2018 Intelligent Systems (**知的システム構成論**)

Inference and Learning of Hidden Markov Model

May.24, 2018

Takehisa YAIRI（**矢入健久**）
E-mail: yairi@ailab.t.u-tokyo.ac.jp

Notice

- The next class will be on May 30 (Wed).
- Hopefully, we will go on to Switching Linear Dynamical Systems, which can be regarded as a hybrid of LDS and HMM

Outline

1. Markov Chain
   - Maximum likelihood estimation of Markov chain
2. Hidden Markov Model (HMM)
3. Inference with HMM
   - Filtering and smoothing
   - Viterbi algorithm -> may be skipped
4. Example : Robot Position Estimation
5. Learning of HMM
   - EM algorithm for HMM (Baum-Welch algorithm)

Markov Process (Markov Chain)

- x_t takes a value in a set of discrete values

$$x_t \in \{1, 2, \ldots, K\}$$

for simplicity, consider integer numbers from 1 to K

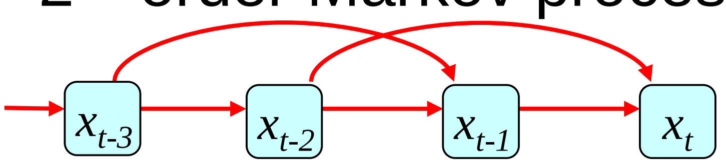
- Prob. dist. of x_t is dependent only on states at previous n-steps (n-th order Markov process)

$$P(x_t \mid x_{t-1}, x_{t-2}, \ldots, x_1, x_0) = P(x_t \mid x_{t-1}, \ldots, x_{t-n})$$

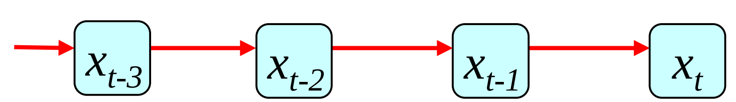
  – Special case of n=1 is called as *Simple Markov Process*

$$P(x_t \mid x_{t-1}, x_{t-2}, \ldots, x_1, x_0) = P(x_t \mid x_{t-1})$$

2nd order Markov process



Simple Markov process



Examples of Markov Process

- Weather
  - $x_t \in$ {Sunny, Rain, Cloudy}
- *Sugoroku*
  - $x_t \in$ {Places that pieces can occupy}
- Musical note
- Web browsing history
- Psychological states of persons
- Status of machines
  - $x_t \in$ {Normal, Abnormal}

Model Parameters of Markov Chain

- We focus on *simple* Markov process
- Model parameters:

- Initial probabilities $\pi_i \equiv p(x_1 = i)$ (i=1,2,..,K)

$$\boldsymbol{\pi} \equiv [\pi_1, \pi_2, \cdots \pi_K]^T$$ K-dimensional vector

$$\pi_1 + \pi_2 + \cdots + \pi_K = \sum_{i=1}^{K} \pi_i = \boldsymbol{\pi}^T \mathbf{1}_K = 1$$

where $\mathbf{1}_K \equiv [\underbrace{1,1,\cdots,1}_{K}]^T$

- Transition probabilities $A_{i,j} \equiv p(x_{t+1} = j \mid x_t = i)$

$$\mathbf{A} \equiv \lfloor A_{i,j} \rfloor$$ K x K matrix

$$\sum_{j=1}^{K} A_{i,j} = 1$$ (i=1,2,..,K) $\Leftrightarrow$ $\mathbf{A}\mathbf{1}_K = \mathbf{1}_K$

Likelihood Function of Markov Model (1)

- Model parameters: $\Theta = \{\mathbf{A}, \boldsymbol{\pi}\}$
- Data: $D = \mathbf{x}_{1:T} = [x_1, x_2, \cdots, x_2]^T$
- Log-likelihood function:

$$l(\Theta \mid D) \equiv \ln p(\mathbf{x}_{1:T} \mid \Theta) = \ln\left\{p(x_1)\prod_{t=1}^{T-1} p(x_{t+1} \mid x_t)\right\}$$

Note we can write

$$p(x_1) = \pi_{x_1} = \prod_{i=1}^{K} \pi_i^{\mathrm{I}(x_1 = i)}$$

$$p(x_{t+1} \mid x_t) = A_{x_t, x_{t+1}} = \prod_{i=1}^{K}\prod_{j=1}^{K} A_{i,j}^{\mathrm{I}(x_t = i, x_{t+1} = j)}$$

where $\mathrm{I}(\mathrm{x})$ is the indicator function defined as,

$$\mathrm{I}(x) = \begin{cases} 1 & \text{(if x is true)} \\ 0 & \text{(otherwise)} \end{cases}$$

Likelihood Function of Markov Model (2)

By substituting them,

$$l(\Theta \mid D) = \ln\left\{\prod_{i=1}^{K} \pi_i^{\mathbf{I}(x_1=i)} \prod_{t=1}^{T-1} \prod_{i=1}^{K} \prod_{j=1}^{K} A_{i,j}^{\mathbf{I}(x_t=i,x_{t+1}=j)}\right\}$$

$$= \sum_{i=1}^{K} \mathbf{I}(x_1 = i) \ln \pi_i + \sum_{i=1}^{K} \sum_{j=1}^{K} N_{i,j} \ln A_{i,j}$$

where $N_{i,j} \equiv \sum_{i=1}^{K} \sum_{j=1}^{K} \mathbf{I}(x_t = i, x_{t+1} = j)$

Counts of transitions from i to j in $\boldsymbol{x}_{1:T}$

Maximize $l(\Theta|D)$ with constraints :

$$\sum_{i=1}^{K} \pi_i = 1$$ and $$\sum_{j=1}^{K} A_{i,j} = 1$$ (i=1,2,..,K)

Lagrange multipliers !

Maximum Likelihood Estimation of Markov Model (1)

Define the Lagrangian as,

$$L(\Theta,\lambda,\boldsymbol{\mu}) = l(\Theta \mid D) + \lambda\left(1-\sum_{i=1}^{K}\pi_i\right) + \sum_{i=1}^{K}\mu_i\left(1-\sum_{j=1}^{K}A_{i,j}\right)$$

$$= \sum_{i=1}^{K}\mathbf{I}(x_1=i)\ln\pi_i + \sum_{i=1}^{K}\sum_{j=1}^{K}N_{i,j}\ln A_{i,j}$$

$$+\lambda\left(1-\sum_{i=1}^{K}\pi_i\right) + \sum_{i=1}^{K}\mu_i\left(1-\sum_{j=1}^{K}A_{i,j}\right)$$

$$\frac{\partial L}{\partial \pi_i} = \frac{\mathbf{I}(x_1=i)}{\pi_i} - \lambda = 0 \quad \Rightarrow \quad \pi_i = \frac{\mathbf{I}(x_1=i)}{\lambda}$$

constraint

$$\sum_{i=1}^{K}\pi_i = \frac{1}{\lambda}\sum_{i=1}^{K}\mathbf{I}(x_1=i) = \frac{1}{\lambda} = 1 \quad \Rightarrow \quad \hat{\pi}_i = \mathbf{I}(x_1=i)$$

Maximum Likelihood Estimation of Markov Model (2)

Derivatives of Lagrangian w.r.t. $A_{i,j}$:

$$\frac{\partial L}{\partial A_{i,j}} = \frac{N_{i,j}}{A_{i,j}} - \mu_i = 0 \quad \Rightarrow \quad A_{i,j} = \frac{N_{i,j}}{\mu_i}$$

Substitute this into the constraint:

$$\sum_{j=1}^{K} A_{i,j} = \frac{1}{\mu_i} \sum_{j=1}^{K} N_{i,j} = 1 \quad \Rightarrow \quad \mu_i = \sum_{j=1}^{K} N_{i,j}$$

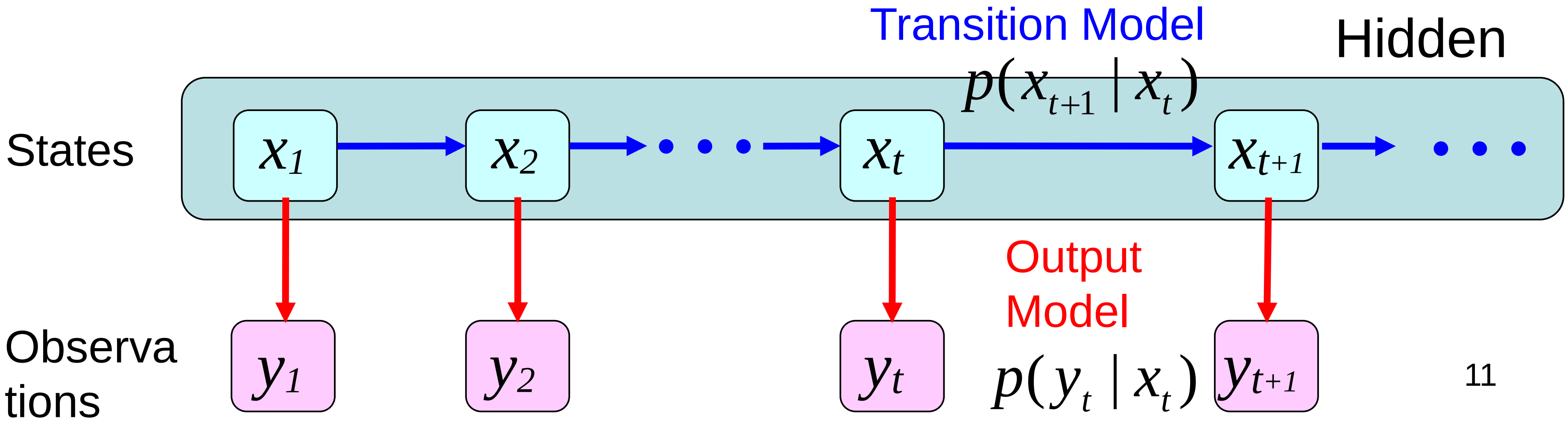
As a result,

$$\hat{A}_{i,j} = \frac{N_{i,j}}{\sum_{j=1}^{K} N_{i,j}}$$

Just frequencies ..

Hidden Markov Model

- Transition of internal state x_t of the system is modeled by Markov process
- Unable to access to (observe) x_t directly
- Observation y_t is available instead
  - Prob. dist. of y_t is determined by x_t $P(y_t \mid x_t)$
  - y_t can be a discrete or continuous scalar or vector



Elements of HMM (with Discrete Observations)

- Hidden state : $x_t \in \{1, 2, \ldots, K\}$
- Observation (output) : $y_t \in \{1, 2, \ldots, M\}$

Here we deal with discrete scalar output

- Transition model : $\mathbf{A} \equiv \left[A_{i,j} \right]$

$$A_{i,j} \equiv p\left(x_{t+1} = j \mid x_t = i\right)$$

Probability of moving from state *i* to state *j*

- Output (Observation) model : $\mathbf{B} \equiv \left[B_{i,k} \right]$

$$B_{i,k} \equiv p(y_t = k \mid x_t = i)$$

Probability of observing *k* at state *i*

- Initial probability distribution : $\boldsymbol{\pi} \equiv \left[\pi_1, \pi_2, \cdots \pi_K \right]^T$

$$\pi_i \equiv p\left(x_1 = i\right)$$

State Estimation Using HMM Filtering (1)

- Filtering : Estimation of posterior distribution of current state, given all observations to date
- i.e., Compute $P(x_t \mid y_{1:t})$

[Hints for derivation]

- Consider deriving a recursive form for computing $\mathrm{P}(x_{t+1}|y_{1:t+1})$ from $\mathrm{P}(x_t \mid y_{1:t})$
- Transition & output models will be contained in the result
- Take advantage of Markov property (conditional independence)
- Use Bayes' rule and marginalization techniques

State Estimation Using HMM Filtering (2)

Posterior at time t+1

$$P(x_{t+1} \mid y_{1:t+1}) = P(x_{t+1} \mid y_{t+1}, y_{1:t})$$ ⇦ $y_{1:t+1} = \{y_{t+1}, y_{1:t}\}$

$$= \frac{P(y_{t+1} \mid x_{t+1}, y_{1:t}) \cdot P(x_{t+1} \mid y_{1:t})}{P(y_{t+1} \mid y_{1:t})}$$ ⇦ Bayes' rule

Independent of x_{t+1}

$$= c \cdot P(y_{t+1} \mid x_{t+1}, y_{1:t}) \cdot P(x_{t+1} \mid y_{1:t})$$ ⇦

$$= c \cdot P(y_{t+1} \mid x_{t+1}) \cdot P(x_{t+1} \mid y_{1:t})$$ ⇦ Markov Property

$$= c \cdot P(y_{t+1} \mid x_{t+1}) \cdot \sum_{i=1}^{K} P(x_{t+1}, x_t = i \mid y_{1:t})$$ ⇦ Marginalization

$$= c \cdot P(y_{t+1} \mid x_{t+1}) \cdot \sum_{i=1}^{K} \{P(x_{t+1} \mid x_t = i, y_{1:t}) \cdot P(x_t = i \mid y_{1:t})\}$$ ⇦ Bayes'

$$= c \cdot \underbrace{P(y_{t+1} \mid x_{t+1})}_{\text{Observation Model}} \cdot \sum_{i=1}^{K} \{\underbrace{P(x_{t+1} \mid x_t = i)}_{\text{Transition Model}} \cdot \underbrace{P(x_t = i \mid y_{1:t})}_{\text{Posterior at time t}}\}$$ ⇦ Markov

State Estimation Using HMM Filtering (3)

$$P(x_{t+1} \mid y_{1:t+1}) = c \cdot P(y_{t+1} \mid x_{t+1}) \cdot \sum_{i=1}^{K} \{P(x_{t+1} \mid x_t = i) \cdot \underbrace{P(x_t = i \mid y_{1:t})}\}$$

Define the forward prob. $\alpha_t(i) \equiv P(x_t = i \mid y_{1:t})$

Assume the observation at t+1 $y_{t+1} \in \{1, 2, \cdots, K\}$

Then, $\alpha_{t+1}(j) = c \cdot B_{j,y_t} \sum_{i=1}^{K} A_{i,j} \alpha_t(i)$

Filtered dist.

k-th column of B

By defining $\boldsymbol{\alpha}_t \equiv [\alpha_t(1), \cdots, \alpha_t(K)]^T$ $\mathbf{B}_{\bullet,k} \equiv [B_{1,k}, \cdots B_{K,k}]^T$

$$\boxed{\boldsymbol{\alpha}_{t+1} \propto \mathbf{B}_{\bullet,y_t} \circ \left(\mathbf{A}^T \boldsymbol{\alpha}_t\right) \quad \text{with } \mathbf{1}_K^T \boldsymbol{\alpha}_{t+1} = \sum_{j=1}^{K} \alpha_{t+1}(j) = 1}$$

Element-wise (Hadamard) product

State Estimation Using HMM Smoothing (1)

- Smoothing : Estimation of posterior of a past state, given all observations up to the present
- i.e., Compute $P(x_t \mid y_{1:T})$ (where $t < T$)

Ref. Derivation of Bayesian Smoothing"

$$P(x_t \mid y_{1:T}) = P(x_t \mid y_{1:t}, y_{t+1:T})$$

$$= c \cdot P(x_t \mid y_{1:t}) \cdot P(y_{t+1:T} \mid x_t, y_{1:t})$$

$$= c \cdot \boxed{P(x_t \mid y_{1:t})} \cdot \boxed{P(y_{t+1:T} \mid x_t)}$$

Filtering distribution (Forward probability)

called "Backward probability"

State Estimation Using HMM Smoothing (2)

$$P(y_{t+1:T} \mid x_t) = \sum_{j=1}^{K} P(y_{t+1:T}, x_{t+1} = j \mid x_t)$$

$$= \sum_{j=1}^{K} P(y_{t+1:T} \mid x_{t+1} = j, x_t) \cdot P(x_{t+1} = j \mid x_t)$$

$$= \sum_{j=1}^{K} P(y_{t+1:T} \mid x_{t+1} = j) \cdot P(x_{t+1} = j \mid x_t)$$

$$= \sum_{j=1}^{K} P(y_{t+1}, y_{t+2:T} \mid x_{t+1} = j) \cdot P(x_{t+1} = j \mid x_t)$$

$$= \sum_{j=1}^{K} \underbrace{P(y_{t+1} \mid x_{t+1} = j)}_{\text{Observation Model}} \cdot P(y_{t+2:T} \mid x_{t+1} = j) \cdot \underbrace{P(x_{t+1} = j \mid x_t)}_{\text{Transition Model}}$$

Backward recursive form !

State Estimation Using HMM Smoothing (3)

$$P(y_{t+1:T} \mid x_t) = \sum_{j=1}^{K} P(y_{t+1} \mid x_{t+1} = j) \cdot P(y_{t+2:T} \mid x_{t+1} = j) \cdot P(x_{t+1} = j \mid x_t)$$

define

$$\beta_t(i) \equiv P(y_{t+1:T} \mid x_t = i) \qquad B_{j,y_{t+1}} \qquad \beta_{t+1}(j) \qquad A_{1:K,j}$$

j-th column

$$\beta_t(i) = \sum_{j=1}^{K} B_{j,y_{t+1}} \beta_{t+1}(j) A_{i,j}$$

k-th column of B

By defining $\boldsymbol{\beta}_t \equiv [\beta_t(1), \cdots, \beta_t(K)]^T$ $\qquad \mathbf{B}_{\bullet,k} \equiv [B_{1,k}, \cdots B_{K,k}]^T$

$$\boxed{\boldsymbol{\beta}_t = \mathbf{B}_{\bullet,y_{t+1}} \circ (\boldsymbol{A}\boldsymbol{\beta}_{t+1}) \quad \text{with} \quad \boldsymbol{\beta}_T = \mathbf{1}_K}$$

Backward equation

Element-wise (Hadamard) product

State Estimation Using HMM Smoothing (4)

Finally, smoothed distribution is obtained by

$$\gamma_t(i) \equiv P(x_t = i \mid y_{1:T})$$

$$\propto P(x_t = i \mid y_{1:t}) \cdot P(y_{t+1:T} \mid x_t = i)$$

$$\propto \alpha_t(i) \cdot \beta_t(i) \qquad \text{and} \qquad \sum_{i=1}^{K} \gamma_t(i) = 1$$

In the vector form,

$$\boxed{\boldsymbol{\gamma}_t \propto \boldsymbol{\alpha}_t \circ \boldsymbol{\beta}_t \qquad \text{with} \qquad \mathbf{1}_K{}^T \boldsymbol{\gamma}_t = 1}$$

Smoothed distribution

State Estimation Using HMM Smoothing (5)

One more thing,...
When we consider the learning of HMM,
we will need the joint posterior distribution of x_t and x_{t+1} given all outputs $y_{1:T}$ (*), i.e.,

$$\xi_{t,t+1}(i,j) \equiv P(x_t = i, x_{t+1} = j \mid y_{1:T})$$

⇩

How can we compute this ?

(*) In LDS, we also needed the covariance between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$ given $\boldsymbol{y}_{1:T}$

State Estimation Using HMM Smoothing (6)

$$P(x_t, x_{t+1} \mid y_{1:T}) = P(x_t, x_{t+1}, y_{1:T}) / P(y_{1:T})$$

$$\propto P(x_t, x_{t+1}, y_{1:T}) = P(x_t, x_{t+1}, y_{1:t}, y_{t+1}, \boxed{y_{t+2:T}})$$

$$\propto P(y_{t+2:T} \mid \cancel{x_t}, x_{t+1}, \cancel{y_{1:t}}, \cancel{y_{t+1}}) \cdot P(x_t, x_{t+1}, y_{1:t}, \boxed{y_{t+1}})$$

$$\propto P(y_{t+2:T} \mid x_{t+1}) \cdot P(y_{t+1} \mid \cancel{x_t}, x_{t+1}, \cancel{y_{1:t}}) \cdot P(x_t, \boxed{x_{t+1}}, y_{1:t})$$

$$\propto P(y_{t+2:T} \mid x_{t+1}) \cdot P(y_{t+1} \mid x_{t+1}) \cdot P(x_{t+1} \mid x_t, \cancel{y_{1:t}}) \cdot P(\boxed{x_t}, y_{1:t})$$

$$\propto \underbrace{P(y_{t+2:T} \mid x_{t+1})}_{\boldsymbol{\beta}_{t+1}} \cdot \underbrace{P(y_{t+1} \mid x_{t+1})}_{\mathbf{B}_{\bullet, y_{t+1}}} \cdot \underbrace{P(x_{t+1} \mid x_t)}_{\mathbf{A}} \cdot \underbrace{P(x_t \mid y_{1:t})}_{\boldsymbol{\alpha}_t}$$

State Estimation Using HMM Smoothing (7)

From this result, we obtain

$$\xi_{t,t+1}(i,j) = A_{i,j} \cdot \alpha_t(i) \cdot B_{j,y_{t+1}} \cdot \beta_{t+1}(j)$$

Define a matrix $\boldsymbol{\Xi}_{t,t+1}$ whose (i,j)-th element is $\xi_{t,t+1}(i,j)$

$$\boldsymbol{\Xi}_{t,t+1} = \begin{bmatrix} \xi_{t,t+1}(1,1) & \cdots & \xi_{t,t+1}(1,K) \\ \vdots & \ddots & \vdots \\ \xi_{t,t+1}(K,1) & \cdots & \xi_{t,t+1}(K,K) \end{bmatrix}$$

Then, we can compute it by matrix-vector manipulation

$$\boldsymbol{\Xi}_{t,t+1} \propto \boldsymbol{A} \circ \left(\boldsymbol{\alpha}_t \left(\boldsymbol{B}_{\bullet, y_{t+1}} \circ \boldsymbol{\beta}_{t+1} \right)^T \right)$$

with $\sum_{i=1}^{K} \sum_{j=1}^{K} \xi_{t,t+1}(i,j) = \boldsymbol{1}_K{}^T \boldsymbol{\Xi}_{t,t+1} \boldsymbol{1}_K = 1$

State Estimation Using HMM
Decoding Problem

- Decoding : Find the most likely state sequence, given all observations
- i.e., Find $\hat{x}_{1:T} = \arg\max_{x_{1:T}} P(y_{1:T}, x_{1:T})$

Naive approach (exhaustive search):

For all possible sequences of $x_{1:T}$, compute

$$P(y_{1:T}, x_{1:T}) = P(x_1) \cdot \prod_{t=2}^{T} P(x_t \mid x_{t-1}) \cdot \prod_{t=1}^{T} P(y_t \mid x_t)$$

Then determine the sequence that maximize the probability

Although it looks good..

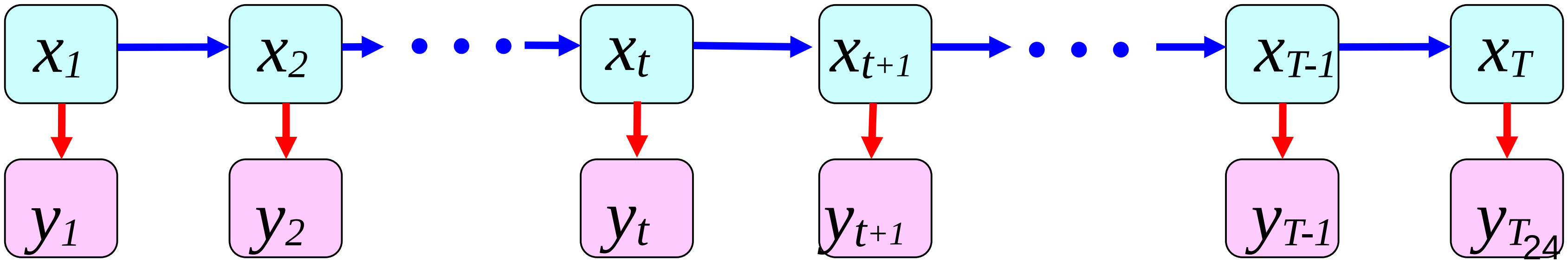
# of patterns of $x_{1:T}$: K^{T+1} Exponential complexity !

Impractical

State Estimation Using HMM Viterbi Algorithm (1)

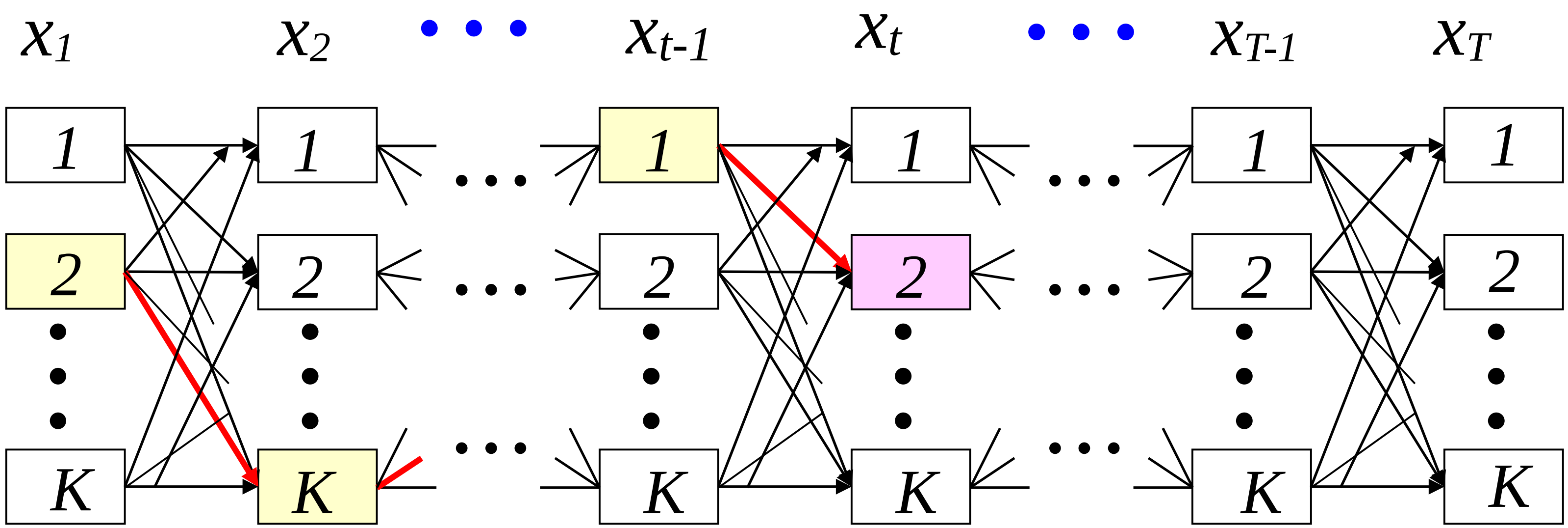
- Decoding problem of HMM is a kind of "optimal path" problem (cf. optimal control)
- Viterbi Algorithm : decoding algorithm based on dynamic programming (DP)
- Idea : Among all paths such that $x_t = i$, you only have to consider the path that maximizes

$$P\left(x_t = i \mid x_{t-1}\right)P(y_{1:t-1}, x_{1:t-1})$$



State Estimation Using HMM Viterbi Algorithm (2)

Possible state sequences (paths)



E.g. Among all paths that $x_t = 2$, the one that maximizes

$P(x_t = 2 \mid x_{t-1}) P(y_{1:t-1}, x_{1:t-1})$ should be considered

Only have to store K paths up to previous time

State Estimation Using HMM
Viterbi Algorithm (3)

As $$P(y_{1:t}, x_{1:t}) = P(y_t \mid x_t) \cdot P(x_t \mid x_{t-1}) \cdot P(y_{1:t-1}, x_{1:t-1})$$

K^t patterns

$$\max_{x_1,\cdots,x_{t-1}} P(y_{1:t}, x_{1:t})$$

$$= \max_{x_1,\cdots,x_{t-1}} P(y_t \mid x_t) \cdot P(x_t \mid x_{t-1}) \cdot P(y_{1:t-1}, x_{1:t-1})$$

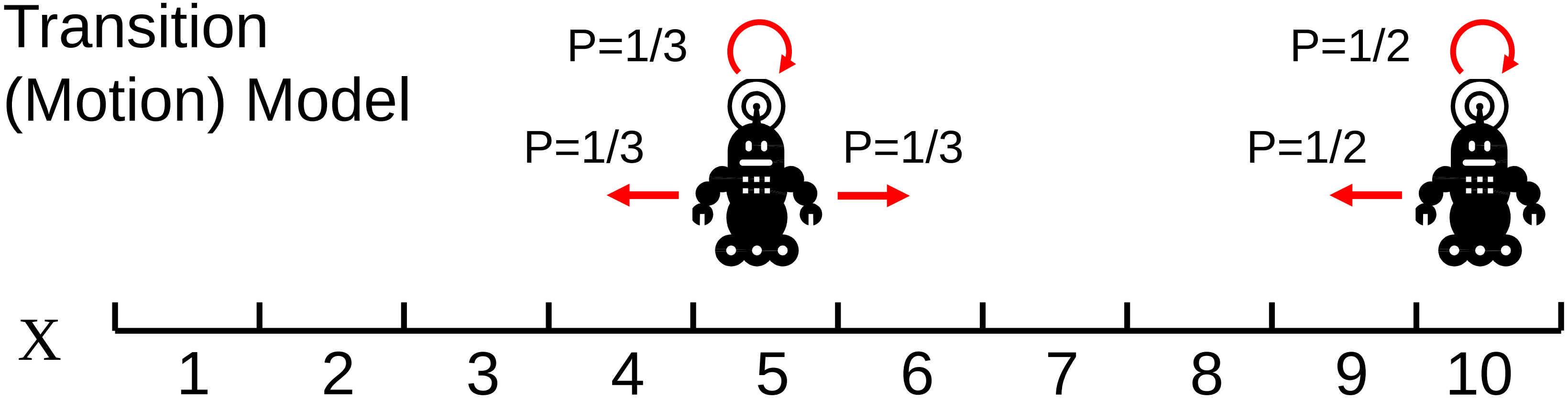
$$= P(y_t \mid x_t) \cdot \max_{x_1,\cdots,x_{t-1}} P(x_t \mid x_{t-1}) \cdot P(y_{1:t-1}, x_{1:t-1})$$

$$= P(y_t \mid x_t) \cdot \max_{x_{t-1}} \left(P(x_t \mid x_{t-1}) \cdot \max_{x_1,\cdots,x_{t-1}} P(y_{1:t-1}, x_{1:t-1}) \right)$$

Just K patterns !

Example : 1-dim Robot Position Estimation from Noisy Observation (1)

Transition (Motion) Model

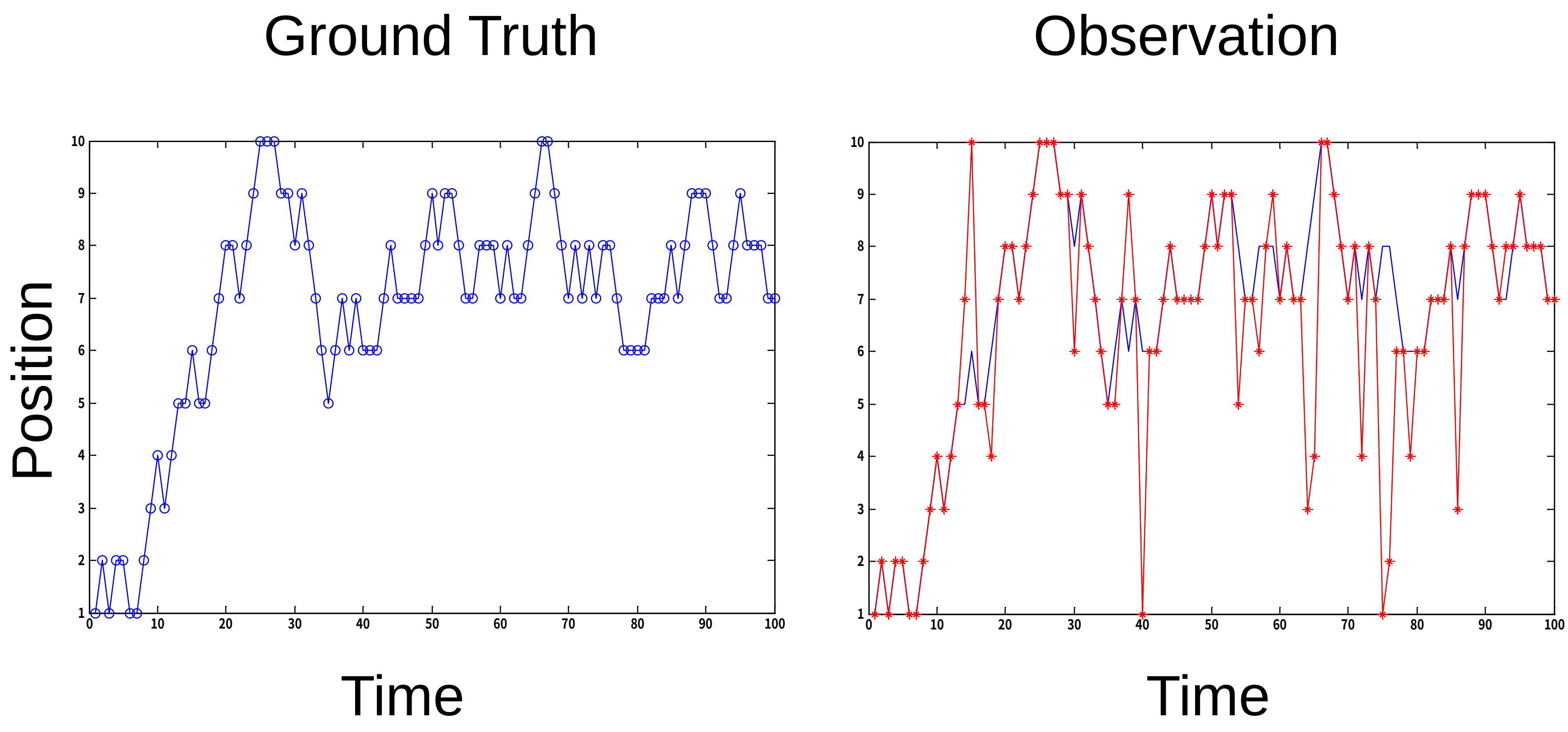


Observation (Sensor) Model

$$\begin{cases} P(x_t = i \mid x_t = i) = 0.8 & \text{Prob. of returning true position} \\ P(y_t = k \mid x_t = i) = 0.0222 \quad (k \neq i) & \text{Prob. of returning incorrect position} \end{cases}$$

Initial Position $P(x_0 = 1) = 1$

Example : 1-dim Robot Position Estimation from Noisy Observation (2)



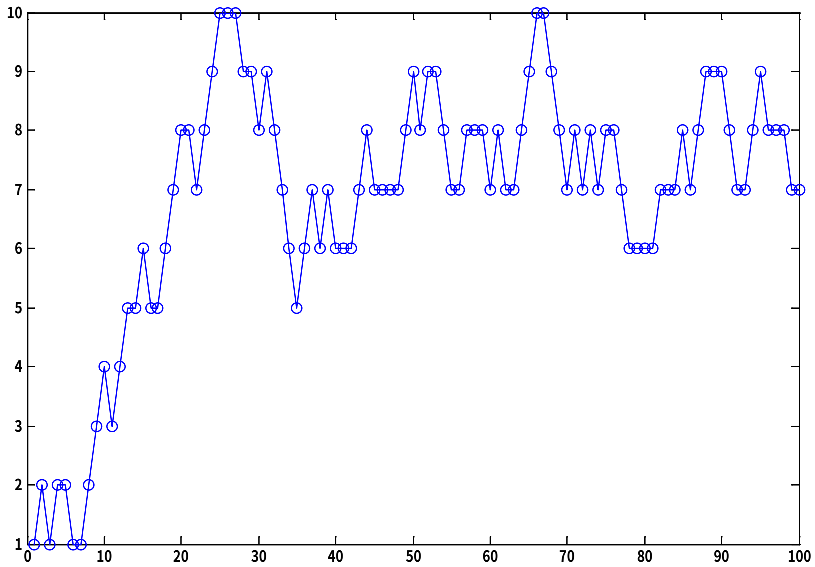
Mean Square Root Error = 0.156

Example : 1-dim Robot Position Estimation from Noisy Observation (3)

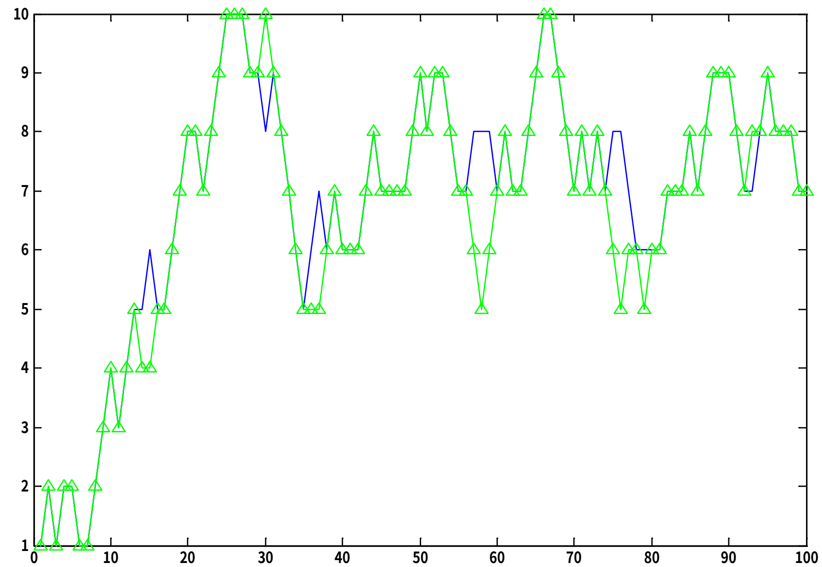
Ground Truth

Estimation by **Viterbi**

Position



Time

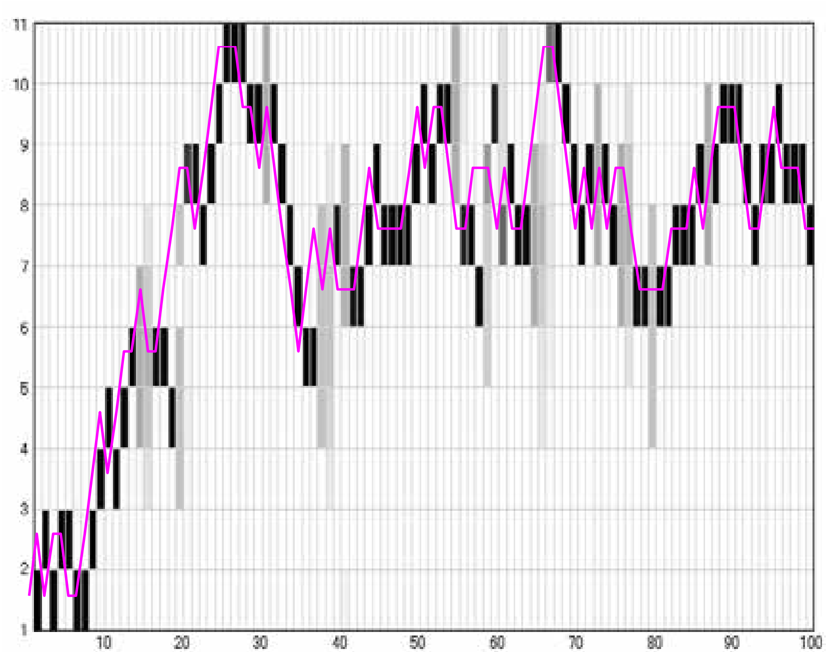


Time

Mean Square Root Error = **0.068**

Example : 1-dim Robot Position Estimation from Noisy Observation (4)

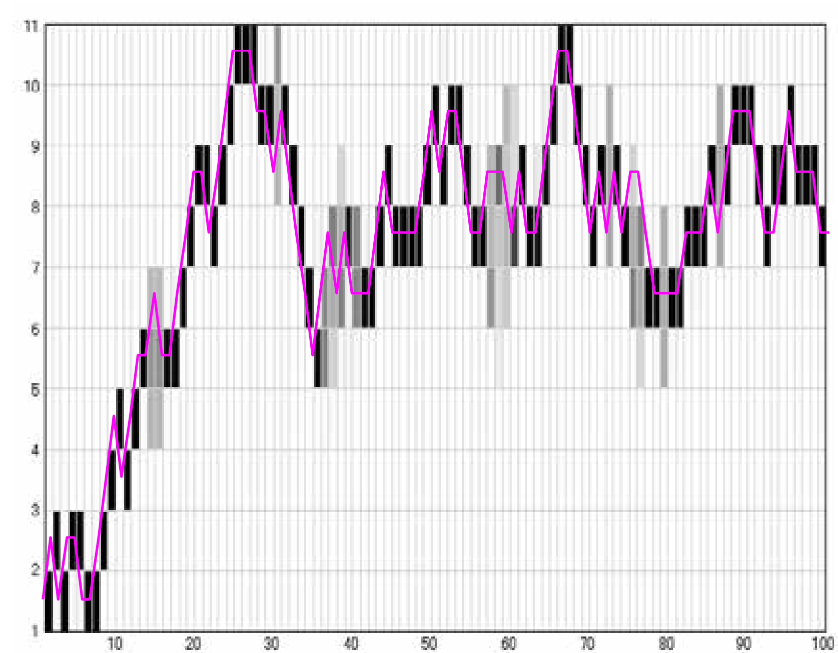
Filtering by Forward Algo.



Position

Time

Smoothing by Forward-Backward



Time

Relationship with Naïve Bayes

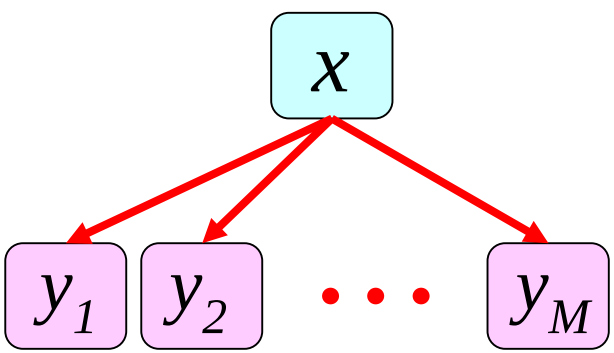
- Naïve Bayes Classifier is an inference method for static systems

$$p(x \mid \mathbf{y}) \propto p(x, \mathbf{y}) = p(\mathbf{y} \mid x) p(x) = p(x) \prod_{j=1}^{M} p(y_j \mid x)$$

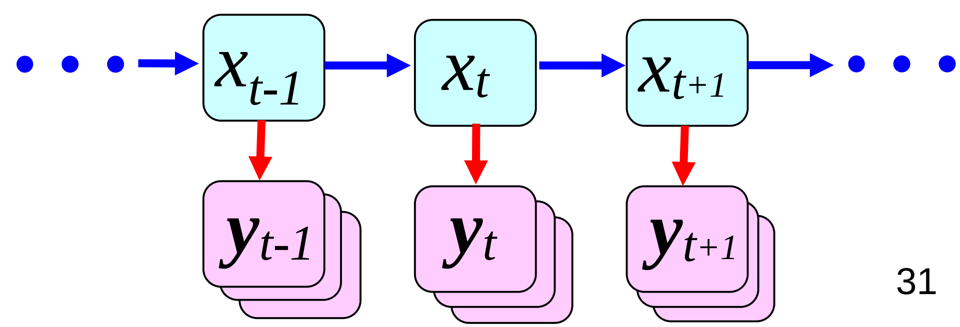
- HMM can be viewed as a dynamic extension of Naïve Bayes

$$p(x_{0:T} \mid \mathbf{y}_{1:T}) \propto p(x_{1:T}, \mathbf{y}_{1:T}) = p(x_0) \prod_{t=1}^{T} p(x_t \mid x_{t-1}) p(y_t \mid x_t)$$

Naïve Bayes Classifier



Hidden Markov Model



Learning of HMM

Supervised Learning of HMM (Discrete Outputs)

- Given:
  - Observation sequence : $\boldsymbol{y}_{1:T}$
  - State sequence : $\boldsymbol{x}_{1:T}$
- Find:
  - Model parameters : $\Theta=\{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$

$$
\begin{cases}
\text{Transition model} & A_{i,j} = p\left(x_{t+1} = j \mid x_t = i\right) \\
\text{Output model} & B_{i,k} = p(y_t = k \mid x_t = i) \\
\text{Initial probability} & \pi_i = p\left(x_1 = i\right)
\end{cases}
$$

Likelihood Function in Supervised Learning (1)

Log-likelihood

$$l(\Theta \mid D) \equiv \boxed{\ln p(\boldsymbol{x}_{1:T}, \boldsymbol{y}_{1:T} \mid \Theta)} = \ln\left\{p(x_1)\prod_{t=1}^{T-1} p(x_{t+1} \mid x_t)\prod_{t=1}^{T} p(y_t \mid x_t)\right\}$$

complete data log-likelihood

Note we can write

$$p(x_1) = \pi_{x_1} = \prod_{i=1}^{K} \pi_i^{\mathrm{I}(x_1 = i)}$$

$$p(x_{t+1} \mid x_t) = A_{x_t, x_{t+1}} = \prod_{i=1}^{K}\prod_{j=1}^{K} A_{i,j}^{\mathrm{I}(x_t = i, x_{t+1} = j)}$$

$$p(y_t \mid x_t) = B_{x_t, y_t} = \prod_{i=1}^{K}\prod_{k=1}^{M} B_{i,k}^{\mathrm{I}(x_t = i, y_t = k)}$$

where $\mathrm{I}(x)$ is the indicator function

Likelihood Function in Supervised Learning (2)

By substituting them,

$$l(\Theta \mid D) = \ln\left\{\prod_{i=1}^{K} \pi_i^{\mathbf{I}(x_1=i)} \prod_{t=1}^{T-1}\prod_{i=1}^{K}\prod_{j=1}^{K} A_{i,j}^{\mathbf{I}(x_t=i,x_{t+1}=j)} \prod_{t=1}^{T}\prod_{i=1}^{K}\prod_{k=1}^{M} B_{i,k}^{\mathbf{I}(x_t=i,y_t=k)}\right\}$$

$$= \sum_{i=1}^{K} \mathbf{I}(x_1 = i)\ln \pi_i + \sum_{i=1}^{K}\sum_{j=1}^{K} N_{i,j} \ln A_{i,j} + \sum_{i=1}^{K}\sum_{k=1}^{M} M_{i,k} \ln B_{i,k}$$

where

$$\left.\begin{aligned} N_{i,j} &\equiv \sum_{i=1}^{K}\sum_{j=1}^{K} \mathbf{I}(x_t = i, x_{t+1} = j) \\ M_{i,k} &\equiv \sum_{i=1}^{K}\sum_{k=1}^{M} \mathbf{I}(x_t = i, y_t = k) \end{aligned}\right\} \text{Can be counted from data}$$

Maximize $l(\Theta|D)$ with constraints :

$$\sum_{i=1}^{K} \pi_i = 1 \;,\; \sum_{j=1}^{K} A_{i,j} = 1 \quad \text{and} \quad \sum_{k=1}^{M} B_{i,k} = 1 \quad \text{(i=1,2,..,K)}$$

Maximum Likelihood Estimation of Supervised Hidden Markov Model (1)

Define the Lagrangian as,

$$L(\Theta,\lambda,\boldsymbol{\mu},\boldsymbol{\nu}) = l(\Theta\,|\,D) + \lambda\left(1-\sum\nolimits_{i=1}^{K}\pi_i\right) + \sum_{i=1}^{K}\mu_i\left(1-\sum\nolimits_{j=1}^{K}A_{i,j}\right) + \sum_{i=1}^{K}\nu_i\left(1-\sum\nolimits_{k=1}^{M}B_{i,k}\right)$$

$$= \sum\nolimits_{i=1}^{K}\mathbf{I}(x_1=i)\ln\pi_i + \sum\nolimits_{i=1}^{K}\sum\nolimits_{j=1}^{K}N_{i,j}\ln A_{i,j} + \sum\nolimits_{i=1}^{K}\sum\nolimits_{k=1}^{M}M_{i,k}\ln B_{i,k}$$
$$+\lambda\left(1-\sum\nolimits_{i=1}^{K}\pi_i\right) + \sum_{i=1}^{K}\mu_i\left(1-\sum\nolimits_{j=1}^{K}A_{i,j}\right) + \sum_{i=1}^{K}\nu_i\left(1-\sum\nolimits_{k=1}^{M}B_{i,k}\right)$$

$$\frac{\partial L}{\partial \pi_i} = \frac{\mathbf{I}(x_1=i)}{\pi_i} - \lambda = 0 \quad\Rightarrow\quad \pi_i = \frac{\mathbf{I}(x_1=i)}{\lambda}$$

By considering the constraint

$$\sum\nolimits_{i=1}^{K}\pi_i = \tfrac{1}{\lambda}\sum\nolimits_{i=1}^{K}\mathbf{I}(x_1=i) = \tfrac{1}{\lambda} = 1 \quad\Rightarrow\quad \hat{\pi}_i = \mathbf{I}(x_1=i)$$

Maximum Likelihood Estimation of Supervised Hidden Markov Model (2)

Derivatives of Lagrangian w.r.t. $A_{i,j}$ and $B_{i,k}$:

$$\frac{\partial L}{\partial A_{i,j}} = \frac{N_{i,j}}{A_{i,j}} - \mu_i = 0 \quad \Rightarrow \quad A_{i,j} = \frac{N_{i,j}}{\mu_i}$$

$$\frac{\partial L}{\partial B_{i,k}} = \frac{M_{i,k}}{B_{i,k}} - \nu_i = 0 \quad \Rightarrow \quad B_{i,k} = \frac{M_{i,k}}{\nu_i}$$

Substitute them into the constraints, we obtain

$$\mu_i = \sum_{j=1}^{K} N_{i,j} \quad \text{and} \quad \nu_i = \sum_{k=1}^{M} M_{i,k}$$

As a result,

$$\hat{A}_{i,j} = \frac{N_{i,j}}{\sum_{j=1}^{K} N_{i,j}} \quad \text{and} \quad \hat{B}_{i,k} = \frac{M_{i,k}}{\sum_{k=1}^{M} M_{i,k}}$$

Almost the same with Simple Markov Chain

Unsupervised Learning of HMM (Discrete Outputs)

- Given:
  - Observation sequence : $\boldsymbol{y}_{1:T}$ Incomplete data
- Find:
  - Model parameters : $\Theta=\{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$
  - State sequence : $\boldsymbol{x}_{1:T}$

EM algorithm, again

(Review) EM Algorithm in General

Given:

Observation sequence

- Data : Y = $\boldsymbol{y}_{1:T}$
- Initial parameter values: $\Theta^{(0)} = \{\boldsymbol{A}^{(0)}, \boldsymbol{B}^{(0)}, \boldsymbol{\pi}^{(0)}\}$

Repeat until convergence

Smoothed dist.

1. [E-step] Compute posterior dist. $q^{*}(X) = p\left(X \mid Y, \Theta^{(t)}\right)$

   and $\mathrm{Q}\left(\Theta \mid \Theta^{(t)}\right) = \mathrm{E}_{q^{*}(X)}\left[\ln p(Y, X \mid \Theta)\right]$

2. [M-step] Maximize Q(Θ|$\Theta^{(t)}$) w.r.t. Θ

$$\Theta^{(t+1)} \leftarrow \arg\max_{\Theta} \mathrm{Q}\left(\Theta \mid \Theta^{(t)}\right)$$

3. $t \leftarrow t+1$

E-step of Learning HMM

For given parameter estimates $\Theta^{(t)} = \{\boldsymbol{A}^{(t)}, \boldsymbol{B}^{(t)}, \boldsymbol{\pi}^{(t)}\}$, perform forward and backward algorithm

Forward: $\boldsymbol{\alpha}_t \equiv [\alpha_t(1), \cdots, \alpha_t(K)]^T$ where $\alpha_t(i) \equiv P(x_t = i \mid y_{1:t})$

(Filtering)

$$\boldsymbol{\alpha}_{t+1} \propto \boldsymbol{B}^{(t)}{}_{\bullet, y_t} \circ \left(\boldsymbol{A}^{(t)^T} \boldsymbol{\alpha}_t \right) \text{ with } \sum_{j=1}^{K} \alpha_{t+1}(j) = 1$$

Backward: $\boldsymbol{\beta}_t \equiv [\beta_t(1), \cdots, \beta_t(K)]^T$ where $\beta_t(i) \equiv P(y_{t+1:T} \mid x_t = i)$

$$\boldsymbol{\beta}_t = \boldsymbol{B}^{(t)}{}_{\bullet, y_{t+1}} \circ \left(\boldsymbol{A}^{(t)} \boldsymbol{\beta}_{t+1} \right) \text{ with } \boldsymbol{\beta}_T = \mathbf{1}_K$$

Smoothing: $\boldsymbol{\gamma}_t \equiv [\gamma_t(1), \cdots, \gamma_t(K)]^T$ $\quad \boldsymbol{\gamma}_t \propto \boldsymbol{\alpha}_t \circ \boldsymbol{\beta}_t$ with $\mathbf{1}_K{}^T \boldsymbol{\gamma}_t = 1$

$$\boldsymbol{\Xi}_{t,t+1} \equiv \begin{bmatrix} \xi_{t,t+1}(1,1) & \cdots & \xi_{t,t+1}(1,K) \\ \vdots & \ddots & \vdots \\ \xi_{t,t+1}(K,1) & \cdots & \xi_{t,t+1}(K,K) \end{bmatrix}$$

where

$\gamma_t(i) \equiv P(x_t = i \mid y_{1:T})$

$\xi_{t,t+1}(i,j) \equiv P(x_t = i, x_{t+1} = j \mid y_{1:T})$

$$\boldsymbol{\Xi}_{t,t+1} \propto \boldsymbol{A} \circ \left(\boldsymbol{\alpha}_t \left(\boldsymbol{B}_{\bullet, y_{t+1}} \circ \boldsymbol{\beta}_{t+1} \right)^T \right) \text{ with } \boldsymbol{1}_K{}^T \boldsymbol{\Xi}_{t,t+1} \boldsymbol{1}_K = 1$$

M-step of Learning HMM

Complete data log-likelihood:

$$\ln p\left(\boldsymbol{x}_{1:T}, \boldsymbol{y}_{1:T} \mid \Theta\right)$$

$$= \sum_{i=1}^{K} \mathbf{I}\left(x_1 = i\right) \ln \pi_i + \sum_{i=1}^{K} \sum_{j=1}^{K} N_{i,j} \ln A_{i,j} + \sum_{i=1}^{K} \sum_{k=1}^{M} M_{i,k} \ln B_{i,k}$$

where $N_{i,j} \equiv \sum_{i=1}^{K} \sum_{j=1}^{K} \mathbf{I}\left(x_t = i, x_{t+1} = j\right)$

$$M_{i,k} \equiv \sum_{i=1}^{K} \sum_{k=1}^{M} \mathbf{I}\left(x_t = i, y_t = k\right)$$

Expected complete data-log-likelihood:

$$\mathrm{Q}\left(\Theta \mid \Theta^{(t)}\right) = \mathrm{E}_{q^*(X)}\left[\ln p\left(\boldsymbol{y}_{1:T}, \boldsymbol{x}_{1:T} \mid \Theta\right)\right]$$

$$= \sum_{i=1}^{K} \boxed{\mathrm{E}\left[\mathbf{I}\left(x_1 = i\right)\right]} \ln \pi_i + \sum_{i=1}^{K} \sum_{j=1}^{K} \boxed{\mathrm{E}\left[N_{i,j}\right]} \ln A_{i,j} + \sum_{i=1}^{K} \sum_{k=1}^{M} \boxed{\mathrm{E}\left[M_{i,k}\right]} \ln B_{i,k}$$

M-step of Learning HMM

We can compute the expected values in Q function:

$$\mathrm{E}\left[\mathbf{I}\left(x_1 = i\right)\right] = \gamma_1(i)$$

$$\mathrm{E}\left[N_{i,j}\right] = \sum_{t=1}^{T-1} \xi_{t,t+1}(i, j)$$

$$\mathrm{E}\left[M_{i,k}\right] = \sum_{t=1}^{T} \gamma_t(i) \mathbf{I}\left(y_t = k\right) = \sum_{y_t=k} \gamma_t(i)$$

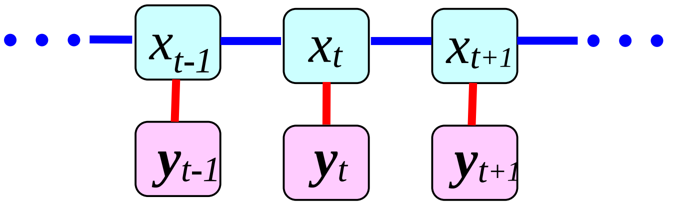
By maximizing $Q(\Theta|\Theta^{(t)})$, we obtain new parameter estimates:

$$\begin{cases} \pi_i^{(t+1)} = \mathrm{E}\left[\mathbf{I}\left(x_1 = i\right)\right] = \gamma_1(i) \\ A_{i,j}^{(t+1)} = \mathrm{E}\left[N_{i,j}\right] \Big/ \sum_{j=1}^{K} \mathrm{E}\left[N_{i,j}\right] = \sum_{t=1}^{T-1} \xi_{t,t+1}(i, j) \Big/ \sum_{j=1}^{K} \sum_{t=1}^{T-1} \xi_{t,t+1}(i, j) \\ B_{i,k}^{(t+1)} = \mathrm{E}\left[M_{i,k}\right] \Big/ \sum_{k=1}^{M} \mathrm{E}\left[M_{i,k}\right] = \sum_{y_t=k} \gamma_t(i) \Big/ \sum_{k=1}^{M} \sum_{y_t=k} \gamma_t(i) \end{cases}$$

(Advanced) Linear-Chain Conditional Random Fields

- Conditional Random Fields (CRF) : A discriminative approach to labeling structured data (including time-series)
- CRF models the conditional probability $\mathrm{p}(x_{1:\mathrm{T}} \mid \boldsymbol{y}_{1:\mathrm{T}})$

$$p\left(x_{1:T} \mid \boldsymbol{y}_{1:T}\right) = \frac{1}{Z\left(\boldsymbol{y}_{1:T}\right)} \prod_{t=1}^{T} \psi_t\left(x_t, x_{t-1}, \boldsymbol{y}_t\right)$$



- CRF is represented by undirected graph
- CRF is better than HMM in prediction accuracy
- Supervised training of CRF is more difficult (complicated) than that of HMM
  - Numerical optimization is necessary