Hidden Markov Models

- Elements
  - hidden state X
  - clock
    - at each tick of the clock, the state updates using
  - dynamical model
    - $$X_{i+1} \sim P(X_{i+1}|X_i = x_i)$$
  - emission at each state, depending on the state alone Y - this is observed

$$Y_i \sim P(Y_i|X_i = x_i)$$

Problems

- Estimating a model
  - given a set of data $Y_i=y_i$, what model produced the data?
- Inference
  - given a string and a model, what set of hidden states produced the data?
- Typically X is discrete

Examples

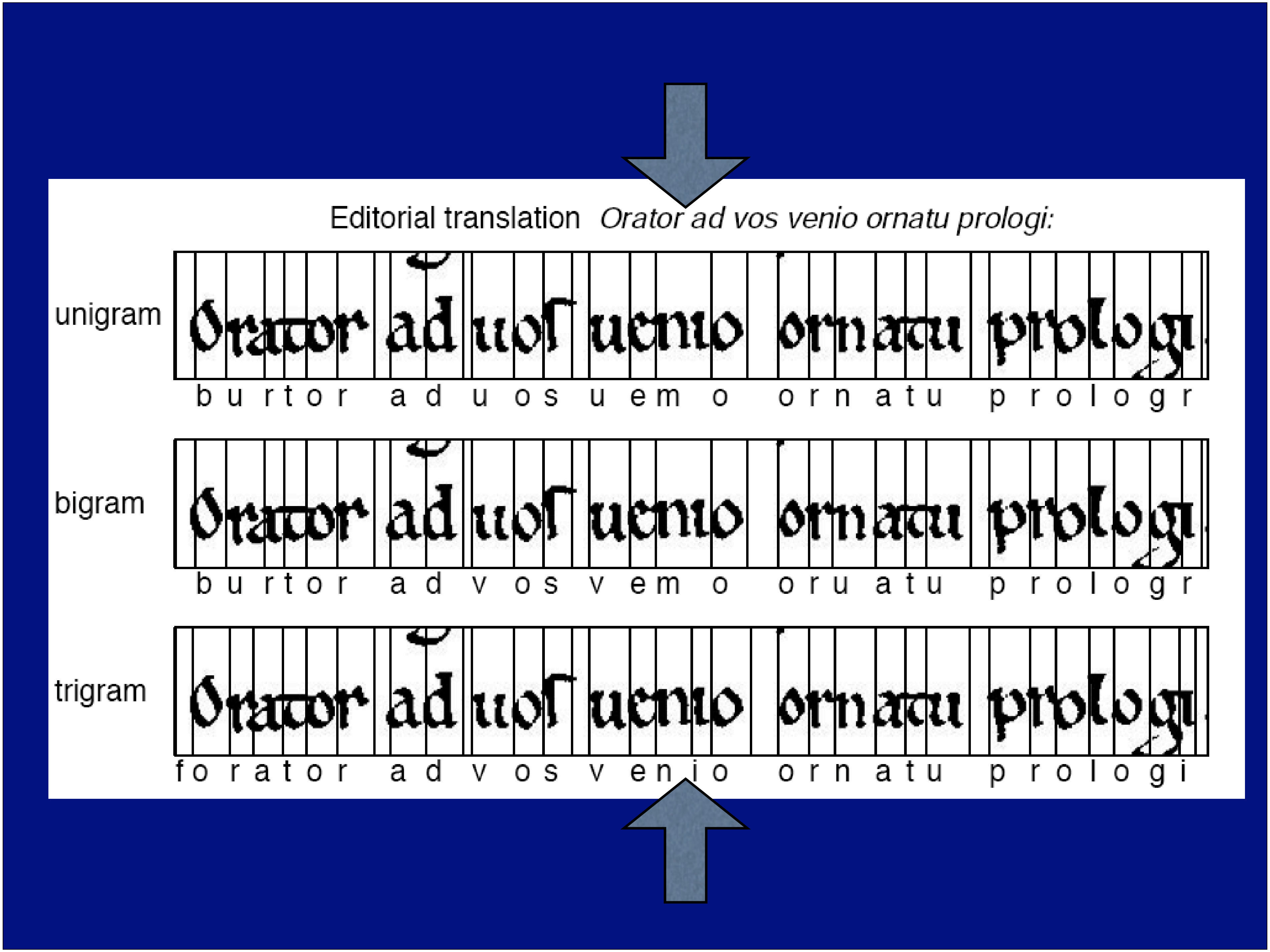
- We observe audio, and wish to infer words
  - much infrastructure required to link this problem to the model
- We observe ink, and wish to infer letters
  - Or any substitution cypher
- We observe people in video, and wish to infer activities

Pragmatics

- X is usually a discrete space
  - n-gram letter models
  - n-gram word models
- It usually has many elements
  - because if it doesn't, the model is not much help
  - but this makes the dynamical model hard to learn (too many transitions)
- Strategies
  - lots of zeros
  - find a model elsewhere

Example

- Search scribal handwriting for strings
  - observations are ink
  - clock obtained by segmentation
    - which can occur at the same time as inference
  - hidden states are letters
  - dynamical model learned by counting in transcribed text


Editorial translation *Orator ad vos venio ornatu prologi:*
unigram
b u r t o r a d u o s u e m o o r n a t u p r o l o g r
bigram
b u r t o r a d v o s v e m o o r u a t u p r o l o g r
trigram
f o r a t o r a d v o s v e n i o o r n a t u p r o l o g i

Estimating Transition Probabilities

- Maximum likelihood estimates are given by counts

$$P_{\mathrm{MLE}}(w_1, w_2, \ldots, w_n) = \frac{C(w_1, w_2, \ldots, w_n)}{N}$$

$$P_{\mathrm{MLE}}(w_1 | w_2, \ldots, w_n) = \frac{C(w_1, w_2, \ldots, w_n)}{C(w_2, \ldots, w_n)}$$

Counting and words

Word Frequency	Frequency of Frequency
1	3993
2	1292
3	664
4	410
5	243
6	199
7	172
8	131
9	82
10	91
11–50	540
51–100	99
> 100	102

Table 1.2 Frequency of frequencies of word types in *Tom Sawyer*.

From Manning and Schutze; recall there are 8, 018 word types

This means many counts will be zero

In person	*she*		*was*		*inferior*		*to*		*both*		*sisters*	
1-gram	$P(\cdot)$		$P(\cdot)$		$P(\cdot)$		$P(\cdot)$		$P(\cdot)$		$P(\cdot)$	
1	the	0.034	the	0.034	the	0.034	the	0.034	the	0.034	the	0.034
2	to	0.032	to	0.032	to	0.032	**to**	**0.032**	to	0.032	to	0.032
3	and	0.030	and	0.030	and	0.030			and	0.030	and	0.030
4	of	0.029	of	0.029	of	0.029			of	0.029	of	0.029
...												
8	was	0.015	**was**	**0.015**	was	0.015			was	0.015	was	0.015
...												
13	**she**	**0.011**			she	0.011			she	0.011	she	0.011
...												
254					both	0.0005			**both**	**0.0005**	both	0.0005
...												
435					sisters	0.0003					**sisters**	**0.0003**
...												
1701					**inferior**	**0.00005**						

MLE probabilities under a trigram model, from Manning and Schutze

In person	*she*	*was*	*inferior*	*to*	*both*	*sisters*

2-gram	$P(\cdot\|person)$		$P(\cdot\|she)$		$P(\cdot\|was)$		$P(\cdot\|inferior)$		$P(\cdot\|to)$		$P(\cdot\|both)$	
1	and	0.099	had	0.141	not	0.065	**to**	**0.212**	be	0.111	of	0.066
2	who	0.099	**was**	**0.122**	a	0.052			the	0.057	to	0.041
3	to	0.076			the	0.033			her	0.048	in	0.038
4	in	0.045			to	0.031			have	0.027	and	0.025
...												
23	**she**	**0.009**							Mrs	0.006	she	0.009
...												
41									what	0.004	**sisters**	**0.006**
...												
293									**both**	**0.0004**		
...												
∞					**inferior**	**0**						

MLE probabilities under a trigram model, from Manning and Schutze

In person	she	was	inferior	to	both	sisters

3-gram	$P(\cdot \mid In,person)$		$P(\cdot \mid person,she)$		$P(\cdot \mid she,was)$		$P(\cdot \mid was,inf.)$		$P(\cdot \mid inferior,to)$		$P(\cdot \mid to,both)$	
1	UNSEEN		did	0.5	not	0.057	UNSEEN		the	0.286	to	0.222
2			**was**	**0.5**	very	0.038			Maria	0.143	Chapter	0.111
3					in	0.030			cherries	0.143	Hour	0.111
4					to	0.026			her	0.143	Twice	0.111
...												
∞					**inferior**	**0**			**both**	**0**	**sisters**	**0**

MLE probabilities under a trigram model, from Manning and Schutze

In person	*she*	*was*	*inferior*		*to*	*both*	*sisters*
4-gram	$P(\cdot\|u,I,p)$	$P(\cdot\|I,p,s)$	$P(\cdot\|p,s,w)$		$P(\cdot\|s,w,i)$	$P(\cdot\|w,i,t)$	$P(\cdot\|i,t,b)$
1	UNSEEN	UNSEEN	in	1.0	UNSEEN	UNSEEN	UNSEEN
...							
∞			**inferior**	0			

MLE probabilities under a 4-gram model, from Manning and Schutze

Smoothing

- Estimating the probability of events that haven't occurred
- Laplace's law
  - add one to each count then renormalize
  - N=number of objects; B=vocabulary size

$$P_{\text{Lap}}(w_1, w_2, \ldots, w_n) = \frac{C(w_1, w_2, \ldots, w_n) + 1}{N + B}$$

  - Issues
    - probabilities depend on vocabulary size
    - much probability goes to unseen events
      - e.g. 44e6 words, 4e5 word types, 1.6e11 bigram types,

$r = f_{MLE}$	$f_{empirical}$	f_{Lap}
0	0.000027	0.000137
1	0.448	0.000274
2	1.25	0.000411
3	2.24	0.000548
4	3.23	0.000685
5	4.21	0.000822
6	5.23	0.000959
7	6.21	0.00109
8	7.21	0.00123
9	8.26	0.00137

Comparison of observed frequencies of bigrams vs very good estimates of what should have been observed vs Laplace smoothing estimates; from Manning and Schutze, after Church and Gale

- 44e6 words, 4e5 word types, 1.6e11 bigram types,

Lidstone's law; Jeffreys-Perks law

- Add some small number, rather than 1
  - if this is 0.5 Jeffreys-Perks, otherwise Lidstone
- Gives

$$P_{\text{Lid}}(w_1, w_2, \ldots, w_n) = \frac{C(w_1, w_2, \ldots, w_n) + \lambda}{N + B\lambda}$$

- Issues
  - small number means less probability on unseen events but where does number come from?
  - estimates are linear in MLE - doesn't seem reasonable at low probabilities

Held out estimates

- Assume
  - we have two data sets
    - counts will not in general be the same
- Strategy
  - identify bigrams with the same frequency in the first
  - estimate probability of each frequency in the second

Held out estimates

- Write
  - C1 for count in data set 1
  - C2 for count in data set 2
  - Nr for the number of bigrams with frequency r in dataset 1

$$T_r = \sum_{\text{ngrams such that } C_1 = \text{r}} C_2(\text{ngram})$$

  - if w_1, ... w_n has C1=r, then

$$P_{\text{ho}}(w_1, \ldots, w_n) = \frac{T_r}{N_r N_2}$$

Deleted estimation or Cross-validation

- But why the asymmetry?
- Instead, we could form

$$T_r^{ab} = \sum_{\text{ngrams such that } C_a = \text{r}} C_b(\text{ngram})$$

$$N_r^a = \sum_{\text{ngrams such that } C_a = \text{r}} 1$$

$$P_{\text{del}}(w_1, \ldots, w_n) = \frac{T_r^{01} + T_r^{10}}{N(N_r^0 + N_r^1)}$$

Good - Turing smoothing

- Improved estimate of frequency for object that occurs r times
  - fit (r, N_r) with some function S
    - S(r) is smoothed estimate of frequency r
- Good-Turing estimate is

$$P_{gt} = \frac{r^*}{N} \qquad r^* = \frac{(r+1)S(r+1)}{S(r)}$$

$$P_{gt}(0) = \frac{N_1}{N_0 N}$$

- Notice that this is poor for large r, so we use it for r<k

$r = f_{MLE}$	$f_{empirical}$	f_{Lap}	f_{del}	f_{GT}	N_r	T_r
0	0.000027	0.000137	0.000037	0.000027	74 671 100 000	2 019 187
1	0.448	0.000274	0.396	0.446	2 018 046	903 206
2	1.25	0.000411	1.24	1.26	449 721	564 153
3	2.24	0.000548	2.23	2.24	188 933	424 015
4	3.23	0.000685	3.22	3.24	105 668	341 099
5	4.21	0.000822	4.22	4.22	68 379	287 776
6	5.23	0.000959	5.20	5.19	48 190	251 951
7	6.21	0.00109	6.21	6.21	35 709	221 693
8	7.21	0.00123	7.18	7.24	27 710	199 779
9	8.26	0.00137	8.18	8.25	22 280	183 971

Comparison of observed frequencies of bigrams vs very good estimates of what should have been observed vs Laplace, deleted, Good-Turing; from Manning and Schutze, after Church and Gale; final columns number of bigrams with that frequency in training, further text

- 44e6 words, 4e5 word types, 1.6e11 bigram types,

Mixture estimates

$$
\begin{aligned}
P_{mix}(w_n|w_2, w_n - 1) &= \lambda_1 P(w_n) + \\
&\quad \lambda_2 P(w_n|w_1) + \\
&\quad \lambda_3 P(w_n|w_2, w_n - 1)
\end{aligned}
$$

- Weights are non-negative, convex
  - can estimate best set of weights using EM
  - more than trigrams are possible

Dynamical models - inference

- We know $P(X_{i+1}|X_i)$ $P(Y_i|X_i)$

- We want to estimate a set of states to maximize

$$P(X_0, \ldots, X_n | Y_0, \ldots, Y_n, \theta) = \frac{P(X_0, \ldots, X_n, Y_0, \ldots, Y_n | \theta)}{P(Y_0, \ldots, Y_n | \theta)}$$

Inference - model assumptions

- Our model has the properties:

$$P(X_{i+1}|X_0, \ldots, X_n) = P(X_{i+1}|X_i)$$

$$P(Y_i|X_0, \ldots, X_n) = P(Y_i|X_i)$$

- So that

$$\begin{aligned} P(X_0, \ldots, X_n, Y_0, \ldots, Y_n|\theta) \quad &= \quad (P(Y_0|X_0)P(X_0))] \times \\ &\quad\ (P(Y_1|X_1)P(X_1|X_0)) \times \\ &\quad\ \ldots \times \\ &\quad\ (P(Y_n|X_n)P(X_n|X_n - 1)) \end{aligned}$$

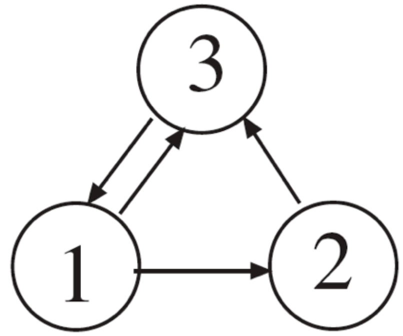
Inference

- Which means

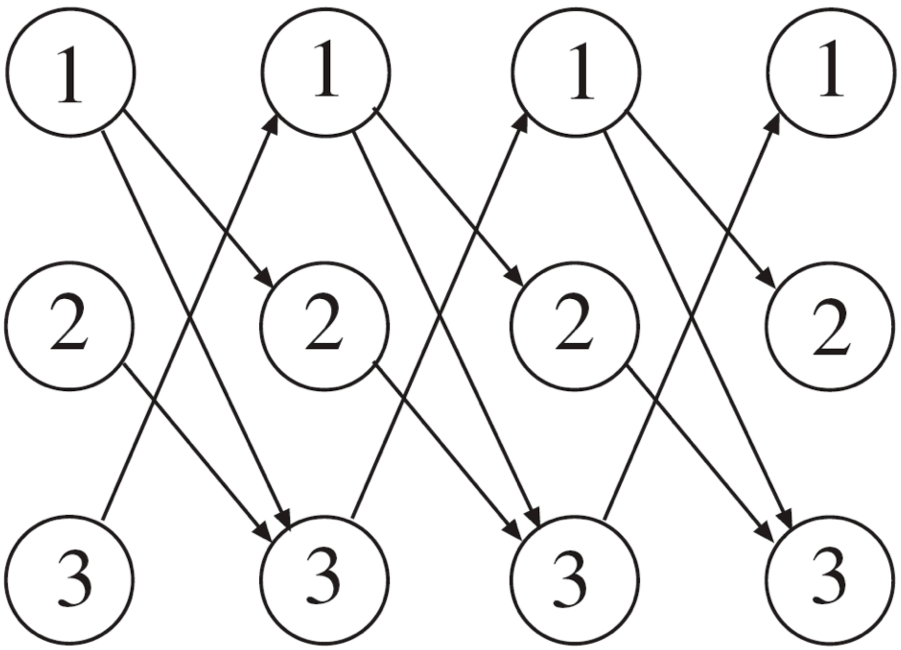
$$\begin{aligned}\log P(X_0,\ldots,X_n,Y_0,\ldots,Y_n|\theta) &= \log P(Y_0|X_0)+\log P(X_0)+\\ &\log P(Y_1|X_1)+\log P(X_1|X_0)+\\ &\ldots+\\ &\log P(Y_n|X_n)+\log P(X_n|X_n-1)\end{aligned}$$

- Set up a trellis
  - one column for each clock tick
  - one node for each state
  - one directed edge for each transition
  - weight with logs

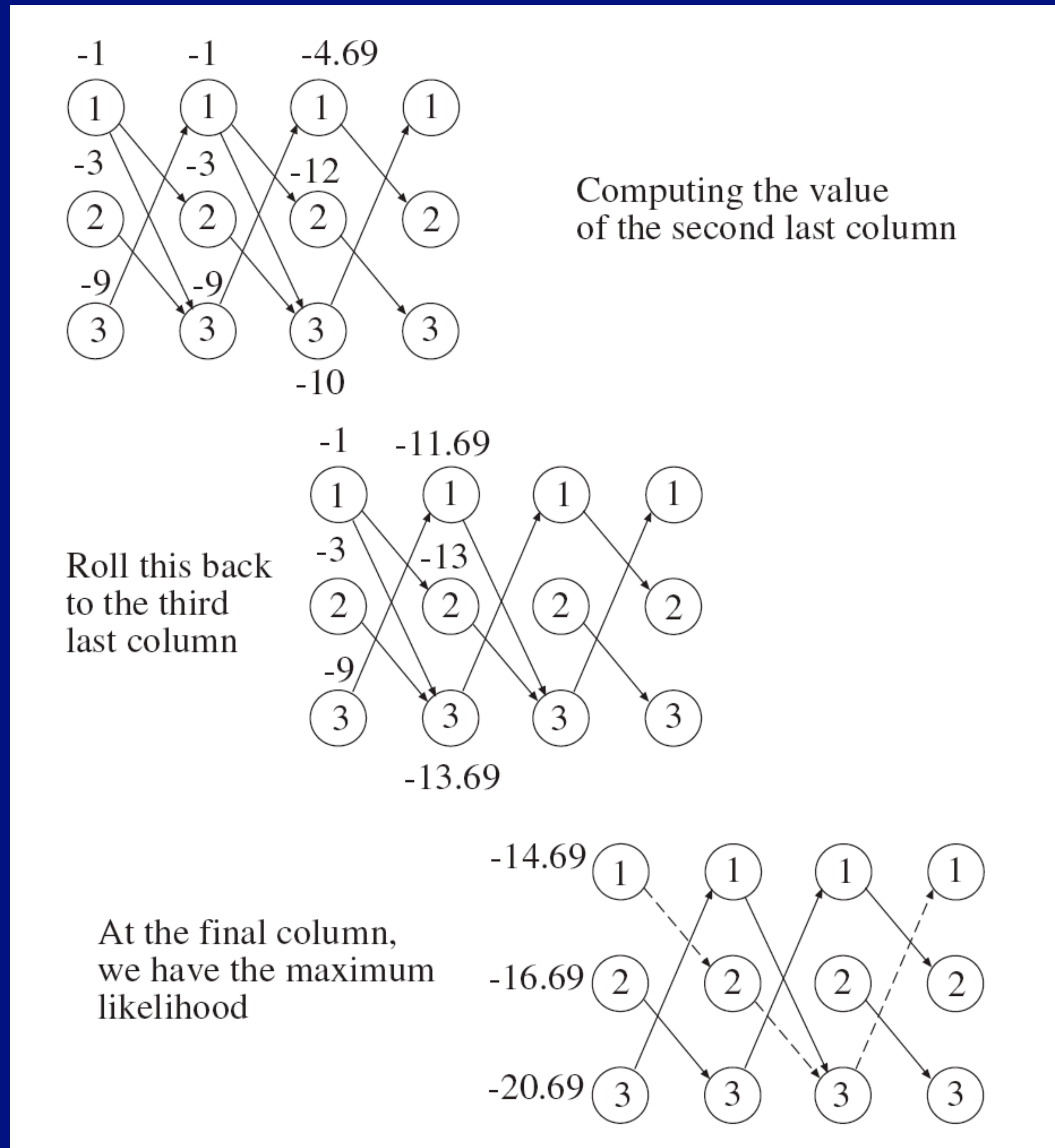
Simple state transition model



Trellis for four ticks



Dynamic programming reveals the maximum likelihood path (set of states)



More inference

- Dynamic programming can compute expectations

$$E(f) = \frac{\sum_{x_0,\ldots,x_n} \left(f(X_0 = x_0) \ldots, f(X_n = x_n) \right) P(X_0 = x_0, \ldots, X_n = x_n, Y_0, \ldots, Y_n|\theta)}{P(Y_0, \ldots, Y_n|\theta)}$$

- Notice

$$P(Y_0, \ldots, Y_n|\theta) = \sum_{x_0,\ldots,x_n} P(X_0 = x_0, \ldots, X_n = x_n, Y_0, \ldots, Y_n|\theta)$$

- So all we care about is:

$$N(f) = \sum_{x_0,\ldots,x_n} \left(f(X_0 = x_0) \ldots, f(X_n = x_n) \right) P(X_0 = x_0, \ldots, X_n = x_n, Y_0, \ldots, Y_n|\theta)$$

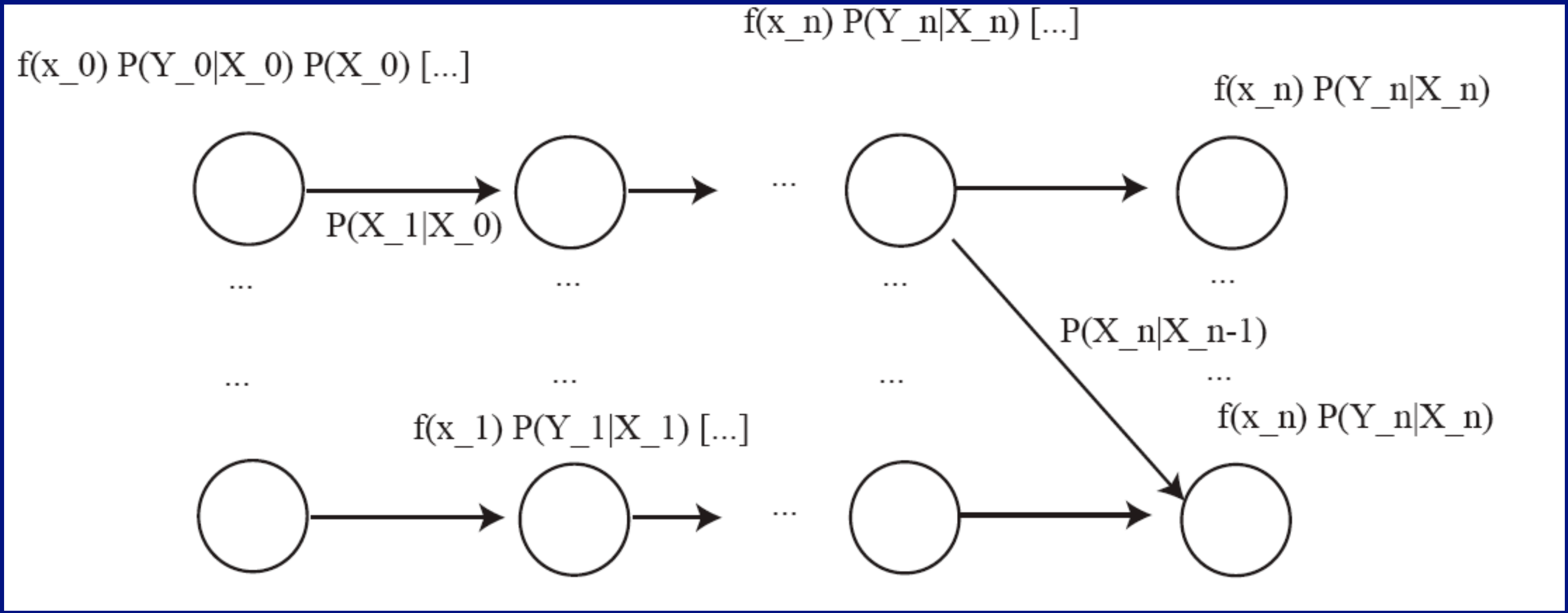
But the sum decomposes

$$N(f) = \sum_{x_0,\ldots,x_n} \left(f(x_0)\ldots f(x_n)\right) P(X_0,\ldots,X_n,Y_0,\ldots,Y_n|\theta)$$

- is the same as

$$\sum_{x_0}\left[f(x_0)P(Y_0|X_0)P(X_0)\left[\sum_{x_1} f(x_1)P(Y_1|X_1)P(X_1|X_0)\left[\sum_{x_2} f(x_2)P(Y_2|X_2)P(X_2|X_1)\left[\ldots\right]\right]\right]\right]$$

- notice that each bracket depends on only the previous



Dynamic programming yields expectations

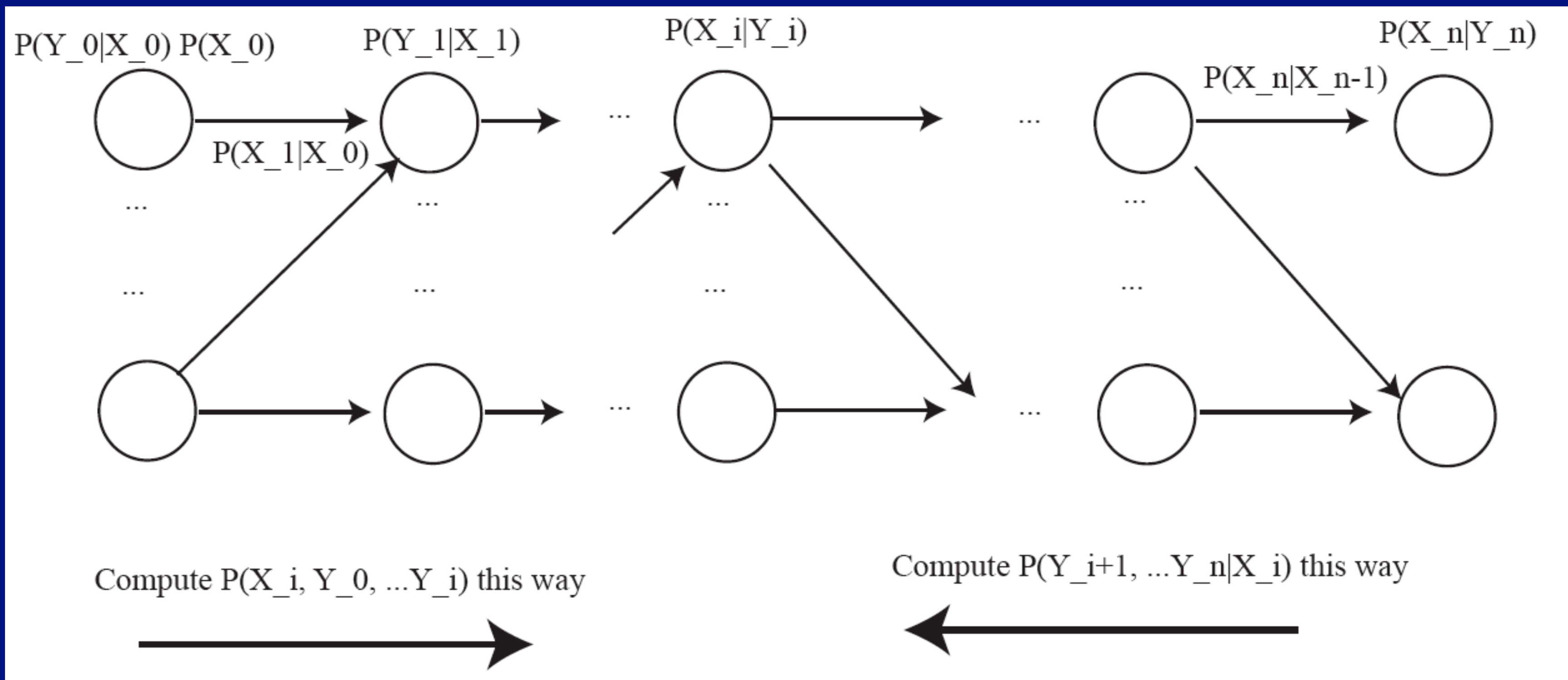
We can compute other things, too

- Consider $$P(X_i, Y_0, \ldots Y_n) = \sum_{x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n} P(X_0, \ldots, X_n, Y_0, \ldots, Y_n | \theta)$$

$$\left[\sum_{x_0, \ldots, x_{i-1}} P(X_0, \ldots, X_i, Y_0, \ldots, Y_i) \right] \left[\sum_{x_{i+1}, \ldots, x_n} P(X_{i+1}, \ldots, X_n, Y_{i+1}, \ldots, Y_n | X_i) \right]$$

P(X_i, Y_0, ..., Y_i)
Compute this moving backward in time

P(Y_i+1, ..., Y_n|X_i)
Compute this moving forward in time



P(Y_0|X_0) P(X_0)
P(Y_1|X_1)
P(X_i|Y_i)
P(X_n|Y_n)
P(X_n|X_n-1)
P(X_1|X_0)
Compute P(X_i, Y_0, ...Y_i) this way
Compute P(Y_i+1, ...Y_n|X_i) this way

Training a dynamical model

- For the moment, assume
  - that transition probabilities are known
- If hidden state were known at each tick, training the emission model would be easy
  - parameter estimation for continuous emission model
  - counting for discrete model
- Idea:
  - new variable to indicate which hidden state is occupied

Simplest case:

- no dynamics
- emission is Normal, with a mean that depends on state; fixed covariance Σ
- there are k states
- Y is continuous
- $P(X)$ is known $= \pi$
- write $\mu_1 \dots \mu_k$ for means;
- we have N observations $y^{(1)} \dots y^{(n)}$
- we need to estimate $\mu_1 \dots \mu_k$

$$P(Y \mid \mu_1 \dots \mu_k, \pi)$$

$$= \frac{1}{K}\left[\sum_i e^{-\frac{(y-\mu_i)^T \Sigma^{-1} (y-\mu_i)}{2}} \cdot \pi_i\right]$$

- normalizing constant for Gaussians
- This is a mixture of Gaussians.

Maximising log-likelihood of this is not a good idea.

$$\sum_{j \in \text{data}} \log P(Y = y_j \mid \mu_1 \cdots \mu_K, \pi)$$

$$= \sum_{j \in \text{data}} \left[\log \left[\sum_{i \in \text{states}} e^{\frac{(y_j - \mu_i)^T \Sigma^{-1} (y_j - \mu_i)}{2}} \cdot \pi_i \right] \right]$$

But this is very difficult to work with; in particular, multiple local minima, etc.

This is $P(\underbrace{y_1 \cdots y_N}_{D} \mid \underbrace{\mu_1 \cdots \mu_K, \pi}_{\Theta})$

- Sometimes known as incomplete data log-likelihood
- This is because if we knew the state for each y, estimating μ's would be 'easy'.

Algorithmic recipe

EM = expectation-maximization

- write H for hidden data.
- write $P(D, H | \Theta)$ — CDLLH = complete data log-likelihood
- assume we have an estimate $\Theta^{(n)}$
- we want a better estimate
- $$Q(\Theta; \Theta^{(n)}) = E_{X|\Theta^{(n)}} [\log P(D, H | \Theta)]$$

i.e compute an expected log-likelihood
- this incorporates all we know about H to date

$$\Theta^{(n+1)} = \arg\max Q(\Theta; \Theta^{(n)})$$

Easy way to encode hidden state is with characteristic functions

$$\delta_{ij} = \begin{cases} 1 & \text{if state} = i \text{ on } j\text{th data item} \\ 0 & \text{otherwise} \end{cases}$$

In this case

$$\log P(D, \mathcal{H} | \theta) =$$

$$\sum_{j \in data} \left[\sum_{i \in states} \left\{ -(y_j - \mu_i)' \underset{\approx}{\Sigma}^{-1} (y_j - \mu_i) \right\} \cdot \delta_{ij} \right]$$

$$+ K + \log P(\mathcal{H} | \theta)$$

And

$$\log P(\mathcal{H} | \theta) = \sum_{j \in data} \left[\sum_{i \in states} \pi_i \cdot \delta_{ij} \right]$$

You should think of δ_{ij} as <u>switches</u>

Now consider $Q(\Theta; \Theta^n)$

1) $\log P(D, \mathcal{H} | \Theta)$ is linear in $\mathcal{H}$ (δ_{ij})

2) So we can get Q by replacing δ_{ij} with $E_{\delta_{ij}|\Theta^{(n)}, D}[\delta_{ij}]$

3) $E_{\delta_{ij}|\Theta^{(n)}, D}[\delta_{ij}] = 1 \cdot P(\delta_{ij} = 1 | D, \Theta^{(n)}) + 0 \cdot \ldots$

$$P(\delta_{ij} = 1 | D, \Theta) = P(\delta_{ij} = 1 | y_j, \Theta^{(n)})$$

$$= \frac{P(y_j | \delta_{ij} = 1, \Theta^{(n)}) \cdot P(\delta_{ij} = 1 | \Theta^{(n)})}{\left[\sum_u P(y_j | \delta_{uj} = 1, \Theta^{(n)}) P(\delta_{uj} = 1 | \Theta^{(n)}) \right]}$$

$\longrightarrow$ this is $P(y_j | \Theta^{(n)})$

now this is

$$\frac{e^{-(y_j - \mu_i)' \frac{\Sigma^{-1}}{2} (y_j - \mu_i)} \cdot \pi_i}{\sum_u \left[e^{-(y_j - \mu_u)' \frac{\Sigma^{-1}}{2} (y_j - \mu_u)} \pi_u \right]}$$

Procedure:

- start with $\theta^{(0)}$
- form $E_{\delta_{ij} | \theta^{(0)}, D} [\delta_{ij}]$
- plug into COLLH
- max wrt θ

Soft counts interpretation

y_j counts toward $\mu_i^{(n+1)}$ by $E[\delta_{ij}]$

this gives

$$\mu_i^{(n+1)} = \frac{\sum_j E[\delta_{ij}] \cdot y_j}{\sum_j E[\delta_{ij}]}$$

You can get this result w/ differentiation, too.

<u>HMM with dynamics, discrete measurements</u>:

- assume $P(X_{i+1} \mid X_i = x)$ known
  $P(X_0)$ known
- assume discrete states
- emission: $P(Y = y_u \mid X = v) = p_{uv}$
  - this is a <u>table</u>
  - indep. of time
- Missing variable $\delta_{ij}^k = \begin{cases} 1 & \text{if } j\text{'th elem of } u\text{'th seq has } X = x_i \\ 0 & \text{otherwise} \end{cases}$

CDLLH:

$$P(D, H|\theta) = P(D|H, \theta) P(H|\theta)$$

Now $\log P(D|H,\theta) =$

$$\sum_{u \in segs} \left[\sum_{j \in elems} \left\{ \sum_{i \in states} \log P\left(Y_j^{(u)} = y_j \,\middle|\, X_j^{(u)} = x_i\right) \delta_{ij}^{u} \right\} \right]$$

~~and $\log P(H|\theta) = \sum_{u \in segs} \left[\sum_{j \in elems} \left[\sum_{i \in states} \left\{ \sum_{k \in states} \log P(x_i|x_k) \right. \right. \right.$~~

$\log P(H|\theta)$

$$= \sum_{u \in segs} \left[\sum_{j \in elems} \left\{ \sum_{i \in states} \left(\sum_{k \in states} \log P(x_i|x_k) \cdot \delta_{k\,j-1}^{(u)} \, \delta_{ij}^{u} \right) \right\} \right]$$

All this looks hairy.

Notice that if $P(x_i | x_j)$ is known, then the second term is not involved in estimation

$$E_{\delta|\theta^{(n)}}\left[\delta_{ij}^{u}\right] = P\left(X_i^{(u)} = x_j \mid D, \theta^{(n)}\right)$$

But we know how to estimate this!

M-Step:

- notice that $\log P(H|\theta)$ doesn't do anything
- notice that CDLCH is linear in hidden vars
- so we can use soft counts interp (or set grad to zero, etc.)

and we get

$$P(Y = y_e \mid X = x_m, \Theta^{(n)})$$

$$= \frac{\{\text{soft count of in } x_m\text{, emitted } y_e\}}{\{\text{soft count of in } x_m\}}$$

$$= \frac{\sum_{u \in segs} \sum_{j \in els} 1\{Y_j^u = y_e\} \cdot P(X_j^u = x_m \mid D, \Theta^{(n)})}{\sum_{u \in segs} \sum_{j \in els} P(X_j^u = x_m \mid D, \Theta^{(n)})}$$

<u>Learning the dynamics</u>:

- notice that, if transition probs are not known, maximising the second term yields them w/ soft counts

-

$$P(X_{j+1} = x_e \mid X_j = x_m, D, \theta^{(n)})$$

$$= \frac{\{\text{soft count of } x_e \to x_m \text{ transitions}\}}{\{\text{soft count of all transitions } x_e \to\}}$$