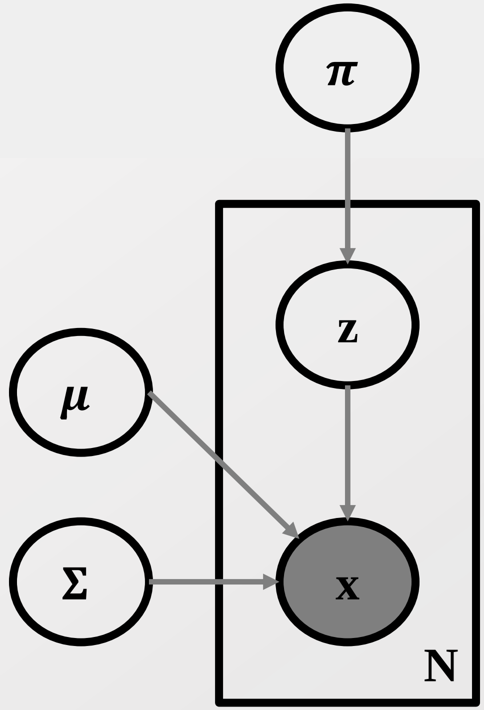
Hidden Markov Model

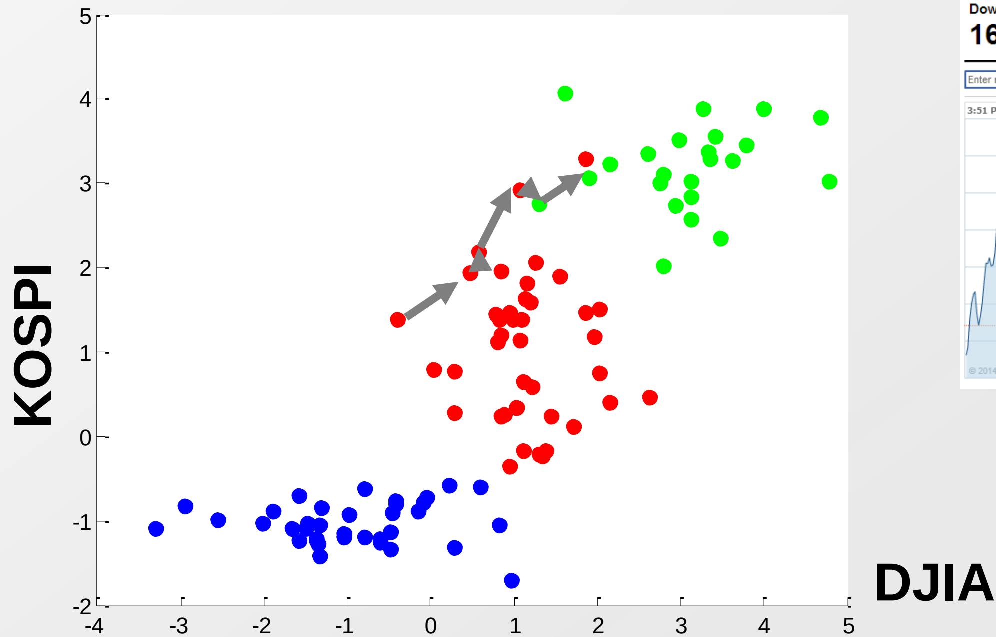
Il-Chul Moon
Department of Industrial and Systems Engineering
KAIST
icmoon@kaist.ac.kr

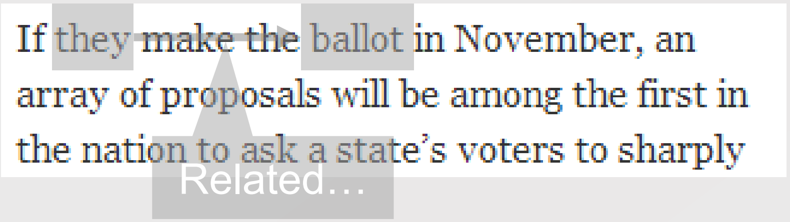
HIDDEN MARKOV MODEL

Times Series Data for GMM



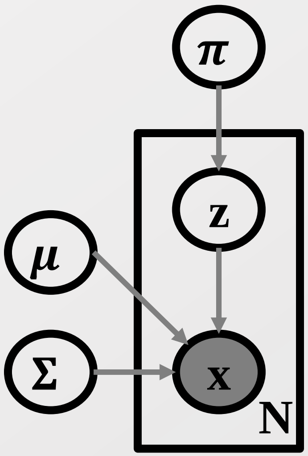
- Imagine the following case
  - Data points on the plane
  - Have a temporal trace of data points
  - Now, any broken assumption in the analysis?
- Any real world applications
  - Many, many, many…
  - Stock market analysis, text mining…

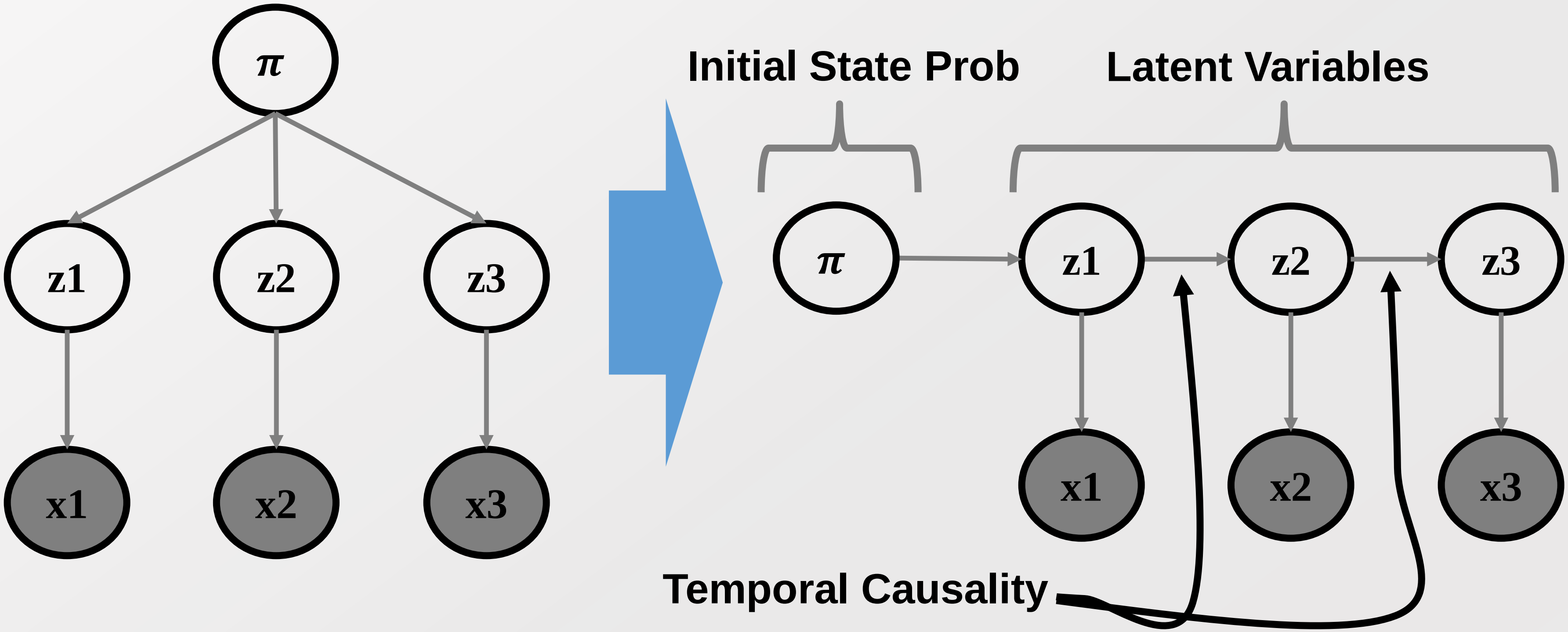




What to Model and How to Model

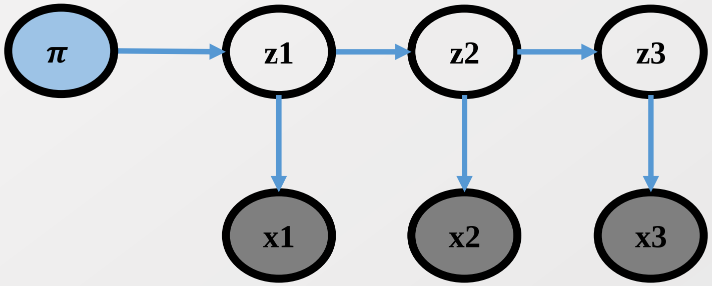
- Previously, all data points are independent trials
  - Now, they are not any further
- Temporal relation: causality from time t to time t+1
- Overall trend: latent state variables

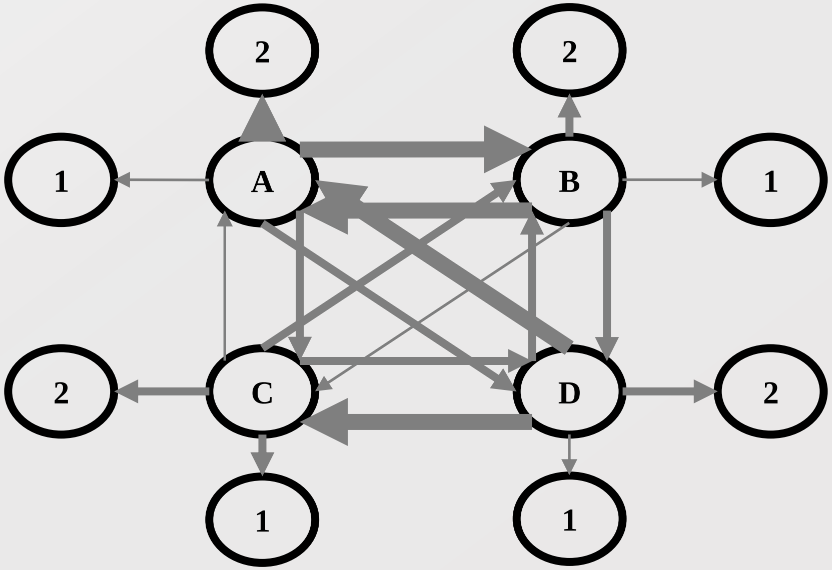




Hidden Markov Model

- Observation, *x*
  - Can be either discrete or continuous
    - Just a difference in probability distributions
    - Will only handle discrete case in this course
  - $x_1 \ldots x_T$: Observation from time 1 to time T
  - $x_i \in \{c_1, \ldots, c_m\}$: *m* types of observation values
- Latent state, z
  - Vector variable with K elements
    - Let's say that there are K types of state values corresponding to each element
  - Can be either discrete or continuous
    - If continuous → Kalman filter, and this is a continuous version of HMM
    - Out of scope of this course
- Initial State probabilities
  - $P(z_1) \sim Mult(\pi_1, \ldots, \pi_k)$
- Transition probabilities
  - $P(z_t | z_{t-1}^i = 1) \sim Mult(a_{i,1}, \ldots, a_{i,k})$
  - Or, $P\left(z_t^j = 1 \middle| z_{t-1}^i = 1\right) = a_{i,j}$
- Emission probabilities
  - $P(x_t | z_t^i = 1) \sim Mult(b_{i,1}, \ldots, b_{i,m}) \sim f(x_t | \theta_i)$
  - Or, $P\left(x_t^j = 1 \middle| z_t^i = 1\right) = b_{i,j}$
- **A stochastic generative model**





Main Questions on HMM

Initial State probabilities

$P(z_1) \sim Mult(\pi_1, \dots, \pi_k)$

Transition probabilities

$P(z_t | z_{t-1}^i = 1) \sim Mult(a_{i,1}, \dots, a_{i,k})$

Or, $P\left(z_t^j = 1 \middle| z_{t-1}^i = 1\right) = a_{i,j}$

Emission probabilities

$P(x_t | z_t^i = 1) \sim Mult(b_{i,1}, \dots, b_{i,m}) \sim f(x_t | \theta_i)$

Or, $P\left(x_t^j = 1 \middle| z_t^i = 1\right) = b_{i,j}$

- Given the topology of the Bayesian network, HMM, or M
- Evaluation question
  - Given $\boldsymbol{\pi}$, $\boldsymbol{a}$, $\boldsymbol{b}$, X
  - Find P(X|M, $\boldsymbol{\pi}$, $\boldsymbol{a}$, $\boldsymbol{b}$)
  - How much is X likely to be observed in the trained model?
- Decoding question
  - Given $\boldsymbol{\pi}$, $\boldsymbol{a}$, $\boldsymbol{b}$, X
  - Find $argmax_z P(Z|X, M, \boldsymbol{\pi}, \boldsymbol{a}, \boldsymbol{b})$
  - What would be the most probable sequences of latent states?
- Learning question
  - Given X
  - Find $argmax_{\pi,\ a,\ b} P(X|M, \boldsymbol{\pi}, \boldsymbol{a}, \boldsymbol{b})$
  - What would be the underlying parameters of the HMM given the observations?
- Decoding questions and learning questions are very similar to
  - Supervised and unsupervised learning
- Anyhow, we often need to find $\boldsymbol{\pi}, \boldsymbol{a}, \boldsymbol{b}$ prior to the supervised learning with X

Obtaining π, a, b given X and M

Initial State probabilities

$$P(z_1) \sim Mult(\pi_1, \dots, \pi_k)$$

Transition probabilities

$$P(z_t | z_{t-1}^i = 1) \sim Mult(a_{i,1}, \dots, a_{i,k})$$

Or, $P\left(z_t^j = 1 \middle| z_{t-1}^i = 1\right) = a_{i,j}$

Emission probabilities

$$P(x_t | z_t^i = 1) \sim Mult(b_{i,1}, \dots, b_{i,m}) \sim f(x_t | \theta_i)$$

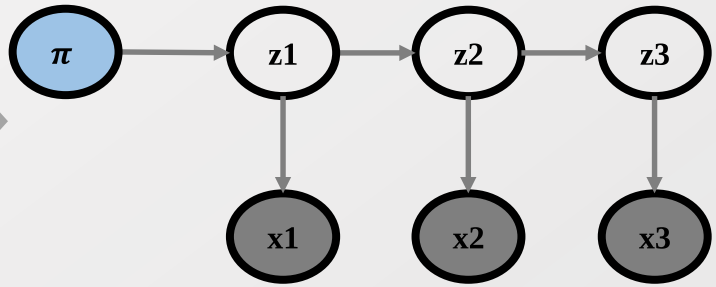
Or, $P\left(x_t^j = 1 \middle| z_t^i = 1\right) = b_{i,j}$

N seq.

LLLLLLLFFFFFFFF
12342531643242
……………
FFFLLFLFLFLLLLFLLLFL
35261524361524 36152

M$_i$ observations for i-th sequence

Counting, MLE, MAP…



- Finding π, a, b from the data in the supervised learning approach requires X as well as Z
- Example scenario
  - Loaded dice and fair dice
  - Two dices yield different probability distributions from one to six
  - Dealer changes the dice as he wishes
- Probability estimation
  - Use MLE, MAP and counting…
  - Find out
    - Dealer starts with a certain dice type: $P(z_1^L = 1) = 1/2$
    - Dealer switches the dice: $P(z_t^L = 1 | z_{t-1}^L = 1) = 0.7, P(z_t^L = 1 | z_{t-1}^F = 1) = 0.5$
    - Loaded dice: $P(X=1) = P(X=2) = P(X=3) = P(X=4) = P(X=5) = 1/10$, $P(X=6) = 1/2$
    - Fair dice: $P(X=1) = P(X=2) = P(X=3) = P(X=4) = P(X=5) = P(X=6) = 1/6$
- What if the X is continuous? Use a known distribution, and estimate its parameters

Joint Probability

- Let's assume that we have a training dataset with X and Z

Initial State probabilities

$P(z_1) \sim Mult(\pi_1, \dots, \pi_k)$

Transition probabilities

$P(z_t | z_{t-1}^i = 1) \sim Mult(a_{i,1}, \dots, a_{i,k})$

Or, $P\left(z_t^j = 1 \middle| z_{t-1}^i = 1\right) = a_{i,j}$

Emission probabilities

$P(x_t | z_t^i = 1) \sim Mult(b_{i,1}, \dots, b_{i,m}) \sim f(x_t | \theta_i)$

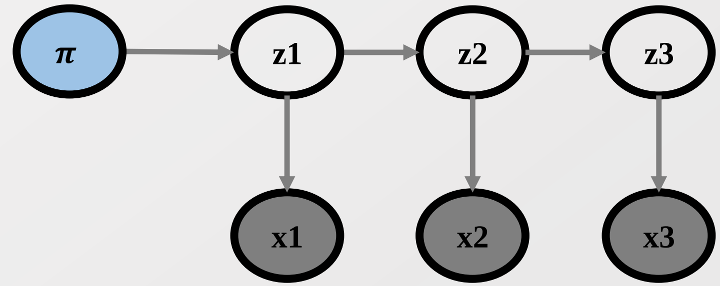
Or, $P\left(x_t^j = 1 \middle| z_t^i = 1\right) = b_{i,j}$

N seq.

LLLLLLLFFFFFFFF
12342531643242
……………
FFFLLFLFLFLLLLFLLLFL
35261524361524361 52

M_i observations for i-th sequence

- Can we compute the joint probability, $P(X, Z)$
  - Yes. Easily by the virtue of the network structure
- Anyway, a Bayesian network, so…
  - Factorize
  - $P(X, Z) = P(x_1, \dots, x_t, z_1, \dots, z_t)$
  - $= P(z_1)P(x_1|z_1)P(z_2|z_1)P(x_2|z_2)P(z_3|z_2)P(x_3|z_3)$
    - Nothing but a combination of initial, transition, and emission probabilities
  - $= \pi_{idx(z_1=1)} b_{idx(x_1=1),idx(z_1=1)} a_{idx(z_1=1),idx(z_2=1)} \cdots$



- Assume that we have 166 as X
  - Let's check Z=LLL and FFF
  - $P(166, LLL) = \frac{1}{2} \times \frac{1}{10} \times \frac{7}{10} \times \frac{1}{2} \times \frac{7}{10} \times \frac{1}{2} = 0.0061$
  - $P(166, FFF) = \frac{1}{2} \times \frac{1}{6} \times \frac{1}{2} \times \frac{1}{6} \times \frac{1}{2} \times \frac{1}{6} = 5.7870e - 04$
  - What about FLL, FFL, FLF……? Exponential combination to check

Marginal Probability

Initial State probabilities

$$P(z_1) \sim Mult(\pi_1, \dots, \pi_k)$$

Transition probabilities

$$P(z_t | z_{t-1}^i = 1) \sim Mult(a_{i,1}, \dots, a_{i,k})$$

Or, $P\left(z_t^j = 1 \middle| z_{t-1}^i = 1\right) = a_{i,j}$

Emission probabilities

$$P(x_t | z_t^i = 1) \sim Mult(b_{i,1}, \dots, b_{i,m}) \sim f(x_t | \theta_i)$$

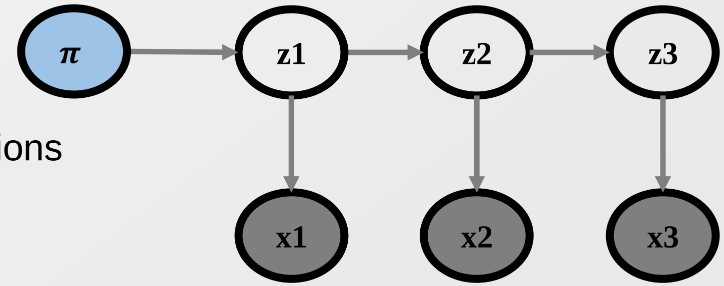
Or, $P\left(x_t^j = 1 \middle| z_t^i = 1\right) = b_{i,j}$

N seq.

LLLLLLLFFFFFFFF
12342531643242
……………
FFFLLFLFLFLLLLFLLLFL
35261524361524361 52

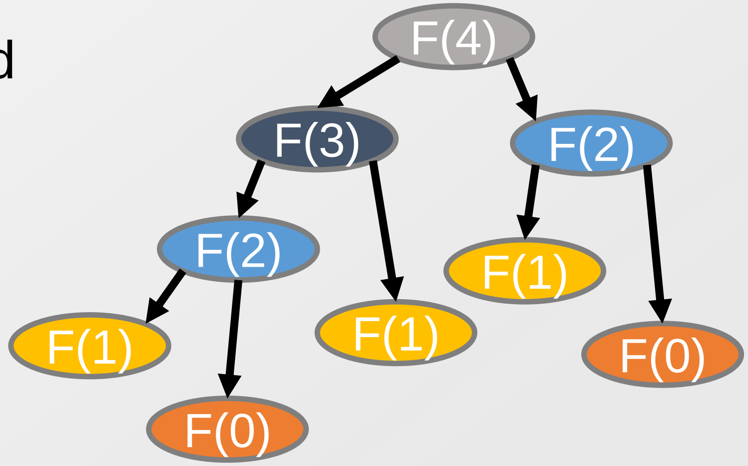
M_i observations for i-th sequence

- Eventually, we only want to use X and marginalize Z
  - Just like GMM,
    $P(X|\theta) = \sum_Z P(X, Z|\theta)$
  - In HMM,
    $P(X|\pi, a, b) = \sum_Z P(X, Z|\pi, a, b)$
- $P(X) = \sum_Z P(X, Z) = \sum_{z_1} \dots \sum_{z_t} P(x_1, \dots, x_t, z_1, \dots, z_t)$
- $= \sum_{z_1} \dots \sum_{z_t} \pi_{z_1} \prod_{t=2}^{T} a_{z_{t-1}, z_t} \prod_{t=1}^{T} b_{z_t, x_t}$
  - Many summations yield an exponential number of combinations
- Need to avoid a repetitive computing
  - Compute only necessary terms for a single time
  - Let's work on the formula
    - P(A,B,C)=P(A)P(B|A)P(C|A,B)
  - $P\left(x_1, \dots, x_t, z_t^k = 1\right) = \sum_{z_{t-1}} P\left(x_1, \dots, x_{t-1}, x_t, z_{t-1}, z_t^k = 1\right)$
  - $= \sum_{z_{t-1}} P(x_1, \dots, x_{t-1}, z_{t-1}) P\left(z_t^k = 1 | x_1, \dots, x_{t-1}, z_{t-1}\right) P\left(x_t | z_t^k = 1, x_1, \dots, x_{t-1}, z_{t-1}\right)$
    - By the virtue of the structure
  - $= \sum_{z_{t-1}} P(x_1, \dots, x_{t-1}, z_{t-1}) P\left(z_t^k = 1 | z_{t-1}\right) P\left(x_t | z_t^k = 1\right)$
  - $= P\left(x_t | z_t^k = 1\right) \sum_{z_{t-1}} P(x_1, \dots, x_{t-1}, z_{t-1}) P\left(z_t^k = 1 | z_{t-1}\right)$
  - $= b_{z_t^k, x_t} \sum_{z_{t-1}} P(x_1, \dots, x_{t-1}, z_{t-1}) a_{z_{t-1}, z_t^k}$
  - Now, we see a repeating structure of terms
  - $P\left(x_1, \dots, x_t, z_t^k = 1\right) = \alpha_t^k = b_{k, x_t} \sum_i \alpha_{t-1}^i a_{i,k}$



Detour: Dynamic Programming

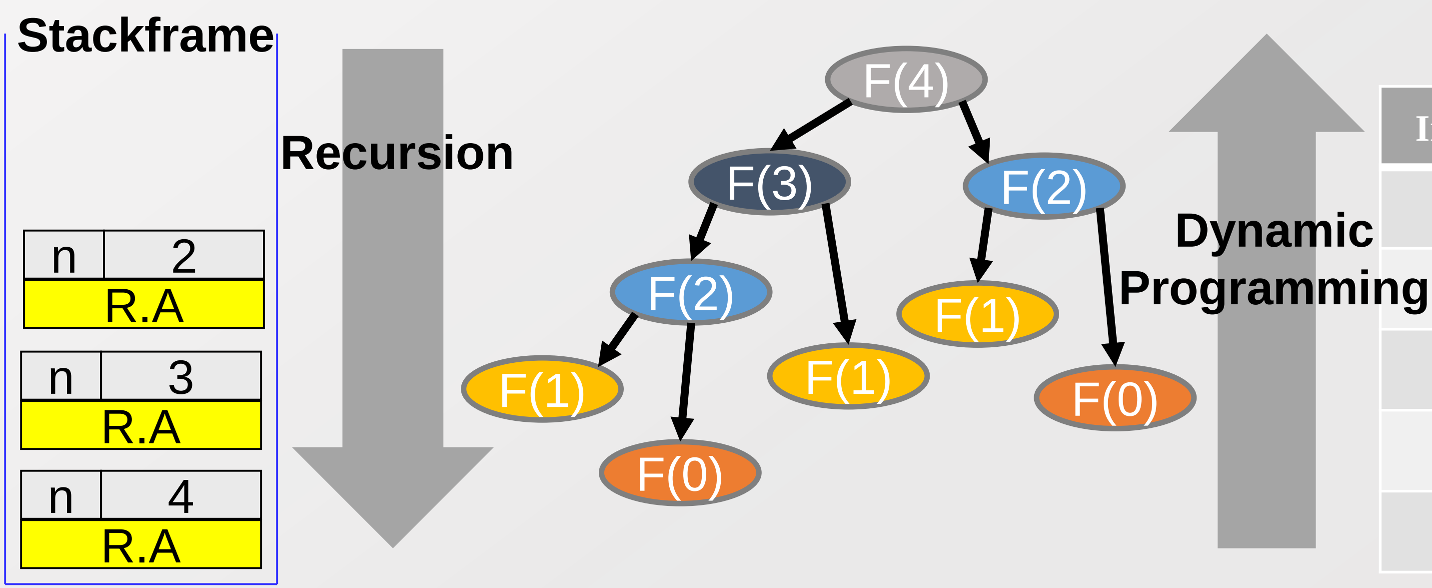
- Dynamic programming:
  - A general algorithm design technique for solving problems defined by or formulated as ***recurrences with overlapping sub-instances***
  - In this context, Programming == Planning
- Main storyline
  - Setting up a recurrence
    - Relating a solution of a larger instance to solutions of some smaller instances
    - Solve small instances once
    - Record solutions in a table
    - Extract a solution of a larger instance from the table



Instance	Solution
F(0)	0
F(1)	1
F(2)	1
F(3)	2
F(4)	?

Detour: Memoization

- Key technique of dynamic programming
  - Simply put
    - Storing the results of previous function calls to reuse the results again in the future
  - More philosophical sense
    - Bottom-up approach for problem-solving
      - Recursion: Top-down of divide and conquer
      - Dynamic programming: Bottom-up of storing and building

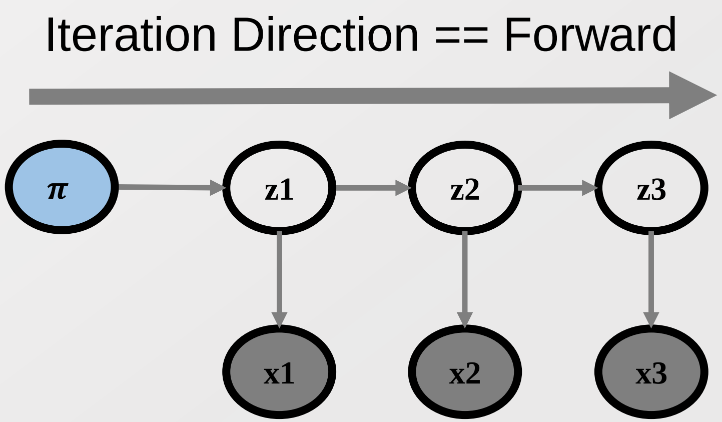


Memoization

Instance	Solution
F(0)	0
F(1)	1
F(2)	1
F(3)	2
F(4)	3

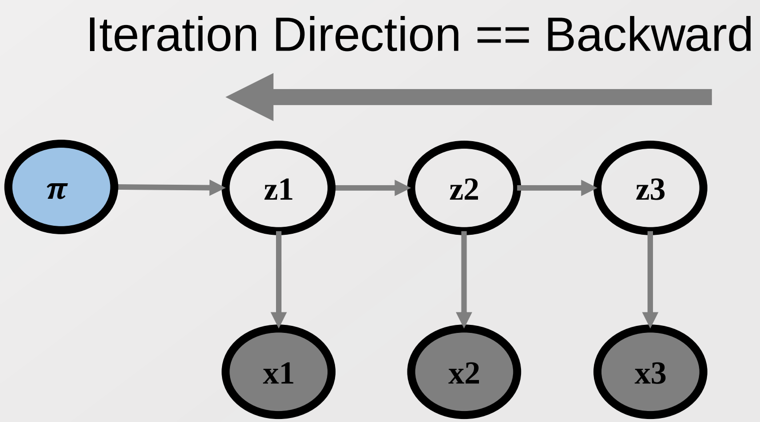
Forward Probability Calculation

- Need to know α_t^k
  - Time X States
  - When we know α_t^k with X, then we know the value of P(X)
    - Answering the evaluation question without Z
- ForwardAlgorithm
  - Initialize
    - $\alpha_1^k = b_{k,x_1}\pi_k$
  - Iterate until time T
    - $\alpha_t^k = b_{k,x_t}\sum_i \alpha_{t-1}^i a_{i,k}$
  - Return $\sum_i \alpha_T^i$
- Proof of correctness
  - $\sum_i \alpha_T^i = \sum_i P\left(x_1, \ldots, x_T, z_T^i = 1\right) = P(x_1, \ldots, x_t)$
- Where to use the memoization table?
  - α_t^k
- Limitation of the forward probability
  - Only takes the input sequence of X before time *t*
  - $P\left(x_1, \ldots, x_t, z_t^k = 1\right) = \alpha_t^k$ and $t \neq T$
  - Need to see a probability distribution of a latent variable at time *t* given the whole *X*
  - Recall the Bayes ball algorithm



Backward Probability Calculation

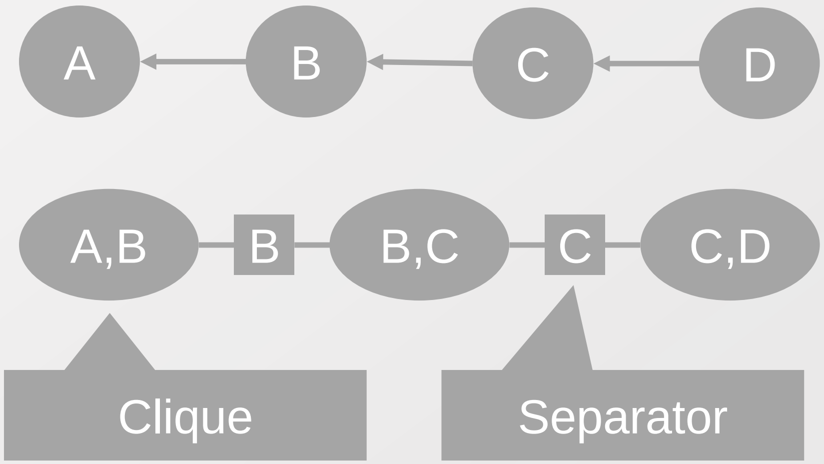
- We need $P(z_t^k = 1|X)$ instead of $P(x_1, \ldots, x_t, z_t^k = 1)$
- Let's derive from the joint probability
  - $P(z_t^k = 1, X) = P(x_1, \ldots, x_t, z_t^k = 1, x_{t+1}, \ldots, x_T)$
  - $= P(x_1, \ldots, x_t, z_t^k = 1) P(x_{t+1}, \ldots, x_T | x_1, \ldots, x_t, z_t^k = 1)$
    - By the virtue of the structure
  - $= P(x_1, \ldots, x_t, z_t^k = 1) P(x_{t+1}, \ldots, x_T | z_t^k = 1)$
    - We already handled $P(x_1, \ldots, x_t, z_t^k = 1)$
      - $P(x_1, \ldots, x_t, z_t^k = 1) = \alpha_t^k$
    - So, we need to compute $P(x_{t+1}, \ldots, x_T | z_t^k = 1)$
      - $P(x_{t+1}, \ldots, x_T | z_t^k = 1) = \beta_t^k$



- $P(x_{t+1}, \ldots, x_T | z_t^k = 1)$
  - $= \sum_{z_{t+1}} P(z_{t+1}, x_{t+1}, \ldots, x_T | z_t^k = 1)$
  - $= \sum_i P(z_{t+1}^i = 1 | z_t^k = 1) P(x_{t+1} | z_{t+1}^i = 1, z_t^k = 1) P(x_{t+2}, \ldots, x_T | x_{t+1}, z_{t+1}^i = 1, z_t^k = 1)$
  - $= \sum_i P(z_{t+1}^i = 1 | z_t^k = 1) P(x_{t+1} | z_{t+1}^i = 1) P(x_{t+2}, \ldots, x_T | z_{t+1}^i = 1)$
  - $= \sum_i a_{k,i} b_{i,x_t} \beta_{t+1}^i$
  - Again, recursive structure. How to calculate this efficiently?
- $P(z_t^k = 1, X) = \alpha_t^k \beta_t^k = (b_{k,x_t} \sum_i \alpha_{t-1}^i a_{i,k}) \times (\sum_i a_{k,i} b_{i,x_t} \beta_{t+1}^i)$

Detour: Potential Functions

- $P(A,B,C,D) = P(A|B)P(B|C)P(C|D)P(D)$
- Let's define a potential function
  - Potential function:
    a function which is not a probability function yet, but once normalized it can be a probability distribution function
  - Potential function on nodes
    - $\psi(a,b), \psi(b,c), \psi(c,d)$
  - Potential function on links
    - $\phi(b), \phi(c)$



- How to setup the function?
  - $P(A,B,C,D) = P(U) = \frac{\prod_N \psi(N)}{\prod_L \phi(L)} = \frac{\psi(a,b)\psi(b,c)\psi(c,d)}{\phi(b)\phi(c)}$
    - $\psi(a,b) = P(A|B), \psi(b,c) = P(B|C), \psi(c,d) = P(C|D)P(D)$
    - $\phi(b) = 1, \phi(c) = 1$
  - $P(A,B,C,D) = P(U) = \frac{\prod_N \psi(N)}{\prod_L \phi(L)} = \frac{\psi^*(a,b)\psi^*(b,c)\psi^*(c,d)}{\phi^*(b)\phi^*(c)}$
    - $\psi^*(a,b) = P(A,B), \psi^*(b,c) = P(B,C), \psi^*(c,d) = P(C,D)$
    - $\phi^*(b) = P(B), \phi^*(c) = P(C)$

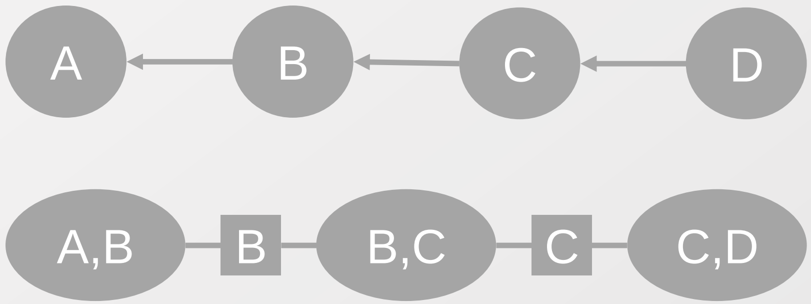
Marginalization is also applicable:

$$\psi(w) = \sum_{v-w} \psi(v)$$

Constructing a potential of a subset (w) of all variables (v)

Detour: Absorption in Clique Graph

- Only applicable to the tree structure of clique graph
- Let's assume
  - $P(B) = \sum_A \psi(A,B)$
  - $P(B) = \sum_C \psi(B,C)$
  - $P(B) = \phi(B)$
  - How to find out the ψs and the ϕs?
    - When the ψs change by the observations: P(A,B) → P(A=1,B)
    - A single ψ change can result in the change of multiple ψs
    - The effect of the observation propagates through the clique graph
    - Belief propagation!



- How to propagate the belief?
  - Absorption (update) rule
  - Assume $\psi^*(A,B)$,$\psi(B,C)$, and $\phi(B)$
  - Define the update rule for separators
    - $\phi^*(B) = \sum_A \psi^*(A,B)$
  - Define the update rule for cliques
    - $\psi^*(B,C)=\psi(B,C)\frac{\phi^*(B)}{\phi(B)}$

Why does this work?

$$\sum_C \psi^*(B,C) = \sum_C \psi(B,C)\frac{\phi^*(B)}{\phi(B)}$$

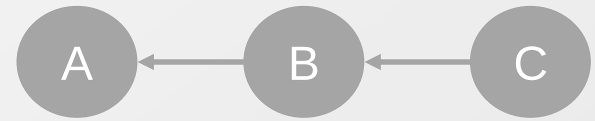
$$= \frac{\phi^*(B)}{\phi(B)}\sum_C \psi(B,C) = \frac{\phi^*(B)}{\phi(B)}\phi(B) = \sum_A \psi^*(A,B)$$

Guarantees the local consistency

→ Global consistency after iterations

Detour: Simple Example of Belief Propagation

- Initialized the potential functions
  - $\psi(a,b) = P(a|b), \psi(b,c) = P(b|c)P(c)$
  - $\phi(b) = 1$
- Example 1. $P(b) = ?$
  - $\phi^*(b) = \sum_a \psi(a,b) = 1$
  - $\psi^*(b,c) = \psi(b,c)\frac{\phi^*(b)}{\phi(b)} = P(b|c)P(c) = P(b,c)$
  - $\phi^{**}(b) = \sum_c \psi(b,c) = \sum_c P(b,c) \ = P(b)$
  - $\psi^*(a,b) = \psi(a,b)\frac{\phi^{**}(b)}{\phi^*(b)} = \frac{P(a|b)P(b)}{1} = P(a,b)$
  - $\phi^{***}(b) = \sum_a \psi^*(a,b) = P(b)$
- Example 2. $P(b|a=1, c=1) = ?$
  - $\phi^*(b) = \sum_a \psi(a,b)\delta(a=1) = P(a=1|b)$
  - $\psi^*(b,c) = \psi(b,c)\frac{\phi^*(b)}{\phi(b)} = P(b|c=1)P(c=1)\frac{P(a=1|b)}{1}$
  - $\phi^{**}(b) = \sum_c \psi(b,c)\,\delta(c=1) = P(b|c=1)P(c=1)P(a=1|b)$
  - $\psi^*(a,b) = \psi(a,b)\frac{\phi^{**}(b)}{\phi^*(b)} = \mathrm{P(a=1|b)}\frac{P(b|c=1)P(c=1)P(a=1|b)}{P(a=1|b)} = P(b|c=1)P(c=1)P(a=1|b)$
  - $\phi^{***}(b) = \sum_a \psi^*(a,b)\,\delta(a=1) = P(b|c=1)P(c=1)P(a=1|b)$

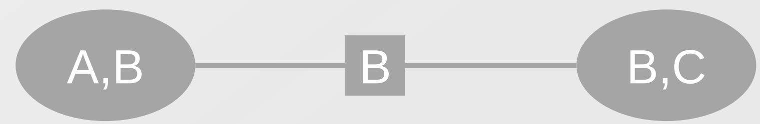


Bayesian Network

1) $\phi^*(b)$ → 2) $\psi^*(b,c)$ →

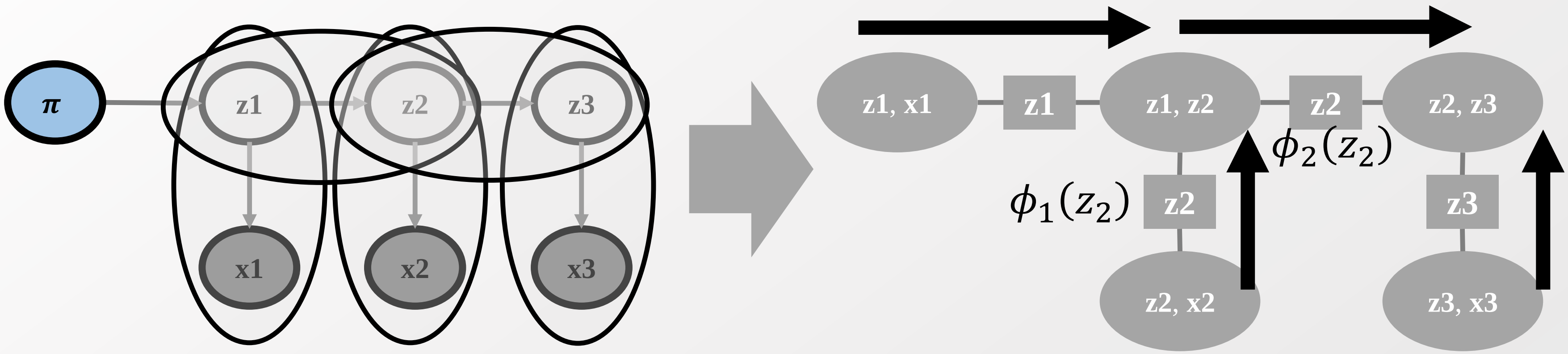
← 4) $\psi^*(a,b)$ ← 3) $\phi^{**}(b)$

5) $\phi^{***}(b)$ →



Clique Graph

Message Passing and Forward-Backward



Define the update rule for separators

$$\boldsymbol{\phi}^*(\boldsymbol{B}) = \sum_{A} \boldsymbol{\psi}^*(\boldsymbol{A}, \boldsymbol{B})$$

Define the update rule for cliques

$$\boldsymbol{\psi}^*(\boldsymbol{B}, \boldsymbol{C}) = \boldsymbol{\psi}(\boldsymbol{B}, \boldsymbol{C}) \frac{\boldsymbol{\phi}^*(\boldsymbol{B})}{\boldsymbol{\phi}(\boldsymbol{B})}$$

Initial State probabilities

$$\boldsymbol{P}(\boldsymbol{z}_1) \sim \boldsymbol{Mult}(\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_k)$$

Transition probabilities

$$\boldsymbol{P}(\boldsymbol{z}_t | \boldsymbol{z}_{t-1}^i = \boldsymbol{1}) \sim \boldsymbol{Mult}(\boldsymbol{a}_{i,1}, \dots, \boldsymbol{a}_{i,k})$$

Or, $\boldsymbol{P}\left(\boldsymbol{z}_t^j = \boldsymbol{1} \middle| \boldsymbol{z}_{t-1}^i = \boldsymbol{1}\right) = \boldsymbol{a}_{i,j}$

Emission probabilities

$$\boldsymbol{P}(\boldsymbol{x}_t | \boldsymbol{z}_t^i = \boldsymbol{1}) \sim \boldsymbol{Mult}(\boldsymbol{b}_{i,1}, \dots, \boldsymbol{b}_{i,m})$$

Or, $\boldsymbol{P}\left(\boldsymbol{x}_t^j = \boldsymbol{1} \middle| \boldsymbol{z}_t^i = \boldsymbol{1}\right) = \boldsymbol{b}_{i,j}$

- $P(z_1, z_2, z_3, x_1, x_2, x_3) = \frac{\prod_N \psi(N)}{\prod_L \phi(L)}$
  - $= \frac{\psi(z_1,x_1)\psi(z_1,z_2)\psi(z_2,z_3)\psi(z_2,x_2)\psi(z_3,x_3)}{\phi(z_1)\phi_1(z_2)\phi_2(z_2)\phi(z_3)}$
- $P(z_1, z_2, z_3, x_1, x_2, x_3)$
  - $= P(z_1)P(x_1|z_1)P(z_2|z_1)P(x_2|z_2)P(z_3|z_2)P(x_3|z_3)$
- Initialized the potential functions
  - $\psi(z_1, x_1) = P(z_1)P(x_1|z_1), \psi(z_1, z_2) = P(z_2|z_1), \psi(z_2, z_3) = P(z_3|z_2), \psi(z_2, x_2) = P(x_2|z_2), \psi(z_3, x_3) = P(x_3|z_3)$
  - $\phi(z_1) = \phi_1(z_2) = \phi_2(z_2) = \phi(z_3) = 1$
- Start absorbing and updating
  - $\phi_2^*(z_2) = \sum_{z_1} \psi^*(z_1, z_2) = \sum_{z_1} \psi(z_1, z_2)\, \phi^*(z_1)\phi_1^*(z_2) = \sum_{z_1} \psi(z_1, z_2)\, \phi^*(z_1)\phi_1^*(z_2)$
    - Because x_2 is already observed, so the summation on x_2 does not happen, use just fixed x_2
  - $= \sum_{z_1} P(z_2|z_1)\, \phi^*(z_1) P(x_2|z_2) = P(x_2|z_2) \sum_{z_1} P(z_2|z_1)\, \phi^*(z_1) = b_{idx(z_2),x_2} \sum_{i \in z_1} \alpha_{2-1}^i a_{i,z_2}$
  - Same as the forward probability calculation
  - This is the upward process, then the downward process is same as the backward probability calculation

Viterbi Decoding

Initial State probabilities

$P(z_1) \sim Mult(\pi_1, \dots, \pi_k)$

Transition probabilities

$P(z_t | z_{t-1}^i = 1) \sim Mult(a_{i,1}, \dots, a_{i,k})$

Or, $P\left(z_t^j = 1 \middle| z_{t-1}^i = 1\right) = a_{i,j}$

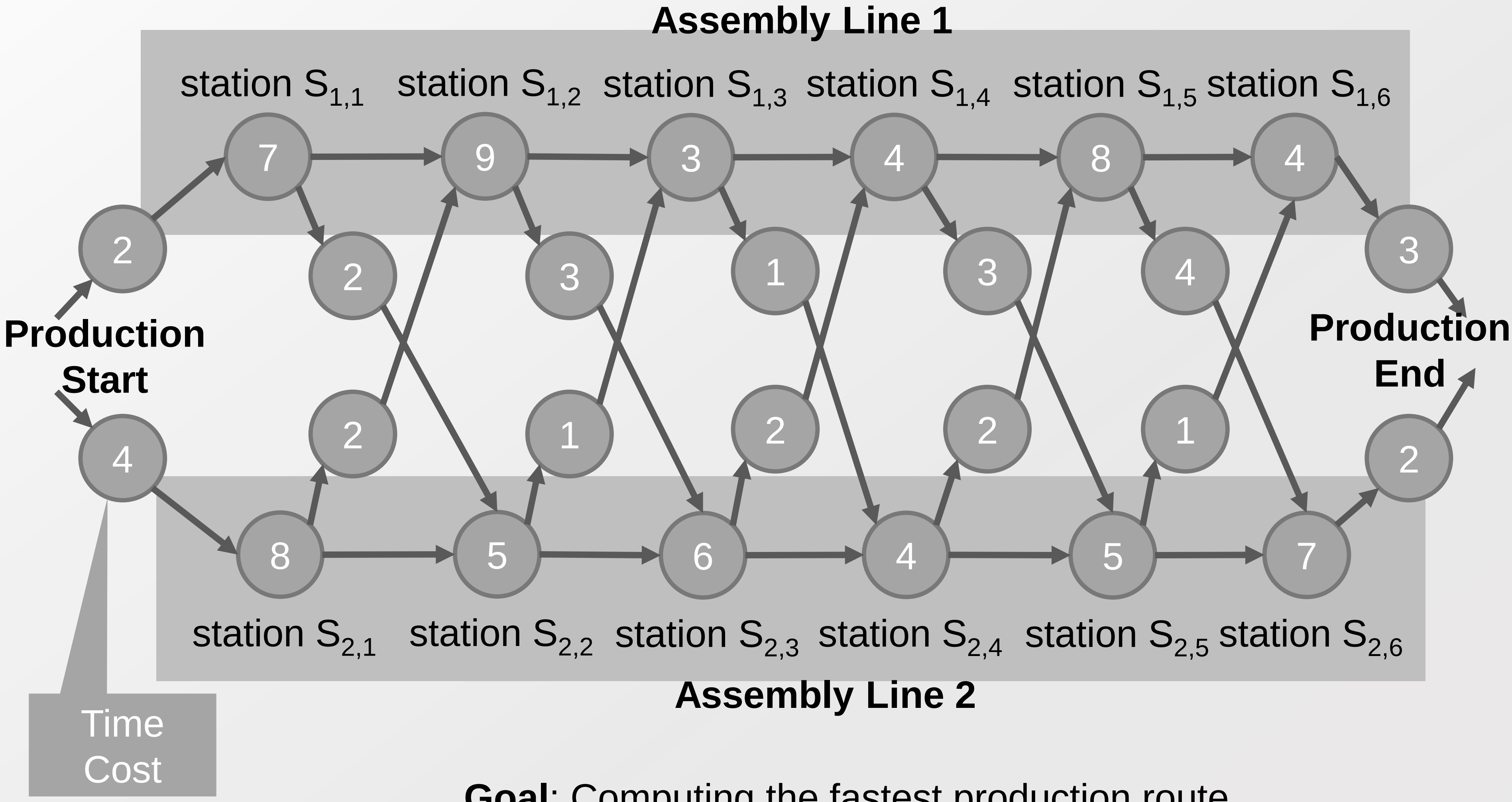
Emission probabilities

$P\left(x_t | z_t^i = 1\right) \sim Mult\left(b_{i,1}, \dots, b_{i,m}\right)$

Or, $P\left(x_t^j = 1 \middle| z_t^i = 1\right) = b_{i,j}$

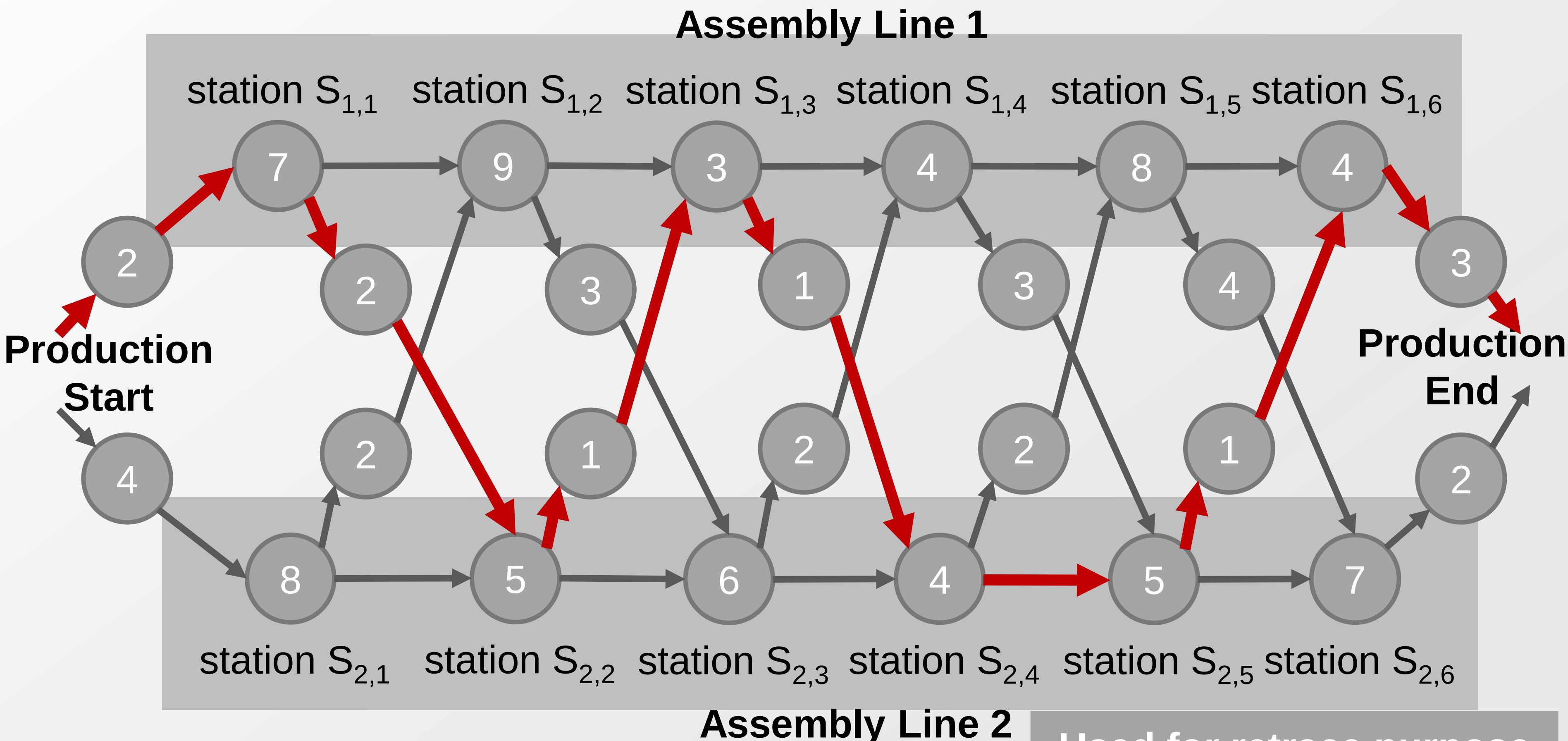
- $P\left(z_t^k = 1, X\right) = \alpha_t^k \beta_t^k = (b_{k,x_t} \sum_i \alpha_{t-1}^i a_{i,k}) \times (\sum_i a_{k,i} b_{i,x_t} \beta_{t+1}^i)$
  - This dictates the most probable assignment to a single latent variable, z_t, given the whole observed sequence, X
  - $k_t^* = argmax_k P(z_t^k = 1 | X) = argmax_k P\left(z_t^k = 1, X\right) = argmax_k \alpha_t^k \beta_t^k$
  - What if we want to have the most probable assignment of Z given X?
    - Exactly the decoding question
    - Different from the most probable assignment of a single latent variable
- Viterbi decoding
  - $k^* = argmax_k P\left(z^k = 1 \middle| X\right) = argmax_k P\left(z^k = 1, X\right)$
  - Need to model the sequence of Z.
    - Let's use the forward approach (Bottom–up)
  - $V_t^k = max_{z_1 \dots z_{t-1}} P(x_1, \dots, x_{t-1}, z_1, \dots, z_{t-1}, x_t, z_t^k = 1)$
    - Most probable sequence of latent states until *t-1* and fixing the state *k* at time *t*
  - $= max_{z_1 \dots z_{t-1}} P\left(x_t, z_t^k = 1 \middle| x_1, \dots, x_{t-1}, z_1, \dots, z_{t-1}\right) P(x_1, \dots, x_{t-1}, z_1, \dots, z_{t-1})$
  - $= max_{z_1 \dots z_{t-1}} P\left(x_t, z_t^k = 1 \middle| z_{t-1}\right) P(x_1, \dots, x_{t-2}, z_1, \dots, z_{t-2}, x_{t-1}, z_{t-1})$
  - $= max_{z_{t-1}} P\left(x_t, z_t^k = 1 \middle| z_{t-1}\right) max_{z_1 \dots z_{t-2}} P(x_1, \dots, x_{t-2}, z_1, \dots, z_{t-2}, x_{t-1}, z_{t-1})$
  - $= max_{i \in z_{t-1}} P\left(x_t, z_t^k = 1 \middle| z_{t-1}^i = 1\right) V_{t-1}^i = max_{i \in z_{t-1}} P(x_t | z_t^k = 1) P\left(z_t^k = 1 \middle| z_{t-1}^i = 1\right) V_{t-1}^i$
  - $= P(x_t | z_t^k = 1) max_{i \in z_{t-1}} P\left(z_t^k = 1 \middle| z_{t-1}^i = 1\right) V_{t-1}^i = b_{k,idx(x_t)} max_{i \in z_{t-1}} a_{i,k} V_{t-1}^i$
  - Keep going until time T

Detour: Assembly Line Scheduling



Goal: Computing the fastest production route

Detour: Tracing Assembly Line Scheduling in DP

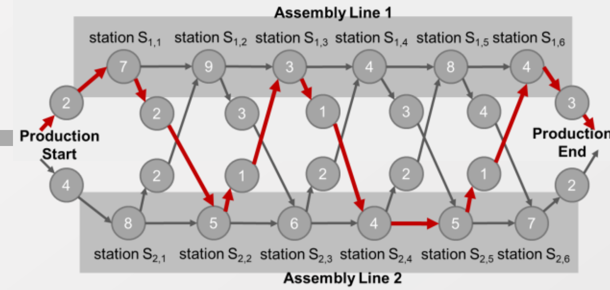


Used for retrace purpose

Time	1	2	3	4	5	6
L1	9	18	20	24	32	35
L2	12	16	22	25	30	37

Trace	1	2	3	4	5	6
L1	S	1	2	1	1	2
L2	S	1	2	1	2	2

Viterbi Decoding Algorithm



Time	1	2	3	4	5	6
L1	9	18	20	24	32	35
L2	12	16	22	25	30	37

Trace	1	2	3	4	5	6
L1	S	1	2	1	1	2
L2	S	1	2	1	2	2

- Need to know V_t^k
  - Time X States
  - Store V_t^k Two variables to store the trace and the probability up to time *t*: Two memoization tables
  - Answering the decoding question with π, a, b, X
- ViterbiDecodingAlgorithm
  - Initialize
    - $V_1^k = b_{k,x_1}\pi_k$
  - Iterate until time T
    - $V_t^k = b_{k,idx(x_t)} max_{i \in z_{t-1}} a_{i,k} V_{t-1}^i$
    - $trace_t^k = argmax_{i \in z_{t-1}} a_{i,k} V_{t-1}^i$
  - Return $\mathrm{P}(\mathrm{X}, Z^*) = max_k V_T^k$, $z_T^* = argmax_k V_T^k$, $z_{t-1}^* = trace_t^{z_t^*}$
- Technical difficulties in the implementation
  - Very frequent underflow problems.
  - Turn this into the log domain → from multiplication to summation

Learning Parameters with Only X

- Importance of π, a, b
  - HMM parameters
  - Forward algorithm (evaluation) and Viterbi algorithm (decoding) depends on knowing π, a, b
- However, knowing π, a, b assumes that we have observed X and Z
  - But, often Z is hard to observe. Need tagging, annotation, etc
  - Often the latent space is what we want to know, so we can't assume that we know Z
- If we don't know Z, we can assign the most probable Z to X
  - However, this is decoding problem, and this requires knowing π, a, b
- Most likely scenario in the real world
  - You have only X
  - You don't have Z, π, a, b, and you need to find out Z, π, a, b
- Strategy
  - Finding the optimized $\bar{\pi}$, $\bar{a}$, $\bar{b}$ with X
  - Finding the most probable Z with X, $\bar{\pi}$, $\bar{a}$, $\bar{b}$
  - How to find the unknown parameter of the latent distribution without supervision?
- **<u>EM algorithm!</u>**
  - **Iteratively optimizing $\boldsymbol{\bar{\pi}}$, $\boldsymbol{\bar{a}}$, $\boldsymbol{\bar{b}}$ and Z**

Detour: EM Algorithm

$$l(\theta) = \ln P(X|\theta) = \ln\left\{\sum_Z q(Z)\frac{P(X,Z|\theta)}{q(Z)}\right\} \geq \sum_Z q(Z)\ln\frac{P(X,Z|\theta)}{q(Z)} = Q(\theta,q)$$
$$Q(\theta,q) = E_{q(Z)}\ln P(X,Z|\theta) + H(q)$$
$$L(\theta,q) = \ln P(X|\theta) - \sum_Z \{q(Z)\ln\frac{q(Z)}{P(Z|X,\theta)}\}$$

- EM algorithm
  - Finds the maximum likelihood solutions for models with latent variables
  - $P(X|\theta) = \sum_Z P(X,Z|\theta) \rightarrow \ln P(X|\theta) = \ln\{\sum_Z P(X,Z|\theta)\}$
- EM algorithm
  - Initialize θ^0 to an arbitrary point
  - Loop until the likelihood converges
    - Expectation step
      - $q^{t+1}(z) = argmax_q Q(\theta^t,q) = argmax_q L(\theta^t,q) = argmin_q KL(q||P(Z|X,\theta^t))$
      - → $q^t(z) = P(Z|X,\theta)$→Assign Z by $P(Z|X,\theta)$
    - Maximization step
      - $\theta^{t+1} = argmax_\theta Q(\theta,q^{t+1}) = argmax_\theta L(\theta,q^{t+1})$
      - → fixed Z means that there is no unobserved variables
      - → Same optimization of ordinary MLE

EM for HMM

$l(\theta) = \ln P(X|\theta) = \ln\left\{\sum_Z q(Z)\frac{P(X,Z|\theta)}{q(Z)}\right\}$

$\geq \sum_Z q(Z)\ln\frac{P(X,Z|\theta)}{q(Z)} = Q(\theta,q)$

$Q(\theta,q) = E_{q(Z)}\ln P(X,Z|\theta) + H(q)$

$L(\theta,q) = \ln P(X|\theta) - \sum_Z\{q(Z)\ln\frac{q(Z)}{P(Z|X,\theta)}\}$

Initial State probabilities

$P(z_1) \sim Mult(\pi_1,\ldots,\pi_k)$

Transition probabilities

$P(z_t|z_{t-1}^i = 1) \sim Mult(a_{i,1},\ldots,a_{i,k})$

Or, $P\left(z_t^j = 1 \middle| z_{t-1}^i = 1\right) = a_{i,j}$

Emission probabilities

$P\left(x_t \middle| z_t^i = 1\right) \sim Mult\left(b_{i,1},\ldots,b_{i,m}\right)$

Or, $P\left(x_t^j = 1 \middle| z_t^i = 1\right) = b_{i,j}$

- EM algorithm for HMM
  - Initialize π^0,a^0,b^0 to an arbitrary point
  - Loop until the likelihood converges
    - Expectation step
      - $q^{t+1}(z) = P(Z|X,\pi^t, a^t, b^t)$ → Assign Z by $P(Z|X,\pi^t, a^t, b^t)$
    - Maximization step
      - $\pi^{t+1}, a^{t+1}, b^{t+1} = argmax_{\pi,\, a,\, b} Q(\pi, a, b, q^{t+1}) = argmax_{\pi,\, a,\, b} E_{q^{t+1}(z)} lnP(X,Z|\pi, a, b) + H(q)$
- Assign Z and optimize π, a, b alternatively
  - Coordinated optimization
  - How to optimize? Derivation of EM update formula from HMM?
  - $P(X,Z|\pi, a, b) = \pi_{z_1}\prod_{t=2}^T a_{z_{t-1},z_t}\prod_{t=1}^T b_{z_t,x_t}$
  - $\ln P(X,Z|\pi, a, b) = \pi_{z_1} + \sum_{t=2}^T \ln a_{z_{t-1},z_t} + \sum_{t=1}^T \ln b_{z_t,x_t}$
  - $E_{q^{t+1}(z)} lnP(X,Z|\pi, a, b) = \sum_Z q^{t+1}(z)\ln P(X,Z|\pi, a, b)$

    $= \sum_Z P(Z|X,\pi^t, a^t, b^t)\ln P(X,Z|\pi, a, b)$

Derivation of EM Update Formula

Similarly, we can compute the update formula for *a* and *b* with the partial derivatives

$$a^{t+1}{}_{i,j} = \frac{\sum_{t=2}^{T} P\left(z_{t-1}^{i} = 1, z_{t}^{j} = 1 \middle| X, \pi^t, a^t, b^t\right)}{\sum_{t=2}^{T} P\left(z_{t-1}^{i} = 1 \middle| X, \pi^t, a^t, b^t\right)}$$

$$b^{t+1}{}_{i,j} = \frac{\sum_{t=1}^{T} P\left(z_{t}^{i} = 1 \middle| X, \pi^t, a^t, b^t\right)\delta(idx(x_t) = j)}{\sum_{t=1}^{T} P\left(z_{t}^{i} = 1 \middle| X, \pi^t, a^t, b^t\right)}$$

- $Q(\pi, a, b, q^{t+1}) = E_{q^{t+1}(z)} lnP(X, Z|\pi, a, b)$
- $= \sum_Z q^{t+1}(z) \ln P(X, Z|\pi, a, b)$
- $= \sum_Z P(Z|X, \pi^t, a^t, b^t) \ln P(X, Z|\pi, a, b)$
  - $\ln P(X, Z|\pi, a, b)$
  - $= \ln \pi_{z_1} + \sum_{t=2}^{T} \ln a_{z_{t-1},z_t} + \sum_{t=1}^{T} \ln b_{z_t,x_t}$
- $= \sum_Z P(Z|X, \pi^t, a^t, b^t)\{\ln \pi_{z_1} + \sum_{t=2}^{T} \ln a_{z_{t-1},z_t} + \sum_{t=1}^{T} \ln b_{z_t,x_t}\}$
- Need to optimize $Q(\pi, a, b, q^{t+1})$ by using π, a, b
  - Remember that π, a, b is actually probabilities. $\sum_i \pi_i = 1$
  - Since there are constraints on π, a, b and Q is smooth → Lagrange method!
- $L(\pi, a, b, q^{t+1})$
- $= Q(\pi, a, b, q^{t+1}) - \lambda_\pi\left(\sum_{i=1}^{K} \pi_i - 1\right) - \sum_{i=1}^{K} \lambda_{a_i} \left(\sum_{j=1}^{K} a_{i,j} - 1\right) - \sum_{i=1}^{K} \lambda_{b_i} \left(\sum_{j=1}^{M} b_{i,j} - 1\right)$
- Now, a typical optimization with partial derivative
- $\frac{dL(\pi, a, b, q^{t+1})}{d\pi_i} = \frac{d}{d\pi_i} \sum_Z P(Z|X, \pi^t, a^t, b^t) \ln \pi_{z_1} - \lambda_\pi\left(\sum_{i=1}^{K} \pi_i - 1\right) = 0$
  - Only the terms with $z_1 = i$ survives
- $\frac{d}{d\pi_i}\{\sum_Z P(Z|X, \pi^t, a^t, b^t) \ln \pi_{z_1} - \lambda_\pi\left(\sum_{i=1}^{K} \pi_i - 1\right)\} = \frac{P\left(z_1^i = 1 \middle| X, \pi^t, a^t, b^t\right)}{\pi_i} - \lambda_\pi = 0 \rightarrow \pi_i = \frac{P\left(z_1^i = 1 \middle| X, \pi^t, a^t, b^t\right)}{\lambda_\pi}$
- $\frac{d}{d\lambda_\pi} L(\pi, a, b, q^{t+1}) = -\left(\sum_{i=1}^{K} \pi_i - 1\right) = 0 \rightarrow \sum_{i=1}^{K} \pi_i = 1$
- Together, $\pi^{t+1}{}_i = \frac{P\left(z_1^i = 1 \middle| X, \pi^t, a^t, b^t\right)}{\sum_{j=1}^{K} P\left(z_1^j = 1 \middle| X, \pi^t, a^t, b^t\right)}$
  - This is an update function for π_i at the M Step

Baum Welch Algorithm

- Answer to the learning question of HMM
- Again, EM for HMM with more details
- EM algorithm for HMM, a.k.a. Baum-Welch, Forward-Backward…
  - Initialize π^0,a^0,b^0 to an arbitrary point
  - Loop until the likelihood converges
    - Expectation step
      - $q^{t+1}(z) = P(Z|X,\pi^t, a^t, b^t)$→Assign Z by $P(Z|X,\pi^t, a^t, b^t)$
    - Maximization step
      - $\pi^{t+1}{}_i = \frac{P\left(z_1^i = 1 \middle| X,\pi^t, a^t, b^t\right)}{\sum_{j=1}^{K} P\left(z_1^j = 1 \middle| X,\pi^t, a^t, b^t\right)}$
      - $a^{t+1}{}_{i,j} = \frac{\sum_{t=2}^{T} P\left(z_{t-1}^i = 1, z_t^j = 1 \middle| X,\pi^t, a^t, b^t\right)}{\sum_{t=2}^{T} P\left(z_{t-1}^i = 1 \middle| X,\pi^t, a^t, b^t\right)}$
      - $b^{t+1}{}_{i,j} = \frac{\sum_{t=1}^{T} P\left(z_t^i = 1 \middle| X,\pi^t, a^t, b^t\right)\delta(idx(x_t)=j)}{\sum_{t=1}^{T} P\left(z_t^i = 1 \middle| X,\pi^t, a^t, b^t\right)}$

$$P\left(z_t^k = 1, X\right) = \alpha_t^k \beta_t^k$$

$$\alpha_t^k = b_{k,x_t} \sum_i \alpha_{t-1}^i a_{i,k}$$

$$\beta_t^k = \sum_i a_{k,i} b_{i,x_t} \beta_{t+1}^i$$

$$P(X) = \sum_i \alpha_T^i$$

Acknowledgement

- This slideset is greatly influenced
  - By Prof. Eric P. Xing at CMU

Further Readings

- Bishop Chapter 13