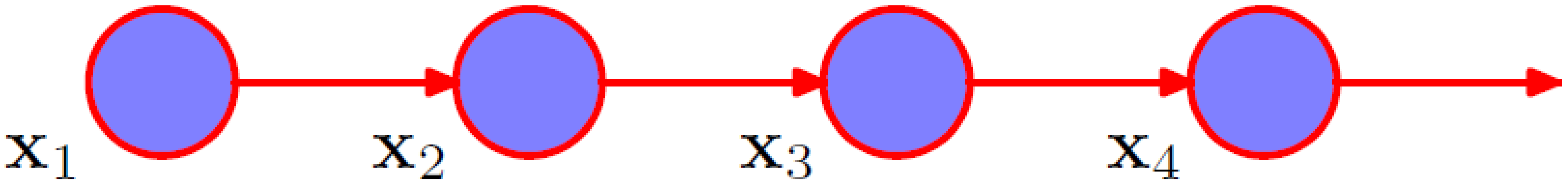
Hidden Markov Models

Based on www-nlp.stanford.edu/fsnlp/**hmm-**chap/**blei-hmm-**ch9.ppt

Models for sequential data

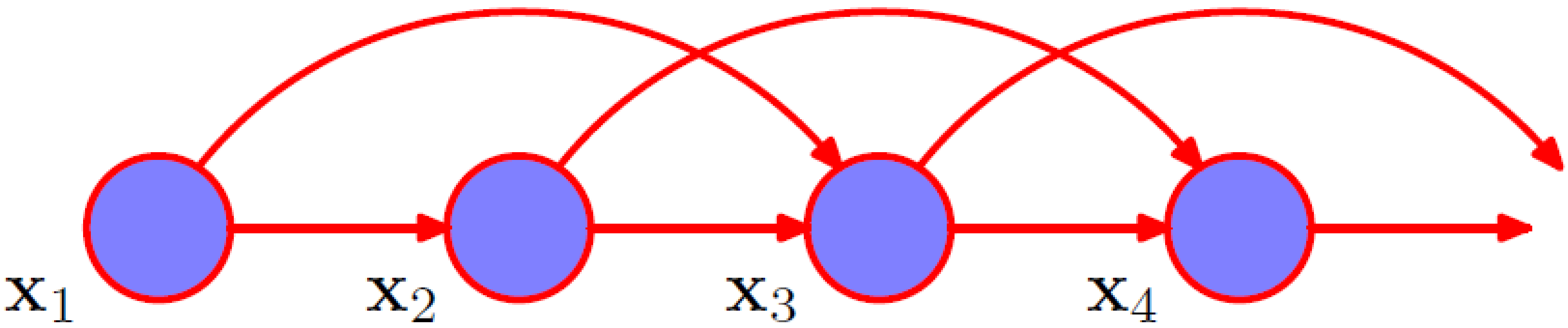
- First-order Markov model: conditions on previous observation:



$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = p(\mathbf{x}_1) \prod_{n=2}^{N} p(\mathbf{x}_n | \mathbf{x}_{n-1}).$$

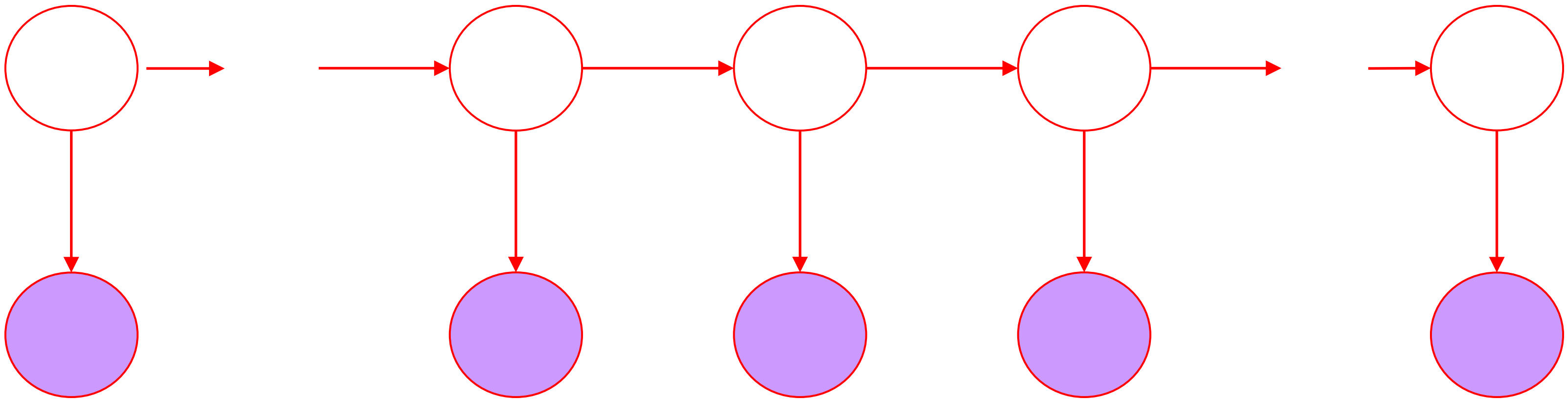
Models for sequential data

- Second-order Markov model conditions on the two previous



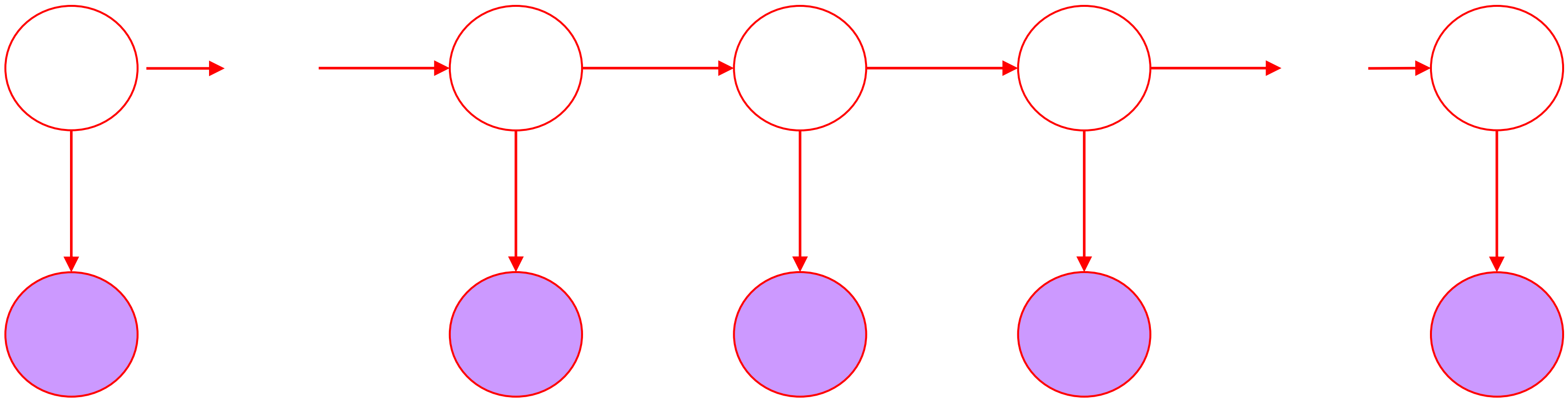
$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = p(\mathbf{x}_1)p(\mathbf{x}_2|\mathbf{x}_1)\prod_{n=3}^{N} p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{x}_{n-2}).$$

Hidden Markov Model



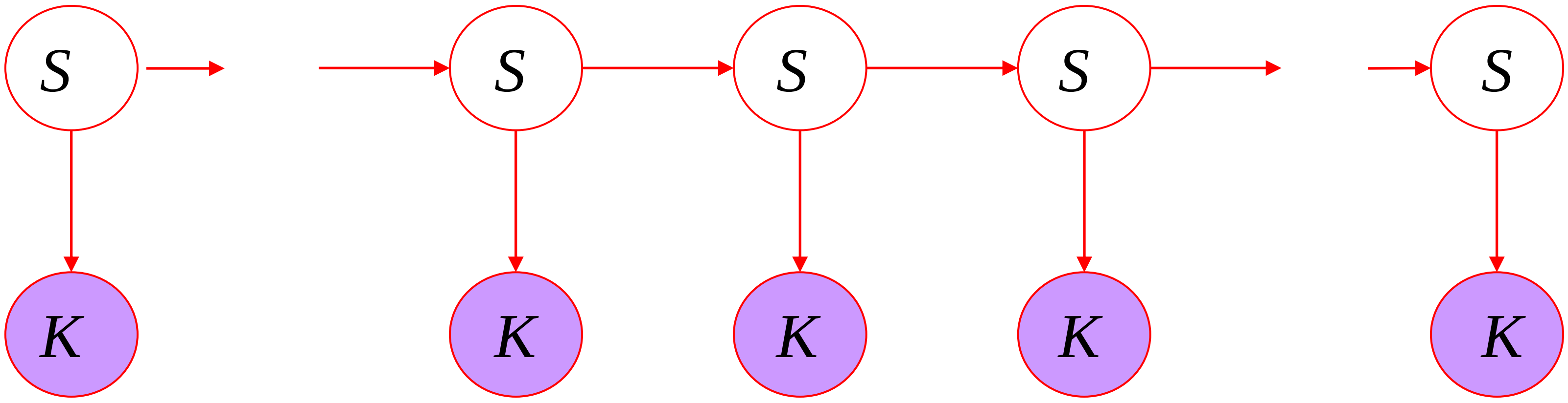
- ***Hidden states*** – Markov chain:
  - Dependent only on the previous state
  - "The past is independent of the future given the present."

Hidden Markov Model



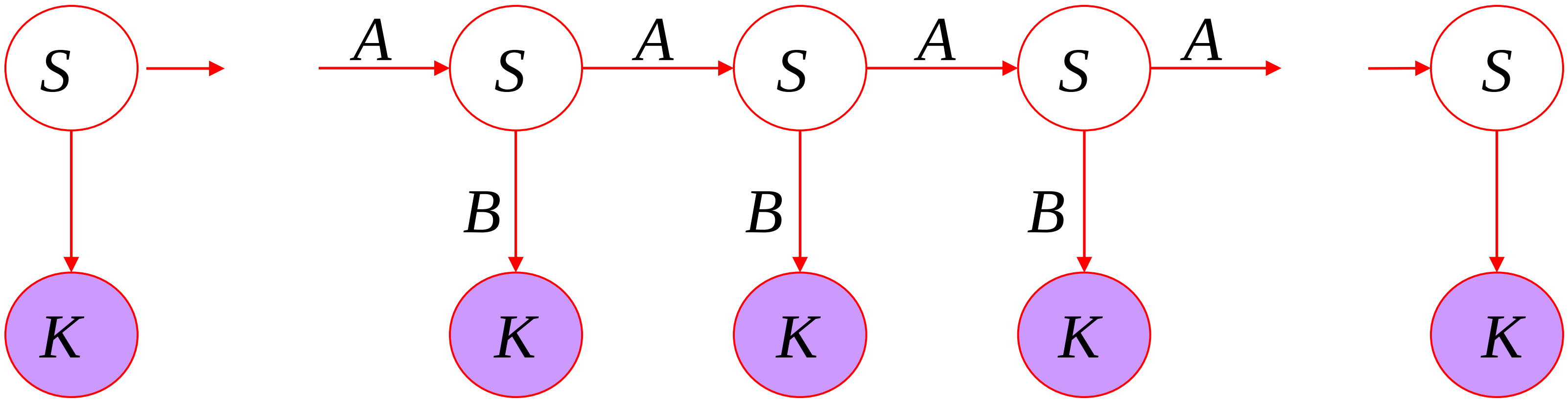
- Shaded nodes are ***observed variables***
- Dependent only on their corresponding hidden state

HMM Formalism



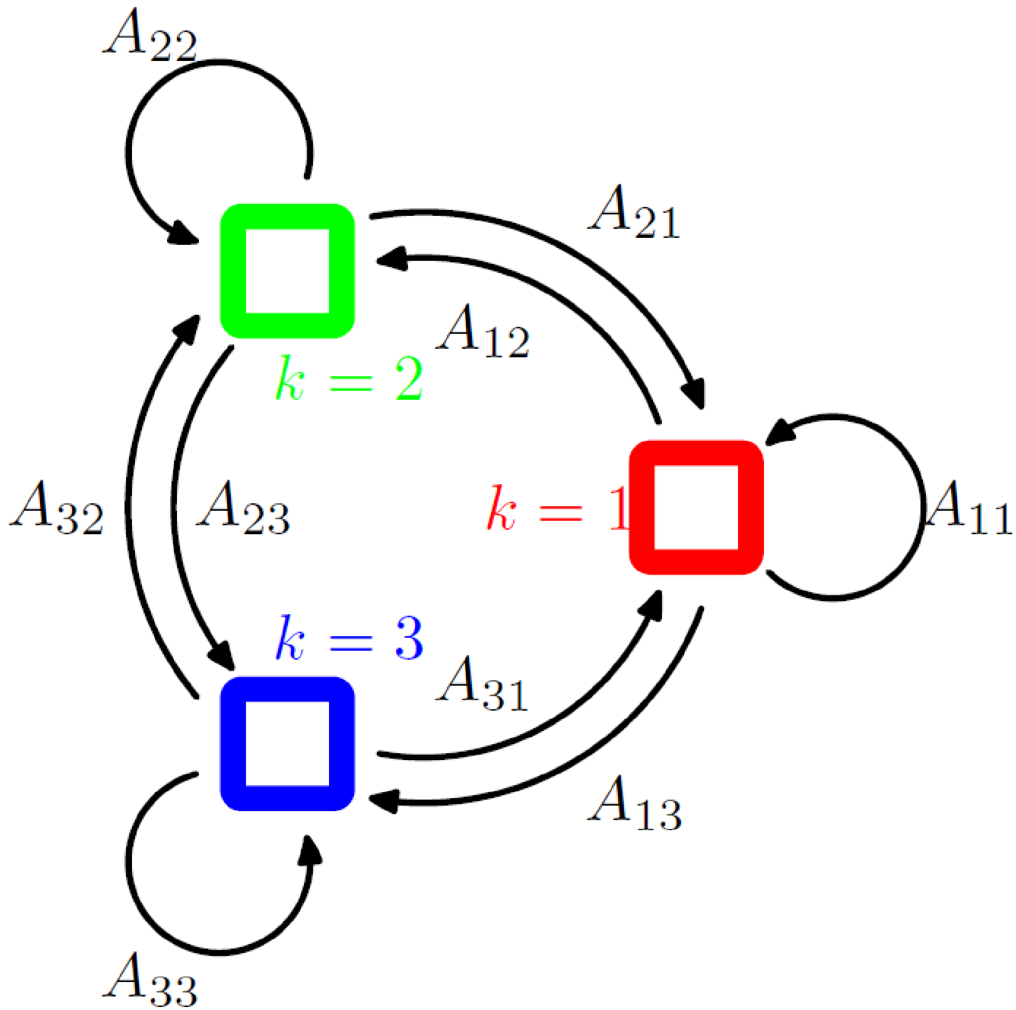
- S : $\{s_1 \ldots s_N\}$ are the values for the hidden states
- K : $\{k_1 \ldots k_M\}$ are the values for the observations

HMM Formalism

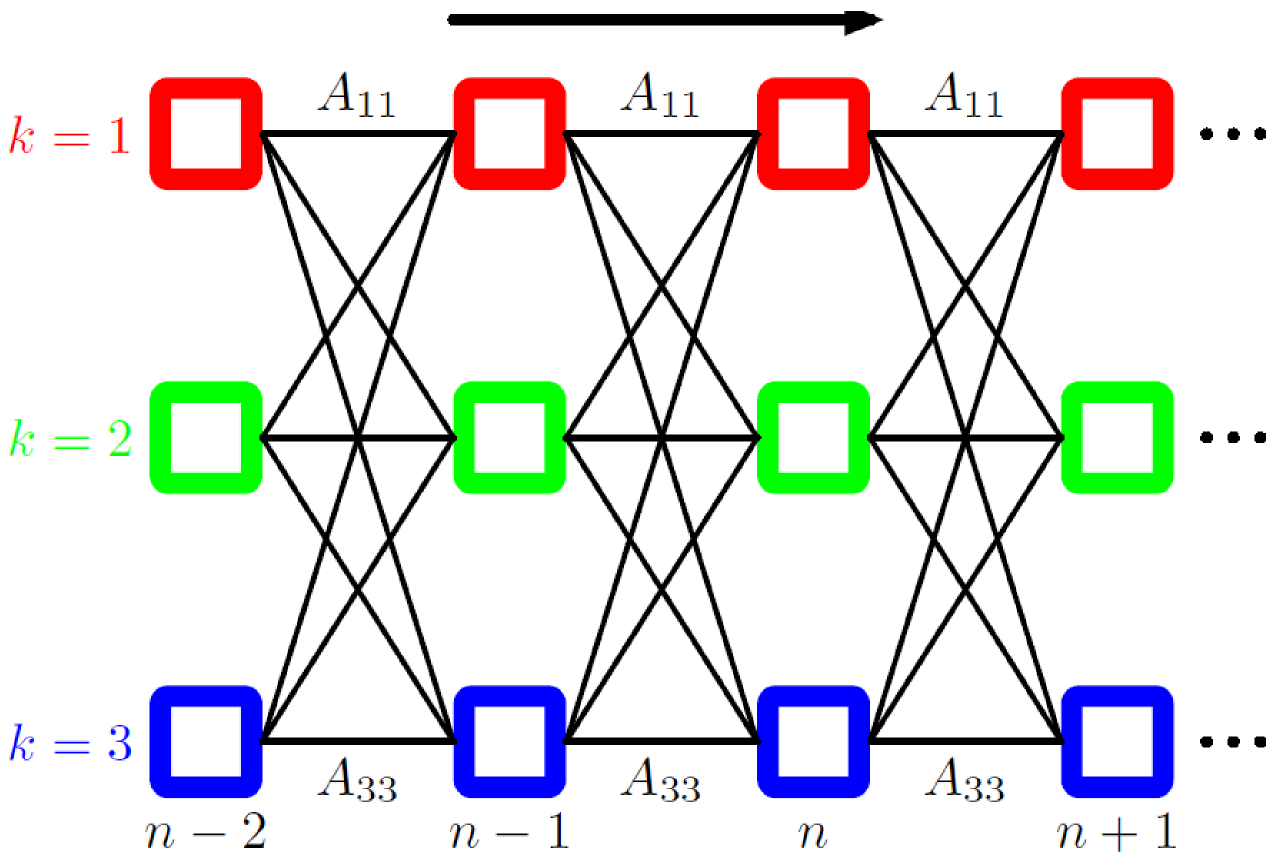


- Parameters: $\{S, K, \Pi, A, B\}$
- Initial hidden state probabilities: $\Pi = \{\pi_i\}$
- Transition probabilities. $A = \{a_{ij}\}$ are the state transition probabilities.
- Emission probabilities. $B = \{b_{ik}\}$ are the observation state probabilities (HMM can also work with continues emission probabilities).

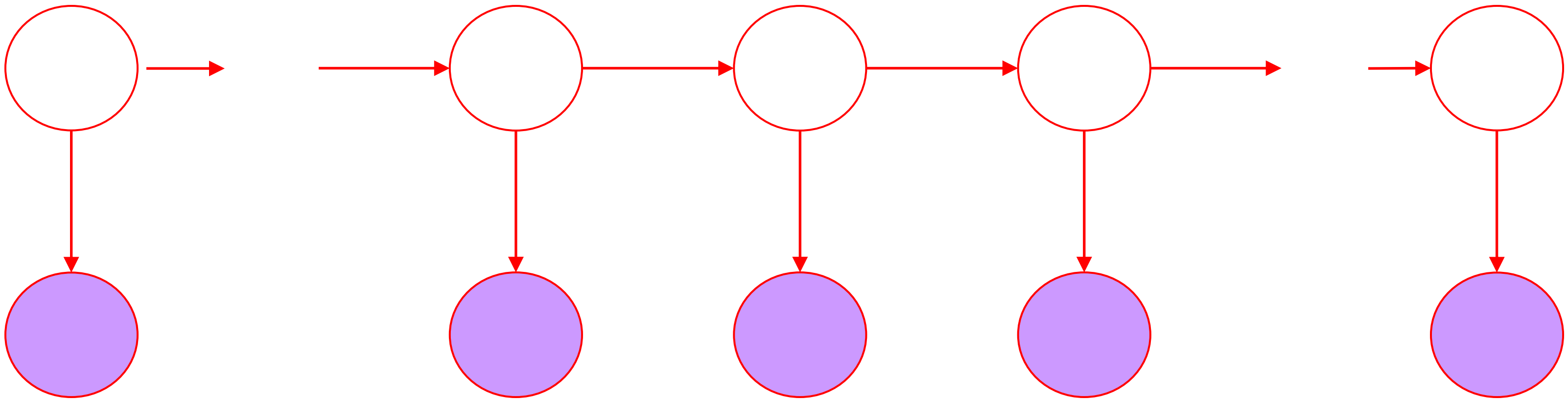
HMM hidden state example



HMM hidden state example

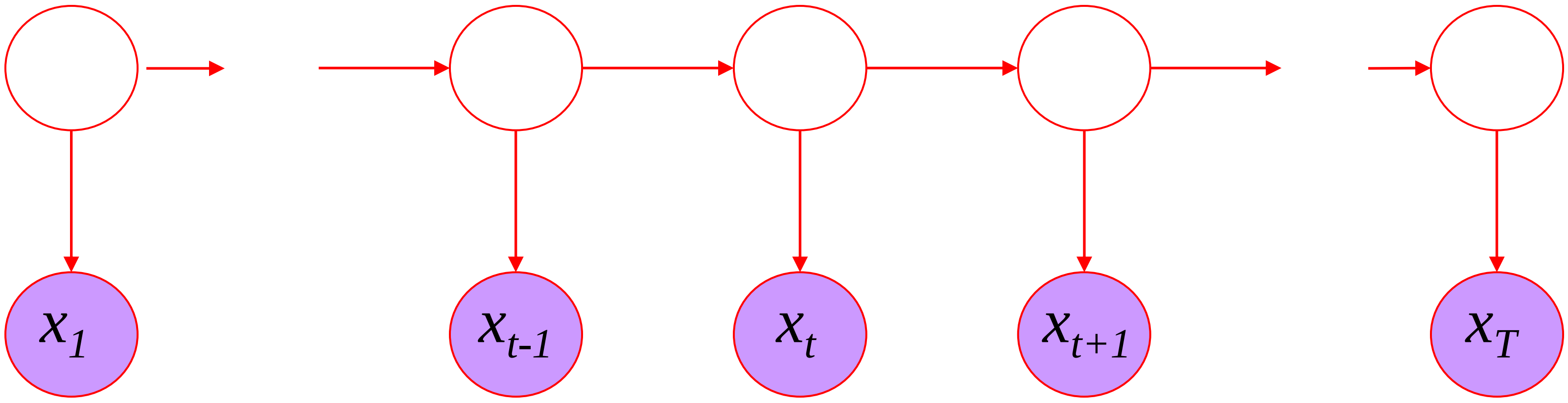


Inference in an HMM



- Compute the probability of a given observation sequence
- Given an observation sequence, compute the most likely hidden state sequence
- Given an observation sequence and set of possible models, which model most closely fits the data?

Decoding

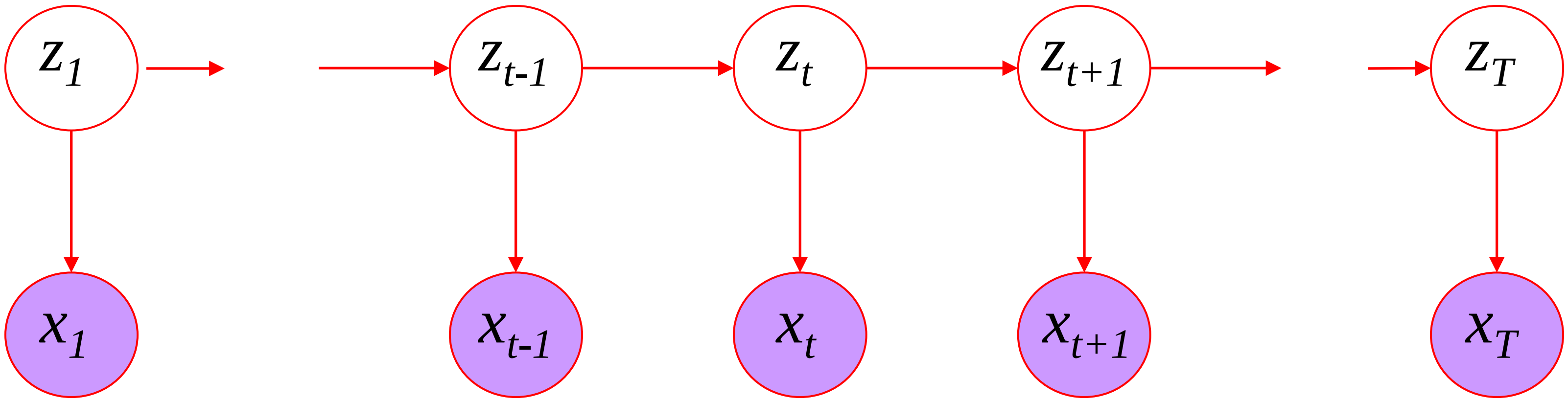


Given an observation sequence and a model, compute the probability of the observation sequence

$$X = (x_1 \dots x_T), \theta = (A, B, \Pi)$$

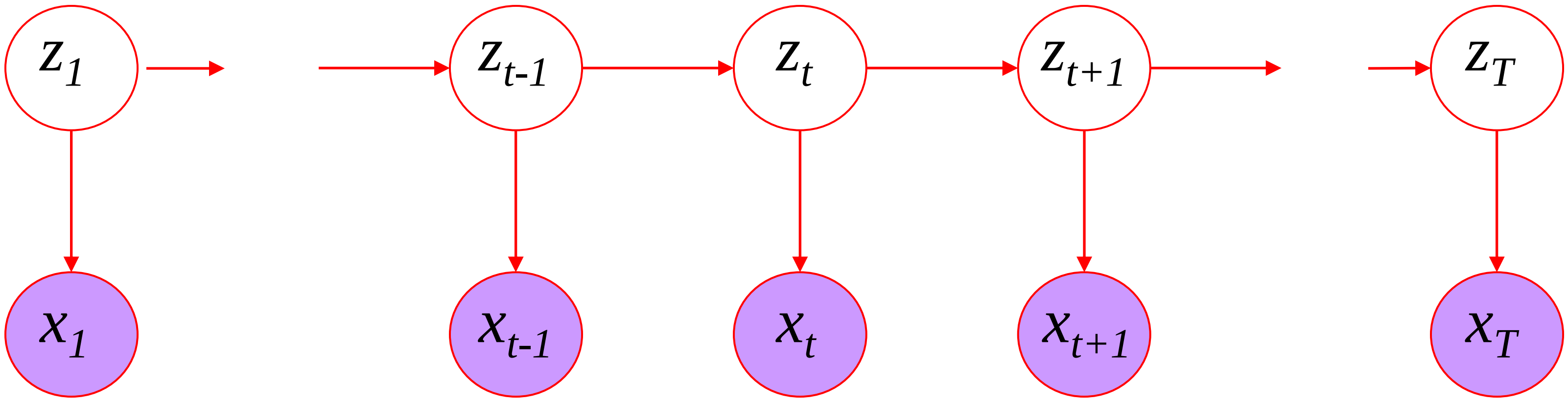
Compute $P(X \mid \theta)$

Decoding



$$P(X \mid Z, \theta) = b_{z_1 x_1} b_{z_2 x_2} \ldots b_{z_T x_T}$$

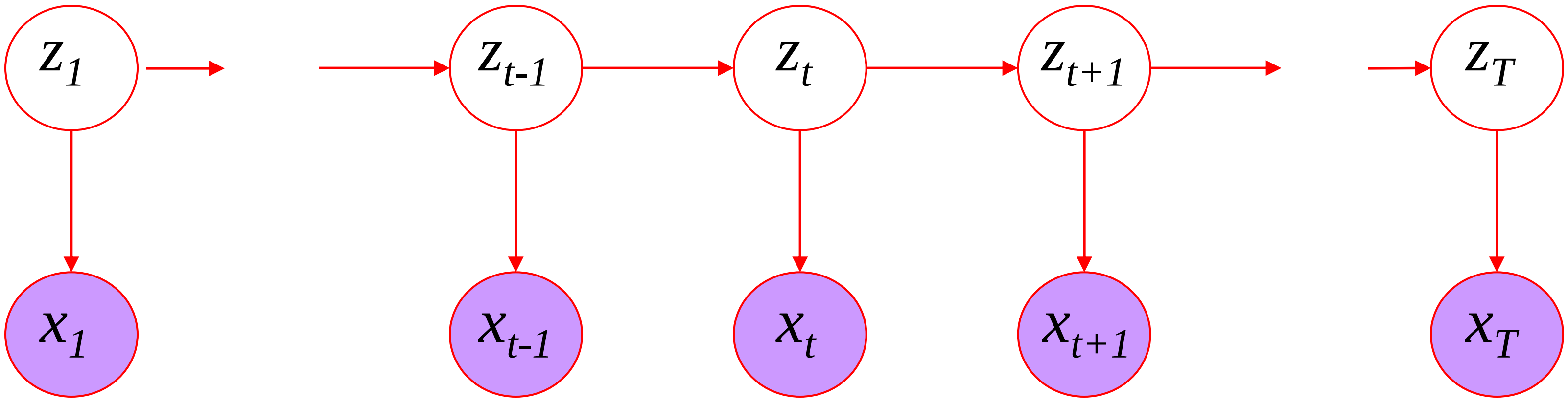
Decoding



$$P(X \mid Z, \theta) = b_{z_1 x_1} b_{z_2 x_2} \ldots b_{z_T x_T}$$

$$P(Z \mid \theta) = \pi_{z_1} a_{z_1 z_2} a_{z_2 z_3} \ldots a_{z_{T-1} z_T}$$

Decoding

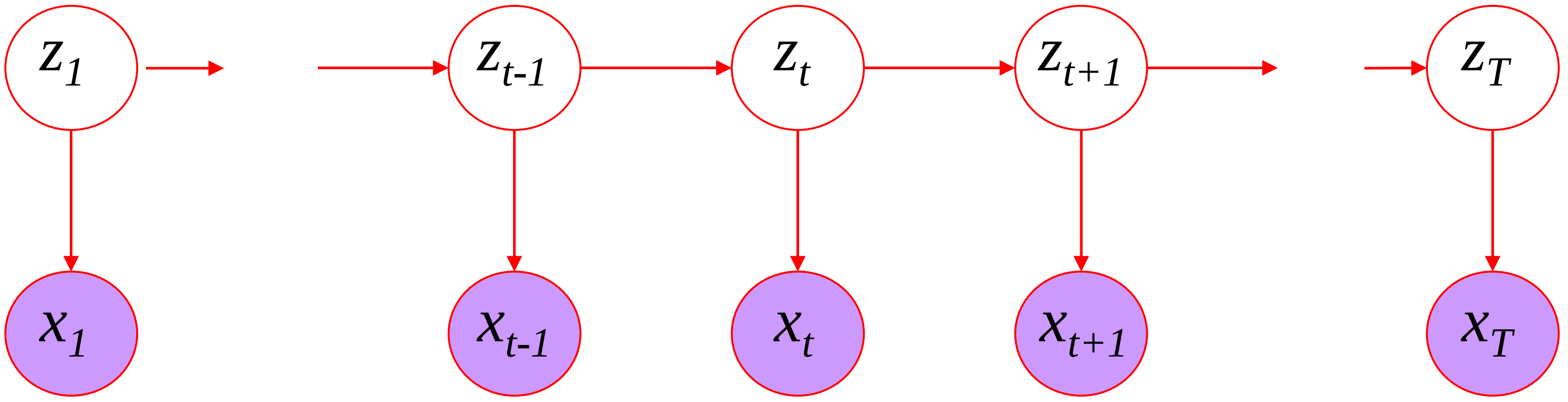


$$P(X \mid Z, \theta) = b_{z_1 x_1} b_{z_2 x_2} \ldots b_{z_T x_T}$$

$$P(Z \mid \theta) = \pi_{z_1} a_{z_1 z_2} a_{z_2 z_3} \ldots a_{z_{T-1} z_T}$$

$$P(X, Z \mid \theta) = P(X \mid Z, \theta) P(Z \mid \theta)$$

Decoding



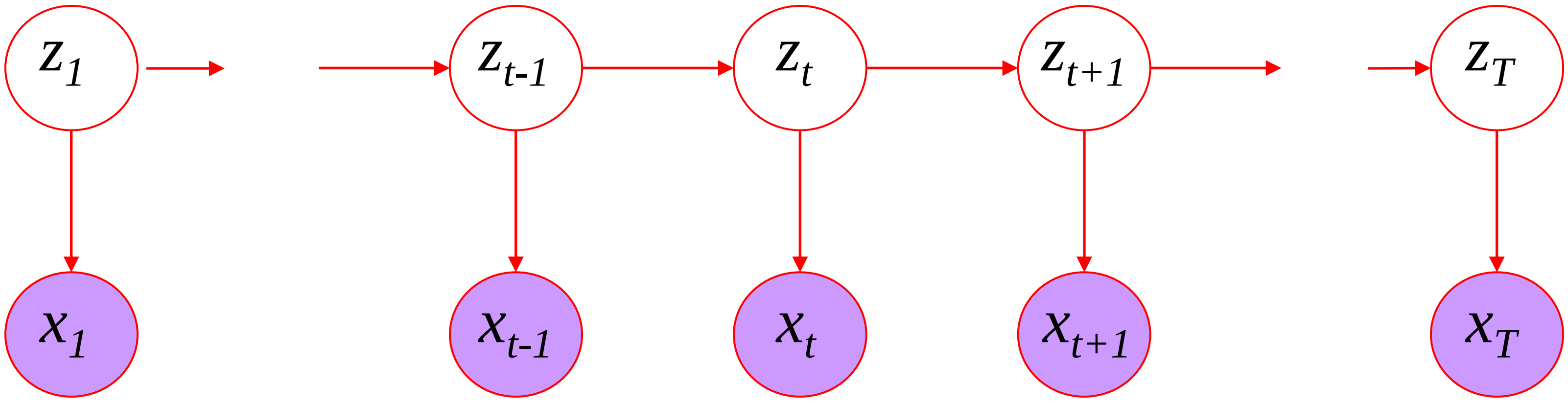
$$P(X \mid Z, \theta) = b_{z_1 x_1} b_{z_2 x_2} \ldots b_{z_T x_T}$$

$$P(Z \mid \theta) = \pi_{z_1} a_{z_1 z_2} a_{z_2 z_3} \ldots a_{z_{T-1} z_T}$$

$$P(X, Z \mid \theta) = P(X \mid Z, \theta) P(Z \mid \theta)$$

$$P(X \mid \theta) = \sum_Z P(X \mid Z, \theta) P(Z \mid \theta)$$

Decoding

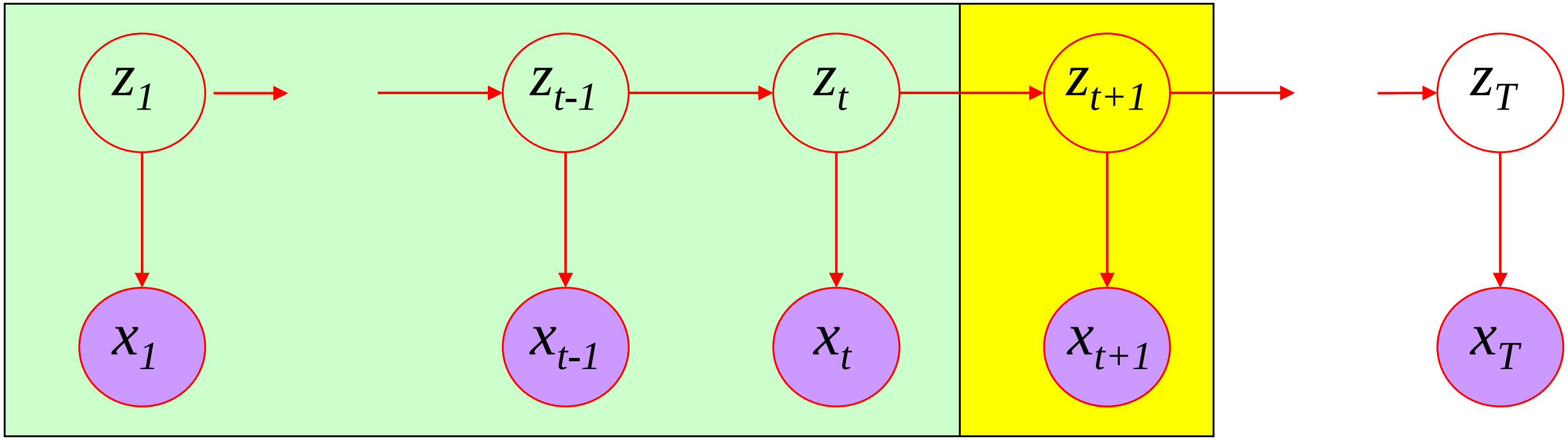


$$P(X \mid \theta) = \sum_{\{z_1 \ldots z_T\}} \pi_{z_1} b_{z_1 x_1} \prod_{t=1}^{T-1} a_{z_t z_{t+1}} b_{z_{t+1} x_{t+1}}$$

☹ Doesn't factorize over t.

☹ The sum contains N^T terms: T variables, each with N states - the number of terms grows exponentially with the length of the chain

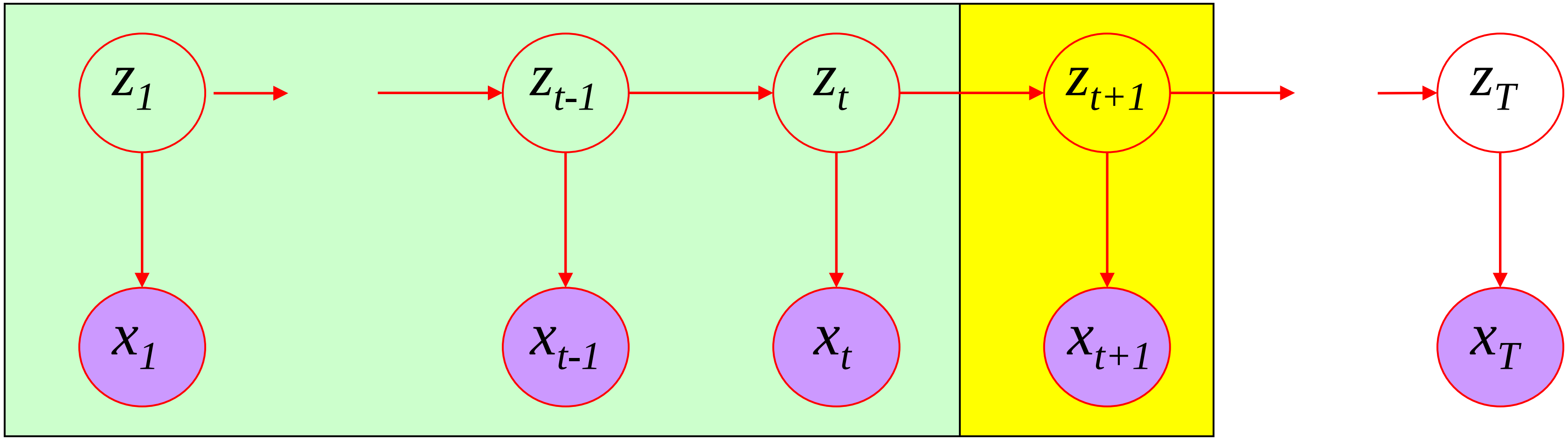
Forward Procedure



- Special structure gives us an efficient solution using *dynamic programming*.
- **Intuition**: Probability of the first *t* observations is the same for all possible *t*+1 length state sequences.
- **Define:**

$$\alpha_i(t) = P(x_1 \ldots x_t, z_t = i \mid \theta)$$

Forward Procedure



$$\alpha_j(t+1)$$

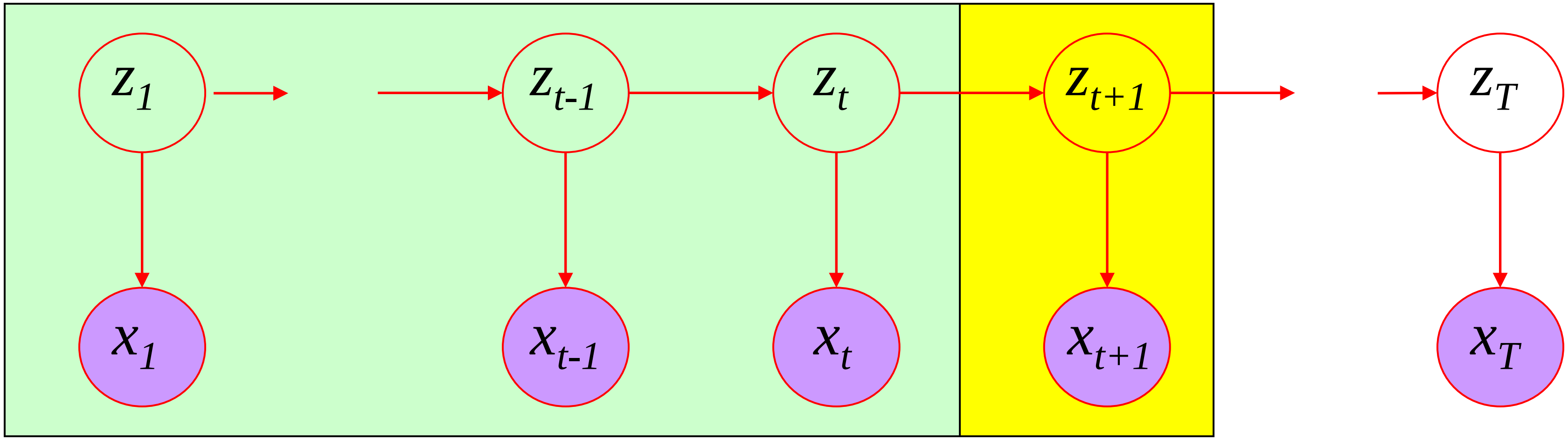
$$= P(x_1 \ldots x_{t+1}, z_{t+1} = j)$$

$$= P(x_1 \ldots x_{t+1} \mid z_{t+1} = j) P(z_{t+1} = j)$$

$$= P(x_1 \ldots x_t \mid z_{t+1} = j) P(x_{t+1} \mid z_{t+1} = j) P(z_{t+1} = j)$$

$$= P(x_1 \ldots x_t, z_{t+1} = j) P(x_{t+1} \mid z_{t+1} = j)$$

Forward Procedure



$$\alpha_j(t+1)$$

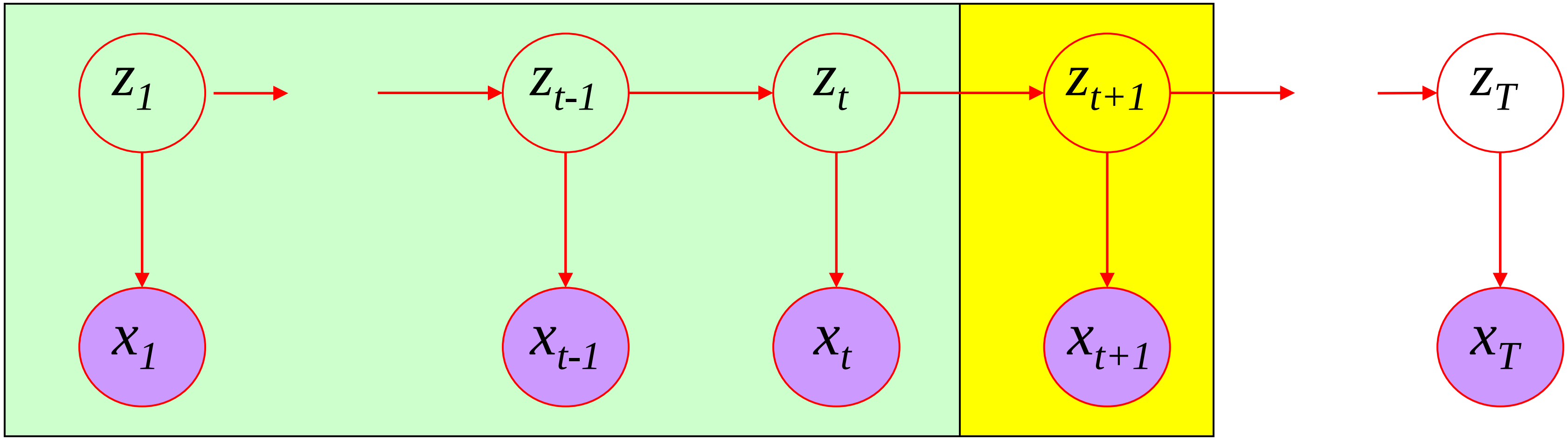
$$= P(x_1 ... x_{t+1}, z_{t+1} = j)$$

$$= P(x_1 ... x_{t+1} \mid z_{t+1} = j) P(z_{t+1} = j)$$

$$= P(x_1 ... x_t \mid z_{t+1} = j) P(x_{t+1} \mid z_{t+1} = j) P(z_{t+1} = j)$$

$$= P(x_1 ... x_t, z_{t+1} = j) P(x_{t+1} \mid z_{t+1} = j)$$

Forward Procedure



$$\alpha_j(t+1)$$

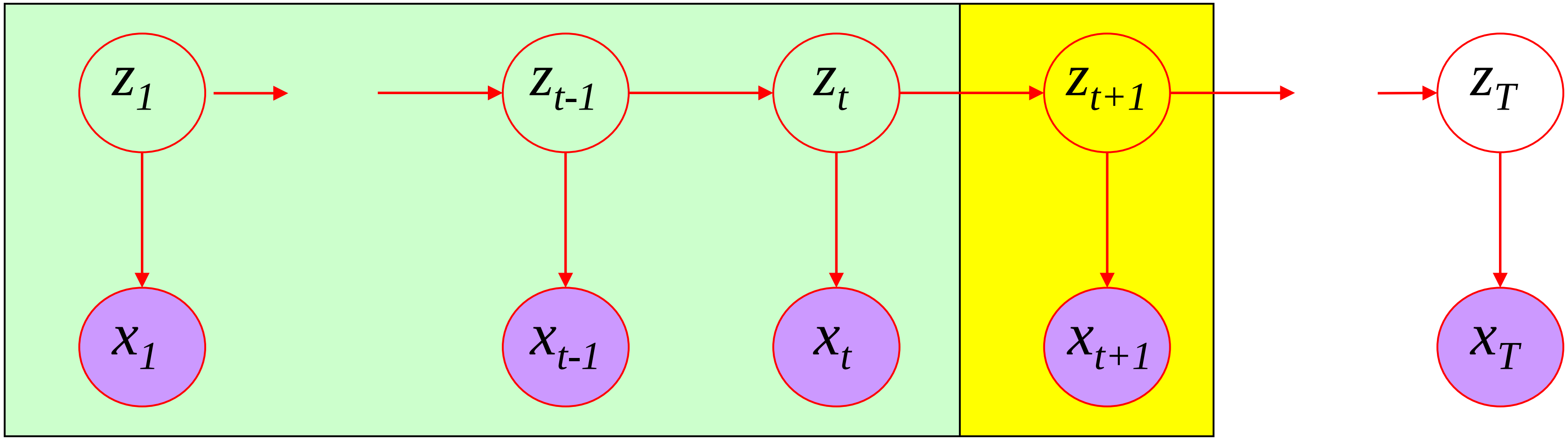
$$= P(x_1 \ldots x_{t+1}, z_{t+1} = j)$$

$$= P(x_1 \ldots x_{t+1} \mid z_{t+1} = j) P(z_{t+1} = j)$$

$$= P(x_1 \ldots x_t \mid z_{t+1} = j) P(x_{t+1} \mid z_{t+1} = j) P(z_{t+1} = j)$$

$$= P(x_1 \ldots x_t, z_{t+1} = j) P(x_{t+1} \mid z_{t+1} = j)$$

Forward Procedure



$$\alpha_j(t+1)$$

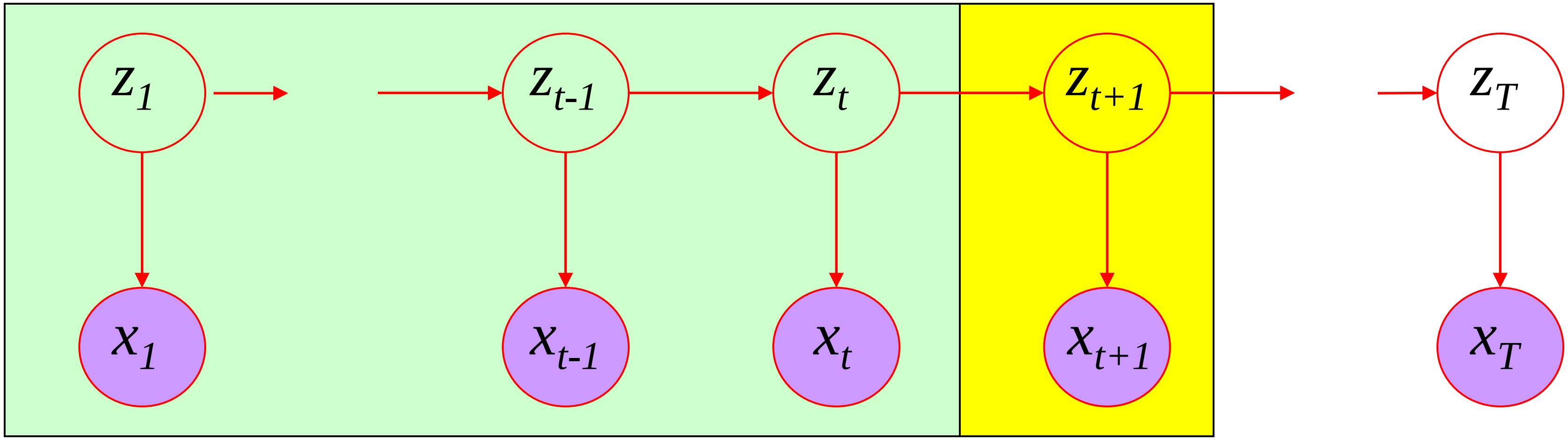
$$= P(x_1 \ldots x_{t+1}, z_{t+1} = j)$$

$$= P(x_1 \ldots x_{t+1} \mid z_{t+1} = j) P(z_{t+1} = j)$$

$$= P(x_1 \ldots x_t \mid z_{t+1} = j) P(x_{t+1} \mid z_{t+1} = j) P(z_{t+1} = j)$$

$$= P(x_1 \ldots x_t, z_{t+1} = j) P(x_{t+1} \mid z_{t+1} = j)$$

Forward Procedure



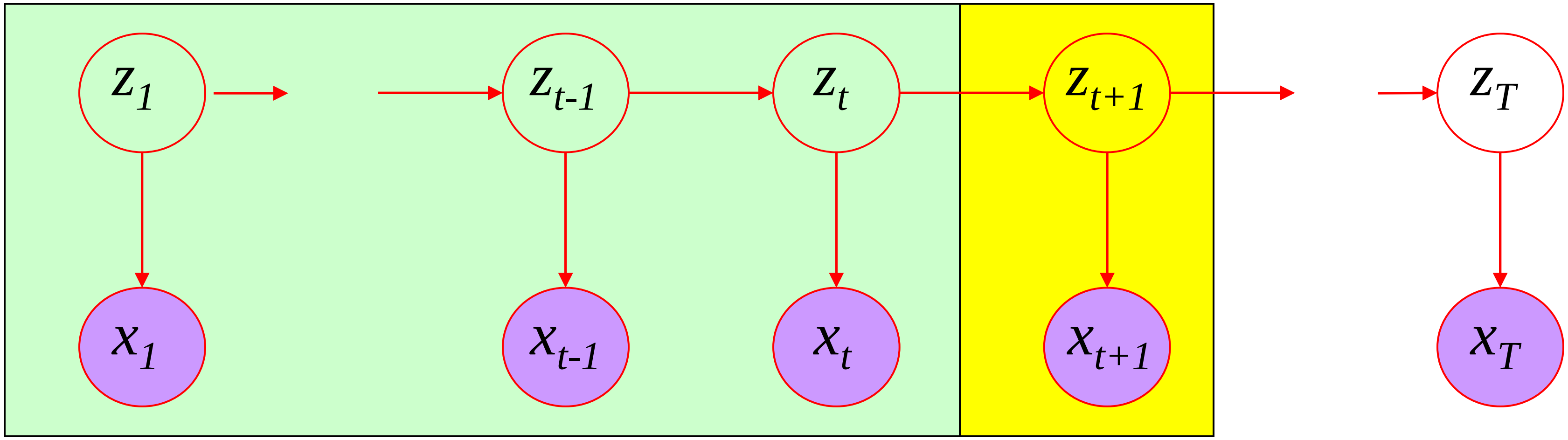
$$= \sum_{i=1\ldots N} P(x_1 \ldots x_t, z_t = i, z_{t+1} = j) P(x_{t+1} \mid z_{t+1} = j)$$

$$= \sum_{i=1\ldots N} P(x_1 \ldots x_t, z_{t+1} = j \mid z_t = i) P(z_t = i) P(x_{t+1} \mid z_{t+1} = j)$$

$$= \sum_{i=1\ldots N} P(x_1 \ldots x_t, z_t = i) P(z_{t+1} = j \mid z_t = i) P(x_{t+1} \mid z_{t+1} = j)$$

$$= \sum_{i=1\ldots N} \alpha_i(t) a_{ij} b_{j x_{t+1}}$$

Forward Procedure



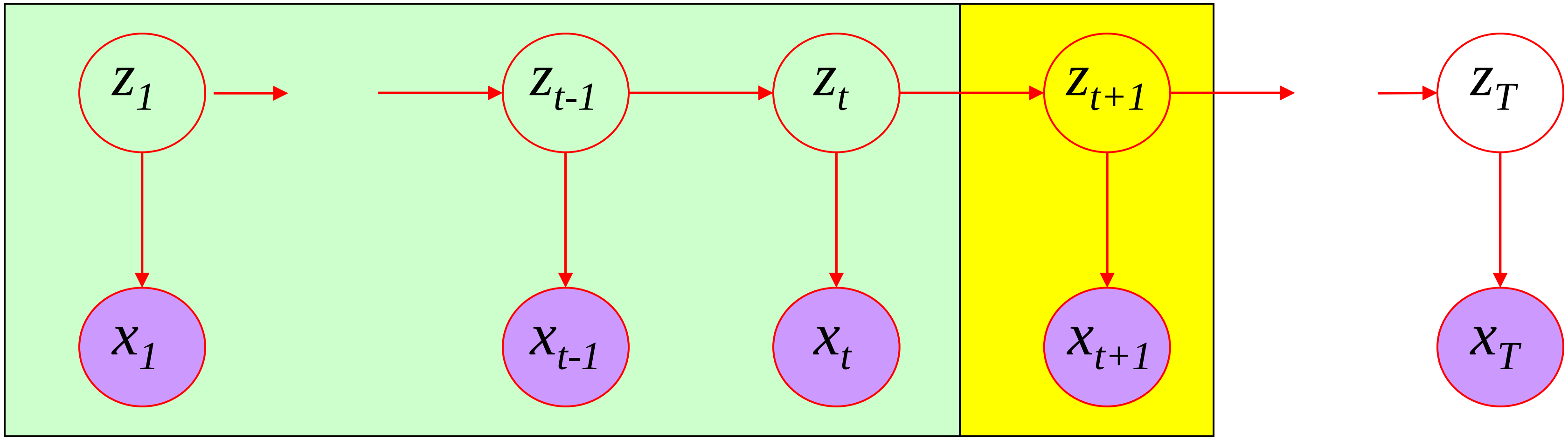
$$= \sum_{i=1\ldots N} P(x_1 \ldots x_t, z_t = i, z_{t+1} = j) P(x_{t+1} \mid z_{t+1} = j)$$

$$= \sum_{i=1\ldots N} P(x_1 \ldots x_t, z_{t+1} = j \mid z_t = i) P(z_t = i) P(x_{t+1} \mid z_{t+1} = j)$$

$$= \sum_{i=1\ldots N} P(x_1 \ldots x_t, z_t = i) P(z_{t+1} = j \mid z_t = i) P(x_{t+1} \mid z_{t+1} = j)$$

$$= \sum_{i=1\ldots N} \alpha_i(t) a_{ij} b_{j x_{t+1}}$$

Forward Procedure



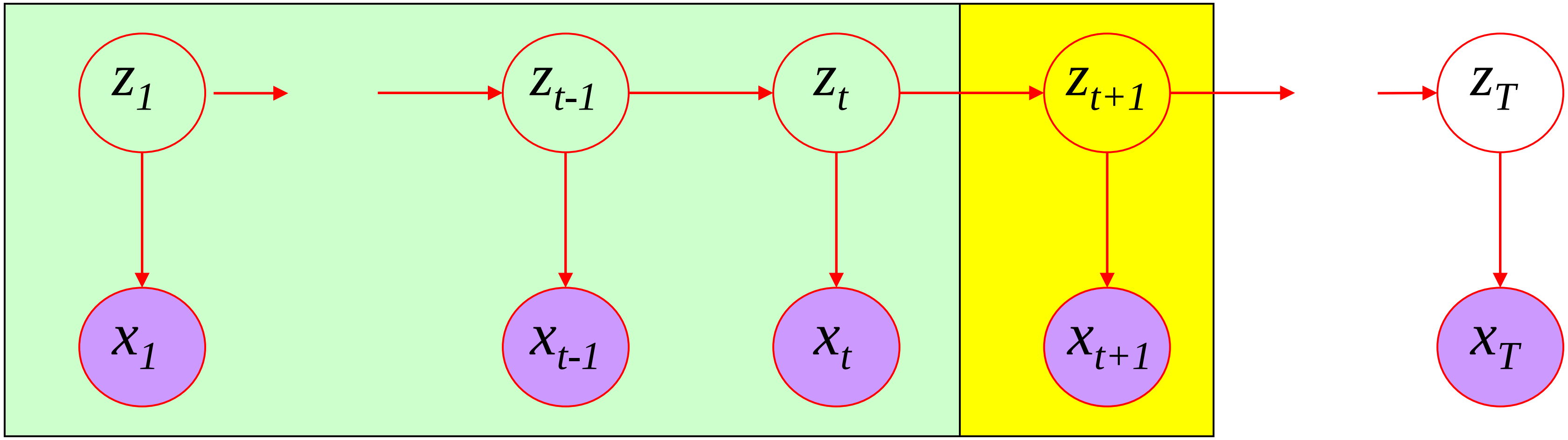
$$= \sum_{i=1\ldots N} P(x_1 \ldots x_t, z_t = i, z_{t+1} = j) P(x_{t+1} \mid z_{t+1} = j)$$

$$= \sum_{i=1\ldots N} P(x_1 \ldots x_t, z_{t+1} = j \mid z_t = i) P(z_t = i) P(x_{t+1} \mid z_{t+1} = j)$$

$$= \sum_{i=1\ldots N} P(x_1 \ldots x_t, z_t = i) P(z_{t+1} = j \mid z_t = i) P(x_{t+1} \mid z_{t+1} = j)$$

$$= \sum_{i=1\ldots N} \alpha_i(t) a_{ij} b_{j x_{t+1}}$$

Forward Procedure



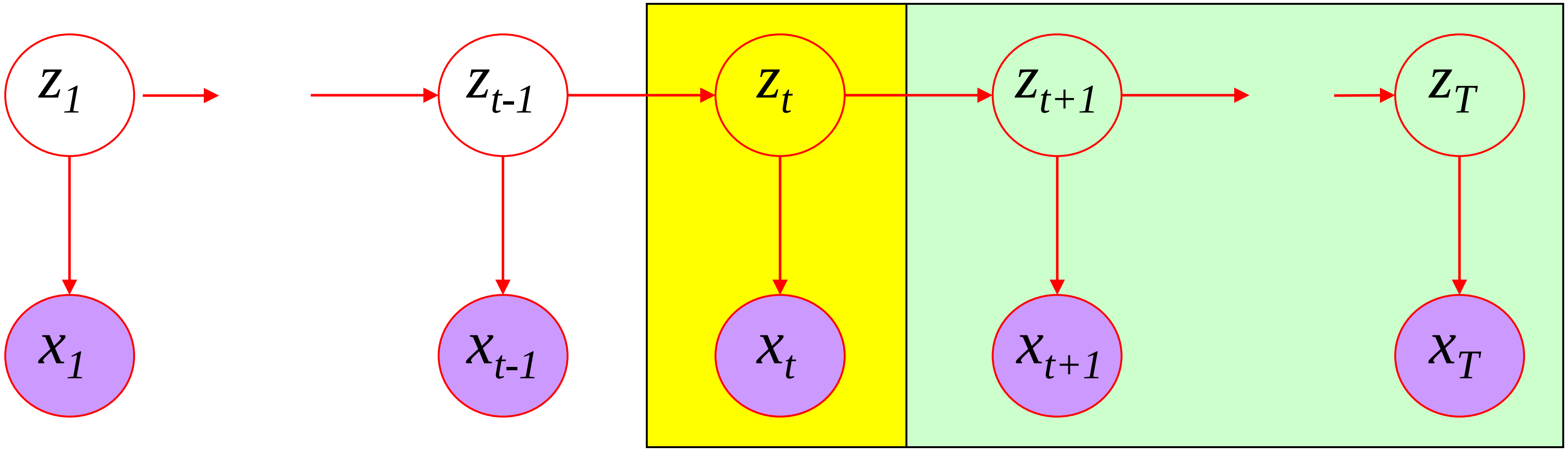
$$= \sum_{i=1 \ldots N} P(x_1 \ldots x_t, z_t = i, z_{t+1} = j) P(x_{t+1} | z_{t+1} = j)$$

$$= \sum_{i=1 \ldots N} P(x_1 \ldots x_t, z_{t+1} = j | z_t = i) P(z_t = i) P(x_{t+1} | z_{t+1} = j)$$

$$= \sum_{i=1 \ldots N} P(x_1 \ldots x_t, z_t = i) P(z_{t+1} = j | z_t = i) P(x_{t+1} | z_{t+1} = j)$$

$$= \sum_{i=1 \ldots N} \alpha_i(t) a_{ij} b_{j x_{t+1}}$$

Backward Procedure



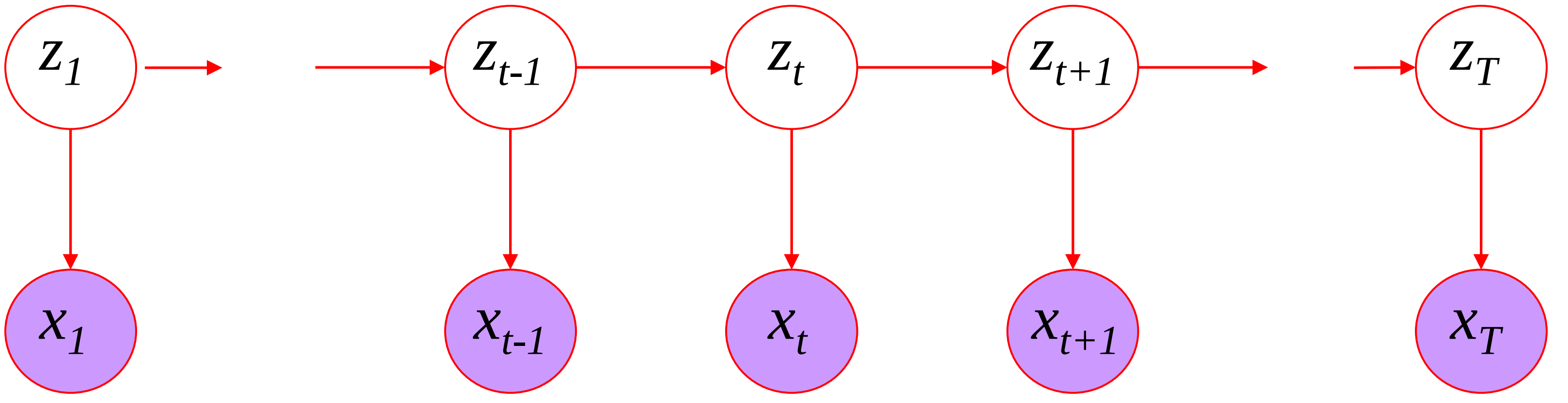
$$\beta_i(T+1) = 1$$

$$\beta_i(t) = P(x_t \ldots x_T \mid z_t = i)$$

$$\beta_i(t) = \sum_{j=1\ldots N} a_{ij} b_{ix_t} \beta_j(t+1)$$

Probability of the rest of the states given the first state

Decoding Solution



$$P(X \mid \theta) = \sum_{i=1}^{N} \alpha_i(T)$$

Forward Procedure

$$P(X \mid \theta) = \sum_{i=1}^{N} \pi_i \beta_i(1)$$

Backward Procedure

$$P(X \mid \theta) = \sum_{i=1}^{N} \alpha_i(t) \beta_i(t)$$

Combination

Best State Sequence

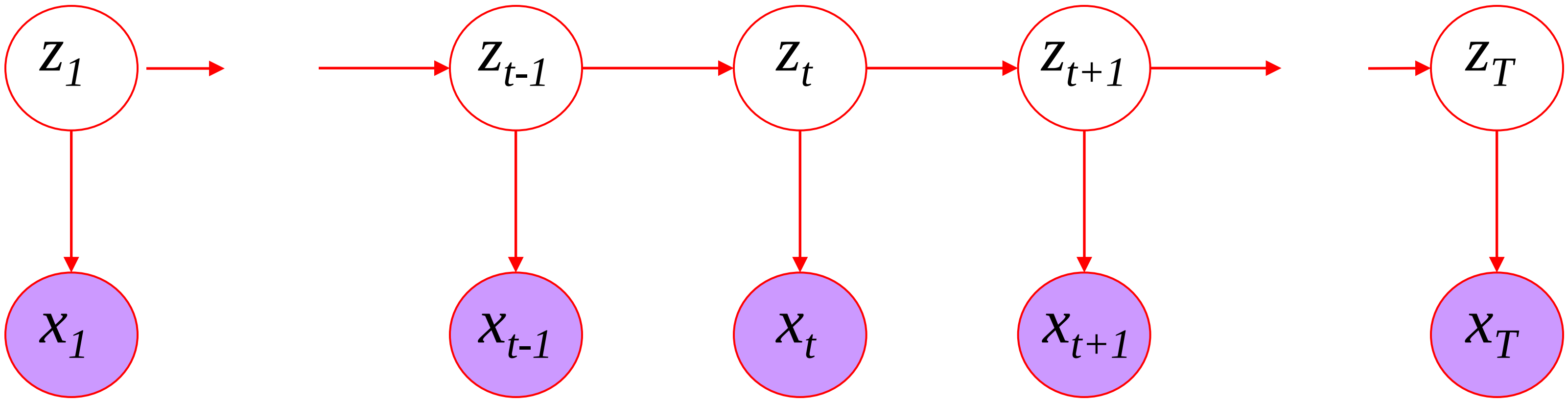
- Find the state sequence that best explains the observations

$$\arg\max_Z P(Z \mid X; A, B) =$$

$$\arg\max_Z \frac{P(X, Z; A, B)}{\sum_Z P(X, Z; A, B)} = \arg\max_Z P(X, Z; A, B)$$

- Viterbi Algorithm: same as forward procedure except that instead of tracking the total probability, we track the maximum probability and record its corresponding state sequence.

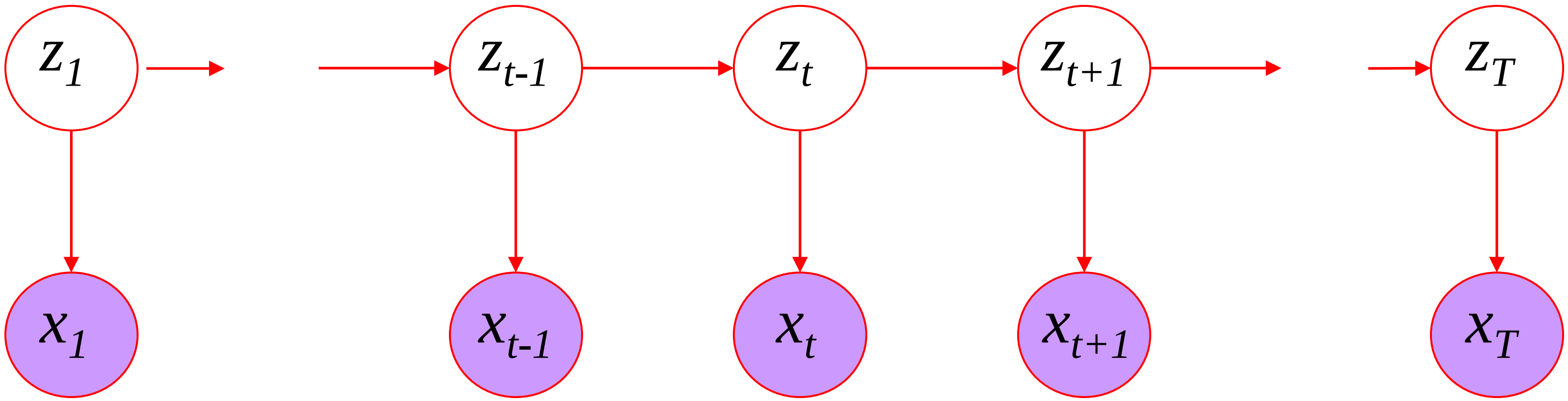
Viterbi Algorithm



$$\delta_j(t) = \max_{z_1 \ldots z_{t-1}} P(z_1 \ldots z_{t-1}, x_1 \ldots x_{t-1}, z_t = j, x_t)$$

The state sequence which maximizes the probability of seeing the observations to time t-1, landing in state j, and seeing the observation at time t

Viterbi Algorithm



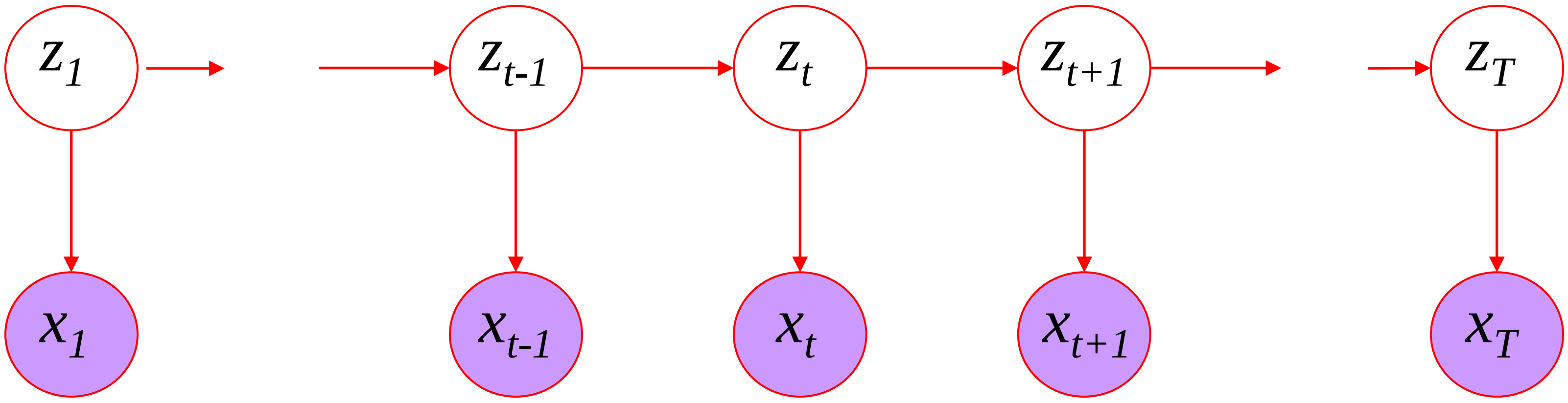
$$\delta_j(t) = \max_{z_1 \ldots z_{t-1}} P(z_1 \ldots z_{t-1}, x_1 \ldots x_{t-1}, z_t = j, x_t)$$

$$\delta_j(t+1) = \max_i \delta_i(t) a_{ij} b_{j x_{t+1}}$$

$$\psi_j(t+1) = \arg\max_i \delta_i(t) a_{ij} b_{j x_{t+1}}$$

Recursive Computation

Viterbi Algorithm



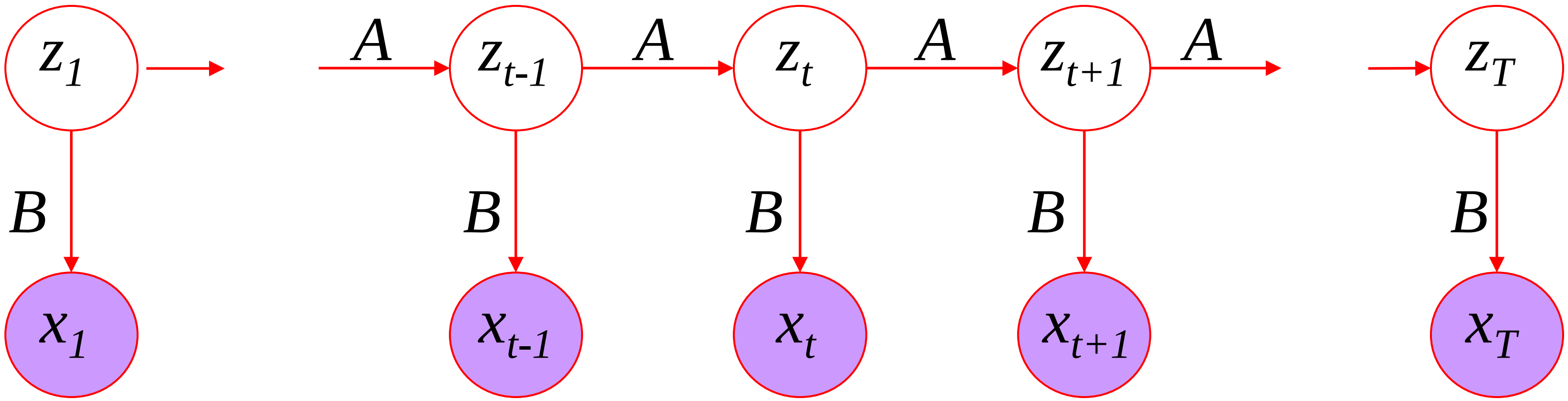
$$\hat{Z}_T = \arg\max_i \delta_i(T)$$

$$\hat{Z}_t = \psi_{\hat{Z}_{t+1}}(t+1)$$

$$P(\hat{Z}) = \arg\max_i \delta_i(T)$$

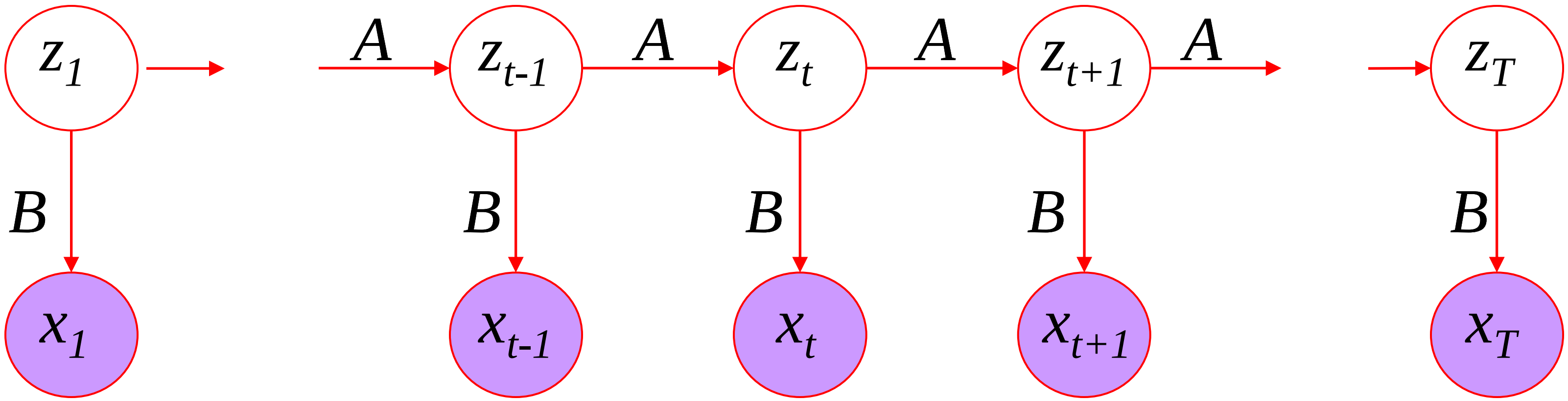
Compute the most likely state sequence by working backwards

Parameter Estimation



- Given an observation sequence, find the model that is most likely to produce that sequence.
- No analytic method -> EM

Parameter Estimation: E-step



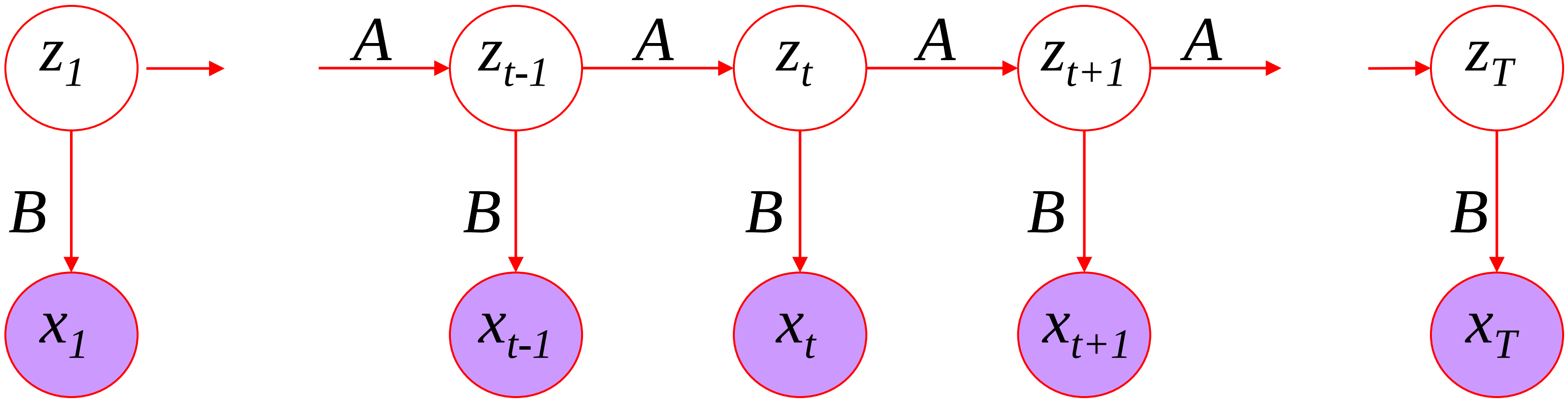
$$p_t(i,j) = \frac{\alpha_i(t) a_{ij} b_{j x_{t+1}} \beta_j(t+1)}{\sum_{m=1 \ldots N} \alpha_m(t) \beta_m(t)}$$

Probability of traversing an arc

$$\gamma_i(t) = \sum_{j=1 \ldots N} p_t(i,j)$$

Probability of being in state *i*

Parameter Estimation: M-step



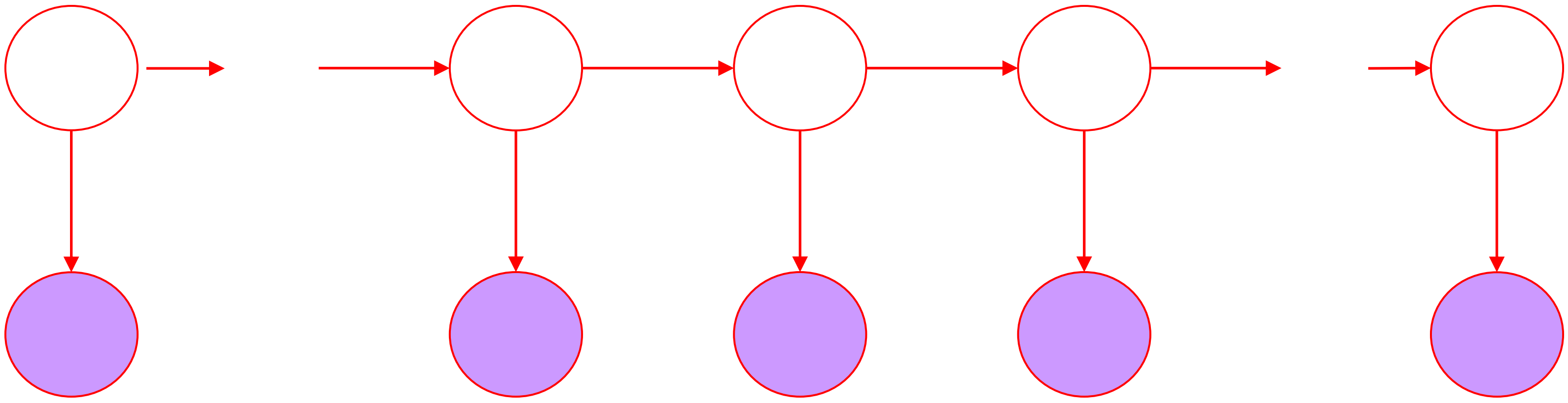
$$\hat{\pi}_i = \gamma_i(1)$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T} p_t(i,j)}{\sum_{t=1}^{T} \gamma_i(t)}$$

$$\hat{b}_{ik} = \frac{\sum_{\{t:x_t=k\}} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_i(t)}$$

Now we can compute the new estimates of the model parameters.

HMM Applications



- Analysis of biological sequences
- Tagging speech
- Speech recognition
- Many others