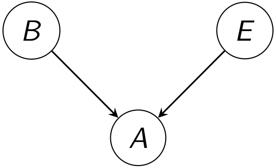
CS-541: Artificial Intelligence
Lecture 9

Abdul Rafae Khan
Department of Computer Science
Stevens Institute of Technology
akhan4@stevens.edu

April 18, 2022

Recap: Bayesian Network



Let $X = (X_1, \cdots, X_n)$ are random variables
Bayesian Network is a directed acyclic graph which specifies joint distribution over X as a product of local conditional distributions

$$\mathbb{P}(X_1 = x_1, \cdots, X_n = x_n) \stackrel{\text{def}}{=\!=} \prod_{i=1}^{n} p(x_i | x_{parents(i)})$$

Recap: Probabilistic Inference

Input

Bayesian network: $\mathbb{P}(X_1, \cdots, X_n)$

Evidence: $E = e$ where $E \subseteq X$ is subset of variables

Query: $Q \subseteq X$ is subset of variables

$\Downarrow$

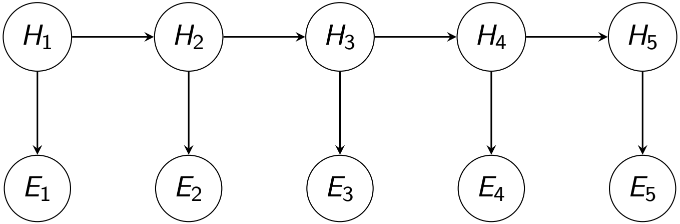
Output

$$\mathbb{P}(Q = q | E = e) \text{ for all values of } q$$

For example: If coughing but no itchy eyes, do you have a cold?

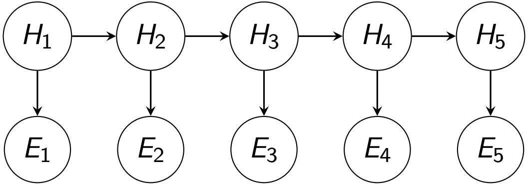
$$\mathbb{P}(C | H = 1, I = 0)$$

Object tracking: Hidden Markov Model



- An object starts at H_1 uniformly drawn over all possible locations.
- Then it **transitions** to an adjacent location. E.g., if $H_2 = 3$, then $H_3 \in 2, 4$ with equal probability.
- At each time step, we obtain a sensor reading E_i given H_i

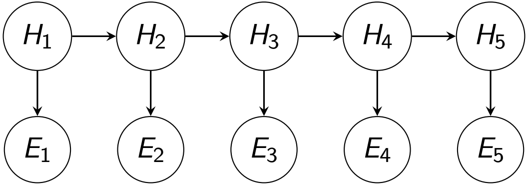
Object tracking: Hidden Markov Model



$H_i \in \{1, \cdots, K\}$ location of object at time step i
$E_i \in \{1, \cdots, K\}$ location of object at time step i
Start $p(h_1)$: e.g., uniform over all locations
Transition $p(h_i|h_{i1})$: e.g., uniform over adjacent loc.
Emission $p(e_i|h_i)$: e.g., uniform over adjacent loc.

$$\mathbb{P}(H = h, E = e) = \underbrace{p(h_1)}_{start} \prod_{i=2}^{n} \underbrace{p(h_i|h_{i-1})}_{transition} \prod_{i=2}^{n} \underbrace{p(e_i|h_i)}_{emission}$$

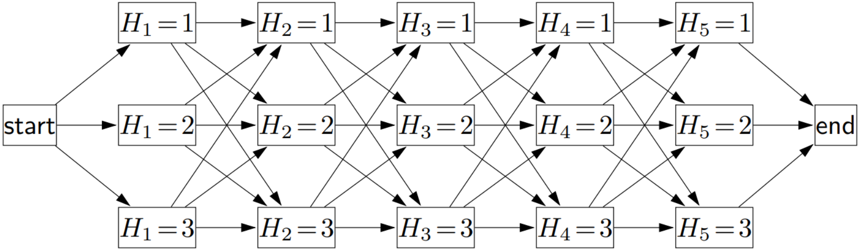
Object tracking: Hidden Markov Model



Question (smoothing)
$P(H_3|E_1 = e_1, E_2 = e_2, E_3 = e_3, E_4 = e_4, E_5 = e_5)$

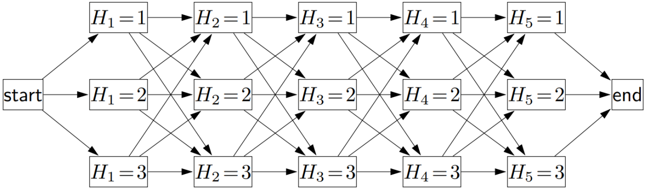
Distribution of some hidden variable H_i conditioned on all the evidence, including the future.

Lattice Representation



Edge *start* $\Rightarrow H_1 = 1$ has a probability $p(h_1)p(e_1|h_1)$
Edge $H_{i-1} = h_{i-1} \Rightarrow H_i = h_i$ has weight $p(h_i|h_{i-1})p(e_i|h_i)$
Each path from *start* to *end* is an assignment with weights equal to the product of edge weights

Lattice Representation



Forward: $F_i(h_i) = \sum_{h_{i-1}} F_{i-1}(h_{i-1}) w(h_{i-1}, h_i$
Sum of weights of path from *start* to $H_i = h_i$
Backward: $B_i(h_i) = \sum_{h_{i+1}} B_{i+1}(h_{i+1}) w(h_i, h_{i+1})$
Sum of weights of path from $H_i = h_i$ to *end*
Define: $S_i = F_i(h_i) B_i(h_i)$
Sum of weights of path from *start* to *end* through $H_i = h_i$

Forward-Backward

Smoothing queries:

$$\mathbb{P}(H_i = h_i | E_i = e_i) \propto S_i(h_i)$$

Forward-Backward Algorithm:

$$\begin{array}{l} \text{Compute } F_1, F_2, \cdots, F_n \\ \text{Compute } B_n, B_{n-1}, \cdots, B_1 \\ \text{Compute } S_i \text{ for each } i \text{ and normalize} \end{array}$$

Running time: $O(nK^2)$

Gibbs Sampling

Setup:

$$Weight(x)$$

```
Initialize x to a random complete assignment
Loop through i = 1, ..., n until convergence:
    Compute weight of x ∪ {X_i : v} for each v
    Choose x ∪ {X_i : v} with probability prop. to weight
```

Gibbs Sampling

Setup:

$$\mathbb{P}(X = x) \propto \mathit{Weight}(x)$$

```
Initialize x to a random complete assignment
Loop through i = 1, ..., n until convergence:
    Set X_i = v with prob. P(X_i = v | X_{-i} = x_{-i})
```

Note: X_{-i} denotes all variables except X_i

Image Denoising

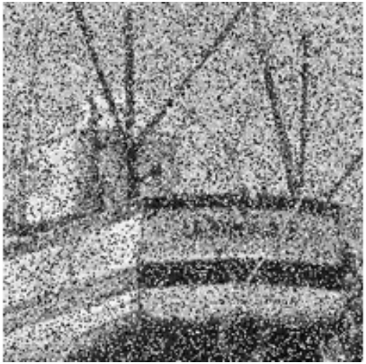
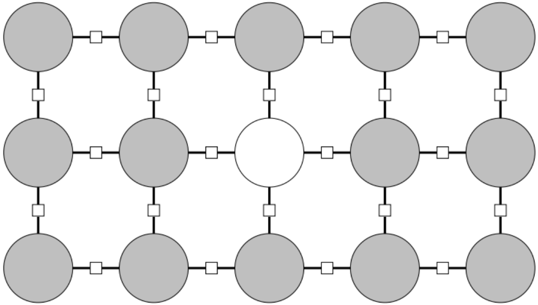
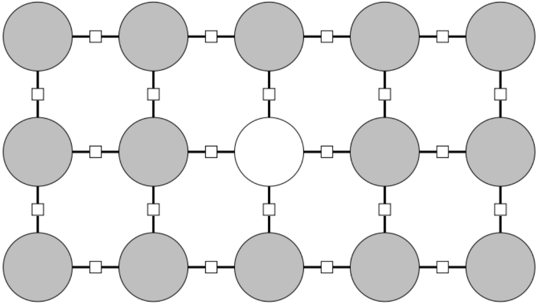


Image Denoising



- $X_i \in \{0, 1\}$ 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 $t_{ij}(x_i, x_j) = [x_i = x_j] + 1$

Image Denoising



If neighbors are $1, 1, 1, 0$ and X_i not observed:
$\mathbb{P}(X_i = 1 | X_i = x_i) = \frac{2221}{2221+1112} = 0.8$
If neighbors are $0, 1, 0, 1$ and X_i not observed:
$\mathbb{P}(X_i = 1 | X_i = x_i) = \frac{1212}{1212+2121} = 0.5$

Summary so far

Model (Bayesian network or factor graph):

$$\mathbb{P}(X = x) = \prod_{i=1}^{n} p(x_i | x_{parents(i)})$$

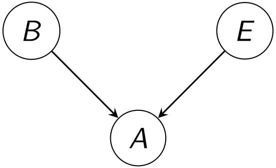
Probabilistic inference:

$$P(Q | E = e)$$

Algorithms:

- Forward-backward: HMMs, exact
- Gibbs sampling: general, approximate

How to set the parameters?



b	p(B=b)
1	ϵ
0	$1-\epsilon$

e	p(E=e)
1	ϵ
0	$1-\epsilon$

b	e	a	$p(a\|b,e)$
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

Supervised Learning

Input (Training data):
$\mathcal{D}_{train}$ (an example is an assignment to X)

Output (Parameters):
θ (local conditional probabilities)

Example: One variable

Setup:

One variable R representing the rating of a movie 1, 2, 3, 4, 5

$R \qquad \mathbb{P}(R = r) = p(r)$

Parameters:

$$\theta = (p(1), p(2), p(3), p(4), p(5))$$

Training data:

$$\mathcal{D}_{train} = \{1, 3, 4, 4, 4, 4, 4, 5, 5, 5\}$$

Example: One variable

Learning:

$$\mathcal{D}_{train} \implies \theta$$

Intuition: $p(r) \propto$ number of occurrences of r in $\mathcal{D}_{train}$

Example:

$$\mathcal{D}_{train} = \{1, 3, 4, 4, 4, 4, 4, 5, 5, 5\}$$

$$\Downarrow$$

θ:

r	$count(r)$
1	1
2	0
3	1
4	5
5	3

Example: One variable

Learning:

$$\mathcal{D}_{train} \implies \theta$$

Intuition: $p(r) \propto$ number of occurrences of r in $\mathcal{D}_{train}$

Example:

$$\mathcal{D}_{train} = \{1, 3{,}4, 4, 4, 4, 4, 5, 5, 5\}$$

$$\Downarrow$$

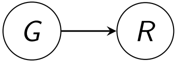
θ:

r	$p(r)$
1	0.1
2	0.0
3	0.1
4	0.5
5	0.3

Example: two variables

Variables:

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



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

$\mathcal{D}_{train} = \{(d, 4), (d, 4), (d, 5), (c, 1), (c, 5)\}$
Parameters: $= (p_G, p_R)$

Example: two variables

$$G \rightarrow R \qquad \mathbb{P}(G = g | R = r) = p_G(g) p_R(r|g)$$

$\mathcal{D}_{train} = \{(d, 4), (d, 4), (d, 5), (c, 1), (c, 5)\}$
Intuitive strategy: Estimate each local conditional distribution (p_G and p_R) separately

θ:

g	$count_G(g)$
d	3
c	2

Example: two variables

$$G \longrightarrow R \qquad \mathbb{P}(G = g | R = r) = p_G(g) p_R(r|g)$$

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

Intuitive strategy: Estimate each local conditional distribution (p_G and p_R) separately

θ:

g	$p_G(g)$
d	$3/5$
c	$2/5$

g	r	$count_R(g, r)$
d	4	2
d	5	1
c	1	1
c	5	1

Example: two variables

$$G \longrightarrow R \qquad \mathbb{P}(G = g | R = r) = p_G(g) p_R(r|g)$$

$\mathcal{D}_{train} = \{(d, 4), (d, 4), (d, 5), (c, 1), (c, 5)\}$
Intuitive strategy: Estimate each local conditional distribution (p_G and p_R) separately

θ:

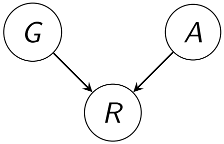
g	$p_G(g)$
d	$3/5$
c	$2/5$

g	r	$p_R(r\|g)$
d	4	$2/3$
d	5	$1/3$
c	1	$1/2$
c	5	$1/2$

Example: v-structure

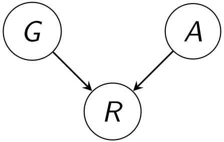
Variables:

- Genre $G \in$ drama,comedy
- Won award $A \in$ 0,1
- Rating $R \in$ 1,2,3,4,5



$$\mathbb{P}(G = g, A = a, R = r) = p_G(g)pA(a)p_R(r|g, a)$$

Example: v-structure

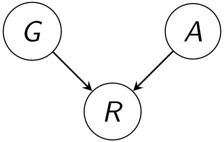


$\mathcal{D}_{train} = \{(d, 0, 3), (d, 1, 5), (d, 0, 1), (c, 0, 5), (c, 1, 4)\}$
Parameters: (p_G, p_A, p_R)

θ:

g	$count_G(g)$
d	3
c	2

Example: v-structure



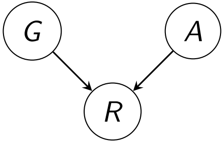
$\mathcal{D}_{train} = \{(d, 0, 3), (d, 1, 5), (d, 0, 1), (c, 0, 5), (c, 1, 4)\}$
Parameters: (p_G, p_A, p_R)

θ:

g	$p_G(g)$
d	$3/5$
c	$2/5$

a	$count_A(a)$
0	3
1	2

Example: v-structure



$\mathcal{D}_{train} = \{(d, 0, 3), (d, 1, 5), (d, 0, 1), (c, 0, 5), (c, 1, 4)\}$
Parameters: (p_G, p_A, p_R)

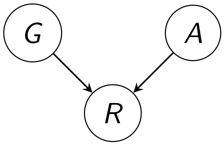
θ:

g	$p_G(g)$
d	$^3/_5$
c	$^2/_5$

a	$p_A(a)$
0	$^3/_5$
1	$^2/_5$

g	a	r	$count_R(g, a, r)$
d	0	1	1
d	0	3	1
d	1	5	1
c	0	5	1
c	0	4	1

Example: v-structure



$\mathcal{D}_{train} = \{(d, 0, 3), (d, 1, 5), (d, 0, 1), (c, 0, 5), (c, 1, 4)\}$
Parameters: (p_G, p_A, p_R)

θ:

g	$p_G(g)$
d	$3/5$
c	$2/5$

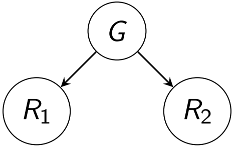
a	$p_A(a)$
0	$3/5$
1	$2/5$

g	a	r	$p_R(r \mid g, a)$
d	0	1	$1/2$
d	0	3	$1/2$
d	1	5	1
c	0	5	1
c	0	4	1

Example: inverted v-structure

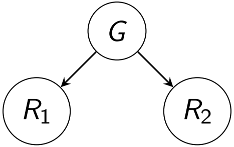
Variables:

- Genre $G \in$ drama,comedy
- John's rating $R_1 \in 1, 2, 3, 4, 5$
- Jane's rating $R_2 \in 1, 2, 3, 4, 5$



$$\mathbb{P}(G = g, R_1 = r_1, R_2 = r_2) = p_G(g) p_{R_1}(r_1|g) p_{R_2}(r_2|g)$$

Example: inverted v-structure

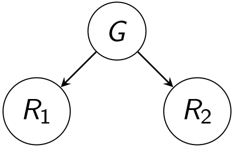


$\mathcal{D}_{train} = \{(d, 4, 5), (d, 4, 4), (d, 5, 3), (c, 1, 2), (c, 5, 4)\}$
Parameters: (p_G, p_{R_1}, p_{R_2})

θ:

g	$count_G(g)$
d	3
c	2

Example: inverted v-structure



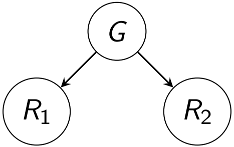
$\mathcal{D}_{train} = \{(d, 4, 5), (d, 4, 4), (d, 5, 3), (c, 1, 2), (c, 5, 4)\}$
Parameters: (p_G, p_{R_1}, p_{R_2})

θ:

g	$p_G(g)$
d	$^3/_5$
c	$^2/_5$

g	r_1	$count_{R_1}(g, r_1)$
d	4	2
d	5	1
c	1	1
c	4	1

Example: inverted v-structure



$\mathcal{D}_{train} = \{(d, 4, 5), (d, 4, 4), (d, 5, 3), (c, 1, 2), (c, 5, 4)\}$
Parameters: (p_G, p_{R_1}, p_{R_2})

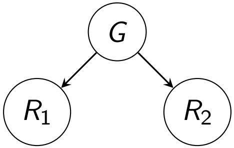
θ:

g	$p_G(g)$
d	$3/5$
c	$2/5$

g	r_1	$p_{R_1}(r_1 \mid g)$
d	4	$2/3$
d	5	$1/3$
c	1	$1/2$
c	4	$1/2$

g	r_2	$count_{R_2}(g, r_2)$
d	3	1
d	4	1
d	5	1
c	2	1
c	4	1

Example: inverted v-structure



$\mathcal{D}_{train} = \{(d, 4, 5), (d, 4, 4), (d, 5, 3), (c, 1, 2), (c, 5, 4)\}$
Parameters: (p_G, p_{R_1}, p_{R_2})

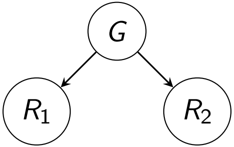
θ:

g	$p_G(g)$
d	$^3/_5$
c	$^2/_5$

g	r_1	$p_{R_1}(r_1 \mid g)$
d	4	$^2/_3$
d	5	$^1/_3$
c	1	$^1/_2$
c	4	$^1/_2$

g	r_2	$p_{R_2}(r_2 \mid g)$
d	3	$^1/_3$
d	4	$^1/_3$
d	5	$^1/_3$
c	2	$^1/_2$
c	4	$^1/_2$

Example: inverted v-structure

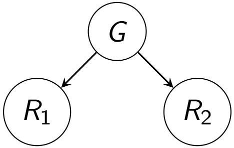


$\mathcal{D}_{train} = \{(d, 4, 5), (d, 4, 4), (d, 5, 3), (c, 1, 2), (c, 5, 4)\}$
Parameters: (p_G, p_R)

θ:

g	$count_G(g)$
d	3
c	2

Example: inverted v-structure



$\mathcal{D}_{train} = \{(d,4,5),(d,4,4),(d,5,3),(c,1,2),(c,5,4)\}$

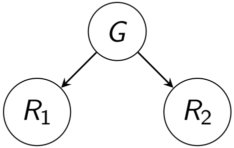
Parameters: (p_G, p_R)

θ:

g	$p_G(g)$
d	$^3/_5$
c	$^2/_5$

g	r	$count_R(g,r)$
d	3	1
d	4	3
d	5	2
c	1	1
c	2	1
c	4	1
c	5	1

Example: inverted v-structure



$\mathcal{D}_{train} = \{(d,4,5),(d,4,4),(d,5,3),(c,1,2),(c,5,4)\}$

Parameters: (p_G, p_R)

θ:

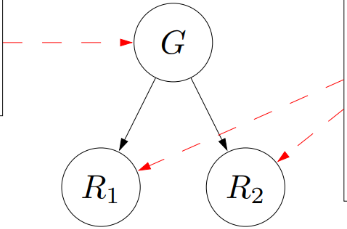
g	$p_G(g)$
d	$^3/_5$
c	$^2/_5$

g	r	$p_R(r \mid g)$
d	3	$^1/_6$
d	4	$^3/_6$
d	5	$^2/_6$
c	1	$^1/_4$
c	2	$^1/_4$
c	4	$^1/_4$
c	5	$^1/_4$

Parameter sharing

Key idea: The local conditional distributions of different variables use the same parameters.

g	$p_G(g)$
c	2/5
d	3/5

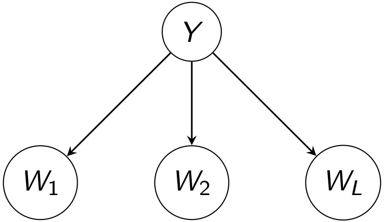


g	r	$p_R(r \mid g)$
d	3	1/6
d	4	3/6
d	5	2/6
c	1	1/4
c	2	1/4
c	4	1/4
c	5	1/4

Impact: more reliable estimates, less expressive model

Example: Naive Bayes

Variables: • Genre $Y \in \{comedy, drama\}$
• Movie review (sequence of words): $W_1, \cdots, W_L$



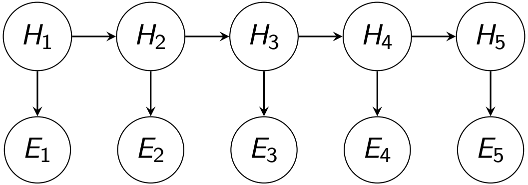
$$\mathbb{P}(Y = y, W_1 = w_1, \cdots, W_L = w_L) = p_{genre}(y) \prod_{j=1}^{L} p_{word}(w_j | y)$$

Parameters: $\theta = (p_{genre}, p_{word})$

Example: HMMs

Variables:

- $H_1, \cdots, H_n$ (e.g., actual positions)
- $E_1, \cdots, E_n$ (e.g., sensor readings)



$$\mathbb{P}(H = h, E = e) = p_{start}(h_1) \prod_{i=2}^{n} p_{trans}(h_i | h_{i1}) \prod_{i=1}^{n} p_{emit}(e_i | h_i)$$

Parameters: $= (p_{start}, p_{trans}, p_{emit})$

General case

Bayesian network: variables $X_1, \cdots, X_n$
Parameters: collection of distributions $\theta = \{p_d : d \in D\}(e.g., D = \{start, trans, emit\})$
Each variable X_i is generated from distribution p_{d_i}:

$$\mathbb{P}(X_1 = x_1, \cdots, X_n = x_n) = \prod_{i=1}^{n} p_{d_i}(x_i | x_{parents}(i))$$

Parameter sharing: d_i could be same for multiple i

General case: Learning Algorithm

Input: training examples $\mathcal{D}_{train}$ of full assignments
Output: parameters $\theta = p_d : d \in \mathcal{D}$
Algorithm: maximum likelihood for Bayesian networks

```
Count:
    For each x ∈ D_train:
        For each variable x_i:
            Increment count_{d_i}(x_{parents(i)}, x_i)
Normalize:
    For each d and local assignment x_{parents(i)}:
        Set p_d(x_i | x_{parents(i)}) ∝ count_d(x_{parents(i)}, x_i)
```

Maximum likelihood

Objective:

$$\max_{\theta} \prod_{x \in \mathcal{D}_{train}} \mathbb{P}(X = x; \theta)$$

Algorithm on previous slide exactly computes maximum likelihood parameters (closed form solution).

Maximum likelihood

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

$$\max_{\theta} \prod_{x \in \mathcal{D}_{train}} \mathbb{P}(X = x; \theta)$$

Maximum likelihood

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

$$\begin{aligned} &\max_{\theta} \prod_{x \in \mathcal{D}_{train}} \mathbb{P}(X = x; \theta) \\ = &\max_{p_G(\cdot), p_R(\cdot|c), p_R(\cdot|d)} \Big(p_G(d) p_R(4|d) p_G(d) p_R(5|d) p_G(c) p_R(5|c) \Big) \end{aligned}$$

Maximum likelihood

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

$$\begin{aligned}
&\max_{\theta} \prod_{x \in \mathcal{D}_{train}} \mathbb{P}(X = x; \theta) \\
&= \max_{p_G(\cdot), p_R(\cdot|c), p_R(\cdot|d)} \Big(p_G(d) p_R(4|d) p_G(d) p_R(5|d) p_G(c) p_R(5|c) \Big) \\
&= \max_{p_G(\cdot)} \Big(p_G(d) p_G(d) p_G(c) \Big) \max_{p_R(\cdot|c)} \Big(p_R(5|c) \Big) \max_{p_R(\cdot|d)} \Big(p_R(4|d) p_R(5|d) \Big)
\end{aligned}$$

Maximum likelihood

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

$$
\begin{aligned}
&\max_{\theta} \prod_{x \in \mathcal{D}_{train}} \mathbb{P}(X = x; \theta) \\
= &\max_{p_G(\cdot), p_R(\cdot|c), p_R(\cdot|d)} \Big(p_G(d) p_R(4|d) p_G(d) p_R(5|d) p_G(c) p_R(5|c) \Big) \\
= &\max_{p_G(\cdot)} \Big(p_G(d) p_G(d) p_G(c) \Big) \max_{p_R(\cdot|c)} \Big(p_R(5|c) \Big) \max_{p_R(\cdot|d)} \Big(p_R(4|d) p_R(5|d) \Big)
\end{aligned}
$$

Solution:

$$p_G(d) = \frac{2}{3}, p_G(c) = \frac{1}{3}, p_R(5|c) = 1, p_R(4|d) = \frac{1}{2}, p_R(5|d) = \frac{1}{2}$$

Maximum likelihood

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

$$\max_{p_G(\cdot), p_R(\cdot|c), p_R(\cdot|d)} \left(p_G(d) p_R(4|d) p_G(d) p_R(5|d) p_G(c) p_R(5|c) \right)$$

$$\max_{p_G(\cdot)} \left(pG(d) pG(d) pG(c) \right) \max_{p_R(\cdot|c)} p_R(5|c) \max_{p_R(\cdot|d)} \left(p_R(4|d) p_R(5|d) \right)$$

Solution:

$$p_G(d) = \frac{2}{3}, p_G(c) = \frac{1}{3}, p_R(5|c) = 1, p_R(4|d) = \frac{1}{2}, p_R(5|d) = \frac{1}{2}$$

- Key: decomposes into subproblems, one for each distribution d and assignment $x_{parents}$
- For each subproblem, solve in closed form (Lagrange multipliers for sum-to-1 constraint)

Scenario 1

Setup:

- You have a coin with an unknown probability of heads $p(H)$.
- You flip it 100 times, resulting in 23 heads, 77 tails.
- What is estimate of $p(H)$?

Scenario 1

Setup:

- You have a coin with an unknown probability of heads $p(H)$.
- You flip it 100 times, resulting in 23 heads, 77 tails.
- What is estimate of $p(H)$?

Maximum likelihood estimate:

$$p(H) = 0.23 \quad p(T) = 0.77$$

Scenario 2

Setup:

- You flip a coin once and get heads.
- What is estimate of $p(H)$?

Maximum likelihood estimate:

$$p(H) = 1 \quad p(T) = 0$$

Intuition: This is a bad estimate; real $p(H)$ should be closer to half
When have less data, maximum likelihood overfits, want a more reasonable estimate.

Regularization: Laplace Smoothing

Maximum likelihood:

$$p(H) = 1 \quad p(T) = 0$$

Maximum likelihood with Laplace smoothing:

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

Example: two variable

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

Amount of smoothing: $\lambda = 1$

θ:

g	$count_G(g)$
d	1
c	1

Example: two variable

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

Amount of smoothing: $\lambda = 1$

θ:

g	$count_G(g)$
d	$1 + 2$
c	$1 + 1$

Example: two variable

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

Amount of smoothing: $\lambda = 1$

θ:

g	$p_G(g)$
d	$^3/_5$
c	$^2/_5$

g	r	$count_R(r\|g)$
d	1	1
d	2	1
d	3	1
d	4	1
d	5	1
c	1	1
c	2	1
c	3	1
c	4	1
c	5	1

Example: two variable

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

Amount of smoothing: $\lambda = 1$

θ:

g	$p_G(g)$
d	$^3/_5$
c	$^2/_5$

g	r	$p_R(r \mid g)$
d	1	$^1/_7$
d	2	$^1/_7$
d	3	$^1/_7$
d	4	$^2/_7$
d	5	$^2/_7$
c	1	$^1/_6$
c	2	$^1/_6$
c	3	$^1/_6$
c	4	$^1/_6$
c	5	$^2/_6$

Regularization: Laplace smoothing

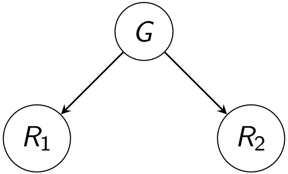
For each distribution d and partial assignment $(x_{parents(i)}, x_i)$, add λ to $count_d(x_{parents(i)}, xi)$.
Then normalize to get probability estimates.

Interpretation: hallucinate λ occurrences of each local assignment
Larger $\lambda \Rightarrow$ more smoothing $\Rightarrow$ probabilities closer to uniform.

Data wins out in the end:

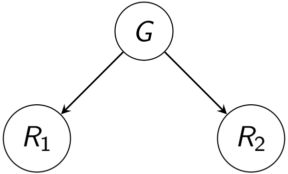
$$p(H) = \frac{1+1}{1+2} = \frac{2}{3} \quad P(H) = \frac{998+1}{998+2} = 0.999$$

Unsupervised Learning



What if we don't observe some of the variables?
$\mathcal{D}_{train} = \{(?, 4, 5), (?, 4, 4), (?, 5, 3), (?, 1, 2), (?, 5, 4)\}$

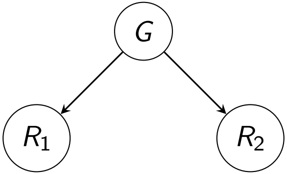
Unsupervised Learning



What if we don't observe some of the variables?
$\mathcal{D}_{train} = \{(?, 4, 5), (?, 4, 4), (?, 5, 3), (?, 1, 2), (?, 5, 4)\}$ Two appraoches:
1) Try to work with the count & normalize routine and come up with maximum likelihood values
2) Try to guess what the missing values are

Maximum marginal likelihood

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

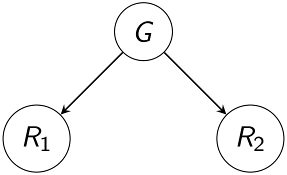


$H = G$ $E = (R_1, R_2)$ $e = (1, 2)$
$\theta = (p_G, p_R)$

Maximum marginal likelihood

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

Example:



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

$\theta = (p_G, p_R)$

Maximum marginal likelihood objective:

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

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

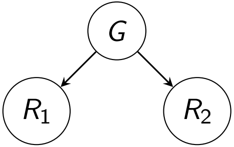
Expectation Maximization

Intuition: generalization of the K-means algorithm
Variables: H is hidden, $E = e$ is observed

Algorithm: Expectation Maximization (EM)

```
Initialize θ
E-step:
    Compute q(h) = P(H = h|E = e; θ) for each h
    Create weighted points: (h, e) with weight q(h)
M-step:
    Compute maximum likelihood (just count and normalize) to
get θ
Repeat until convergence.
```

Example: one iteration of EM



$$\mathcal{D}_{train} = \{(?, 2, 2), (?, 1, 2)\}$$

θ:

g	$p_G(g)$
c	0.5
d	0.5

g	r	$p_R(r \mid g)$
c	1	0.4
c	2	0.6
d	1	0.6
d	2	0.4

E-step

(r_1, r_2)	g	$\mathbb{P}(G = g, R_1 = r_1, R_2 = r_2)$	$q(g)$
(2, 2)	c	$0.5 \cdot 0.6 \cdot 0.6 = 0.18$	$\frac{0.18}{0.18+0.08} = 0.69$
(2, 2)	d	$0.5 \cdot 0.4 \cdot 0.4 = 0.08$	$\frac{0.08}{0.18+0.08} = 0.31$
(1, 2)	c	$0.5 \cdot 0.4 \cdot 0.6 = 0.12$	$\frac{0.12}{0.12+0.12} = 0.5$
(1, 2)	d	$0.5 \cdot 0.6 \cdot 0.4 = 0.12$	$\frac{0.12}{0.12+0.12} = 0.5$

M-step

θ:

g	count	$p_G(g)$
c	0.69 + 0.5	0.59
d	0.31 + 0.5	0.41

g	r	count	$p_R(r \mid g)$
c	1	0.5	0.21
c	2	0.5 + 0.69 + 0.69	0.79
d	1	0.5	0.31
d	2	0.5 + 0.31 + 0.31	0.69

Application: Decryption

Substitution cipher:

Plain	abcdefghijklmnopqrstuvwxyz
Encryption Key	plokmijnuhbygvtfcrdxeszaqw

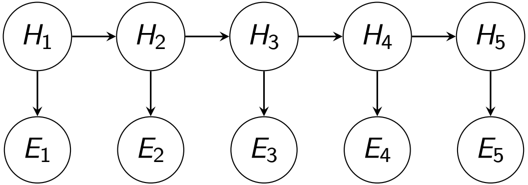
Table: Substitution Table (unknown)

Plain Text (unknown): hello world
Encrypted Text (known): nmyyt ztryk

Example: Decryption as an HMM

Variables:

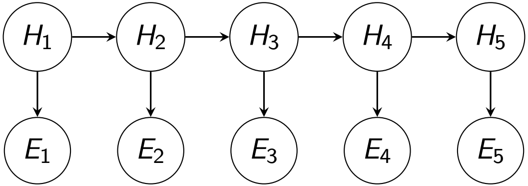
- $H_1, \cdots, H_n$ (e.g., characters of plain text)
- $E_1, \cdots, E_n$ (e.g., characters of encrypted text)



$$\mathbb{P}(H = h, E = e) = p_{start}(h_1) \prod_{i=2}^{n} p_{trans}(h_i | h_{i1}) \prod_{i=1}^{n} p_{emit}(e_i | h_i)$$

Parameters: $= (p_{start}, p_{trans}, p_{emit})$

Example: Decryption as an HMM

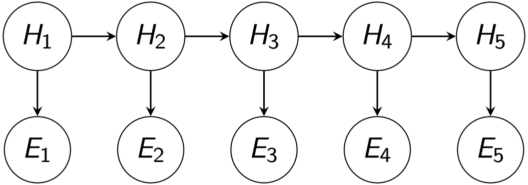


Strategy:

- p_{start}: set to uniform
- p_{trans}: estimate on tons of English text
- p_{emit}: substitution table, from EM

Intuition: rely on language model (p_{trans}) to favor plain texts h that look like English

Example: Decryption as an HMM



E-step: forward-backward algorithm computes

$$q_i(h) \stackrel{\mathrm{def}}{=\!=} \mathbb{P}(H_i = h | E_1 = e_1, \cdots E_n = e_n)$$

M-step: count (fractional) and normalize

$$count_{emit}(h, e) = P_n \sum_{i=1}^{n} q_i(h) \cdot [e_i = e]$$

$$p_{emit}(e|h) \propto count_{emit}(h, e)$$

Recap

Smoothing
Gibbs Sampling
Probability from data
Maximum Likelihood
Laplace Smoothing
Expectation Maximization

References

Stuart Russell and Xiaodong Song (2021)
CS 188 — Introduction to Artificial Intelligence
University of California, Berkeley

Chelsea Finn and Nima Anari (2021)
CS221 — Artificial Intelligence: Principles and Techniques
Stanford University

The End