CS221 Problem Workout

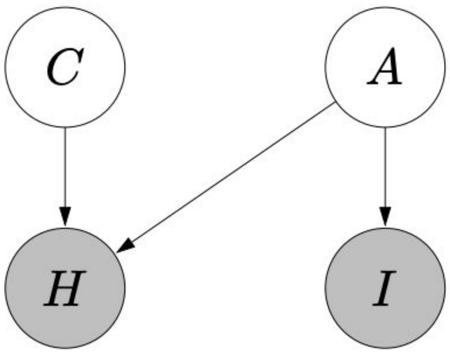
Week 8

Stanford University

Outline

- **Bayesian networks: Learning**
  - Maximum likelihood
  - Smoothing
  - EM Algorithm
- Problem discussion

Bayesian networks: Learning



Given local probability distributions, i.e. P(x | parents(x))

Find conditional P(Q | E=e)

Inference

Given observations / samples

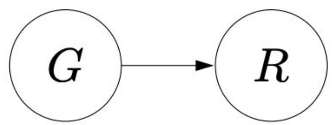
Find the local distributions, i.e. P(x | Parents(x))

Learning

Example

Variables:

- Genre $G \in \{\text{drama}, \text{comedy}\}$
- Rating $R \in \{1, 2, 3, 4, 5\}$



$$\mathbb{P}(G = g, R = r) = p_G(g) p_R(r \mid g)$$

$$\mathcal{D}_{\text{train}} = \{(\text{d}, 4), (\text{d}, 4), (\text{d}, 5), (\text{c}, 1), (\text{c}, 5)\}$$

Parameters: $\theta = (p_G, p_R)$

Example borrowed from lecture slides

Outline

- Bayesian networks: Learning
  - **Maximum likelihood**
  - Smoothing
  - EM Algorithm
- Problem discussion

Maximum likelihood

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

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

Algorithm: count and normalize

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

Slide borrowed from lecture slides

Outline

- Bayesian networks: Learning
  - Maximum likelihood
  - **Smoothing**
  - EM Algorithm
- Problem discussion

Smoothing

Why?

- What if count is 0? Should P be 0?

How to solve?

- Initialize all counts with a non-zero constant $\boldsymbol{\lambda}$

Observations

- Larger $\boldsymbol{\lambda}$ -> more uniform distributions, less influenced by data
- Smaller $\boldsymbol{\lambda}$ -> more influenced by data
- Infinite data -> effect of $\boldsymbol{\lambda}$ vanishes

Final algorithm

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

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

Algorithm: count and normalize

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

Slide modified from lecture slides

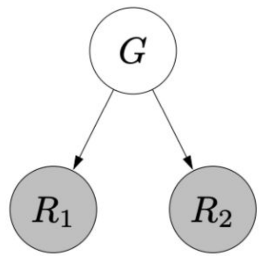
Outline

- Bayesian networks: Learning
  - Maximum likelihood
  - Smoothing
  - **EM Algorithm**
- Problem discussion

EM Algorithm

Variables: H is hidden, $E = e$ is observed

Example:



$H = G \qquad E = (R_1, R_2) \qquad e = (1, 2)$

$\theta = (p_G, p_R)$

Maximum marginal likelihood objective:

$$\max_\theta \prod_{e \in \mathcal{D}_{\text{train}}} \mathbb{P}(E = e; \theta)$$

$$= \max_\theta \prod_{e \in \mathcal{D}_{\text{train}}} \sum_h \mathbb{P}(H = h, E = e; \theta)$$

Slide borrowed from lecture slides

EM Algorithm

Initialize θ randomly

Until convergence:

 # E Step

for each e in Data:

for each h:

$q(h; e) = P(H = h \mid E = e; \theta)$ … inference

Update table from {e, count(e)} to {(h,e), (q(h; e) x count(e)}

Now no variables are hidden

 # M step

update(θ) using Table {(h,e), (q(h; e) x count(e)} … MLE

Summary

- Given data learn the parameters of bayesian net

MLE

$p \propto \text{count}(x_i; \text{parents}(x_i))$

Smoothing

$p \propto \text{count}(x_i; \text{parents}(x_i)) + \boldsymbol{\lambda}$

EM

Data is *incomplete*

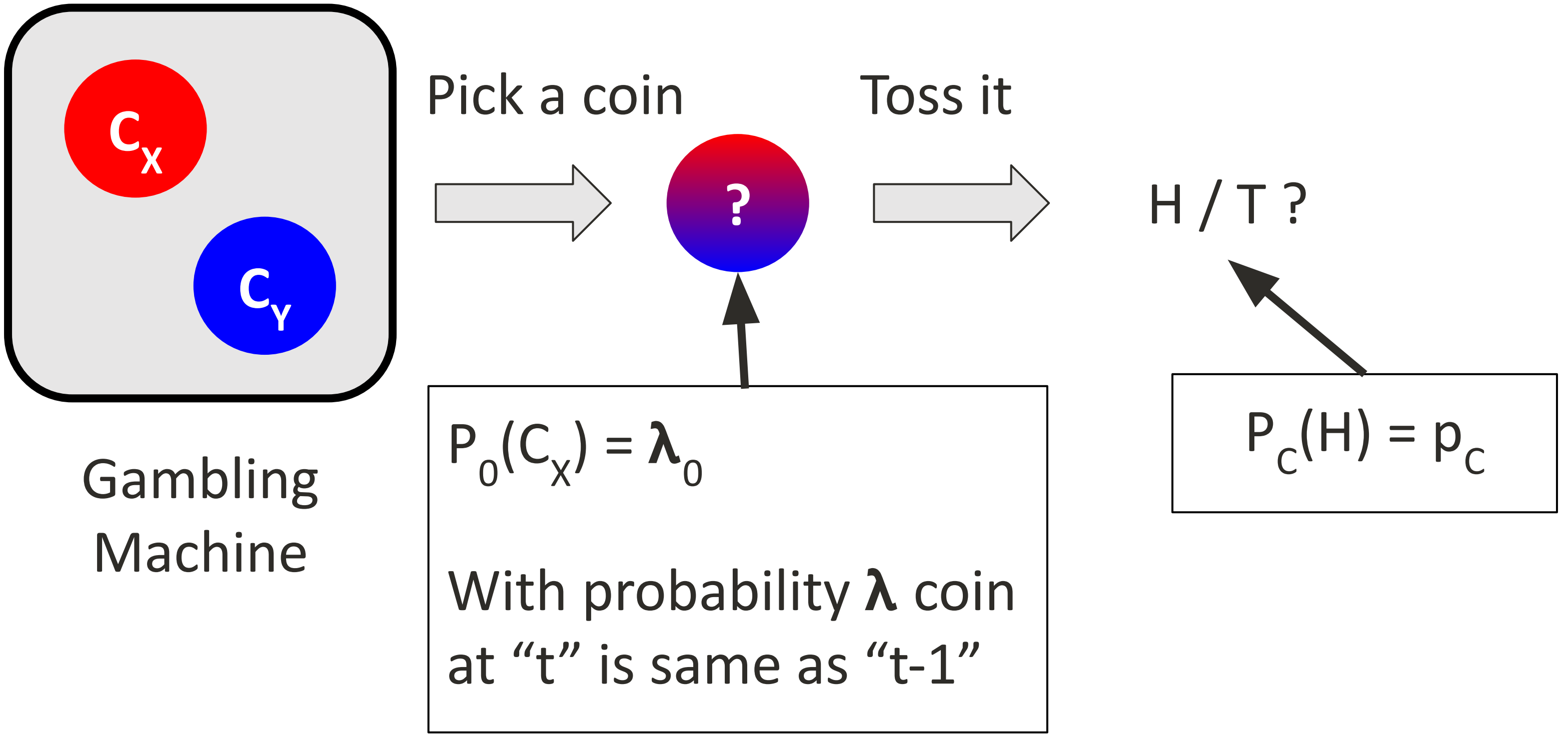
E step: compute counts

M step: MLE

Outline

- Bayesian networks: Learning
  - Maximum likelihood
  - Smoothing
  - EM Algorithm
- **Problem discussion**

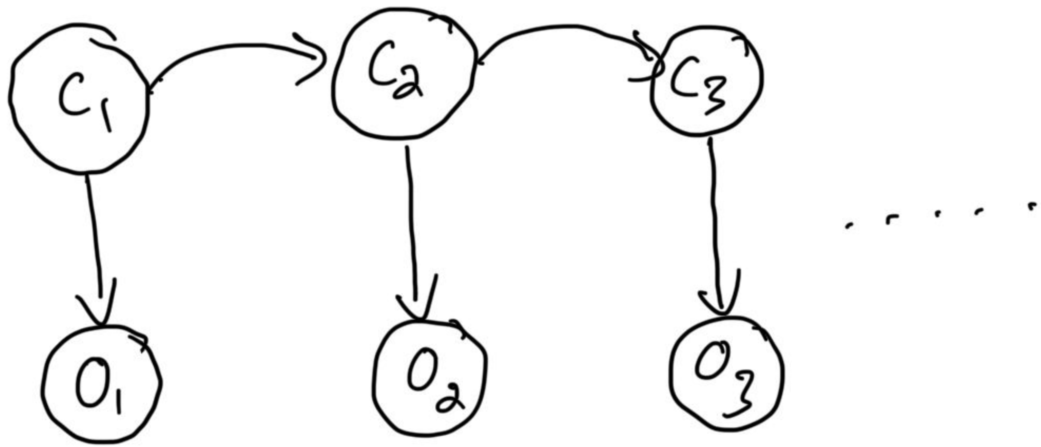
Problem: P2, Winter 2021 Exam 2



How does the bayesian net look?

$$P(C_1) = \begin{cases} \lambda_0 & \text{if } C_1 = X \\ 1-\lambda_0 & \text{else} \end{cases}$$

$$P(C_{i+1} \mid C_i) = \begin{cases} \lambda & \text{if } C_{i+1} = C_i \\ 1-\lambda & \text{else} \end{cases}$$



$$P(O_i \mid C_i) = \begin{cases} p_{C_i} & \text{if } O_i = H \\ 1-p_{C_i} & \text{else} \end{cases}$$

Learning using EM algorithm

- Data = {H, H, T}
- $\boldsymbol{\lambda}_0$ and $\boldsymbol{\lambda}$ are given. To find: p_X and p_Y

- Why do we need EM?
  - What is not observed?
  - C_i is not observed

- How do we use EM?
  - Compute $q(c_i)$ using p'_X and p'_Y
  - Use ML to update p'_X and p'_Y

Given q's compute update

	T=1	T=2	T=3
X	0.1	0.5	0.3
Y	0.9	0.5	0.7

Data = {H, H, T}

Compute:

C_i	O_i	Count
?	?	?

Given q's compute update

	T=1	T=2	T=3
X	0.1	0.5	0.3
Y	0.9	0.5	0.7

Data = {H, H, T}

C_i	O_i	Count
X	H	0.1
X	H	0.5
X	T	0.3

$$P(H \mid X) = p'_X$$
$$= (0.1 + 0.5) / (0.1 + 0.5 + 0.3) \ldots \text{MLE}$$

Thank You