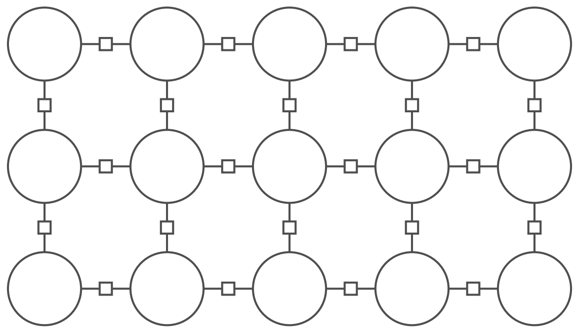
Bayesian Networks III



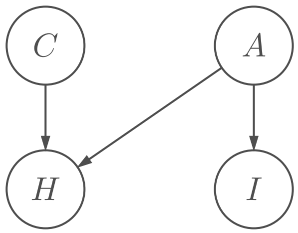
Lecture: Bayesian networks

Learning: Supervised learning

Learning: Smoothing

Learning: EM Algorithm

Review: Bayesian network



Random variables:

cold C, allergies A, cough H, itchy eyes I

Joint distribution:

$$\mathbb{P}(C = c, A = a, H = h, I = i) = p(c)p(a)p(h \mid c, a)p(i \mid a)$$

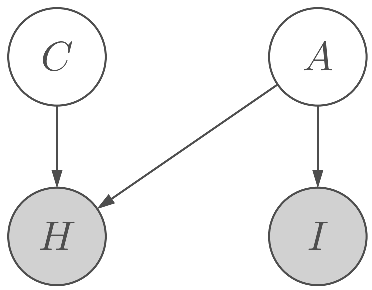
Definition: Bayesian network

Let $X = (X_1, \dots, X_n)$ 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:

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

Review: probabilistic inference



Question: $\mathbb{P}(C \mid H = 1, I = 1)$

Input

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

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

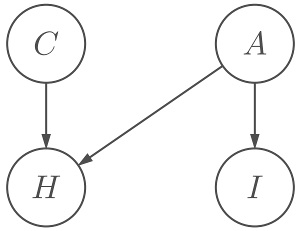
Query: $Q \subseteq X$ is subset of variables

Output

$\mathbb{P}(Q \mid E = e) \longleftrightarrow \mathbb{P}(Q = q \mid E = e)$ for all values q

Algorithms: Gibbs sampling, forward-backward, particle filtering

Where do parameters come from?



c	$p(c)$
1	?
0	?

a	$p(a)$
1	?
0	?

c	a	h	$p(h \mid c, a)$
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

a	i	$p(i \mid a)$
0	0	?
0	1	?
1	0	?
1	1	?

Learning task

Training data

$\mathcal{D}_{\text{train}}$ (an example is an assignment to X)

$\downarrow$

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:

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

Example: one variable

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

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

$$\downarrow$$

θ:

r	count(r)	$p(r)$
1	1	0.1
2	0	0.0
3	1	0.1
4	5	0.5
5	3	0.3

Example: two variables

Variables:

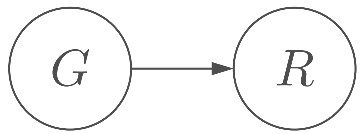
- Genre $G \in \{\text{drama}, \text{comedy}\}$
- Rating $R \in \{1, 2, 3, 4, 5\}$

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

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

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

Example: two variables



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

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

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

θ:

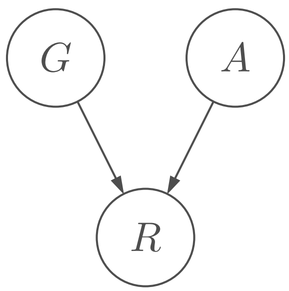
g	count$_G(g)$	$p_G(g)$
d	3	3/5
c	2	2/5

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

Example: v-structure

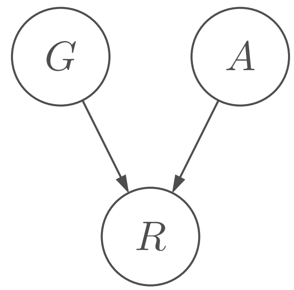
Variables:

- $G \in \{\text{drama}, \text{comedy}\}$ (genre)
- $A \in \{0, 1\}$ (award)
- $R \in \{1, 2, 3, 4, 5\}$ (rating)



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

Example: v-structure



$\mathcal{D}_{\textsf{train}} = \{(\textsf{d}, 0, 3), (\textsf{d}, 1, 5), (\textsf{d}, 0, 1), (\textsf{c}, 0, 5), (\textsf{c}, 1, 4)\}$

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

θ:

g	$\textsf{count}_G(g)$	$p_G(g)$
d	3	3/5
c	2	2/5

a	$\textsf{count}_A(a)$	$p_A(a)$
0	3	3/5
1	2	2/5

g	a	r	$\textsf{count}_R(g, a, r)$	$p_R(r \mid g, a)$
d	0	1	1	1/2
d	0	3	1	1/2
d	1	5	1	1
c	0	5	1	1
c	1	4	1	1

Example: inverted-v structure

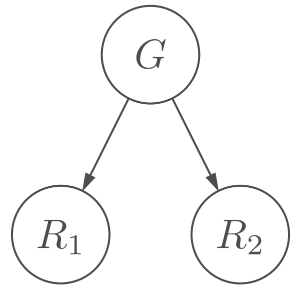
Variables:

- Genre $G \in \{\text{drama}, \text{comedy}\}$
- Jim's rating $R_1 \in \{1, 2, 3, 4, 5\}$
- Martha's rating $R_2 \in \{1, 2, 3, 4, 5\}$

$$
\begin{tikzcd}
 & G \arrow[dl] \arrow[dr] & \\
R_1 & & R_2
\end{tikzcd}
$$

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

Example: inverted-v structure



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

Parameters: $\theta = (p_G, p_{R_1}, p_{R_2})$

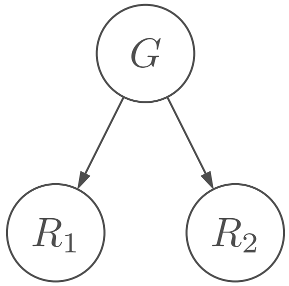
θ:

g	$\text{count}_G(g)$	$p_G(g)$
d	3	3/5
c	2	2/5

g	r_1	$\text{count}_{R_1}(g, r)$	$p_{R_1}(r \mid g)$
d	4	2	2/3
d	5	1	1/3
c	1	1	1/2
c	5	1	1/2

g	r_2	$\text{count}_{R_2}(g, r)$	$p_{R_2}(r \mid g)$
d	3	1	1/3
d	4	1	1/3
d	5	1	1/3
c	2	1	1/2
c	4	1	1/2

Example: inverted-v structure



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

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

θ:

g	$\text{count}_G(g)$	$p_G(g)$
d	3	3/5
c	2	2/5

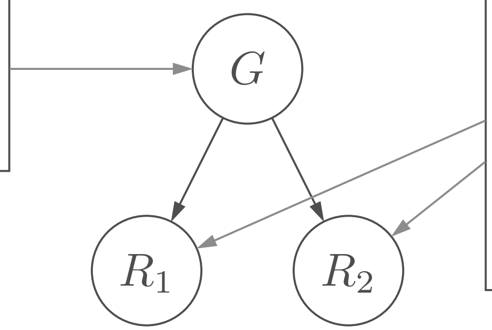
g	r	$\text{count}_R(g, r)$	$p_R(r \mid g)$
d	3	1	1/6
d	4	3	3/6
d	5	2	2/6
c	1	1	1/4
c	2	1	1/4
c	4	1	1/4
c	5	1	1/4

Parameter sharing

Key idea: parameter sharing

The local conditional distributions of different variables can share the same parameters.

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



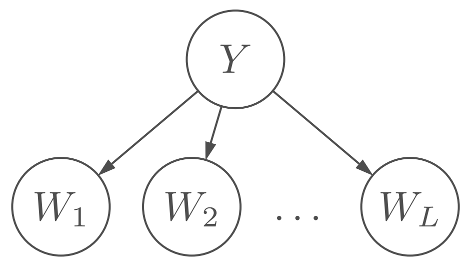
g	r	$p_R(r \mid g)$
d	3	1/6
d	4	3/6
d	5	2/6
c	1	1/4
c	2	1/4
c	4	1/4
c	5	1/4

Impact: more reliable estimates, less expressive model

Example: Naive Bayes

Variables:

- Genre $Y \in \{\text{comedy}, \text{drama}\}$
- Movie review (sequence of words): $W_1, \dots, W_L$



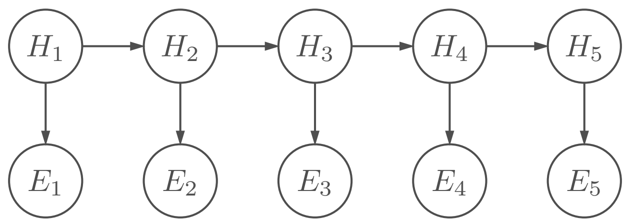
$$\mathbb{P}(Y = y, W_1 = w_1, \dots, W_L = w_L) = p_{\text{genre}}(y) \prod_{j=1}^{L} p_{\text{word}}(w_j \mid y)$$

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

Example: HMMs

Variables:

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



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

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

$\mathcal{D}_{\text{train}}$ is a set of full assignments to (H, E)

General case

Bayesian network: variables $X_1, \dots, X_n$

Parameters: collection of distributions $\theta = \{p_d : d \in D\}$ (e.g., $D = \{\text{start}, \text{trans}, \text{emit}\}$)

Each variable X_i is generated from distribution p_{d_i}:

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

Parameter sharing: d_i could be same for multiple i

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

Maximum likelihood

Maximum likelihood objective:

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

Algorithm: maximum likelihood

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

Closed form — no iterative optimization!

Maximum likelihood

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

$$\max_\theta \prod_{x \in \mathcal{D}_{\text{train}}} \mathbb{P}(X = x; \theta) = \max_{p_G(\cdot), p_R(\cdot \mid \mathsf{c}), p_R(\cdot \mid \mathsf{d})} (p_G(\mathsf{d}) p_R(4 \mid \mathsf{d}) p_G(\mathsf{d}) p_R(5 \mid \mathsf{d}) p_G(\mathsf{c}) p_R(5 \mid \mathsf{c}))$$

$$= \max_{p_G(\cdot)} (p_G(\mathsf{d}) p_G(\mathsf{d}) p_G(\mathsf{c})) \max_{p_R(\cdot \mid \mathsf{c})} p_R(5 \mid \mathsf{c}) \max_{p_R(\cdot \mid \mathsf{d})} (p_R(4 \mid \mathsf{d}) p_R(5 \mid \mathsf{d}))$$

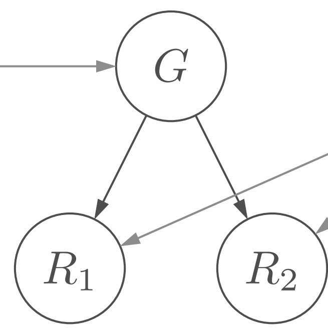
Solution:

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

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

Summary

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

- Parameter sharing: variables powered by parameters (passing by reference)
- Maximum likelihood = count and normalize

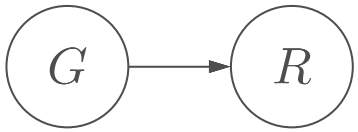
Lecture: Bayesian networks

Learning: Supervised learning

Learning: Smoothing

Learning: EM Algorithm

Review: maximum likelihood



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

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

θ:

g	$\text{count}_G(g)$	$p_G(g)$
d	3	3/5
c	2	2/5

g	r	$\text{count}_R(g, r)$	$p_R(r \mid g)$
d	4	2	2/3
d	5	1	1/3
c	1	1	1/2
c	5	1	1/2

Do we really believe that $p_R(r = 2 \mid g = \text{c}) = 0$?

Overfitting!

Laplace smoothing example

Idea: just add $\lambda = 1$ to each count

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

θ:

g	$\text{count}_G(g)$	$p_G(g)$
d	1+3	4/7
c	1+2	3/7

g	r	$\text{count}_R(g, r)$	$p_R(g, r)$
d	1	1	1/8
d	2	1	1/8
d	3	1	1/8
d	4	1+2	3/8
d	5	1+1	2/8
c	1	1+1	2/7
c	2	1	1/7
c	3	1	1/7
c	4	1	1/7
c	5	1+1	2/7

Now $p_R(r = 2 \mid g = \text{c}) = \frac{1}{7} > 0$

Laplace smoothing

Key idea: maximum likelihood with Laplace smoothing

For each distribution d and partial assignment $(x_{\text{Parents}(i)}, x_i)$:

Add λ to $\text{count}_d(x_{\text{Parents}(i)}, x_i)$.

Further increment counts $\{\text{count}_d\}$ based on $\mathcal{D}_{\text{train}}$.

Hallucinate λ occurrences of each local assignment

Interplay between smoothing and data

Larger $\lambda \Rightarrow$ more smoothing $\Rightarrow$ probabilities closer to uniform

g	$\text{count}_G(g)$	$p_G(g)$
d	1/2+1	3/4
c	1/2	1/4

g	$\text{count}_G(g)$	$p_G(g)$
d	1+1	2/3
c	1	1/3

Data wins out in the end (suppose only see $g =$ d):

g	$\text{count}_G(g)$	$p_G(g)$
d	1+1	2/3
c	1	1/3

g	$\text{count}_G(g)$	$p_G(g)$
d	1+998	0.999
c	1	0.001

Summary

g	$\text{count}_G(g)$	$p_G(g)$
d	$\lambda + 1$	$\frac{1+\lambda}{1+2\lambda}$
c	λ	$\frac{\lambda}{1+2\lambda}$

- Pull distribution closer to uniform distribution

- Smoothing gets washed out with more data

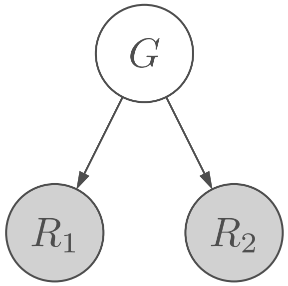
Lecture: Bayesian networks

Learning: Supervised learning

Learning: Smoothing

Learning: EM Algorithm

Motivation



Genre $G \in \{\text{drama}, \text{comedy}\}$
Jim's rating $R_1 \in \{1, 2, 3, 4, 5\}$
Martha's rating $R_2 \in \{1, 2, 3, 4, 5\}$

If observe all the variables: maximum likelihood $=$ count and normalize

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

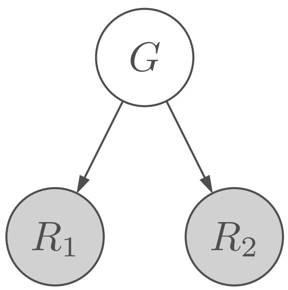
What if we **don't observe** some of the variables?

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

Maximum marginal likelihood

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:

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

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

Expectation Maximization (EM)

Intuition: generalization of the K-means algorithm

cluster centroids = parameters θ cluster assignments = hidden variables H

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

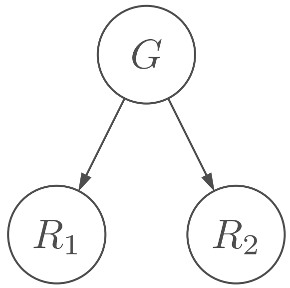
Algorithm: Expectation Maximization (EM)

Initialize θ randomly

Repeat until convergence:

- E-step:
  - Compute $q(h) = \mathbb{P}(H = h \mid E = e; \theta)$ for each h (probabilistic inference)
  - Create fully-observed weighted examples: (h, e) with weight $q(h)$
- M-step:
  - Maximum likelihood (count and normalize) on weighted examples to get θ

Example: one iteration of EM



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

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 $\longrightarrow$

(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	$0.5 \cdot 0.6 \cdot 0.4 = 0.12$	$\frac{0.12}{0.12+0.12} = 0.5$

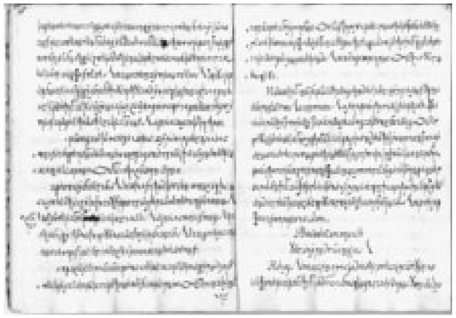
M-step $\longrightarrow$ θ:

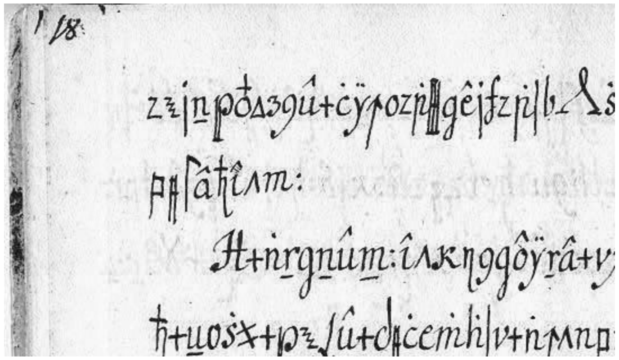
g	count	$p_G(g)$
c	0.69 + 0.5	0.59
d	0.31 + 0.5	0.41

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: decipherment

Copiale cipher (105-page encrypted volume from 1730s):





Cracked in 2011 with the help of EM!

Substitution ciphers

Letter substitution table (unknown):

Plain:	`abcdefghijklmnopqrstuvwxyz`
Cipher:	`plokmijnuhbygvtfcrdxeszaqw`

Plaintext (unknown): `hello world`

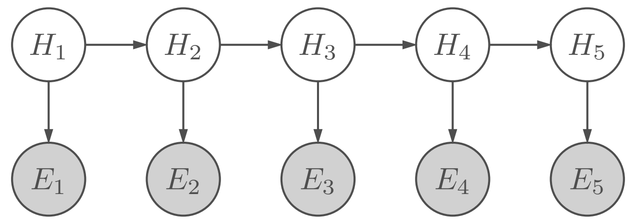
Ciphertext (known): `nmyyt ztryk`

Challenge: Give ciphertext, recover the plaintext

Application: decipherment as an HMM

Variables:

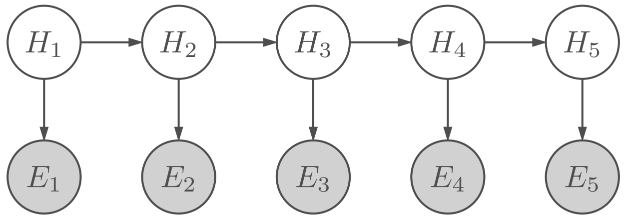
- $H_1, \dots, H_n$ (e.g., characters of plaintext)
- $E_1, \dots, E_n$ (e.g., characters of ciphertext)



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

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

Application: decipherment as an HMM



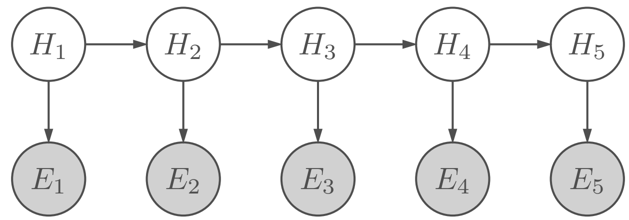
Strategy:

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

Intuitions:

- p_{trans} to favor plaintexts h that look like English
- p_{emit} favors consistent characters substitutions

Application: decipherment as an HMM



E-step: forward-backward computes for each position i and character h

$$q_i(h) \stackrel{\text{def}}{=} \mathbb{P}(H_i = h \mid E_1 = e_1, \dots E_n = e_n)$$

M-step: count (fractional) and normalize for all characters e, h

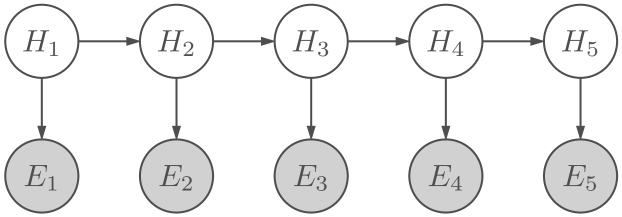
$$\text{count}_{\text{emit}}(h, e) = \sum_{i : e_i = e} q_i(h)$$

$$p_{\text{emit}}(e \mid h) \propto \text{count}_{\text{emit}}(h, e)$$

Decipherment in Python

[code]

Summary

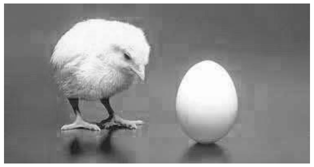


Maximum marginal likelihood:

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

EM algorithm:

$\Leftarrow$ probabilistic inference (E-step)

hidden variables $q(h)$  parameters θ

count and normalize (M-step) $\Rightarrow$

Applications: decipherment, phylogenetic reconstruction, crowdsourcing

Course plan

Search problems
Markov decision processes
Adversarial games

Constraint satisfaction problems
Markov networks
Bayesian networks

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

Low-level → High-level

Machine learning