

The forward-backward algorithm

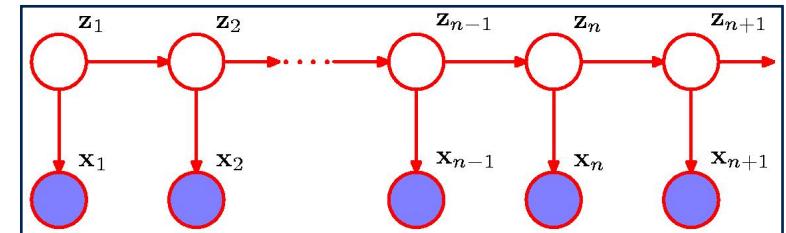
Sargur Srihari
srihari@buffalo.edu

HMM Topics

1. What is an HMM?
2. State-space Representation
3. HMM Parameters
4. Generative View of HMM
5. Determining HMM Parameters Using EM
6. Forward-Backward or $\alpha-\beta$ algorithm
7. HMM Implementation Issues:
 - a) Length of Sequence
 - b) Predictive Distribution
 - c) Sum-Product Algorithm
 - d) Scaling Factors
 - e) Viterbi Algorithm

Forward-Backward Algorithm

- E step: efficient procedure to evaluate $\gamma(z_n)$ and $\xi(z_{n-1}, z_n)$
- Graph of HMM, a tree →
 - Implies that posterior distribution of latent variables can be obtained efficiently using message passing algorithm
- In HMM it is called *forward-backward* algorithm or *Baum-Welch Algorithm*
- Several variants lead to exact marginals
 - Method called *alpha-beta* discussed here



Derivation of Forward-Backward

- Several conditional-independences (A-H) hold

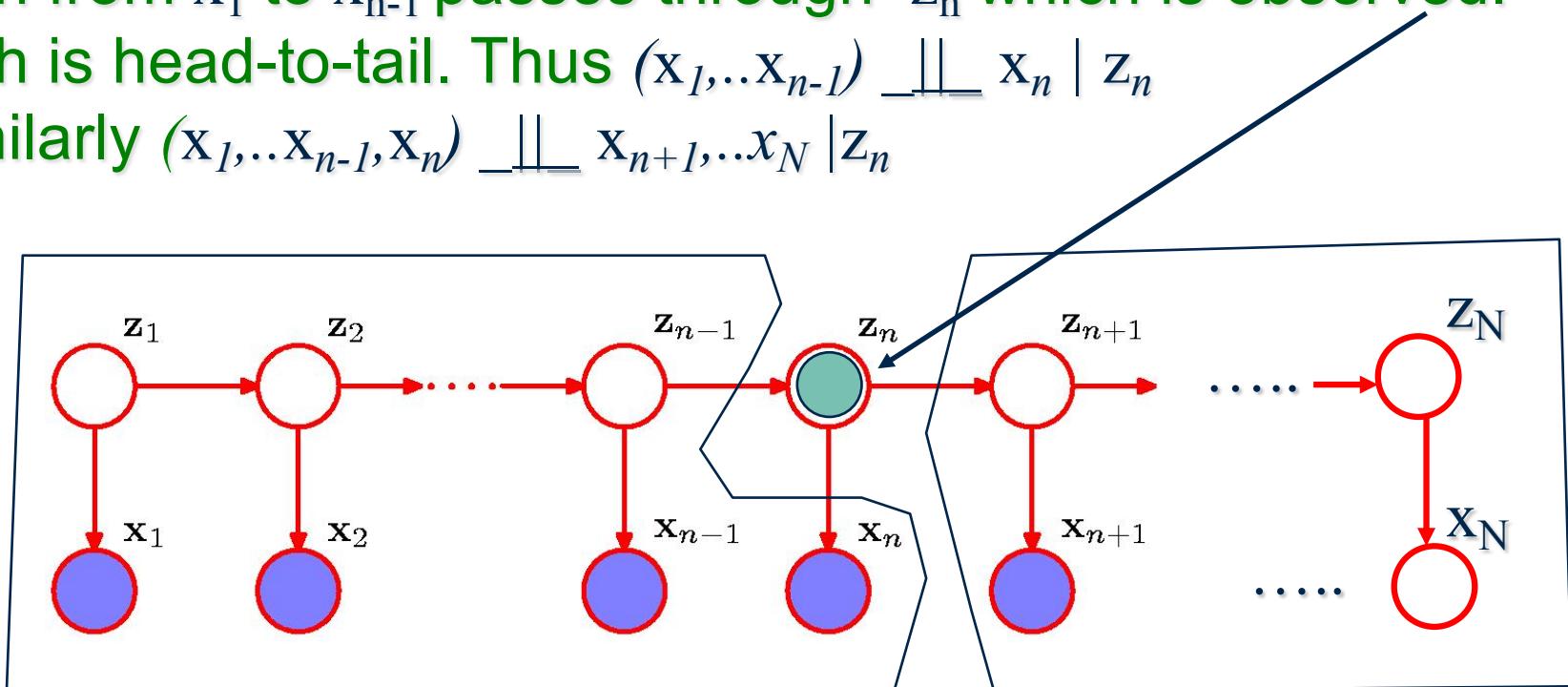
A.
$$p(X|z_n) = p(x_1, \dots, x_n | z_n) p(x_{n+1}, \dots, x_N | z_n)$$

- Proved using d-separation:

Path from x_1 to x_{n-1} passes through z_n which is observed.

Path is head-to-tail. Thus $(x_1, \dots, x_{n-1}) \perp\!\!\!\perp x_n | z_n$

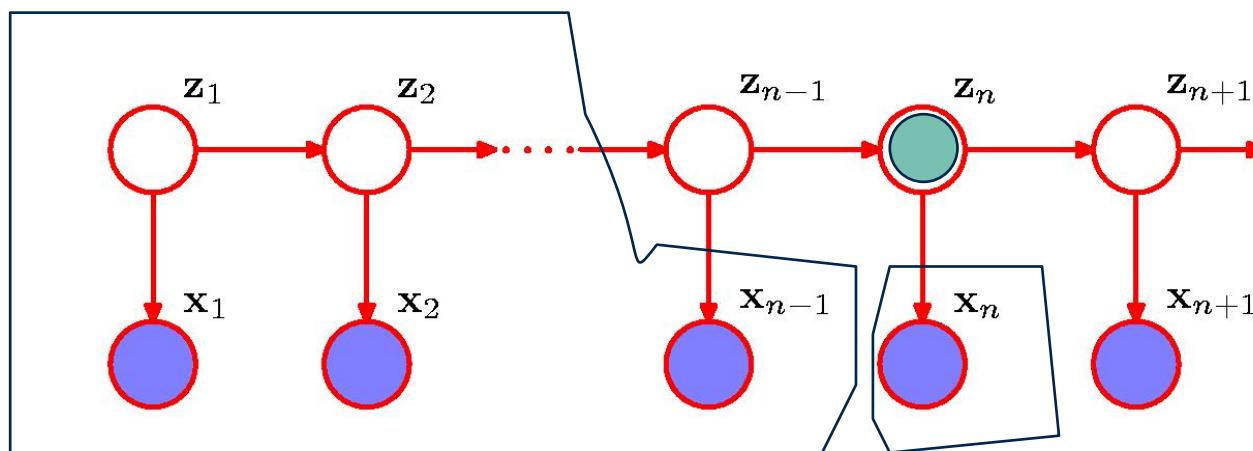
Similarly $(x_1, \dots, x_{n-1}, x_n) \perp\!\!\!\perp x_{n+1}, \dots, x_N | z_n$



Conditional independence B

- Since $(x_1, \dots, x_{n-1}) \perp\!\!\!\perp x_n | z_n$ we have

B. $p(x_1, \dots, x_{n-1} | x_n, z_n) = p(x_1, \dots, x_{n-1} | z_n)$

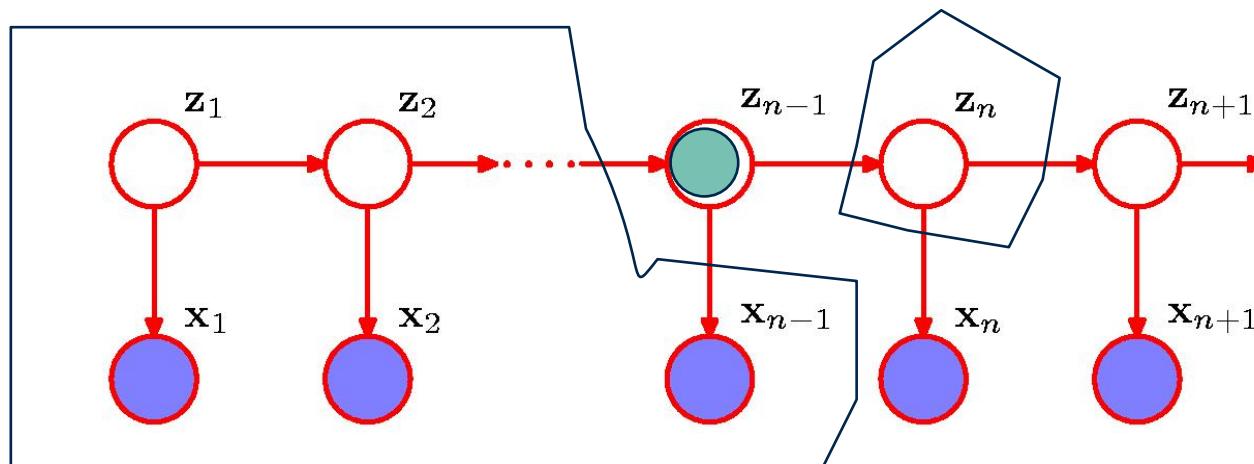


Conditional independence C

- Since

$$(x_1, \dots, x_{n-1}) \perp\!\!\!\perp z_n \mid z_{n-1}$$

C. $p(x_1, \dots, x_{n-1} \mid z_{n-1}, z_n) = p(x_1, \dots, x_{n-1} \mid z_{n-1})$

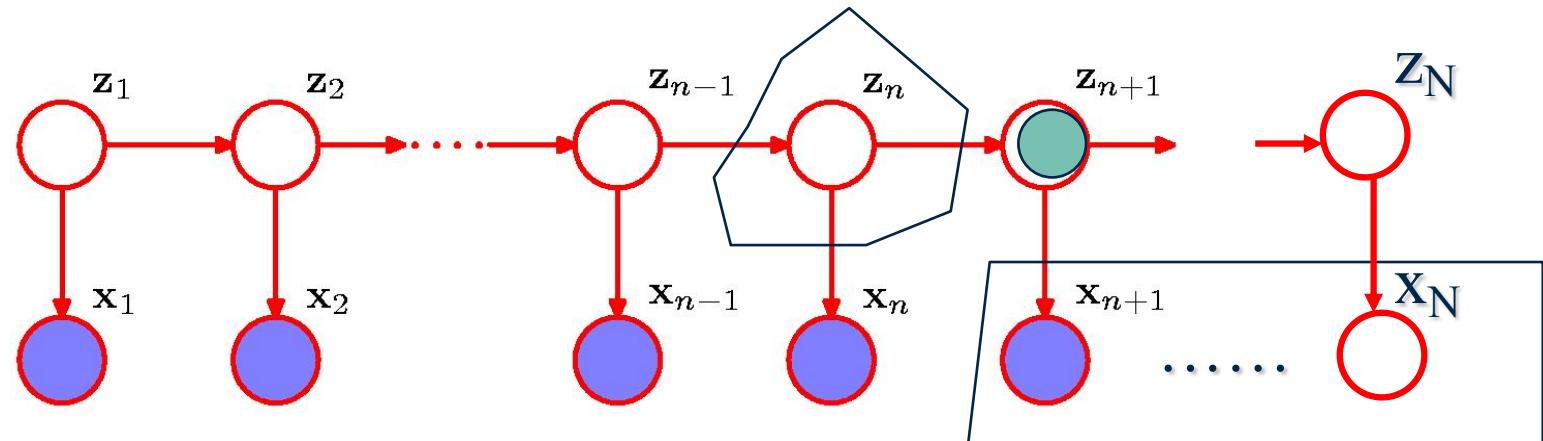


Conditional independence D

- Since

$$(x_{n+1}, \dots, x_N) \perp\!\!\!\perp z_n \mid z_{n+1}$$

D. $p(x_{n+1}, \dots, x_N \mid z_n, z_{n+1}) = p(x_1, \dots, x_N \mid z_{n+1})$

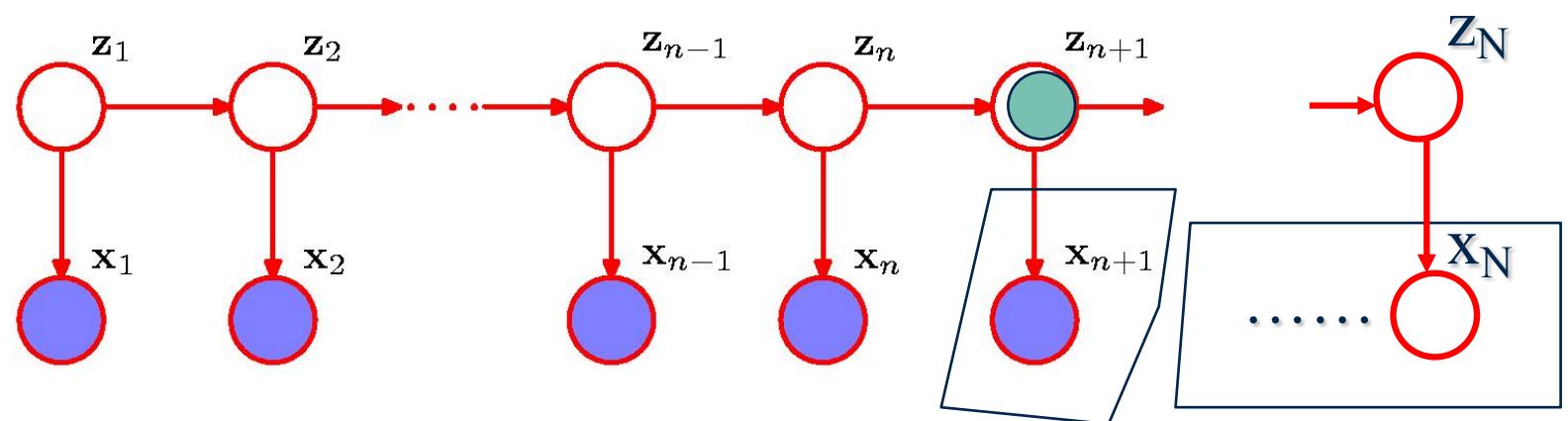


Conditional independence E

- Since

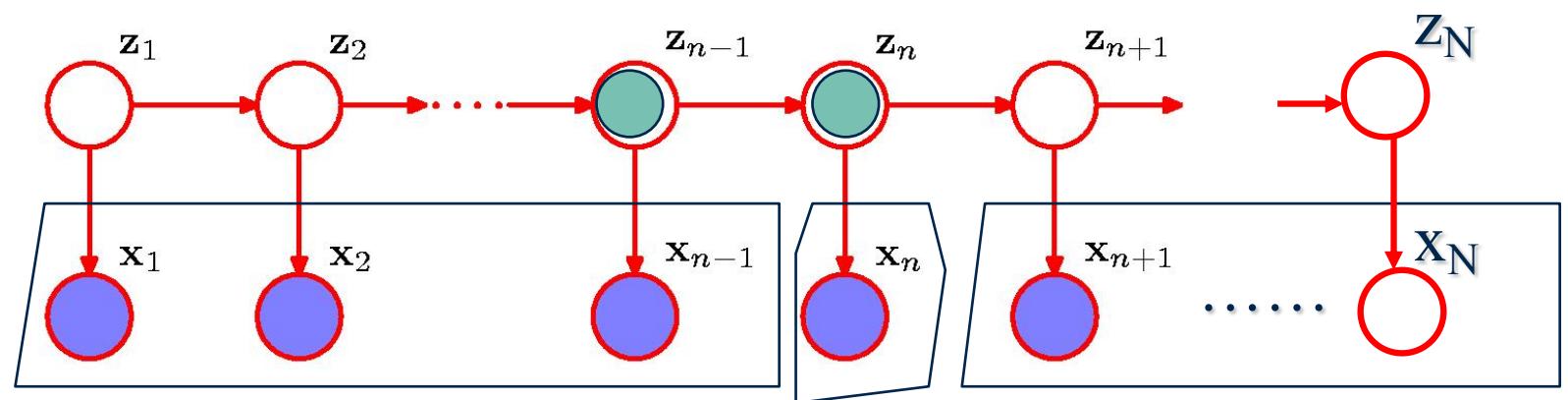
$$(x_{n+2}, \dots, x_N) \perp\!\!\!\perp z_n \mid z_{n+1}$$

E. $p(x_{n+2}, \dots, x_N \mid z_{n+1}, x_{n+1}) = p(x_{n+2}, \dots, x_N \mid z_{n+1})$



Conditional independence F

$$\text{F. } p(\mathbf{X} | \mathbf{z}_{n-1}, \mathbf{z}_n) = p(x_1, \dots, x_{n-1} | z_{n-1}) p(x_n | z_n) \\ p(x_{n+1}, \dots, x_N | z_n)$$

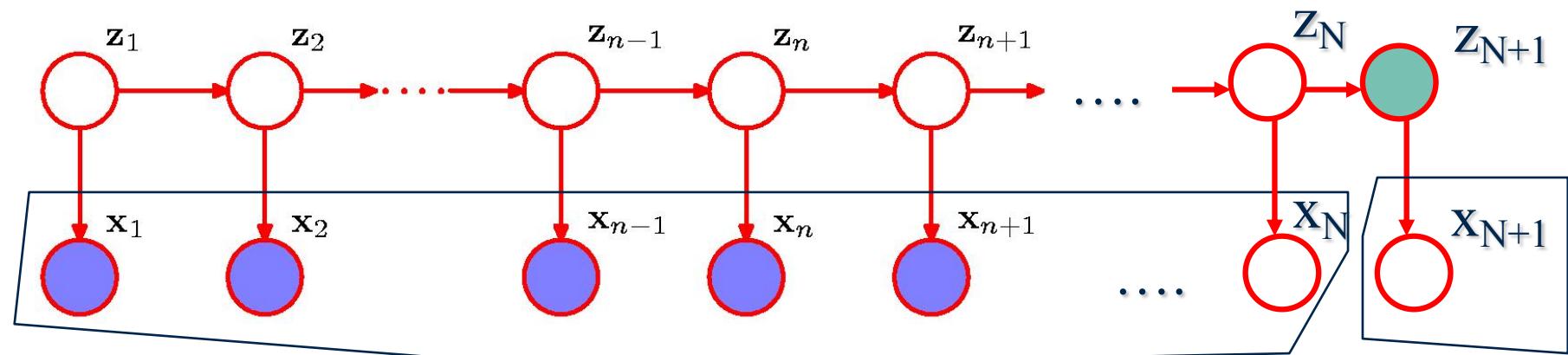


Conditional independence G

Since

$$(x_1, \dots, x_N) \perp\!\!\!\perp x_{N+1} \mid z_{N+1}$$

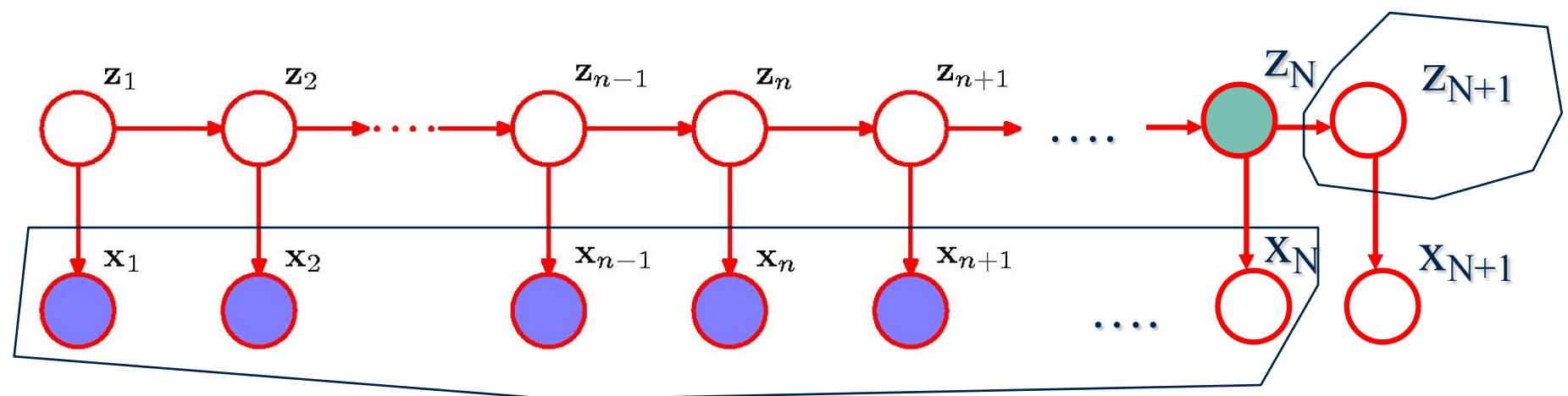
G. $p(x_{N+1} \mid X, z_{N+1}) = p(x_{N+1} \mid z_{N+1})$



Conditional independence H

H.

$$p(z_{N+1}|z_N, X) = p(z_{N+1}|z_N)$$



Evaluation of $\gamma(z_n)$

- Recall that this is to efficiently compute the E step of estimating parameters of HMM
 $\gamma(z_n) = p(z_n|X, \theta^{old})$: Marginal posterior distribution of latent variable z_n
- We are interested in finding posterior distribution $p(z_n|x_1,..x_N)$
- This is a vector of length K whose entries correspond to expected values of z_{nk}

Introducing alpha and beta

- Using Bayes theorem $\gamma(z_n) = p(z_n | X) = \frac{p(X | z_n)p(z_n)}{p(X)}$
- Using conditional independence A

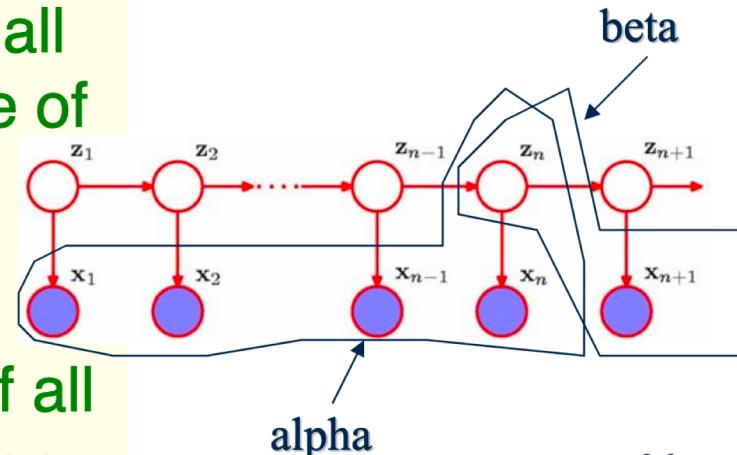
$$\begin{aligned}\gamma(z_n) &= \frac{p(x_1, \dots, x_n | z_n)p(x_{n+1}, \dots, x_N | z_n)p(z_n)}{p(X)} \\ &= \frac{p(x_1, \dots, x_n, z_n)p(x_{n+1}, \dots, x_N | z_n)}{p(X)} = \boxed{\frac{\alpha(z_n)\beta(z_n)}{p(X)}}\end{aligned}$$

- where $\alpha(z_n) \equiv p(x_1, \dots, x_n, z_n)$

which is the probability of observing all given data up to time n and the value of z_n

$$\beta(z_n) \equiv p(x_{n+1}, \dots, x_N | z_n)$$

which is the conditional probability of all future data from time $n+1$ up to N given the value of z_n



36

Recursion relation for alpha

$$\alpha(z_n) = p(x_1, \dots, x_n, z_n)$$

$$= \underline{p(x_1, \dots, x_n | z_n)} p(z_n) \text{ by Bayes rule}$$

$$= \underline{p(x_n | z_n)} p(x_1, \dots, x_{n-1} | z_n) p(z_n) \text{ by conditional independence B}$$

$$= p(x_n | z_n) p(x_1, \dots, x_{n-1}, z_n) \text{ by Bayes rule}$$

$$= p(x_n | z_n) \sum_{z_{n-1}} p(x_1, \dots, x_{n-1}, z_{n-1}, z_n) \text{ by Sum Rule}$$

$$= p(x_n | z_n) \sum_{z_{n-1}} p(x_1, \dots, x_{n-1}, z_n | z_{n-1}) p(z_{n-1}) \text{ by Bayes rule}$$

$$= p(x_n | z_n) \sum_{z_{n-1}} p(x_1, \dots, x_{n-1} | z_{n-1}) p(z_n | z_{n-1}) p(z_{n-1}) \text{ by cond. ind. C}$$

$$= p(x_n | z_n) \sum_{z_{n-1}} p(x_1, \dots, x_{n-1}, z_{n-1}) p(z_n | z_{n-1}) \text{ by Bayes rule}$$

$$= p(x_n | z_n) \sum_{z_{n-1}} \alpha(z_{n-1}) p(z_n | z_{n-1}) \text{ by definition of } \alpha$$

Forward recursion for alpha evaluation

- Recursion Relation is

$$\alpha(z_n) = p(x_n | z_n) \sum_{z_{n-1}} \alpha(z_{n-1}) p(z_n | z_{n-1})$$

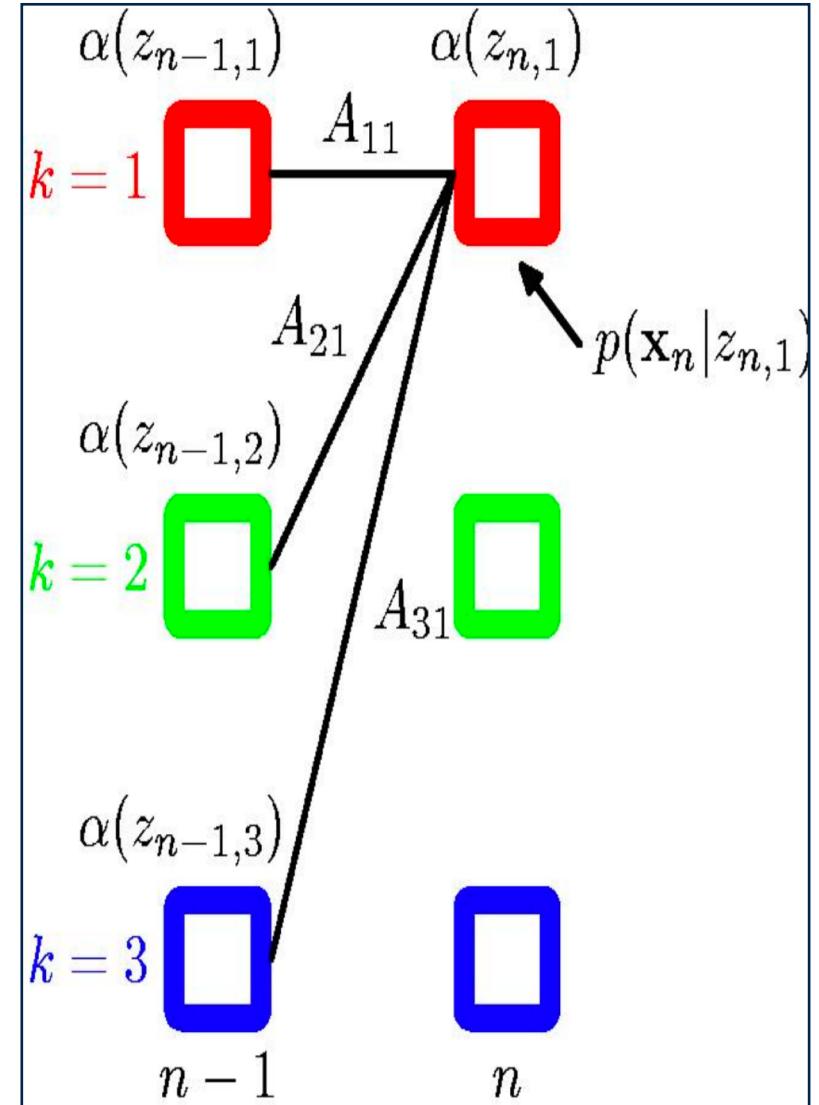
- There are K terms in the summation

- Has to be evaluated for each of K values of z_n
- Each step of recursion is $O(K^2)$

- Initial condition is

$$\alpha(z_1) = p(x_1, z_1) = p(z_1) p(x_1 | z_1) = \prod_{k=1}^K \{\pi_k p(x_1 | \phi_k)\}^{z_{1k}}$$

- Overall cost for the chain is $O(K^2 N)$



Recursion relation for beta

$$\begin{aligned}\beta(z_n) &= p(x_{n+1}, \dots, x_N | z_n) \\&= \sum_{z_{n+1}} p(x_{n+1}, \dots, x_N, z_{n+1} | z_n) \text{ by Sum Rule} \\&= \sum_{z_{n+1}} p(x_{n+1}, \dots, x_N | z_n, z_{n+1}) p(z_{n+1} | z_n) \text{ by Bayes rule} \\&= \sum_{z_{n+1}} p(x_{n+1}, \dots, x_N | z_{n+1}) p(z_{n+1} | z_n) \text{ by Cond. ind. D} \\&= \sum_{z_{n+1}} p(x_{n+2}, \dots, x_N | z_{n+1}) p(x_{n+1} | z_{n+1}) p(z_{n+1} | z_n) \text{ by Cond. ind. E} \\&= \sum_{z_{n+1}} \beta(z_{n+1}) p(x_{n+1} | z_{n+1}) p(z_{n+1} | z_n) \text{ by definition of } \beta\end{aligned}$$

Backward recursion for beta

- Backward message passing

$$\beta(z_n) = \sum_{z_{n+1}} \beta(z_{n+1}) p(x_{n+1}|z_n) p(z_{n+1}|z_n)$$

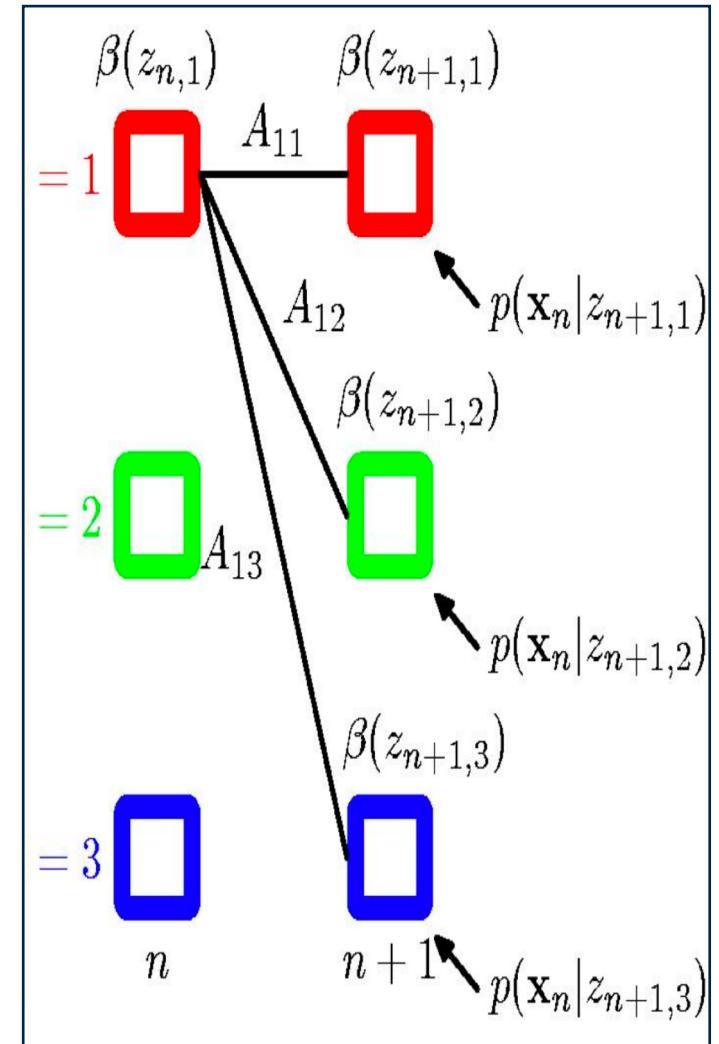
- Evaluates $\beta(z_n)$ in terms of $\beta(z_{n+1})$

- Starting condition for recursion is

$$p(z_N | X) = \frac{p(X, z_N) \beta(z_N)}{p(X)}$$

- Is correct provided we set $\beta(z_N) = 1$ for all settings of z_N

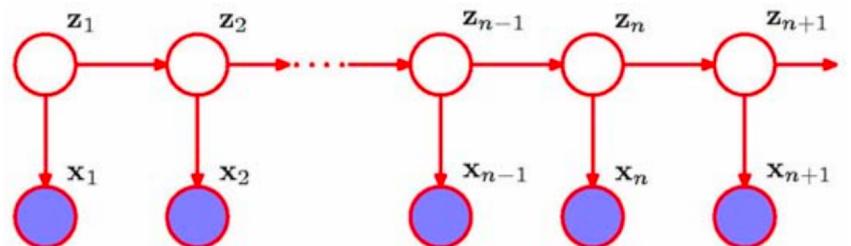
- This is the initial condition for backward computation



M-step Equations

- In the M-step equations $p(x)$ will cancel out

$$\mu_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) x_n}{\sum_{n=1}^N \gamma(z_{nk})}$$



$$p(X) = \sum_{z_n} \alpha(z_n) \beta(z_n)$$

Evaluation of Quantities $\xi(z_{n-1}, z_n)$

- They correspond to the values of the conditional probabilities $p(z_{n-1}, z_n | X)$ for each of the $K \times K$ settings for (z_{n-1}, z_n)

$$\xi(z_{n-1}, z_n) = p(z_{n-1}, z_n | X) \text{ by definition}$$

$$= \frac{p(X | z_{n-1}, z_n) p(z_{n-1}, z_n)}{p(X)} \text{ by Bayes Rule}$$

$$= \frac{p(x_1, \dots, x_{n-1} | z_{n-1}) p(x_n | z_n) p(x_{n+1}, \dots, x_N | z_n) p(z_n | z_{n-1}) p(z_{n-1})}{p(X)} \text{ by cond ind F}$$

$$= \frac{\alpha(z_{n-1}) p(x_n | z_n) p(z_n | z_{n-1}) \beta(z_n)}{p(X)}$$

- Thus we calculate $\xi(z_{n-1}, z_n)$ directly by using results of the α and β recursions

Summary of EM to train HMM

Step 1: Initialization

- Make an initial selection of parameters θ^{old} where $\theta = (\pi, A, \phi)$
 1. π is a vector of K probabilities of the states for latent variable z_1
 2. A is a $K \times K$ matrix of transition probabilities A_{ij}
 3. ϕ are parameters of conditional distribution $p(x_k|z_k)$
- A and π parameters are often initialized uniformly
- Initialization of ϕ depends on form of distribution
 - For Gaussian:
 - parameters μ_k initialized by applying K-means to the data, Σ_k corresponds to covariance matrix of cluster

Summary of EM to train HMM

Step 2: E Step

- Run both forward α recursion and backward β recursion
- Use results to evaluate $\gamma(z_n)$ and $\xi(z_{n-1}, z_n)$ and the likelihood function

Step 3: M Step

- Use results of E step to find revised set of parameters θ^{new} using M-step equations

Alternate between E and M

until convergence of likelihood function

Values for $p(x_n|z_n)$

- In recursion relations, observations enter through conditional distributions $p(x_n|z_n)$
- Recursions are independent of
 - Dimensionality of observed variables
 - Form of conditional distribution
 - So long as it can be computed for each of K possible states of z_n
- Since observed variables $\{x_n\}$ are fixed they can be pre-computed at the start of the EM algorithm

Length of Sequence

- HMM can be trained effectively if length of sequence is sufficiently long
 - True of all maximum likelihood approaches
- Alternatively we can use multiple short sequences
 - Requires straightforward modification of HMM-EM algorithm
- Particularly important in left-to-right models
 - In given observation sequence, a given state transition for a non-diagonal element of A occurs only once

Predictive Distribution

- Observed data is $X = \{x_1, \dots, x_N\}$
- Wish to predict x_{N+1}
- Application in financial forecasting

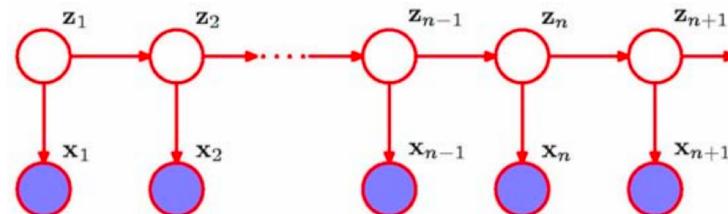
$$\begin{aligned} p(x_{N+1} | X) &= \sum_{z_{N+1}} p(x_{N+1}, z_{N+1} | X) \\ &= \sum_{z_{N+1}} p(x_{N+1} | z_{N+1} | X) p(z_{N+1} | X) \text{ by Product Rule} \\ &= \sum_{z_{N+1}} p(x_{N+1} | z_{N+1}) \sum_{z_N} p(z_{N+1}, z_N | X) \text{ by Sum Rule} \\ &= \sum_{z_{N+1}} p(x_{N+1} | z_{N+1}) \sum_{z_N} p(z_{N+1} | z_N) p(z_N | X) \text{ by conditional ind H} \\ &= \sum_{z_{N+1}} p(x_{N+1} | z_{N+1}) \sum_{z_N} p(z_{N+1} | z_N) \frac{p(z_N, X)}{p(X)} \text{ by Bayes rule} \\ &= \frac{1}{p(X)} \sum_{z_{N+1}} p(x_{N+1} | z_{N+1}) \sum_{z_N} p(z_{N+1} | z_N) \alpha(z_N) \text{ by definition of } \alpha \end{aligned}$$

- Can be evaluated by first running forward α recursion and summing over z_N and z_{N+1}
- Can be extended to subsequent predictions of x_{N+2} , after x_{N+1} is observed, using a fixed amount of storage

Sum-Product and HMM

- HMM graph is a tree and hence *sum-product* algorithm can be used to find local marginals for hidden variables
 - Equivalent to forward-backward algorithm
 - Sum-product provides a simple way to derive alpha-beta recursion formulae
- Transform directed graph to factor graph
 - Each variable has a node, small squares represent factors, undirected links connect factor nodes to variables used

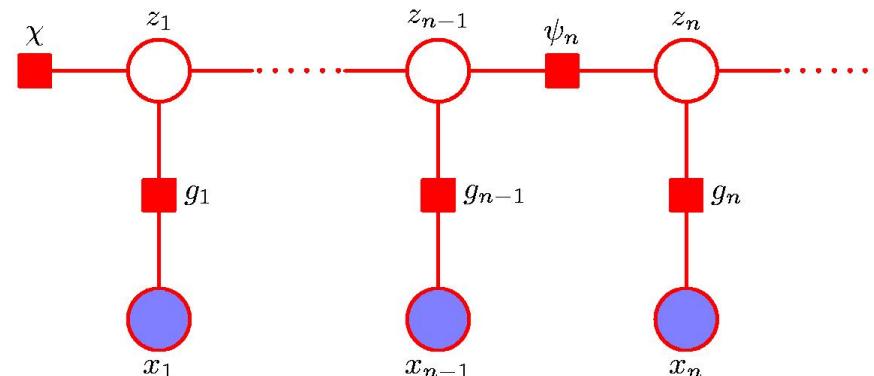
HMM Graph



Joint distribution

$$p(x_1, \dots, x_N, z_1, \dots, z_n) = p(z_1) \left[\prod_{n=2}^N p(z_n | z_{n-1}) \right] \prod_{n=1}^N p(x_n | z_n)$$

Fragment of Factor Graph



Deriving alpha-beta from Sum-product

- Begin with simplified form of factor graph
- Factors are given by

$$h(z_1) = p(z_1)p(x_1 | z_1)$$

$$f_n(z_{n-1}, z_n) = p(z_n | z_{n-1})p(x_n | z_n)$$

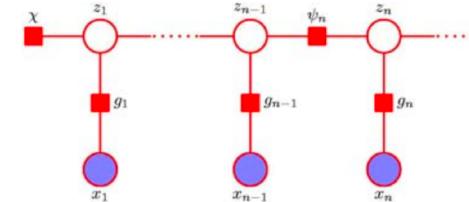
- Messages propagated are

$$\mu_{z_{n-1} \rightarrow f_n}(z_{n-1}) = \mu_{f_n \rightarrow z_{n-1}}(z_{n-1})$$

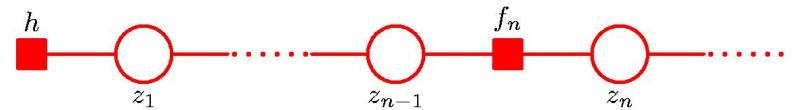
$$\mu_{f_n \rightarrow z_n}(z_n) = \sum_{z_{n-1}} f_n(z_{n-1}, z_n) \mu_{z_{n-1} \rightarrow f_n}(z_{n-1})$$

- Can show that α recursion is computed
- Similarly starting with the root node β recursion is computed
- So also γ and ξ are derived

Fragment of Factor Graph



Simplified by absorbing emission probabilities into transition probability factors



Final Results

$$\alpha(z_n) = p(x_n | z_n) \sum_{z_{n-1}} \alpha(z_{n-1}) p(z_n | z_{n-1})$$

$$\beta(z_n) = \sum_{z_{n+1}} \beta(z_{n+1}) p(x_{n+1} | z_n) p(z_{n+1} | z_n)$$

$$\gamma(z_n) = \frac{\alpha(z_n) \beta(z_n)}{p(X)}$$

$$\xi(z_{n-1}, z_n) = \frac{\alpha(z_{n-1}) p(x_n | z_n) p(z_n | z_{n-1}) \beta(z_n)}{p(X)}$$

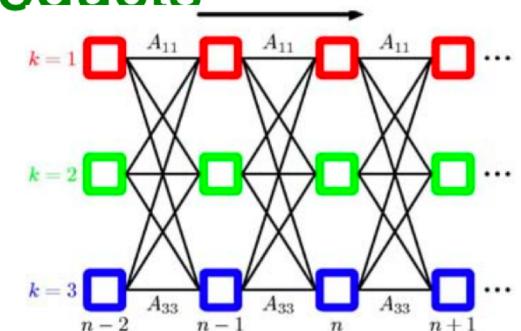
Scaling Factors

- Implementation issue for small probabilities
- At each step of recursion

$$\alpha(z_n) = p(x_n | z_n) \sum_{z_{n-1}} \alpha(z_{n-1}) p(z_n | z_{n-1})$$

- To obtain new value of $\alpha(z_n)$ from previous value $\alpha(z_{n-1})$ we multiply $p(z_n | z_{n-1})$ and $p(x_n | z_n)$
- These probabilities are small and products will underflow
- Logs don't help since we have sums of products

- Solution is rescaling
 - of $\alpha(z_n)$ and $\beta(z_n)$ whose values remain close to unity

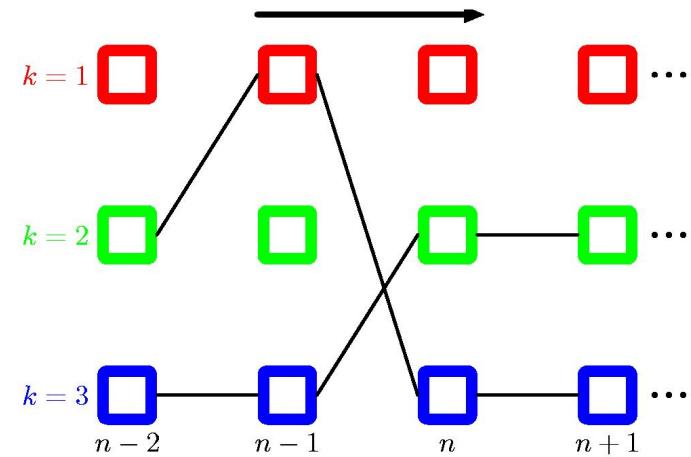


The Viterbi Algorithm

- Finding most probable sequence of hidden states for a given sequence of observables
- In speech recognition: finding most probable phoneme sequence for a given series of acoustic observations
- Since graphical model of HMM is a tree, can be solved exactly using *max-sum* algorithm
 - Known as Viterbi algorithm in the context of HMM
 - Since max-sum works with log probabilities no need to work with re-scaled variables as with forward-backward

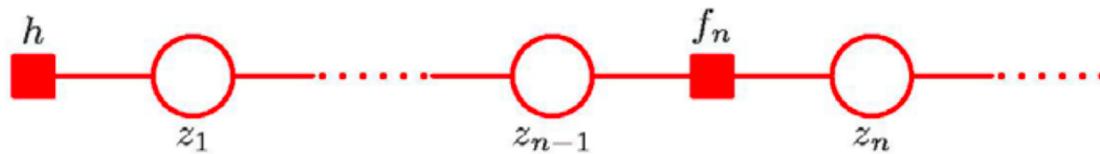
Viterbi Algorithm for HMM

- Fragment of HMM lattice showing two paths
- Number of possible paths grows exponentially with length of chain
- Viterbi searches space of paths efficiently
 - Finds most probable path with computational cost linear with length of chain



Deriving Viterbi from Max Sum

- Start with simplified factor graph



- Treat variable z_N as root node, passing messages to root from leaf nodes
- Messages passed are

$$\mu_{z_n \rightarrow f_{n+1}}(z_n) = \mu_{f_n \rightarrow z_n}(z_n)$$

$$\mu_{f_{n+1} \rightarrow z_{n+1}}(z_{n+1}) = \max_{z_n} \left\{ \ln f_{n+1}(z_n, z_{n+1}) + \mu_{z_n \rightarrow f_{n+1}}(z_n) \right\}$$