hidden markov models

19 oct 2022

CSCI 373

let's travel back to the first days of the semester when we were reconnecting with all our old friends...

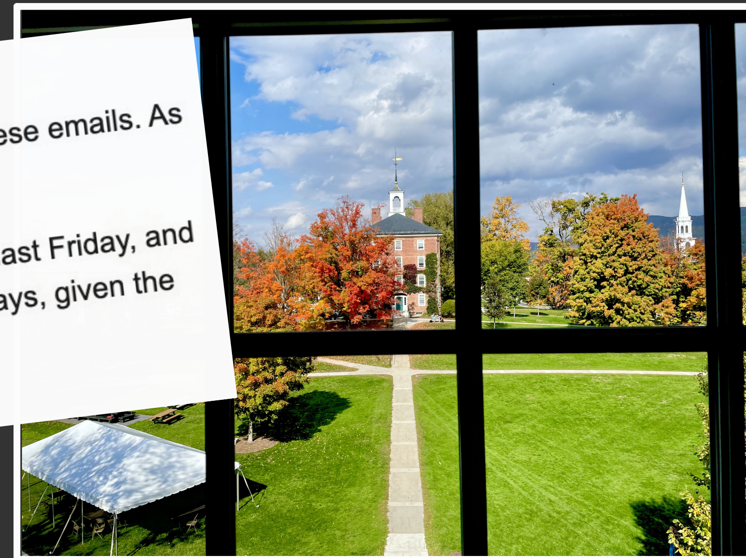
let's travel back to the

Case counts

Starting now, we'll share case counts with you whenever I send these emails. As of today we have:

- **110 current student positives**. This is down from 165 last Friday, and the total is expected to decline further in the next few days, given the number of people almost done with isolation.
- **31 current faculty/staff positives**.

all our old friends...

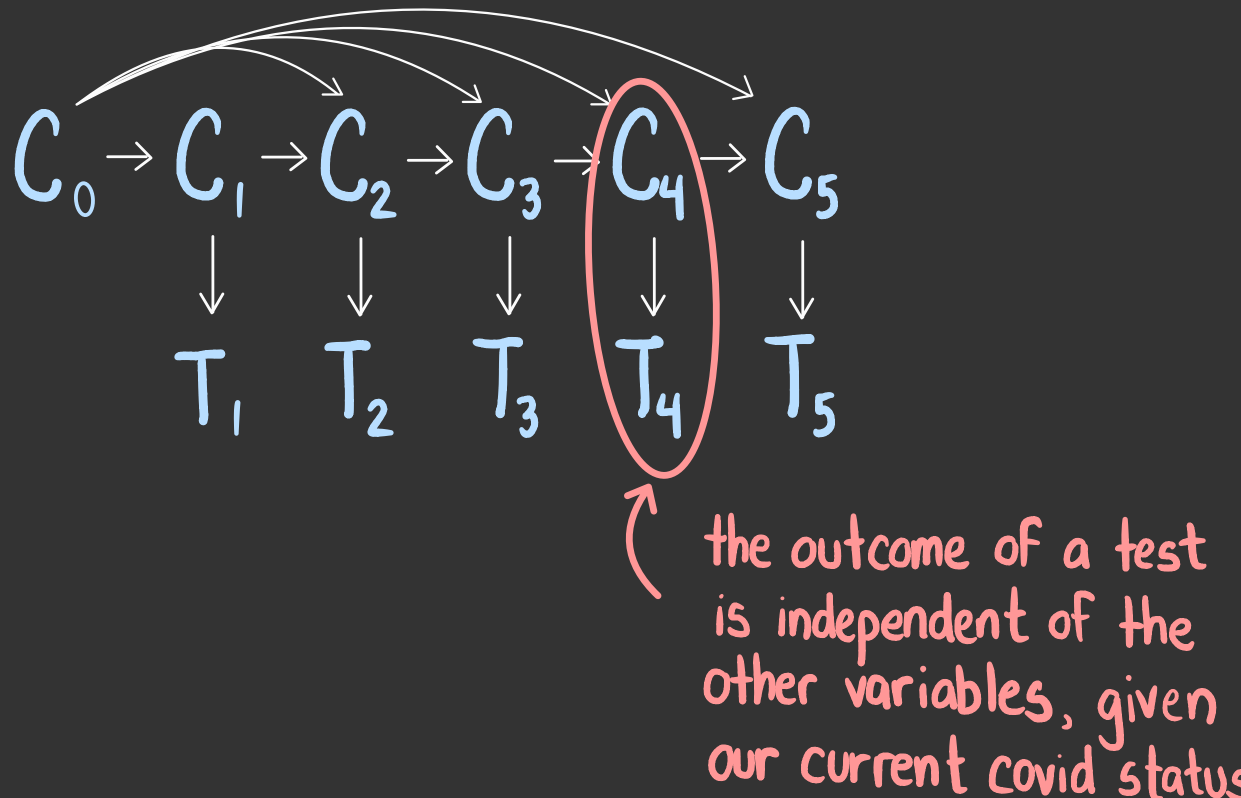


like covid-19

design a bayesian network that allows us to compute the probability of having covid during your first five days on campus

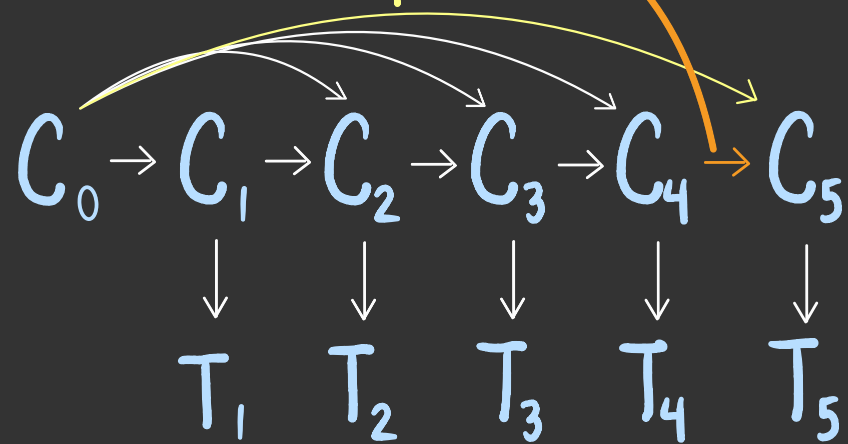
use the following variables:

- $C_0 \in \{-, +\}$ is whether you arrived with covid
- $C_d \in \{-, +\}$ is whether you have covid on day d of being on campus
- $T_d \in \{-, +\}$ is the result of a rapid antigen test taken on day d


C_0
C_1
C_2
C_3
C_4
C_5
T_1
T_2
T_3
T_4
T_5
the outcome of a test is independent of the other variables, given our current covid status

the likelihood of having covid on day d depends on

- whether we already have covid
- how long we've had it



the likelihood of having covid on day d depends on

- whether we already have covid
- ~~how long we've had it~~

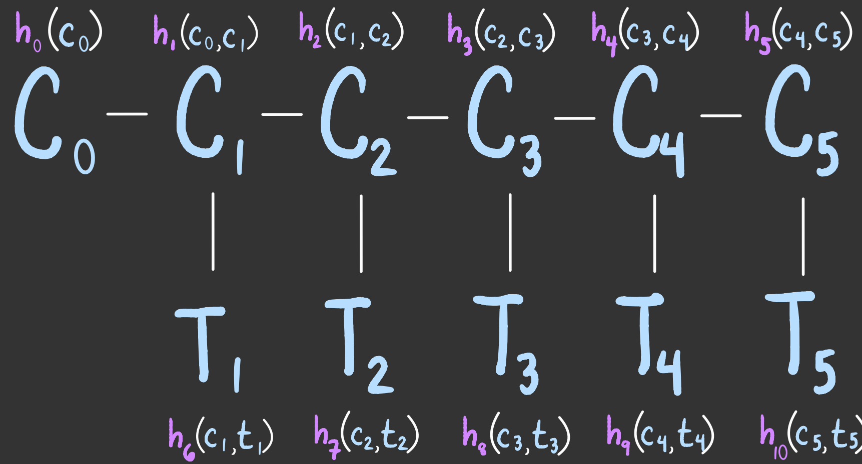
$$
\begin{tikzcd}
C_0 \arrow[r] \arrow[rr, bend left] \arrow[rrr, bend left] \arrow[rrrr, bend left] \arrow[rrrrr, bend left] & C_1 \arrow[r] \arrow[d] & C_2 \arrow[r] \arrow[d] & C_3 \arrow[r] \arrow[d] & C_4 \arrow[r] \arrow[d] & C_5 \arrow[d] \\
 & T_1 & T_2 & T_3 & T_4 & T_5
\end{tikzcd}
$$

why might we make such a simplification?

$$\begin{tikzcd} C_0 \arrow[r] & C_1 \arrow[r] \arrow[d] & C_2 \arrow[r] \arrow[d] & C_3 \arrow[r] \arrow[d] & C_4 \arrow[r] \arrow[d] & C_5 \arrow[d] \\ & T_1 & T_2 & T_3 & T_4 & T_5 \end{tikzcd}$$

Compute the width of elimination order $C_0, T_1, C_1, T_2, C_2, T_3, C_3, T_4, C_4, T_5, C_5$

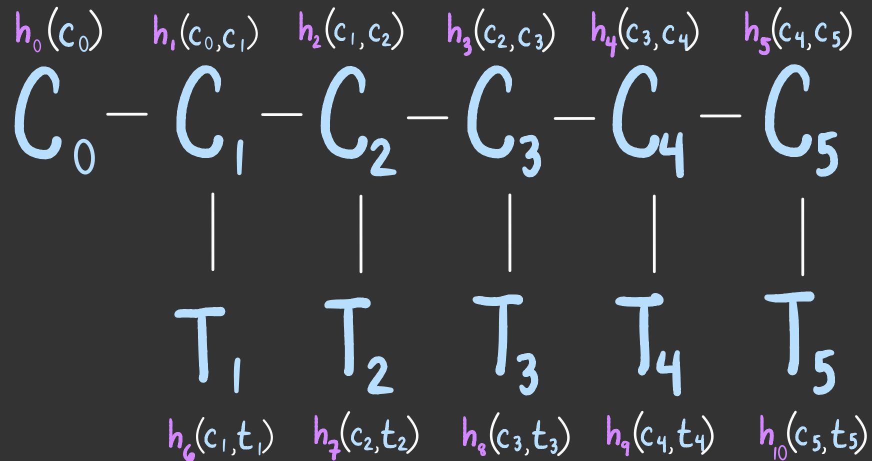
1. create an undirected graph where two variables are **adjacent** if they appear in a **common factor**



Compute the width of elimination order

$C_0, T_1, C_1, T_2, C_2, T_3, C_3, T_4, C_4, T_5, C_5$

2. eliminate each variable by connecting all its neighbors

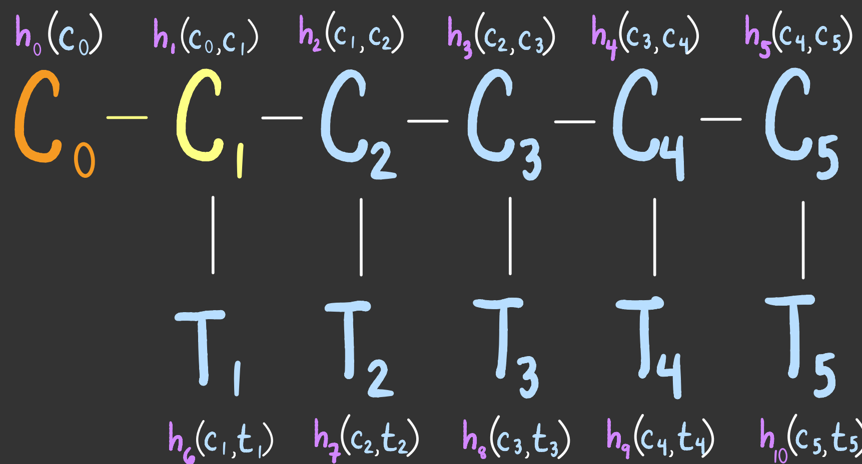


max clique
2

Compute the width of elimination order

$C_0, T_1, C_1, T_2, C_2, T_3, C_3, T_4, C_4, T_5, C_5$

2. eliminate each variable by connecting all its neighbors

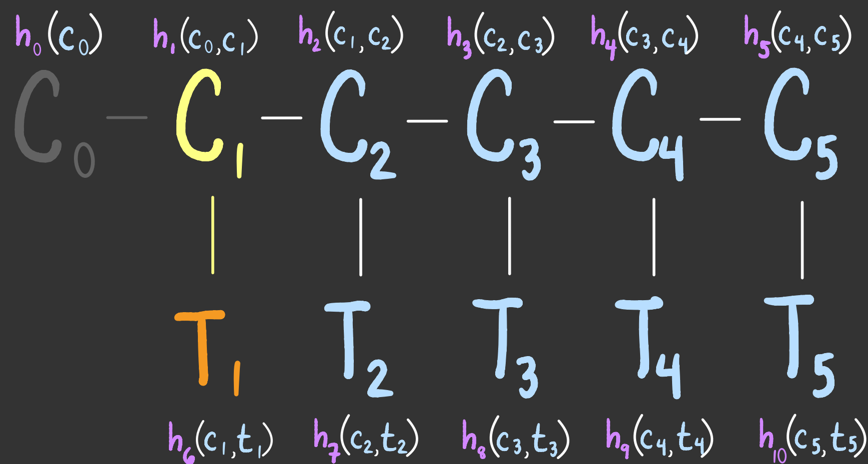


max clique
2

Compute the width of elimination order

$C_0, T_1, C_1, T_2, C_2, T_3, C_3, T_4, C_4, T_5, C_5$

2. eliminate each variable by connecting all its neighbors

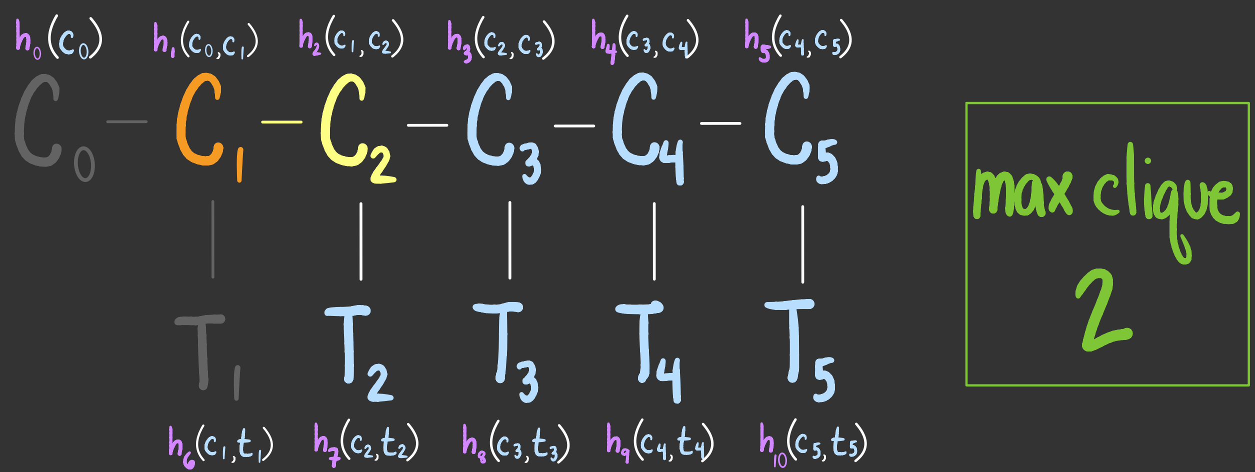


max clique
2

Compute the width of elimination order

$C_0, T_1, C_1, T_2, C_2, T_3, C_3, T_4, C_4, T_5, C_5$

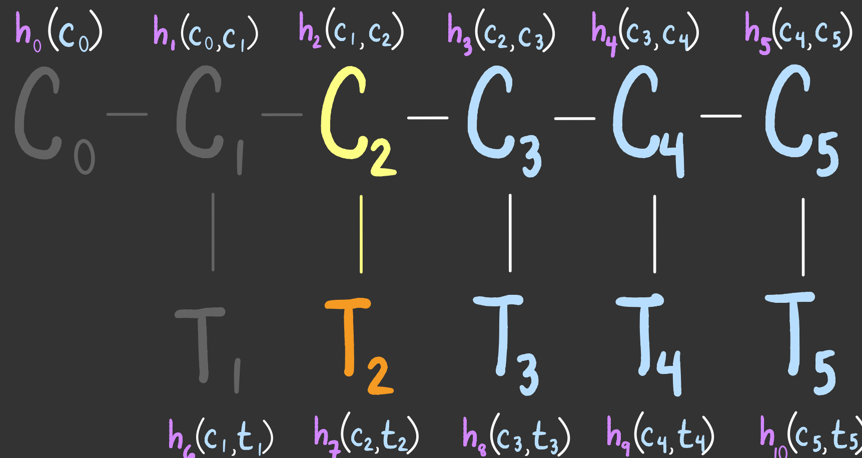
2. eliminate each variable by connecting all its neighbors



Compute the width of elimination order

$C_0, T_1, C_1, T_2, C_2, T_3, C_3, T_4, C_4, T_5, C_5$

2. eliminate each variable by connecting all its neighbors

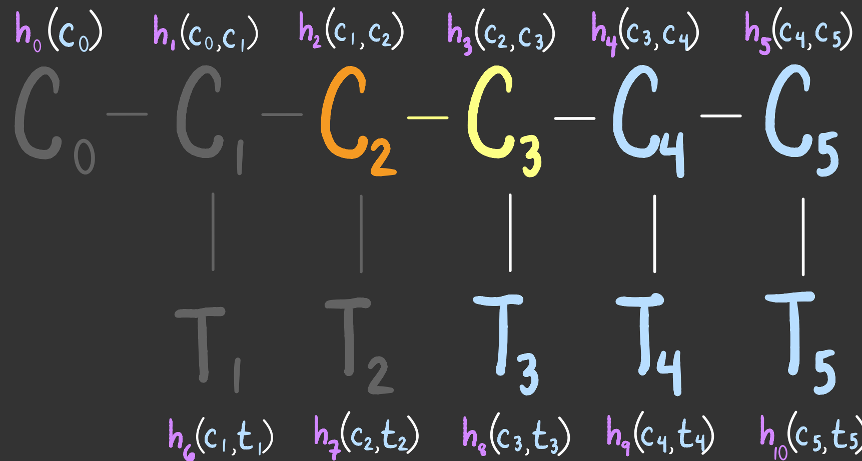


max clique
2

Compute the width of elimination order

$C_0, T_1, C_1, T_2, C_2, T_3, C_3, T_4, C_4, T_5, C_5$

2. eliminate each variable by connecting all its neighbors

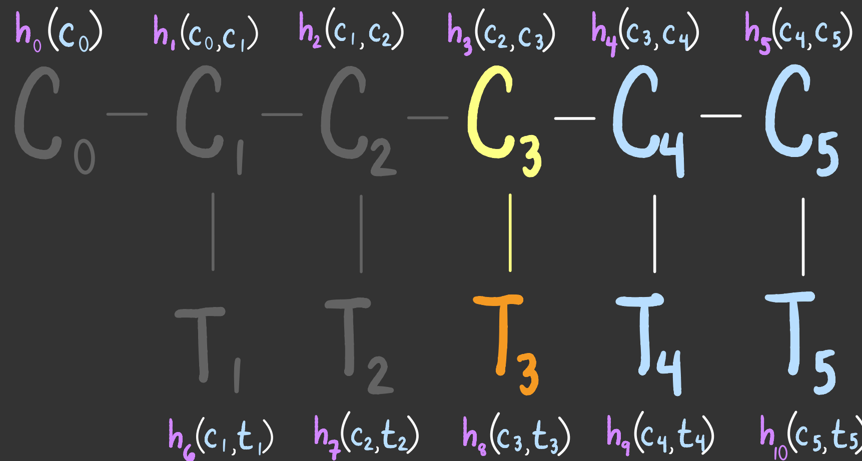


max clique
2

Compute the width of elimination order

$C_0, T_1, C_1, T_2, C_2, T_3, C_3, T_4, C_4, T_5, C_5$

2. eliminate each variable by connecting all its neighbors

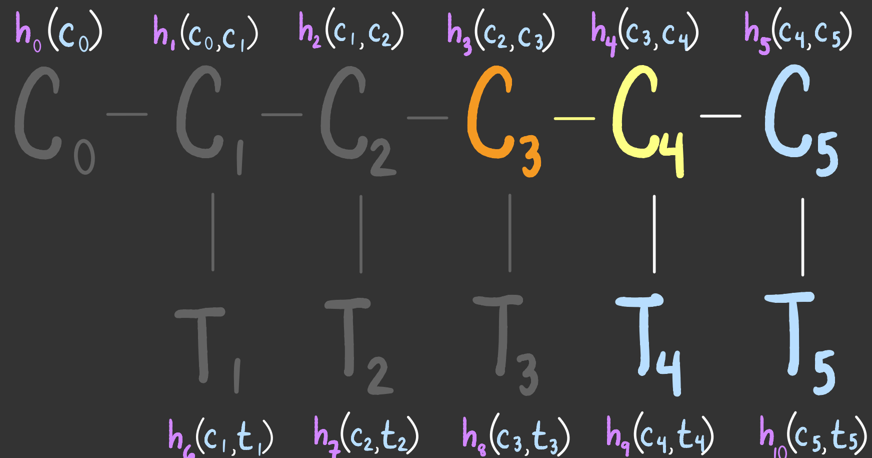


max clique
2

Compute the width of elimination order

$C_0, T_1, C_1, T_2, C_2, T_3, C_3, T_4, C_4, T_5, C_5$

2. eliminate each variable by connecting all its neighbors

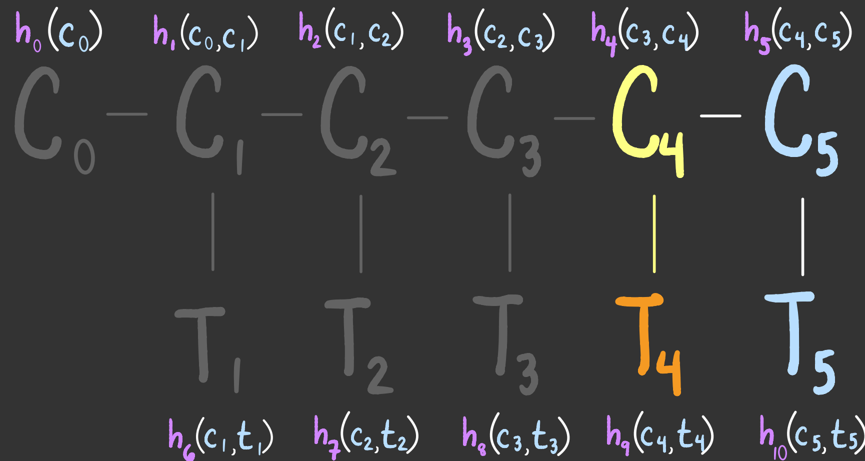


max clique
2

Compute the width of elimination order

$C_0, T_1, C_1, T_2, C_2, T_3, C_3, T_4, C_4, T_5, C_5$

2. eliminate each variable by connecting all its neighbors



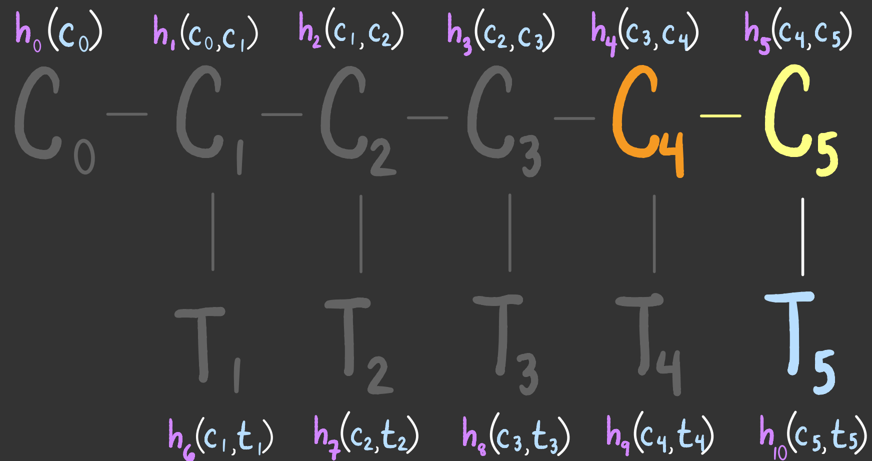
max clique

2

Compute the width of elimination order

$C_0, T_1, C_1, T_2, C_2, T_3, C_3, T_4, C_4, T_5, C_5$

2. eliminate each variable by connecting all its neighbors



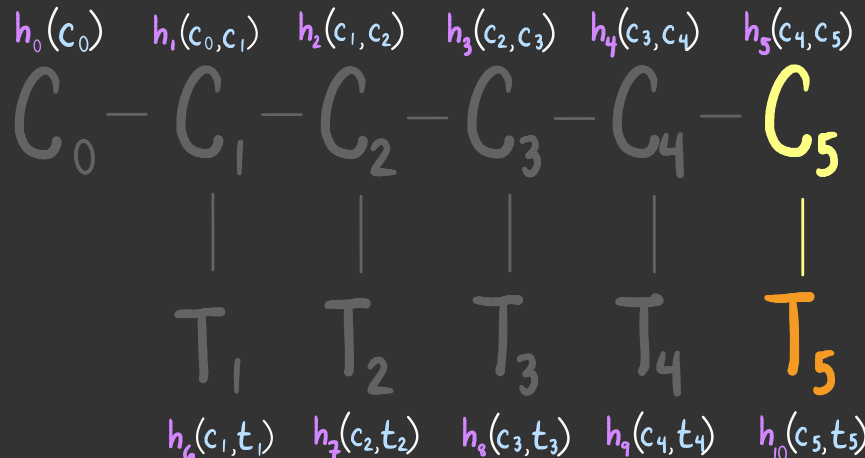
max clique

2

Compute the width of elimination order

$C_0, T_1, C_1, T_2, C_2, T_3, C_3, T_4, C_4, T_5, C_5$

2. eliminate each variable by connecting all its neighbors

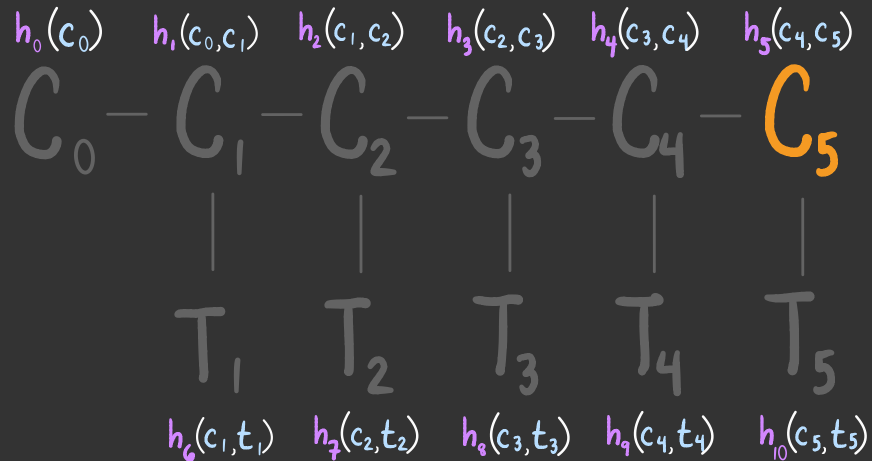


max clique
2

Compute the width of elimination order

$C_0, T_1, C_1, T_2, C_2, T_3, C_3, T_4, C_4, T_5, C_5$

2. eliminate each variable by connecting all its neighbors

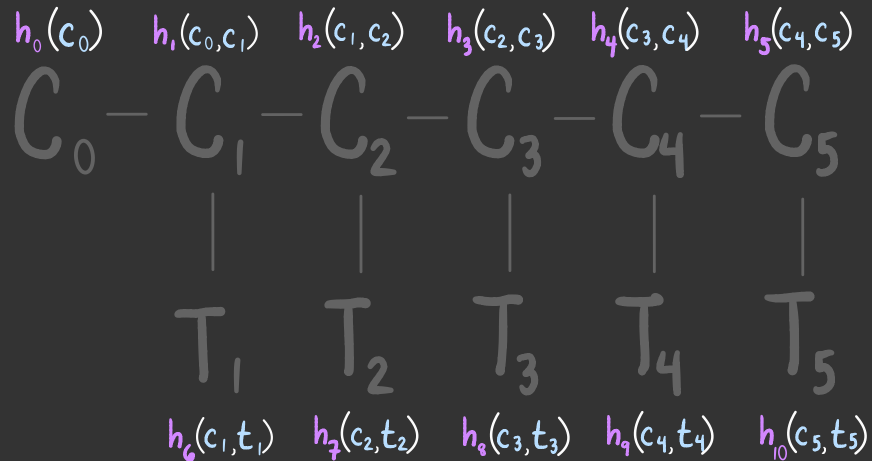


max clique
2

Compute the width of elimination order

$C_0, T_1, C_1, T_2, C_2, T_3, C_3, T_4, C_4, T_5, C_5$

2. eliminate each variable by connecting all its neighbors



width
2

Compute the width of elimination order
$C_0, T_1, C_1, T_2, C_2, T_3, C_3, T_4, C_4, T_5, C_5$

recall:

the worst-case runtime of variable elimination is $O(nd^{w})$, where

- n is the number of bayesian network variables
- d is the size of the largest variable domain
- w is the width of the elimination order

for:

$$\begin{tikzcd}
C_0 \arrow[r] & C_1 \arrow[r] \arrow[d] & C_2 \arrow[r] \arrow[d] & C_3 \arrow[r] \arrow[d] & C_4 \arrow[r] \arrow[d] & C_5 \arrow[d] \\
 & T_1 & T_2 & T_3 & T_4 & T_5
\end{tikzcd}$$

$n = \boxed{?}$

$d = \boxed{?}$

$w = \boxed{?}$

recall:

the worst-case runtime of variable elimination is $O(nd^{w})$, where

- n is the number of bayesian network variables
- d is the size of the largest variable domain
- w is the width of the elimination order

for:

$$\begin{tikzcd}
C_0 \arrow[r] & C_1 \arrow[r] \arrow[d] & C_2 \arrow[r] \arrow[d] & C_3 \arrow[r] \arrow[d] & C_4 \arrow[r] \arrow[d] & C_5 \arrow[d] \\
& T_1 & T_2 & T_3 & T_4 & T_5
\end{tikzcd}$$

$n = 1 + 2 \cdot \underbrace{\text{number of days}}_{\text{call this } m}$

$d = 2$

$w = 2$

so the worst-case runtime is $O((1+2m)\,2^2) = O(4+8m) = O(m)$

so we can run variable elimination on any bayesian network with this graph structure

$$\begin{tikzcd}
C_0 \arrow[r] & C_1 \arrow[r] \arrow[d] & C_2 \arrow[r] \arrow[d] & C_3 \arrow[r] \arrow[d] & \cdots \arrow[r] & C_m \arrow[d] \\
& T_1 & T_2 & T_3 & & T_m
\end{tikzcd}$$

in time $O(m)$

this would not have been the case without our earlier simplifying assumption

$$\begin{tikzcd}
C_0 \arrow[r] \arrow[rr, bend left] \arrow[rrr, bend left] \arrow[rrrr, bend left] \arrow[rrrrr, bend left] & C_1 \arrow[r] \arrow[d] & C_2 \arrow[r] \arrow[d] & C_3 \arrow[r] \arrow[d] & \cdots \arrow[r] & C_m \arrow[d] \\
 & T_1 & T_2 & T_3 & & T_m
\end{tikzcd}$$

this assumption is called a

markov assumption

$$
\begin{tikzcd}
C_0 \arrow[r] \arrow[rr, bend left] \arrow[rrr, bend left] \arrow[rrrr, bend left] \arrow[rrrrr, bend left] & C_1 \arrow[r] \arrow[d] & C_2 \arrow[r] \arrow[d] & C_3 \arrow[r] \arrow[d] & \cdots \arrow[r] & C_m \arrow[d] \\
 & T_1 & T_2 & T_3 & & T_m
\end{tikzcd}
$$

namely, the probability of our current state depends only on the previous state

markov assumption: the probability of our current state depends only on the previous state

$$\begin{tikzcd}
C_0 \arrow[r] & C_1 \arrow[r] \arrow[d] & C_2 \arrow[r] \arrow[d] & C_3 \arrow[r] \arrow[d] & \cdots \arrow[r] & C_m \arrow[d] \\
& T_1 & T_2 & T_3 & & T_m
\end{tikzcd}$$

how would this be expressed as a conditional independence statement?

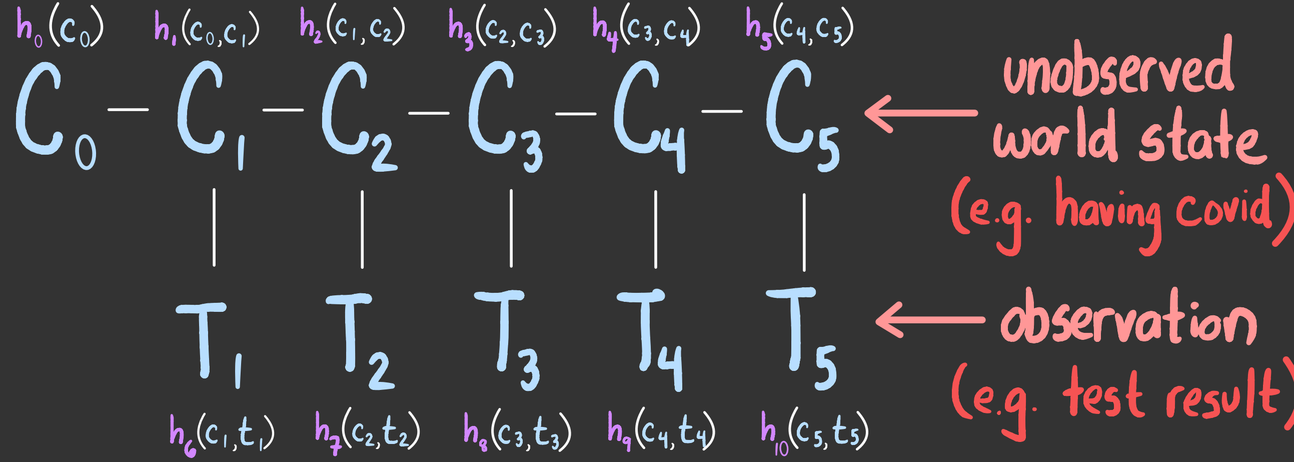
$$\boxed{\;?\;} \perp\!\!\!\perp \boxed{\qquad ? \qquad} \mid \boxed{\;?\;}$$

markov assumption: the probability of our current state depends only on the previous state

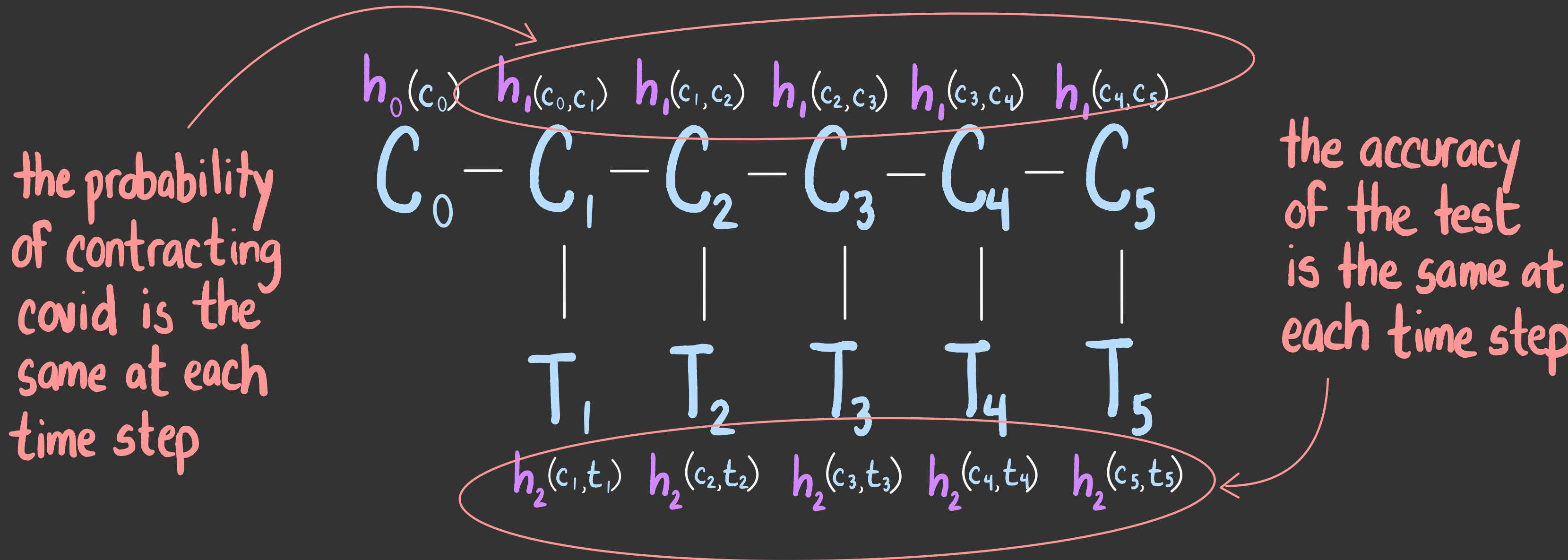
$$\begin{tikzcd}
C_0 \arrow[r] & C_1 \arrow[r] \arrow[d] & C_2 \arrow[r] \arrow[d] & C_3 \arrow[r] \arrow[d] & \cdots \arrow[r] & C_m \arrow[d] \\
 & T_1 & T_2 & T_3 & & T_m
\end{tikzcd}$$

how would this be expressed as a conditional independence statement?

$$C_i \perp\!\!\!\perp \{C_0, \ldots, C_{i-2}\} \mid C_{i-1}$$

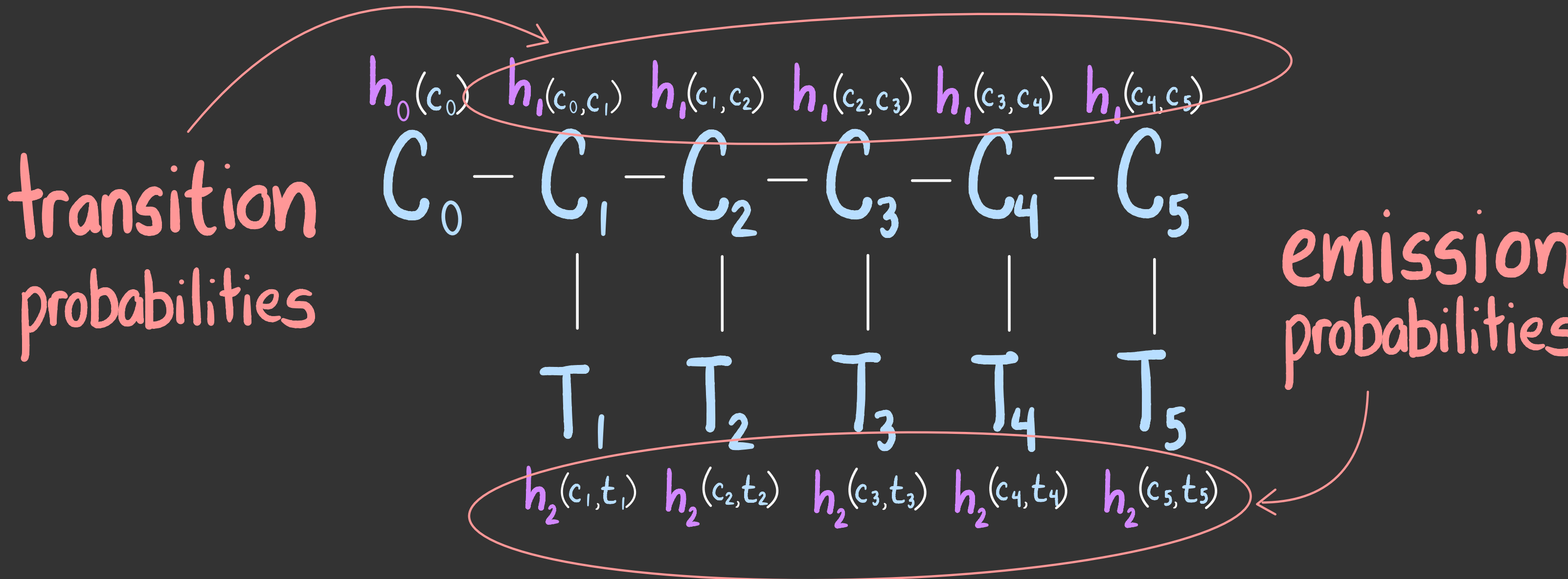


this kind of bayesian network is popular for temporal reasoning when we have noisy observations about the world



for such models, typically we make the additional assumption of

time-invariance



such bayesian networks are called

hidden markov models