Hidden Markov Model
Principles and Applications

Kwak, Namju

Applied Algorithm Lab.
Computer Science Department
KAIST

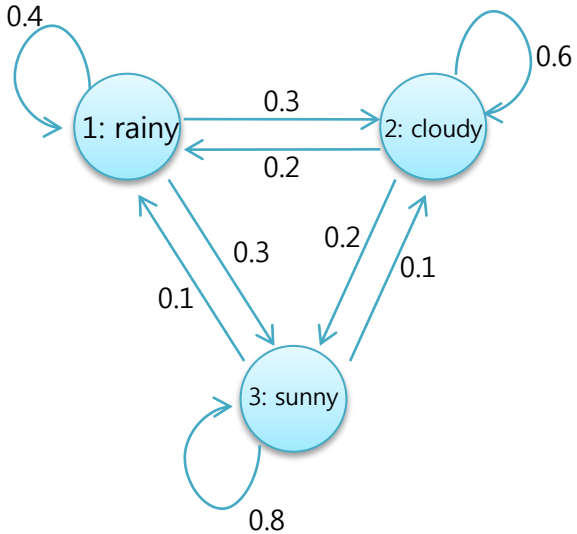
ROSAEC Workshop Tutorial
27th August, 2010

Topics

- Basic Principles of Hidden Markov Model (HMM)
- HMM-related Algorithms
- Application: Voice Recognition
- Application: Anomaly Detection
- Conclusion

Basic Principles of HMM

- Markov Model (MM)



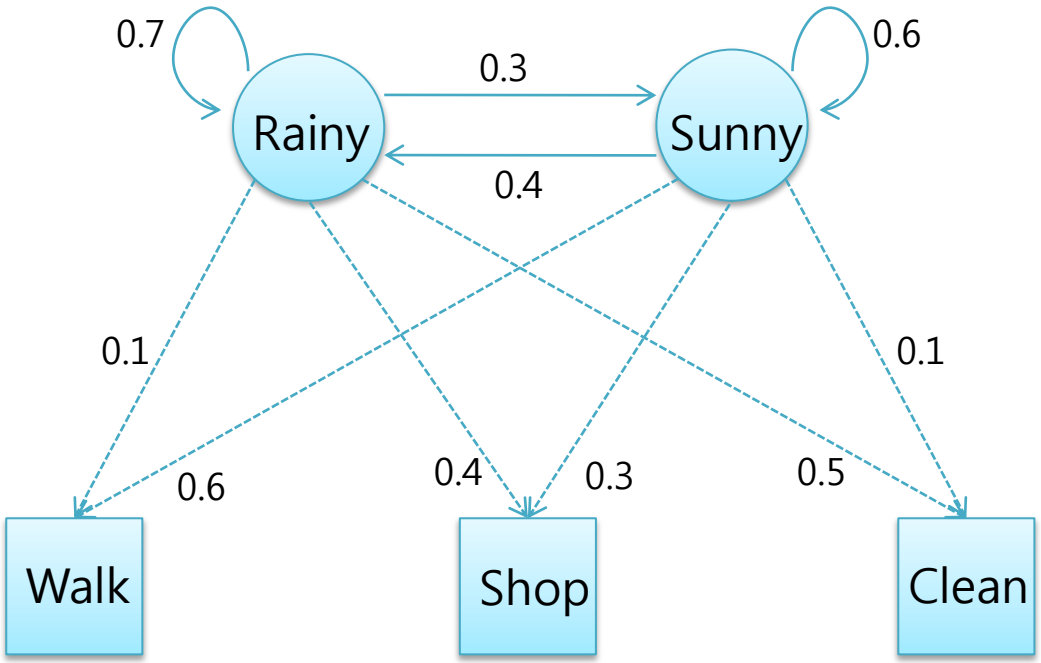
State transition probability matrix

$$(A_{ij}) = \begin{pmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{pmatrix}$$

http://nandaro.tistory.com/entry/Markov-Model

Basic Principles of HMM

- Hidden Markov Model (HMM)



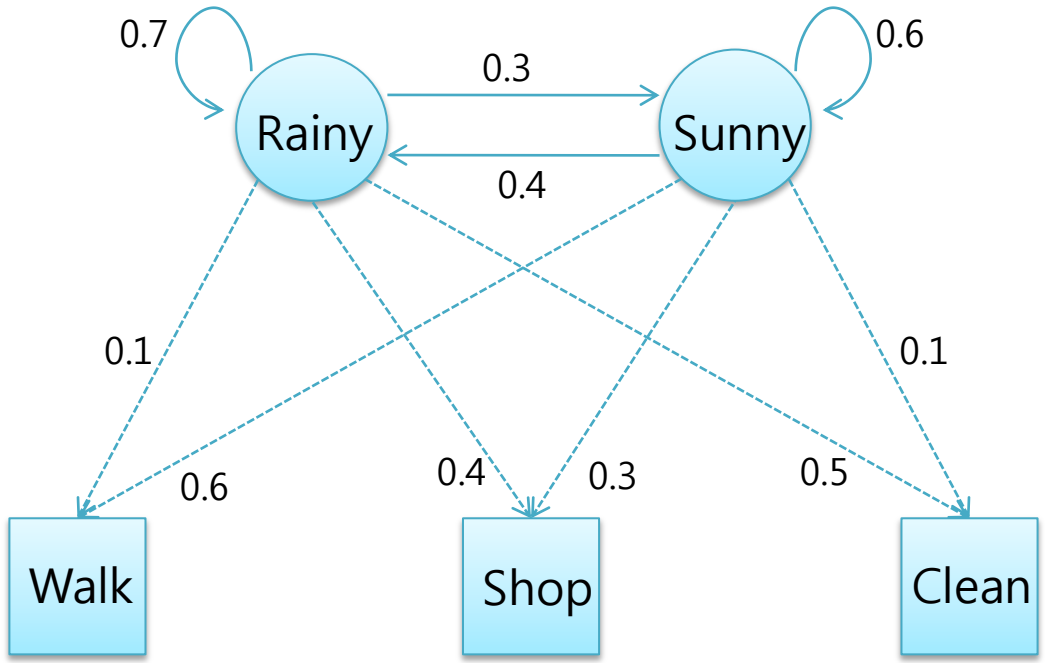
Observation emission probability matrix

$$(b_{ij}) = \begin{pmatrix} 0.1 & 0.4 & 0.5 \\ 0.6 & 0.3 & 0.1 \end{pmatrix}$$

www.wikipedia.org

Basic Principles of HMM

- Hidden Markov Model (HMM)



Bob's girlfriend Alice is on vacation in New York.
Bob knows weather history of New York.
Bob also knows which action Alice is likely to do on rainy and sunny day.
Alice calls Bob and tells him what she did last week.
Can Bob infer the most likely weather sequence of last week?

www.wikipedia.org

Basic Principles of HMM

- Problems of Interest
  - Decoding: For a given observation sequence and a HMM, what is the most likely state sequence (and the likelihood of it)?
  - Learning: For a given observation sequence, how should we set the parameter set of a HMM to maximize the probability that such a sequence occurs?

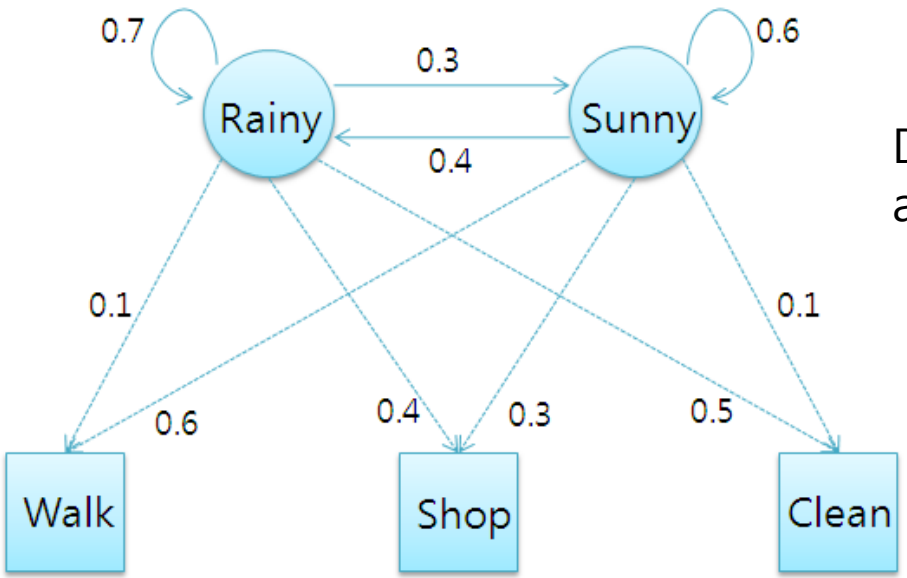
HMM-related Algorithms

- Viterbi Algorithm (Decoding)
  - Decoding: For a given observation sequence and a HMM, what is the most likely state sequence (and the likelihood of it)?

HMM-related Algorithms

- Viterbi Algorithm (Decoding)

$$
\begin{aligned}
V_{0,k} &= \mathrm{P}(x_0 \mid k) \cdot \pi_k \\
V_{t,k} &= \mathrm{P}(x_t \mid k) \cdot \max_{y \in Y} (a_{y,k} V_{t-1,y})
\end{aligned}
$$



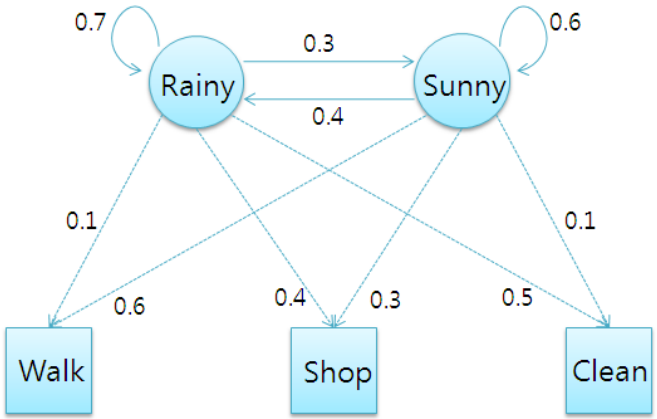
Dynamic programming approach

www.wikipedia.org

HMM-related Algorithms

- Viterbi Algorithm (Decoding)

$$V_{0,k} = P(x_0 \mid k) \cdot \pi_k$$
$$V_{t,k} = P(x_t \mid k) \cdot \max_{y \in Y} (a_{y,k} V_{t-1,y})$$



$$(\pi_i) = \begin{pmatrix} 0.6 \\ 0.4 \end{pmatrix}$$

$$V_{0,Rainy} = P(Walk \mid Rainy) \cdot \pi_{Rainy} = 0.06$$

	Walk	Shop	Clean	Walk	Shop	Clean
Rainy	0.06					
Sunny						

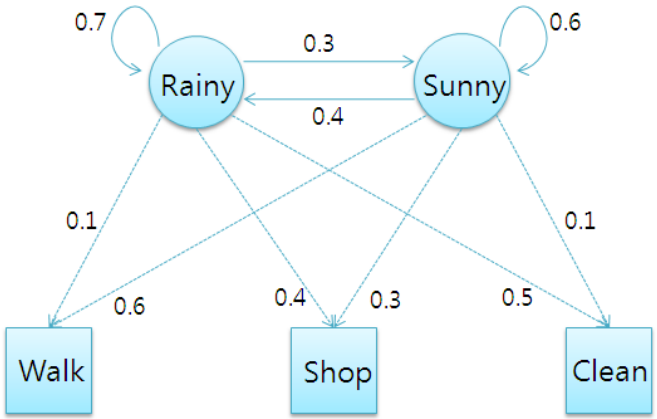
With observation sequence (Walk, Shop, Clean, Walk, Shop, Clean), the state sequence is (Sunny, Rainy, Rainy, Sunny, Sunny, Rainy) with likelihood 0.00014.

www.wikipedia.org

HMM-related Algorithms

- Viterbi Algorithm (Decoding)

$$V_{0,k} = P(x_0 \mid k) \cdot \pi_k$$
$$V_{t,k} = P(x_t \mid k) \cdot \max_{y \in Y} (a_{y,k} V_{t-1,y})$$



$$(\pi_i) = \begin{pmatrix} 0.6 \\ 0.4 \end{pmatrix}$$

$$V_{0,Sunny} = P(Walk \mid Sunny) \cdot \pi_{Sunny} = 0.24$$

	Walk	Shop	Clean	Walk	Shop	Clean
Rainy	0.06					
Sunny	0.24					

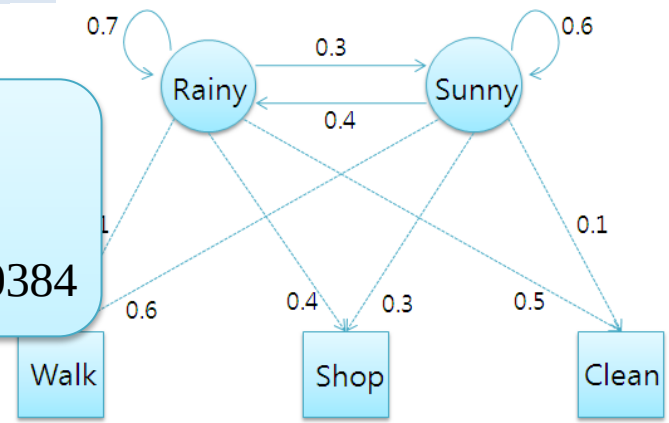
With observation sequence (Walk, Shop, Clean, Walk, Shop, Clean), the state sequence is (Sunny, Rainy, Rainy, Sunny, Sunny, Rainy) with likelihood 0.00014.

www.wikipedia.org

HMM-related Algorithms

- Viterbi Algorithm (Decoding)

$$V_{0,k} = \mathrm{P}(x_0 \mid k) \cdot \pi_k$$
$$V_{t,k} = \mathrm{P}(x_t \mid k) \cdot \max_{y \in Y} (a_{y,k} V_{t-1,y})$$



$$a_{Rainy,Rainy} V_{0,Rainy} = 0.042$$
$$a_{Sunny,Rainy} V_{0,Sunny} = 0.096$$
$$V_{1,Rainy} = P(Shop \mid Rainy) \cdot a_{Sunny,Rainy} V_{0,Sunny} = 0.0384$$

	Walk	Shop	Clean	Walk	Shop	Clean
Rainy	0.06	0.038				
Sunny	0.24					

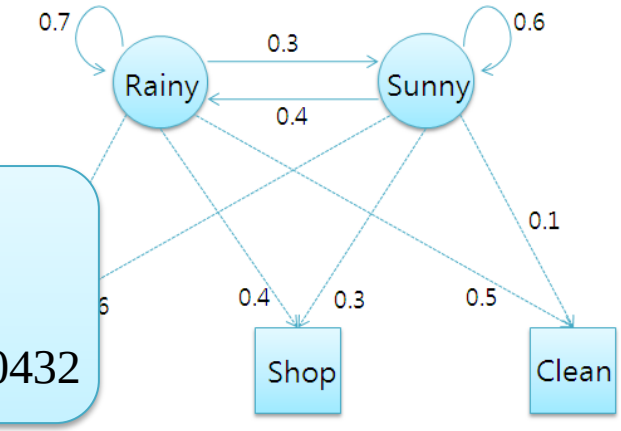
With observation sequence (Walk, Shop, Clean, Walk, Shop, Clean), the state sequence is (Sunny, Rainy, Rainy, Sunny, Sunny, Rainy) with likelihood 0.00014.

www.wikipedia.org

HMM-related Algorithms

- Viterbi Algorithm (Decoding)

$$V_{0,k} = \mathrm{P}(x_0 \mid k) \cdot \pi_k$$
$$V_{t,k} = \mathrm{P}(x_t \mid k) \cdot \max_{y \in Y} (a_{y,k} V_{t-1,y})$$



$$(\pi_i) = \begin{pmatrix} 0.6 \\ 0.4 \end{pmatrix}$$

$$a_{Rainy,Sunny} V_{0,Rainy} = 0.018$$
$$a_{Sunny,Sunny} V_{0,Sunny} = 0.144$$
$$V_{1,Sunny} = P(Shop \mid Sunny) \cdot a_{Sunny,Sunny} V_{0,Sunny} = 0.0432$$

	W	hop	Clean	Walk	Shop	Clean
Rainy	0.0	0.038				
Sunny	0.24	0.043				

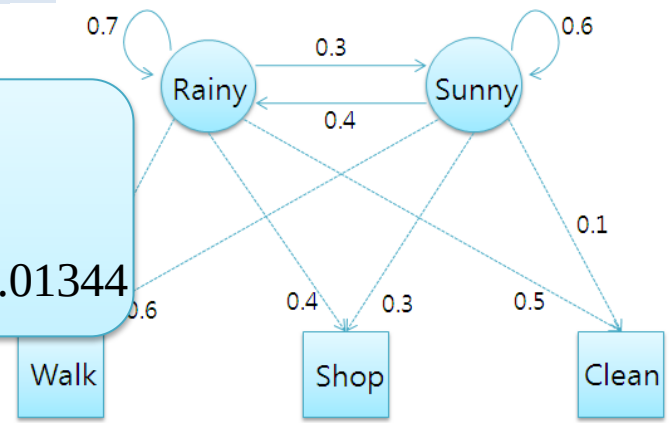
With observation sequence (Walk, Shop, Clean, Walk, Shop, Clean), the state sequence is (Sunny, Rainy, Rainy, Sunny, Sunny, Rainy) with likelihood 0.00014.

www.wikipedia.org

HMM-related Algorithms

- Viterbi Algorithm (Decoding)

$$V_{0,k} = \mathrm{P}(x_0 \mid k) \cdot \pi_k$$
$$V_{t,k} = \mathrm{P}(x_t \mid k) \cdot \max_{y \in Y} (a_{y,k} V_{t-1,y})$$



$$a_{Rainy,Rainy} V_{1,Rainy} = 0.02688$$
$$a_{Sunny,Rainy} V_{1,Sunny} = 0.01728$$
$$V_{2,Rainy} = P(Clean \mid Rainy) \cdot a_{Rainy,Rainy} V_{1,Rainy} = 0.01344$$

	Walk	Shop	Clean	Walk	Shop	Clean
Rainy	0.06	0.038	0.013			
Sunny	0.24	0.043				

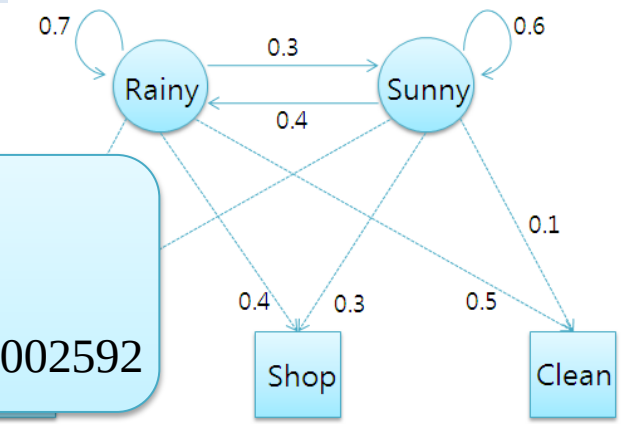
With observation sequence (Walk, Shop, Clean, Walk, Shop, Clean), the state sequence is (Sunny, Rainy, Rainy, Sunny, Sunny, Rainy) with likelihood 0.00014.

www.wikipedia.org

HMM-related Algorithms

- Viterbi Algorithm (Decoding)

$$V_{0,k} = \mathrm{P}(x_0 \mid k) \cdot \pi_k$$
$$V_{t,k} = \mathrm{P}(x_t \mid k) \cdot \max_{y \in Y} (a_{y,k} V_{t-1,y})$$



$$(\pi_i) = \begin{pmatrix} 0.6 \\ 0.4 \end{pmatrix}$$

$$a_{Rainy,Sunny} V_{1,Rainy} = 0.01152$$
$$a_{Sunny,Sunny} V_{1,Sunny} = 0.02592$$
$$V_{2,Sunny} = P(Clean \mid Sunny) \cdot a_{Sunny,Sunny} V_{1,Sunny} = 0.002592$$

	Walk	Shop	Cl	Walk	Shop	Clean
Rainy	0.06	0.038	13			
Sunny	0.24	0.043	0.003			

With observation sequence (Walk, Shop, Clean, Walk, Shop, Clean), the state sequence is (Sunny, Rainy, Rainy, Sunny, Sunny, Rainy) with likelihood 0.00014.

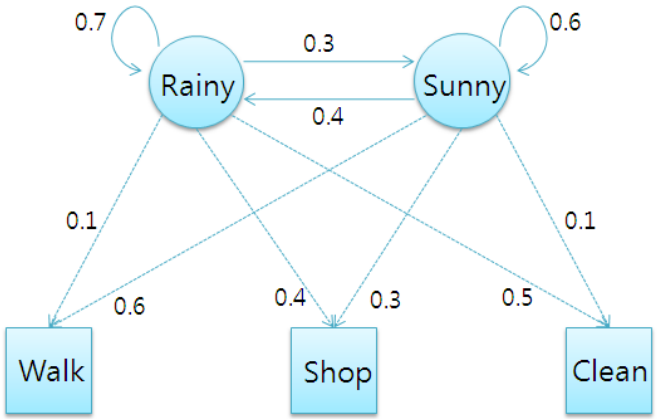
www.wikipedia.org

HMM-related Algorithms

- Viterbi Algorithm (Decoding)

$$V_{0,k} = \mathrm{P}(x_0 \mid k) \cdot \pi_k$$
$$V_{t,k} = \mathrm{P}(x_t \mid k) \cdot \max_{y \in Y} (a_{y,k} V_{t-1,y})$$

$$(\pi_i) = \begin{pmatrix} 0.6 \\ 0.4 \end{pmatrix}$$



	Walk	Shop	Clean	Walk	Shop	Clean
Rainy	0.06	0.038	0.013	0.0009	0.00039	0.00014
Sunny	0.24	0.043	0.003	0.0024	0.00044	0.00003

With observation sequence (Walk, Shop, Clean, Walk, Shop, Clean), the state sequence is (Sunny, Rainy, Rainy, Sunny, Sunny, Rainy) with likelihood 0.00014.

www.wikipedia.org

HMM-related Algorithms

- Baum-Welch (Learning)
  - For a given observation sequence, how should we set the parameter set of a HMM to maximize the probability that such a sequence occurs?
  - Parameters: initial state probability vector, state transition probability matrix, and observation emission probability matrix
  - EM algorithm (Expectation Maximization)

HMM-related Algorithms

- Baum-Welch (Learning)

$$\alpha_1(i) = \pi_i b_i(O_1)$$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1})$$

$$\beta_T(i) = 1$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}$$

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_k \sum_l \alpha_t(k) a_{kl} b_l(O_{t+1}) \beta_{t+1}(l)}$$

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

$$\hat{b}_j(m) = \frac{\sum_{t=1}^{T} \gamma_t(j) 1(O_t = v_m)}{\sum_{t=1}^{T} \gamma_t(j)}$$

$$\hat{\pi}_i = \gamma_1(i)$$

Until convergence

Introduction to Machine Learning, 2nd ed., Alpaydin, MIT Press

HMM-related Algorithms

- Baum-Welch (Learning)

  - Definition

$$\alpha_t(i) = P(O_1 \cdots O_t, q_t = S_i \mid \lambda)$$

  - Initialization

$$\begin{aligned}\alpha_1(i) &= P(O_1, q_1 = S_i \mid \lambda) \\ &= P(O_1 \mid q_1 = S_i, \lambda) P(q_1 = S_i) \\ &= \pi_i b_i(O_1)\end{aligned}$$

  - Recursion

$$\begin{aligned}\alpha_{t+1}(j) &= P(O_1 \cdots O_{t+1}, q_{t+1} = S_j \mid \lambda) \\ &= \cdots \\ &= \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1})\end{aligned}$$

$\alpha_t(i)$

Obs.	O_1	O_2	…	O_t	O_{t+1}	…	O_T
State	…	…	…	S_i	…	…	…

Introduction to Machine Learning, 2nd ed., Alpaydin, MIT Press

HMM-related Algorithms

- Baum-Welch (Learning)

  - Definition

$$\beta_t(i) = P(O_{t+1} \cdots O_T \mid q_t = S_i, \lambda)$$

  - Initialization (arbitrarily to 1)

$$\beta_T(i) = 1$$

  - Recursion

$$\begin{aligned}\beta_t(i) &= P(O_{t+1} \cdots O_T \mid q_t = S_i, \lambda) \\ &= \cdots \\ &= \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)\end{aligned}$$

$\beta_t(i)$

Obs.	O_1	O_2	…	O_t	O_{t+1}	…	O_T
State	…	…	…	S_i	…	…	…

Introduction to Machine Learning, 2nd ed., Alpaydin, MIT Press

HMM-related Algorithms

- Baum-Welch (Learning)

$$\gamma_t(i) = P(q_t = S_i \mid O, \lambda) = \cdots = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}$$

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda) = \cdots = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_k \sum_l \alpha_t(k) a_{kl} b_l(O_{t+1}) \beta_{t+1}(l)}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \qquad \hat{b}_j(m) = \frac{\sum_{t=1}^{T} \gamma_t(j) 1(O_t = v_m)}{\sum_{t=1}^{T} \gamma_t(j)} \qquad \hat{\pi}_i = \gamma_1(i)$$

Introduction to Machine Learning, 2nd ed., Alpaydin, MIT Press

HMM-related Algorithms

- Baum-Welch (Learning)

$$\alpha_1(i) = \pi_i \hat{b}_i(O_1)$$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\hat{a}_{ij}\right]\hat{b}_j(O_{t+1})$$

$$\beta_T(i) = 1$$

$$\beta_t(i) = \sum_{j=1}^{N} \hat{a}_{ij}\hat{b}_j(O_{t+1})\beta_{t+1}(j)$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_k \sum_l \alpha_t(k)a_{kl}b_l(O_{t+1})\beta_{t+1}(l)}$$

E step

M step

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

$$\hat{b}_j(m) = \frac{\sum_{t=1}^{T} \gamma_t(j)1(O_t = v_m)}{\sum_{t=1}^{T} \gamma_t(j)}$$

$$\hat{\pi}_i = \gamma_1(i)$$

Until convergence

"These two steps are alternated until convergence during which, it has been shown, P(O|λ) never decreases."

Introduction to Machine Learning, 2nd ed., Alpaydin, MIT Press

HMM-related Algorithms

- Baum-Welch (Learning)
  - Limitation
    - Not optimal
    - Local maximum (EM approach)
    - Dependence on parameter initialization

Application: Voice Recognition

Spoken sound "catz"

$$\begin{pmatrix} a_1 & a_2 & a_3 & \cdots & a_t \\ b_1 & b_2 & b_3 & \cdots & b_t \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_1 & z_2 & z_3 & \cdots & z_t \end{pmatrix}$$

Features from Fourier transform

Pitch, volume, etc.

HMM for Sound 1 "cup"

HMM for Sound 2 "cat"

HMM for Sound 3 "carrot"

...

HMM for Sound n "castle"

Baum-Welch Learning

Viterbi Likelihood

Viterbi Likelihood

Viterbi Likelihood

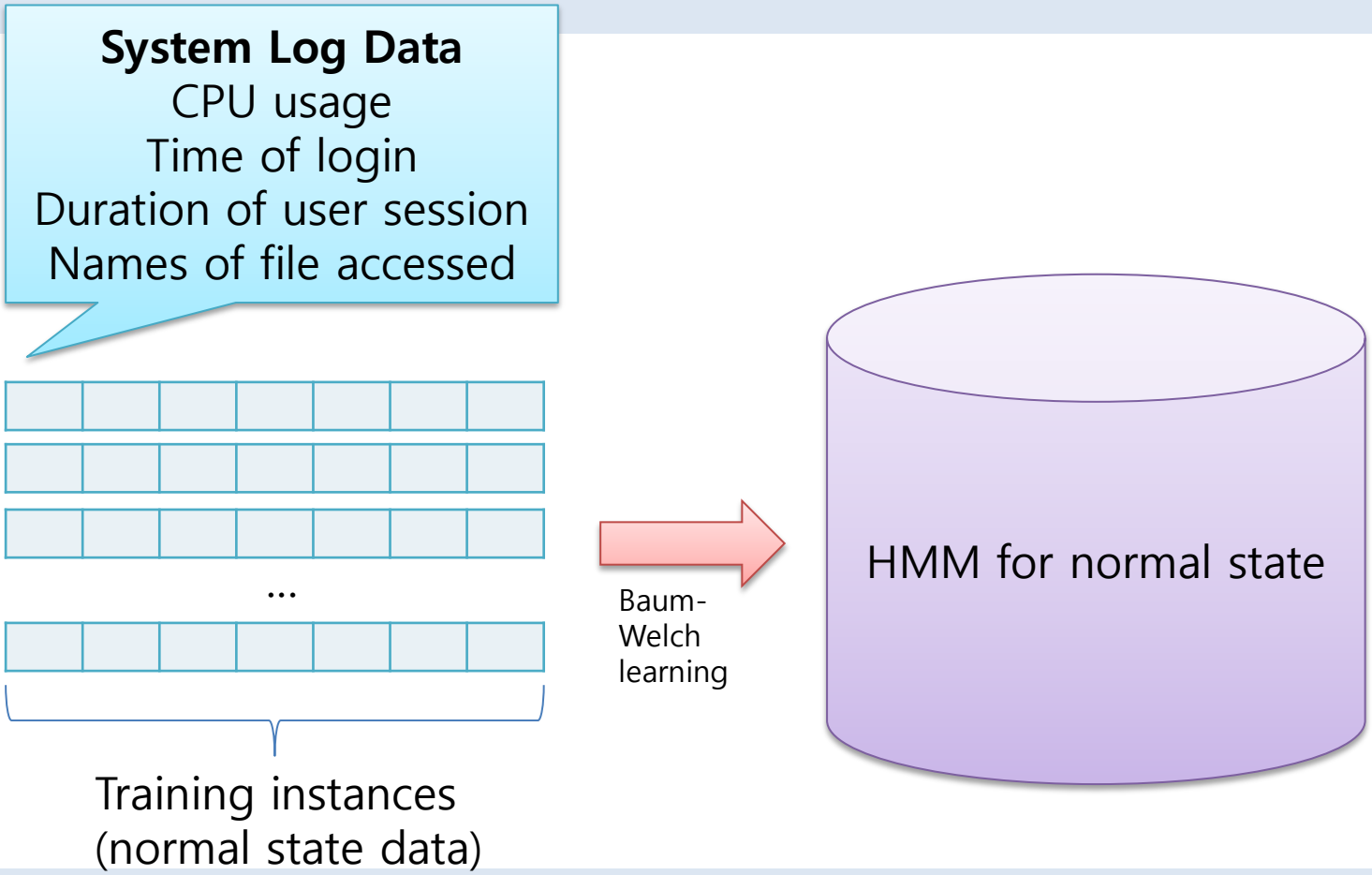
Viterbi Likelihood

Viterbi Decoding

The highest

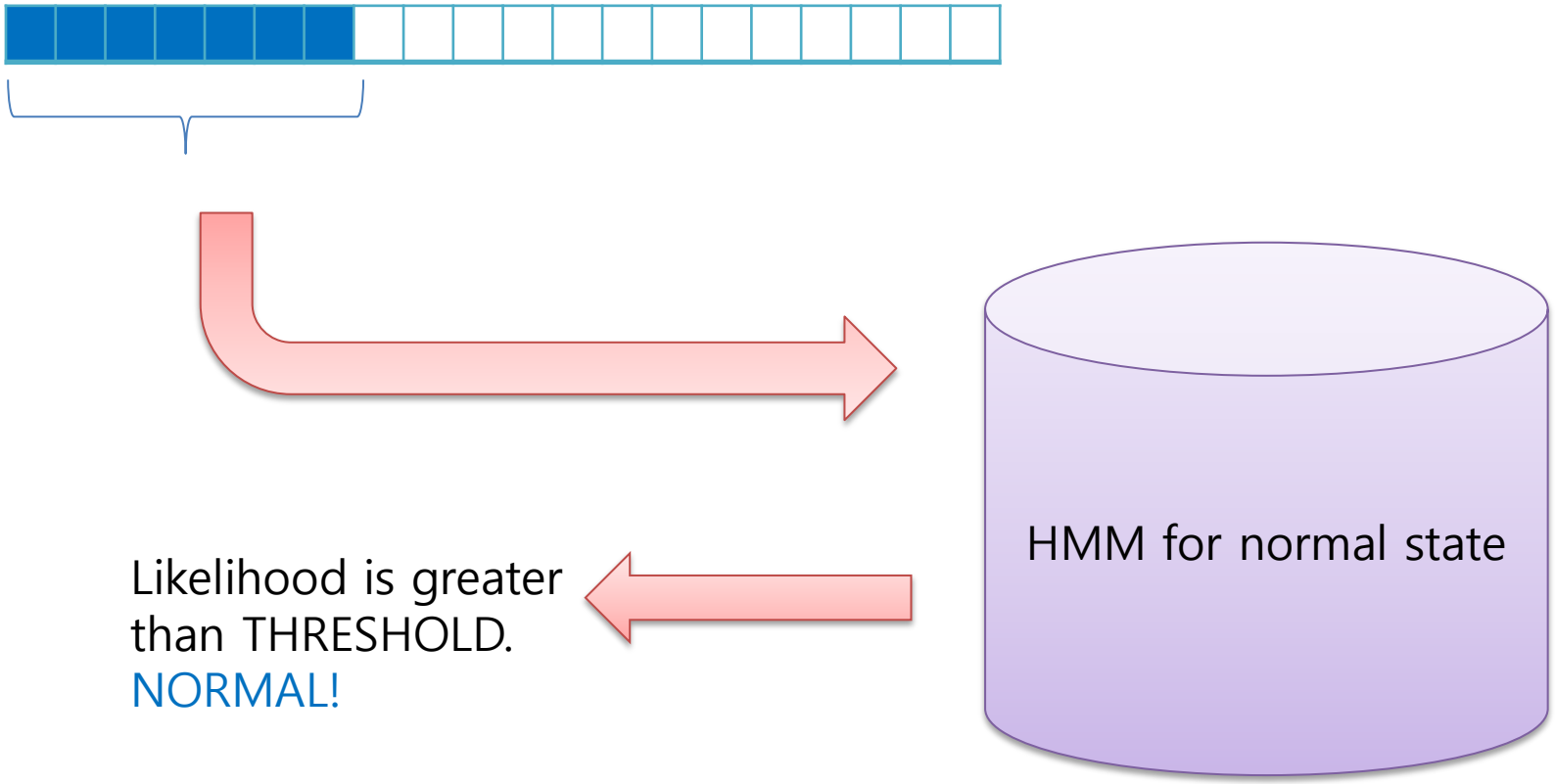
The spoken sound is sound 2 "cat"!!

Application: Anomaly Detection



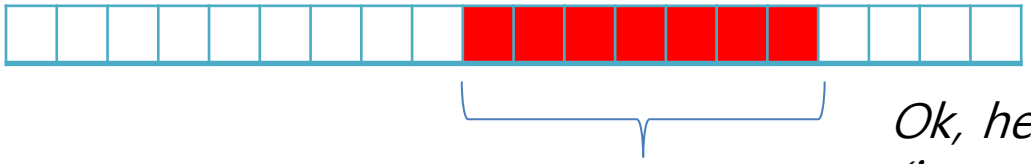
Application: Anomaly Detection

Target system log data

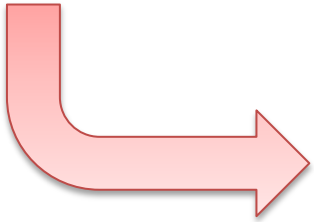


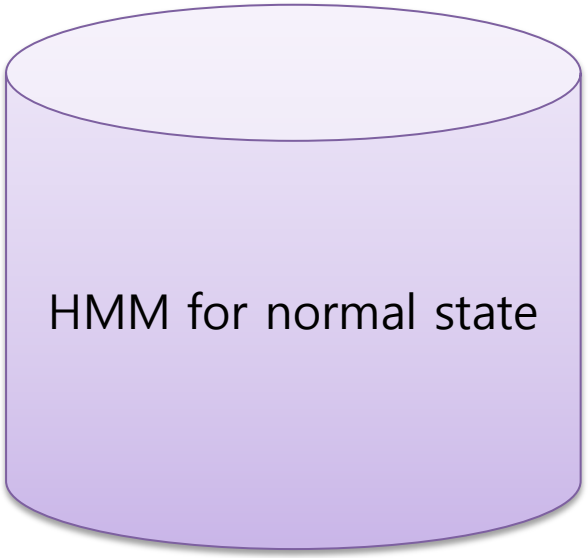
Application: Anomaly Detection

Target system log data



Ok, here is the abnormal point (i.e. crash, deadlock, etc).





Likelihood is less than THRESHOLD.
ABNORMAL!

Conclusion

- Principles
  - Hidden Markov model
  - 3 problems and corresponding algorithms

- Where to use HMMs?
  - Analysis of sequential data
  - Voice recognition, anomaly/intrusion detection, gene analysis/prediction, cryptoanalysis, written character recognition and general time series forecast
  - Somewhere the sequential pattern recognition technique can be useful in PL/SE world (hopefully)

Questions and Answers