Markov networks

Hwanjo Yu

POSTECH

http://di.postech.ac.kr/hwanjoyu

Search problem

Markov decision processes

Reinforcement learning & games

Constraint satisfaction problems

Markov networks

Bayesian networks

Reflex **States** **Variables** **Logic**

"Low-level intelligence" "High-level intelligence"

Machine learning

Roadmap

Markov networks

Conditional independence

Review: probability

Random variables: sunshine $S \in \{0,1\}$, rain $R \in \{0,1\}$

Joint distribution (unknown world)

$\mathbb{P}(S,R) =$

s	r	$\mathbb{P}(S=s,R=r)$
0	0	0.20
0	1	0.08
1	0	0.70
1	1	0.02

Marginal distribution

(aggregate rows)

$\mathbb{P}(S) =$

s	$\mathbb{P}(S=s)$
0	0.28
1	0.72

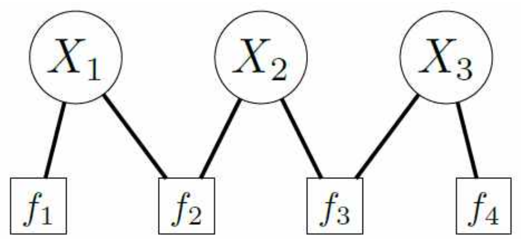
Conditional distribution

(select rows, normalize)

$\mathbb{P}(S \mid R=1) =$

s	$\mathbb{P}(S=s \mid R=1)$
0	0.8
1	0.2

Review: factor graph and CSPs



Definition: factor graph

- Variables:

$X = (X_1, \dots, X_n)$, where $X_i \in \text{Domain}_i$

- Factors:

$f_1, \dots, f_m$, with each $f_j(X) \geq 0$

Definition: assignment weight

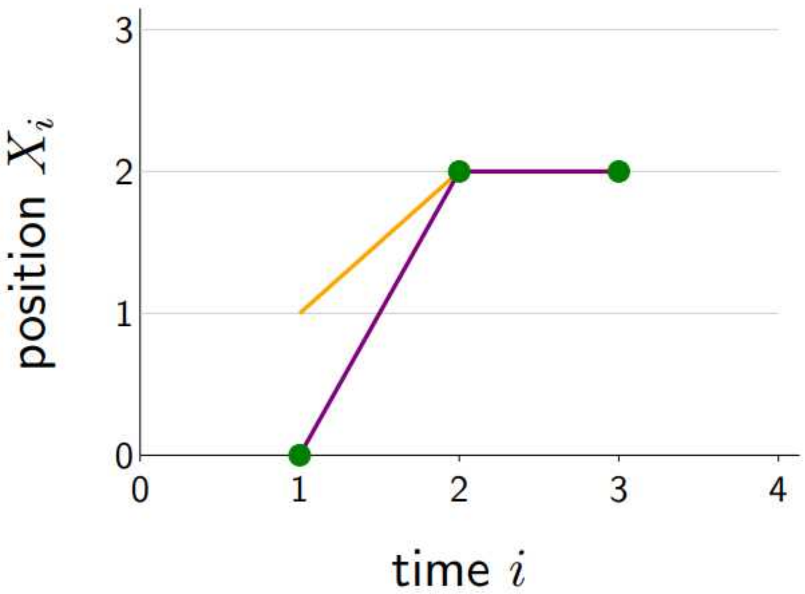
- Each **assignment** $x = (x_1, \dots, x_n)$ has a **weight**:

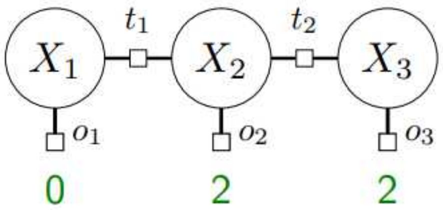
$$\text{Weight}(x) = \prod_{j=1}^{m} f_j(x)$$

Objective: find the maximum weight assignment

$$arg \ \max_x \text{Weight}(x)$$

Example: object tracking





x_1	$o_1(x_1)$
0	2
1	1
2	0

x_2	$o_2(x_2)$
0	0
1	1
2	2

x_3	$o_3(x_3)$
0	0
1	1
2	2

$\|x_i - x_{i+1}\|$	$t_i(x_i, x_{i+1})$
0	2
1	1
2	0

Maximum weight assignment

- CSP objective: find the maximum weight assignment

$$\max_{\mathbf{x}} \mathrm{Weight}(x)$$

x_1	x_2	x_3	Weight(x)
0	1	1	4
0	1	2	4
1	1	1	4
1	1	2	4
1	2	1	2
1	2	2	8

- Maximum weight assignment: $\{x_1: 1, x_2: 2, x_3: 2\}$ (weight 8)
- But this doesn't represent how likely it is.

Markov network

Definition: Markov network

- A Markov network is a factor graph which defines a joint distribution over random variables $X = (X_1, \ldots, X_n)$:

$$\mathbb{P}(X = x) = \frac{\text{Weight}(x)}{Z}$$

x_1	x_2	x_3	Weight(x)	$\mathbb{P}(X = x)$
0	1	1	4	0.15
0	1	2	4	0.15
1	1	1	4	0.15
1	1	2	4	0.15
1	2	1	2	0.08
1	2	2	8	0.31

$$Z = 4 + 4 + 4 + 4 + 2 + 8 = 26$$

Marginal probabilities

Object tracking example:

x_1	x_2	x_3	Weight(x)	$\mathbb{P}(X=x)$
0	1	1	4	0.15
0	1	2	4	0.15
1	1	1	4	0.15
1	1	2	4	0.15
1	2	1	2	0.08
1	2	2	8	0.31

- $\mathbb{P}(X_2 = 1) = 0.15 + 0.15 + 0.15 + 0.15 = 0.62$
- $\mathbb{P}(X_2 = 2) = 0.08 + 0.31 = 0.38$

- Where was the object at time step 2 (X_2)? Different than max weight assignment!

Definition: Marginal probability

- The marginal probability of $X_i = v$ is given by:

$$\mathbb{P}(X_i = v) = \sum_{x:x_i=v} \mathbb{P}(X = x)$$

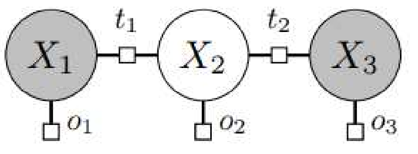
Gibbs sampling

Algorithm: Gibbs sampling

- Initialize x to a random complete assignment
- Loop through $i = 1, \dots, n$ until convergence (converge depending on the initial assignment):
  - For each variable X_i:
    - Set $x_i = v$ with prob. $\mathbb{P}(X_i = v | X_{-i} = x_{-i})$ (notation: $X_{-i} = X \backslash \{X_i\}$)
    - Increment $\text{count}_i(x_i)$
- Estimate $\widehat{\mathbb{P}}(X_i = x_i) = \frac{\text{count}_i(x_i)}{\sum_v \text{count}_i(v)}$

Example: sampling one variable

- $\text{Weight}(x \cup \{X_2: 0\}) = 1$ prob. 0.2
- $\text{Weight}(x \cup \{X_2: 1\}) = 2$ prob. 0.4
- $\text{Weight}(x \cup \{X_2: 2\}) = 2$ prob. 0.4



Application: image denoising

Example: image denoising

- $X_i \in \{0,1\}$ is pixel value in location i
- Subset of pixels are observed

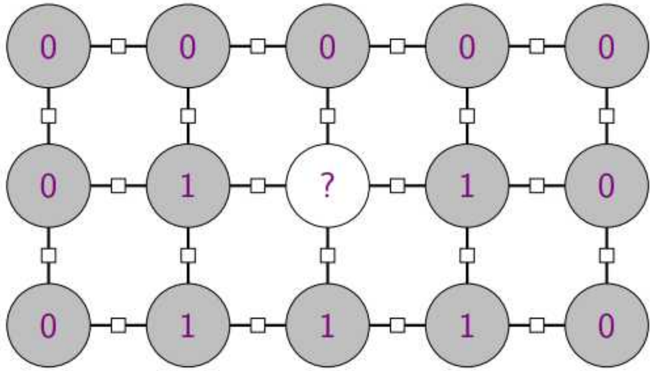
$$o_i(x_i) = [x_i = \text{observed value at } i]$$

- Neighboring pixels more likely to be same than different

$$t_{ij}\left(x_i, x_j\right) = \left[x_i = x_j\right] + 1$$

- Scan through image and update each pixel given rest:

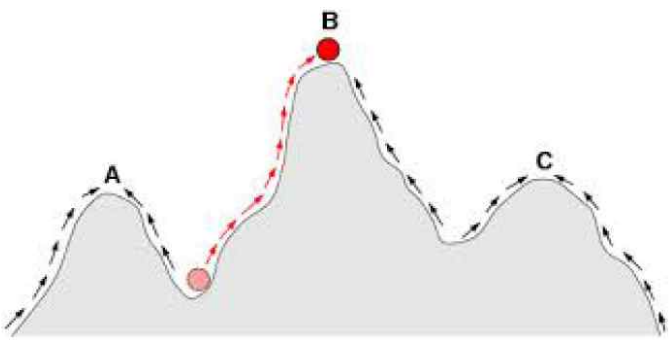
v	weight	$\mathbb{P}(X_i = v \mid X_{-i} = x_{-i})$
0	2*1*1*1	0.2
1	1*2*2*2	0.8



Search versus sampling

Iterated Conditional Modes	Gibbs sampling
maximum weight assignment	marginal probabilities
choose best value	sample a value
converge to local optimum	marginals converge to correct answer*

*under technical conditions (sufficient condition: all weights positive), but could take exponential time



Summary

Markov networks = factor graphs + probability

- Normalize weights to get probability distribution
- Can compute marginal probabilities to focus on variables

CSPs	Markov networks
Variables	Random variables
Weights	Probabilities
Maximum weight assignment	Marginal probabilities $\mathbb{P}(X_i = x_i)$
ICM	Gibbs sampling

Roadmap

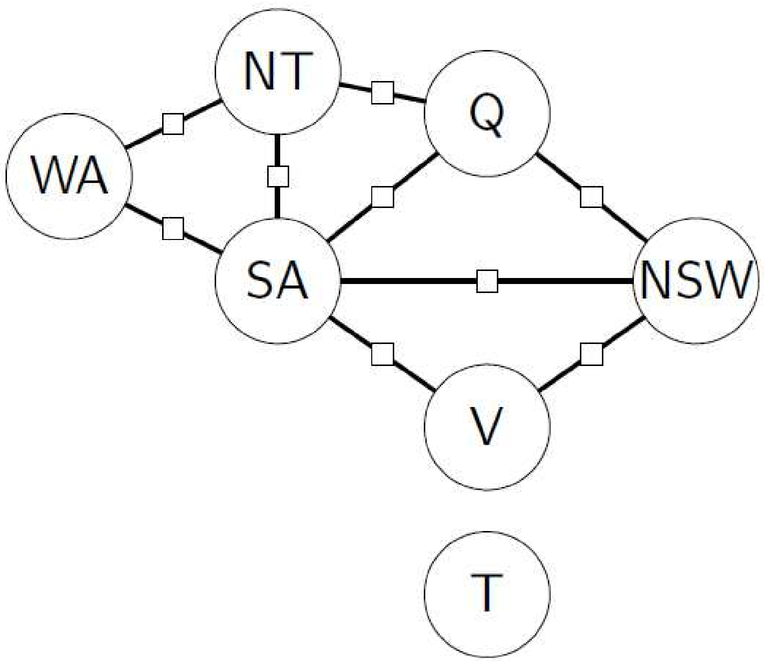
Markov networks

Conditional independence

Motivation

Key idea:

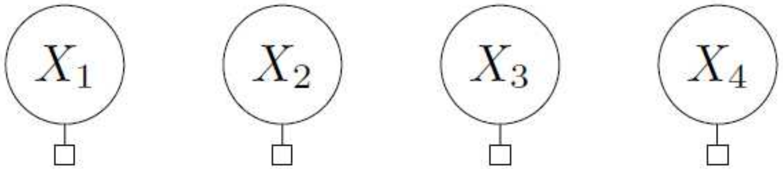
- Leverage graph properties to derive efficient algorithms which are exact.



Motivation

Backtracking search:

exponential time in number of variables n



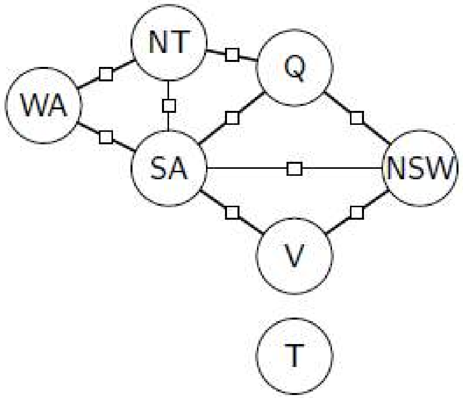
Efficient algorithm:

maximize each variable separately

Independence

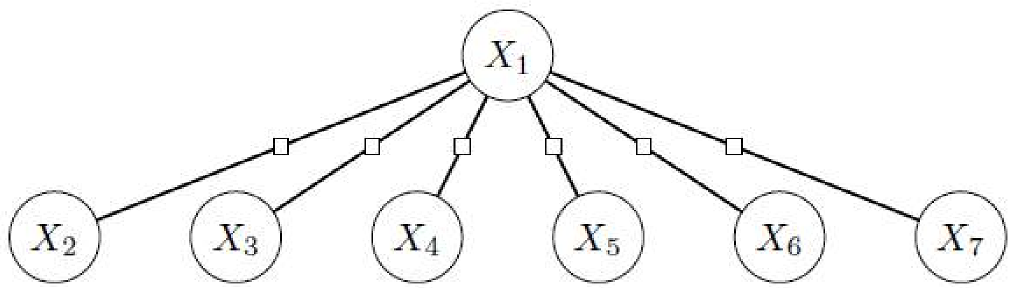
Definition: independence

- Let A and B be a partitioning of variables X.
- We say A and B are **independent** if there are no edges between A and B.
- In symbols: $A \perp B$.



$\{\mathrm{WA}, \mathrm{NT}, \mathrm{SA}, \mathrm{Q}, \mathrm{NSW}, \mathrm{V}\}$ and $\{\mathrm{T}\}$ are independent.

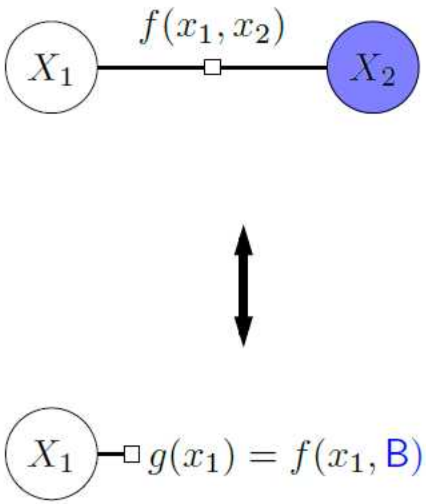
Non-independence



No variables are independent of each other, but feels close...

Conditioning

Goal: try to disconnect the graph



x_1	x_2	$f(x_1, x_2)$
R	R	1
R	B	7
B	R	3
B	B	2

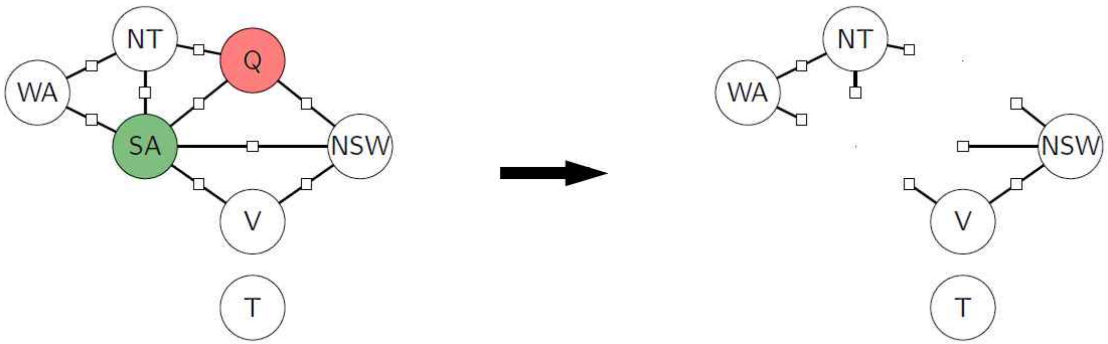
x_1	$g(x_1)$
R	7
B	2

Condition on $X_2 = \text{B}$: remove X_2, f and add g

Conditioning: example

Example: map coloring

- Condition on $\mathrm{Q} = \mathrm{R}$ and $\mathrm{SA} = \mathrm{G}$.



- New factors:

$$[\mathrm{NT} \neq \mathrm{R}] \qquad [\mathrm{WA} \neq \mathrm{G}]$$

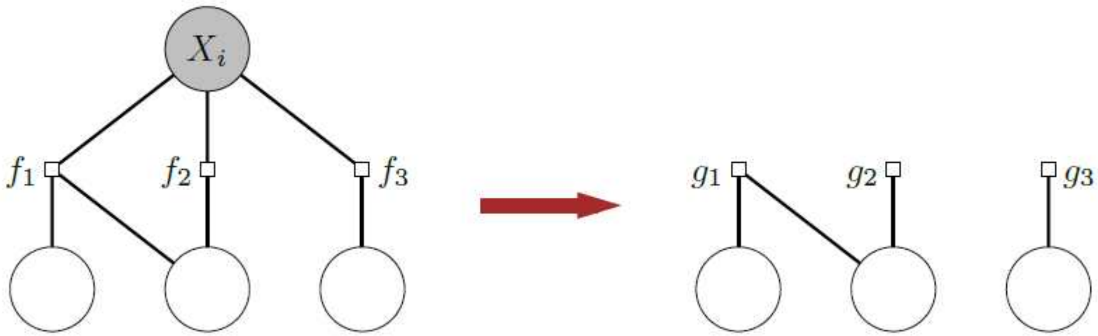
$$[\mathrm{NSW} \neq \mathrm{R}] \qquad [\mathrm{NT} \neq \mathrm{G}]$$

$$[\mathrm{NSW} \neq \mathrm{G}]$$

$$[\mathrm{V} \neq \mathrm{G}]$$

Conditioning: general

Graphically: remove edges from X_i to dependent factors



Definition: conditioning

- To condition on a variable $X_i = v$, consider all factors $f_1, \ldots, f_k$ that depend on X_i.
- Remove X_i and $f_1, \ldots, f_k$.
- Add $g_j(x) = f_j(x \cup \{X_i : v\})$ for $j = 1, \ldots, k$.

Conditional independence

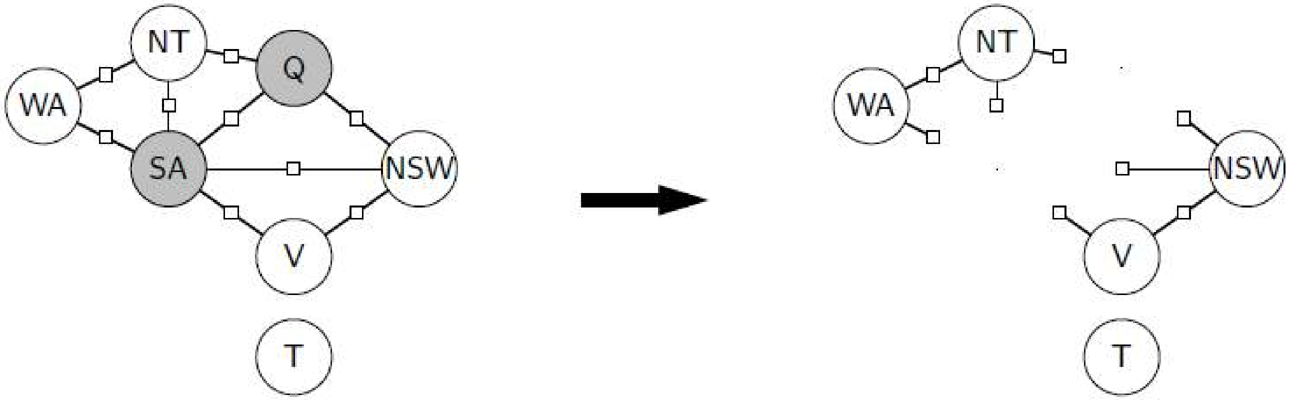
Definition: conditional independence

- Let A, B, C be a partitioning of the variables.
- We say A and B are **conditionally independent** given C if conditioning on C produces a graph in which A and B are independent.
- In symbols: $A \perp B \mid C$.

Equivalently: every path from A to B goes through C.

Conditional independence

Example: map coloring

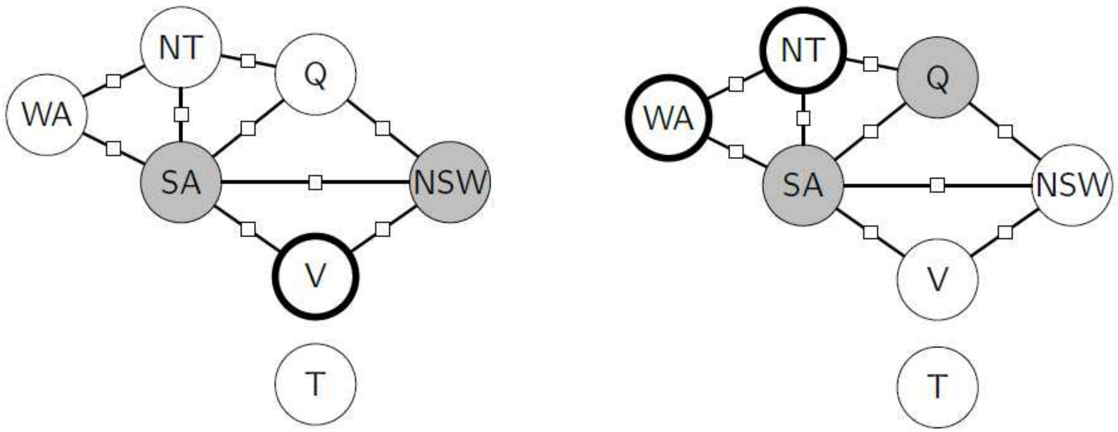


Conditional independence assertion:

$$\{\mathrm{WA}, \mathrm{NT}\} \perp \{\mathrm{V}, \mathrm{NSW}, \mathrm{T}\} \mid \{\mathrm{SA}, \mathrm{Q}\}$$

Markov blanket

How can we separate an arbitrary set of nodes from everything else?

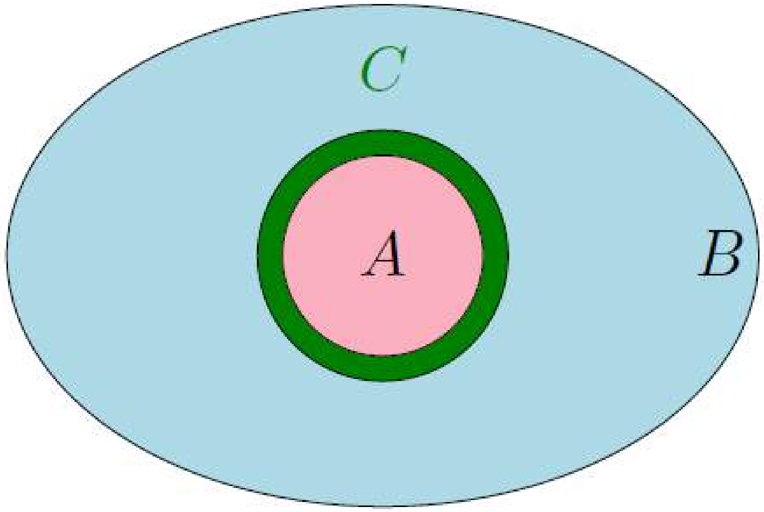


Definition: Markov blanket

- Let $A \subseteq X$ be a subset of variables.
- Define $\mathrm{MarkovBlanket}(A)$ be all the neighbors of A that are not in A.

*the smaller the Markov blanket, the easier the factor graph is to deal with

Markov blanket



Proposition: conditional independence

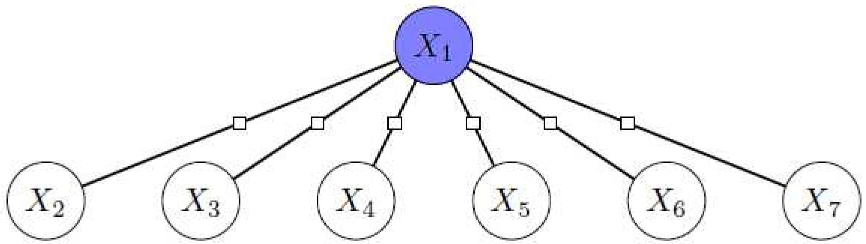
- Let $C = \mathrm{MarkovBlanket}(A)$.
- Let B be $X \setminus (A \cup C)$.
- Then $A \perp B \mid C$.
- C is the smallest set of variables to condition on to make A independent of the rest.

Using conditional independence

For each value $v = \mathrm{R}, \mathrm{G}, \mathrm{B}$:

Condition on $X_1 = v$.

Find the maximum weight assignment (easy).



R	3
G	6
B	1

maximum weight is 6

Summary

- Independence: when sets of variables A and B are disconnected; can solve separately.
- Conditioning: assign variable to value, replaces binary factors with unary factors
- Conditional independence: when C blocks paths between A and B
- Markov blanket: what to condition on to make A conditionally independent of the rest.