Practical Machine Learning

Lecture 4
Hidden Markov Model (HMM)

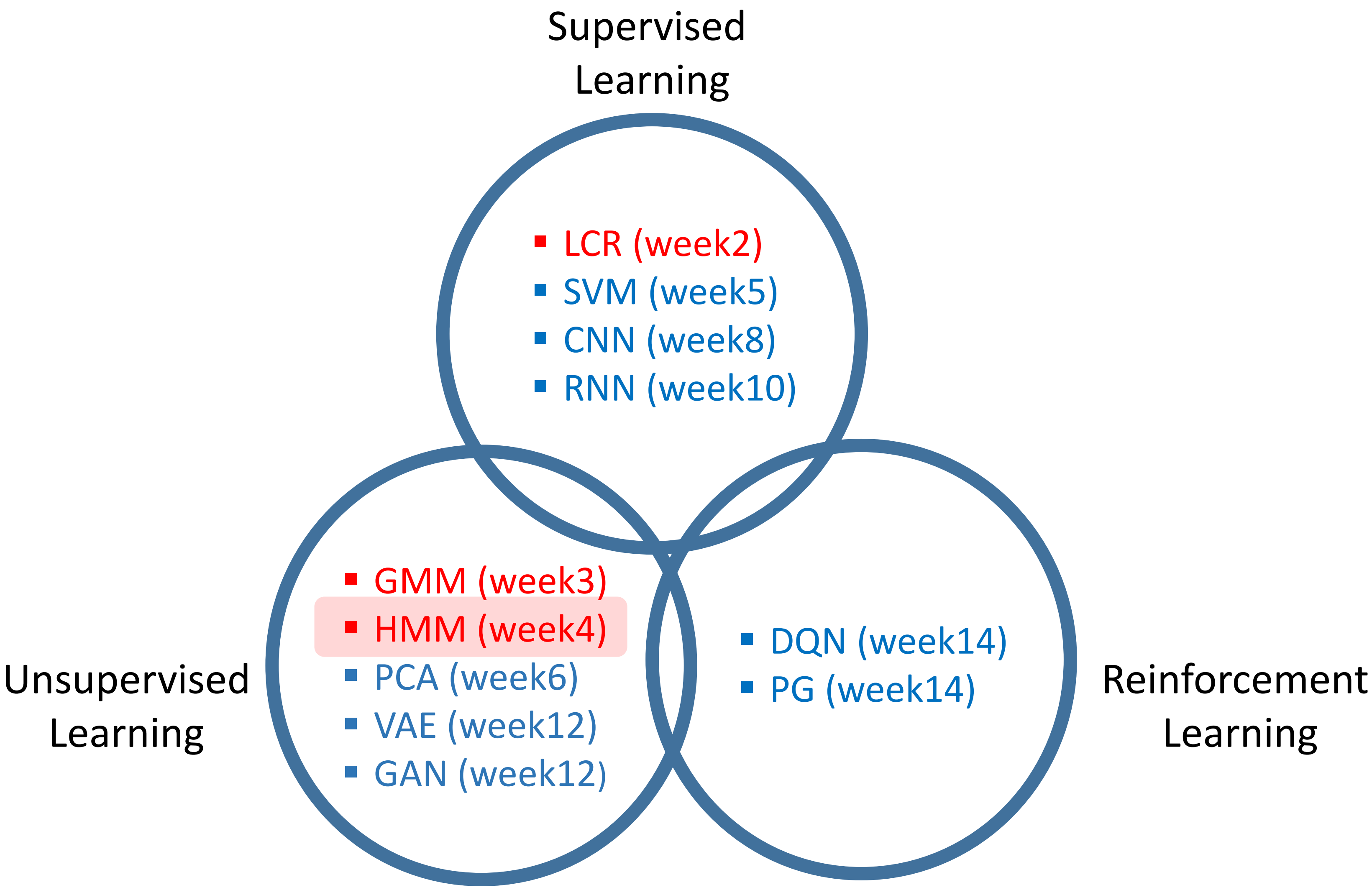
Dr. Suyong Eum

OSAKA UNIVERSITY

Administrative

- Now assignment 3 is on the web: self-regulated open book examination.
  - Short questions every week: let’s say 1 to 3 questions,
  - 40% of the total mark,
  - Individual assignment,
  - Due on Aug 3rd.

Where we are

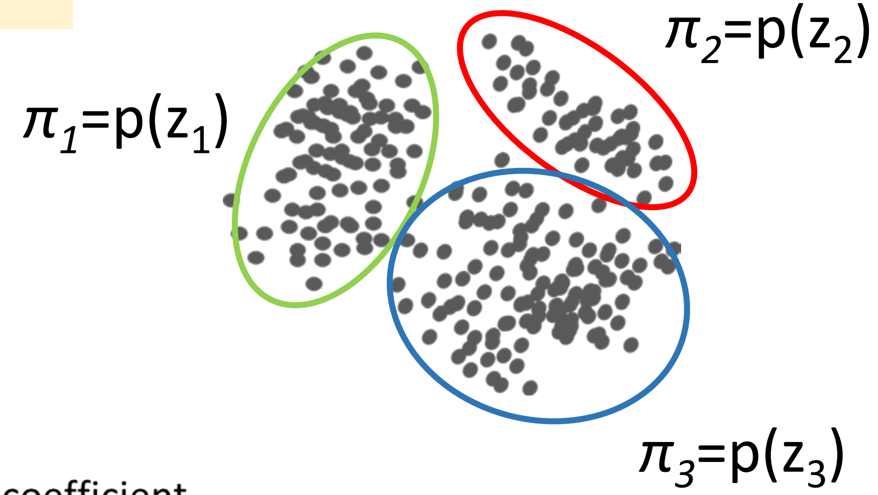

Supervised Learning
LCR (week2)
SVM (week5)
CNN (week8)
RNN (week10)
Unsupervised Learning
GMM (week3)
HMM (week4)
PCA (week6)
VAE (week12)
GAN (week12)
Reinforcement Learning
DQN (week14)
PG (week14)

You are going to learn

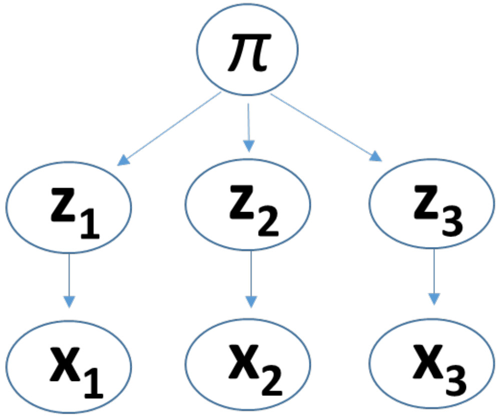
- Introduction to HMM
- Three problems in HMM
  - Evaluation problem
  - Decoding problem: Viterbi algorithm
  - Learning problem: Baum-welch algorithm

Gaussian Mixture Model (GMM) vs Hidden Markov Model (HMM)

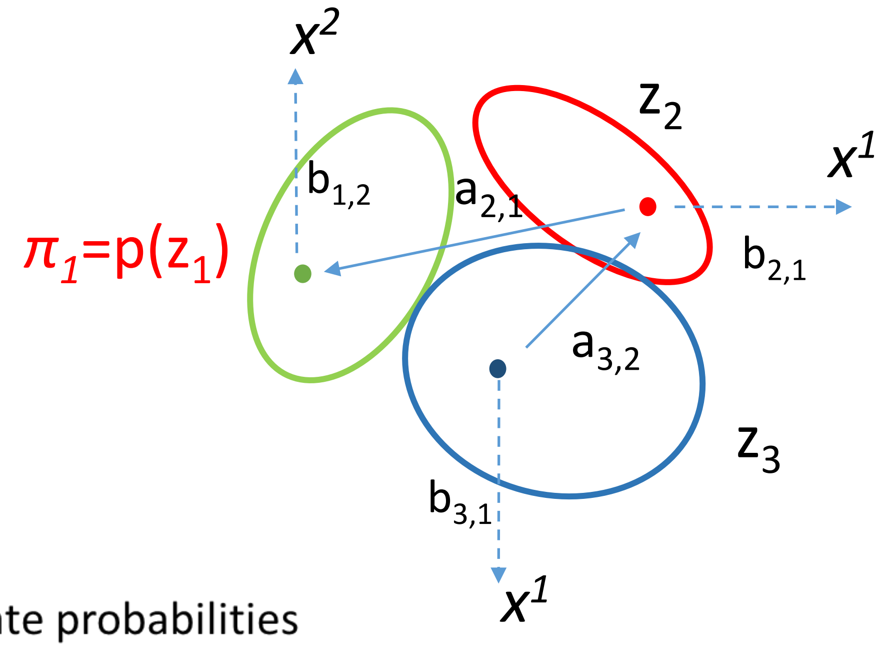
GMM



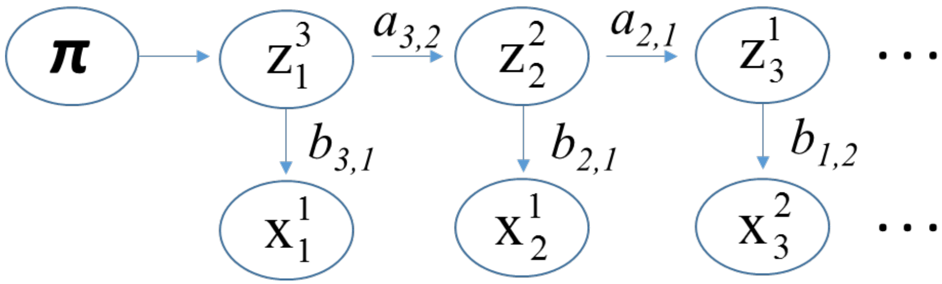
- π : Mixing coefficient
- μ_k : Mean of K^{th} Multivariate Gaussian
- Σ_k : Covariance of K^{th} Multivariate Gaussian
- Z : Latent variables



HMM

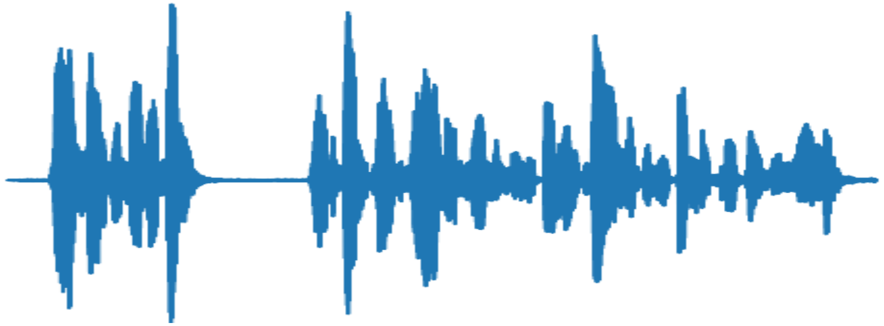


- π : Initial state probabilities
- a : Transition probabilities
- b : Emission probabilities
- Z : Latent variables



Applications of Hidden Markov Model

Speech recognition



I like to eat an apple

I like to it an apple

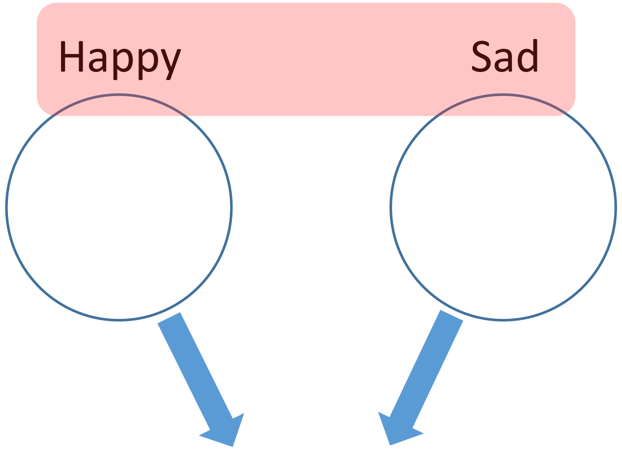
Part of speech tagging

Times flies like an arrow

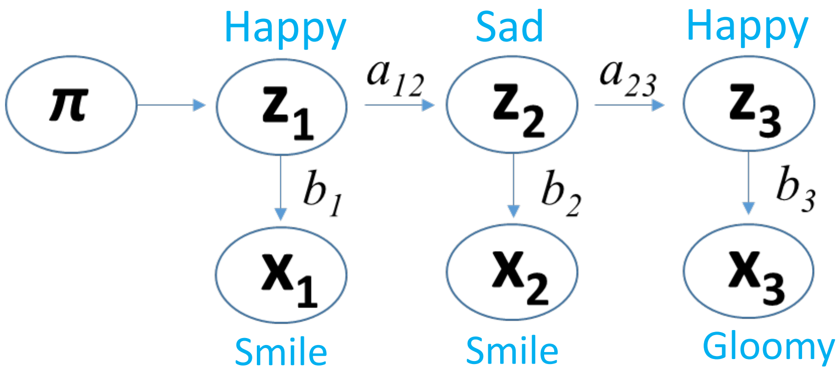
Flies fly

Time series analysis

An example: is your boss Happy or Sad?



- Smile
- Gloomy



- Initial state probability: $\pi = p(z_1)$

Happy	Sad
0.7	0.3

- Transition probability: $p(z_t \mid z_{t-1})$

	Happy (z^1)	Sad (z^2)
Happy (z^1)		
Sad (z^2)		

- Emission probability: $p(x_t \mid z_t)$

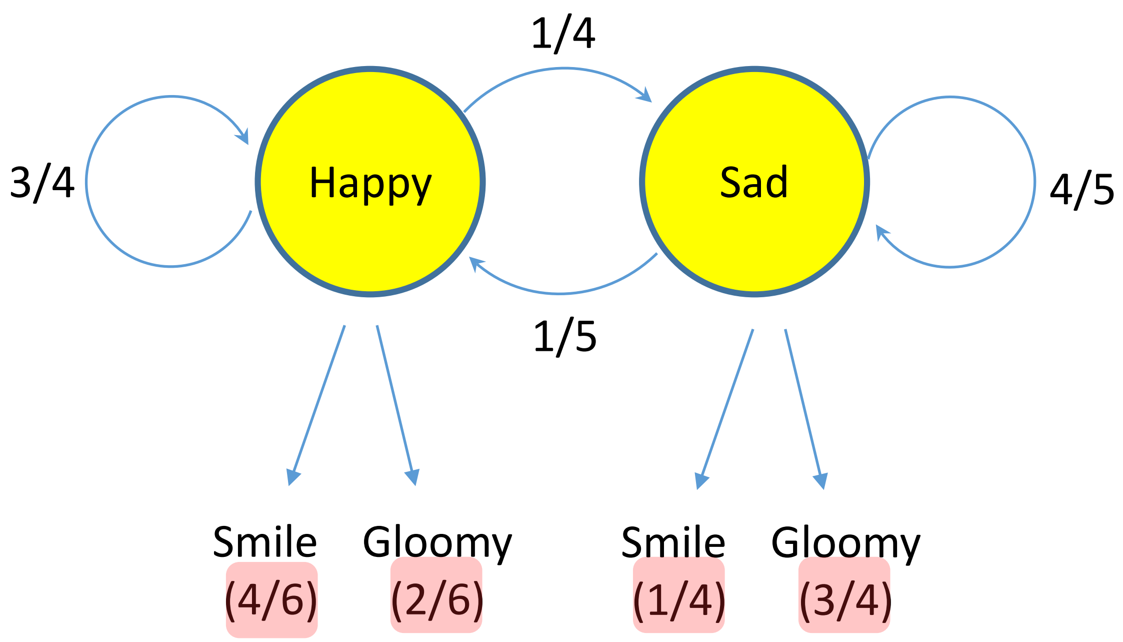
	Happy (z^1)	Sad (z^2)
Smile (x^1)		
Gloomy (x^2)		

Three major problems in HMM

	Input (Given)	Output (Find)	Description
Evaluation problem	π, a, b, X	p(X| π, a, b)	Given a set of observation sequences X= x_1, x_2, …, x_t and the HMM parameters (π, a, b), obtaining the probability p(X| π, a, b)
	e.g.) after observing "Smile-Smile-Gloomy", what is the probability that your boss is happy now?		
Decoding problem	π, a, b, X	p(Z|X, π, a, b)	Given a set of observation sequences X= x_1, x_2, …, x_t and the HMM parameters (π, a, b), obtaining the optimal state sequences
	e.g.) after observing "Smile-Smile-Gloomy", what is his emotional state (happy-happy-sad)?		
Learning problem	X	p(X| π, a, b)	Given a set of observation sequences X= x_1, x_2, …, x_t, adjusting the HMM parameters (π, a, b) to maximize the probability p(X| π, a, b)
	e.g.) Which parameters of HMM generate the observed data (Smile-Smile-Gloomy)?		

Evaluation problem

Starting from data observed



- *π* : Initial state probabilities

Happy	Sad
6/9	3/9

- *a* : Transition probability

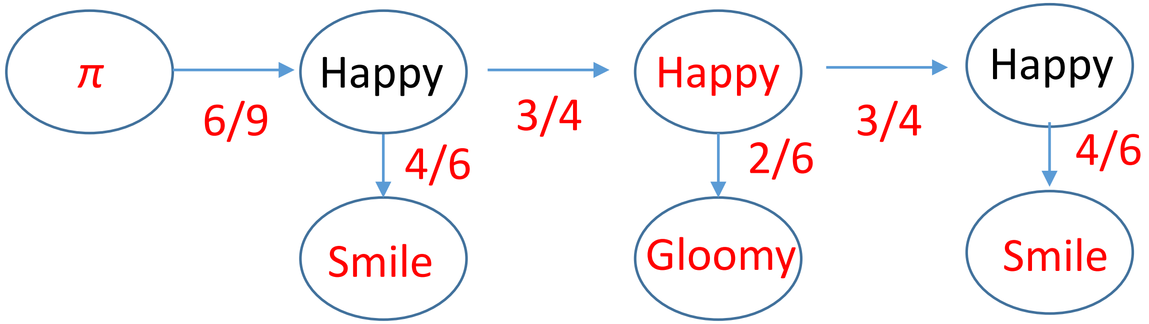
	Happy	Sad
Happy	3/4	1/4
Sad	1/5	4/5

- *b* : Emission probability

Happy		Sad	
Smile	Gloomy	Smile	Gloomy
4/6	2/6	1/4	3/4

Evaluation problem

- What is the probability to observe the data sequence below?
  - p(X| π, a, b)



Cases	Probability (*π x b x a x b x a x b*)
p(Smile-Gloomy-Smile| π, a, b)	6/9 x 4/6 x 1/4 x 3/4 x 1/5 x 4/6 = 0.0111

Cases	Probability (*π x b x a x b x a x b*)
p(Smile-Gloomy-Smile| π, a, b)	6/9 x 4/6 x 3/4 x 2/6 x 3/4 x 4/6 = 0.0556

- *π* : Initial state probabilities

Happy	Sad
6/9	3/9

- *a* : Transition probability

	Happy	Sad
Happy	3/4	1/4
Sad	1/5	4/5

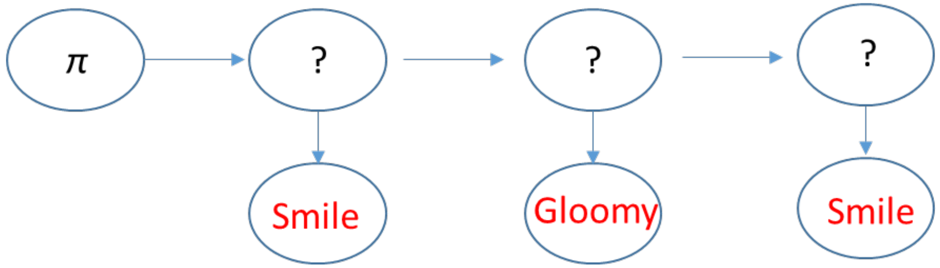
- *b* : Emission probability

Happy		Sad	
Smile	Gloomy	Smile	Gloomy
4/6	2/6	1/4	3/4

Decoding problem with Viterbi algorithm

Decoding problem

- Which sequence of hidden state is likely to generate observed X?
  - p(Z|X, π, a, b)



Cases	Probability (π x b x a x b x a x b)	
p(H-H-H| X, π, a, b)	6/9 x 4/6 x 3/4 x 2/6 x 3/4 x 4/6	0.0556
p(H-H-S| X, π, a, b)	6/9 x 4/6 x 3/4 x 2/6 x 1/4 x 1/4	0.0046
p(H-S-H| X, π, a, b)	6/9 x 4/6 x 1/4 x 3/4 x 1/5 x 4/6	0.0111
p(H-S-S| X, π, a, b)	6/9 x 4/6 x 1/4 x 3/4 x 4/5 x 1/4	0.0167
p(S-S-H| X, π, a, b)	3/9 x 1/4 x 4/5 x 3/4 x 1/5 x 4/6	0.0067
p(S-H-S| X, π, a, b)	3/9 x 1/4 x 1/5 x 2/6 x 1/4 x 1/4	0.0003
p(S-H-H| X, π, a, b)	3/9 x 1/4 x 1/5 x2/6 x 3/4 x 4/6	0.0028
p(S-S-S| X, π, a, b)	3/9 x 1/4 x 4/5 x 3/4 x4/5 x 1/4	0.01

- π : Initial state probabilities

Happy	Sad
6/9	3/9

- *a* : Transition probability

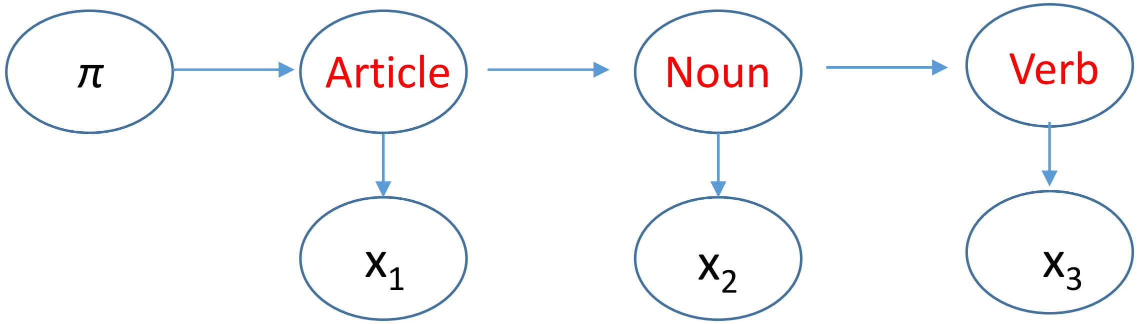
	Happy	Sad
Happy	3/4	1/4
Sad	1/5	4/5

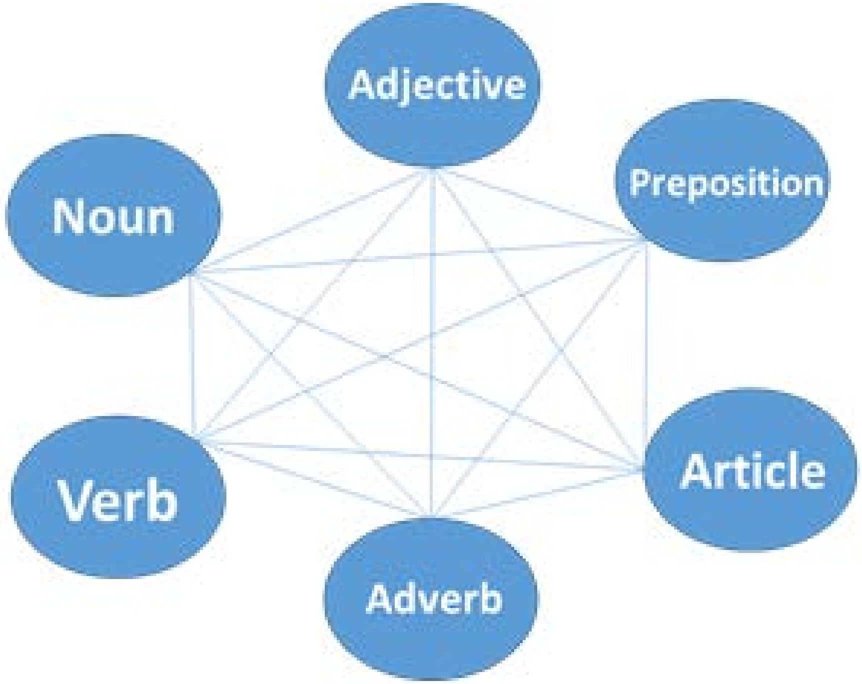
- *b* : Emission probability

Happy		Sad	
Smile	Gloomy	Smile	Gloomy
4/6	2/6	1/4	3/4

of evaluation exponentially increases

- The evaluation cost increases exponentially as the number of hidden variables increases.
- (# of hidden states)$^{(\text{\# of observations})}$ numbers of evalutions is required
  - # of classes / # of hidden states: 6
  - # of observations: 3
  - 6^3: number of evaluation is required.

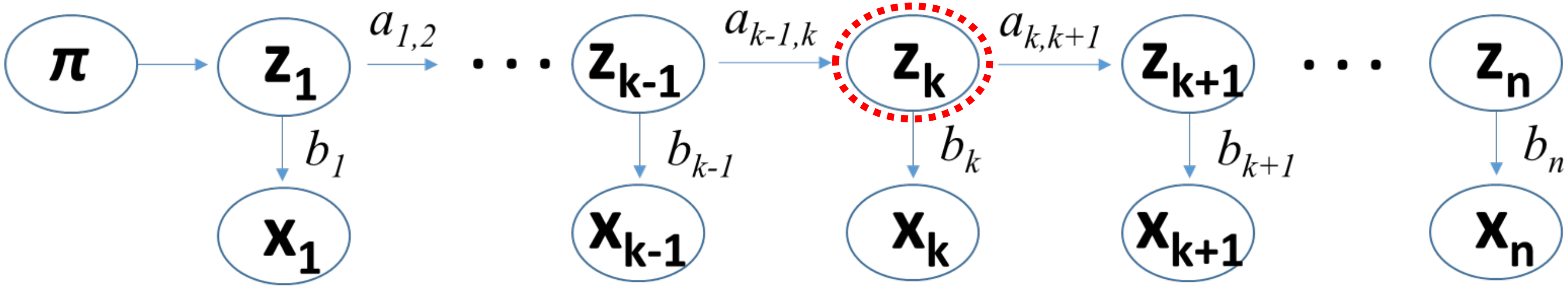




Decoding problem: $p(Z|X, \pi, a, b)$

- Three algorithms in the decoding problem:

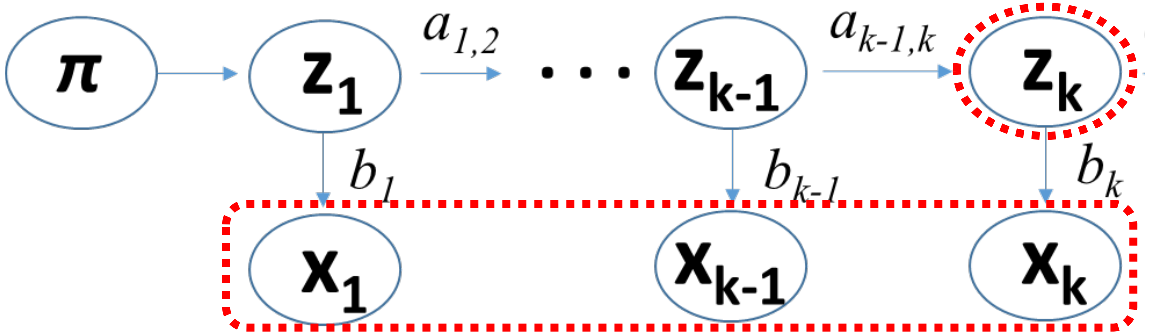
	Forward algorithm	Forward-Backward algorithm	Viterbi algorithm
Notation	$p(z_k \mid x_{1:k})$	$p(z_k \mid x_{1:n})$	$p(z_{1:n} \mid x_{1:n})$



Decoding problem: $p(Z|X, \pi, a, b)$

- Three algorithms in the decoding problem:

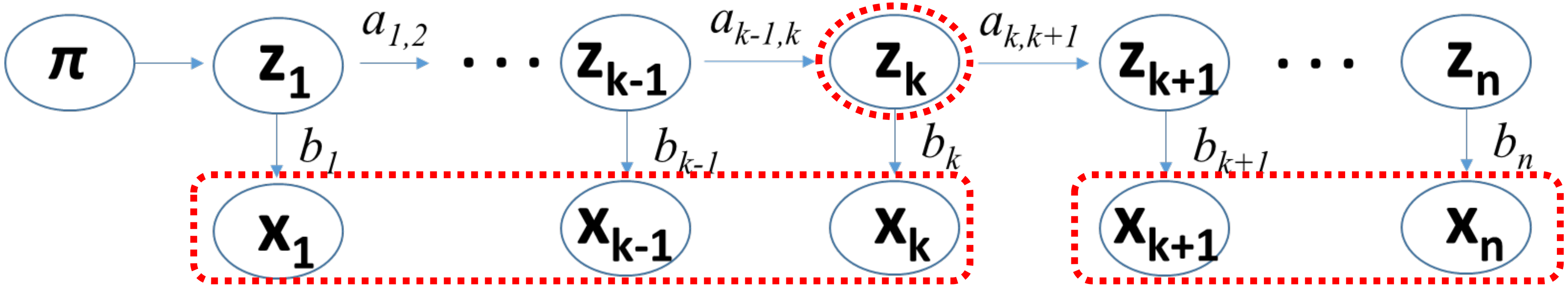
	Forward algorithm	Forward-Backward algorithm	Viterbi algorithm
Notation	$p(z_k \| x_{1:k})$	$p(z_k \| x_{1:n})$	$p(z_{1:n} \| x_{1:n})$



Decoding problem: p(Z|X, π, a, b)

❑ Three algorithms in the decoding problem:

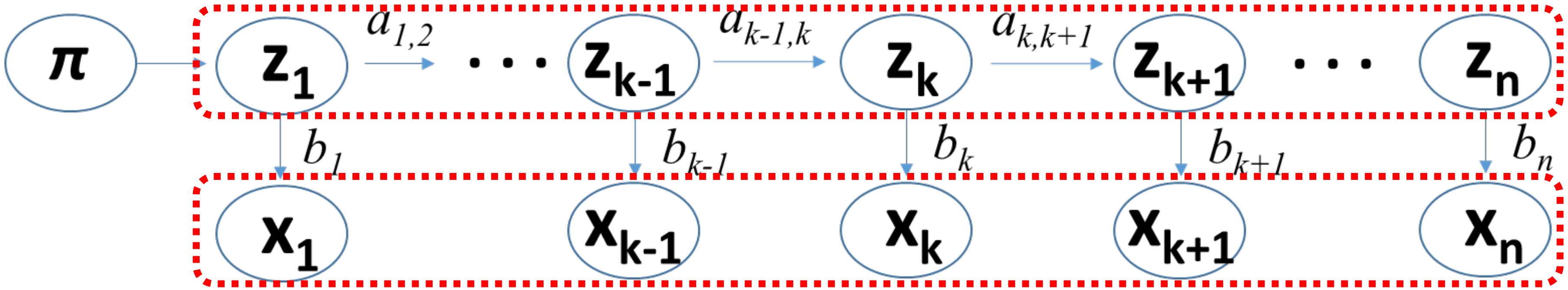
	Forward algorithm	Forward-Backward algorithm	Viterbi algorithm
Notation	$p(z_k \| x_{1:k})$	$p(z_k \| x_{1:n})$	$p(z_{1:n} \| x_{1:n})$



Decoding problem: $p(Z|X, \pi, a, b)$

- Three algorithms in the decoding problem:

	Forward algorithm	Forward-Backward algorithm	Viterbi algorithm
Notation	$p(z_k \mid x_{1:k})$	$p(z_k \mid x_{1:n})$	$p(z_{1:n} \mid x_{1:n})$



Prerequisite items you need to know before going further

- Marginalization
- $p(A,B,C) = p(A)p(B|A)p(C|A,B)$
- $p(A,B,C|D) = p(A|D)p(B|A,D)p(C|A,B,D)$

Marginalization

- Marginalization is a procedure to get rid of an influence of the other random variables from a joint distribution.
- A joint distribution, p(z, x), can be represented as p(x) or p(z) by marginalization out or over the variable z or x, respectively.
  - Marginal distribution of z, p(z) is the result of marginalization over x in p(z,x),
  - And vice versa.

	x_1(Smile)	x_2(Gloomy)	…	x_n(Cry)	
z_1 (Happy)	$p(z_1,x_1)$	$p(z_1,x_2)$	…	$p(z_1,x_n)$	$p(z_1)$
z_2(Sad)	$p(z_2,x_1)$	$p(z_2,x_2)$	…	$p(z_2,x_n)$	$p(z_2)$
	$p(x_1)$	$p(x_2)$	…	$p(x_n)$	

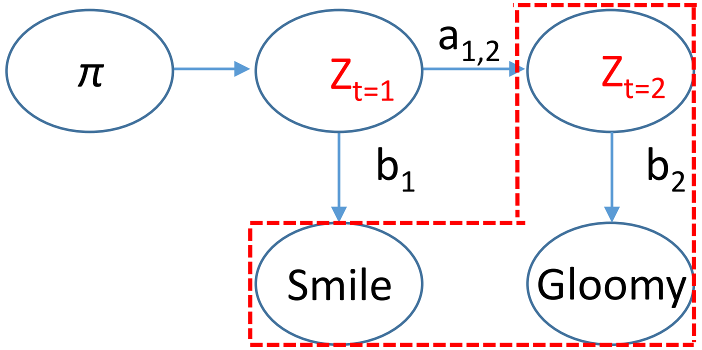
$$p(\mathrm{z}) = \sum_{\mathrm{x}} p(\mathrm{x},\mathrm{z})$$

- Marginal distribution of z
- Marginalization out x

$$p(\mathrm{x}) = \sum_{\mathrm{z}} p(\mathrm{x},\mathrm{z})$$

- Marginal distribution of X
- Marginalization out z

Marginalization – cont.



	$z_{t=2}$, $x_{1:2}$
$z_{t=1}$= Happy	$p(z_{t=1}$=Happy, $z_{t=2}$, $x_{1:2})$
$z_{t=1}$= Sad	$p(z_{t=1}$=Sad, $z_{t=2}$, $x_{1:2})$
	$p(z_{t=2}$, $x_{1:2})$

$$Z \in \{Happy, Sad\}$$

$$X \in \{Smile, Gloomy\}$$

$$p(z_2, x_{1:2}) = \sum_{z_1} p(z_1, z_2, x_{1:2})$$

$$= p(z_1 = happy, z_2, x_{1:2}) + p(z_1 = sad, z_2, x_{1:2})$$

p(A,B,C) = p(A)p(B|A)p(C|A,B)

$$p(x,y) = p(x \mid y)\, p(y)$$

Product rule / Chain rule

$$p(A,B,C) = p(C,B \mid A)\, p(A)$$

$$= p(C \mid B,A)\, p(B \mid A)\, p(A)$$

$$p(A,B,C) = p(A)\, p(B \mid A)\, p(C \mid A,B)$$

$$p(C,B,A) = p(C,B \mid A)\, p(A)$$

$$p(C,B,A) = p(C \mid B,A)\, p(B,A)$$

$$p(C,B \mid A)\, p(A) = p(C \mid B,A)\, p(B,A)$$

$$p(C,B \mid A) = \frac{p(C \mid B,A)\, p(B,A)}{p(A)}$$

$$= p(C \mid B,A)\, p(B \mid A)$$

p(A,B,C|D) = p(A|D)p(B|A,D)p(C|A,B,D)

$$p(A,B,C,D) = p(C,B,A,D)$$

$$p(A,B,C \mid D)\,p(D) = p(C \mid B,A,D)\,p(B,A,D)$$

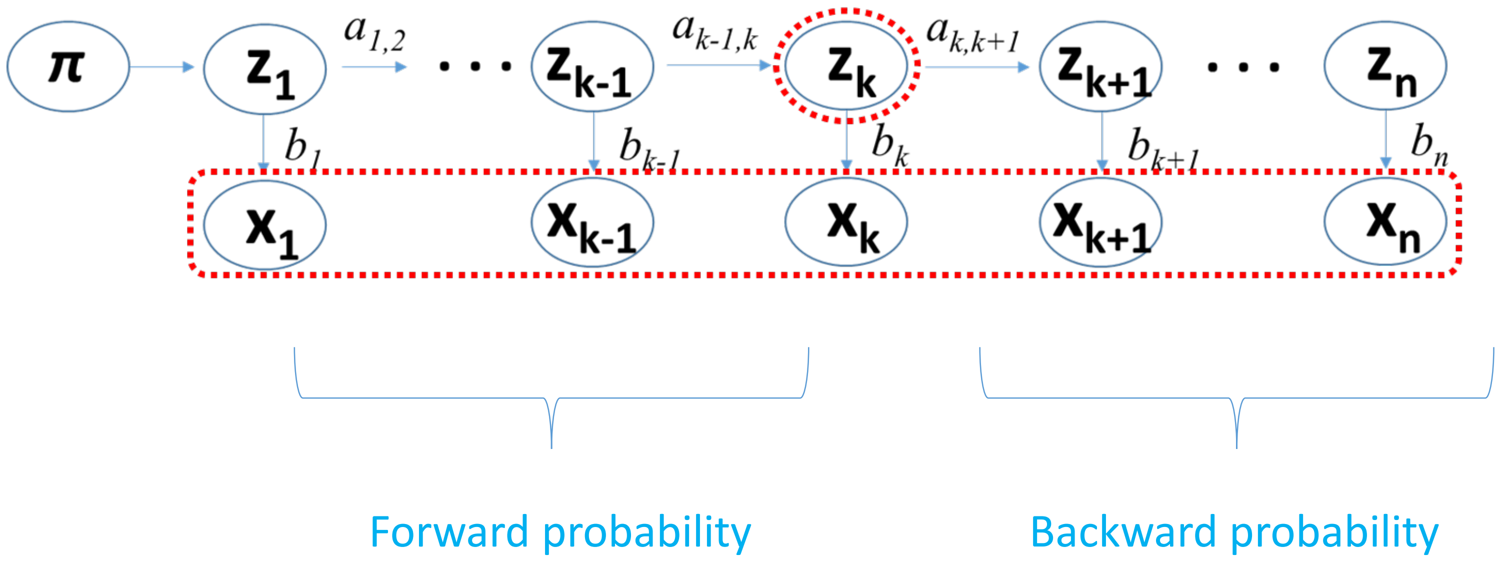
$$p(A,B,C \mid D) = \frac{p(C \mid B,A,D)\,p(B,A,D)}{p(D)}$$

$$= \frac{p(C \mid B,A,D)\,p(B \mid A,D)\,p(A,D)}{p(D)}$$

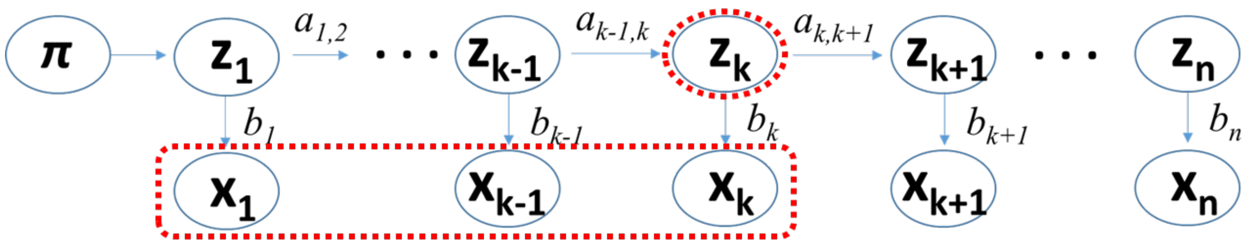
$$= \frac{p(C \mid B,A,D)\,p(B \mid A,D)\,p(A \mid D)\,p(D)}{p(D)}$$

$$p(A,B,C \mid D) = p(A \mid D)\,p(B \mid A,D)\,p(C \mid A,B,D)$$

Forward-Backward algorithm



1) Forward probability (α)



$$p(z_k, x_1, x_2, \ldots, x_k) = \alpha_k(z_k) = \sum_{z_{k-1}} p(z_{k-1}, z_k, x_1, x_2, \ldots, x_k)$$ marginalization

$$= \sum_{z_{k-1}} p(\underbrace{z_{k-1}, x_1, x_2, \ldots, x_{k-1}}_{A}, \underbrace{z_k}_{B}, \underbrace{x_k}_{C})$$ p(A,B,C) = p(A)p(B|A)p(C|A,B)

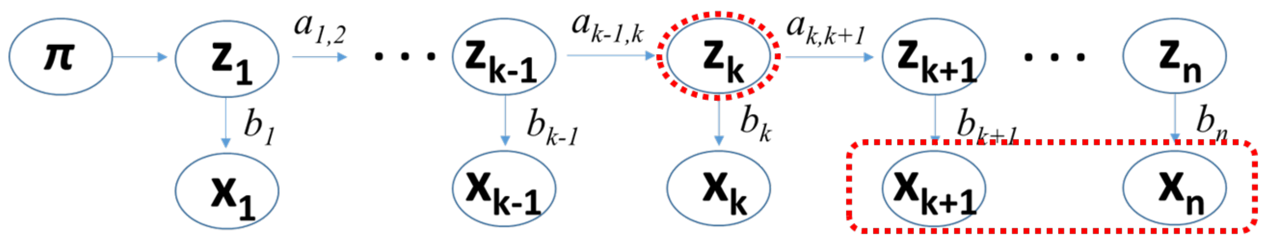
$$\alpha_k(z_k) = \sum_{z_{k-1}} p(z_{k-1}, x_1, x_2, \ldots, x_{k-1})\, p(z_k \mid z_{k-1}, x_1, x_2, \ldots, x_{k-1})\, p(x_k \mid z_k, z_{k-1}, x_1, x_2, \ldots, x_{k-1})$$

$$= \sum_{z_{k-1}} p(z_{k-1}, x_1, x_2, \ldots, x_{k-1})\, p(z_k \mid z_{k-1})\, p(x_k \mid z_k)$$

$$\alpha_k(z_k) = \begin{cases} \sum_{z_{k-1}} \alpha_{k-1}(z_{k-1})\, a_{k-1,k} b_k & k \geq 2 \\ \pi_k\, b_k & k = 1 \end{cases}$$

$$p(z_1, x_1) = p(z_1)\, p(x_1 \mid z_1) = \pi_1\, b_1$$

2) Backward probability (β)



$$p(x_{k+1},\ldots,x_n \mid z_k) = \beta_k(z_k) = \sum_{z_{k+1}} p(z_{k+1}, x_{k+1},\ldots,x_n \mid z_k)$$ marginalization

$$= \sum_{z_{k+1}} p(\underbrace{z_{k+1}}_{A}, \underbrace{x_{k+1}}_{B}, \underbrace{x_{k+2}, x_{k+3},\ldots,x_n}_{C} \mid \underbrace{z_k}_{D})$$

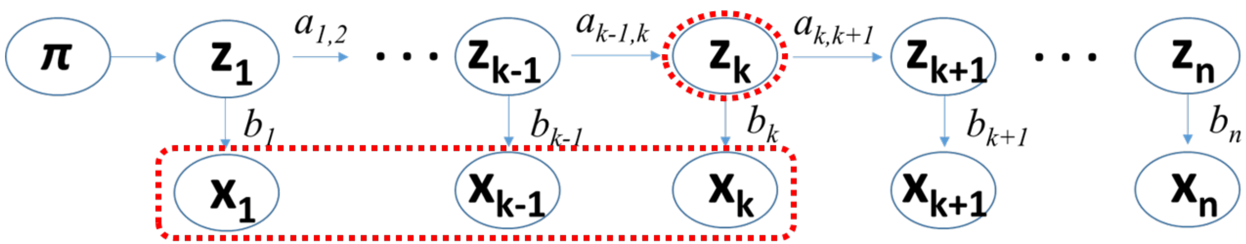
p(A,B,C|D) = p(A|D)p(B|A,D)p(C|A,B,D)

$$\beta_k(z_k) = \sum_{z_{k+1}} p(z_{k+1} \mid z_k)\, p(x_{k+1} \mid z_{k+1}, z_k)\, p(x_{k+2},\ldots,x_n \mid x_{k+1}, z_{k+1}, z_k)$$

$$= \sum_{z_{k+1}} p(z_{k+1} \mid z_k)\, p(x_{k+1} \mid z_{k+1})\, p(x_{k+2},\ldots,x_n \mid z_{k+1})$$

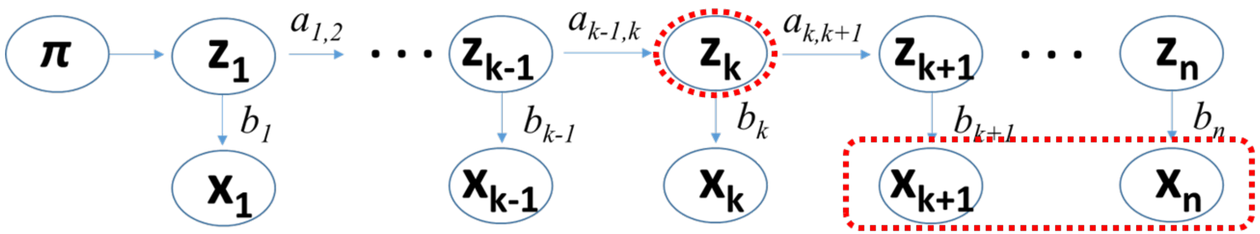
$$\beta_k(z_k) = \sum_{z_{k+1}} a_{k,k+1} b_{k+1} \beta_{k+1}(z_{k+1})$$

Now we know probabilities of forward and backward



$$\alpha_k(z_k) = \begin{cases} \sum_{z_{k-1}} \alpha_{k-1}(z_{k-1})\, a_{k-1,k} b_k & k \geq 2 \\ \pi_k\, b_k & k = 1 \end{cases}$$

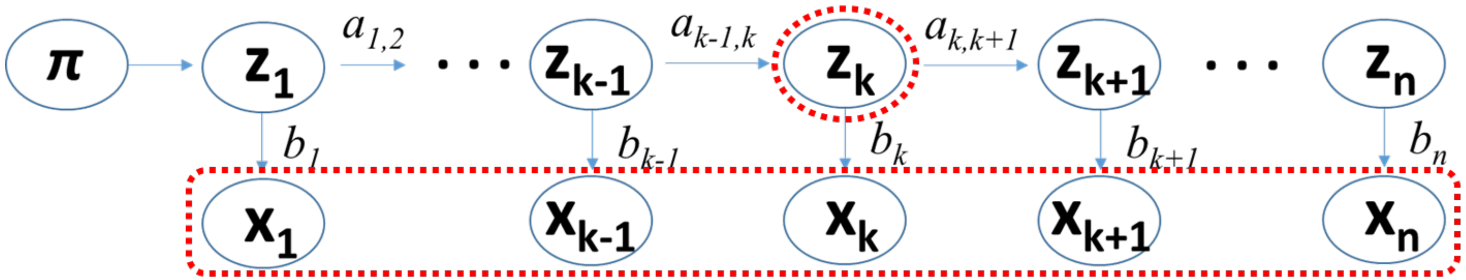
Forward probability α



$$\beta_k(z_k) = \sum_{z_{k+1}} a_{k,k+1} b_{k+1} \beta_{k+1}(z_{k+1})$$

Backward probability β

Back to Forward-Backward algorithm – cont.



$$p(z_k, x_{1:n}) = p(z_k, x_1, x_2, \ldots, x_k, x_{k+1}, \ldots, x_n)$$

$$= p(z_k, x_1, x_2, \ldots, x_k)\ p(x_{k+1}, \ldots, x_n \mid z_k)$$

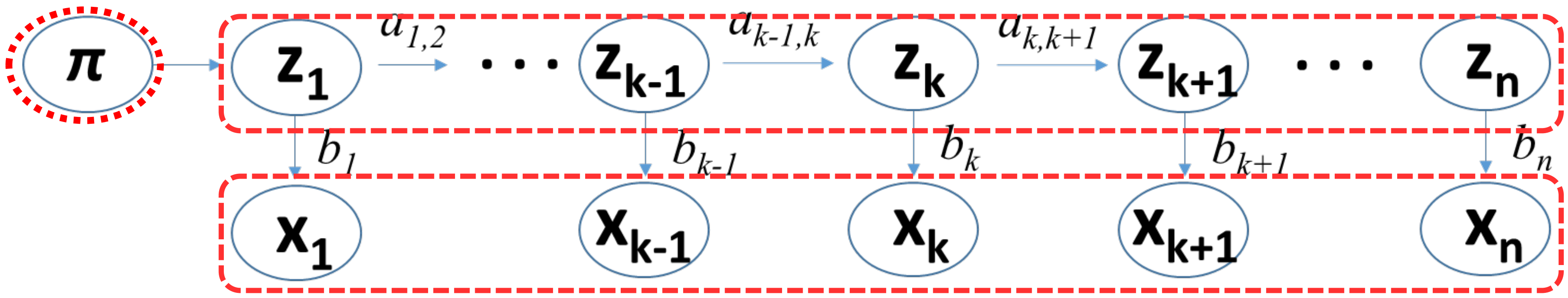
1) Forward probability 2) Backward probability

$$\alpha_k(z_k) = \begin{cases} \sum_{z_{k-1}} \alpha_{k-1}(z_{k-1})\, a_{k-1,k} b_k & k \geq 2 \\ \pi_k\, b_k & k = 1 \end{cases}$$

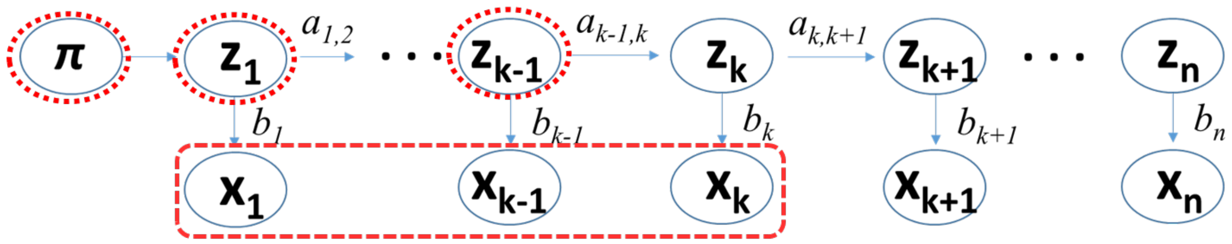
$$\beta_k(z_k) = \sum_{z_{k+1}} a_{k,k+1} b_{k+1} \beta_{k+1}(z_{k+1})$$

Viterbi Decoding

- It is an algorithm which determines the label sequences($z_{1:n}$) from observed data.



Viterbi Decoding – cont.



$$p(A,B) = p(A)p(B \mid A)$$
$$p(B \mid A) \propto p(B,A)$$

$$z_{1:n}^* = \arg\max_{z_{1:n}} p(z_{1:n} \mid x_{1:n}) = \arg\max_{z_{1:n}} p(z_{1:n}, x_{1:n})$$

$$V_k(z_k) = \max_{z_{1:k-1}} p(z_{1:k}, x_{1:k}) = \max_{z_{1:k-1}} p(z_1,\dots,z_{k-1},x_1,\dots,x_{k-1},z_k,x_k)$$

$$= \max_{z_{1:k-1}} p(z_k, x_k \mid z_1,\dots,z_{k-1},x_1,\dots,x_{k-1})\, p(z_1,\dots,z_{k-1},x_1,\dots,x_{k-1})$$

z_{k-1} only affects to z_k

$$= \max_{z_{1:k-1}} p(z_k, x_k \mid z_{k-1})\, p(z_1,\dots,z_{k-1},x_1,\dots,x_{k-1})$$

$$= \max_{z_{k-1}} p(z_k, x_k \mid z_{k-1}) \max_{z_{1:k-2}} p(z_1,\dots,z_{k-1},x_1,\dots,x_{k-1})$$

p(A,B|C) = p(A|C)p(B|A,C)

A B C

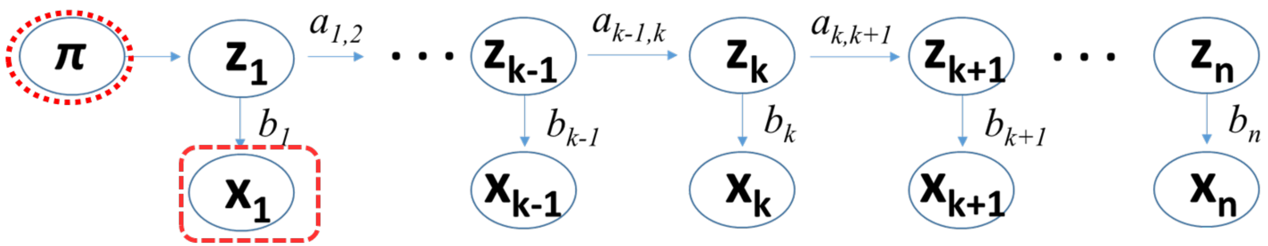
$$= \max_{z_{k-1}} p(z_k, x_k \mid z_{k-1}) V_{k-1}(z_{k-1}) = \max_{z_{k-1}} p(x_k \mid z_k, z_{k-1})\, p(z_k \mid z_{k-1}) V_{k-1}(z_{k-1})$$

Emission probability "b" Transition probability "a"

$$= p(x_k \mid z_k) \max_{z_{k-1}} p(z_k \mid z_{k-1}) V_{k-1}(z_{k-1})$$

$$V_k(z_k) = b_k \max_{z_{k-1}} a_{k-1,k} V_{k-1}(z_{k-1})$$

Viterbi Decoding – cont.



- Initialize

$$V_1(z_1) = b_1 \max_{z_0} a_{0,1} V_0(z_0) = b_1 \pi_1$$

- Iterate until time k → n

$$V_k(z_k) = b_k \max_{z_{k-1}} a_{k-1,k} V_{k-1}(z_{k-1})$$

Select a link which maximizes the connection from Z_(k-1) to Z_k. See the example in the next slide.

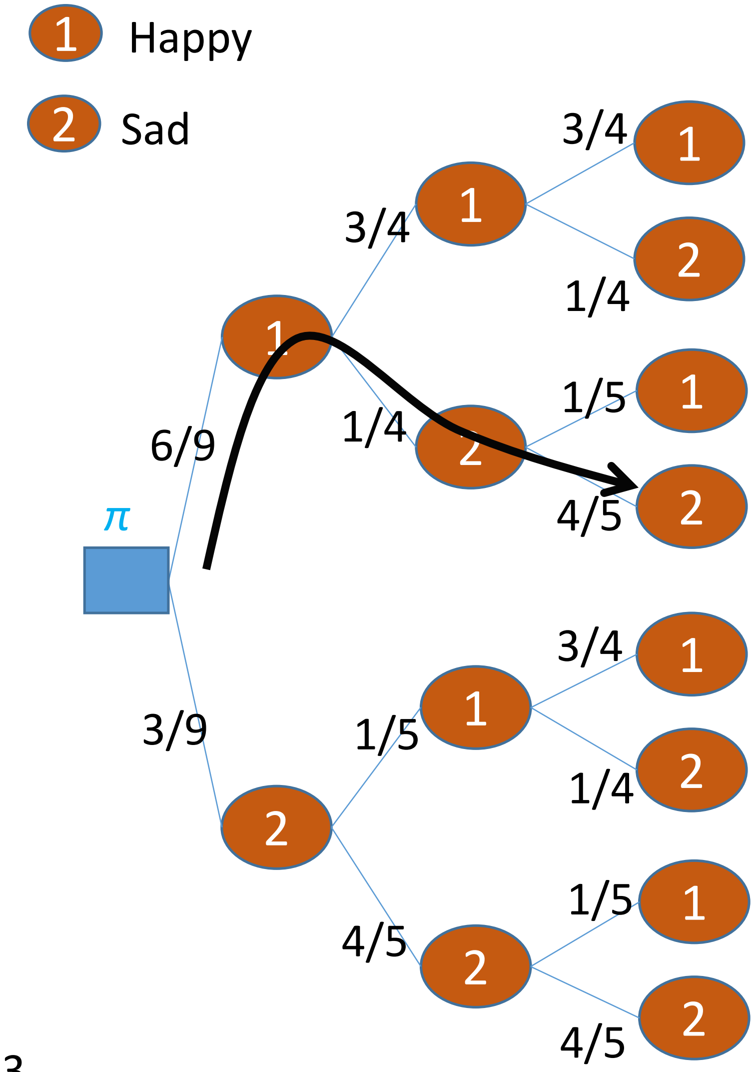
From above, we are interested in the series of "$z_{1:n}$", which maximizes $V_k(z_k)$.

$$z^*_{1:n} = \arg\max_{z_{1:n}} p(z_{1:n} \mid x_{1:n}) = \arg\max_{z_{1:n}} p(z_{1:n}, x_{1:n})$$

Decoding examples with Viterbi algorithm

Example: observed data sequence: Smile-Gloomy-Gloomy

- What is the sequence of hidden states which generates “Smile – Gloomy – Gloomy”?



- π : initial

Happy	Sad
6/9	3/9

- a : transition

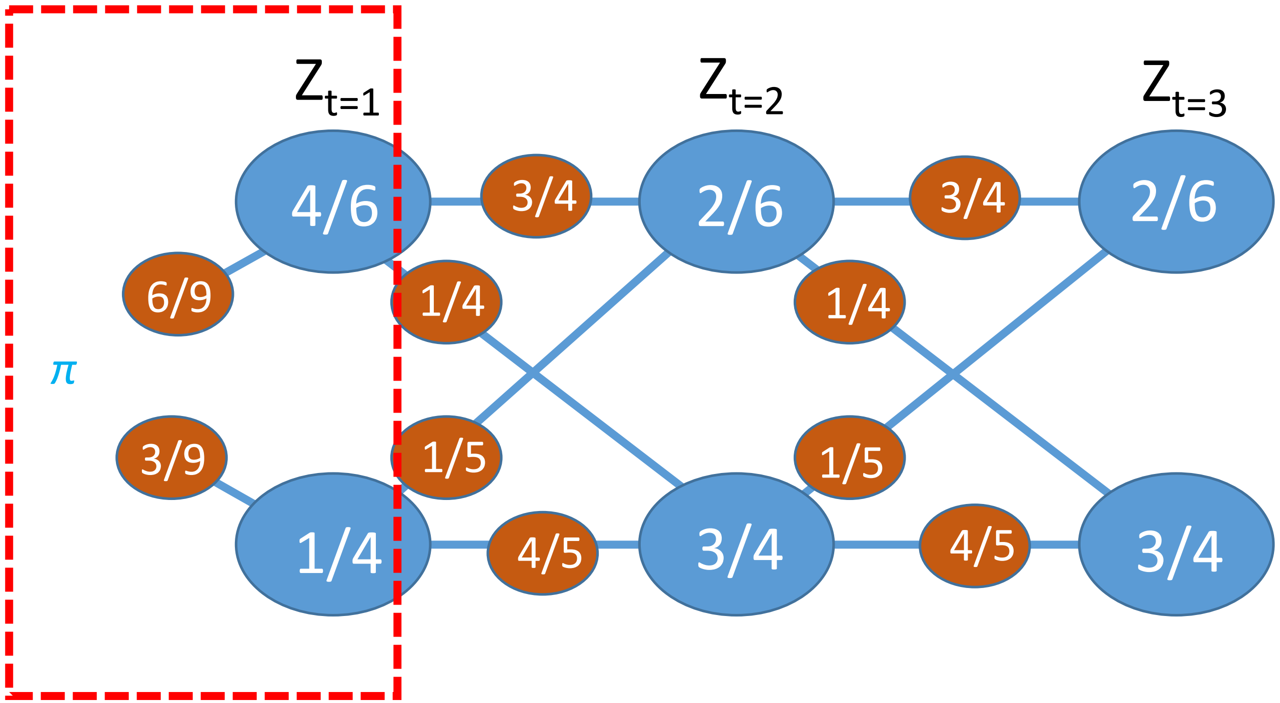
	Happy	Sad
Happy	3/4	1/4
Sad	1/5	4/5

- b : emission

Happy		Sad	
Smile	Gloomy	Smile	Gloomy
4/6	2/6	1/4	3/4

Cases	**Probability (π x b x a x b x a x b)**	
p(H-H-H | X, π, a, b)	6/9 x 4/6 x 3/4 x 2/6 x 3/4 x 2/6	0.02778
p(H-H-S | X, π, a, b)	6/9 x 4/6 x 3/4 x 2/6 x 1/4 x 3/4	0.02083
p(H-S-H | X, π, a, b)	6/9 x 4/6 x 1/4 x 3/4 x 1/5 x 2/6	0.00556
p(H-S-S | X, π, a, b)	6/9 x 4/6 x 1/4 x 3/4 x 4/5 x 3/4	0.05000
p(S-S-H | X, π, a, b)	3/9 x 1/4 x 4/5 x 3/4 x 1/5 x 2/6	0.00335
p(S-H-S | X, π, a, b)	3/9 x 1/4 x 1/5 x 2/6 x 1/4 x 3/4	0.0009
p(S-H-H | X, π, a, b)	3/9 x 1/4 x 1/5 x2/6 x 3/4 x 2/6	0.0014
p(S-S-S | X, π, a, b)	3/9 x 1/4 x 4/5 x 3/4 x4/5 x 3/4	0.03

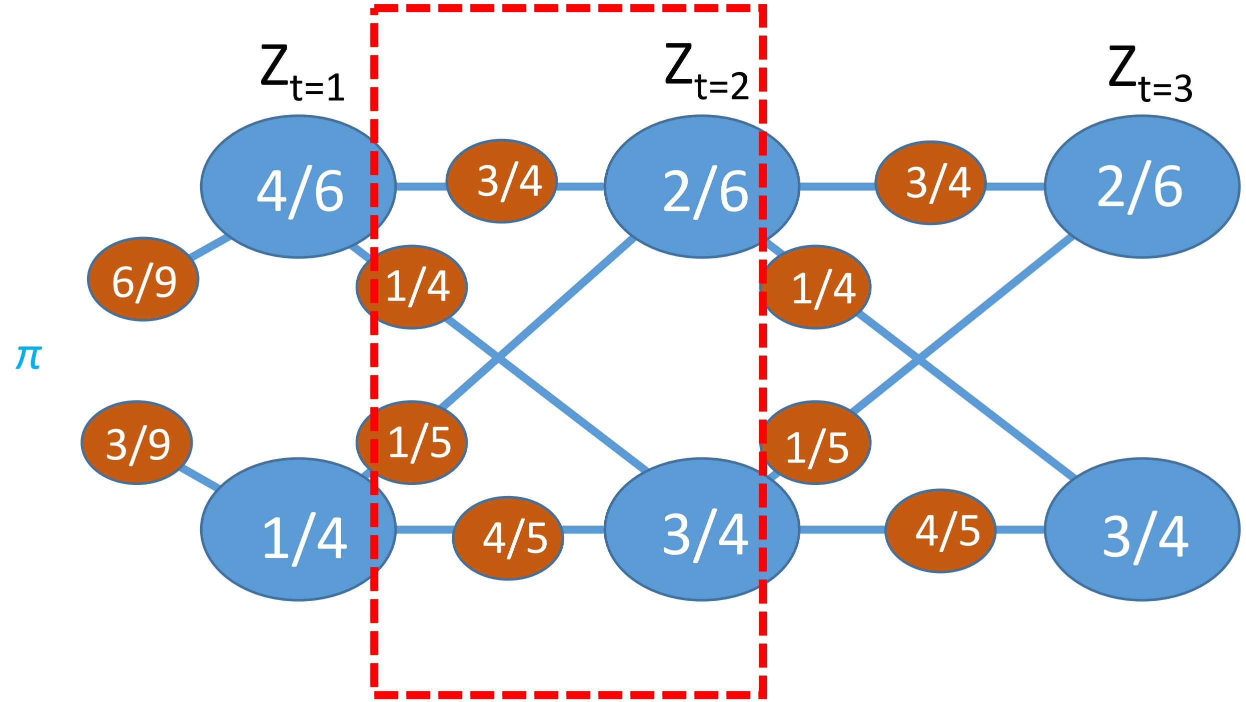
Example: observed data sequence: Smile-Gloomy-Gloomy



$$V_1(z_1) = b_1 \max_{z_0} a_{0,1} V_0(z_0) = b_1 \pi_1$$

	V(Z1)	V(Z2)	V(Z3)
Happy	6/9 x 4/6 = 0.444		
Sad	3/9 x 1/4 = 0.083		

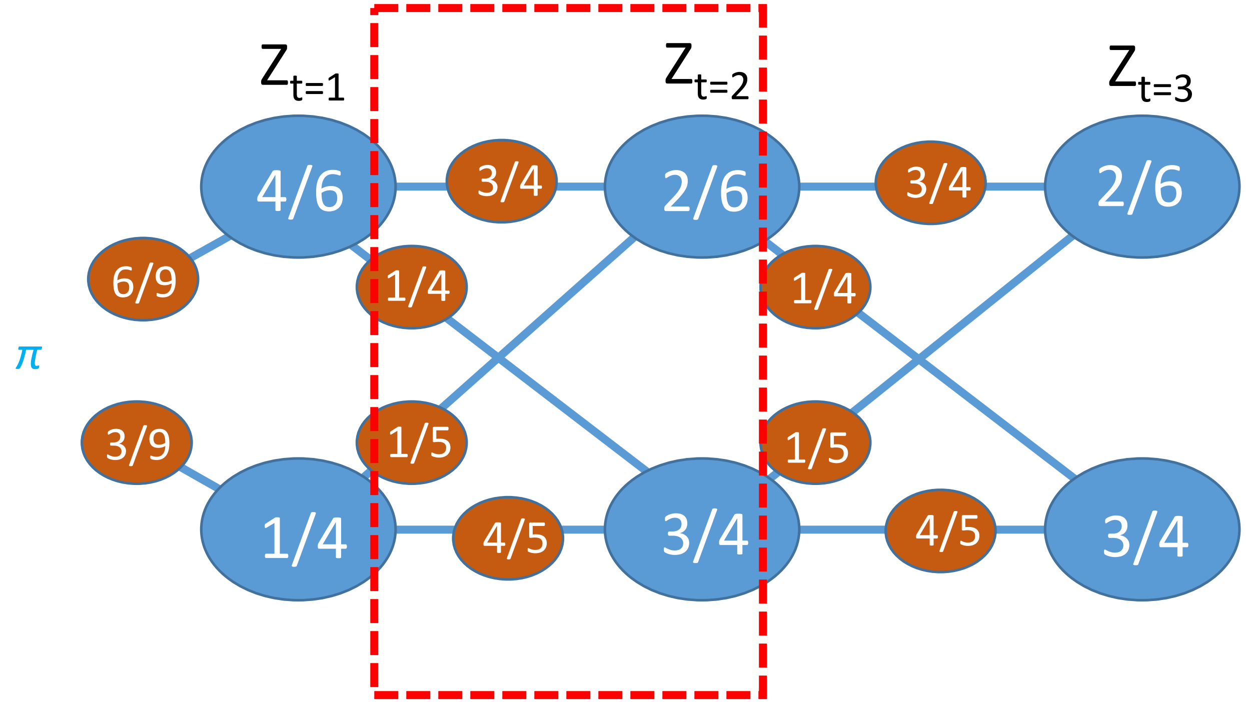
Example: observed data sequence: Smile-Gloomy-Gloomy



$$V_2(z_2) = b_2 \max_{z_1} a_{1,2} V_1(z_1)$$

	V(Z1)	V(Z2)	V(Z3)
Happy	6/9 x 4/6 = 0.444	3/4 x 2/6 x v(z1-happy) = 0.111 1/5 x 2/6 x v(z1-sad) = 0.00553	
Sad	3/9 x 1/4 = 0.083	1/4 x 3/4 x v(z1-happy) = 0.0833 4/5 x 3/4 x v(z1-sad) = 0.0498	

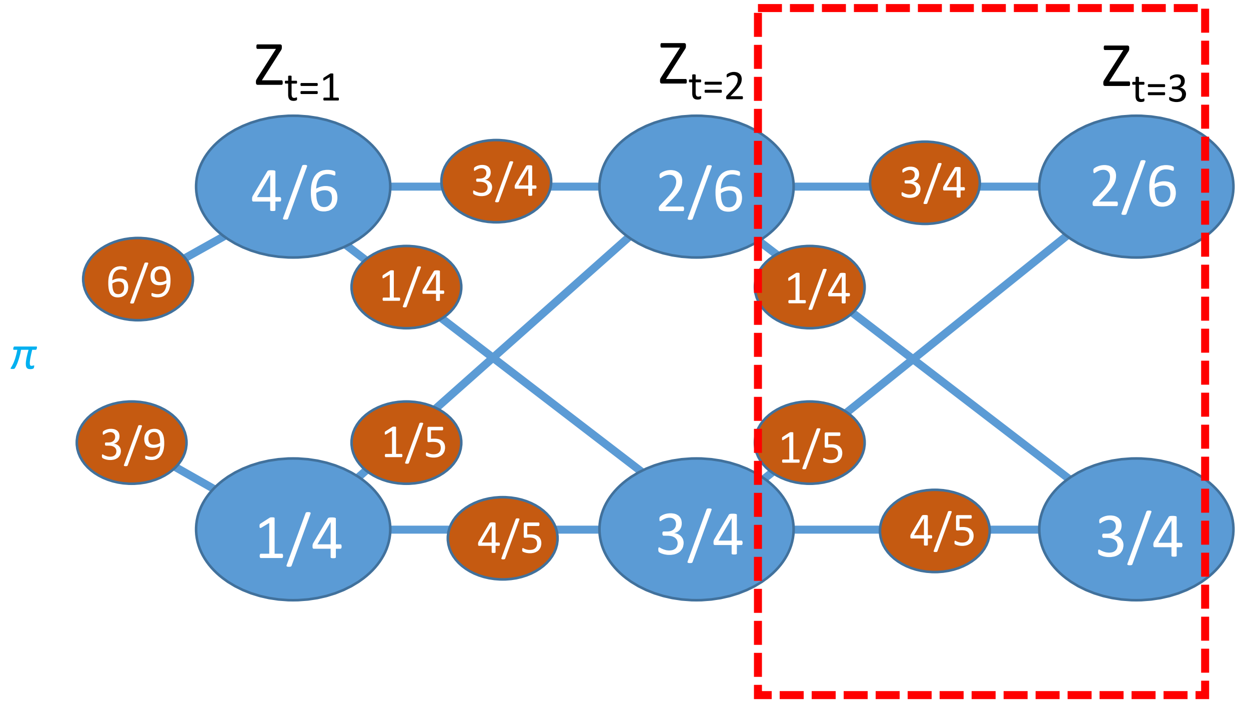
Example: observed data sequence: Smile-Gloomy-Gloomy



$$V_2(z_2) = b_2 \max_{z_1} a_{1,2} V_1(z_1)$$

	V(Z1)	V(Z2)	V(Z3)
Happy	6/9 x 4/6 = 0.444	3/4 x 2/6 x v(z1-happy) = 0.111 ~~1/5 x 2/6 x v(z1-sad) = 0.00553~~	
Sad	3/9 x 1/4 = 0.083	1/4 x 3/4 x v(z1-happy) = 0.0833 ~~4/5 x 3/4 x v(z1-sad) = 0.0498~~	

Example: observed data sequence: Smile-Gloomy-Gloomy

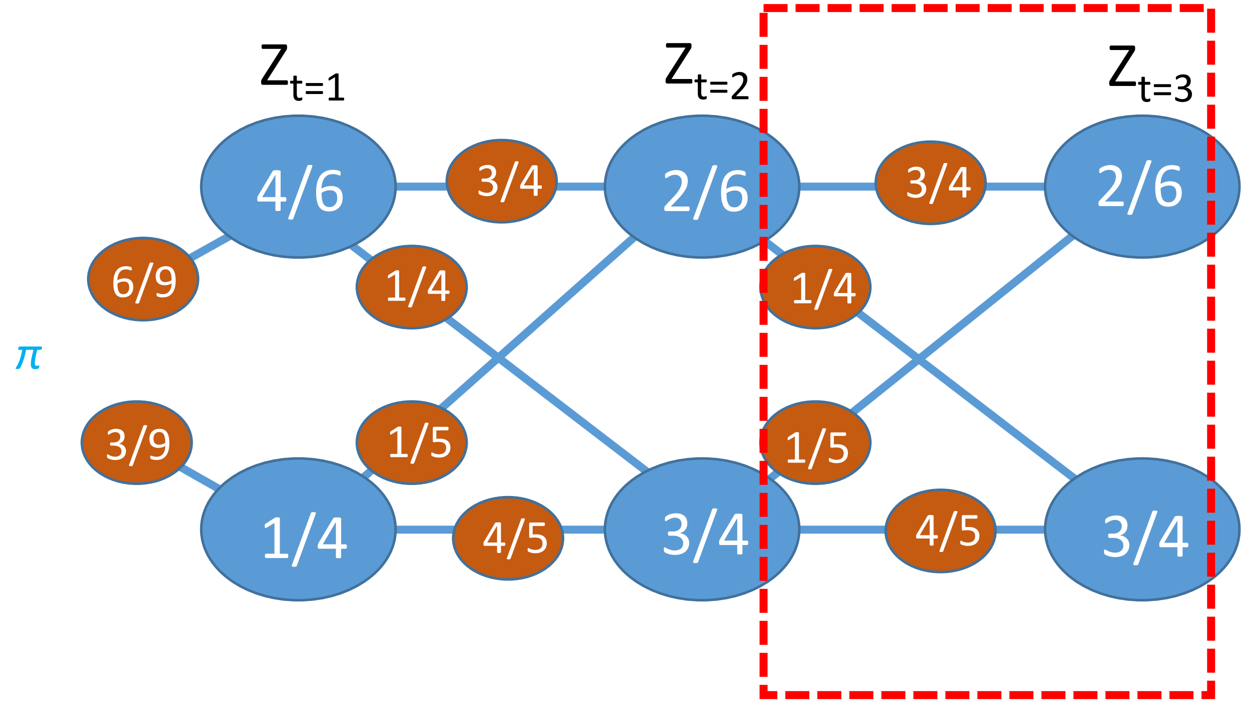


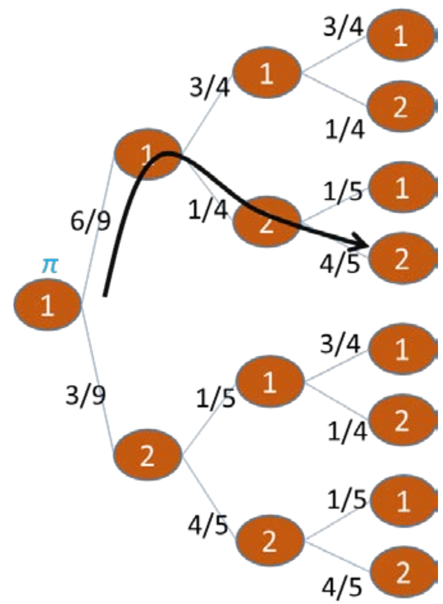
$$V_3(z_3) = b_3 \max_{z_2} a_{2,3} V_2(z_2)$$

	V(Z1)	V(Z2)	V(Z3)
Happy	6/9 x 4/6 = 0.444	3/4 x 2/6 x v(z1-happy) = 0.111 ~~1/5 x 2/6 x v(z1-sad) = 0.00553~~	3/4 x 2/6 x v(z2-happy) = 0.0278 1/5 x 2/6 x v(z2-sad) = 0.00556
Sad	3/9 x 1/4 = 0.083	1/4 x 3/4 x v(z1-happy) = 0.0833 ~~4/5 x 3/4 x v(z1-sad) = 0.0498~~	1/4 x 3/4 x v(z2-happy) = 0.0208 4/5 x 3/4 x v(z2-sad)= 0.04998

Example: observed data sequence: Smile-Gloomy-Gloomy

$$V_3(z_3) = b_3 \max_{z_2} a_{2,3} V_2(z_2)$$





	V(Z1)	V(Z2)	V(Z3)
Happy	6/9 x 4/6 = 0.444	3/4 x 2/6 x v(z1-happy) = 0.111 ~~1/5 x 2/6 x v(z1-sad) = 0.00553~~	3/4 x 2/6 x v(z2-happy) = 0.0278 ~~1/5 x 2/6 x v(z2-sad) = 0.00556~~
Sad	3/9 x 1/4 = 0.083	1/4 x 3/4 x v(z1-happy) = 0.0833 ~~4/5 x 3/4 x v(z1-sad) = 0.0498~~	~~1/4 x 3/4 x v(z2-happy) = 0.0208~~ 4/5 x 3/4 x v(z2-sad)= 0.04998

$Z^*_{1:n}$ ⇒ *HAPPY* → *SAD* → *SAD*

Learning Problem with Baum-Welch algorithm

Prerequisite items you need to know before tackling Baum-Welch algorithm

- ❑ Lagrange method
- ❑ Jensen's inequality
- ❑ Generalized Expectation and Maximization (EM)

Lagrange method

- A method which converts a constrained optimization problem to a non-constrained optimization problem.

$$\min \quad f(x, y) = x^2 + y^2$$
$$s.t \quad x + y = 100$$

$$\Rightarrow$$

$$\min \quad f(x, y, \lambda) = x^2 + y^2 + \lambda(x + y - 100)$$

$$\frac{\partial f(x, y, \lambda)}{\partial x} = 0$$

$$\frac{\partial f(x, y, \lambda)}{\partial y} = 0$$

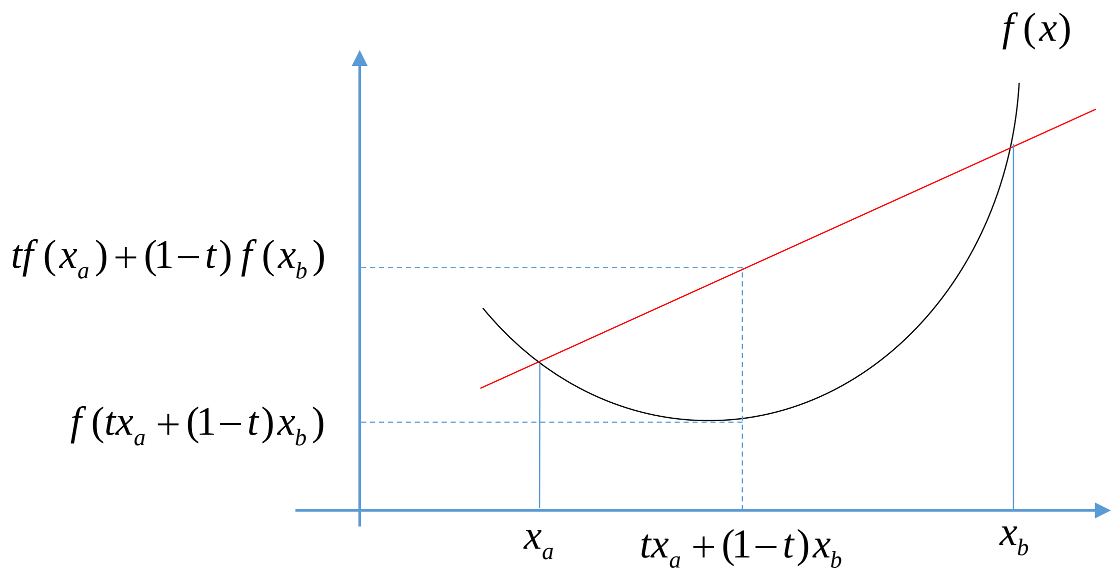
$$\frac{\partial f(x, y, \lambda)}{\partial \lambda} = 0$$

Jensen’s inequality

- When a function is convex or concave, the following inequality condition is satisfied.

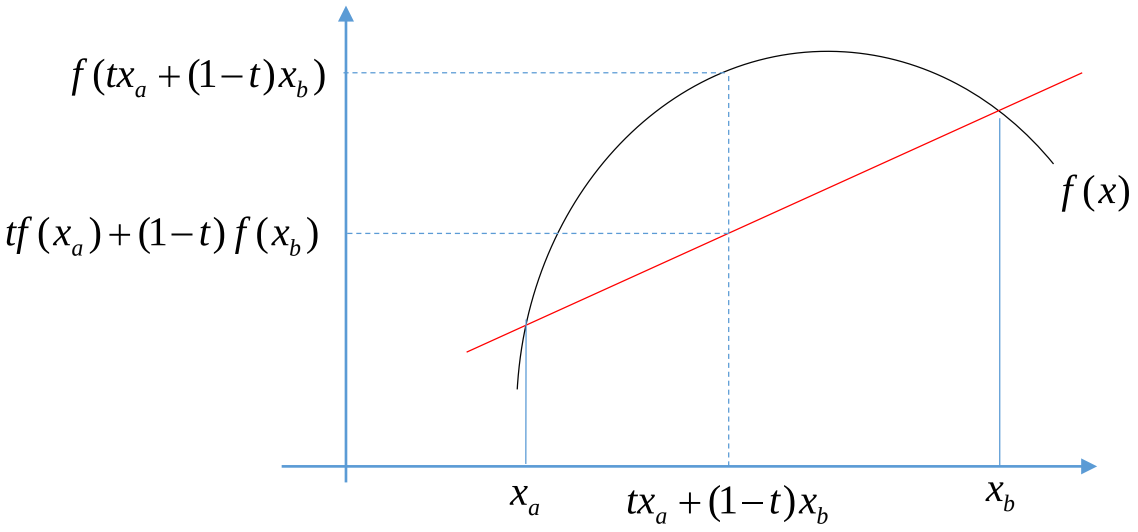
Convex function

$$E[f(X)] \geq f(E[X])$$



Concave function

$$E[f(X)] \leq f(E[X])$$



Generalized EM method

- Log likelihood function?

$$\{\pi, a, b\} \in \theta$$

$$\ln p(X \mid \theta) = \ln \sum_Z p(X, Z \mid \theta)$$

Marginalize over Z

$$= \ln \sum_Z q(Z) \frac{p(X, Z \mid \theta)}{q(Z)}$$

$$E[X] = \sum xp(x)$$

$$= \ln E_Z\left[\frac{p(X, Z \mid \theta)}{q(Z)}\right]$$

Jensen's inequality (a log function is concave)

$$f(E[X]) \geq E[f(X)]$$

$$\geq E_Z \ln\left[\frac{p(X, Z \mid \theta)}{q(Z)}\right]$$

$$\geq E_Z[\ln p(X, Z \mid \theta)] - E_Z[\ln q(Z)]$$

$$\geq \sum_Z q(Z) \ln p(X, Z \mid \theta) - \sum_Z q(Z) \ln q(Z)$$

Generalized EM method – cont.

$$\ln p(X\mid\theta) \ge \sum_Z q(Z)\ln p(X,Z\mid\theta) - \sum_Z q(Z)\ln q(Z)$$

$$\ge \sum_Z q(Z)\ln p(Z\mid X,\theta)p(X\mid\theta) - \sum_Z q(Z)\ln q(Z)$$

$$\ge \sum_Z q(Z)\ln \frac{p(Z\mid X,\theta)p(X\mid\theta)}{q(Z)}$$

$$\ge \sum_Z q(Z)\ln \frac{p(Z\mid X,\theta)}{q(Z)} + \sum_Z q(Z)\ln p(X\mid\theta)$$

$$\ge \sum_Z q(Z)\ln \frac{p(Z\mid X,\theta)}{q(Z)} + \ln p(X\mid\theta)$$

$$\ge \ln p(X\mid\theta) - \sum_Z q(Z)\ln \frac{q(Z)}{p(Z\mid X,\theta)}$$

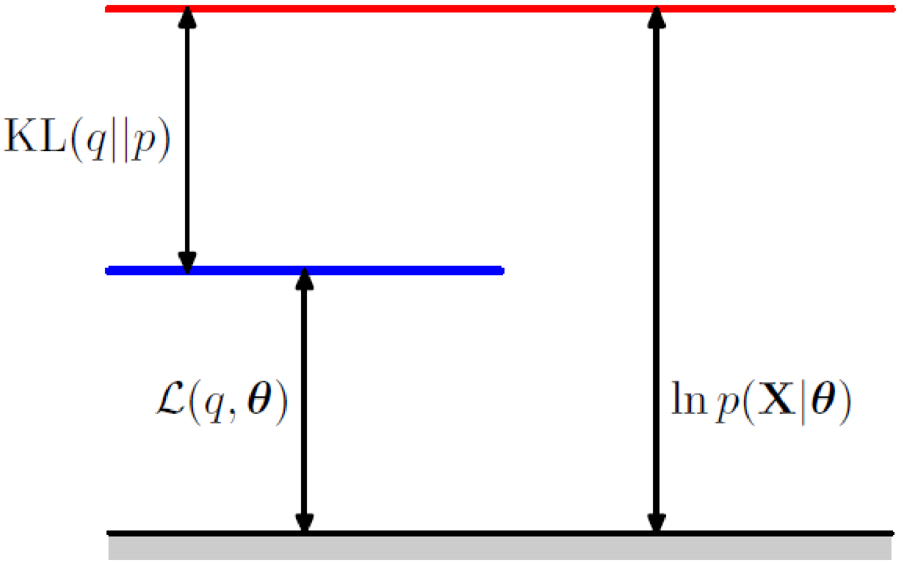
- Product rule

p(A,B|C) = p(A|C)p(B|A,C)

Generalized EM method – cont.

$$L(q,\theta) \geq \ln p(X\mid\theta) - \sum_{Z} q(Z)\ln\frac{q(Z)}{p(Z\mid X,\theta)}$$

- Kullback-Leiber divergence, or KL divergence: KL (q(z)||p(z|z,θ))
- When q(z) is equal to p(Z|X, θ), KL divergence becomes 0 which is the minimum value (KL ≥ 0).



$$\ln p(X\mid\theta) = \ln\sum_{Z} p(X,Z\mid\theta)$$

Learning problem: Baum-Welch algorithm

- Same as other machine learning algorithms, the learning problem is to find parameters of a model (HMM) which explains the observed data set X.
  - p(X| π, a, b)
  - 1) π (initial state probability) , 2) a (transition probability), 3) b (emission probability)
- We have two sets of unknown variables:
  1) Hidden variable z which is a sequence of classes/clusters,
  2) Its relevant parameters: (π, a, b) .
- Same as GMM case, the log-likelihood function of HMM cannot be solved analytically
- We need to apply Expectation and Maximization method, which is called Baum-Welch algorithm in HMM.

Learning problem: Baum-Welch algorithm: Expectation step

- Expect the sequence of hidden variable "z_n" given X, *π, a, b*
  - p(Z|X, π, a, b)
  - The three parameters of HMM is from M-step (or randomly initialized in the first iteration).
    - 1) π (initial state probability), 2) a (transition probability), 3) b (emission probability)

Learning problem: Baum-Welch algorithm: Maximization step

- Maximize the log likelihood function to find the parameters of HMM given the sequence of hidden variables "z_n" and observation "X"

$$\max_{\theta} \sum_{Z} q(Z) \ln p(X, Z \mid \theta)$$

$$\max_{\pi,a,b} \sum_{Z} p(z \mid x, \pi, a, b) \ln\left(\pi \prod_{k=2}^{n} a_{k-1,k} \prod_{k=1}^{n} b_k \right)$$

$$\ln p(X \mid \theta) \geq \sum_{Z} q(Z) \ln p(X, Z \mid \theta) - \sum_{Z} q(Z) \ln q(Z)$$

$$\ln p(X \mid \theta) \geq \ln p(X \mid \theta) - \sum_{Z} q(Z) \ln \frac{q(Z)}{p(Z \mid X, \theta)}$$

$$\max_{\pi,a,b} \sum_{Z} p(z \mid x, \pi, a, b) \left(\ln \pi + \sum_{k=2}^{n} \ln a_{k-1,k} + \sum_{k=1}^{n} \ln b_k \right)$$

$s.t.$

$$\sum_{i=1}^{m} \pi_i = 1, \quad \sum_{i=1}^{n} a_{i,i+1} = 1, \quad \sum_{i=1}^{n} b_i = 1$$

Optimization problem using Lagrange method

Learning problem: Baum-Welch algorithm: Maximization step

$$\pi^{(t+1)} = \frac{p(z_1^i = 1 | X, \pi_t, a_t, b_t)}{\sum_{j=1}^{K} p(z_1^j = 1 | X, \pi_t, a_t, b_t)}$$

Evaluation problem

H H H S H S S

	H	S
H	2/4	2/4
S	1/3	1/3

$$a_{(i,j)}^{(t+1)} = \frac{\sum_{t=2}^{T} p(z_{t-1}^i = 1, z_t^j = 1 | X, \pi_t, a_t, b_t)}{\sum_{t=2}^{T} p(z_{t-1}^i = 1 | X, \pi_t, a_t, b_t)}$$

- Numerator
  - Count transits from one hidden variable to the other
- Denominator
  - Count the hidden variable: see example above

$$b_{(i,j)}^{(t+1)} = \frac{\sum_{t=1}^{T} p(z_t^i = 1 | X, \pi_t, a_t, b_t)\delta(idx(x_t) = j)}{\sum_{t=1}^{T} p(z_t^i = 1 | X, \pi_t, a_t, b_t)}$$

- Numerator
  - Count occurrence of the random variable (x) in which you are interested from a particular hidden variable.
- Denominator
  - Count the hidden variable appeared.

H H H S H S S

S C S S C S C

Happy		Sad	
Smile	Cry	Smile	Cry
2/4	2/4	2/3	1/3

Learning problem: Baum-Welch algorithm

- Initialize the parameters: Θ^0

$\{\pi, a, b\} \in \theta$

Expectation Step

- **E**xpectation: finding Z given Θ^t , X
- p(Z|X, π, a, b): decoding problem

Maximization Step

- **M**aximization: finding Θ^t given X, Z

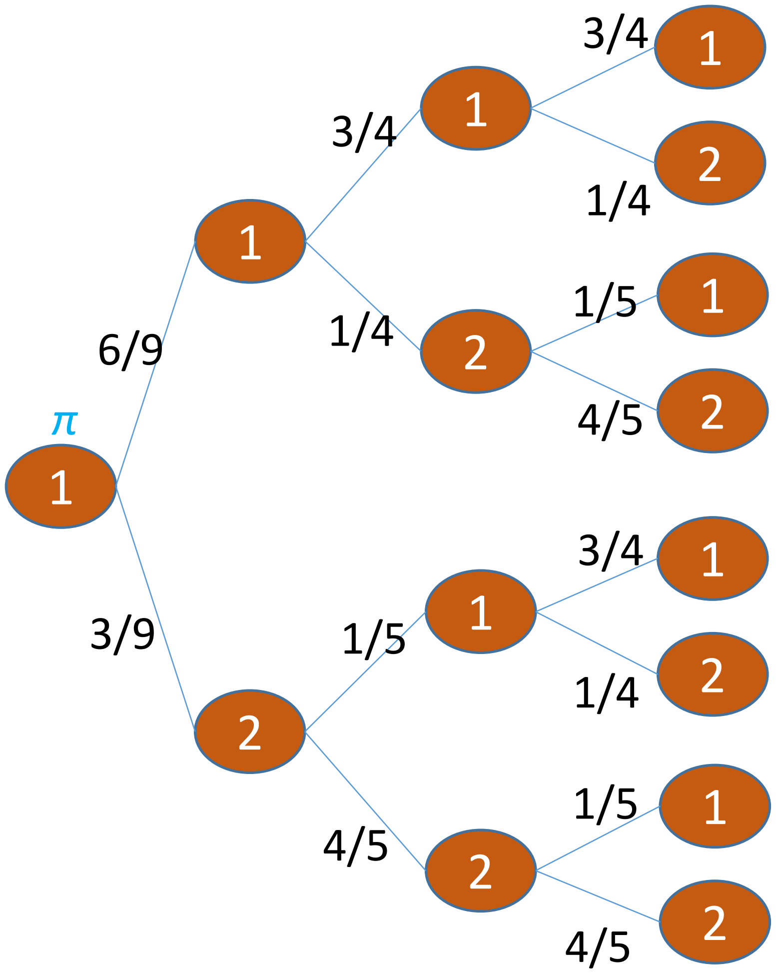
❑ The process is repeated until the three parameters of HMM do not change much.

- 1) π , 2) a (transition probability), 3) b (emission probability)

Backup slide

Decoding example with Viterbi algorithm – (2)

Example: observed data sequence: Smile-Gloomy-Smile



- *π* : initial

Happy	Sad
6/9	3/9

- *a* : transition

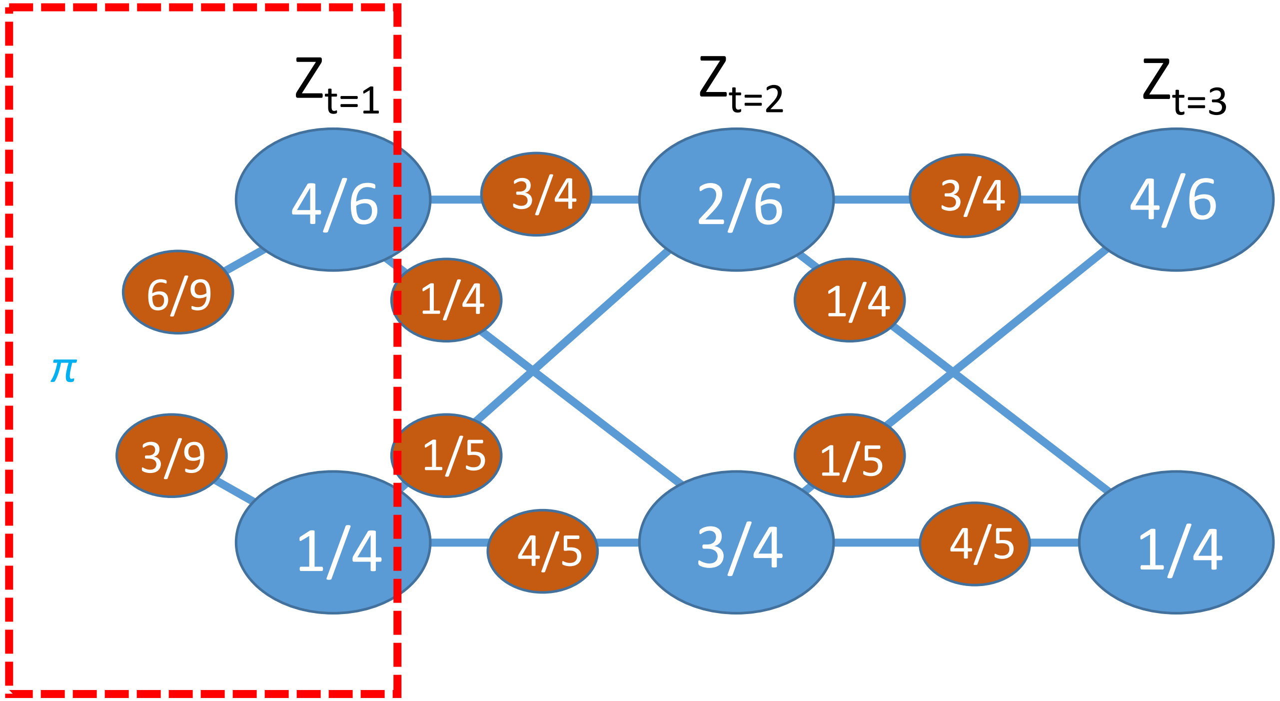
	Happy	Sad
Happy	3/4	1/4
Sad	1/5	4/5

- *b* : emission

Happy		Sad	
Smile	Cry	Smile	Cry
4/6	2/6	1/4	3/4

Cases	**Probability (*π* x *b* x *a* x *b* x *a* x *b*)**	
p(H-H-H| X, π, a, b)	6/9 x 4/6 x 3/4 x 2/6 x 3/4 x 4/6	0.0556
p(H-H-S| X, π, a, b)	6/9 x 4/6 x 3/4 x 2/6 x 1/4 x 1/4	0.0046
p(H-S-H| X, π, a, b)	6/9 x 4/6 x 1/4 x 3/4 x 1/5 x 4/6	0.0111
p(H-S-S| X, π, a, b)	6/9 x 4/6 x 1/4 x 3/4 x 4/5 x 1/4	0.0167
p(S-S-H| X, π, a, b)	3/9 x 1/4 x 4/5 x 3/4 x 1/5 x 4/6	0.0067
p(S-H-S| X, π, a, b)	3/9 x 1/4 x 1/5 x 2/6 x 1/4 x 1/4	0.0003
p(S-H-H| X, π, a, b)	3/9 x 1/4 x 1/5 x2/6 x 3/4 x 4/6	0.0028
p(S-S-S| X, π, a, b)	3/9 x 1/4 x 4/5 x 3/4 x4/5 x 1/4	0.01

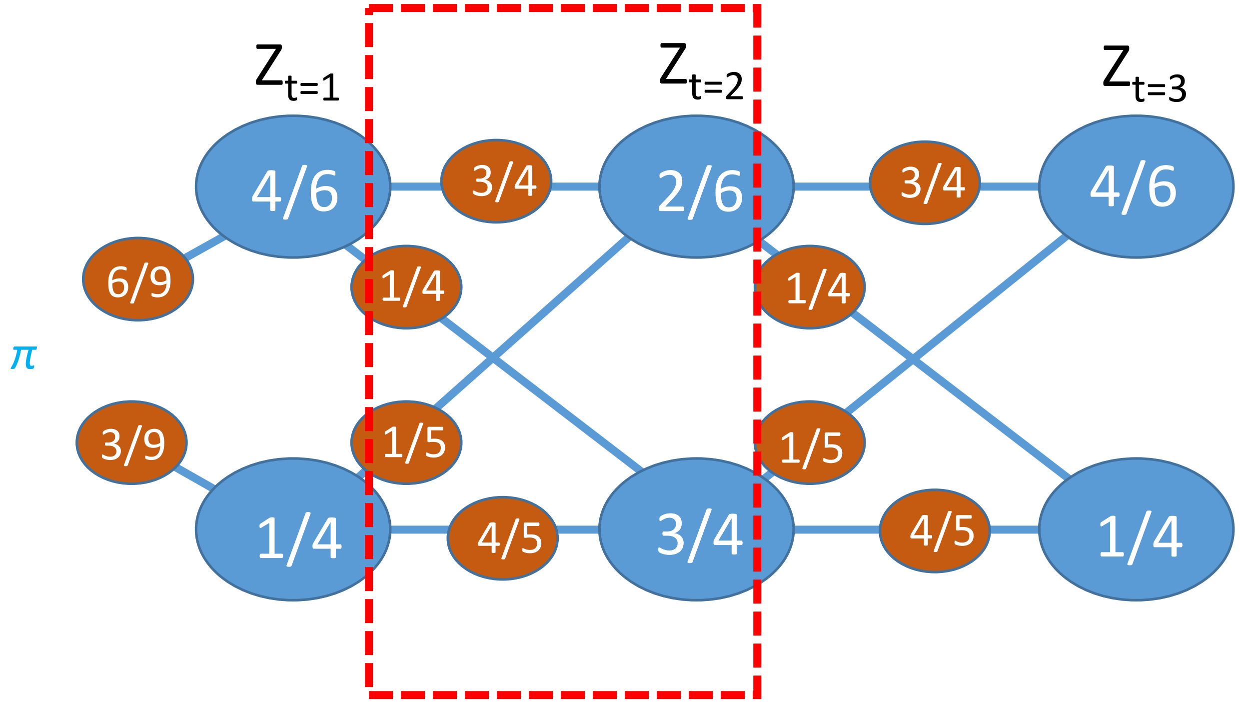
Example: observed data sequence: Smile-Gloomy-Smile



$$V_1(z_1) = b_1 \max_{z_0} a_{0,1} V_0(z_0) = b_1 \pi_1$$

	V(Z1)	V(Z2)	V(Z3)
Happy	6/9 x 4/6 = 0.444		
Sad	3/9 x 1/4 = 0.083		

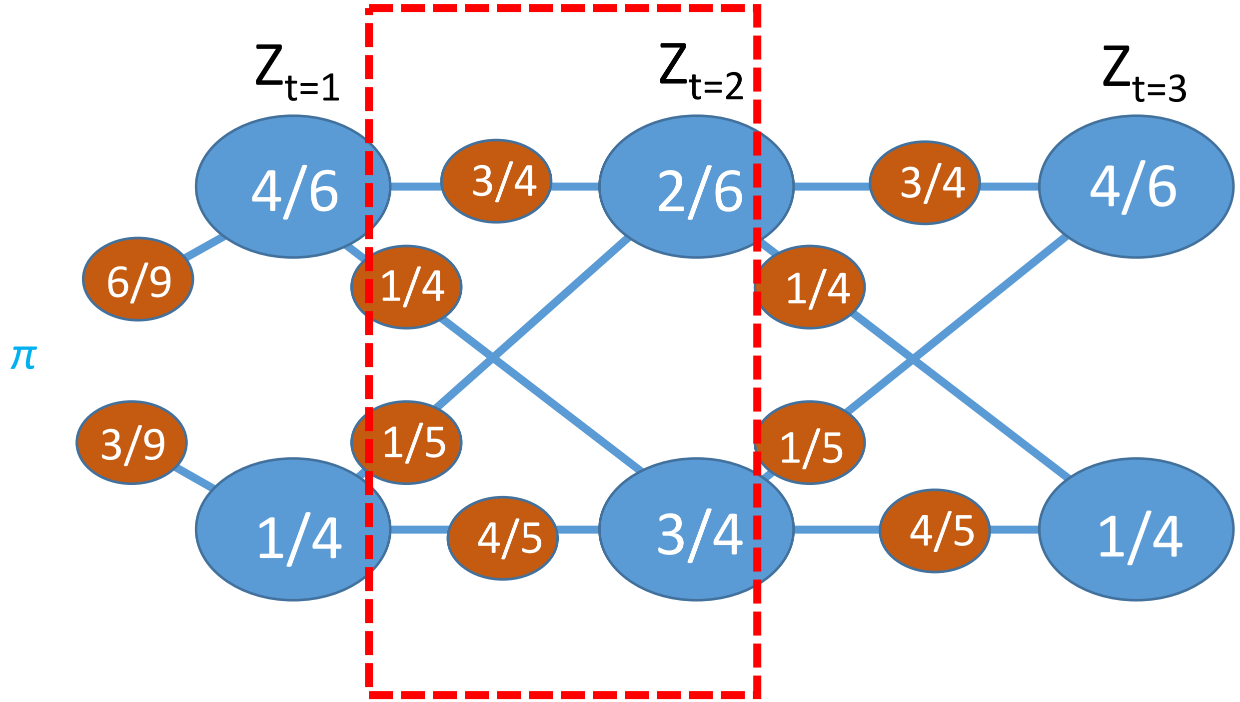
Example: observed data sequence: Smile-Gloomy-Smile



$$V_2(z_2) = b_2 \max_{z_1} a_{1,2} V_1(z_1)$$

	V(Z1)	V(Z2)	V(Z3)
Happy	6/9 x 4/6 = 0.444	3/4 x 2/6 x v(z1-happy) = 0.111 1/5 x 2/6 x v(z1-sad) = 0.00553	
Sad	3/9 x 1/4 = 0.083	1/4 x 3/4 x v(z1-happy) = 0.0833 4/5 x 3/4 x v(z1-sad) = 0.0498	

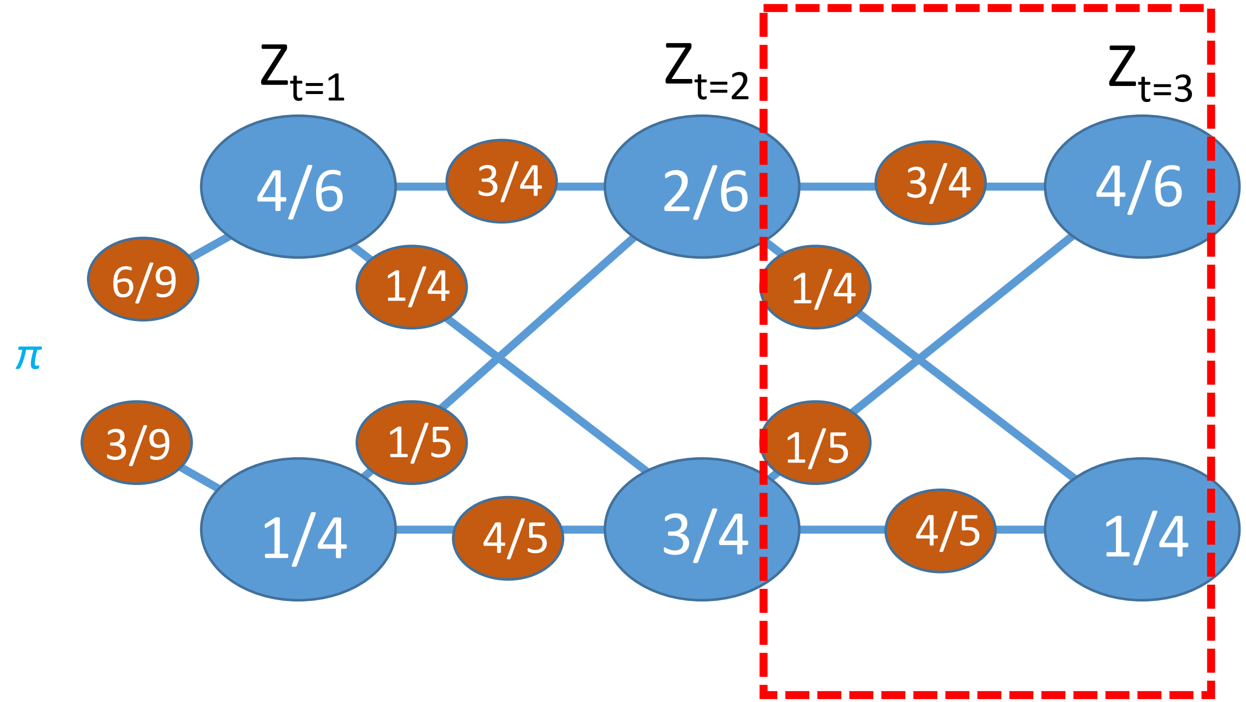
Example: observed data sequence: Smile-Gloomy-Smile



$$V_2(z_2) = b_2 \max_{z_1} a_{1,2} V_1(z_1)$$

	V(Z1)	V(Z2)	V(Z3)
Happy	6/9 x 4/6 = 0.444	3/4 x 2/6 x v(z1-happy) = 0.111 ~~1/5 x 2/6 x v(z1-sad) = 0.00553~~	
Sad	3/9 x 1/4 = 0.083	1/4 x 3/4 x v(z1-happy) = 0.0833 ~~4/5 x 3/4 x v(z1-sad) = 0.0498~~	

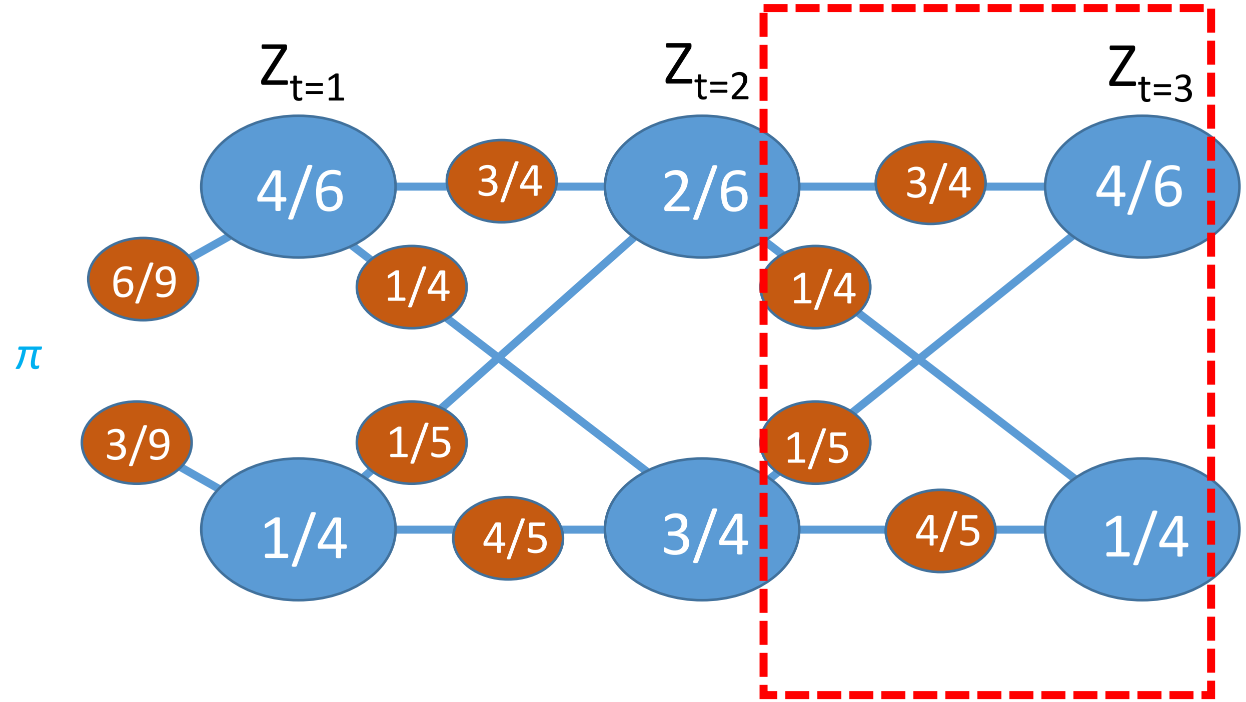
Example: observed data sequence: Smile-Gloomy-Smile



$$V_3(z_3) = b_3 \max_{z_2} a_{2,3} V_2(z_2)$$

	V(Z1)	V(Z2)	V(Z3)
Happy	6/9 x 4/6 = 0.444	3/4 x 2/6 x v(z1-happy) = 0.111 ~~1/5 x 2/6 x v(z1-sad) = 0.00553~~	3/4 x 4/6 x v(z2-happy) = 0.0555 1/5 x 4/6 x v(z2-sad) = 0.0111
Sad	3/9 x 1/4 = 0.083	1/4 x 3/4 x v(z1-happy) = 0.0833 ~~4/5 x 3/4 x v(z1-sad) = 0.0498~~	1/4 x 1/4 x v(z2-happy) = 0.0069 4/5 x 1/4 x v(z2-sad)= 0.01666

Example: observed data sequence: Smile-Gloomy-Smile



$$V_3(z_3) = b_3 \max_{z_2} a_{2,3} V_2(z_2)$$

	V(Z1)	V(Z2)	V(Z3)
Happy	6/9 x 4/6 = 0.444	3/4 x 2/6 x v(z1-happy) = 0.111 ~~1/5 x 2/6 x v(z1-sad) = 0.00553~~	3/4 x 4/6 x v(z2-happy) = 0.0555 ~~1/5 x 4/6 x v(z2-sad) = 0.0111~~
Sad	3/9 x 1/4 = 0.083	1/4 x 3/4 x v(z1-happy) = 0.0833 ~~4/5 x 3/4 x v(z1-sad) = 0.0498~~	~~1/4 x 1/4 x v(z2-happy) = 0.0069~~ 4/5 x 1/4 x v(z2-sad)= 0.01666

HAPPY → *HAPPY* → *HAPPY*