D)$ in time for D variables with K states
- We can exploit the structure of the distribution

Inference In Graphical Models

Exact Inference with Variable Elimination (Bucket Elimination)

Main ideas:

- Exploit structure of the Bayesian network
- Factorisation due to conditional independences
- Push sums inside products

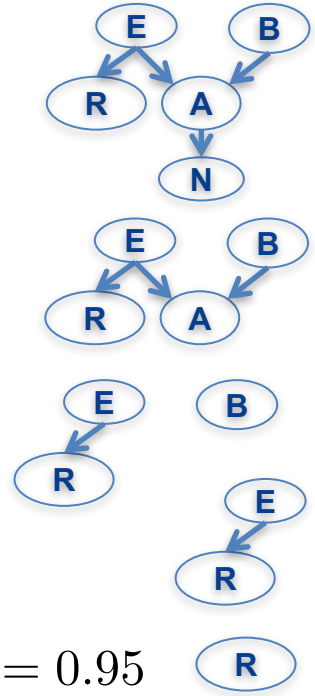
$$P(R) = \sum_e \sum_b \sum_a \sum_n P(e, b, R, a, n)$$

$$= \sum_e \sum_b \sum_a \sum_n P(e)P(b)P(R|e)P(a|e,b)P(n|a)$$

$$= \sum_e \sum_b \sum_a P(e)P(b)P(R|e)P(a|e,b) \underbrace{\sum_n P(n|a)}_{\gamma_n(a)=1}$$

$$= \sum_e \sum_b P(e)P(b)P(R|e) \sum_a P(a|e,b)$$

$$= \sum_e P(e)P(R|e) \sum_b P(b) = \sum_e P(e)P(R|e)$$



$$P(R = 1) = P(E = 0)P(R = 1|E = 0) + P(E = 1)P(R = 1|E = 1) = 0.95$$

Variable Elimination(VE)
Properties

- In general, individual local functions (γ) do not sum to one
  - These tables need to be carried forward the in computations

- Faster than naïve enumeration
  - However, can be exponentially in worst case (for general graphs)
  - Depends on the graph and elimination order
  - Elimination order is extremely important
  - Optimal ordering is NP-hard problem
    - » Clear in some simple graphs, e.g. chain, trees – $O(DK^2)$

- Introducing evidence

query evidence

$$P(B|A=1) = \frac{P(B, A=1)}{P(A=1)} \propto \sum_{e,r,n} P(e, B, r, A=1, n)$$

Other (nuisance) variables

VE is generic but inefficient

- it does not allow us to reuse computation, e.g. when computing multiple queries conditioned on the same evidence

V. The Junction Tree Algorithm

The Junction Tree Algorithm

Overview

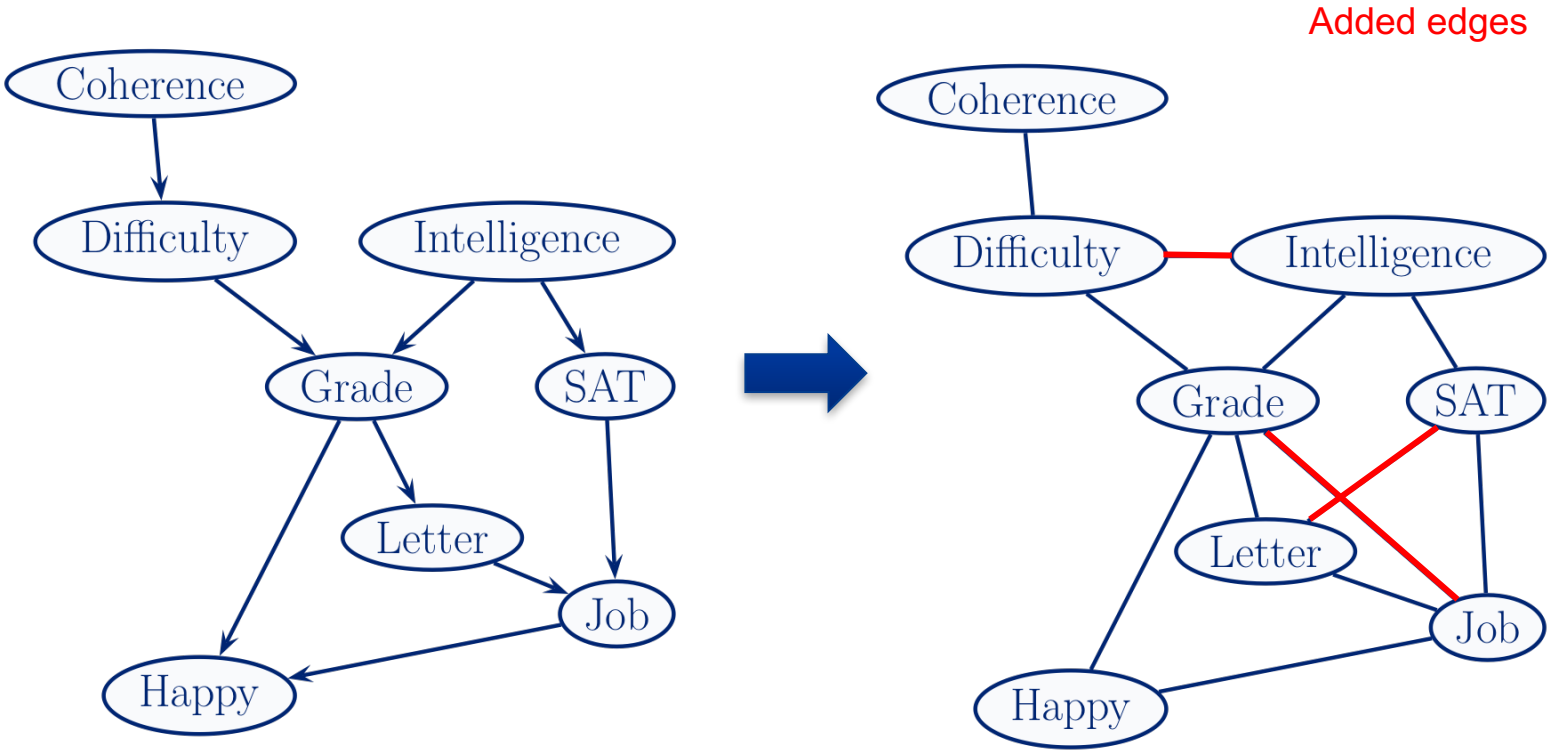
- General purpose algorithm for computing (conditional) marginals on a graph
- It generalises nearly all popular inference algorithms such as belief propagation
- Efficient in that computations can be reused

1. Moralisation
2. Triangulation
3. Construction of the junction tree
4. Assignment of potentials
5. Message passing

The Junction Tree Algorithm

1. Moralisation

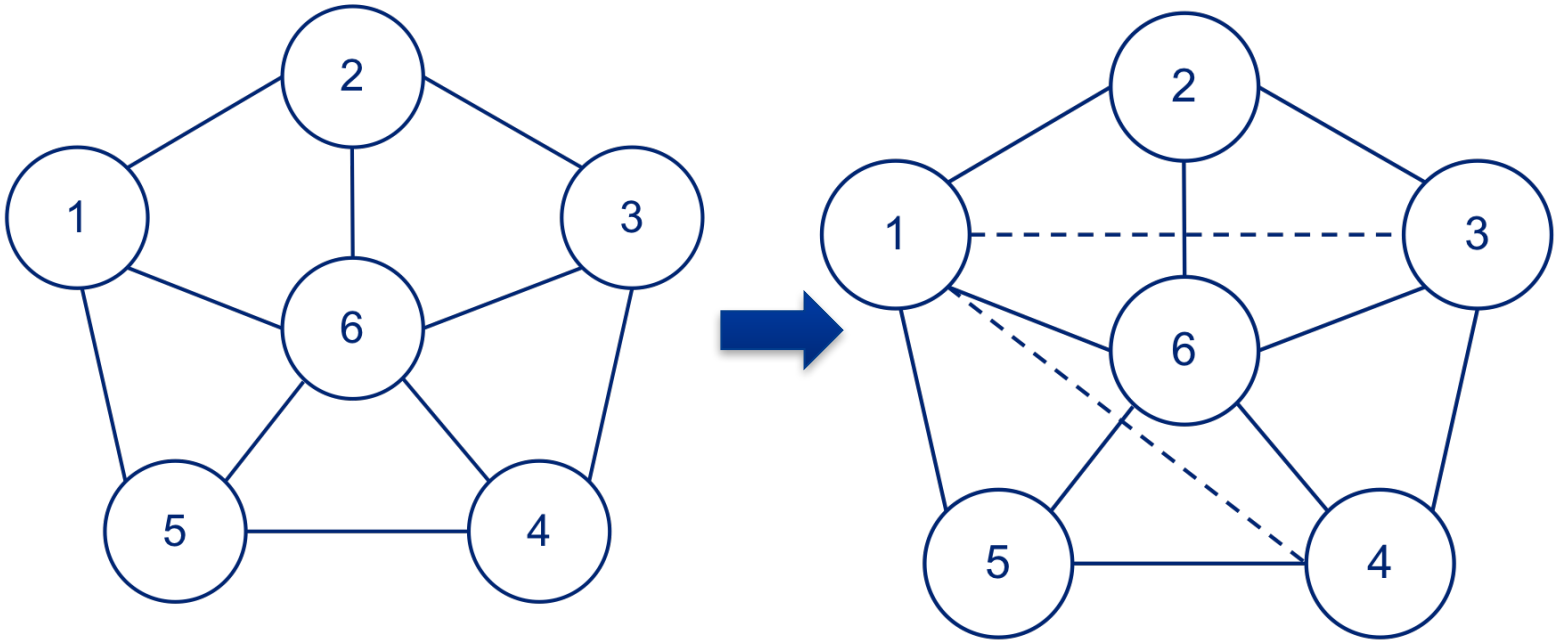
Convert directed graph into undirected graph by marrying parents and dropping link directions



The Junction Tree Algorithm

2. Triangulation

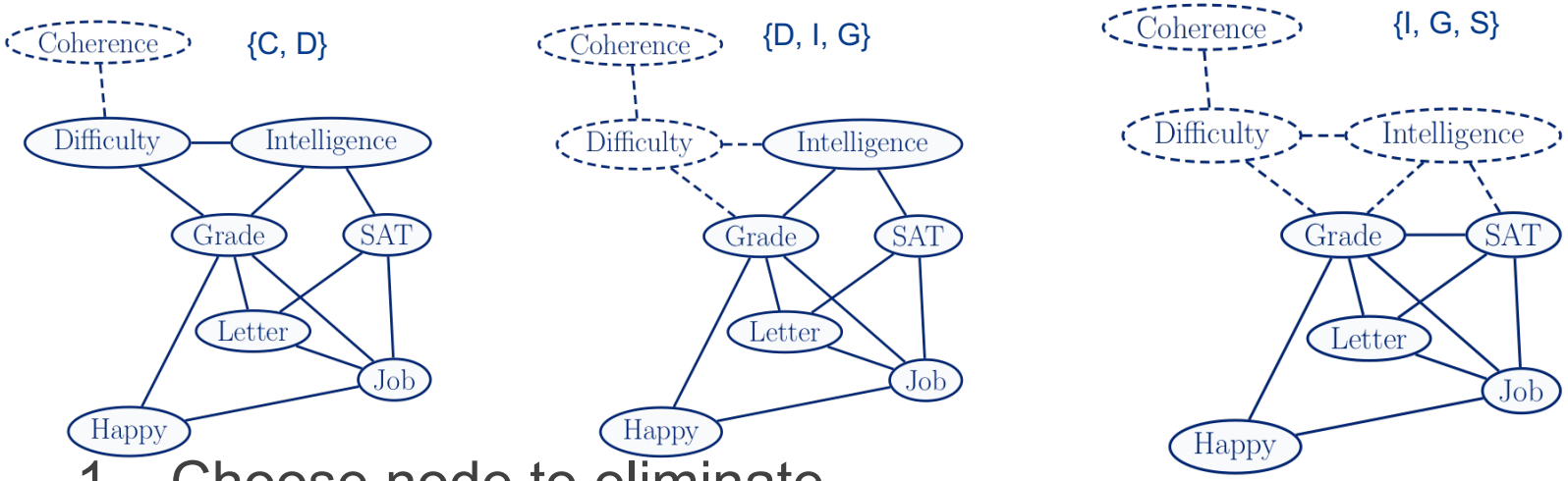
Make sure every loop of length ≥ 4 has a chord



The Junction Tree Algorithm

2. Triangulation by running Elimination Algorithm

In practice, we can triangulate the graph by running *elimination algorithm*



1. Choose node to eliminate
2. Link the remaining nodes neighbours of that node
   - Record corresponding clique
3. Remove node from the graph

Elimination order C, D, I, H, …

- Give us the cliques: {C,D}, {D,I,G}, {I,G,S}, {H, G, J}, {G, S, L, J}

The Junction Tree Algorithm

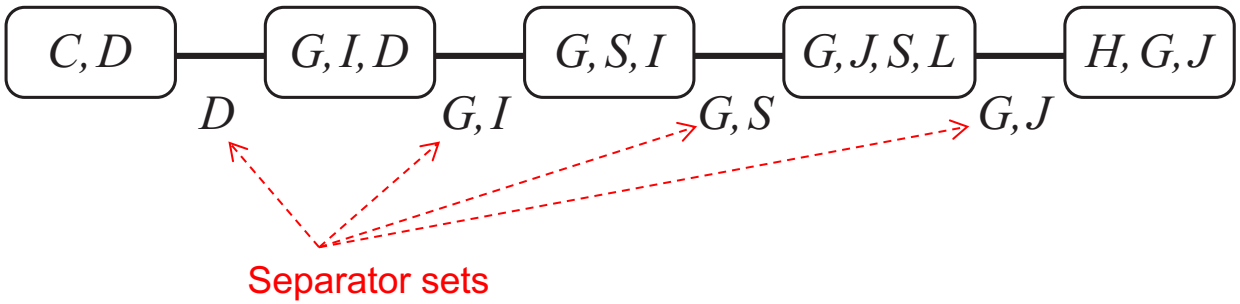
3. Construction of the Junction Tree

Select the maximal cliques and build a tree with the following:

Running Intersection Property (RIP)

Any subset of nodes containing a given variable forms a connected component.

- Maximal cliques: {C,D}, {D,I,G}, {I,G,S}, {H, G, J}, {G, S, L, J}



- The separator sets are the intersection between the neighbouring cliques

Does the graph above satisfy the RIP?

The Junction Tree Algorithm

4. Assignment of potentials

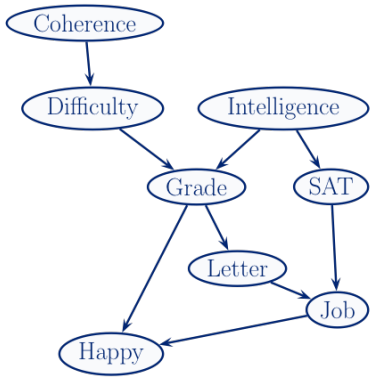
We will denote the potentials on cliques with ψ and the potential on separators with ϕ

Potential in separators are initialised to 1

$$\phi(D) = \phi(G, I) = \phi(G, S) = \phi(G, J) = 1$$

Potentials in cliques are initialised using the CPTs

$$\psi(C, D) = P(C)P(D|C)$$
$$\psi(G, I, D) = P(G|D, I)P(I)$$
$$\psi(G, S, I) = P(S|I)$$
$$\psi(G, J, S, L) = P(L|G)P(J|L, S)$$
$$\psi(H, G, J) = P(H|G, J)$$

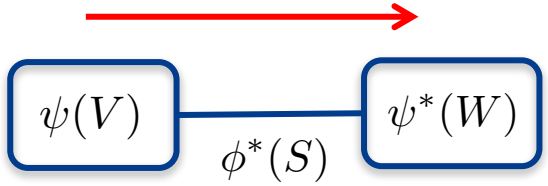


The Junction Tree Algorithm

5. Message Passing

1. Choose a root node
   - Must contain the query variable
2. Update potentials with the following rules:

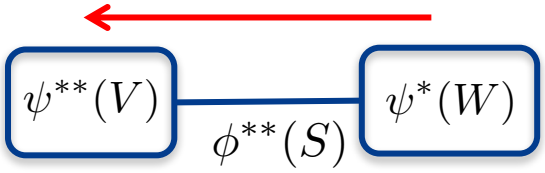
a. Collect evidence



$$\phi^*(S) = \sum_{V \setminus S} \psi(V)$$

$$\psi^*(W) = \psi(W) \frac{\phi^*(S)}{\phi(S)}$$

b. Distribute evidence



$$\phi^{**}(S) = \sum_{W \setminus S} \psi(W)$$

$$\psi^{**}(V) = \psi(V) \frac{\phi^{**}(S)}{\phi^*(S)}$$

- Node V can send exactly one message to neighbour W and it may only be sent when V has received a message from all of its other neighbours
- Collect messages to the root and distribute messages away from it

The Junction Tree Algorithm
Computing (Conditional) Marginals

After running the JTA we have that:

$$P(\mathbf{x}_H, \mathbf{x}_E) = \frac{\prod_{c \in C} \psi_c^{**}(\mathbf{x}_c)}{\prod_{s \in S} \phi_s^{**}(\mathbf{x}_s)}$$

Hidden

Evidence

Where:

Variables in c not in E

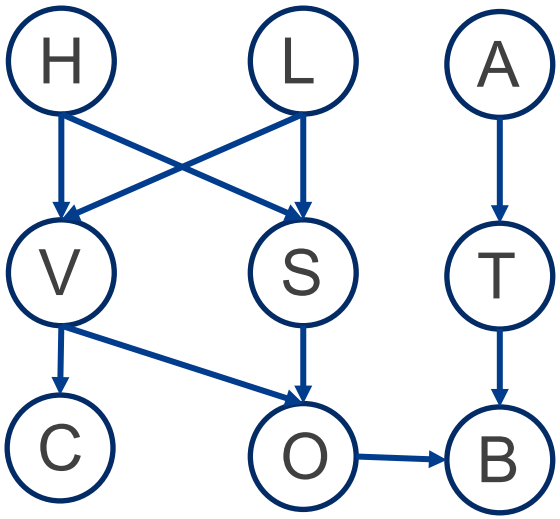
$$\psi_c^{**}(\mathbf{x}_c) = P(\mathbf{x}_{c \setminus E}, \mathbf{x}_E)$$

$$\phi_s^{**}(\mathbf{x}_s) = P(\mathbf{x}_{s \setminus E}, \mathbf{x}_E)$$

Variables in s not in E

- Can compute marginal using the updated root node potential

The JTA – Example (1)



Diagnostic variables

$$H, L, A \in \{true, false\}$$

Intermediate variables

$$V, S, T \in \{low, high\}$$

Measurement variables

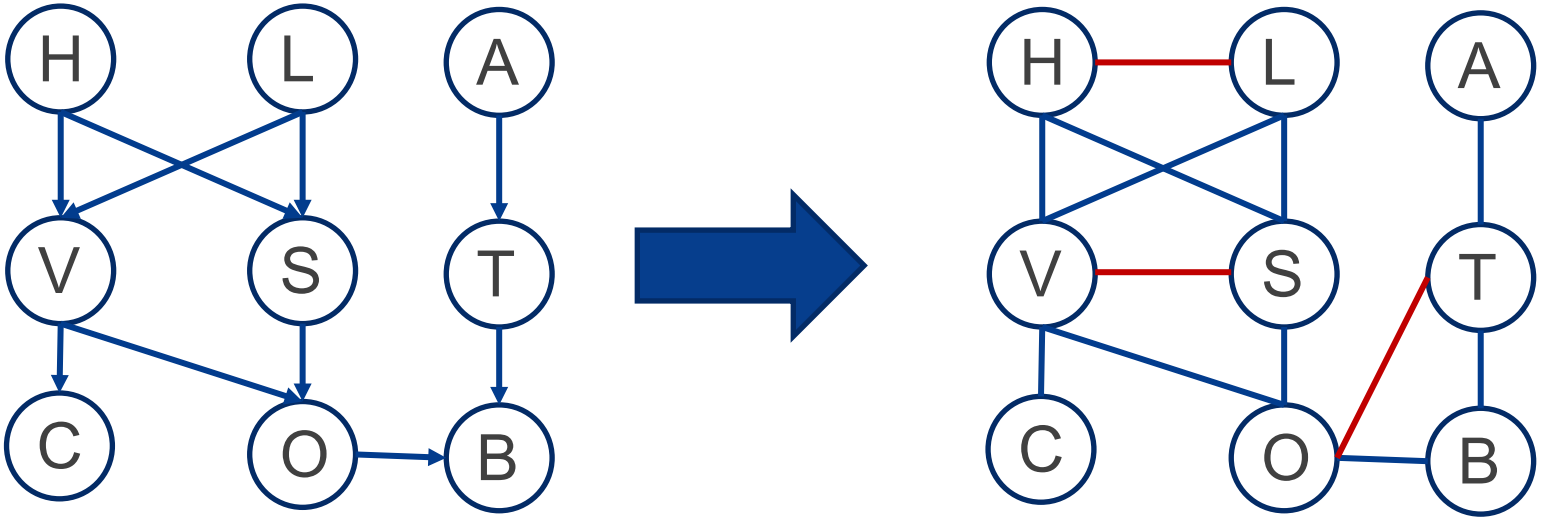
$$C, O, B \in \{low, medium, high\}$$

Building the JT

1. Moralisation
2. Triangulation
3. Construction of the junction tree
4. Assignment of potentials

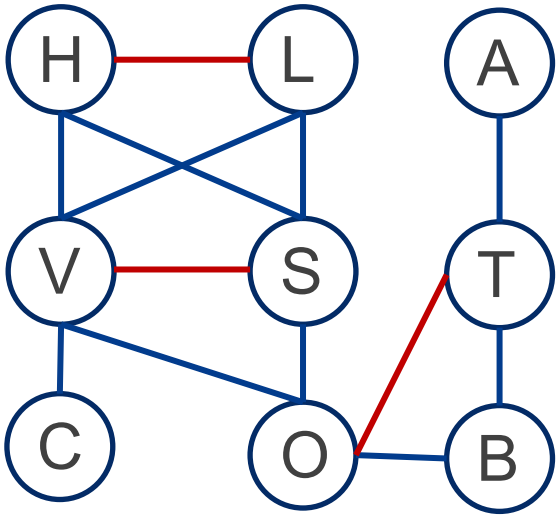
The JTA – Example (2)

Moralisation: Marry parents and drop directions of links



The JTA – Example (3)

Triangulation via elimination algorithm



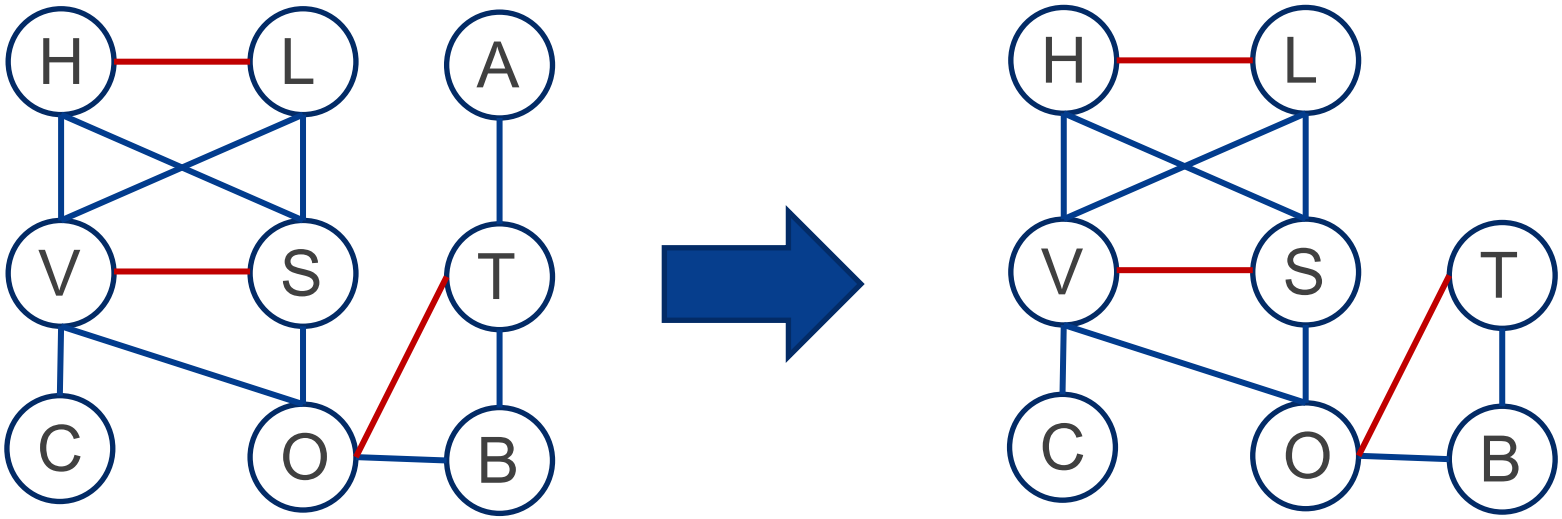
Good elimination order can be

$$A, T, B, O, C, V, H, L, S$$

a) Choose node to eliminate
b) Link the remaining nodes neighbours of that node
c) Record corresponding clique
d) Remove node from the graph

The JTA – Example (4)

Triangulation via de elimination algorithm

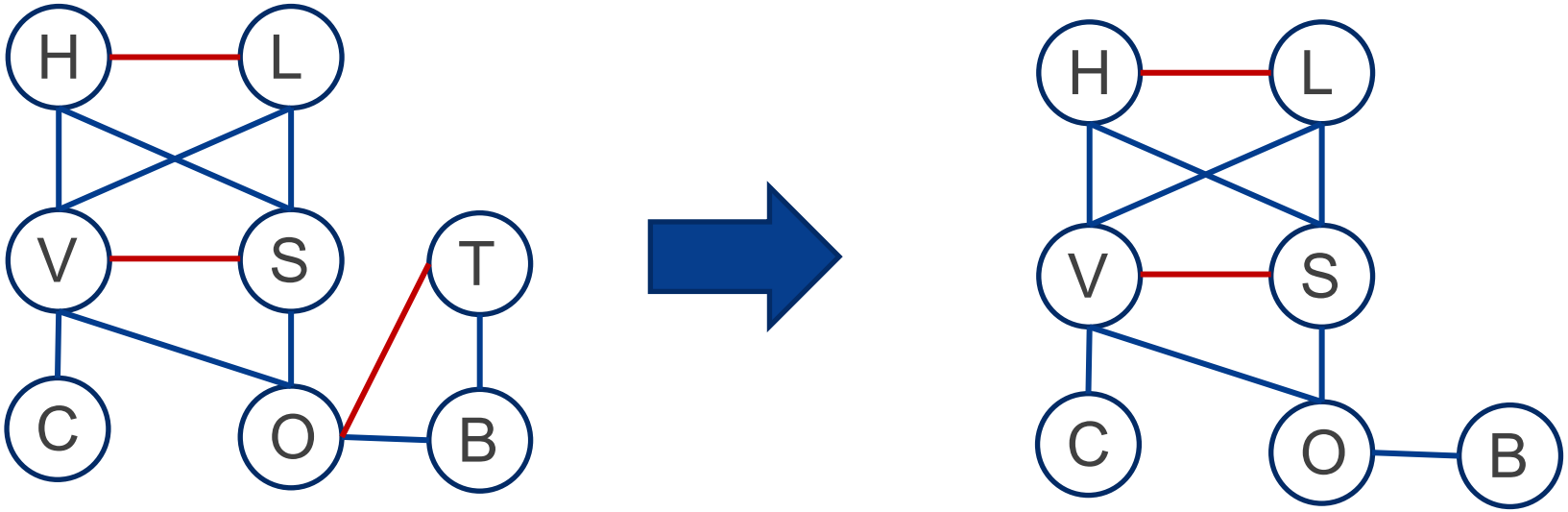


Eliminate A → record {A, T}

{A, T}

The JTA – Example (5)

Triangulation via de elimination algorithm

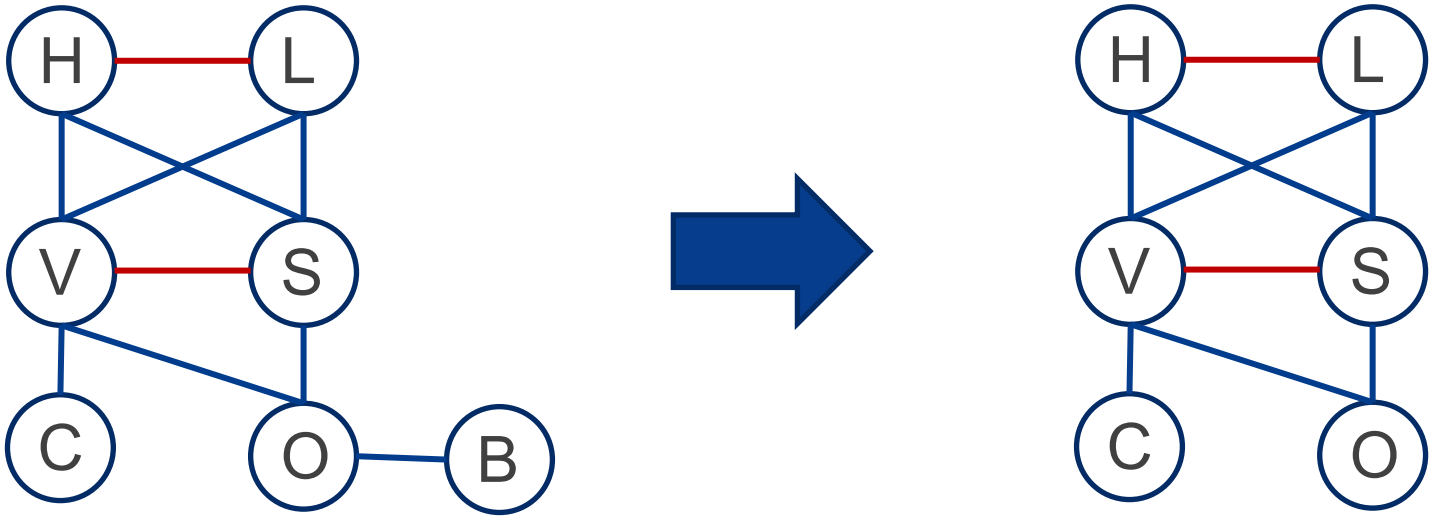


Eliminate T → record {O, T, B}

{A, T}, {O,T,B}

The JTA – Example (6)

Triangulation via de elimination algorithm

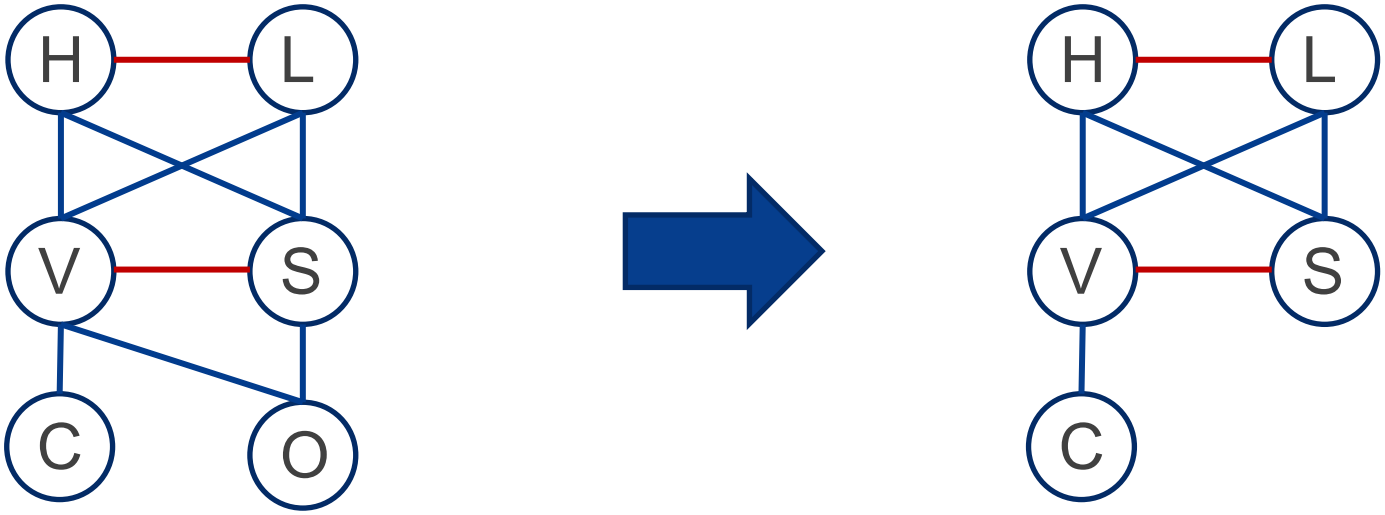


Eliminate B → {O,B} is a subset of {O, T, B} (do nothing)

{A, T}, {O,T,B}

The JTA – Example (7)

Triangulation via de elimination algorithm

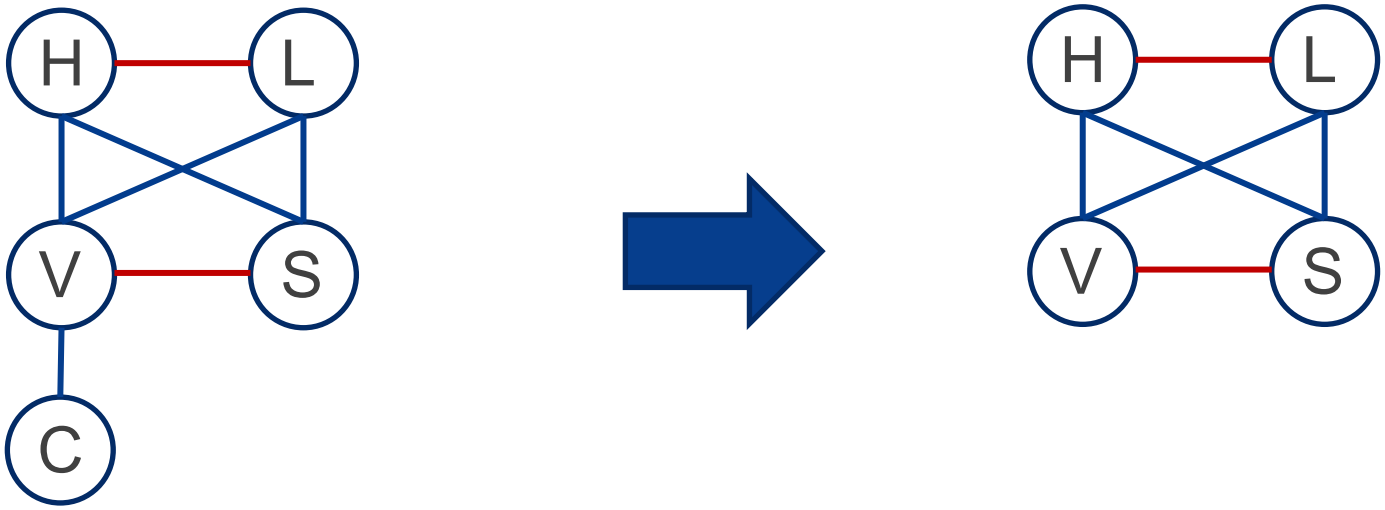


Eliminate O → record {V, S,O}

{A, T}, {O,T,B}, {V, S,O}

The JTA – Example (7)

Triangulation via de elimination algorithm

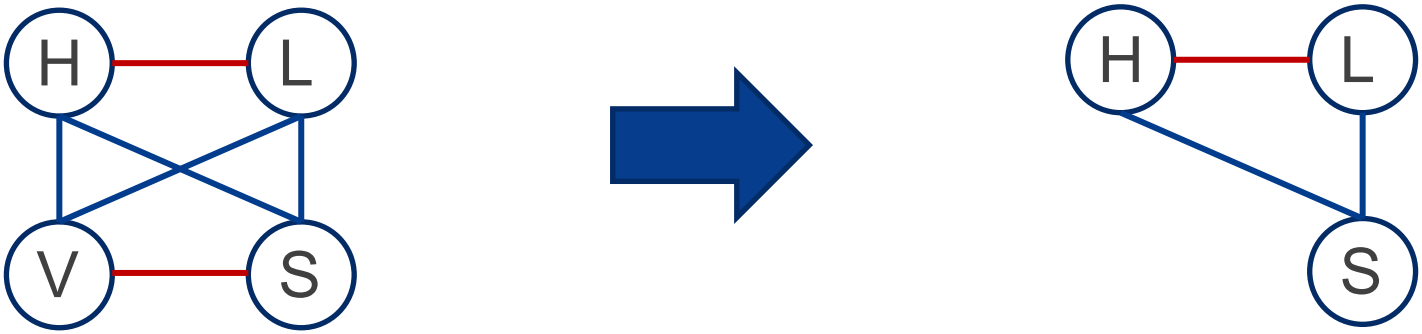


Eliminate C → record {V, C}

{A, T}, {O,T,B}, {V, S,O}, {V, C}

The JTA – Example (7)

Triangulation via de elimination algorithm

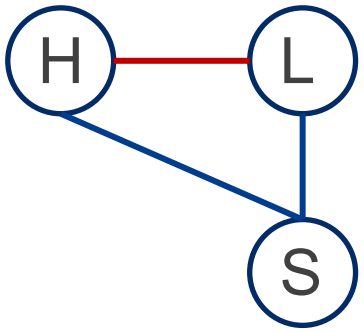


Eliminate V → record {V, H, L, S}

{A, T}, {O,T,B}, {V, S,O}, {V, C}, {V, H, L, S}

The JTA – Example (7)

Triangulation via de elimination algorithm



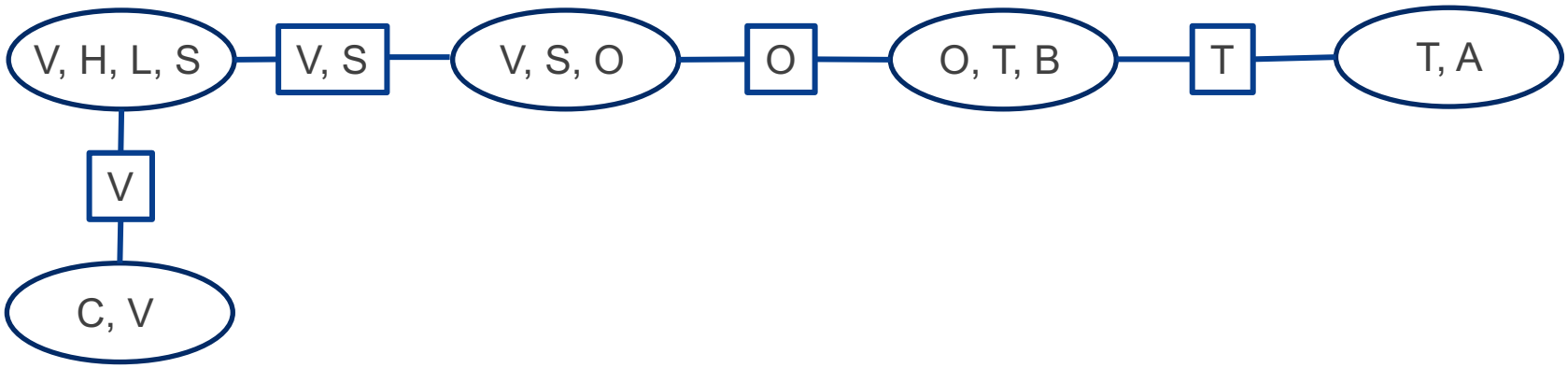
Note we did not need to link neighbours as graph was already triangulated

Eliminate H, L, S → cliques produced are subsets of {V, H, L, S}

{A, T}, {O,T,B}, {V, S,O}, {V, C}, {V, H, L, S}

The JTA – Example (8)

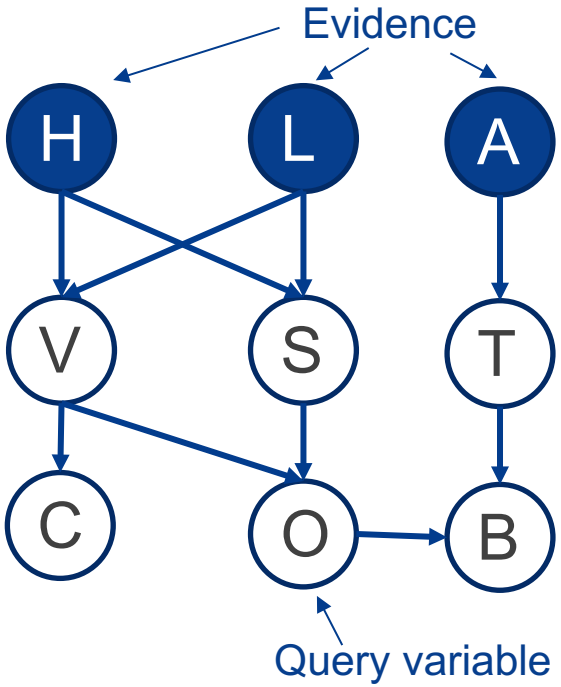
- Creation of the tree:
  - Link the cliques created before so that we construct a tree that satisfies the running intersection property (RIP)

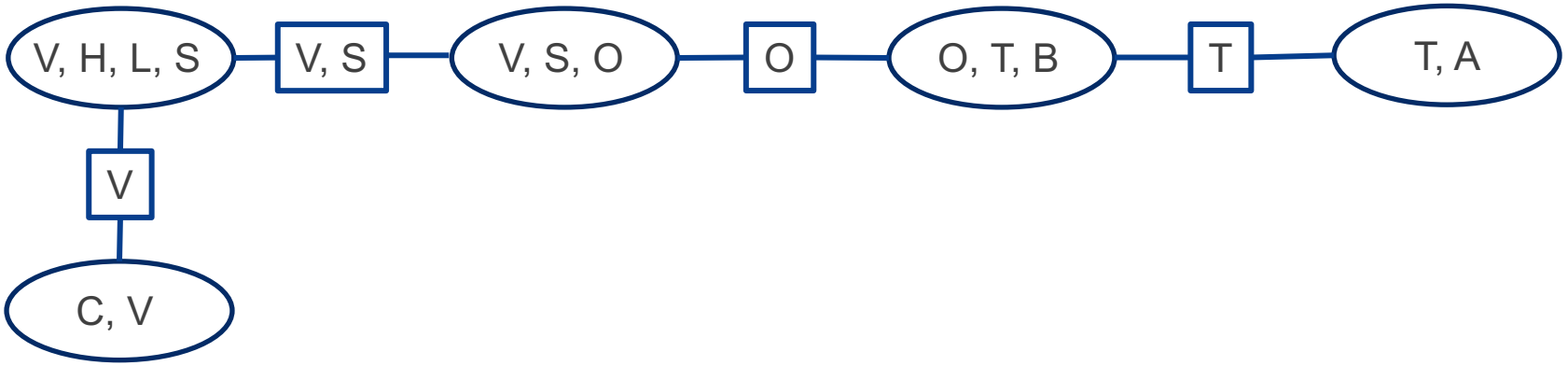


Every variable appearing in two different nodes in the tree always appears in the nodes along the path

The JTA – Example (9)

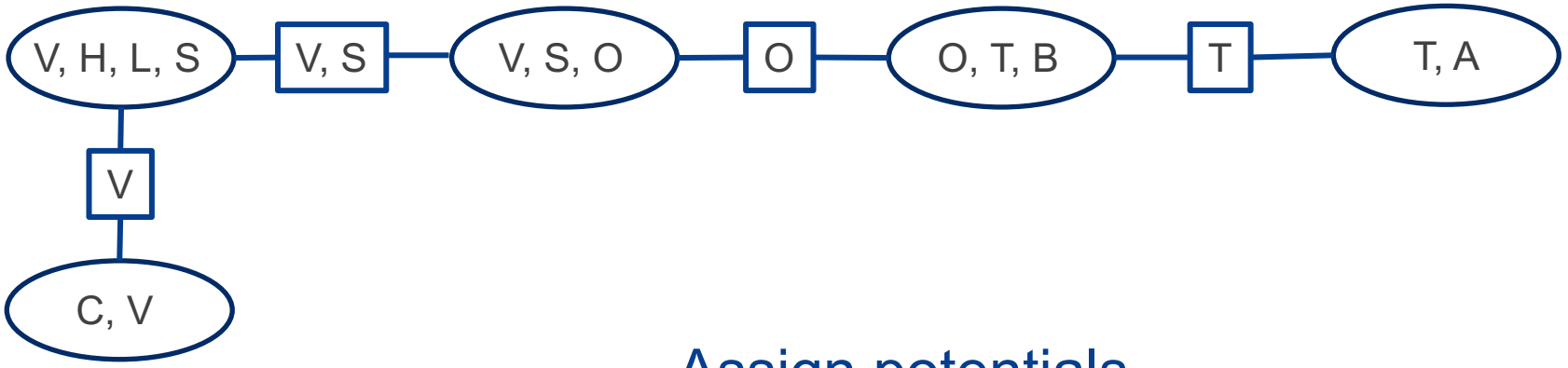
Compute *p(O | H=true, L=true, A=true)*

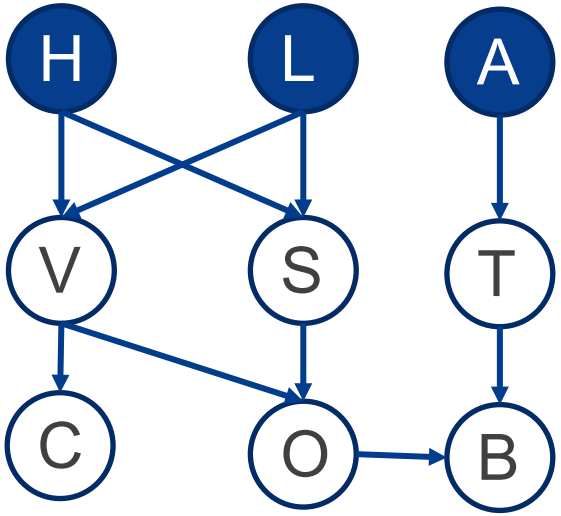




The JTA – Example (10)

Compute *p(O | H=true, L=true, A=true)*





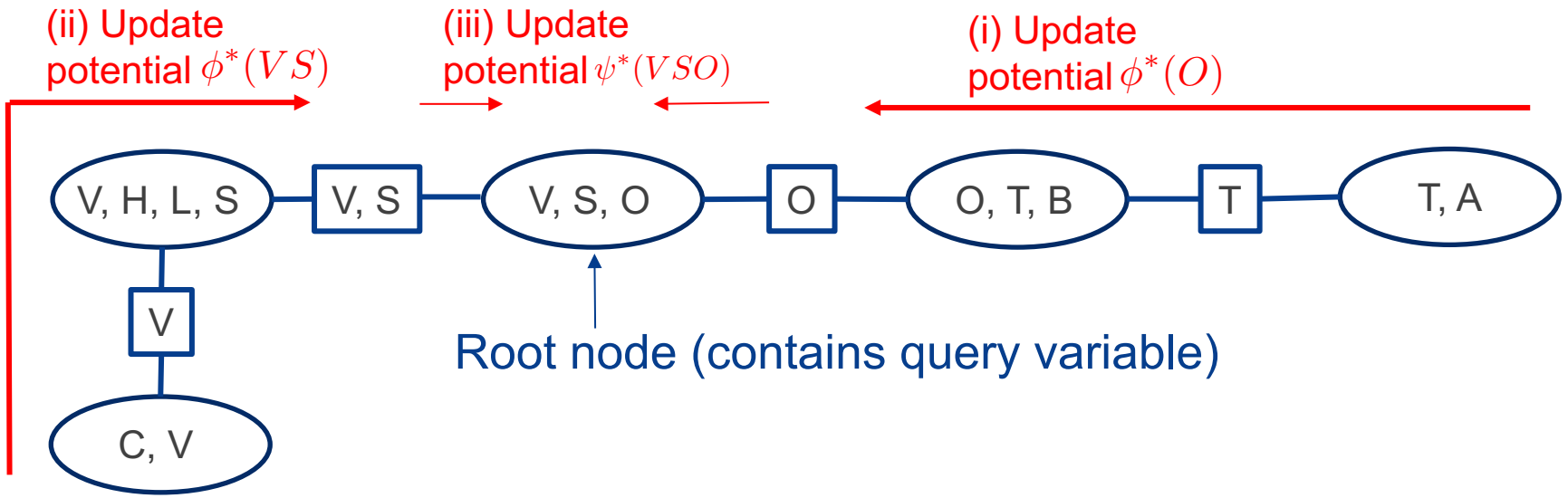
Assign potentials

$$\psi(CV) = P(C|V)$$
$$\psi(VSO) = P(O|V,S)$$
$$\psi(VHLS) = P(V|H,L)P(S|H,L)P(H)P(L)$$
$$\psi(OTB) = P(B|O,T)$$
$$\psi(TA) = P(T|A)P(A)$$

$$\phi(V) = \phi(VS) = \phi(O) = \phi(T) = 1$$

The JTA – Example (11)

Compute *p(O | H=true, L=true, A=true)*

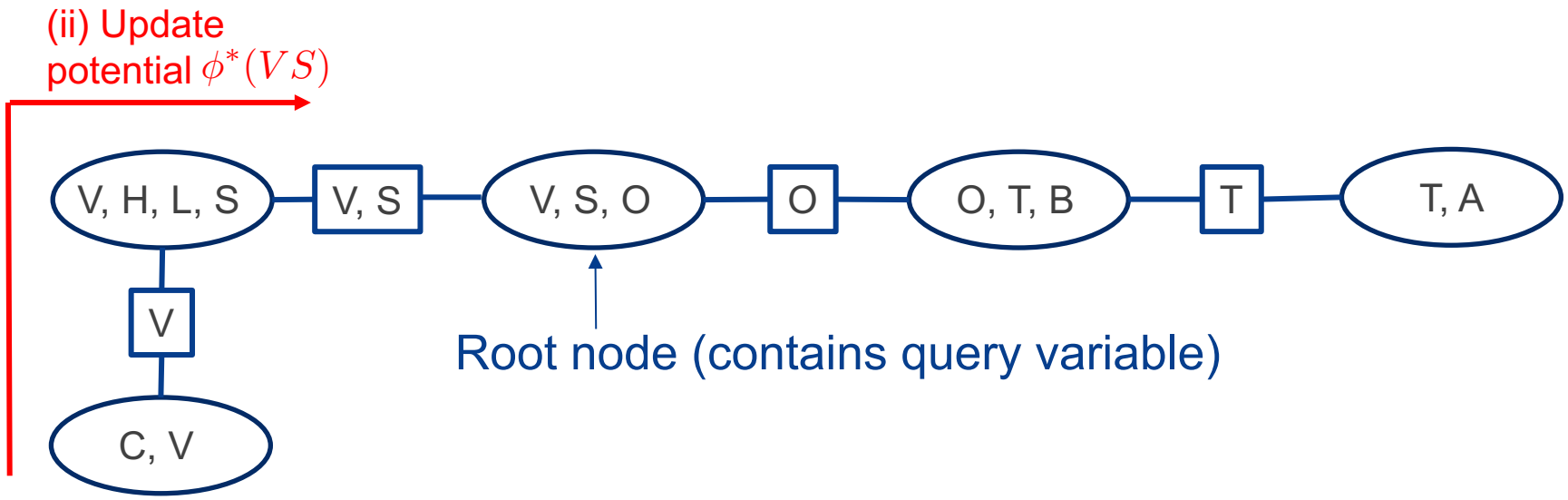


(i) Absorption from {T,A} to {O}

$$\psi^*(OTB) = \frac{\sum_A \psi(TA)\delta(A = true)}{\phi(T)}\psi(OTB)$$
$$= P(T|A = true)P(A = true)P(B|O,T)$$

$$\phi^*(O) = \sum_{BT} \psi^*(OTB)$$
$$= P(A = true)$$

The JTA – Example (12)



(ii) Absorption from {C,V} to {V,S}

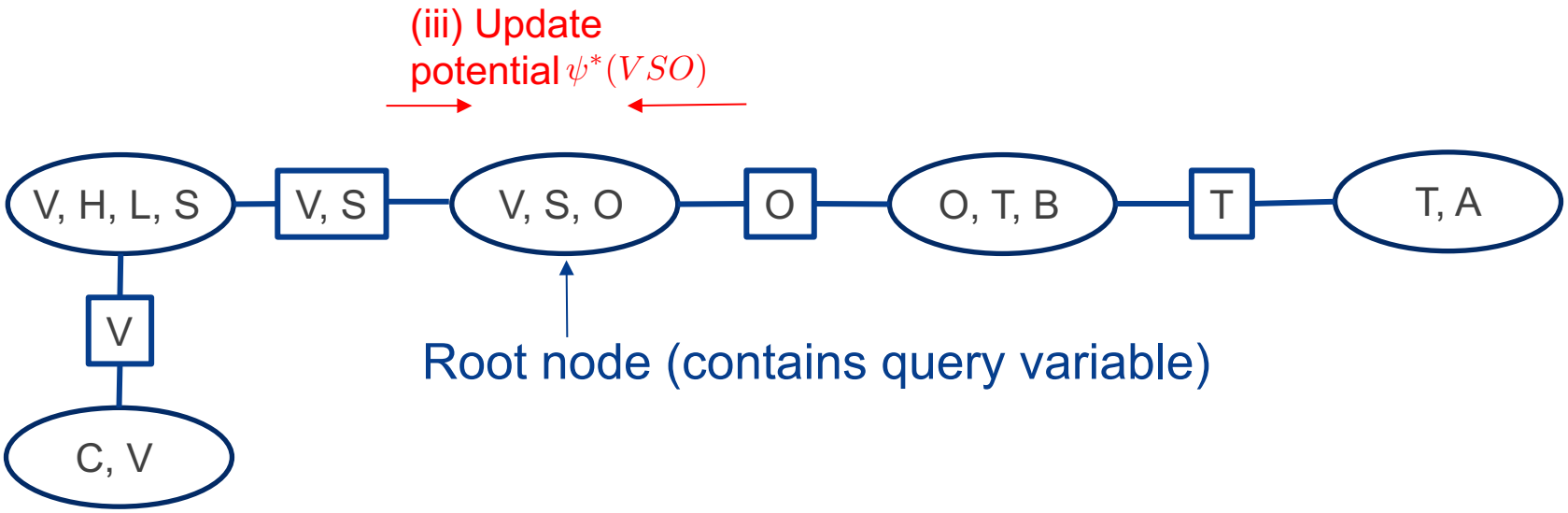
$$\psi^*(VHLS) = \frac{\sum_C \psi(CV)}{\phi(V)}\psi(VHLS) = \psi(VHLS)\underbrace{\sum_C P(C|V)}_{1}$$

$$= \psi(VHLS)$$

$$\phi^*(VS) = \sum_{HL} \psi^*(VHLS)\delta(H = true, L = true)$$

$$= P(V|H = true, L = true)P(S|H = true, L = true)P(H = true)P(L = true)$$

The JTA – Example (12)



(iii) Update of {V, S, O}

$$\psi^*(VSO) = \frac{\phi^*(O)\phi^*(VS)}{\phi(O)\phi(VS)}\psi(VSO)$$

$$= P(A = true)P(V|H = true, L = true)P(S|H = true, L = true)P(H = true)P(L = true)P(O|V, S)$$

The JTA – Example (13)

- We have collected so far all the evidence to the node {V,S,O} and we know that:

$$\psi^{**}(VSO) = \psi^{*}(VSO)$$
$$= P(V, S, O, H = true, L = true, A = true)$$

- Therefore, we can calculate the marginal required as follows:

$$P(O|H = true, L = true, A = true) = \frac{P(O, H = true, L = true, A = true)}{P(H = true, L = true, A = true)}$$
$$= \frac{\sum_{VS} \psi^{*}(VSO)}{\sum_{OVS} \psi^{*}(VSO)}$$

Inference in Graphical models

Final Remarks

- JTA is generic but computationally expensive $O(|C| K^w)$
  - $|C|$ is the number of cliques and w the tree width
  - Exponential in the size of the largest clique
  - Choosing a triangulation to minimize the tree width is NP-hard

- Is there a smarter algorithm out there?
  - Exact inference is #P-hard

- Why do we need to solve the problem exactly?

- Approximate inference – Soon ☺
  - Variational inference, expectation propagation
  - Monte Carlo (rejection sampling, importance sampling, particle filtering)
  - MCMC

Summary & Conclusions

- Conditional independence
- Graphical models
  - D-separation and conditional independences
  - Bayes' ball algorithm
- Inference
  - with variable elimination
  - JTA
- Reading
  - Murphy (MLaPP, 2012): Ch. 10 & Ch. 20 (except Sec. 20.2)