Review Session (Variable-Based Models)

CS 221 AUTUMN 2019

CHRISTOPHER WAITES

JON KOTKER

What are we covering today?

CONSTRAINT SATISFACTION PROBLEMS

BAYESIAN NETWORKS

CSPs: Definitions

Key idea: variables

- Solutions to problems ⇒ assignments to variables (modeling).
- Decisions about variable ordering, etc. chosen by inference.

We have **factors** (constraints) and **inputs**.

Setting these inputs is an **assignment**.

Factor Graphs: How good is our assignment? (maximum weight assignment)

Constraint Satisfaction: Is this a valid assignment? (yes/no; is weight 1?)

CSPs: Inference

Backtracking: Try everything and prune.

Algorithm: backtracking search

Backtrack(x, w, Domains):

- If x is complete assignment: update best and return
- Choose unassigned **VARIABLE** X_i
- Order **VALUES** Domain$_i$ of chosen X_i
- For each value v in that order:
  - $\delta \leftarrow \prod_{f_j \in D(x, X_i)} f_j(x \cup \{X_i : v\})$
  - If $\delta = 0$: continue
  - Domains$'$ $\leftarrow$ Domains via **LOOKAHEAD**
  - Backtrack($x \cup \{X_i : v\}, w\delta$, Domains$'$)

CSPs: Inference

AC-3: Pre-emptively prune with lookahead.

Algorithm: AC-3

Add X_j to set.

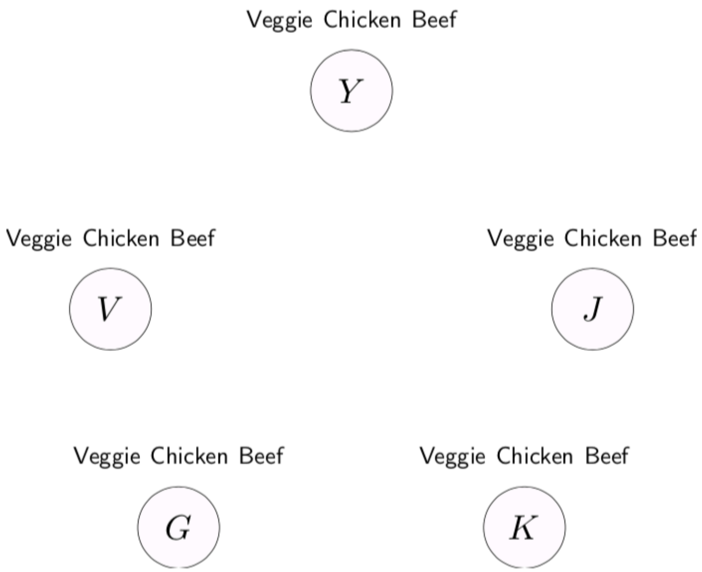
While set is non-empty:

- Remove any X_k from set.
- For all neighbors X_l of X_k:
  - Enforce arc consistency on X_l w.r.t. X_k.
  - If Domain_l changed, add X_l to set.

CSPs: Sample Question
(Fall 2014)

It's Friday night, and you and your friends go out to dinner in anticipation of the Big Game the next day (go Card!). Since you've just been working on your final project for CS221, you are seeing CSPs and Bayesian networks everywhere!

a. (*10 points*)
You and your friends (Veronica, Jarvis, Gabriela, Kanti) sit around a table like this:



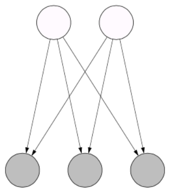
CSPs: Sample Question (Fall 2014)

There are three dishes on the menu: the vegetarian deep dish pizza, the chicken quesadilla, and the beef cheeseburger. Each person will order exactly one dish.

But what started out as a simple dinner has quickly turned into a logistical nightmare because of all the constraints you and your friends impose upon yourselves:

1. Each person must order something different than the people sitting immediately next to him/her.
2. You (Y) are vegetarian.
3. If Veronica (V) orders beef, then Jarvis (J) will order veggie.
4. Kanti (K) and Jarvis (J) cannot both get non-chicken dishes.

Draw the potentials for the above constraints and write the propositional formula above each potential (e.g., [Y = Veggie]). Then for each pair of variables, enforce arc consistency in both directions, crossing out the appropriate values from the domains.



Definition: Bayesian network

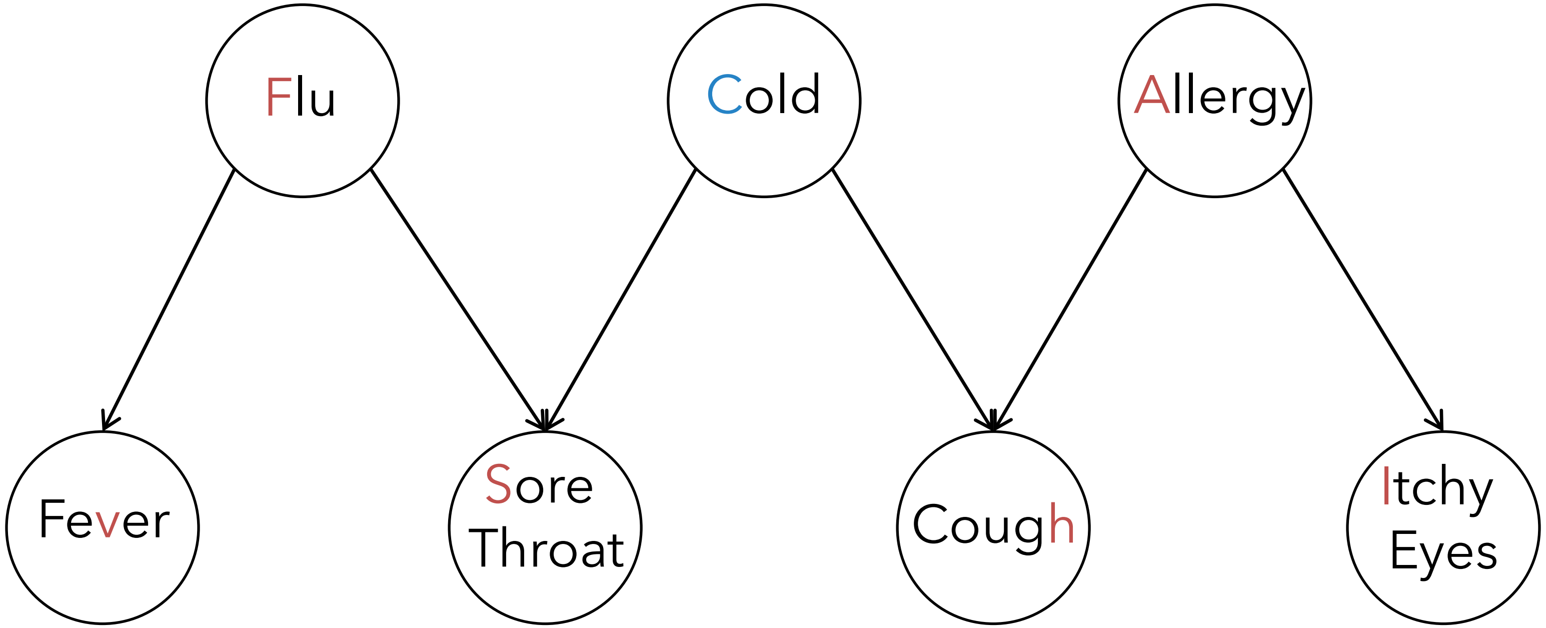
Let $X = (X_1, \ldots, X_n)$ be random variables.

A **Bayesian network** is a directed acyclic graph (DAG) that specifies a joint distribution over X as a product of local conditional distributions, one for each node:

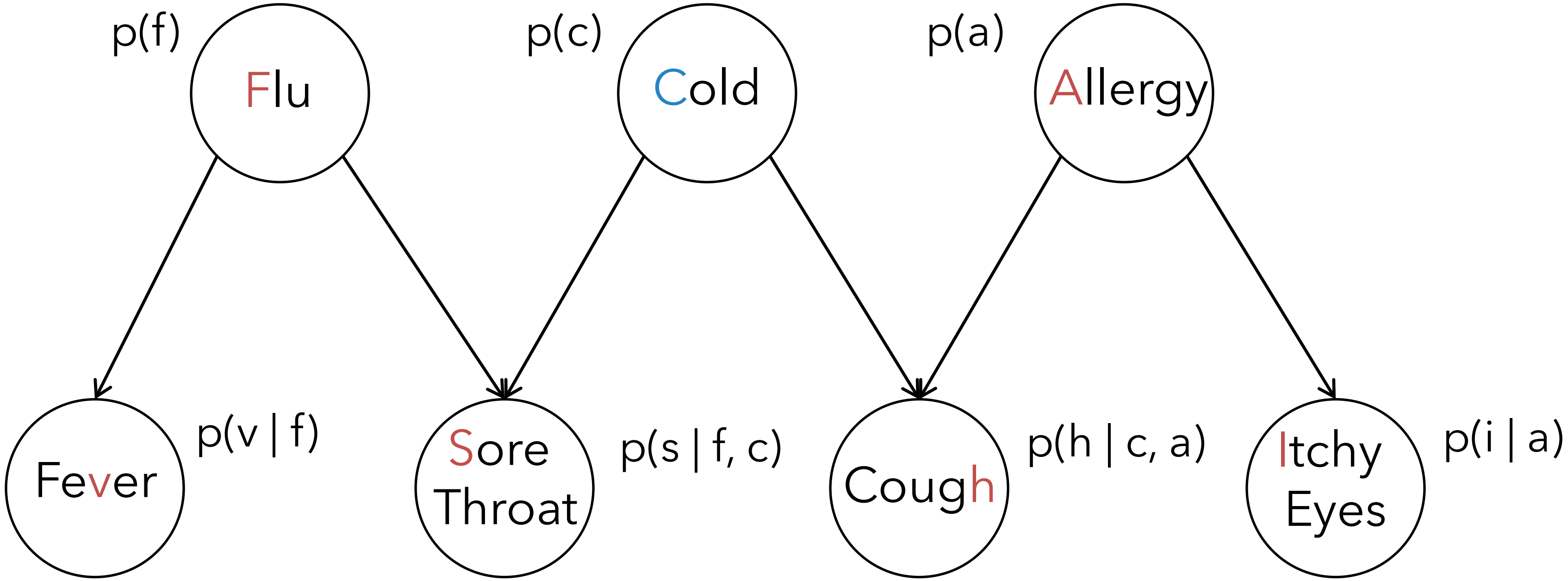
$$\mathbb{P}(X_1 = x_1, \ldots, X_n = x_n) = \prod_{i=1}^{n} p(x_i \mid x_{\text{Parents}(i)})$$

Bayesian Networks

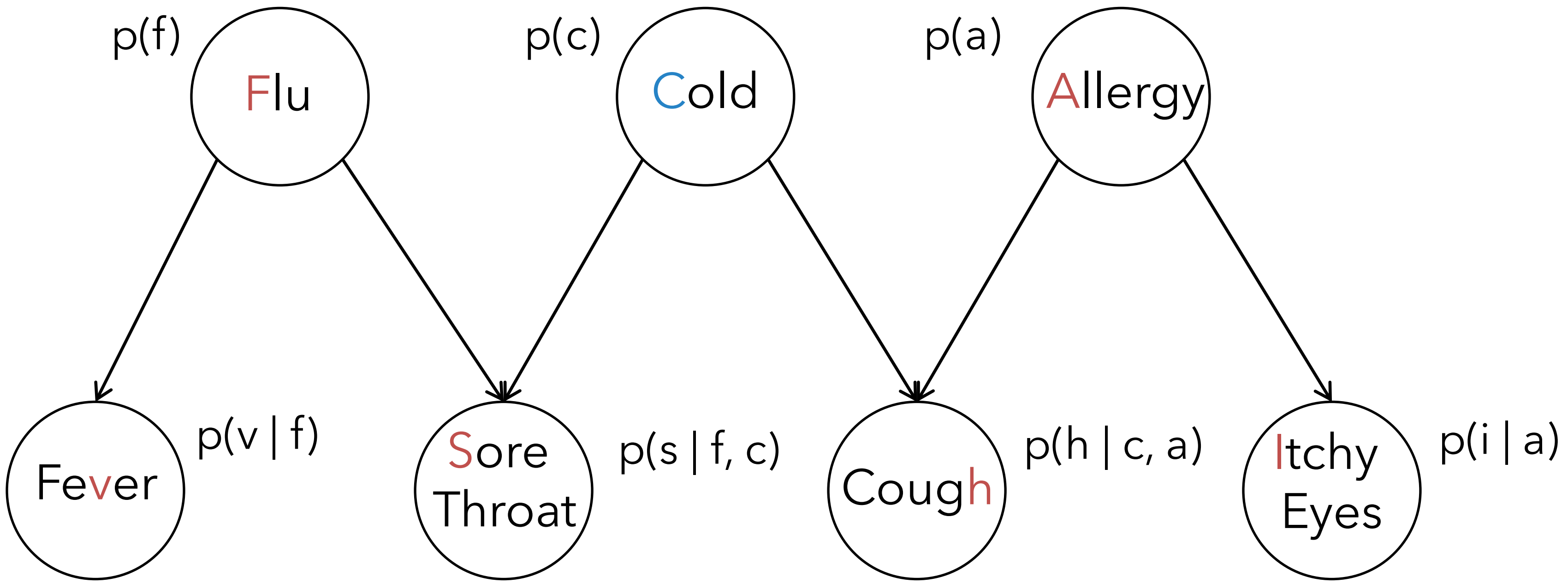
Bayesian Networks



A Bayesian network represents a joint probability distribution.



A Bayesian network represents a joint probability distribution.



P(F=f, C=c, A=a, V=v, S=s, H=h, I=i) = p(f) p(c) p(a) p(v | f) p(s | f, c) p(h |c, a) p(i | a)

Probabilistic Inference Cookbook

Given a query $P(Q \mid E = e)$:

1. Remove (marginalize) variables not ancestors of Q or E.
2. Convert Bayesian network to factor graph.
3. Condition (shade nodes / disconnect) on $E = e$.
4. Remove (marginalize) nodes disconnected from Q.
5. Run probabilistic inference algorithm (manual, variable elimination, Gibbs sampling, particle filtering).

3. The Bayesian Bag of Candies Model (*50 points*)

You have a lot of candy left over from Halloween, and you decide to give them away to your friends. You have four types of candy: **A**pple, **B**anana, **C**aramel, **D**ark-Chocolate. You decide to prepare candy bags using the following process.

- For each candy bag, you first flip a (biased) coin Y which comes up heads ($Y = \mathrm{H}$) with probability λ and tails ($Y = \mathrm{T}$) with probability $1 - \lambda$.
- If Y comes up heads ($Y = \mathrm{H}$), you make a **H**ealthy bag, where you:
  1. Add one **A**pple candy with probability p_1 or nothing with probability $1 - p_1$;
  2. Add one **B**anana candy with probability p_1 or nothing with probability $1 - p_1$;
  3. Add one **C**aramel candy with probability $1 - p_1$ or nothing with probability p_1;
  4. Add one **D**ark-Chocolate candy with probability $1 - p_1$ or nothing with probability p_1.
- If Y comes up tails ($Y = \mathrm{T}$), you make a **T**asty bag, where you:
  1. Add one **A**pple candy with probability p_2 or nothing with probability $1 - p_2$;
  2. Add one **B**anana candy with probability p_2 or nothing with probability $1 - p_2$;
  3. Add one **C**aramel candy with probability $1 - p_2$ or nothing with probability p_2;
  4. Add one **D**ark-Chocolate candy with probability $1 - p_2$ or nothing with probability p_2.

For example, if $p_1 = 1$ and $p_2 = 0$, you would deterministically generate: **H**ealthy bags with one **A**pple and one **B**anana; and **T**asty bags with one **C**aramel and one **D**ark-Chocolate. For general values of p_1 and p_2, bags can contain anywhere between 0 and 4 pieces of candy.

Denote A, B, C, D random variables indicating whether or not the bag contains candy of type **A**pple, **B**anana, **C**aramel, and **D**ark-Chocolate, respectively.

Bayesian Networks: Sample Question

(Fall 2017)

Bayesian Networks: MLE

Input: training examples $\mathcal{D}_{\text{train}}$ of full assignments

Output: parameters $\theta = \{p_d : d \in D\}$

Algorithm: maximum likelihood for Bayesian networks

Count:

For each $x \in \mathcal{D}_{\text{train}}$:

For each variable x_i:

Increment $\text{count}_{d_i}(x_{\text{Parents}(i)}, x_i)$

Normalize:

For each d and local assignment $x_{\text{Parents}(i)}$:

Set $p_d(x_i \mid x_{\text{Parents}(i)}) \propto \text{count}_d(x_{\text{Parents}(i)}, x_i)$

Bayesian Networks: Sample Question (Fall 2017)

b. (*10 points*)

You realize you need to make more candy bags, but you've forgotten the probabilities you used to generate them. So you try to estimate them looking at the 5 bags you've already made:

$$
\begin{aligned}
&bag\ 1: && (\mathbf{H}\text{ealthy}, \{\mathbf{A}\text{pple}, \mathbf{B}\text{anana}\}) \\
&bag\ 2: && (\mathbf{T}\text{asty}, \{\mathbf{C}\text{aramel}, \mathbf{D}\text{ark-Chocolate}\}) \\
&bag\ 3: && (\mathbf{H}\text{ealthy}, \{\mathbf{A}\text{pple}, \mathbf{B}\text{anana}\}) \\
&bag\ 4: && (\mathbf{T}\text{asty}, \{\mathbf{C}\text{aramel}, \mathbf{D}\text{ark-Chocolate}\}) \\
&bag\ 5: && (\mathbf{H}\text{ealthy}, \{\mathbf{A}\text{pple}, \mathbf{B}\text{anana}\})
\end{aligned}
$$

Estimate λ, p_1, p_2 by maximum likelihood.

Bayesian Networks: Smoothing

Key idea: Hallucinate λ extra observations.

Assumption: All values we listed are possible.

Maximum likelihood:

$$p(\mathrm{H}) = \frac{1}{1} \quad p(\mathrm{T}) = \frac{0}{1}$$

Maximum likelihood with Laplace smoothing:

$$p(\mathrm{H}) = \frac{1+1}{1+2} = \frac{2}{3} \quad p(\mathrm{T}) = \frac{0+1}{1+2} = \frac{1}{3}$$

Bayesian Networks: EM algorithm

E-step:

- Compute $q(h) = \mathbb{P}(H = h \mid E = e; \theta)$ for each h (use any probabilistic inference algorithm)
- Create weighted points: (h, e) with weight $q(h)$

M-step:

- Compute maximum likelihood (just count and normalize) to get θ

Repeat until convergence.

Bayesian Networks: Sample Question (Fall 2017)

c. (*15 points*) You find out your little brother had been playing with your candy bags, and had mixed them up (in a uniformly random way). Now you don't even know which ones were **H**ealthy and which ones were **T**asty. So you need to re-estimate λ, p_1, p_2, but now without knowing whether the bags were **H**ealthy or **T**asty.

$$\begin{aligned}
&bag\ 1: && (?\ ,\ \{\mathbf{A}\text{pple}, \mathbf{B}\text{anana}, \mathbf{C}\text{aramel}\}) \\
&bag\ 2: && (?\ ,\ \{\mathbf{C}\text{aramel}, \mathbf{D}\text{ark-Chocolate}\}) \\
&bag\ 3: && (?\ ,\ \{\mathbf{A}\text{pple}, \mathbf{B}\text{anana}, \mathbf{C}\text{aramel}\}) \\
&bag\ 4: && (?\ ,\ \{\mathbf{C}\text{aramel}, \mathbf{D}\text{ark-Chocolate}\}) \\
&bag\ 5: && (?\ ,\ \{\mathbf{A}\text{pple}, \mathbf{B}\text{anana}, \mathbf{C}\text{aramel}\})
\end{aligned}$$

You remember the EM algorithm is just what you need. Initialize with $\lambda = 0.5, p_1 = 0.5, p_2 = 0$, and run one step of the EM algorithm.