Hidden Markov Models

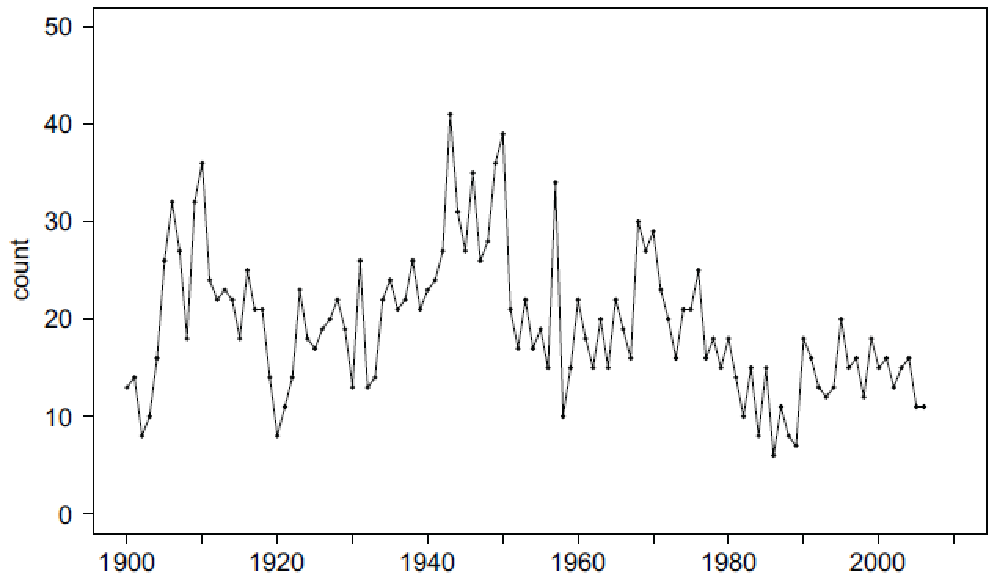
Tomasz Burzykowski

tomasz.burzykowski@uhasselt.be

Earthquakes data

Table 1.1 *Number of major earthquakes (magnitude 7 or greater) in the world, 1900–2006; to be read across rows.*

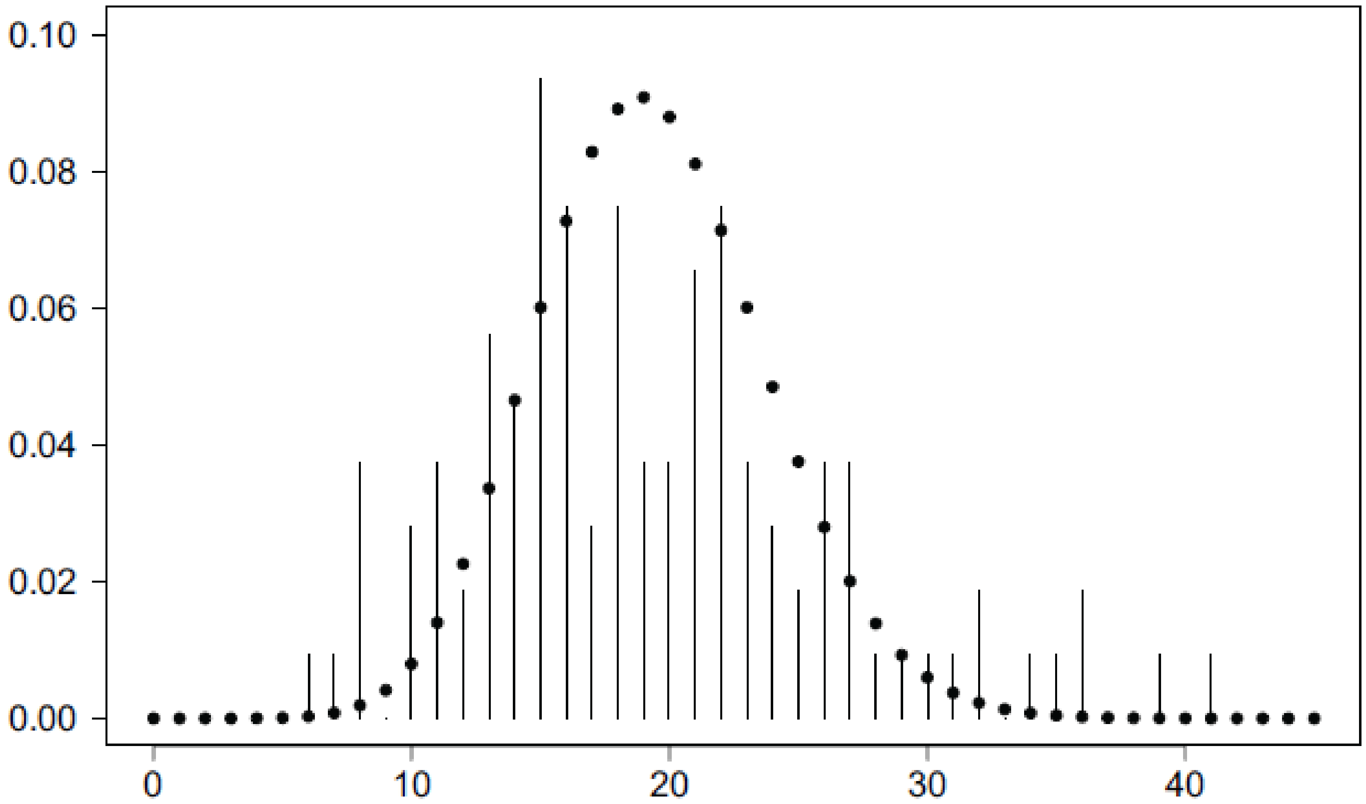
13	14	8	10	16	26	32	27	18	32	36	24	22	23	22	18	25	21	21	14
8	11	14	23	18	17	19	20	22	19	13	26	13	14	22	24	21	22	26	21
23	24	27	41	31	27	35	26	28	36	39	21	17	22	17	19	15	34	10	15
22	18	15	20	15	22	19	16	30	27	29	23	20	16	21	21	25	16	18	15
18	14	10	15	8	15	6	11	8	7	18	16	13	12	13	20	15	16	12	18
15	16	13	15	16	11	11													



REF: Zucchini & MacDonald (2009)

Earthquakes data

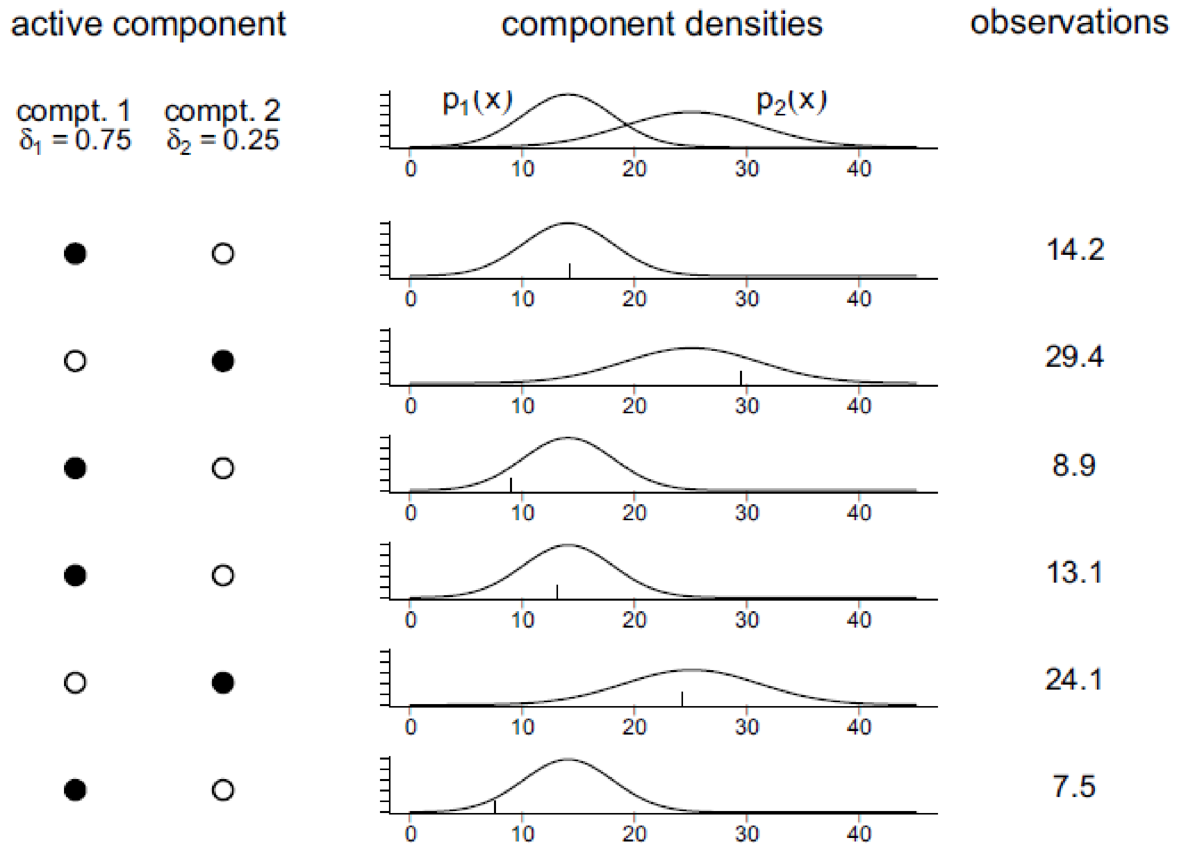
- Fitted Poisson
- Overdispersion: sample mean 19.4, variance 51.6
- Mixture?



REF: Zucchini & MacDonald (2009)

Earthquakes data

- Mixture: $P(X{=}x){=}\sum_q P(Q{=}q)P(X{=}x \mid Q{=}q) \equiv \sum_q \pi_q P_q(x)$
- Example: two Poisson rates, λ_1 and λ_2
- $\mathrm{E}(X) = \pi_1\lambda_1 + (1-\pi_1)\lambda_2$; $\mathrm{Var}(X) = \mathrm{E}(X) + \pi_1(1-\pi_1)(\lambda_1-\lambda_2)^2$

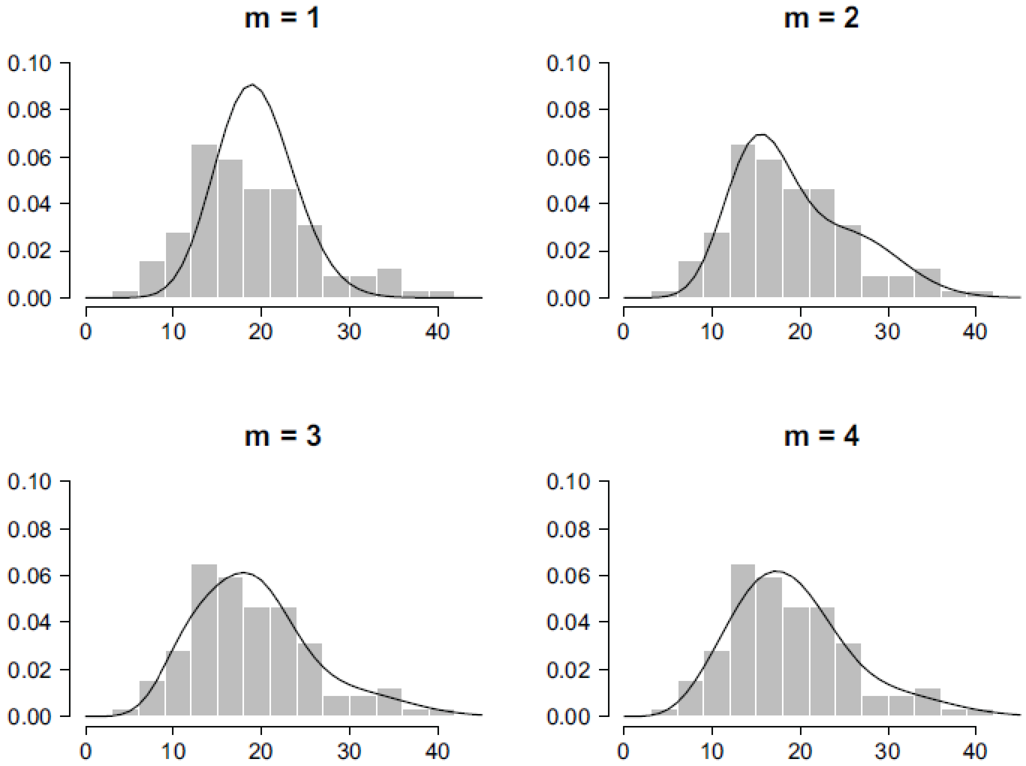


REF: Zucchini & MacDonald (2009)

Earthquakes data

- Mixtures with 1 to 4 components

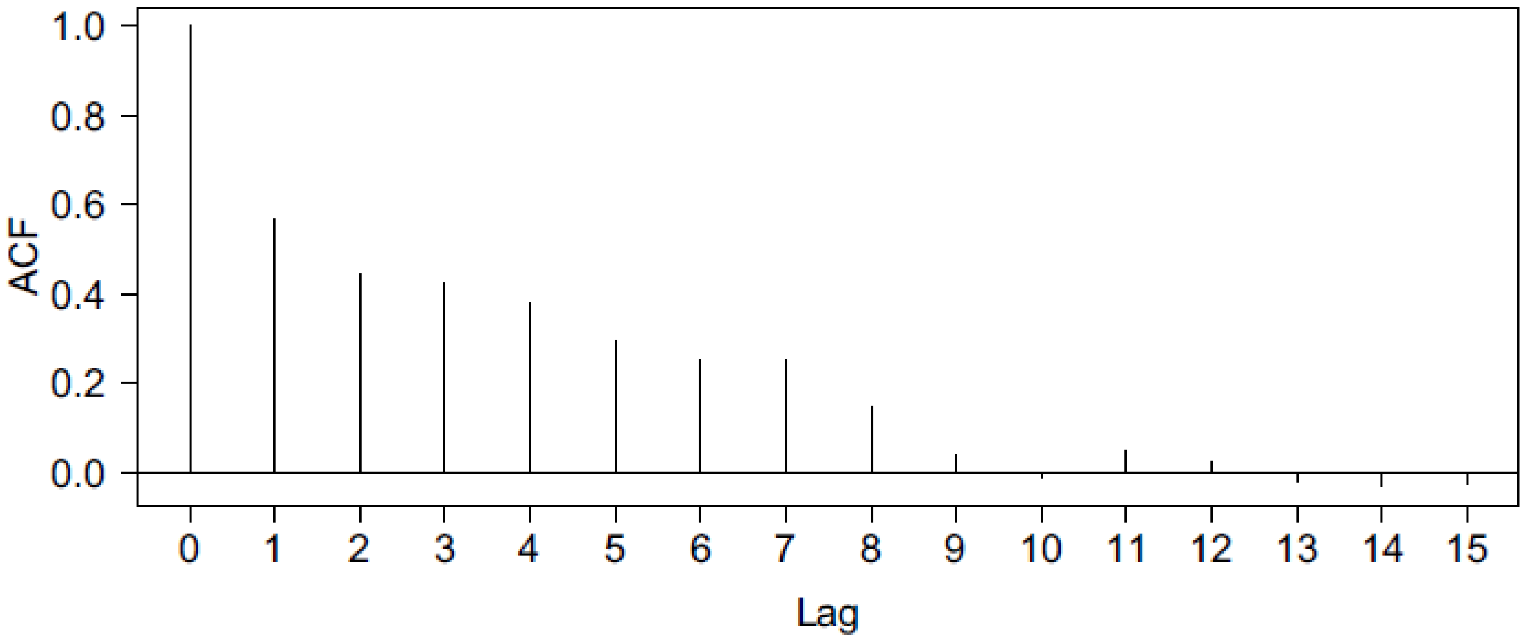
model	i	δ_i	λ_i	$-\log L$	mean	variance
$m = 1$	1	1.000	19.364	391.9189	19.364	19.364
$m = 2$	1	0.676	15.777	360.3690	19.364	46.182
	2	0.324	26.840			
$m = 3$	1	0.278	12.736	356.8489	19.364	51.170
	2	0.593	19.785			
	3	0.130	31.629			
$m = 4$	1	0.093	10.584	356.7337	19.364	51.638
	2	0.354	15.528			
	3	0.437	20.969			
	4	0.116	32.079			
observations					19.364	51.573



REF: Zucchini & MacDonald (2009)

Earthquakes data

- Serial dependence
- Mixtures assume independence



REF: Zucchini & MacDonald (2009)

Markov Chains

- A sequence of discrete random variables ("states") $Q_1, Q_2, Q_3, \ldots$
  - We will assume a finite set of values 1,2, ..., m

- In general:

$$P(Q_1,Q_2,\ldots,Q_{t+1})=P(Q_1)P(Q_2|Q_1)P(Q_3|Q_1,Q_2)\cdot\ldots\cdot P(Q_{t+1}|Q_1,\ldots,Q_t)$$

- 1st-order M. chain: $P(Q_{t+1}|Q_1,Q_2,\ldots,Q_t) \equiv P(Q_{t+1}|Q_t)$
- 2nd-order M. chain: $P(Q_{t+1}|Q_1,Q_2,\ldots,Q_t) \equiv P(Q_{t+1}|Q_{t-1},Q_t)$

- Etc.

Markov Chains

- Transition probabilities: $P(Q_{s+t} = i \mid Q_s = j)$
- Homogenous chain: no dependence on s

$[\boldsymbol{A}(t)]_{ij} \equiv a_{ij}(t) \equiv P(Q_{s+t} = i \mid Q_s = j)$

- It follows that $\boldsymbol{A}(t+u)=\boldsymbol{A}(t)\boldsymbol{A}(u)$ and $\boldsymbol{A}(t) = \boldsymbol{A}(1)^t$
- Hence, define $\boldsymbol{A} \equiv \boldsymbol{A}(1)$, with $a_{ij} \equiv a_{ij}(1)$

- Note that, for each row i, $\sum_j a_{ij} = 1$

Markov Chains

- Consider $\boldsymbol{u}(t) = (P(Q_t = 1), P(Q_t = 2), \ldots, P(Q_t = m))$
  - unconditional "state" distribution

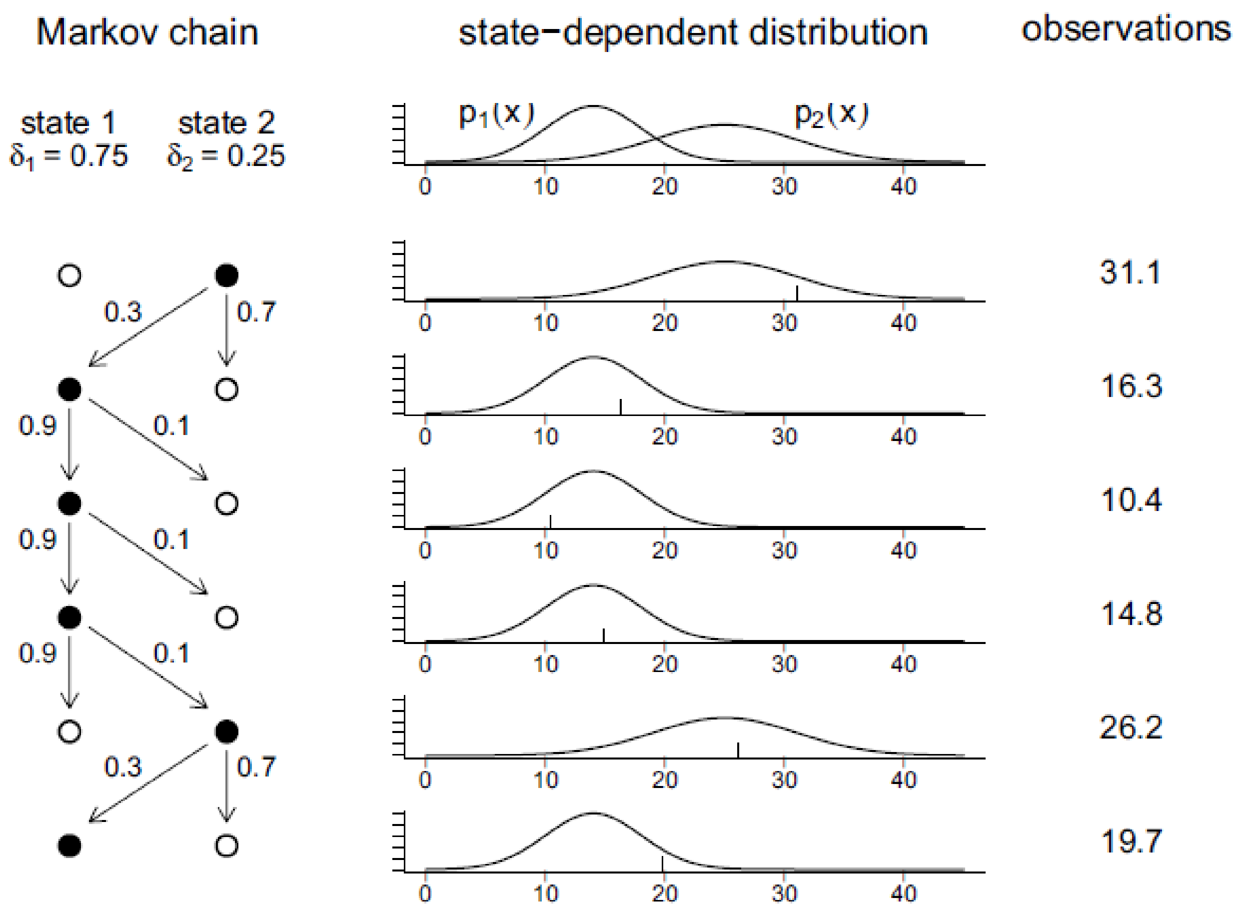
- $\boldsymbol{u}(1)$ – the initial "state" distribution
- Note: $\boldsymbol{u}(t) = \boldsymbol{u}(1)\boldsymbol{A}^t$

- A stationary distribution: $\boldsymbol{u}^*\boldsymbol{A} = \boldsymbol{u}^*$
  - If $\boldsymbol{u}(1) = \boldsymbol{u}^*$, then $\boldsymbol{u}(t) = \boldsymbol{u}^*$; the same distribution at all t

- Stationary M. chain: $\boldsymbol{u}(t)$ the same for all t

Earthquakes data

- To deal with serial dependence in the data, assume that the Poisson rates depend on “states” forming a Markov chain



REF: Zucchini & MacDonald (2009)

The "Fair Bet Casino"

- The game is to flip coins, which results in only two possible outcomes: **H**ead or **T**ail.
- The **F**air coin will give **H**eads and **T**ails with same probability ½.
- The **B**iased coin will give **H**eads with prob. ¾.

The "Fair Bet Casino" (cont'd)

- Thus, we define the probabilities:
  - $P(H|F) = P(T|F) = ½$
  - $P(H|B) = ¾$, $P(T|B) = ¼$
  - The crooked dealer changes between Fair and Biased coins with probability 10%.

The Fair Bet Casino Problem

- **Input:** A sequence $\underline{x} = x_1x_2x_3\ldots x_n$ of coin tosses made by two possible coins (***F*** or ***B***).
- **Output:** A sequence $\underline{q} = q_1q_2q_3\ldots q_n$, with each q_i being either *F* or *B* indicating that x_i is the result of tossing the Fair or Biased coin respectively.

Problem…

Fair Bet Casino Problem

Any observed outcome of coin tosses could have been generated by any sequence of states!

Need to incorporate a way to grade different sequences differently.

Decoding Problem

P($\underline{x}$ |fair coin) vs. P($\underline{x}$ |biased coin)

- Suppose first that dealer never changes coins. Some definitions…:
  - P($\underline{x}$|fair coin): prob. of the dealer using the *F* coin and generating the outcome $\underline{x}$.
  - P($\underline{x}$|biased coin): prob. of the dealer using the *B* coin and generating outcome $\underline{x}$.
  - *k* the number of ***H***eads in $\underline{x}$.

$P(\underline{x}\,|\text{fair coin})$ vs. $P(\underline{x}\,|\text{biased coin})$

- $P(\underline{x}|\text{fair coin}) = P(x_1 \ldots x_n|\text{fair coin}) =$
  $\Pi_{i=1,n}\, p(x_i|\text{fair coin}) = (1/2)^n$

- $P(\underline{x}|\text{biased coin}) = P(x_1 \ldots x_n|\text{biased coin}) =$
  $\Pi_{i=1,n}\, p(x_i|\text{biased coin}) = (3/4)^k (1/4)^{n-k} = 3^k/4^n$

  - k - the number of ***H***eads in $\underline{x}$.

P($\underline{x}$ |fair coin) vs. P($\underline{x}$ |biased coin)

- P($\underline{x}$ |fair coin) = P($\underline{x}$ |biased coin)
  when $k = n / log_2 3$
  $k \sim 0.67n$

Log-odds Ratio

- We define ***log-odds ratio*** as follows:

$$\log_2(P(\underline{x}\,|\text{fair coin}) \,/\, P(\underline{x}\,|\text{biased coin}))$$
$$= \Sigma^{n}_{i=1} \log_2(p(x_i \mid F) \,/\, p(x_i \mid B))$$
$$= n - k \log_2 3$$

- Not really log-odds, but log-likelihood ratio
- 0 if $k = n / \log_2 3$
- *Biased (fair)* coin most likely used if log-OR<0 (>0)

Computing Log-odds Ratio in Sliding Windows

$$x_1 x_2 \boxed{x_3 x_4 x_5 x_6 x_7} x_8 \ldots x_n$$

Consider a *"sliding window"* of the outcome sequence. Find the log-odds for this short window.

Biased coin most likely used

Fair coin most likely used

Disadvantages:
- the length of the window is not known in advance
- different windows may classify the same position differently

Hidden Markov Model (HMM)

- Can be viewed as an abstract machine with *m hidden* states that emits symbols from an alphabet Σ with *r* symbols.
- Each state has its own emission probability distribution.
- The machine switches between states according to some probability distribution.
- While in a certain state, the machine makes 2 decisions:
  - What state should I move to next?
  - What symbol - from the alphabet Σ - should I emit?

Why “Hidden”?

- Observers can see the emitted symbols of an HMM but have *no ability to know which state the HMM is currently in*.
- Thus, the goal is to infer the most likely hidden states of an HMM based on the given sequence of emitted symbols.

HMM Parameters

Σ: set of all *r* possible emission characters

Ex.: $\Sigma = \{H, T\}$ for coin tossing

$\Sigma = \{1, 2, 3, 4, 5, 6\}$ for dice tossing

Q: set of *m* hidden states, each emitting symbols from Σ

Q={F,B} for coin tossing

HMM Parameters (cont'd)

$A = (a_{kl})$: an m x m matrix of probability of changing from state k to state l

$$a_{FF} = 0.9 \qquad a_{FB} = 0.1$$
$$a_{BF} = 0.1 \qquad a_{BB} = 0.9$$

$E = (e_k(b))$: an m x r matrix of probability of emitting symbol b during a step in which the HMM is in state k

$$e_F(T) = \tfrac{1}{2} \qquad e_F(H) = \tfrac{1}{2}$$
$$e_B(T) = \tfrac{1}{4} \qquad e_B(H) = \tfrac{3}{4}$$

Markov Chain Property

$$
\begin{aligned}
P(q_1 q_2 q_3 \ldots q_n) &= P(q_1)P(q_2|q_1)P(q_3|q_2 q_1)P(q_n|q_{n-1} \ldots q_2 q_1) \\
&\equiv P(q_1)P(q_2|q_1)P(q_3|q_2)\ldots P(q_n|q_{n-1}) \\
&= P(q_1)\, a_{q_1,q_2} a_{q_2,q_3} \ldots a_{q_{n-1},q_n}
\end{aligned}
$$

Additionally:

Given the state, the emission of different symbols is independent.

The emission of different symbols in different states is independent.

HMM for Fair Bet Casino

- The *Fair Bet Casino* in *HMM* terms:

  $\Sigma = \{0, 1\}$ (0 for ***T***ails and 1 ***H***eads)

  Q = {*F*,*B*} – *F* for Fair & *B* for Biased coin.

- Transition Probabilities *A*

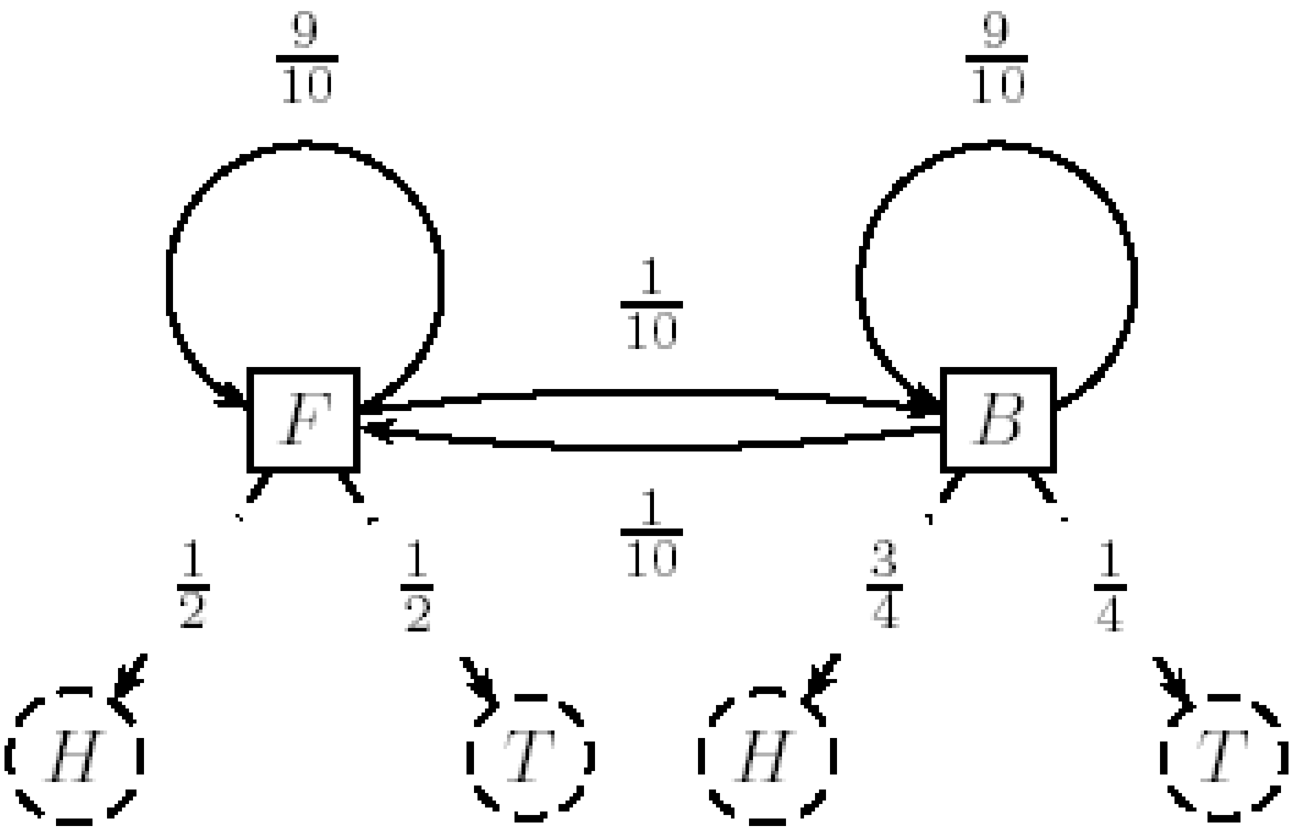
	Biased	Fair
Biased	$a_{BB} = 0.9$	$a_{FB} = 0.1$
Fair	$a_{BF} = 0.1$	$a_{FF} = 0.9$

HMM for Fair Bet Casino (cont'd)

Emission Probabilities *E*

	Tails(0)	Heads(1)
Fair	$e_F(0) = ½$	$e_F(1) = ½$
Biased	$e_B(0) = ¼$	$e_B(1) = ¾$

HMM for Fair Bet Casino (cont'd)



HMM model for the *Fair Bet Casino* Problem

Hidden Paths

- A *path* $\underline{q} = q_1 \dots q_n$ in the HMM is defined as a sequence of states.
- Consider path $\underline{q}$ = FFFBBBBBFFF and sequence $\underline{x}$ = 01011101001

Probability of x_i was emitted from state q_i

$\underline{x}$	0	1	0	1	1	1	0	1	0	0	1
$\underline{q}$ =	F	F	F	B	B	B	B	B	F	F	F
$P(x_i \mid q_i)$	½	½	½	¾	¾	¾	¼	¾	½	½	½
$P(q_{i-1} \rightarrow q_i)$	½	$^9/_{10}$	$^9/_{10}$	$^1/_{10}$	$^9/_{10}$	$^9/_{10}$	$^9/_{10}$	$^9/_{10}$	$^1/_{10}$	$^9/_{10}$	$^9/_{10}$

Transition probability from state q_{i-1} to state q_i

P($\underline{x}$) Calculation

- P($\underline{x}$): Probability of observing sequence $\underline{x}$, given the model *M*.

$$P(\underline{x}) = \Sigma_{\underline{q}} P(\underline{x} \mid \underline{q}) \cdot P(\underline{q})$$

$$= \Sigma_{\underline{q}}\{P(q_0 \rightarrow q_1) \cdot P(x_1 \mid q_1) \cdot P(q_1 \rightarrow q_2) \cdot \ldots \cdot P(q_{n-1} \rightarrow q_n) \cdot P(x_n \mid q_n)\}$$

$$= \Sigma_{\underline{q}}\{a_{q_0,q_1} \cdot e_{q_1}(x_1) \cdot a_{q_1,q_2} \cdot \ldots \cdot a_{q_{n-1},q_n} \cdot e_{q_n}(x_n)\}$$

$$= \Sigma_{\underline{q}}\{a_{q_0,q_1} \cdot \Pi^{n}_{t=1} e_{q_t}(x_t) \cdot \Pi^{n-1}_{t=1} a_{q_t,q_{t+1}}\}$$

  - $P(q_0 \rightarrow q_1) = a_{q_0,q_1}$: for starting in state q_1 (imaginary state 0 “start”)

- Requires $2nm^n$ computations
  - Sum of m^n terms, each being a product with $2n$ multiplications
  - Impossible numerically: for m=5 states, n=100, $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$

$P(\underline{x})$ Calculation: Forward Algorithm

- One can write

$$P(\underline{x}) \quad = \quad \sum_{i=1}^{m} P(\underline{x},\, q_n = Q_i)$$

- "Forward variable": $P(x_1, \ldots, x_t, q_t = Q_i)$

- The following holds

$$P(x_1, q_1 = Q_i) = e_{Q_i}(x_1)\, a_{q_0, Q_i}$$

$$P(x_1, \ldots, x_{t+1}, q_{t+1} = Q_i) = \left\{ \sum_{j=1}^{m} P(x_1, \ldots, x_t, q_t = Q_j)\, a_{Q_j, Q_i} \right\} e_{Q_i}(x_{t+1})$$

- Recursion!
- Requires ~ nm^2 computations
  - m values (t=1,2,…), each a sum of m products
  - Feasible numerically: for m=5 states, n=100, $25 \cdot 100 \approx 2500$, not 10^{72}

$P(\underline{x})$ Calculation: Backward Algorithm

- One can also write

$$P(\underline{x}) = \sum_{i=1}^{m} P(\underline{x}, q_t=Q_i) =$$

$$\sum_i P(x_1,\ldots,x_t, q_t=Q_i)\, P(x_{t+1}, \ldots, x_n | x_1,\ldots,x_t, q_t=Q_i) =$$

$$\sum_i P(x_1,\ldots,x_t, q_t=Q_i)\, P(x_{t+1}, \ldots, x_n | q_t=Q_i)$$

- $P(x_1,\ldots,x_t, q_t=Q_i)$ come from the forward algorithm

Backward Algorithm

- "Backward variable": $P(x_{t+1}, \ldots, x_n | q_t=Q_j)$

- The following holds

$$P(x_n \mid q_{n-1}=Q_i) = \sum_{j=1}^{m} a_{Q_i,Q_j} e_{Q_j}(x_n)$$

$$P(x_t, \ldots, x_n \mid q_{t-1}=Q_i) = \left\{\sum_{j=1}^{m} P(x_{t+1}, \ldots, x_n \mid q_t=Q_j)\, a_{Q_i,Q_j}\right\} e_{Q_j}(x_t)$$

- Recursion again!

P($\underline{x}$) Calculation

- We can use matrix notation:

  $$P(\underline{x}) = \Sigma_{\underline{q}}\{a_{q_0,q_1} \cdot e_{q_1}(x_1) \cdot a_{q_1,q_2} \cdot \ldots \cdot a_{q_{n-1},q_n} \cdot e_{q_n}(x_n)\}$$

  $$= \boldsymbol{a_0}\boldsymbol{E}(x_1)\boldsymbol{A}\boldsymbol{E}(x_2) \cdot \ldots \cdot \boldsymbol{A}\boldsymbol{E}(x_n)\mathbf{1}^T$$

  where $\boldsymbol{a_0}$ is the initial state distribution, $\boldsymbol{A}$ is the transition probability matrix, $\mathbf{1}$ is the vector of ones, and $\boldsymbol{E}(x)=diag(p_1(x), p_2(x),\ldots,p_m(x))$.

- Let $\boldsymbol{B}_t \equiv \boldsymbol{A}\boldsymbol{E}(x_t)$, then $P(\underline{x}) = \boldsymbol{a_0}\boldsymbol{E}(x_1)\boldsymbol{B}_2 \cdot \ldots \cdot \boldsymbol{B}_n\mathbf{1}^T$

- If $\boldsymbol{a_0}$ is the stationary distribution, then $\boldsymbol{a_0}\boldsymbol{A}=\boldsymbol{a_0}$, *so*

  $$P(\underline{x})=\boldsymbol{a_0}\boldsymbol{E}(x_1)\boldsymbol{B}_2 \cdot \ldots \cdot \boldsymbol{B}_n\mathbf{1}^T=\boldsymbol{a_0}\boldsymbol{A}\boldsymbol{E}(x_1)\boldsymbol{B}_2 \cdot \ldots \cdot \boldsymbol{B}_n\mathbf{1}^T= \boldsymbol{a_0}\boldsymbol{B}_1\boldsymbol{B}_2 \cdot \ldots \cdot \boldsymbol{B}_n\mathbf{1}^T$$

P($\underline{x}$) Calculation

- In matrix notation, P($\underline{x}$) = $\boldsymbol{a}_0\boldsymbol{E}(x_1)\boldsymbol{AE}(x_2)\cdot\ldots\cdot\boldsymbol{AE}(x_n)\mathbf{1}^T$
- Define $\boldsymbol{\alpha}_t = \boldsymbol{a}_0\boldsymbol{E}(x_1)\boldsymbol{AE}(x_2)\cdot\ldots\cdot\boldsymbol{AE}(x_t)=\boldsymbol{a}_0\boldsymbol{E}(x_1)\prod_{s=2}^{t}\boldsymbol{AE}(x_s)$
- P($\underline{x}$) = $\boldsymbol{\alpha}_n\mathbf{1}^T$
- And $\boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t-1}\boldsymbol{AE}(x_t)$ for t>1, with $\boldsymbol{\alpha}_1 = \boldsymbol{a}_0\boldsymbol{E}(x_1)$. Hence, recursion.
  - Note: elements of $\boldsymbol{\alpha}_t$ are forward probabilities.

“Optimal” State Sequence?

Given: a sequence of symbols generated by an HMM.

Goal: find the path of states most likely to generate the observed sequence.

Individually Most Likely States (Local Decoding)

- For each t, we may look for $\max_i \mathrm{P}(q_t=Q_i \mid \underline{x})$

$$\mathrm{P}(q_t=Q_i \mid \underline{x}) = \mathrm{P}(\underline{x}, q_t=Q_i) \,/\, \mathrm{P}(\underline{x}) =$$

$$\frac{\mathrm{P}(x_1, \ldots, x_t, q_t=Q_i)\,\mathrm{P}(x_{t+1}, \ldots, x_n \mid q_t=Q_i)}{\Sigma_j\, \mathrm{P}(x_1, \ldots, x_t, q_t=Q_j)\,\mathrm{P}(x_{t+1}, \ldots, x_n \mid q_t=Q_j)}$$

- Thus, we can use forward-backward algorithms

Individually Most Likely States (cont'd)

- The resulting sequence of states can be problematic.

- This is because the states are optimized *individually*, without regard of the probability of occurrence of a *sequence* of states.

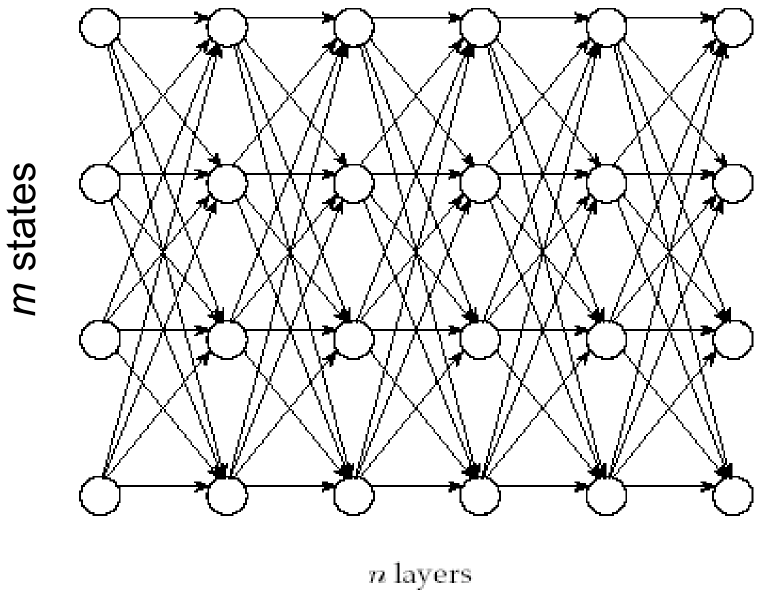
- Alternative: find a path that maximizes $P(\underline{q}|\underline{x})$ over all possible paths $\underline{q}$.

Global Decoding Problem

- **Goal:** Find an optimal hidden path of states given observations.
- **Input:** Sequence of observations $\underline{x} = x_1 \ldots x_n$ generated by an HMM $M(\Sigma, Q, A, E)$
- **Output:** A path that maximizes $P(\underline{q} \mid \underline{x})$ over all possible paths $\underline{q}$.

Building Manhattan for Global Decoding Problem

- Andrew Viterbi used the Manhattan grid model to solve the *Decoding Problem*.
- Every choice of $\underline{q}$ corresponds to a path in the graph
- The only valid direction in the graph is *eastward.*
- This graph has $m^2(n-1)$ edges, each with a weight.



Decoding Problem as Finding a Longest Path in a DAG

- The *Decoding Problem* is reduced to finding a longest (largest score) path in the *directed acyclic graph (DAG)* above.

- **Notes:** the length of the path is defined as the *product* of its edges' weights.

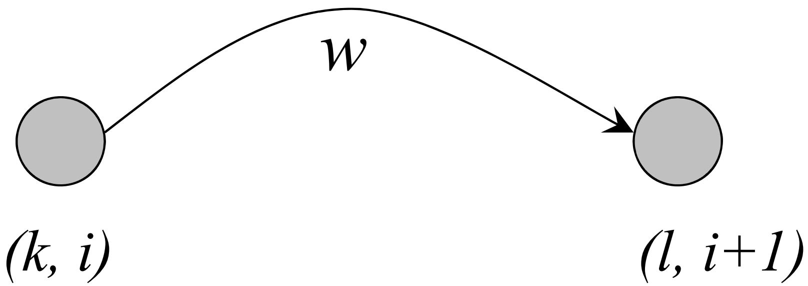
Global Decoding Problem (cont'd)

- Idea: $\max_{\underline{q}} P(\underline{q}|\underline{x}) = \max_{\underline{q}} P(\underline{q},\underline{x})/P(\underline{x}) = \max_{\underline{q}} P(\underline{q},\underline{x})$
- So, $\text{argmax}_{\underline{q}} P(\underline{q}|\underline{x}) = \text{argmax}_{\underline{q}} P(\underline{q},\underline{x})$

- Every path in the graph has the probability $P(\underline{q},\underline{x})$
- The Viterbi algorithm finds the path that maximizes $P(\underline{q},\underline{x})$ among all possible paths
  - It runs in $\boldsymbol{O(nm^2)}$ time.

Global Decoding Problem (cont'd)

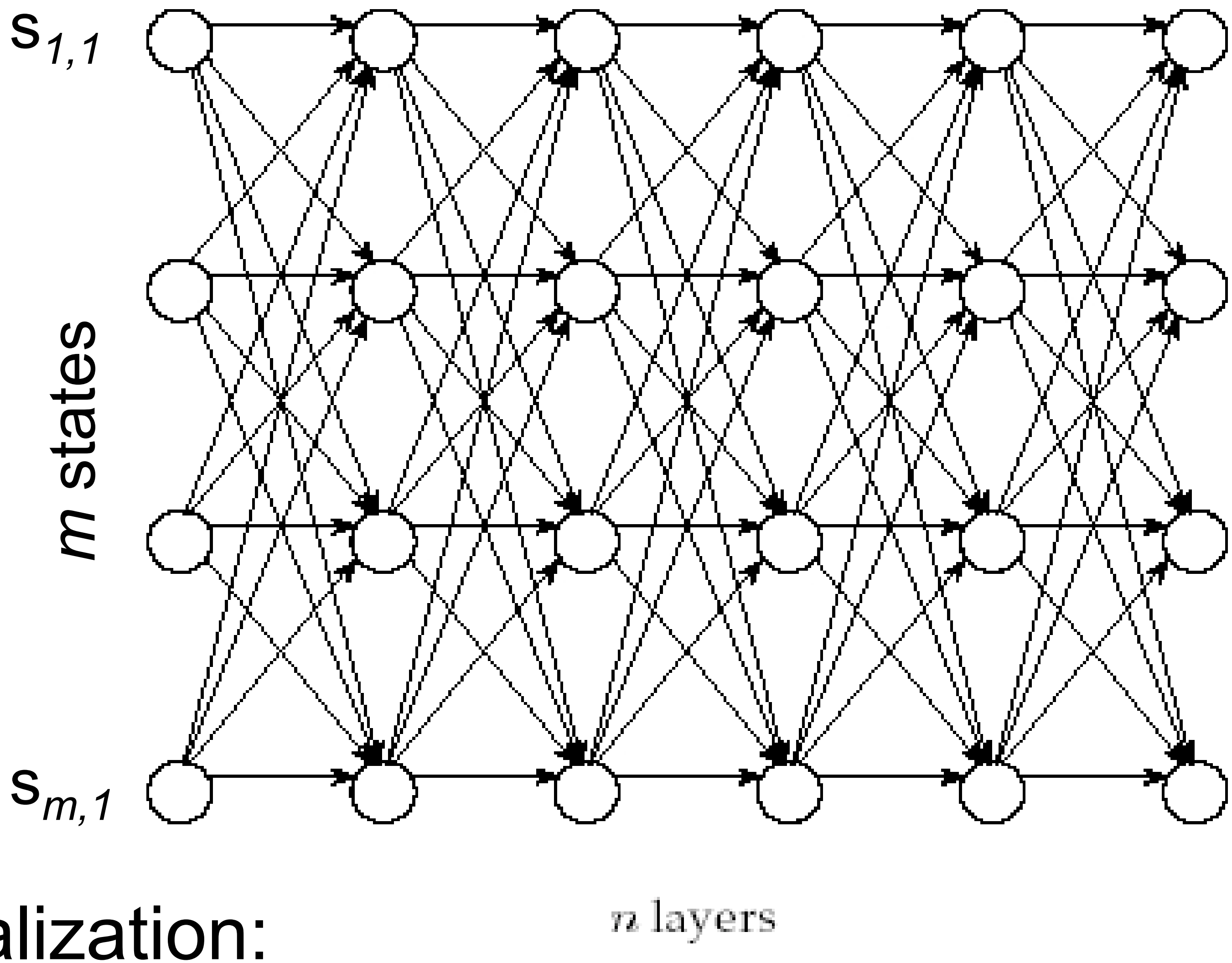
$$P(\underline{x}|\underline{q}) = a_{q_0,q_1} \cdot \Pi \{ e_{q_i}(x_i) \,.\, a_{q_i,q_{i+1}} \}$$



The weight ***w*** is given by:

$$w = e_l(x_{i+1}).\ a_{kl}$$

Edit Graph for Decoding Problem



- Initialization:

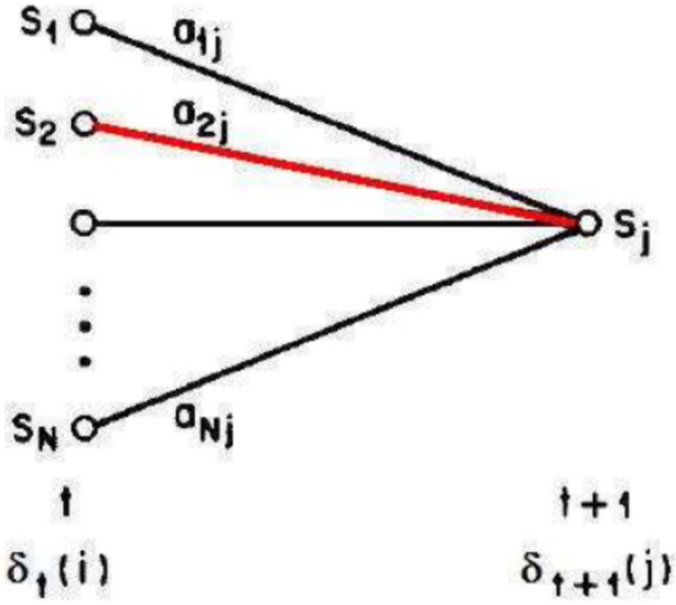
  $s_{k,1} = e_k(x_1) \cdot a_{begin,k}$

Decoding Problem and Dynamic Programming

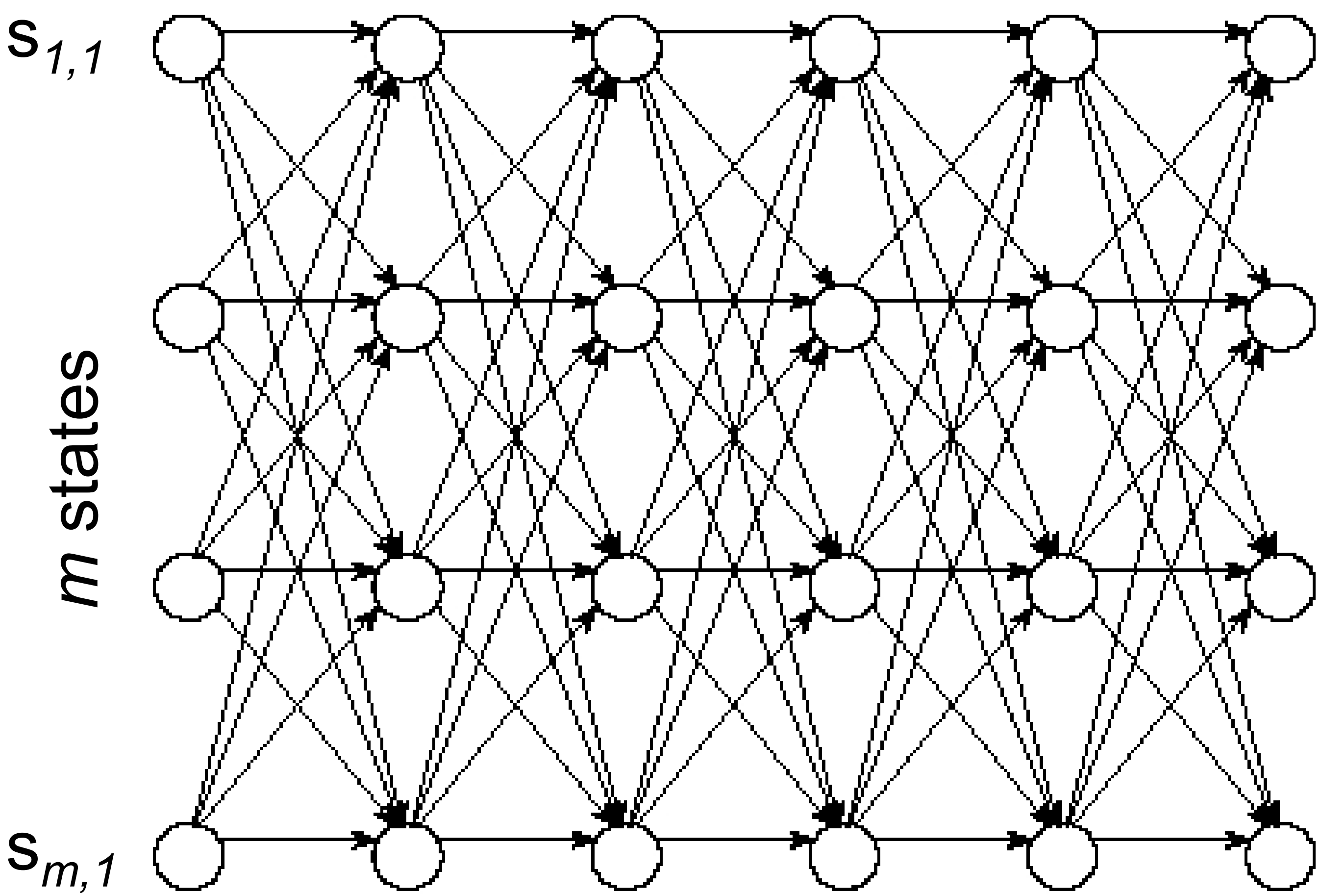
$s_{l,i+1}$ = $\max_{k \in Q}\{s_{k,i} \cdot$ weight of edge between (k,i) and $(l,i+1)\}=$

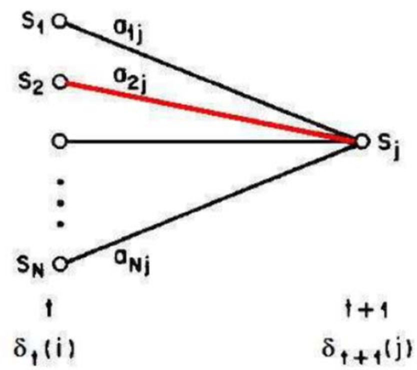
$\max_{k \in Q}\{s_{k,i} \cdot a_{kl} \cdot e_l\,(x_{i+1})\}=$

$e_l\,(x_{i+1}) \cdot \max_{k \in Q}\{s_{k,i} \cdot a_{kl}\}$



Edit Graph for Decoding Problem



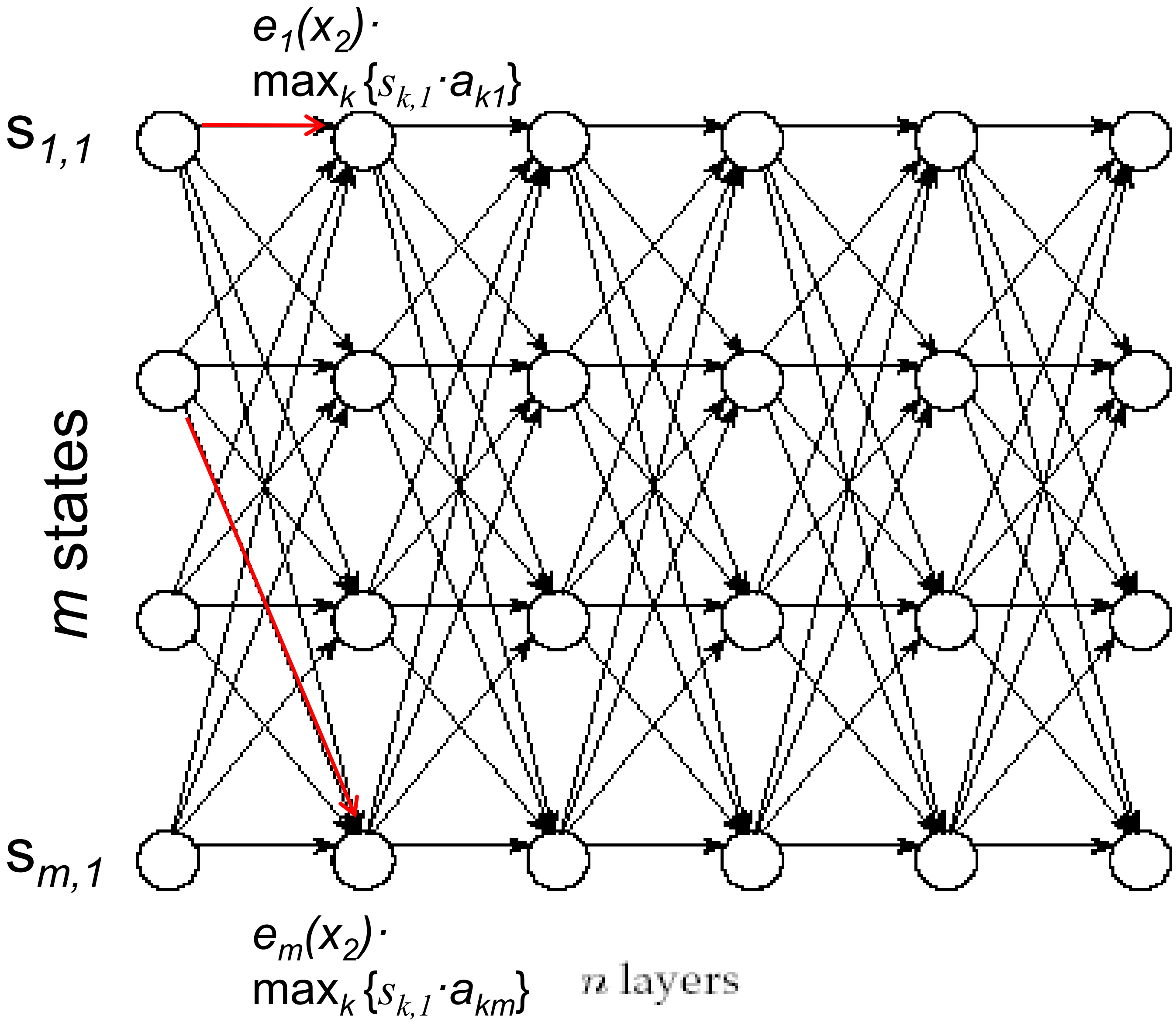


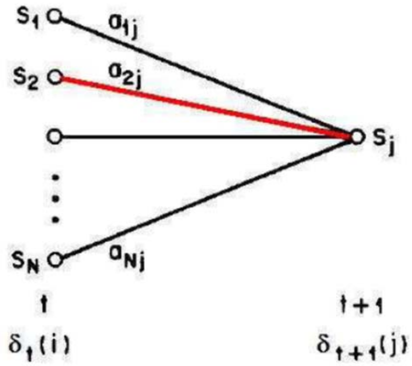
$s_{l,i+1} = \max_{k \in Q}\{s_{k,i} \cdot$ weight of edge between (k,i) and $(l,i+1)\}=$

$\max_{k \in Q}\{s_{k,i} \cdot a_{kl} \cdot e_l(x_{i+1})\}=$

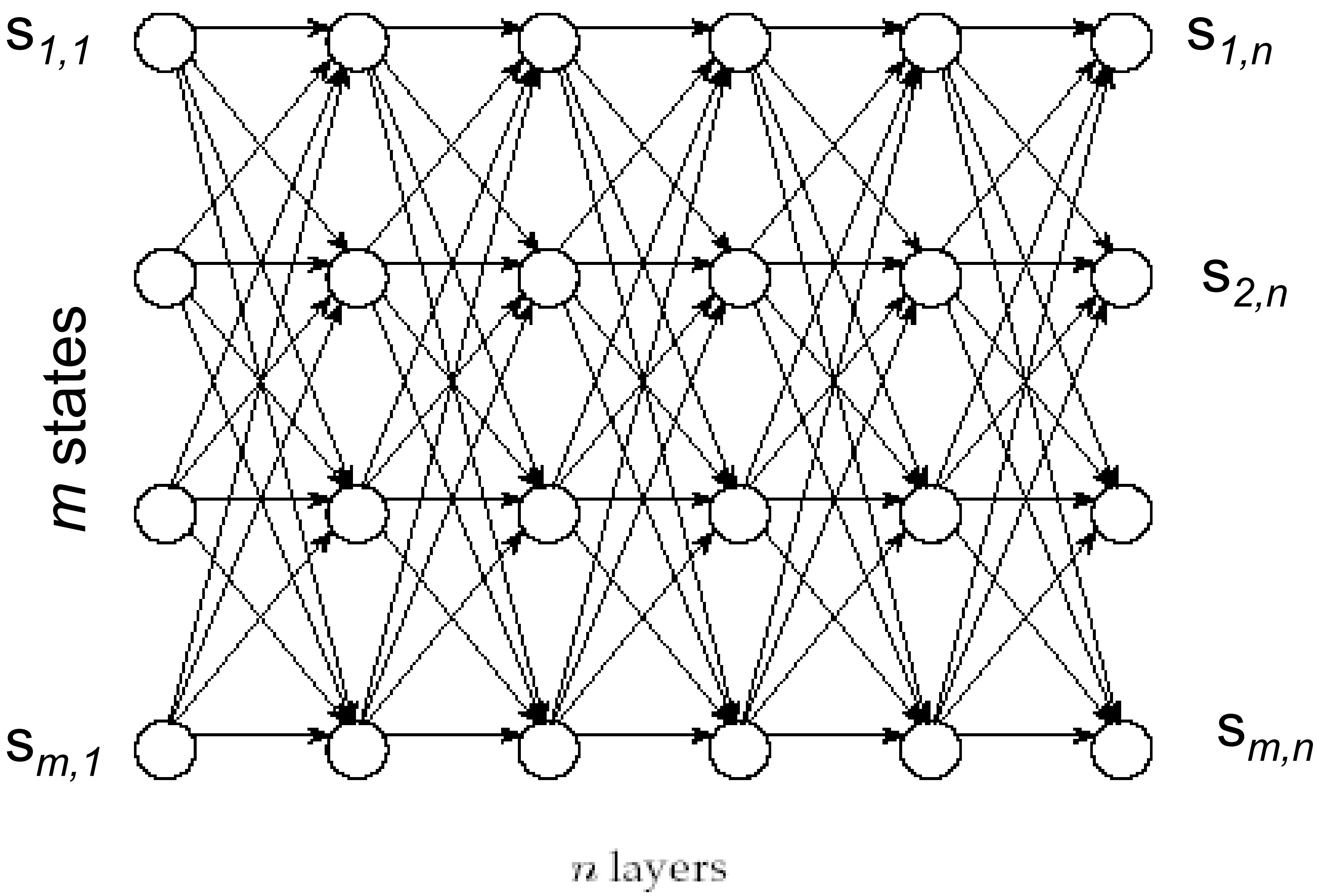
$e_l(x_{i+1}) \cdot \max_{k \in Q}\{s_{k,i} \cdot a_{kl}\}$

Edit Graph for Decoding Problem

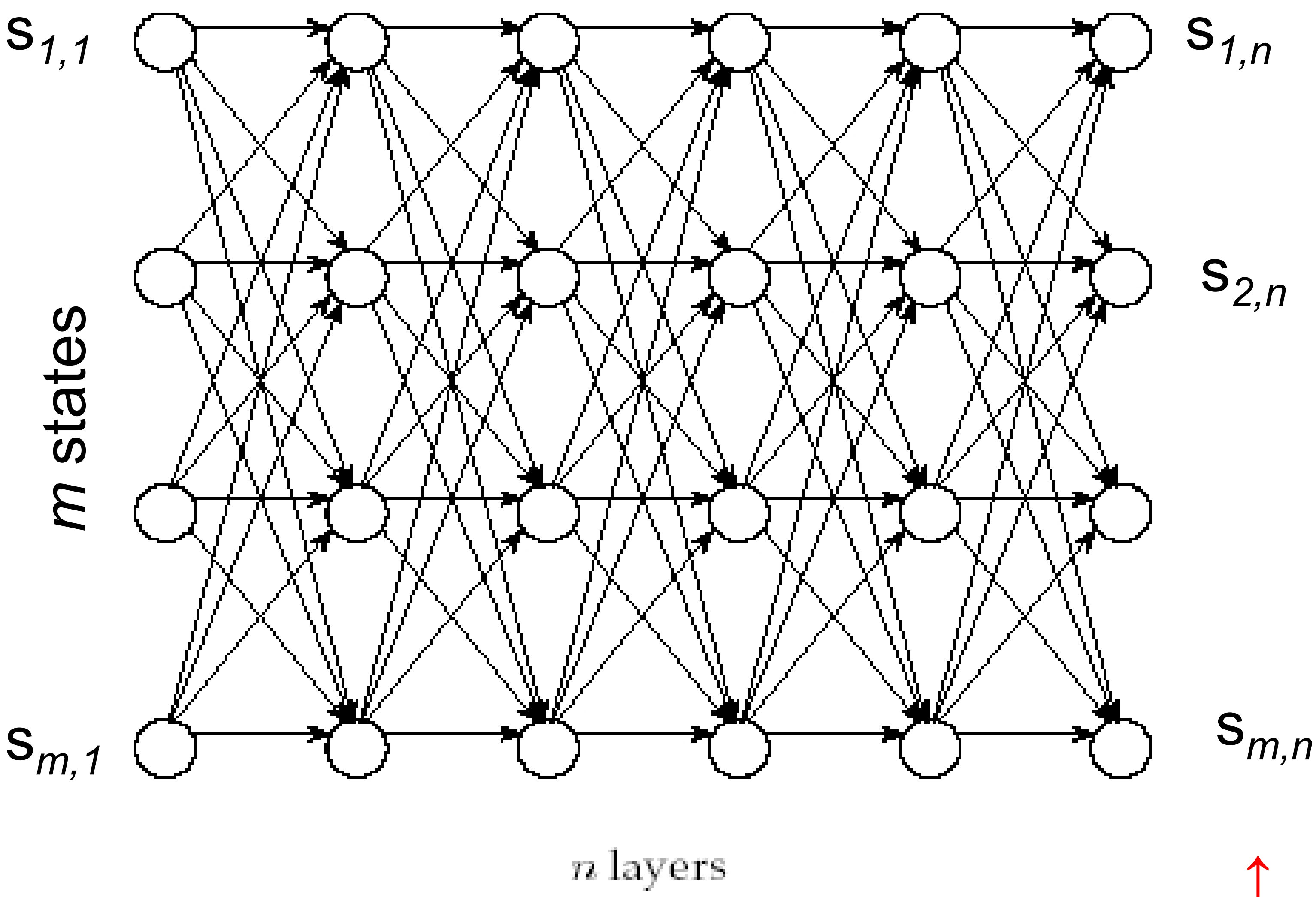




Edit Graph for Decoding Problem

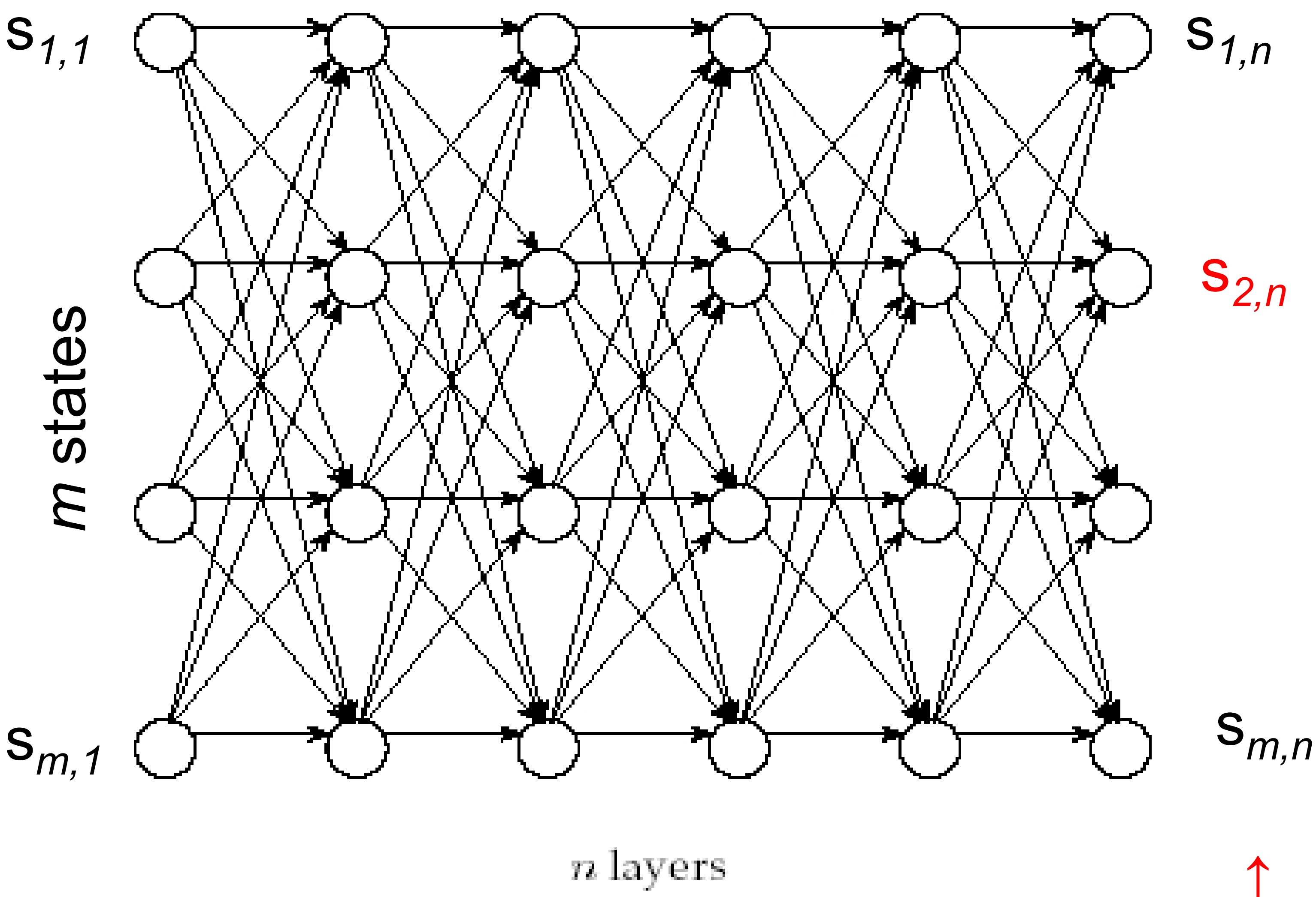


Edit Graph for Decoding Problem



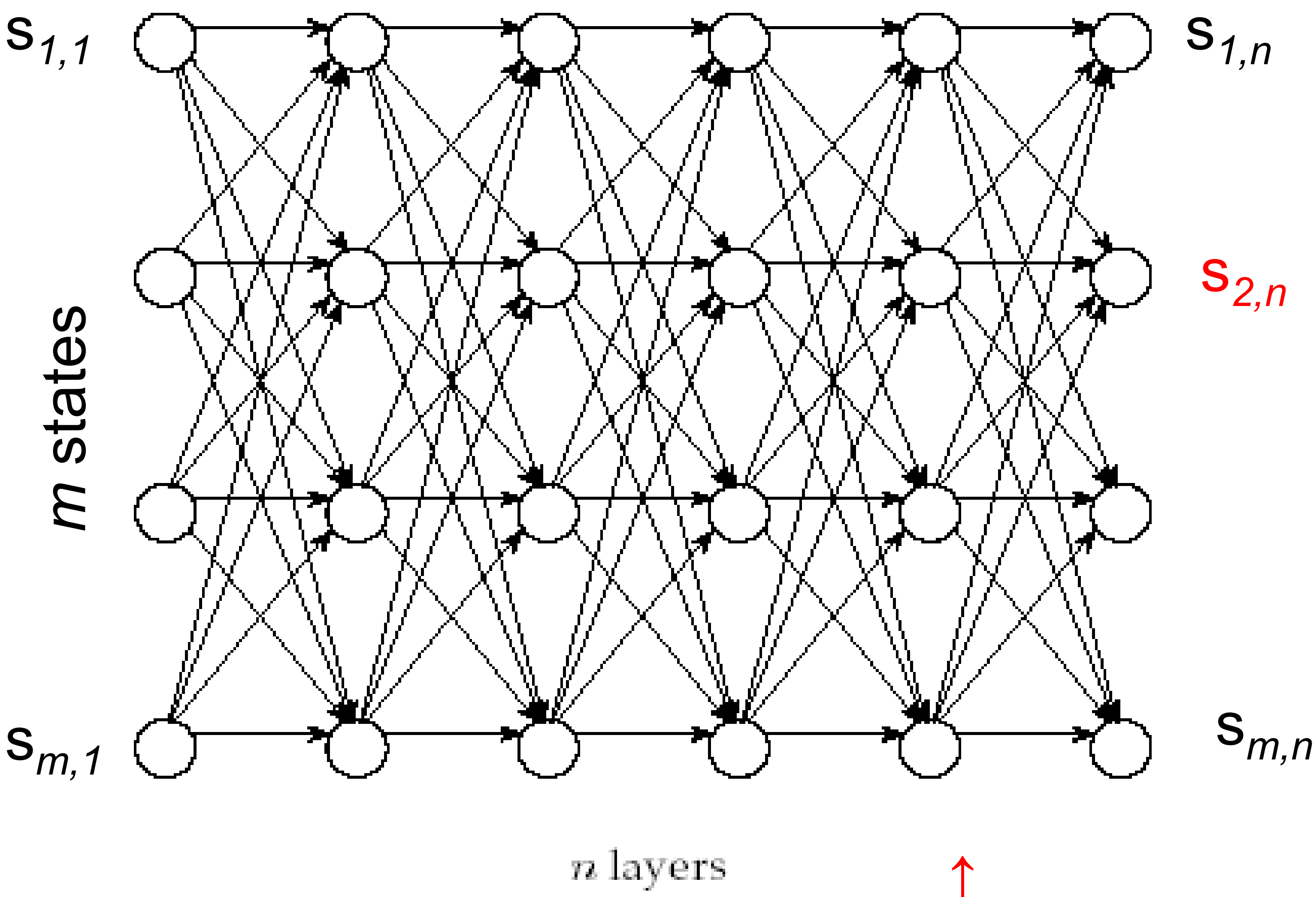
Find max

Edit Graph for Decoding Problem



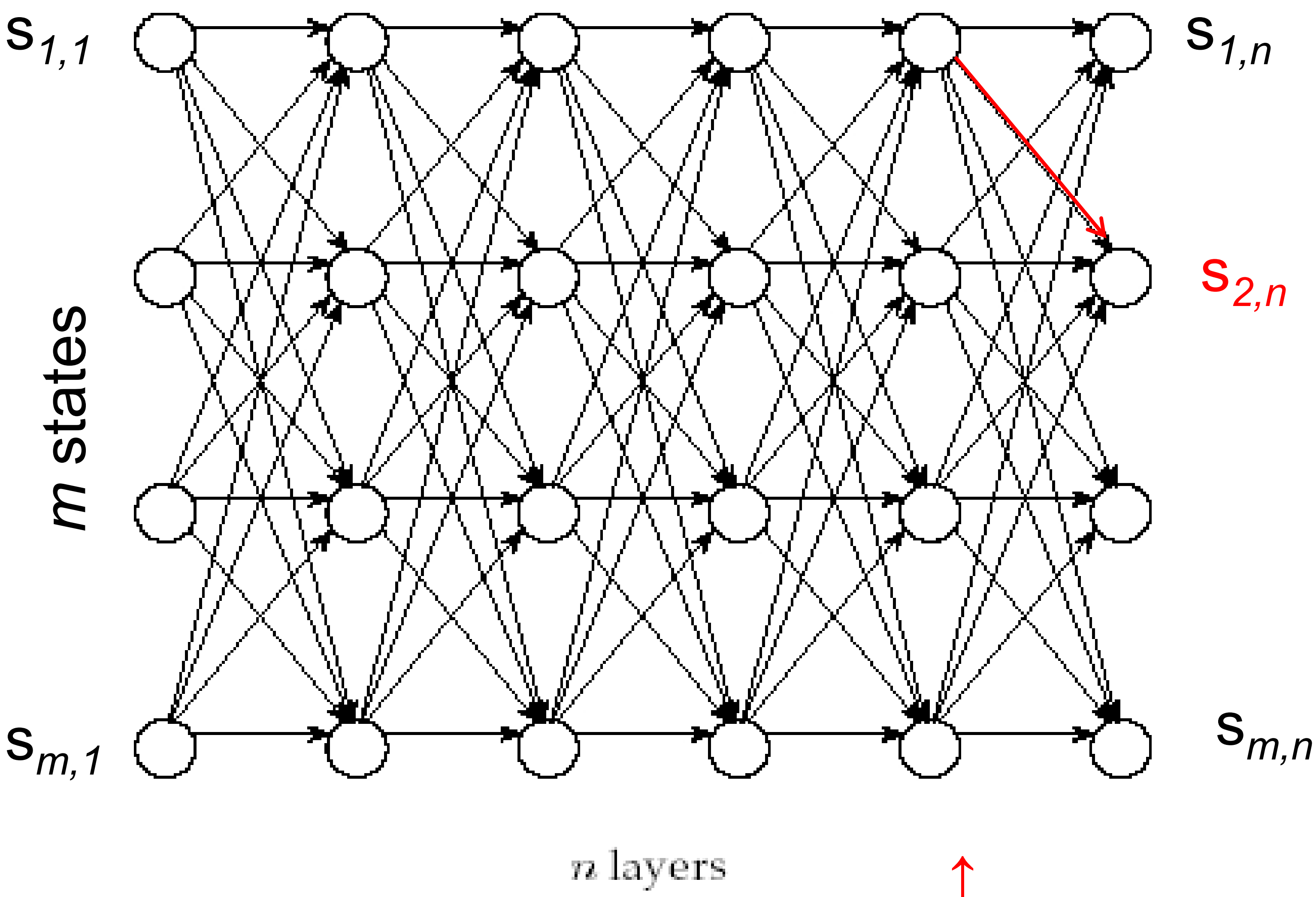
Find max

Edit Graph for Decoding Problem



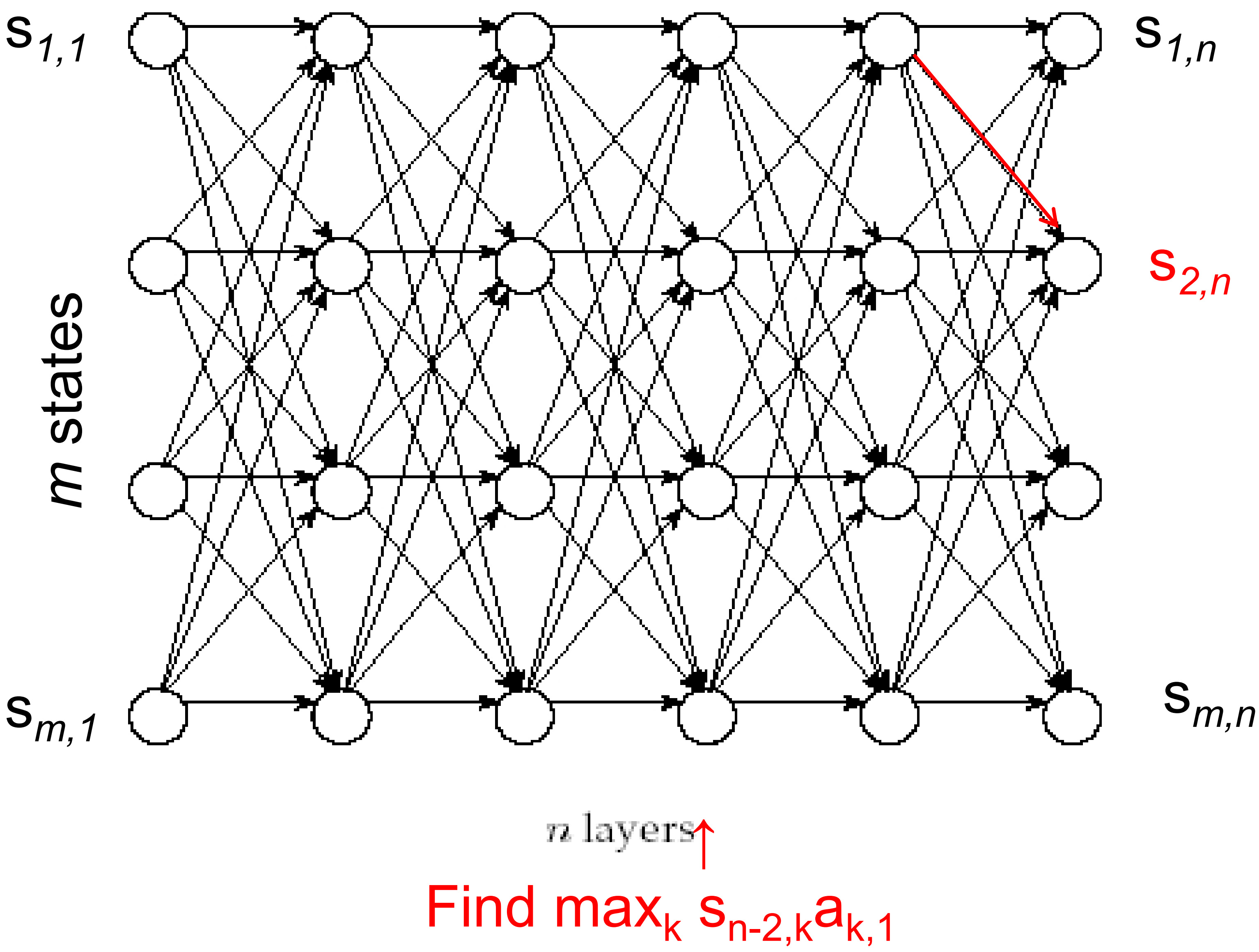
Find $\max_k s_{n-1,k} a_{k,2}$

Edit Graph for Decoding Problem

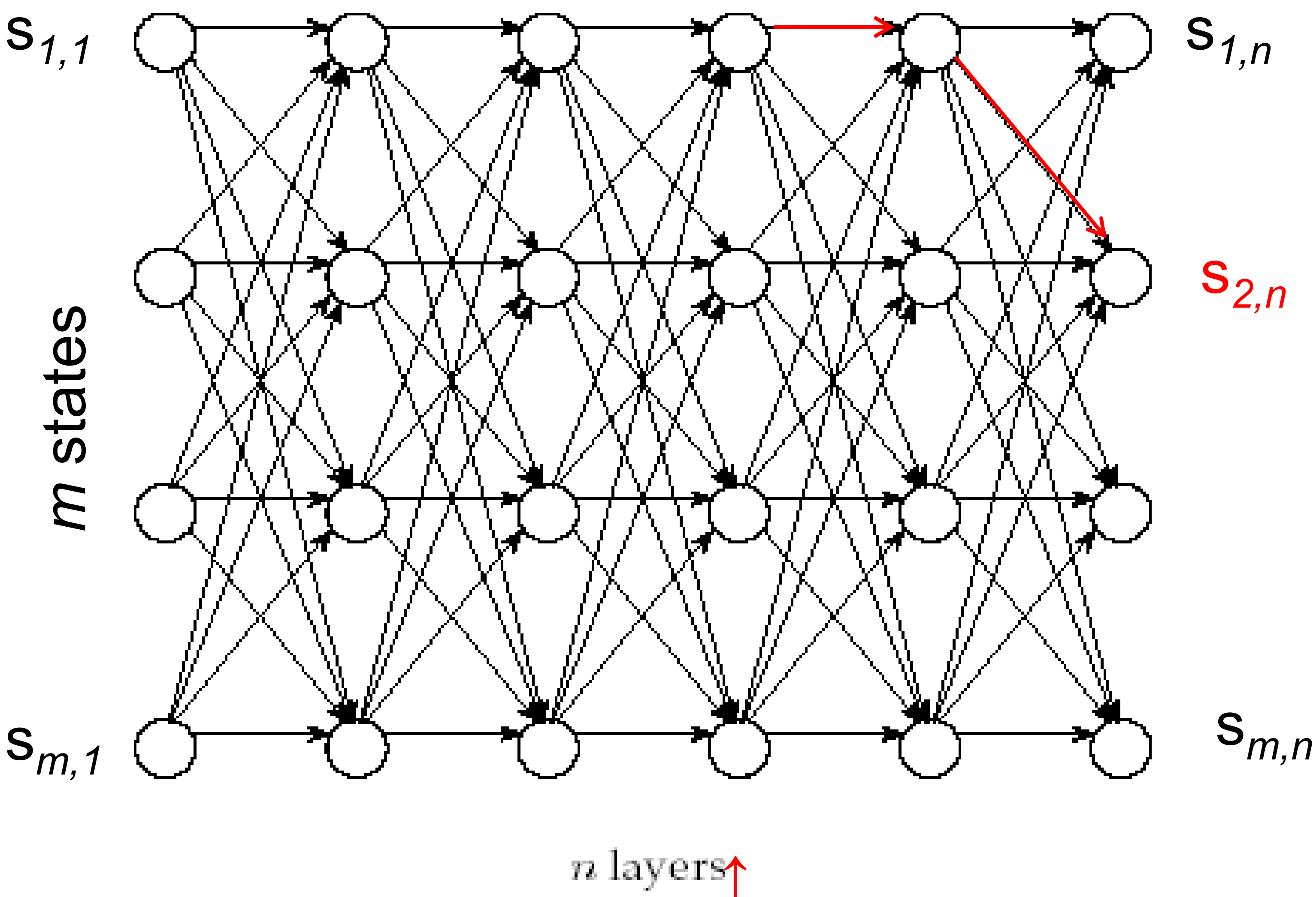


Find $\max_k s_{n-1,k} a_{k,2}$

Edit Graph for Decoding Problem

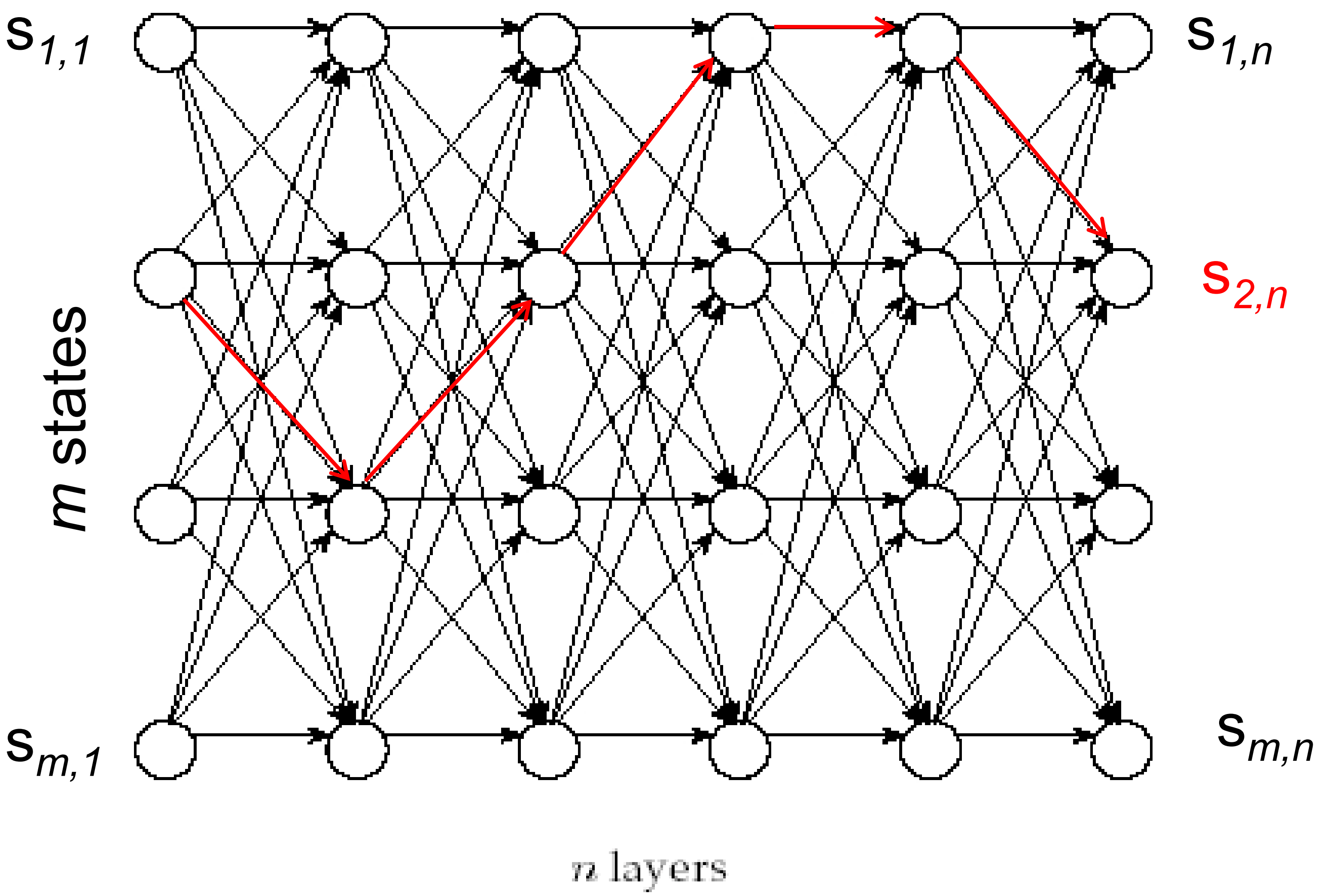


Edit Graph for Decoding Problem



Find $\max_k s_{n-2,k} a_{k,1}$

Edit Graph for Decoding Problem



Viterbi Algorithm

- The value of the product can become extremely small, which leads to overflowing.
- To avoid overflowing, use log value instead.

$$s_{l,i+1} = \log e_l(x_{i+1}) + \max_{k \in Q}\{s_{k,i} + \log(a_{kl})\}$$

- Note similarity to forward algorithm:

$$P(x_1, \ldots, x_{t+1}, q_{t+1}=Q_k) = e_{Q_k}(x_{t+1}) \cdot \Sigma_j P(x_1, \ldots, x_t, q_t=Q_j)\, a_{Q_j, Q_k}$$

$$s_{l,i+1} = e_l\,(x_{i+1}) \cdot \max_{k \in Q}\{s_{k,i} \cdot a_{kl}\}$$

State Prediction

- Consider $P(q_t = Q_i \mid \underline{x})$

- We get

$1 \leq t < n$: $P(x_1, \ldots, x_t, q_t = Q_i) P(x_{t+1}, \ldots, x_n \mid q_t = Q_i) / P(\underline{x})$

$t = n$: $P(x_1, \ldots, x_n, q_n = Q_i) / P(\underline{x})$

$t > n$: $\boldsymbol{\alpha}_n [\boldsymbol{A}^{t-n}]_{(.,i)} / P(\underline{x})$

- For $t > n$, we can write $P(q_{n+h} = Q_i \mid \underline{x}) = \boldsymbol{\alpha}_n [\boldsymbol{A}^h]_{(.,i)} / P(\underline{x})$
  - with $h \rightarrow +\infty$, $\boldsymbol{\alpha}_n \boldsymbol{A}^h / P(\underline{x}) \rightarrow$ stationary distribution

HMM Parameter Estimation

- So far, we have assumed that the transition and emission probabilities are known.

- However, in most HMM applications, the probabilities are not known. It's very hard to estimate the probabilities.

HMM Parameter Estimation (cont'd)

- Let Θ be a vector combining the unknown transition and emission probabilities.
- Given training sequences $\underline{x}^1, \ldots, \underline{x}^M$, let $P(\underline{x}|\Theta)$ be the prob. of $\underline{x}$ given the assignment of parameters Θ.
- Then our goal is to find MLE:

$$max_{\Theta} \quad \Pi_{j=1} P(\underline{x}^i|\Theta)$$

HMM Parameter Estimation (cont'd)

- Issue: constrained parametrization, as $\boldsymbol{A}\mathbf{1}^{\mathsf{T}}=\mathbf{1}^{\mathsf{T}}$
  - row entries sum to one
- Consider "working" parameters ϑ_{ij} for $i \neq j$
- Let $a_{ij}=\exp(\vartheta_{ij})/\{1+\sum_k \exp(\vartheta_{ik})\}$ for $i \neq j$ and $a_{ii}= 1/\{1+\sum_k \exp(\vartheta_{ik})\}$
- Additional contraints for, e.g., emission-distribution parameters can be handled by transformations
  - Poisson rates: use $\ln(\lambda)$

HMM Parameter Estimation (cont'd)

- In matrix notation, $P(\underline{x}) = \boldsymbol{a}_0\boldsymbol{E}(x_1)\boldsymbol{A}\boldsymbol{E}(x_2) \cdot \ldots \cdot \boldsymbol{A}\boldsymbol{E}(x_n)\mathbf{1}^{\mathsf{T}} = \boldsymbol{\alpha}_n\mathbf{1}^{\mathsf{T}}$ with $\boldsymbol{\alpha}_1 = \boldsymbol{a}_0\boldsymbol{E}(x_1)$ and recursion $\boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t-1}\boldsymbol{A}\boldsymbol{E}(x_t)$ for $t>1$.

- Hence, one could directly maximize $\Pi_{j=1}\, P(\underline{x}^i|\Theta)$

- Issue: numerical underflow

- Solution: log-tansformation and scaling:
  for stationary $\boldsymbol{a}_0$, $\ln P(\underline{x}) = \sum_{t=1} \ln[\{\boldsymbol{\alpha}_{t-1}/(\boldsymbol{\alpha}_{t-1}\mathbf{1}^{\mathsf{T}})\}\boldsymbol{A}\boldsymbol{E}(x_t)\mathbf{1}^{\mathsf{T}}]$
  for non-stationary, $\ln\{\boldsymbol{a}_0\,\boldsymbol{E}(x_1)\mathbf{1}^{\mathsf{T}}\}+\sum_{t=2}\ln[\{\boldsymbol{\alpha}_{t-1}/(\boldsymbol{\alpha}_{t-1}\mathbf{1}^{\mathsf{T}})\}\boldsymbol{A}\boldsymbol{E}(x_t)\mathbf{1}^{\mathsf{T}}]$

HMM Parameter Estimation (cont'd)

- Assume state paths are known for the training set.
- Maximum likelihood estimators for a_{ij} and $e_i(s)$ are

  $a_{ij}=A_{ij} / \Sigma_{j'} A_{ij'}$ and $e_i=E_i(s) / \Sigma_{s'} E_i(s')$

  where A_{ij} and $E_i(s)$ are the numbers of times a particular transition and emission are used in training sequences.

Baum-Welch Algorithm

- Assume further that state paths are not known.

- B-W is a version of the E-M algorithm.
  - E-step: A_{ij} and $E_i(s)$ are estimated using current estimates of a_{ij} and $e_i(s)$.
  - M-step: Using the estimates, a_{ij} and $e_i(s)$ are updated:

    $a_{ij}=A_{ij} / \Sigma_{j'} A_{ij'}$ and $e_i=E_i(s)/ \Sigma_{s'} E_i(s')$

Baum-Welch Algorithm (cont'd)

- The probability that transition $Q_i \rightarrow Q_j$ is used at position t in $\underline{x}$ is
  $P(q_t=Q_i\,,\ q_{t+1}=Q_j \mid \underline{x}) =$
  $P(x_1, \ldots, x_t, q_t=Q_i)a_{ij}e_j(x_{t+1})P(x_{t+2}, \ldots, x_n \mid q_{t+1}=Q_j)\,/\,P(\underline{x})$

- The expected number of times transition $Q_i \rightarrow Q_j$ is used in the training sequences is
  $A_{ij}=\Sigma^M_{k=1}\Sigma_t P(x^k_1, \ldots, x^k_t, q_t=Q_i)a_{ij}e_j(x^k_{t+1})P(x^k_{t+2}, \ldots, x^k_n \mid q_{t+1}=Q_j)\,/\,P(\underline{x}^k)$

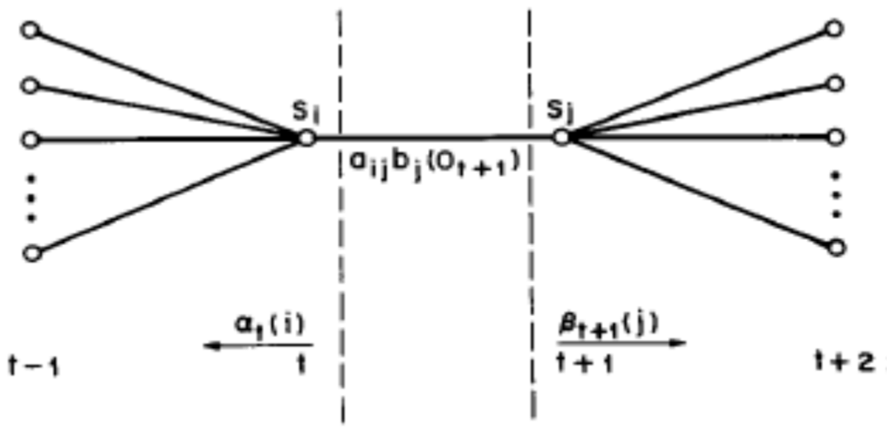


Fig. 6. Illustration of the sequence of operations required for the computation of the joint event that the system is in state S_i at time t and state S_j at time $t+1$.

Baum-Welch Algorithm (cont'd)

- Similarly, the expected number of times letter *s* is emitted in state Q_i is

$E_i(s) = \Sigma_k \Sigma_{t'} P(x^k{}_1, \ldots, x^k{}_{t'}, q_{t'}=Q_i)P(x^k{}_{t'+1}, \ldots, x^k{}_n \mid q_{t'}=Q_i)/P(\underline{x}^k)$

where the sum is over positions *t'*, at which *s* was observed.

- The expected number of times sequence starts in state Q_i is

$a_{begin,i} = \Sigma_k P(x^k{}_1, q_1=Q_i)P(x^k{}_2, \ldots, x^k{}_n \mid q_1=Q_i)/P(\underline{x}^k)$

Baum-Welch Algorithm (cont'd)

- Initialize a_{ij} and $e_i(s)$, A_{ij}, and $E_i(s)$.
- E-step:
  - for each training sequence k
    - use the forward algorithm to compute $P(x^k_1, \ldots, x^k_t, q_t=Q_i)$
    - use the backward algorithm to compute $P(x^k_t, \ldots, x^k_n \mid q_t=Q_j)$
  - calculate the expected values A_{ij} and $E_i(s)$
- M-step: compute the updated values of a_{ij} and $e_i(s)$
- Iterate until convergence

Viterbi Training

- A modification of the B-W algorithm.
- Given a_{ij} and $e_i(s)$, most likely paths are found for the training sequences, and used to re-compute A_{ij} and $E_i(s)$.
- B-W maximizes $P(\underline{x}^1, \ldots, \underline{x}^M \mid \Theta)$
- Viterbi training maximizes $P(\underline{x}^1, \ldots, \underline{x}^M \mid \Theta, \underline{q}(\underline{x}^1), \ldots, \underline{q}(\underline{x}^M))$

Baum-Welch/Viterbi Training (cont'd)

- Caution:

  If there is a small group of sequences in the training set which are highly similar, the model will overspecialize to the small group

  $\Rightarrow$ use a method of sequence weighting

HMM Estimation Issues

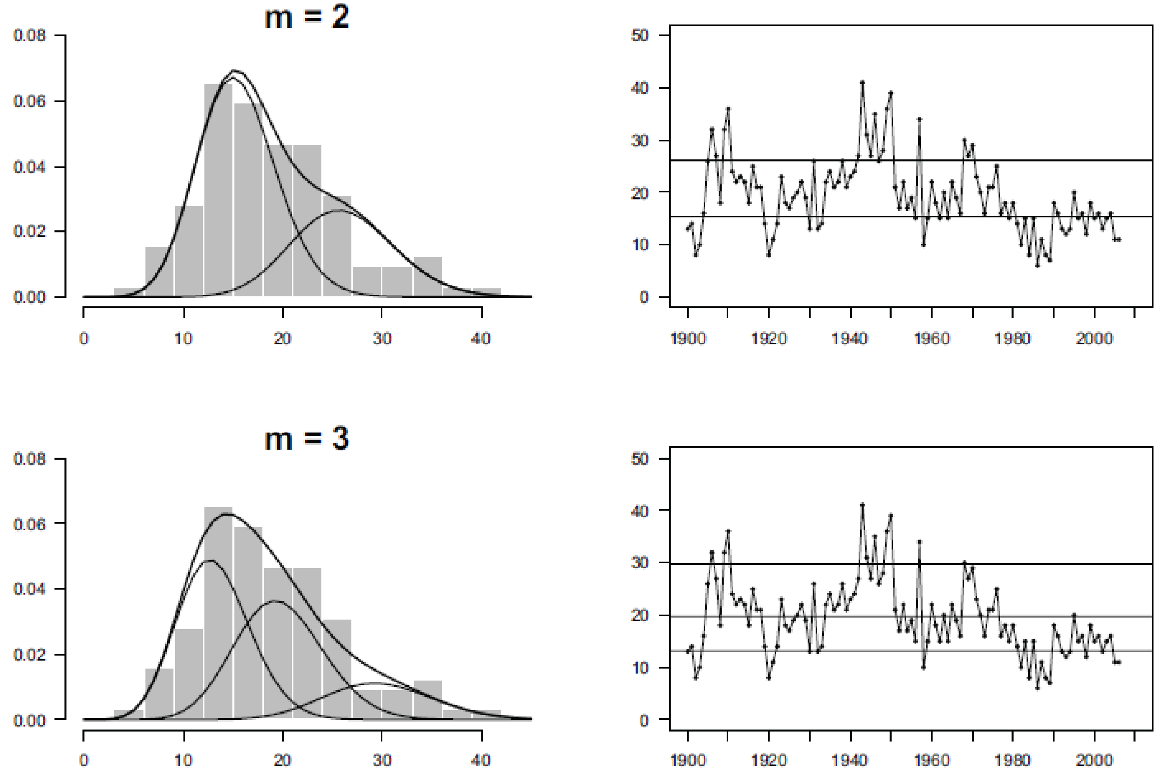
- Local maxima can occur
- Use various starting values and check stability of solutions
- For the emission-parameters, use observed quantiles
  - Poisson HMM: if 3 states, use quartiles of the count distribution
- Transition probabilities: less trivial
  - Uniform off-diagonal probabilities (all 0.01)?

REF: Zucchini & MacDonald (2009)

Earthquakes data

- Direct (stationary) likelihood maximization

	mean	variance
observations:	19.364	51.573
'one-state HMM':	19.364	19.364
two-state HMM:	19.086	44.523
three-state HMM:	18.322	50.709
four-state HMM:	18.021	49.837



REF: Zucchini & MacDonald (2009)

Earthquakes data

- EM

Table 4.1 *Two-state model for earthquakes, fitted by EM.*

Iteration	γ_{12}	γ_{21}	λ_1	λ_2	δ_1	$-l$
0	0.100000	0.10000	10.000	30.000	0.50000	413.27542
1	0.138816	0.11622	13.742	24.169	0.99963	343.76023
2	0.115510	0.10079	14.090	24.061	1.00000	343.13618
30	0.071653	0.11895	15.419	26.014	1.00000	341.87871
50	0.071626	0.11903	15.421	26.018	1.00000	341.87870
convergence	0.071626	0.11903	15.421	26.018	1.00000	341.87870
stationary model	0.065961	0.12851	15.472	26.125	0.66082	342.31827

Table 4.2 *Three-state model for earthquakes, fitted by EM.*

Iteration	λ_1	λ_2	λ_3	δ_1	δ_2	$-l$
0	10.000	20.000	30.000	0.33333	0.33333	342.90781
1	11.699	19.030	29.741	0.92471	0.07487	332.12143
2	12.265	19.078	29.581	0.99588	0.00412	330.63689
30	13.134	19.713	29.710	1.00000	0.00000	328.52748
convergence	13.134	19.713	29.710	1.00000	0.00000	328.52748
stationary model	13.146	19.721	29.714	0.44364	0.40450	329.46028

REF: Zucchini & MacDonald (2009)

Earthquakes data

- EM

- Three-state model with initial distribution (1,0,0), fitted by EM:

$$\mathbf{\Gamma} = \begin{pmatrix} 0.9393 & 0.0321 & 0.0286 \\ 0.0404 & 0.9064 & 0.0532 \\ 0.0000 & 0.1903 & 0.8097 \end{pmatrix},$$

$$\boldsymbol{\lambda} = (13.134, 19.713, 29.710).$$

- Three-state model based on stationary Markov chain, fitted by direct numerical maximization:

$$\mathbf{\Gamma} = \begin{pmatrix} 0.9546 & 0.0244 & 0.0209 \\ 0.0498 & 0.8994 & 0.0509 \\ 0.0000 & 0.1966 & 0.8034 \end{pmatrix},$$

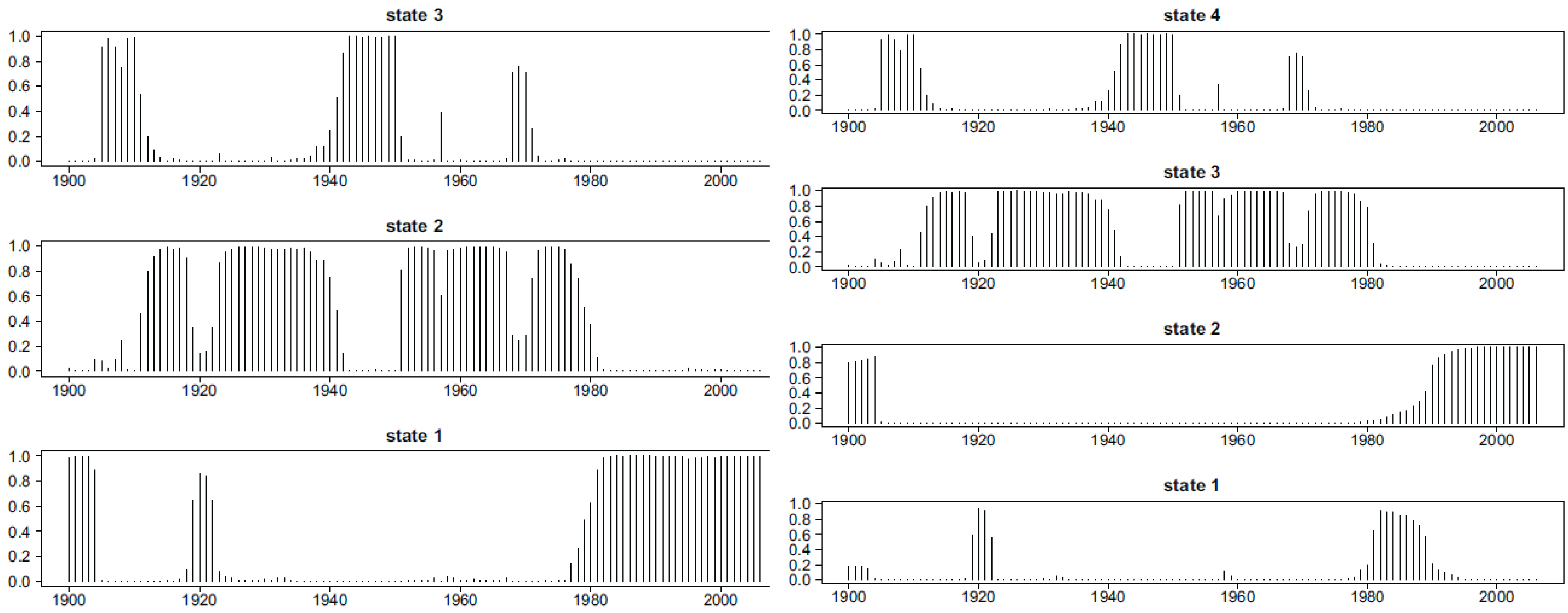
$$\boldsymbol{\delta} = (0.4436, 0.4045, 0.1519),$$

$$\text{and } \boldsymbol{\lambda} = (13.146, 19.721, 29.714).$$

REF: Zucchini & MacDonald (2009)

Earthquakes data

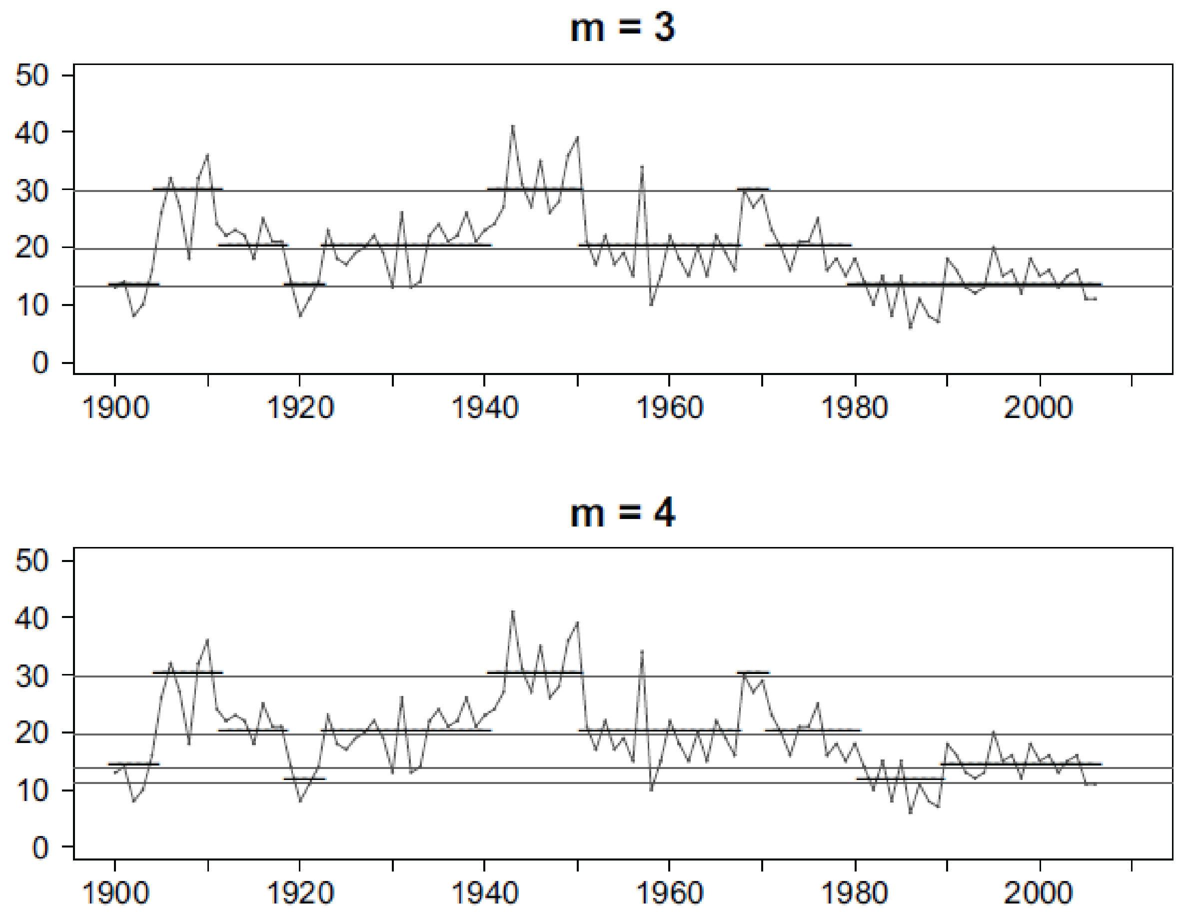
- Most likely states (non-stationary)



REF: Zucchini & MacDonald (2009)

Earthquakes data

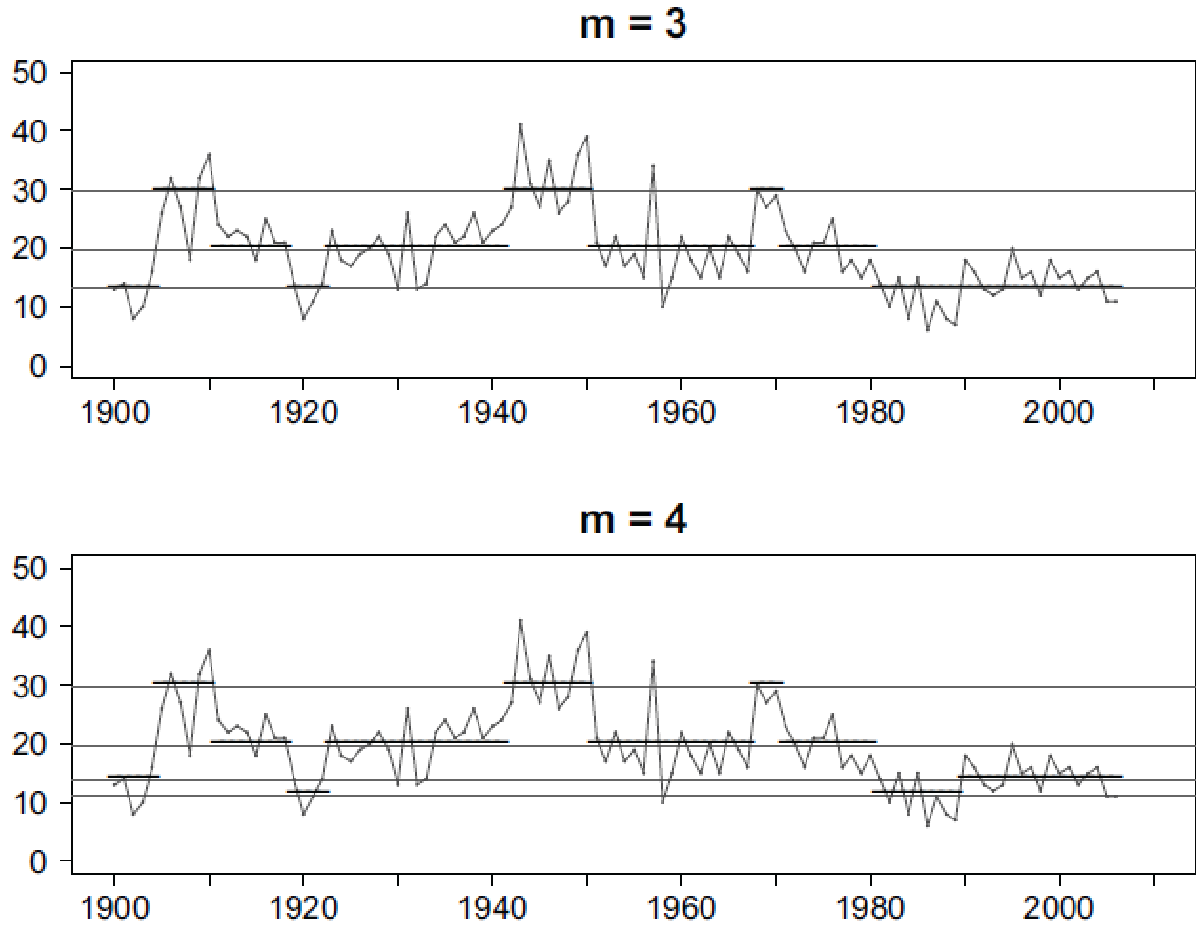
- Local decoding
- Four-state HMM “splits” the “lowest” state
  - the state visited only for 1919-1922 & 1981-1989



REF: Zucchini & MacDonald (2009)

Earthquakes data

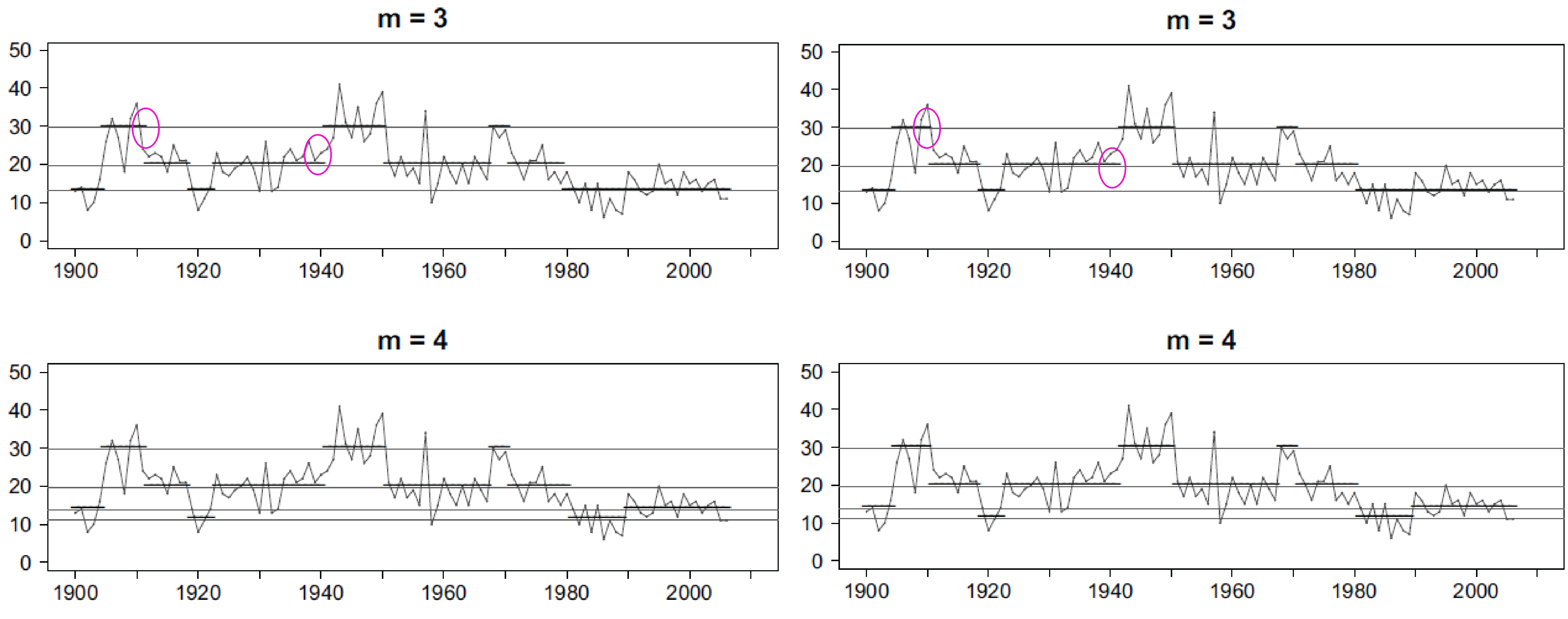
- Global decoding (Viterbi)
- Four-state HMM "splits" the "lowest" state
  - the state visited only for 1919-1922 & 1981-1989



REF: Zucchini & MacDonald (2009)

Earthquakes data

- Local and global (Viterbi) decoding
- Difference: 1911 & 1941



REF: Zucchini & MacDonald (2009)

Earthquakes data

- State predictions

year	2007	2008	2009	2016	2026	2036
state=1	0.951	0.909	0.871	0.674	0.538	0.482
2	0.028	0.053	0.077	0.220	0.328	0.373
3	0.021	0.038	0.052	0.107	0.134	0.145

Table 4.2 *Three-state model for earthquakes, fitted by EM.*

Iteration	λ_1	λ_2	λ_3	δ_1	δ_2	$-l$
0	10.000	20.000	30.000	0.33333	0.33333	342.90781
1	11.699	19.030	29.741	0.92471	0.07487	332.12143
2	12.265	19.078	29.581	0.99588	0.00412	330.63689
30	13.134	19.713	29.710	1.00000	0.00000	328.52748
convergence	13.134	19.713	29.710	1.00000	0.00000	328.52748
stationary model	13.146	19.721	29.714	0.44364	0.40450	329.46028

REF: Zucchini & MacDonald (2009)

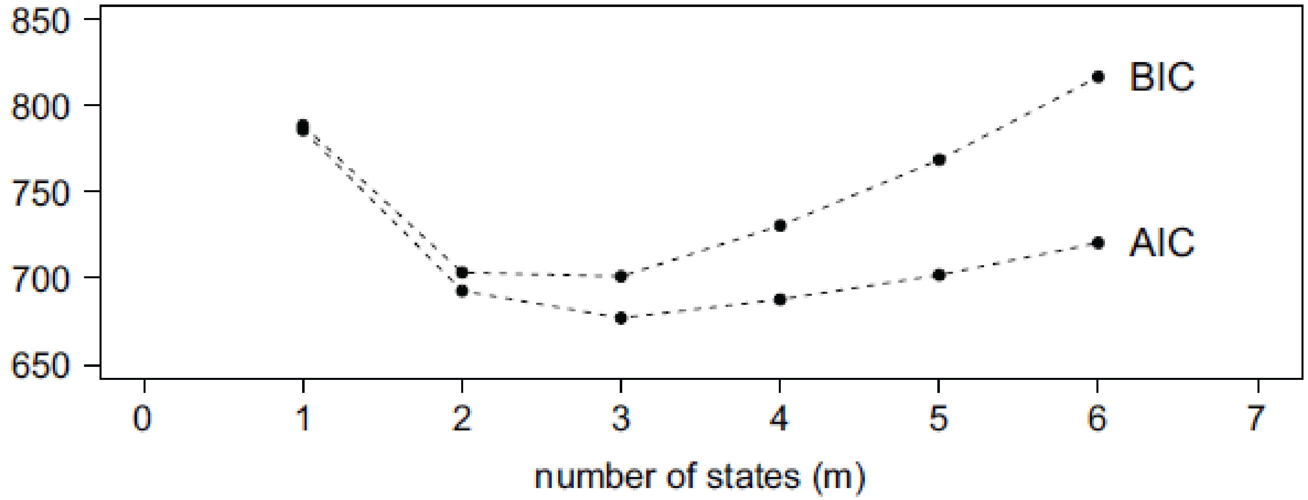
HMM Selection

- Number of states?
- AIC or BIC

REF: Zucchini & MacDonald (2009)

Earthquakes data

- Model selection



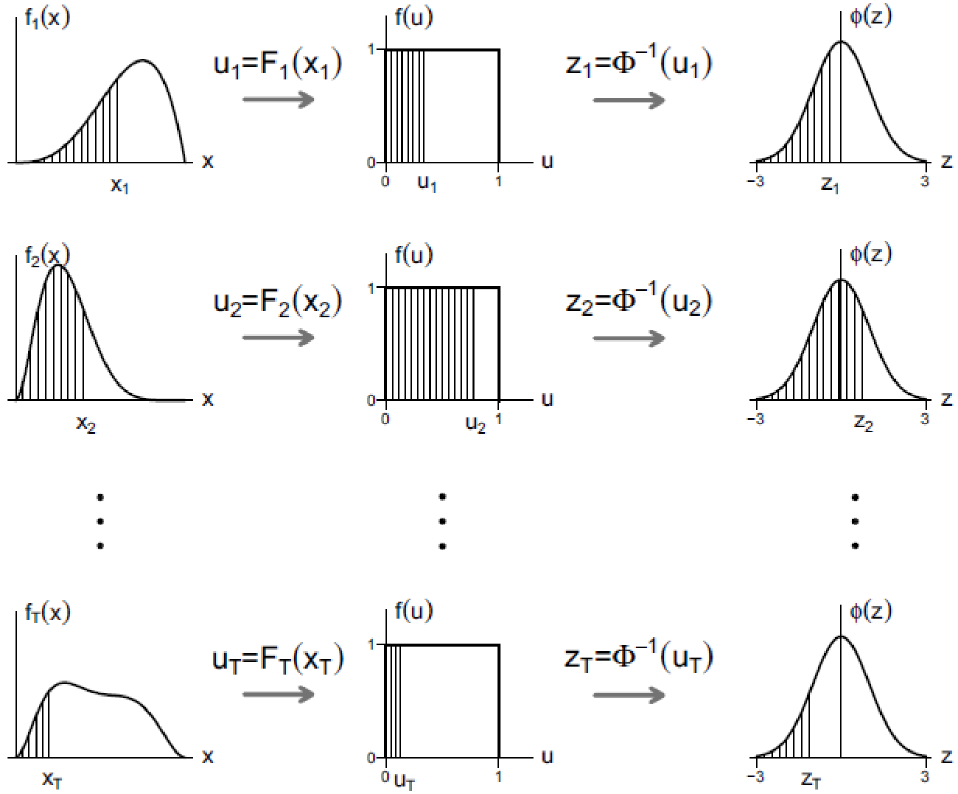
model	k	$-\log L$	AIC	BIC
'1-state HM'	1	391.9189	785.8	788.5
2-state HM	4	342.3183	692.6	703.3
3-state HM	9	329.4603	**676.9**	**701.0**
4-state HM	16	327.8316	687.7	730.4
5-state HM	25	325.9000	701.8	768.6
6-state HM	36	324.2270	720.5	816.7
indep. mixture (2)	3	360.3690	726.7	734.8
indep. mixture (3)	5	356.8489	723.7	737.1
indep. mixture (4)	7	356.7337	727.5	746.2

Not a reasonable choice
Multimodal likelihoood

REF: Zucchini & MacDonald (2009)

HMM Diagnostics

- *Uniform pseudo-residuals*: r_t = P($X_t \le x_t$) = $F_t(x_t)$ ~ U(0,1)
  - If the model X_t ~ F_t, where X_t continuous, is correct
  - Not easy to detect outliers (0.97 and 0.999)
- *Normal pseudo-residuals*: $z_t = \Phi^{-1}\{F_t(x_t)\}$ ~ N(0,1)
  - z_t =0 if x_t = median
  - Easier to detect outliers



REF: Zucchini & MacDonald (2009)

HMM Diagnostics

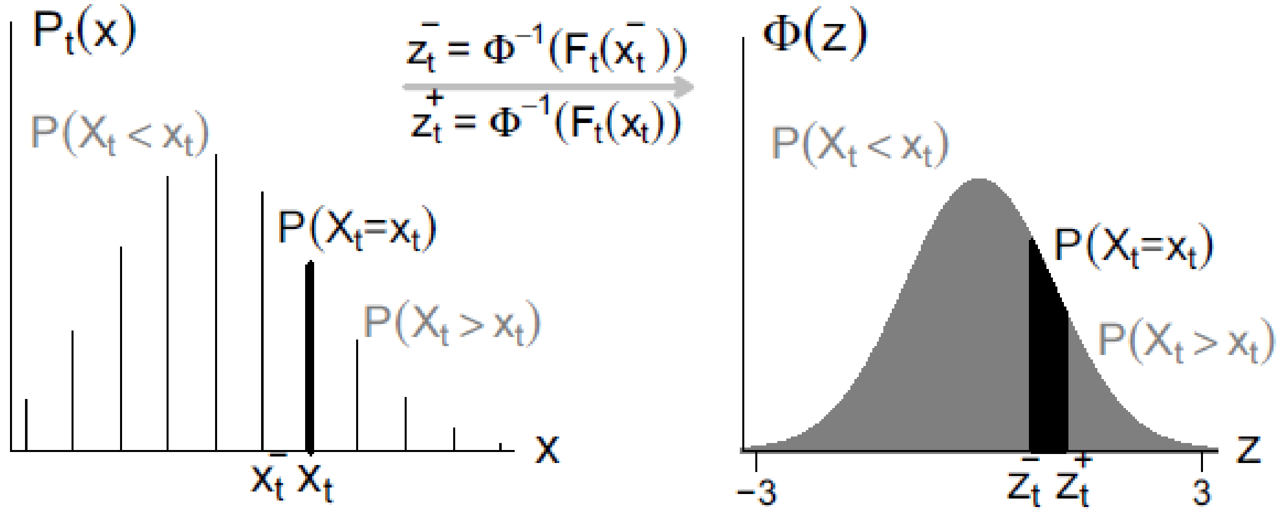
- *Uniform pseudo-residual segments* for discrete X_t:

$$[r_t^-, r_t^+] = [F_t(x_t^-), F_t(x_t)]$$

  - x_t^- - largest realization of X_t strictly less than x_t
- *Normal pseudo-residual segments* for discrete X_t:

$$[z_t^-, z_t^+] = [\Phi^{-1}(r_t^-), \Phi^{-1}(r_t^+)]$$

- For a Q-Q plot, order $z_t^* = \Phi^{-1}\{(r_t^- + r_t^+)/2\}$
  - z_t^* can be used for diagnostics as well



REF: Zucchini & MacDonald (2009)

HMM Diagnostics

- *Ordinary pseudo-residuals*: $z_t = \Phi^{-1}\{P(X_t \le x_t | x_1,\ldots, x_{t-1}, x_{t+1},\ldots,x_n)\}$
  - If the HMM is correct, z_t should be N(0,1) distributed

- *Ordinary pseudo-residual segments*: $[z_t^-, z_t^+]$ with
  $z_t^- = \Phi^{-1}\{P(X_t < x_t | x_1,\ldots, x_{t-1}, x_{t+1},\ldots,x_n)\}$ and
  $z_t^+ = \Phi^{-1}\{P(X_t \le x_t | x_1,\ldots, x_{t-1}, x_{t+1},\ldots,x_n)\}$

- Note that

$$P(X_t = x \mid x_1,\ldots, x_{t-1}, x_{t+1},\ldots,x_n) = P(x_1,\ldots, x,\ldots,x_n)/P(x_1,\ldots, x_{t-1}, x_{t+1},\ldots,x_n) =$$

$$\boldsymbol{a}_0\boldsymbol{E}(x_1)\boldsymbol{B}_2\cdot\ldots\cdot\boldsymbol{B}_{t-1}\boldsymbol{A}\boldsymbol{E}(x)\boldsymbol{B}_{t+1}\cdot\ldots\cdot\boldsymbol{B}_n\mathbf{1}^T / \boldsymbol{a}_0\boldsymbol{E}(x_1)\boldsymbol{B}_2\cdot\ldots\cdot\boldsymbol{B}_{t-1}\boldsymbol{A}\boldsymbol{B}_{t+1}\cdot\ldots\cdot\boldsymbol{B}_n\mathbf{1}^T$$

$$\propto \boldsymbol{\alpha}_{t-1}\boldsymbol{A}\boldsymbol{E}(x)\boldsymbol{B}_{t+1}\cdot\ldots\cdot\boldsymbol{B}_n\mathbf{1}^T$$

REF: Zucchini & MacDonald (2009)

HMM Diagnostics

- *Forecast pseudo-residuals*: $z_t = \Phi^{-1}\{P(X_t \leq x_t | x_1,\dots, x_{t-1})\}$
  - Deviation from the median of the one-step-ahead forecast
  - If indicating an outlier, unacceptable description of the series by the HMM
  - Possible monitoring of a series

- *Forecast pseudo-residual segments*: $[z_t^-, z_t^+]$ with $z_t^- = \Phi^{-1}\{P(X_t < x_t | x_1,\dots, x_{t-1})\}$ and $z_t^+ = \Phi^{-1}\{P(X_t \leq x_t | x_1,\dots, x_{t-1})\}$

- Note that $P(X_t = x_t | x_1,\dots, x_{t-1}) = P(x_1,\dots, x_t) / P(x_1,\dots, x_{t-1}) =$ $\boldsymbol{a}_0\boldsymbol{E}(\mathrm{x}_1)\boldsymbol{B}_2\cdot\ldots\cdot\boldsymbol{B}_{t-1}\boldsymbol{AE}(x)\mathbf{1}^{\mathrm{T}}/\boldsymbol{a}_0\boldsymbol{E}(\mathrm{x}_1)\boldsymbol{B}_2\cdot\ldots\cdot\boldsymbol{B}_{t-2}\boldsymbol{AE}(x)\mathbf{1}^{\mathrm{T}}$ $= \boldsymbol{\alpha}_{t-1}\boldsymbol{AE}(x)\mathbf{1}^{\mathrm{T}}/\boldsymbol{\alpha}_{t-1}\mathbf{1}^{\mathrm{T}}$

REF: Zucchini & MacDonald (2009)

Earthquakes data

- Autocorrelation functions
  - two states: $\rho(k) = 0.5713 \times 0.8055^k$;
  - three states: $\rho(k) = 0.4447 \times 0.9141^k + 0.1940 \times 0.7433^k$;
  - four states: $\rho(k) = 0.2332 \times 0.9519^k + 0.3682 \times 0.8174^k + 0.0369 \times 0.7252^k$.

k:	1	2	3	4	5	6	7	8
observations	0.570	0.444	0.426	0.379	0.297	0.251	0.251	0.149
2-state model	0.460	0.371	0.299	0.241	0.194	0.156	0.126	0.101
3-state model	0.551	0.479	0.419	0.370	0.328	0.292	0.261	0.235
4-state model	0.550	0.477	0.416	0.366	0.324	0.289	0.259	0.234

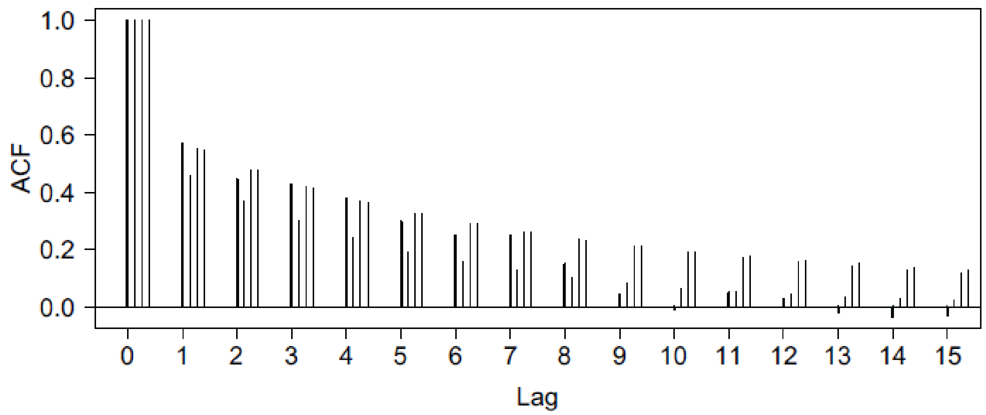
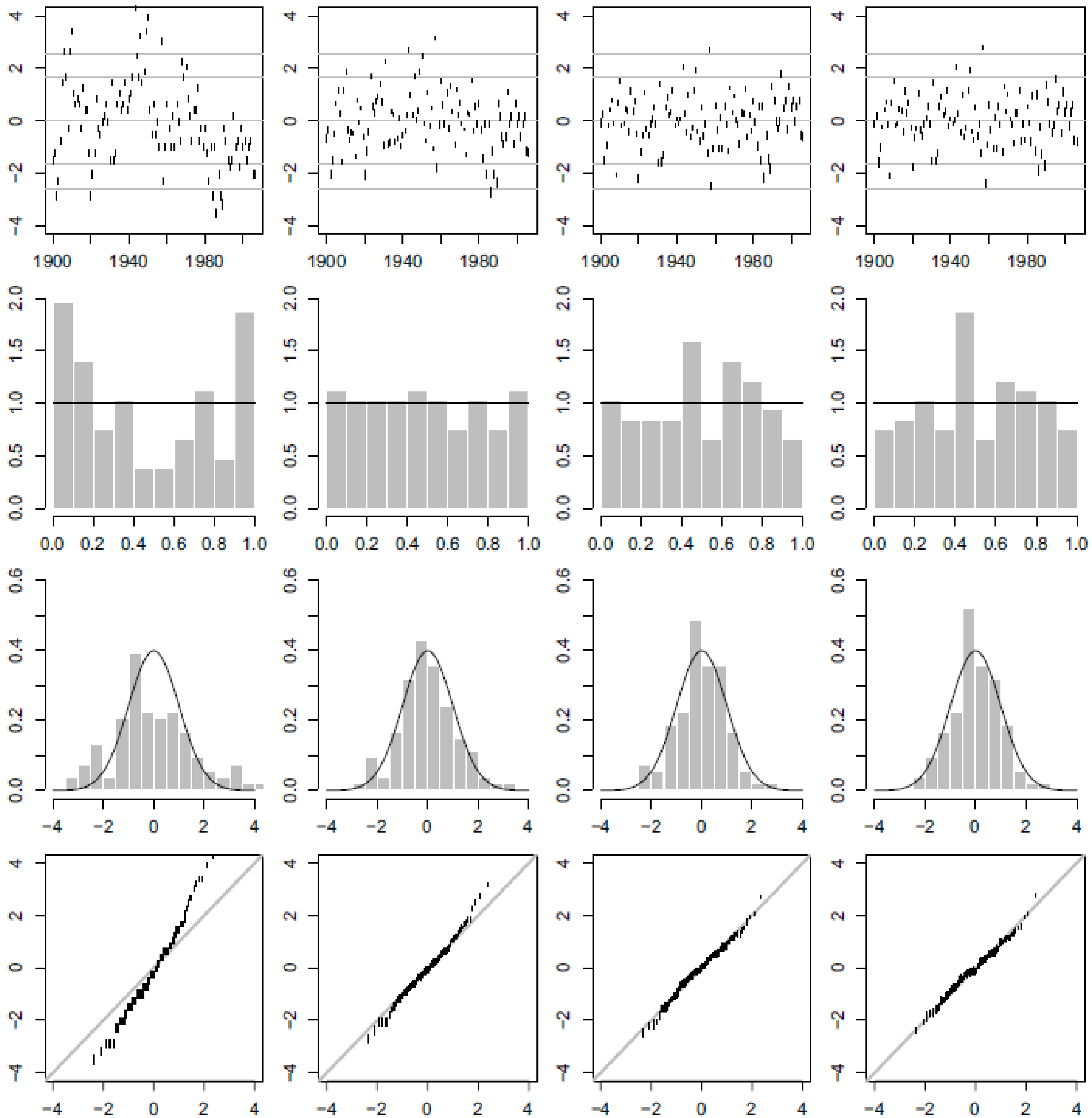


Figure 6.2 *Earthquakes data: sample ACF and ACF of three models. The bold bars on the left represent the sample ACF, and the other bars those of the HMMs with (from left to right) two, three and four states.*

REF: Zucchini & MacDonald (2009)

Earthquakes data

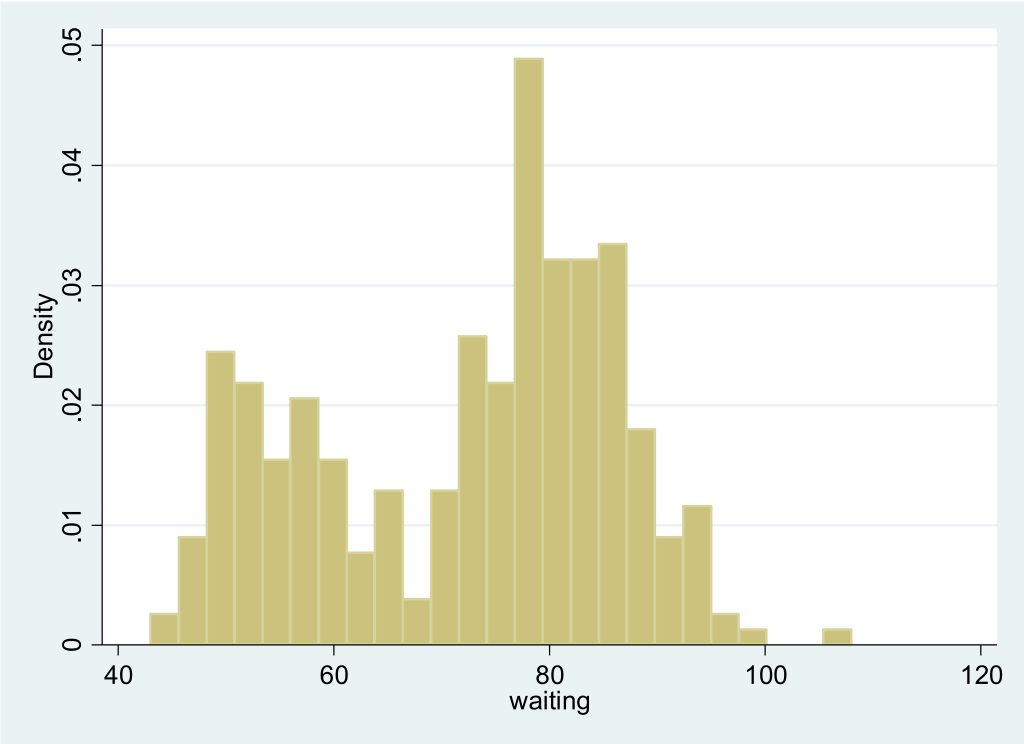
- Ordinary pseudo-residuals (m=1,2,3,4)

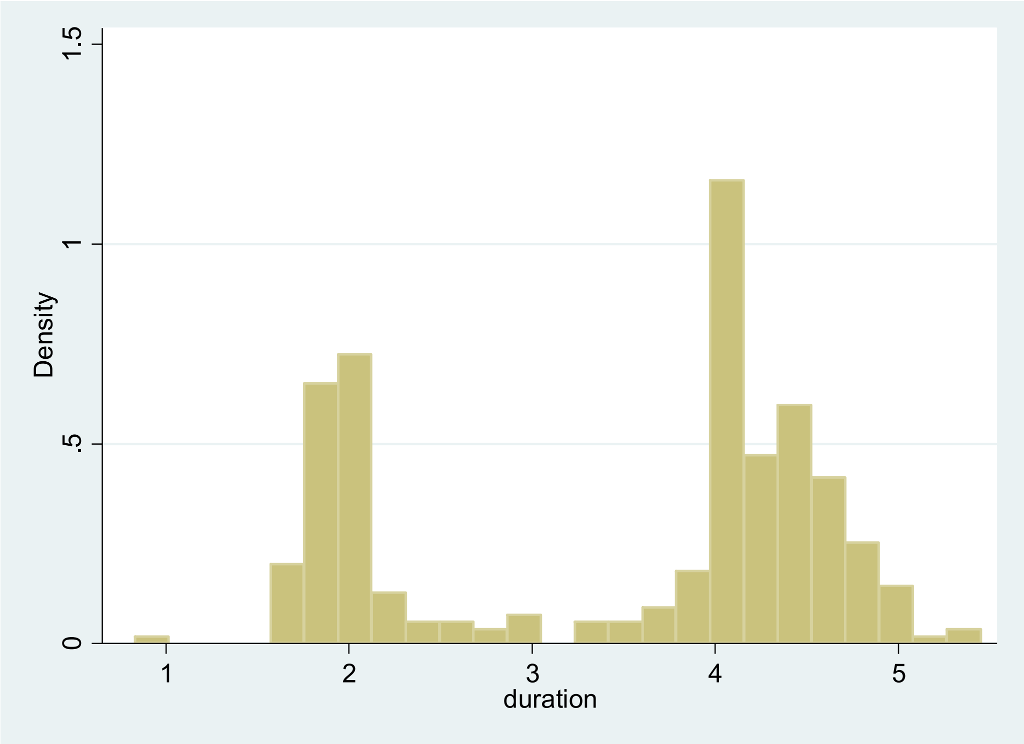


REF: Zucchini & MacDonald (2009)

Old Faithful geyser data

- 299 pairs of measurements
- Time interval between the starts of successive eruptions *w* and the duration of the subsequent eruption *d* (min)
  - Recorded to the nearest second. Except for a few Short, Medium, Long observations (replaced by 2, 3, 4 minutes)
  - Interval data: (0,3), (2.5, 3.5), and (3, 20), or observation ± 0.5 sec





Old Faithful geyser data

- Duration
- Mixtures of normals
- Continuous likelihood: $P(X=x)=\sum_q \pi_q f_q(x)$
  - did not work for m=4
- Discrete likelihood: $P(X=x)=\sum_q \pi_q \, {}_a\!\int^b f_q(x)$

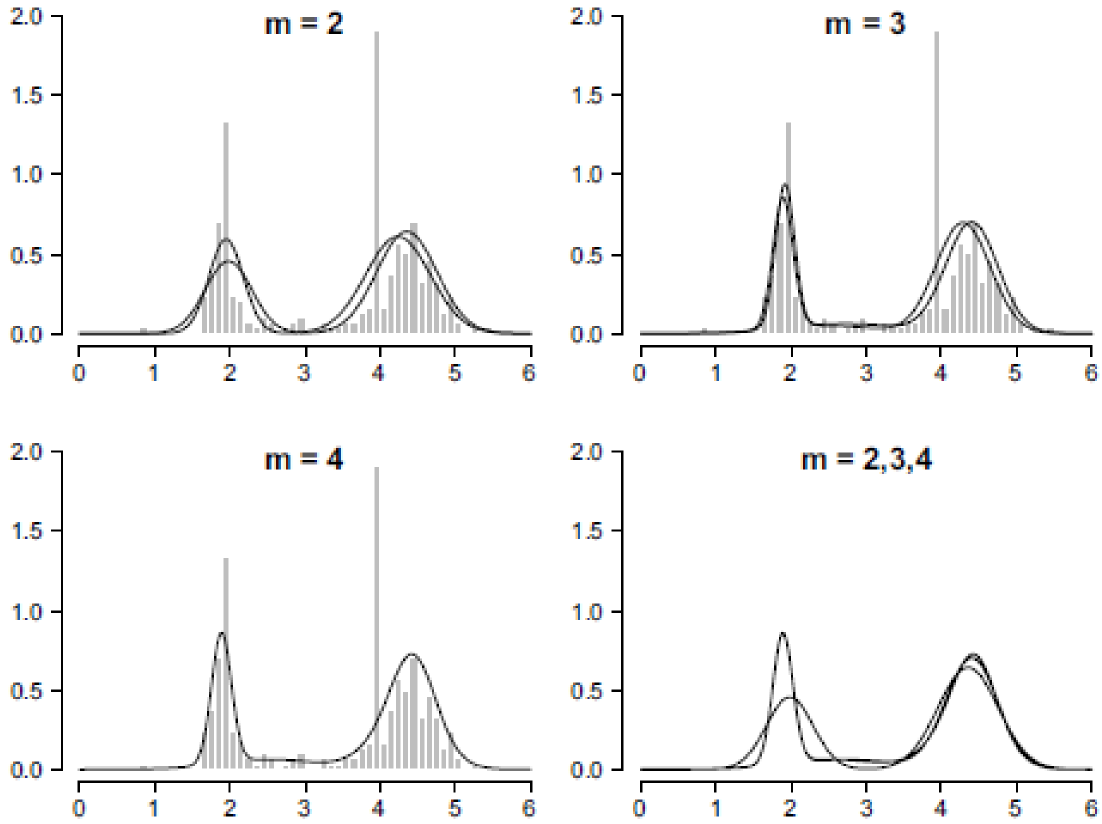


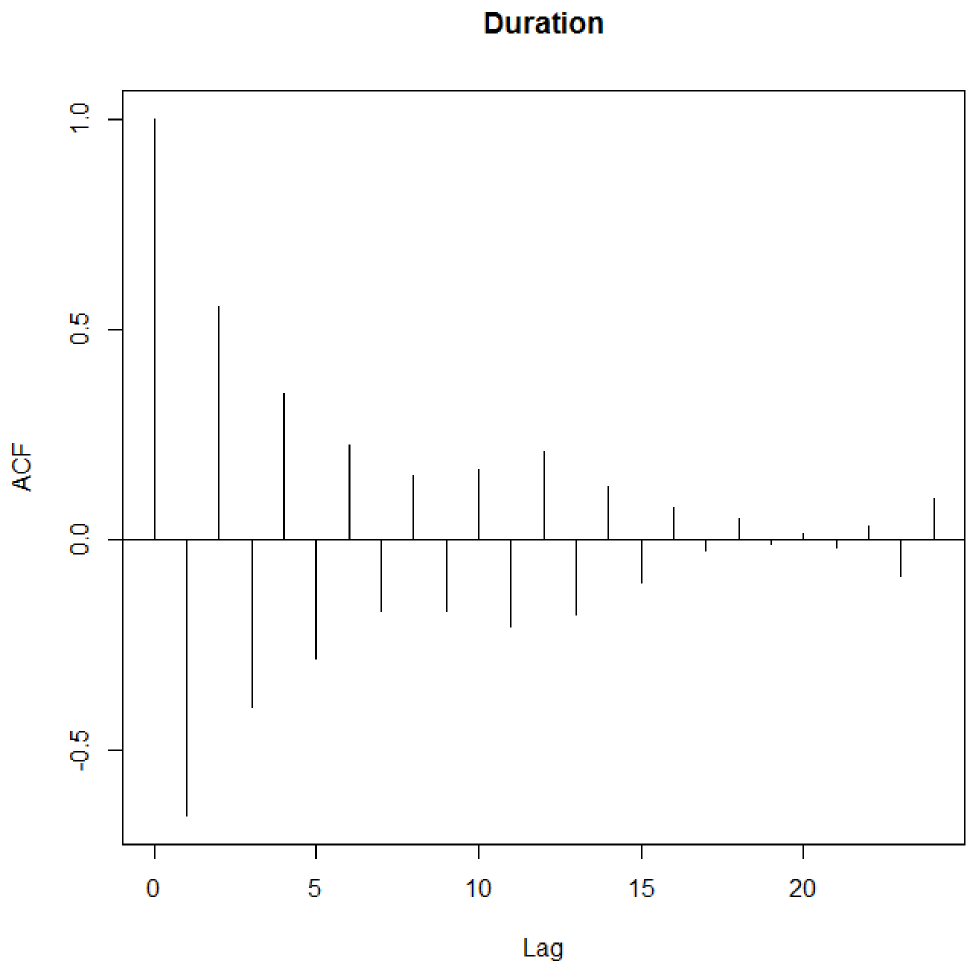
Figure 1.5 Old Faithful durations: histogram of observations (with S,M,L replaced by 2, 3, 4), compared to independent mixtures of 2–4 normal distributions. Thick lines (only for m = 2 and 3): p.d.f. of model based on continuous likelihood. Thin lines (all cases): p.d.f. of model based on discrete likelihood.

REF: Zucchini & MacDonald (2009)

Old Faithful geyser data

- Duration

- Autocorrelation function



Old Faithful geyser data

- Duration, HMM, discrete likelihood

model	k	$-\log L$	AIC	BIC
2-state HM	6	1168.955	2349.9	2372.1
3-state HM	12	1127.185	2278.4	**2322.8**
4-state HM	20	1109.147	**2258.3**	2332.3
indep. mixture (2)	5	1230.920	2471.8	2490.3
indep. mixture (3)	8	1203.872	2423.7	2453.3
indep. mixture (4)	11	1203.636	2429.3	2470.0

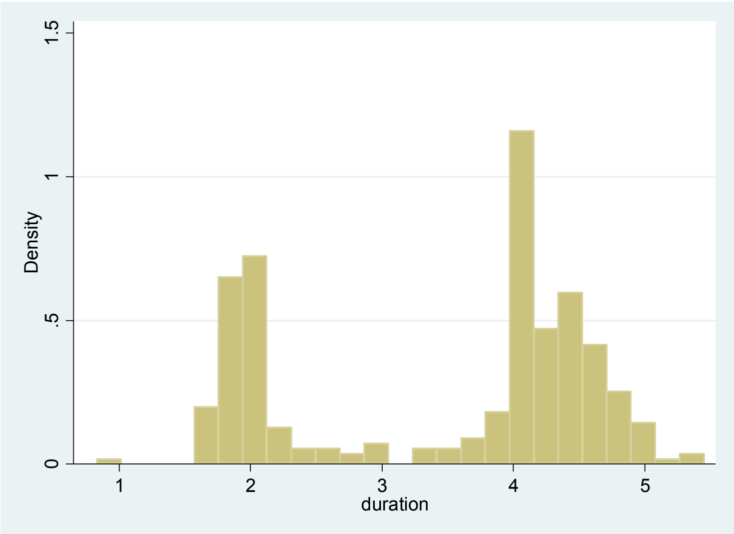
	$\boldsymbol{\Gamma}$	
0.000	0.000	1.000
0.053	0.113	0.834
0.546	0.337	0.117

i	1	2	3
δ_i	0.291	0.195	0.514
μ_i	1.894	3.400	4.459
σ_i	0.139	0.841	0.320

REF: Zucchini & MacDonald (2009)

Old Faithful geyser data

- Duration, HMM



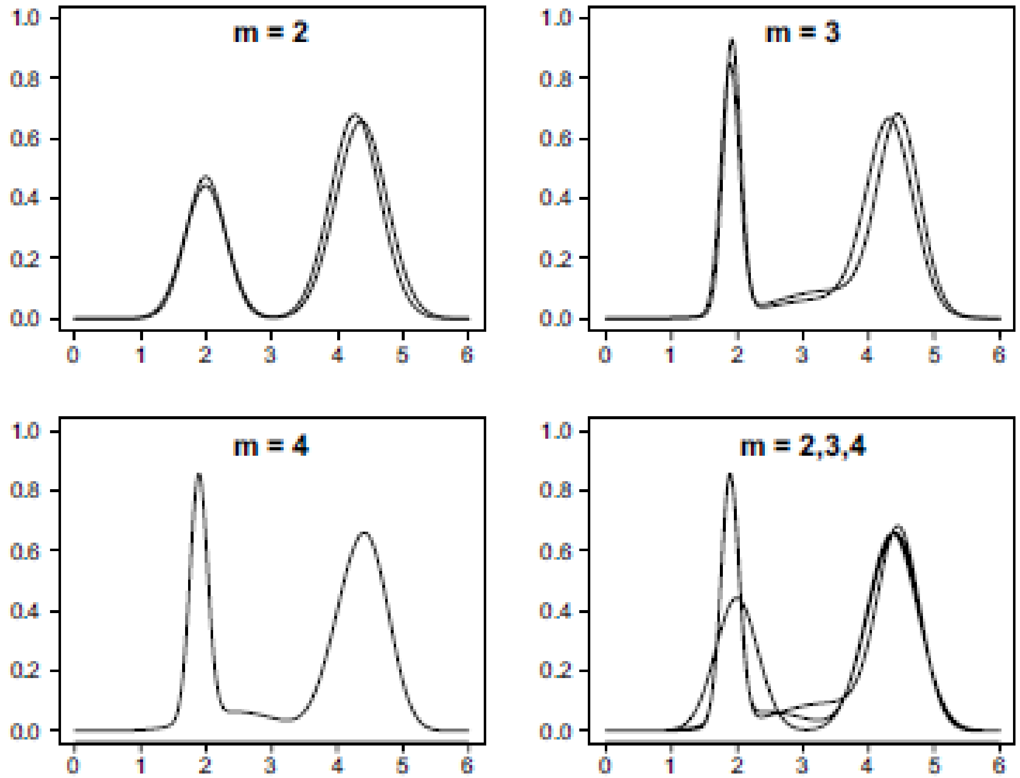
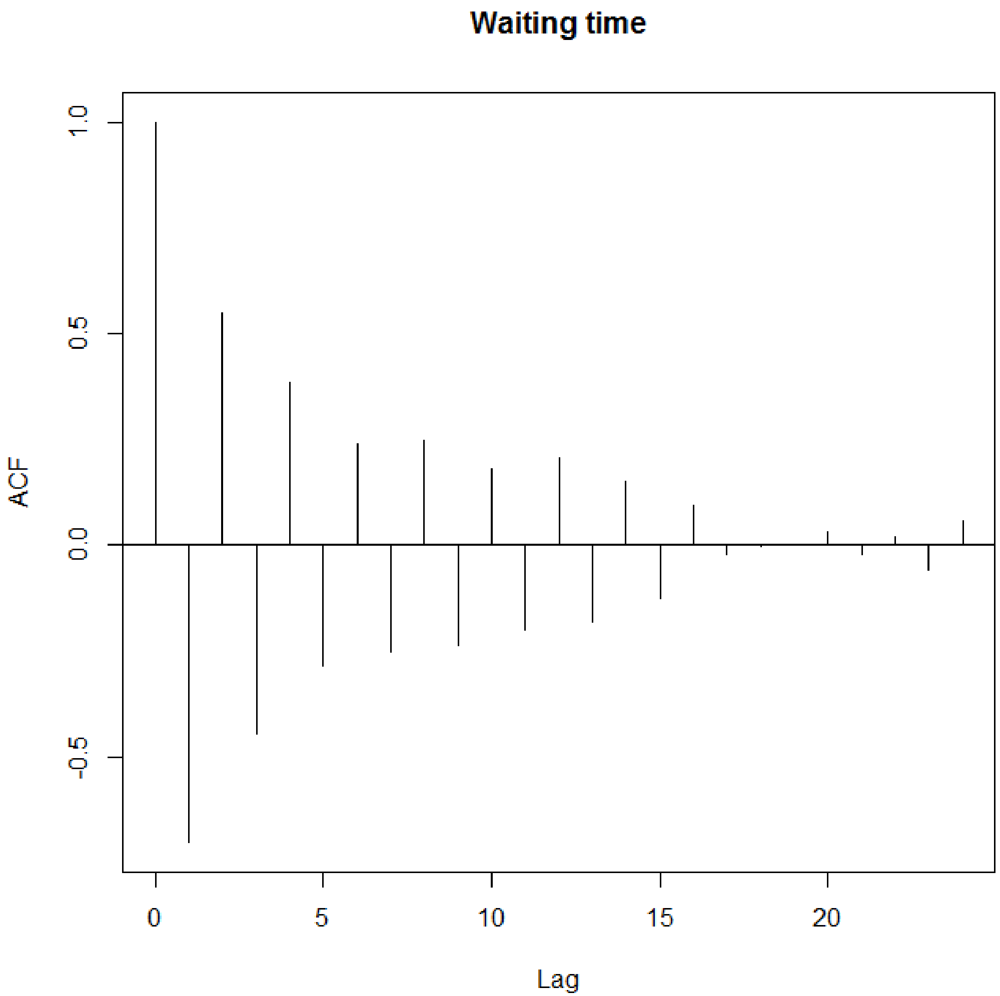


Figure 10.2 *Old Faithful durations, normal–HMMs. Thick lines ($m = 2$ and 3 only): models based on continuous likelihood. Thin lines (all panels): models based on discrete likelihood.*

REF: Zucchini & MacDonald (2009)

Old Faithful geyser data

- Waiting time
- Autocorrelation function



Old Faithful geyser data

- Waiting time, HMM, discrete likelihood

model	k	$-\log L$	AIC	BIC
2-state HM	6	1092.794	2197.6	2219.8
3-state HM	12	1051.138	2126.3	**2170.7**
4-state HM	20	1038.600	**2117.2**	2191.2

Γ		
0.000	0.000	1.000
0.298	0.575	0.127
0.662	0.276	0.062

i	1	2	3
δ_i	0.342	0.259	0.399
μ_i	55.30	75.30	84.93
σ_i	5.809	3.808	5.433

REF: Zucchini & MacDonald (2009)

Old Faithful geyser data

- Waiting time, HMM

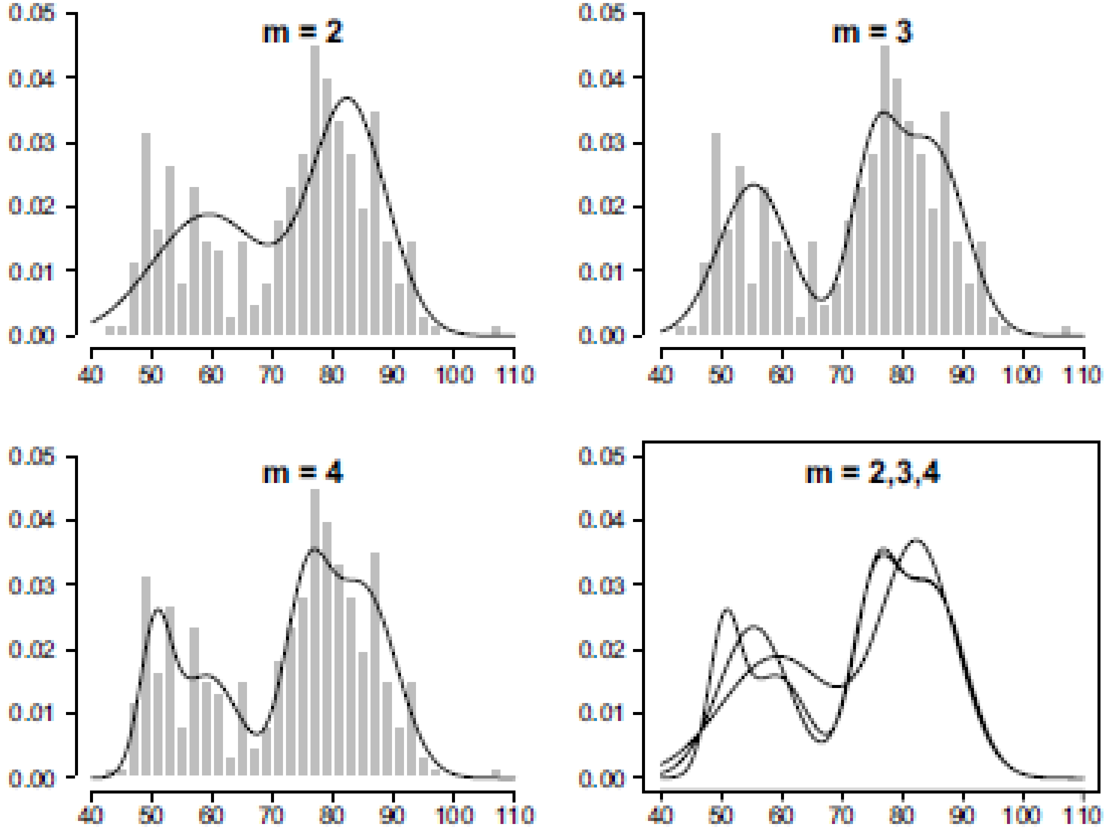


Figure 10.3 *Old Faithful waiting times, normal–HMMs. Models based on continuous likelihood and models based on discrete likelihood are essentially the same. Notice that the model for $m = 3$ is identical, or almost identical, to the three-state model of Robert and Titterington (1998): see their Figure 7.*

REF: Zucchini & MacDonald (2009)

Use of Hidden Markov Models

- HMM is a flexible statistical tool that can be used to analyze serially-correlated data
  - continuous, discrete, multivariate
- Diagnostic methods are available
- Numerical complexity (can be addressed)
- R packages: HiddenMarkov, HMM, HMMCont, hmm.discnp