

# Probabilistic Models of Time Series and Sequences

Zitao Liu

## Agenda

- Introduction
- Markov Models
- Hidden Markov Models
  - Inference
  - Learning
- Linear Dynamical Systems
  - Inference
  - Learning
- Recent Work

# Agenda

- Introduction
- Markov Models
- Hidden Markov Models
  - Inference
  - Learning
- Linear Dynamical Systems
  - Inference
  - Learning
- Recent Work

## Introduction

---

- Time series are everywhere...



Stock Price Series

## Introduction

---

- Time series are everywhere...



Hear Beat Rate Series

## Introduction

---

- Time series are everywhere...



Weather Temperature Series

## Introduction

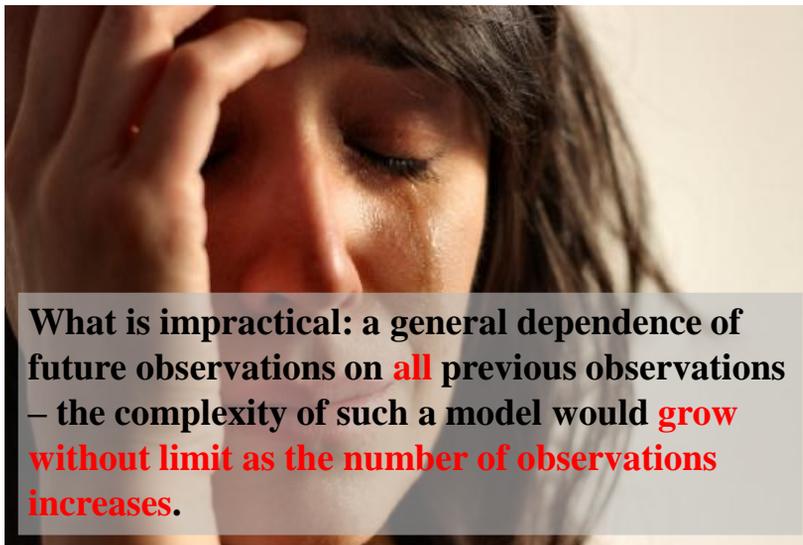
---



- **What We Want:**  
Be able to **predict** the **next value** in a time series given observations of the **previous values**.

## Introduction

---



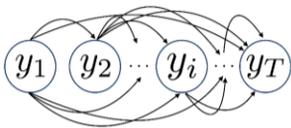
What is impractical: a general dependence of future observations on **all** previous observations – the complexity of such a model would **grow without limit as the number of observations increases**.

# Agenda

- Introduction
- **Markov Models**
- Hidden Markov Models
  - Inference
  - Learning
- Linear Dynamical Systems
  - Inference
  - Learning
- Recent Work

## **Markov Models**

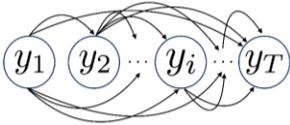
---



Fully dependent

## Markov Models

---



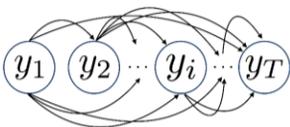
Fully dependent



Fully independent

## Markov Models

---



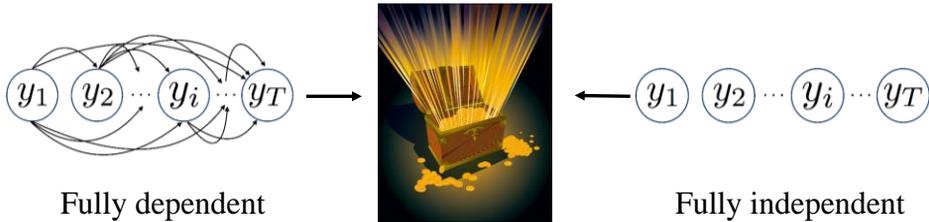
Fully dependent



Fully independent

## Markov Models

---



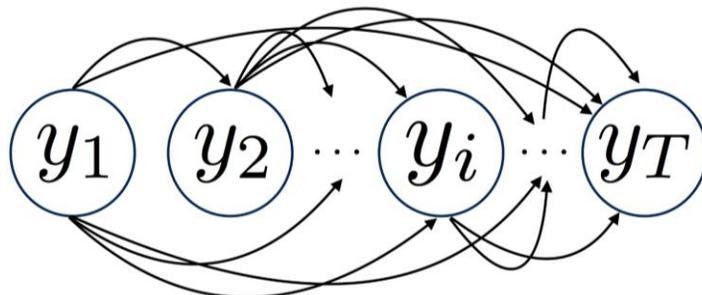
## Markov Models

**Markov Models: Future predictions are independent of all but the most recent observations.**

## Markov Models

---

**Markov Models: Future predictions are independent of all but the most recent observations.**

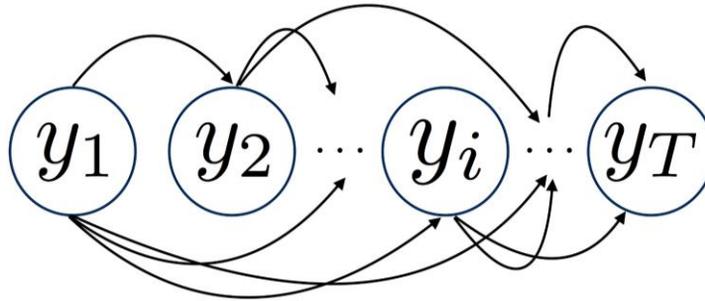


Fully dependent

## Markov Models

---

**Markov Models:** Future predictions are independent of all but **the most recent observations.**

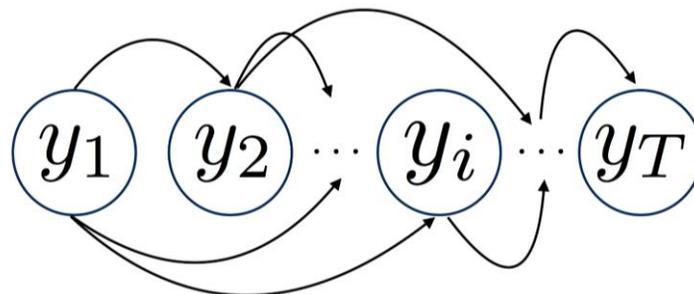


Partially dependent

## Markov Models

---

**Markov Models:** Future predictions are independent of all but **the most recent observations.**

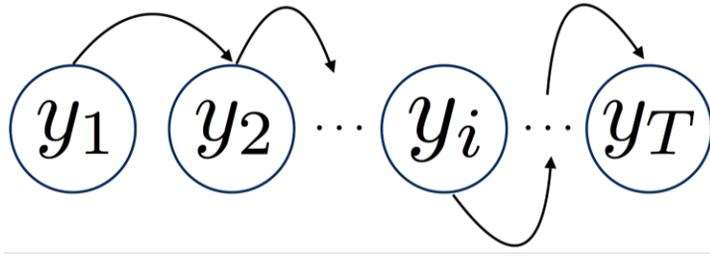


Partially dependent

## Markov Models

---

**Markov Models:** Future predictions are independent of all but **the most recent observations**.



Partially dependent

## Markov Models

---

- Make it for formal (**joint probability**) and simple (consider only 4 observations  $\{y_1, y_2, y_3, y_4\}$ ).

## Markov Models

---

- Make it for formal (**joint probability**) and simple (consider only 4 observations  $\{y_1, y_2, y_3, y_4\}$ ).
- Fully dependent:

$$p(y_1, y_2, y_3, y_4) = p(y_1)p(y_2|y_1)p(y_3|y_1, y_2)p(y_4|y_1, y_2, y_3)$$

## Markov Models

---

- Make it for formal (**joint probability**) and simple (consider only 4 observations  $\{y_1, y_2, y_3, y_4\}$ ).
- Fully dependent:

$$p(y_1, y_2, y_3, y_4) = p(y_1)p(y_2|y_1)p(y_3|y_1, y_2)p(y_4|y_1, y_2, y_3)$$

- Fully independent:

$$p(y_1, y_2, y_3, y_4) = p(y_1)p(y_2)p(y_3)p(y_4)$$

## Markov Models

---

- Make it for formal (**joint probability**) and simple (consider only 4 observations  $\{y_1, y_2, y_3, y_4\}$ ).

- Fully dependent:

$$p(y_1, y_2, y_3, y_4) = p(y_1)p(y_2|y_1)p(y_3|y_1, y_2)p(y_4|y_1, y_2, y_3)$$

- Fully independent:

$$p(y_1, y_2, y_3, y_4) = p(y_1)p(y_2)p(y_3)p(y_4)$$

- Partially dependent (Depend on most recent observation):

$$p(y_1, y_2, y_3, y_4) = p(y_1)p(y_2|y_1)p(y_3|y_2)p(y_4|y_3)$$

## Markov Models

---

- Make it for formal (**joint probability**) and simple (consider only 4 observations  $\{y_1, y_2, y_3, y_4\}$ ).

- Fully dependent:

$$p(y_1, y_2, y_3, y_4) = p(y_1)p(y_2|y_1)p(y_3|y_1, y_2)p(y_4|y_1, y_2, y_3)$$

- Fully independent:

$$p(y_1, y_2, y_3, y_4) = p(y_1)p(y_2)p(y_3)p(y_4)$$

- Partially dependent (Depend on most recent observation):

$$p(y_1, y_2, y_3, y_4) = p(y_1)p(y_2|y_1)p(y_3|y_2)p(y_4|y_3)$$

- Partially dependent (Depend on two most recent observation):

$$p(y_1, y_2, y_3, y_4) = p(y_1)p(y_2|y_1)p(y_3|y_1, y_2)p(y_4|y_2, y_3)$$

## Markov Models

---

- Make it for formal (**joint probability**) and simple (consider only 4 observations  $\{y_1, y_2, y_3, y_4\}$ ).

- Fully dependent:

$$p(y_1, y_2, y_3, y_4) = p(y_1)p(y_2|y_1)p(y_3|y_1, y_2)p(y_4|y_1, y_2, y_3)$$

- Fully independent (Depend on no previous observations)
 

First-order Markov chain

$$p(y_1, y_2, y_3, y_4) = p(y_1)p(y_2)p(y_3)p(y_4)$$

- Partially dependent (Depend on most recent observation)
 

Second-order Markov chain

$$p(y_1, y_2, y_3, y_4) = p(y_1)p(y_2|y_1)p(y_3|y_1, y_2)p(y_4|y_1, y_2, y_3)$$

- Partially dependent (Depend on two most recent observations)
 
$$p(y_1, y_2, y_3, y_4) = p(y_1)p(y_2|y_1)p(y_3|y_1, y_2)p(y_4|y_2, y_3)$$

## Markov Models

---

- Make it more formal...

### First-order Markov chain

$$p(y_1, y_2, \dots, y_T) = p(y_1) \sum_{t=2}^T p(y_t|y_{t-1})$$

### Second-order Markov chain

$$p(y_1, y_2, \dots, y_T) = p(y_1)p(y_2|y_1) \sum_{t=3}^T p(y_t|y_{t-1}, y_{t-2})$$

## Markov Models

---

- A toy example.

Consider a simple 3-state Markov model of the weather. We assume that once a day (e.g., at noon), the weather is observed as being one of the following:

- State 1: rain or (snow)
- State 2: cloudy
- State 3: sunny

Assume that the weather on day  $t$  is characterized by a single one of the three states above, and that the matrix  $A$  of state transition probabilities is:

$$A = \{A_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

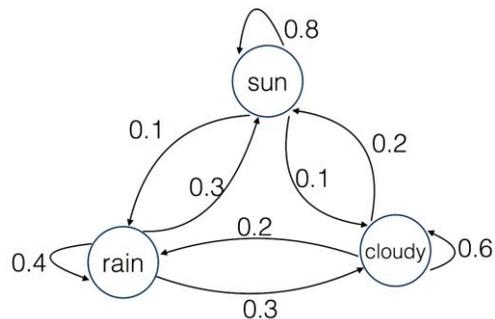
## Markov Models

---

- A toy example.

- State 1: rain or (snow)
- State 2: cloudy
- State 3: sunny

$$A = \{A_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$



## Markov Models

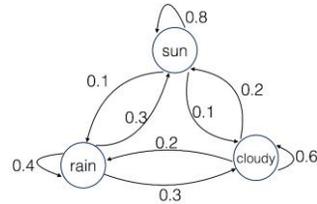
---

- Question?

Given that the weather on day 1 ( $t = 1$ ) is sunny (state 3), what is the probability that weather for the next 7 days will be :

“sun-sun-rain-rain-sun-cloudy-sun”

“ $y_2 - y_3 - y_4 - y_5 - y_6 - y_7 - y_8$ ”



## Markov Models

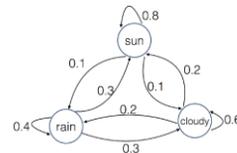
---

- Question?

Given that the weather on day 1 ( $t = 1$ ) is sunny (state 3), what is the probability that weather for the next 7 days will be :

“sun-sun-rain-rain-sun-cloudy-sun”

“ $y_2 - y_3 - y_4 - y_5 - y_6 - y_7 - y_8$ ”



$$\begin{aligned}
 P(y_1, y_2, \dots, y_7, y_8 | A) &= P(y_1 = s)P(y_2 = s | y_1 = s)P(y_3 = s | y_2 = s) \\
 &\quad P(y_4 = r | y_3 = s)P(y_5 = r | y_4 = r)P(y_6 = s | y_5 = r) \\
 &\quad P(y_7 = c | y_6 = s)P(y_8 = s | y_7 = c) \\
 &= 1 \cdot A_{33} \cdot A_{33} \cdot A_{31} \cdot A_{11} \cdot A_{13} \cdot A_{32} \cdot A_{23} \\
 &= 1 \cdot 0.8 \cdot 0.8 \cdot 0.1 \cdot 0.4 \cdot 0.3 \cdot 0.1 \cdot 0.2 = 1.536 \times 10^{-4}
 \end{aligned}$$

## Markov Models

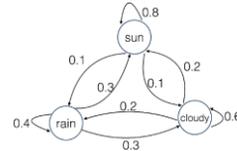
---

- Question?

Given that the weather on day 1 ( $t = 1$ ) is sunny (state 3), what is the probability that weather for the next 7 days will be :

“sun-sun-rain-rain-sun-cloudy-sun”

“ $y_2 - y_3 - y_4 - y_5 - y_6 - y_7 - y_8$ ”



$$\begin{aligned}
 P(y_1, y_2, \dots, y_7, y_8 | A) &= P(y_1 = s)P(y_2 = s | y_1 = s)P(y_3 = s | y_2 = s) \\
 &\quad P(y_4 = r | y_3 = s)P(y_5 = r | y_4 = r)P(y_6 = s | y_5 = r) \\
 &\quad P(y_7 = c | y_6 = s)P(y_8 = s | y_7 = c) \\
 &= 1 \cdot A_{33} \cdot A_{33} \cdot A_{31} \cdot A_{11} \cdot A_{13} \cdot A_{32} \cdot A_{23} \\
 &= 1 \cdot 0.8 \cdot 0.8 \cdot 0.1 \cdot 0.4 \cdot 0.3 \cdot 0.1 \cdot 0.2 = 1.536 \times 10^{-4}
 \end{aligned}$$

- Other questions we may interested in? For example, what's the probability of the day after tomorrow is cloudy?

## Markov Models

---

- All above are observable Markov models.

A single stochastic process with Markov assumptions.



**Too restrictive to be applicable to many real problems.**

## Markov Models

---

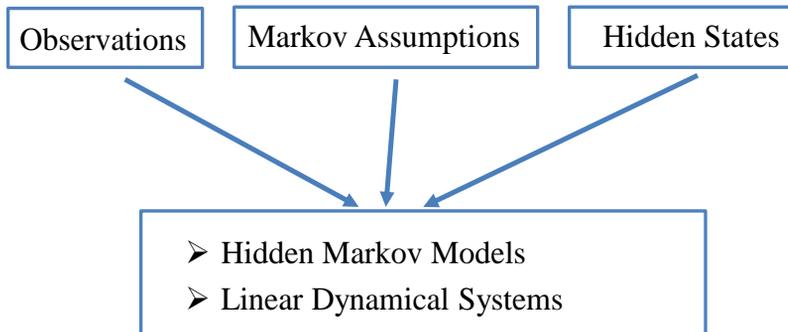
- All above are observable Markov models.



## Markov Models

---

- All above are observable Markov models.

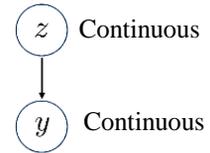
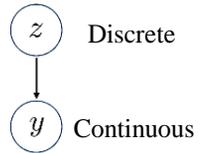
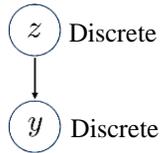


A more **general** framework, while still retaining inference **tractability**.

## Markov Models

---

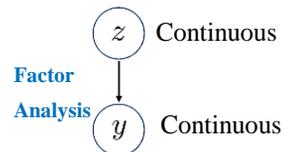
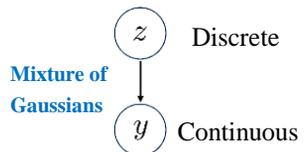
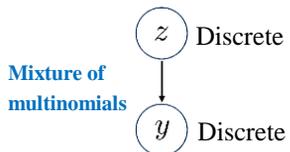
- Connections.



## Markov Models

---

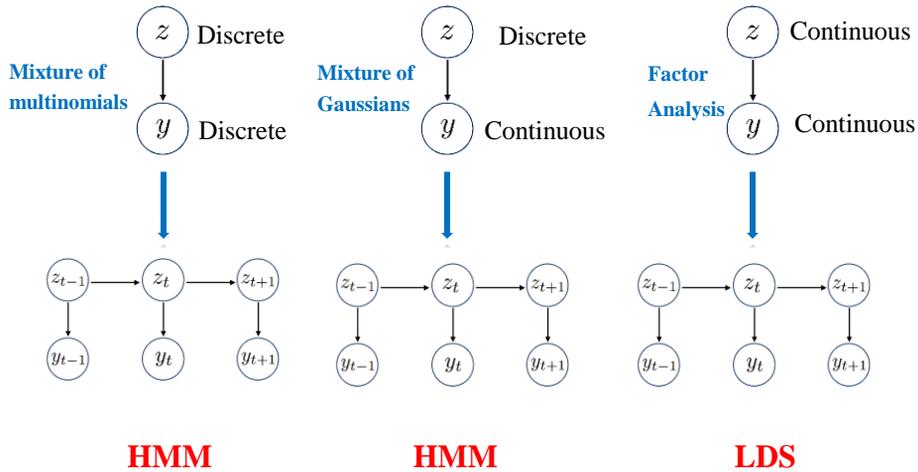
- Connections.



## Markov Models

---

- Connections.



## Agenda

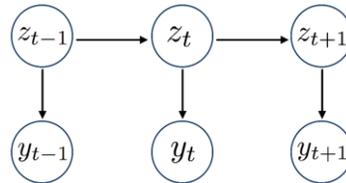
- Introduction
- Markov Models
- Hidden Markov Models
  - Inference
  - Learning
- Linear Dynamical Systems
  - Inference
  - Learning
- Recent Work

## Hidden Markov Models

---

- Overview

- A sequence of length  $T$
- Observations  $\{y_t\}$ : **discrete** or **continuous**
- Hidden states  $\{z_t\}$ : **discrete**



- Joint distribution of the model.

$$p(y_1, y_2, \dots, y_T, z_1, z_2, \dots, z_T) = p(z_1) \prod_{t=2}^T p(z_t | z_{t-1}) \prod_{t=1}^T p(y_t | z_t)$$

## Hidden Markov Models

---

- Hidden state transitions  $p(z_t | z_{t-1})$
- Latent variables are the discrete **multinomial** variables  $z_t$  describing which component of the mixture is responsible for generating the corresponding observation  $y_t$ .
- $z_t \in \{1, 2, 3, \dots, d\}$ , where  $d$  is the dimension of the hidden states.

$$p(z_t = i | z_{t-1} = j) \equiv A_{ij} \quad 0 \leq A_{ij} \leq 1, \sum_i^d A_{ij} = 1$$

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1d} \\ \vdots & \ddots & \vdots \\ A_{d1} & \cdots & A_{dd} \end{bmatrix}$$

## Hidden Markov Models

---

- State-to-observation emissions  $p(y_t|z_t)$

- When  $y_t$  is discrete,  $p(y_t|z_t)$  is a **probabilistic table**.

$$p(y_t = i|z_t = j) \equiv C_{ij} \quad C = \begin{bmatrix} C_{11} & \cdots & C_{1d} \\ \vdots & \ddots & \vdots \\ C_{K1} & \cdots & C_{Kd} \end{bmatrix}$$

$$0 \leq C_{ij} \leq 1, \sum_i^K C_{ij} = 1$$

## Hidden Markov Models

---

- State-to-observation emissions  $p(y_t|z_t)$

- When  $y_t$  is discrete,  $p(y_t|z_t)$  is a **probabilistic table**.

$$p(y_t = i|z_t = j) \equiv C_{ij} \quad C = \begin{bmatrix} C_{11} & \cdots & C_{1d} \\ \vdots & \ddots & \vdots \\ C_{K1} & \cdots & C_{Kd} \end{bmatrix}$$

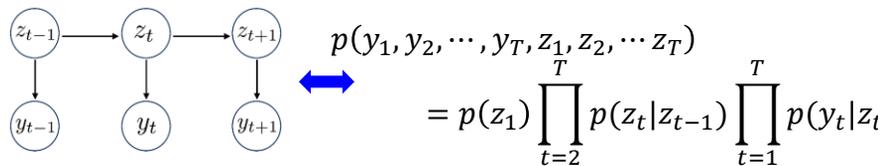
$$0 \leq C_{ij} \leq 1, \sum_i^K C_{ij} = 1$$

- When  $y_t$  is continuous,  $p(y_t|z_t)$  is a **continuous density function**, i.e., Gaussian, etc.

$$p(y_t|z_t) \text{ is Gaussian, } C = \{\mu_1, \Sigma_1, \mu_2, \Sigma_2, \dots, \mu_d, \Sigma_d\}.$$

## Hidden Markov Models

---



- What we can do if we have the model ? - **Inference**
  - Q1: Given an observation sequence and the model, compute the probability of the sequence.
  - Q2: Given the observation sequence and the model, compute the most likely (ML) hidden state sequence.
- How can we get the model ?- **Learning**
  - Q3: Learning of parameters of HMM (ML parameter estimate).

## Hidden Markov Models

---

- Notations
  - Let  $\mathbf{y} = \{y_1, y_2, \dots, y_T\}$  be a sequence of  $T$  observations.
  - Let  $\mathbf{z} = \{z_1, z_2, \dots, z_T\}$  be a sequence of  $T$  internal states.
  - Let  $d$  be the dimension of the hidden states  $\mathbf{z}$ .
  - Let  $\xi = p(z_1)$  be the initial probability for  $z_1$  and  $\xi_i = p(z_1 = i)$ .
  - Let  $A$  be the  $d \times d$  transition matrix of  $\mathbf{z}$ .
  - Let  $C$  be the parameters in the emission function.
  - Let  $\Omega$  be the entire HMM model, i.e.,  $\Omega = \{\xi, A, C\}$ .

## Hidden Markov Models

---

- Q1: Probability of an observed sequence  $p(y_1, y_2, \dots, y_T | \Omega)$ :  
**forward/backward algorithm.**

$$\begin{aligned}
 p(\mathbf{y}) &= \sum_{\mathbf{z}} p(\mathbf{y}, \mathbf{z}) = \sum_{\mathbf{z}} p(\mathbf{y}|\mathbf{z})p(\mathbf{z}) \\
 &= \sum_{\mathbf{z}} \left[ \underbrace{\prod_{t=1}^T p(y_t|z_t)}_{p(\mathbf{y}|\mathbf{z})} \cdot p(z_1) \cdot \underbrace{\prod_{t=2}^T p(z_t|z_{t-1})}_{p(\mathbf{z})} \right]
 \end{aligned}$$

## Hidden Markov Models

---

- Q1: Probability of an observed sequence  $p(y_1, y_2, \dots, y_T | \Omega)$ :  
**forward/backward algorithm.**

$$\begin{aligned}
 p(\mathbf{y}) &= \sum_{\mathbf{z}} p(\mathbf{y}, \mathbf{z}) = \sum_{\mathbf{z}} p(\mathbf{y}|\mathbf{z})p(\mathbf{z}) \\
 &= \sum_{\mathbf{z}} \left[ \underbrace{\prod_{t=1}^T p(y_t|z_t)}_{p(\mathbf{y}|\mathbf{z})} \cdot p(z_1) \cdot \underbrace{\prod_{t=2}^T p(z_t|z_{t-1})}_{p(\mathbf{z})} \right]
 \end{aligned}$$

Naïve solution:  $O(d^T)$



## Hidden Markov Models

---

- Q1: Probability of an observed sequence  $p(y_1, y_2, \dots, y_T | \Omega)$ :  
**forward algorithm.**

$$p(\mathbf{y}) = \sum_{\mathbf{z}} p(\mathbf{y}, \mathbf{z}) = \sum_{\mathbf{z}} p(\mathbf{y} | \mathbf{z}) \longrightarrow p(\mathbf{y}) = \sum_{i=1}^d p(y_1, y_2, \dots, y_T, z_T = i)$$

## Hidden Markov Models

---

- Q1: Probability of an observed sequence  $p(y_1, y_2, \dots, y_T | \Omega)$ :  
**forward algorithm.**

$$p(\mathbf{y}) = \sum_{\mathbf{z}} p(\mathbf{y}, \mathbf{z}) = \sum_{\mathbf{z}} p(\mathbf{y} | \mathbf{z}) \longrightarrow p(\mathbf{y}) = \sum_{i=1}^d p(y_1, y_2, \dots, y_T, z_T = i)$$

Can we get  $p(y_1, y_2, \dots, y_T, z_T = i)$   
from  $\Omega$  and  $\sum_{j=1}^d p(y_1, y_2, \dots, y_{T-1}, z_{T-1} = j)$  ?

## Hidden Markov Models

---

- Q1: Probability of an observed sequence  $p(y_1, y_2, \dots, y_T | \Omega)$ :  
**forward algorithm.**

$$p(\mathbf{y}) = \sum_{\mathbf{z}} p(\mathbf{y}, \mathbf{z}) = \sum_{\mathbf{z}} p(\mathbf{y} | \mathbf{z}) \longrightarrow p(\mathbf{y}) = \sum_{i=1}^d p(y_1, y_2, \dots, y_T, z_T = i)$$

Can we get  $p(y_1, y_2, \dots, y_T, z_T = i)$   
from  $\Omega$  and  $\sum_{j=1}^d p(y_1, y_2, \dots, y_{T-1}, z_{T-1} = j)$  ?

$$\begin{aligned} p(y_1, y_2, \dots, y_T, z_T = i) &= \sum_{j=1}^d p(y_1, y_2, \dots, y_{T-1}, y_T, z_{T-1} = j, z_T = i) \\ &= \sum_{j=1}^d p(y_1, y_2, \dots, y_{T-1}, z_{T-1} = j) p(z_T = i | z_{T-1} = j) p(y_T | z_T = i) \quad \text{Recursion!} \end{aligned}$$

## Hidden Markov Models

---

**Recursion we find:**

$$p(y_1, y_2, \dots, y_t, z_t = i) = \sum_{j=1}^d p(y_1, y_2, \dots, y_{t-1}, z_{t-1} = j) p(z_t = i | z_{t-1} = j) p(y_t | z_t = i)$$

Let  $\alpha_t(i) \equiv p(y_1, y_2, \dots, y_t, z_t = i)$ , which represents the total probability of all the observations up through time  $t$  and that we are in state  $i$  at time  $t$ .

**Recursive rules:**

$$\alpha_t(i) = \sum_j^d \alpha_{t-1}(j) \cdot A_{ij} \cdot p(y_t | z_t = i)$$

**Base case:**

$$\alpha_1(i) = p(y_1, z_1 = i) = p(y_1 | z_1 = i) p(z_1 = i) = \xi_i \cdot p(y_1 | z_1 = i)$$

**Our goal:**

$$p(\mathbf{y}) = \sum_{i=1}^d \alpha_T(i)$$

## Hidden Markov Models

---

- Q1: Probability of an observed sequence  $p(y_1, y_2, \dots, y_T | \Omega)$ :  
**backward algorithm.**

**Very similar to forward algorithm!!!**

Let  $\beta_t(i) \equiv p(y_{t+1}, y_{t+2}, \dots, y_T | z_t = i)$

**Recursive rules:**

$$\beta_t(i) = \sum_j^d \beta_{t+1}(j) \cdot A_{ij} \cdot p(y_{T+1} | z_{T+1} = j)$$

**Base case:**

$$\beta_T(i) = p(y_T, z_{T-1} = i) = p(y_1 | z_1 = i) p(z_1 = i) = \xi_i \cdot p(y_1 | z_1 = i)$$

**Our goal:**

$$p(\mathbf{y}) = \sum_{i=1}^d \beta_1(i)$$

## Hidden Markov Models

---

- Q2: Find the most likely sequence of states.

“**most likely**” state sequence, i.e.,  $\tilde{\mathbf{z}}$ .

Option 1:  $\tilde{z}_t = \operatorname{argmax}_{z_t} p(z_t | \mathbf{y}, \Omega)$

Option 2:  $\tilde{\mathbf{z}} = \operatorname{argmax}_{\mathbf{z}} p(\mathbf{z} | \mathbf{y}, \Omega)$

## Hidden Markov Models

---

- Q2: Find the most likely sequence of states (Option 1: **piece together most likely states**, i.e.,  $\operatorname{argmax}_{z_t} p(z_t | \mathbf{y}, \Omega)$ ).

$$\begin{aligned} \operatorname{argmax}_{z_t} p(z_t | \mathbf{y}) &= \operatorname{argmax}_{z_t} \frac{p(z_t, \mathbf{y})}{p(\mathbf{y})} = \operatorname{argmax}_{z_t} p(z_t, \mathbf{y}) \\ &= \operatorname{argmax}_{z_t} p(y_1, \dots, y_t, z_t) p(y_{t+1}, \dots, y_T | z_t) \\ &= \operatorname{argmax}_i \alpha_t(i) \beta_t(i) \end{aligned}$$

$$\alpha_t(i) \equiv p(y_1, y_2, \dots, y_t, z_t = i)$$

$$\beta_t(i) \equiv p(y_{t+1}, y_{t+2}, \dots, y_T | z_t = i)$$

## Hidden Markov Models

---

- Q2: Find the most likely sequence of states (Option 1: **piece together most likely states**, i.e.,  $\operatorname{argmax}_{z_t} p(z_t | \mathbf{y}, \Omega)$ ).

$$\begin{aligned} \operatorname{argmax}_{z_t} p(z_t | \mathbf{y}) &= \operatorname{argmax}_{z_t} \frac{p(z_t, \mathbf{y})}{p(\mathbf{y})} = \operatorname{argmax}_{z_t} p(z_t, \mathbf{y}) \\ &= \operatorname{argmax}_{z_t} p(y_1, \dots, y_t, z_t) p(y_{t+1}, \dots, y_T | z_t) \\ &= \operatorname{argmax}_i \alpha_t(i) \beta_t(i) \end{aligned}$$

$$\alpha_t(i) \equiv p(y_1, y_2, \dots, y_t, z_t = i)$$

$$\beta_t(i) \equiv p(y_{t+1}, y_{t+2}, \dots, y_T | z_t = i)$$

**Problem:** what if the transition between the two most likely states is 0?

## Hidden Markov Models

---

- Q2: Find the most likely sequence of states (Option 2: Find the best “continuous” sequence of states,  $\operatorname{argmax}_{\mathbf{z}} p(\mathbf{z}|\mathbf{y}, \Omega)$ ).
- **The Viterbi Algorithm.**

$$\operatorname{argmax}_{\mathbf{z}} p(\mathbf{z}|\mathbf{y}) = \operatorname{argmax}_{\mathbf{z}} \frac{p(\mathbf{y}, \mathbf{z})}{p(\mathbf{y})} = \operatorname{argmax}_{\mathbf{z}} p(\mathbf{y}, \mathbf{z})$$

Remember in Q1, we want to compute  $p(\mathbf{y}) = \sum_{\mathbf{z}} p(\mathbf{y}, \mathbf{z})$

Almost the same with forward algorithm!

## Hidden Markov Models

---

- Q2: Find the most likely sequence of states (Option 2: Find the best “continuous” sequence of states,  $\operatorname{argmax}_{\mathbf{z}} p(\mathbf{z}|\mathbf{y}, \Omega)$ ).
- **The Viterbi Algorithm.**

$$\operatorname{argmax}_{\mathbf{z}} p(\mathbf{z}|\mathbf{y}) = \operatorname{argmax}_{\mathbf{z}} \frac{p(\mathbf{y}, \mathbf{z})}{p(\mathbf{y})} = \operatorname{argmax}_{\mathbf{z}} p(\mathbf{y}, \mathbf{z})$$

**Define**  $\delta_t(i) \equiv \max_{z_1, z_2, \dots, z_{t-1}} p(z_1, \dots, z_{t-1}, y_1, \dots, y_t, z_t = i)$ .

**Remember**  $\alpha_t(i) \equiv p(y_1, y_2, \dots, y_t, z_t = i)$

$\delta_t(i)$  represents the highest probability along a single path that accounts for the first  $t$  observations and ends at state  $z_t = i$ .

**Recursive rules:**  $\delta_{t+1}(j) = \max_i [\delta_t(i) A_{ij}] p(y_{t+1} | z_{t+1} = j)$

## Hidden Markov Models

---

- Q3: Learning of parameters of HMM (ML parameter estimate),  $\Omega$ .  
**The Baum-Welch Algorithm.**

Add-on Notations:

- Let  $z_{t,i}$  be a binary variable, where  $z_{t,i} = 1$  means  $z_t = i$ .
- Let  $\gamma(z_{t,i})$  be the expectation of  $z_{t,i}$ , i.e.,  $\gamma(z_{t,i}) = \mathbb{E}[z_{t,i}] = p(z_{t,i}|\mathbf{y})$ .
- Let  $\eta(z_{t-1,j}, z_{t,i})$  be the expectation of  $z_{t-1,j}z_{t,i}$ , i.e.,  
 $\eta(z_{t-1,j}, z_{t,i}) = \mathbb{E}[z_{t-1,j}z_{t,i}] = p(z_{t-1,j}, z_{t,i}|\mathbf{y})$ .
- Let  $p(y_t|z_t, C) = \prod_{k=1}^d p(y_t|C_k)^{z_{t,k}}$ .

## Hidden Markov Models

---

- Q3: Learning of parameters of HMM (ML parameter estimate),  $\Omega$ .  
**The Baum-Welch Algorithm(M-step).**

$$\text{maximize } Q = \mathbb{E}_{\mathbf{z}}[\log p(\mathbf{y}, \mathbf{z})]$$

$$\begin{aligned} Q &= \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{y}) \log p(\mathbf{y}, \mathbf{z}) \\ &= \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{y}) \left[ \log p(z_1) + \sum_{t=2}^T \log p(z_t|z_{t-1}) + \sum_{t=1}^T \log p(y_t|z_t) \right] \\ &= \sum_{z_1} p(z_1|\mathbf{y}) \log p(z_1) + \sum_{t=2}^T \sum_{z_{t-1}, z_t} p(z_{t-1}, z_t|\mathbf{y}) \log p(z_t|z_{t-1}) + \sum_{t=1}^T \sum_{z_t} p(z_t|\mathbf{y}) \log p(y_t|z_t) \\ &= \sum_{k=1}^d \gamma(z_{1,k}) \log \xi_i + \sum_{t=2}^T \sum_{j=1}^d \sum_{k=1}^d \eta(z_{t-1,j}, z_{t,k}) \log A_{jk} + \sum_{t=1}^T \sum_{k=1}^d \gamma(z_{t,k}) \log p(y_t|C_k) \end{aligned}$$

## Hidden Markov Models

---

- Q3: Learning of parameters of HMM (ML parameter estimate),  $\Omega$ .  
**The Baum-Welch Algorithm(M-step).**

$$Q = \sum_{k=1}^d \gamma(z_{1,k}) \log \xi_k + \sum_{t=2}^T \sum_{j=1}^d \sum_{k=1}^d \eta(z_{t-1,j}, z_{t,k}) \log A_{jk} + \sum_{t=1}^T \sum_{k=1}^d \gamma(z_{t,k}) \log p(y_t | C_k)$$

## Hidden Markov Models

---

- Q3: Learning of parameters of HMM (ML parameter estimate),  $\Omega$ .  
**The Baum-Welch Algorithm(M-step).**

$$Q = \sum_{k=1}^d \gamma(z_{1,k}) \log \xi_k + \sum_{t=2}^T \sum_{j=1}^d \sum_{k=1}^d \eta(z_{t-1,j}, z_{t,k}) \log A_{jk} + \sum_{t=1}^T \sum_{k=1}^d \gamma(z_{t,k}) \log p(y_t | C_k)$$

Maximization with respect to  $\xi_k$  and  $A_{jk}$  is easily achieved using Lagrange multipliers.

$$\xi_k = \frac{\gamma(z_{1,k})}{\sum_{j=1}^d \gamma(z_{1,j})} \quad A_{jk} = \frac{\sum_{t=2}^T \eta(z_{t-1,j}, z_{t,k})}{\sum_{l=1}^d \sum_{t=2}^T \eta(z_{t-1,j}, z_{t,l})}$$

## Hidden Markov Models

---

- Q3: Learning of parameters of HMM (ML parameter estimate),  $\Omega$ .  
**The Baum-Welch Algorithm(M-step).**

$$Q = \sum_{k=1}^d \gamma(z_{1,k}) \log \xi_k + \sum_{t=2}^T \sum_{j=1}^d \sum_{k=1}^d \eta(z_{t-1,j}, z_{t,k}) \log A_{jk} + \sum_{t=1}^T \sum_{k=1}^d \gamma(z_{t,k}) \log p(y_t | C_k)$$

Maximization with respect to  $C_k$ : when  $y_t$  is **discrete**.

Parameterization: 
$$p(y_t | C_k) = \prod_i^n \prod_k^d \mu_{i,k}^{y_{t,i} z_{t,k}}$$

Estimation: 
$$\mu_{i,k} = \frac{\sum_{t=1}^T \gamma(z_{t,k}) y_{t,i}}{\sum_{t=1}^T \gamma(z_{t,k})}$$

## Hidden Markov Models

---

- Q3: Learning of parameters of HMM (ML parameter estimate),  $\Omega$ .  
**The Baum-Welch Algorithm(M-step).**

$$Q = \sum_{k=1}^d \gamma(z_{1,k}) \log \xi_k + \sum_{t=2}^T \sum_{j=1}^d \sum_{k=1}^d \eta(z_{t-1,j}, z_{t,k}) \log A_{jk} + \sum_{t=1}^T \sum_{k=1}^d \gamma(z_{t,k}) \log p(y_t | C_k)$$

Maximization with respect to  $C_k$ : when  $y_t$  is **continuous**.

Parameterization: 
$$p(y_t | C_k) \sim \mathcal{N}(\mu_k, \Sigma_k)$$

Estimation: 
$$\mu_k = \frac{\sum_{t=1}^T \gamma(z_{t,k}) y_t}{\sum_{t=1}^T \gamma(z_{t,k})}$$

$$\Sigma_k = \frac{\sum_{t=1}^T \gamma(z_{t,k}) (y_t - \mu_k)(y_t - \mu_k)'}{\sum_{t=1}^T \gamma(z_{t,k})}$$

## Hidden Markov Models

---

- Q3: Learning of parameters of HMM (ML parameter estimate),  $\Omega$ .  
**The Baum-Welch Algorithm(E-step).**

Compute  $\gamma(z_{t,k}), \eta(z_{t-1,j}, z_{t,k})$  for  $t = 1, \dots, T$  and  $j, k = 1, \dots, d$ .

## Hidden Markov Models

---

- Q3: Learning of parameters of HMM (ML parameter estimate),  $\Omega$ .  
**The Baum-Welch Algorithm(E-step).**

Compute  $\gamma(z_{t,k}), \eta(z_{t-1,j}, z_{t,k})$  for  $t = 1, \dots, T$  and  $j, k = 1, \dots, d$ .

$$\gamma(z_{t,k}) = \frac{\alpha_t(k)\beta_t(k)}{p(\mathbf{y})} \propto \alpha_t(k)\beta_t(k)$$

$$\begin{aligned} \eta(z_{t-1,j}, z_{t,k}) &= \frac{\alpha_{t-1}(j)\beta_t(k)A_{jk}p(y_t|C_k)}{p(\mathbf{y})} \\ &\propto \alpha_{t-1}(j)\beta_t(k)A_{jk}p(y_t|C_k) \end{aligned}$$

$$\alpha_t(i) \equiv p(y_1, y_2, \dots, y_t, z_t = i)$$

$$\beta_t(i) \equiv p(y_{t+1}, y_{t+2}, \dots, y_T | z_t = i)$$

## Hidden Markov Models

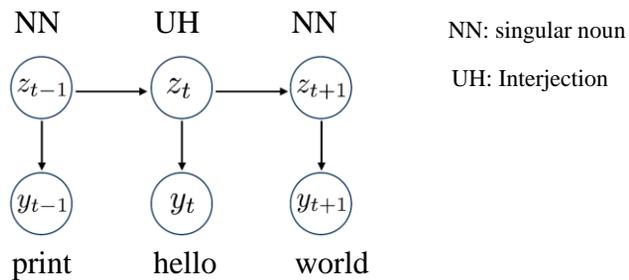
---

- Applications.
  - Speech recognition
  - Natural Language Processing
  - Bio-sequence analysis

## Hidden Markov Models

---

- Applications.
  - Speech recognition
  - Natural Language Processing
  - Bio-sequence analysis



Tagging results from <http://cogcomp.cs.illinois.edu/demo/pos/>

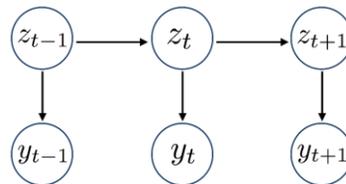
# Agenda

- Introduction
- Markov Models
- Hidden Markov Models
  - Inference
  - Learning
- **Linear Dynamical Systems**
  - Inference
  - Learning
- Recent Work

## Linear Dynamical Systems

- Overview

- A sequence of length  $T$
- Observations  $\{y_t\}$ : **continuous**
- Hidden states  $\{z_t\}$ : **continuous**



- Defined by two linear equations.

$$z_t = Az_{t-1} + \zeta_t \quad \zeta_t \sim \mathcal{N}(0, Q)$$

$$y_t = Cz_t + \epsilon_t \quad \epsilon_t \sim \mathcal{N}(0, R)$$

$$z_1 \sim \mathcal{N}(\xi, \Psi)$$

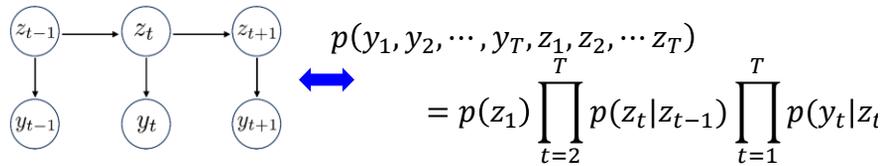
- $A, Q, \Psi$ :  $d \times d$  matrices.

- $C$ : an  $n \times d$  matrix.

- $R$  is an  $n \times n$  matrix.

## Linear Dynamical Systems

---



- What we can do if we have the model ? – **Inference & Prediction**
  - Q1: Given an observation sequence up to time  $t$  and the model, compute the probability of the current hidden state  $z_t$ . (**Kalman filtering**)
  - Q2: Given the observation sequence and the model, compute the probability of hidden state  $z_T$ , where  $T > t$ . (**Kalman smoothing**)
  - Q3: Given the model, compute the expected values at future timestamps.
- How can we get the model ?- **Learning**
  - Q4: Learning of parameters of LDS(ML parameter estimate).

## Linear Dynamical Systems

---

- Notations
  - Let  $\hat{z}_{t|t-1} \equiv \mathbb{E}[z_t | y_1, \dots, y_{t-1}]$  be the **priori** estimation.
  - Let  $\hat{z}_{t|t} \equiv \mathbb{E}[z_t | y_1, \dots, y_{t-1}, y_t]$  be the **posteriori** estimation.
  - Let  $P_{t|t-1} \equiv \mathbb{E}[(z_t - \hat{z}_{t|t-1})(z_t - \hat{z}_{t|t-1})']$  be the **priori** estimate error covariance.
  - Let  $P_{t|t} \equiv \mathbb{E}[(z_t - \hat{z}_{t|t})(z_t - \hat{z}_{t|t})']$  be the **posteriori** estimate error covariance.
  - Let  $\hat{z}_t \equiv \hat{z}_{t|T} \equiv \mathbb{E}[z_t | \mathbf{y}]$ .
  - Let  $M_t \equiv M_{t|T} \equiv \mathbb{E}[z_t z_t' | \mathbf{y}]$ .
  - Let  $M_{t,t-1} \equiv M_{t,t-1|T} \equiv \mathbb{E}[z_t z_{t-1}' | \mathbf{y}]$ .
  - Let  $P_{t|T} \equiv \mathbb{V}\mathbb{A}\mathbb{R}[z_t | \mathbf{y}]$ .
  - Let  $P_{t,t-1|T} \equiv \mathbb{V}\mathbb{A}\mathbb{R}[z_t z_{t-1}' | \mathbf{y}]$ .
  - Let  $\Omega$  be the entire LDS model, i.e.,  $\Omega = \{\xi, \Psi, A, C, Q, R\}$ .

## Linear Dynamical Systems

---

- Q1: Given an observation sequence up to time  $t$  and the model, compute the probability of the current hidden state  $z_t$ . (**Kalman filtering**)

$$\mathbb{E}[z_t | \{y_1, y_2, \dots, y_t\}]$$

Kalman filtering is a **forward recursive** algorithm, as follows:

```
// Time Update:
 $\hat{\mathbf{z}}_{t|t-1} = A\hat{\mathbf{z}}_{t-1|t-1}$ 
 $P_{t|t-1} = AP_{t-1|t-1}A' + Q$ 
// Measure Update:
 $K_t = P_{t|t-1}C'(CP_{t|t-1}C' + R)^{-1}$ 
 $\hat{\mathbf{z}}_{t|t} = \hat{\mathbf{z}}_{t|t-1} + K_t(\mathbf{y}_t - C\hat{\mathbf{z}}_{t|t-1})$ 
 $P_{t|t} = P_{t|t-1} - K_tCP_{t|t-1}$ 
```

## Linear Dynamical Systems

---

- Q1: Given an observation sequence up to time  $t$  and the model, compute the probability of the current hidden state  $z_t$ . (**Kalman filtering**)

$$\mathbb{E}[z_t | \{y_1, y_2, \dots, y_t\}]$$

Kalman filtering is a **forward recursive** algorithm, as follows:

```
for t = 2 → T do
  // Time Update:
   $\hat{\mathbf{z}}_{t|t-1} = A\hat{\mathbf{z}}_{t-1|t-1}$ 
   $P_{t|t-1} = AP_{t-1|t-1}A' + Q$ 
  // Measure Update:
   $K_t = P_{t|t-1}C'(CP_{t|t-1}C' + R)^{-1}$ 
   $\hat{\mathbf{z}}_{t|t} = \hat{\mathbf{z}}_{t|t-1} + K_t(\mathbf{y}_t - C\hat{\mathbf{z}}_{t|t-1})$ 
   $P_{t|t} = P_{t|t-1} - K_tCP_{t|t-1}$ 
end for
```

**Prior estimate** →  $\hat{\mathbf{z}}_{t|t-1}$

**Some weight (Kalman Gain)** →  $K_t$

**Error of prior estimate** →  $(\mathbf{y}_t - C\hat{\mathbf{z}}_{t|t-1})$

## Linear Dynamical Systems

- Q2: Given the observation sequence and the model, compute the probability of hidden state  $z_T$ , where  $T > t$ . (**Kalman Smoothing**)

$$\mathbb{E}[z_t | \mathbf{y}]$$

Kalman smoothing is a **backward recursive** algorithm, as follows:

```

for t = T-1 → 1 do
   $M_{t|T} = P_{t|T} + \hat{\mathbf{z}}_{t|T} \hat{\mathbf{z}}_{t|T}'$ 
   $J_{t-1} = P_{t-1|t-1} A' (P_{t|t-1})^{-1}$ 
   $P_{t,t-1|T} = P_{t|t} J_{t-1}' + J_t (P_{t+1,t|T} - A P_{t|t}) J_{t-1}'$ 
   $M_{t,t-1|T} = P_{t,t-1|T} + \hat{\mathbf{z}}_{t|T} \hat{\mathbf{z}}_{t-1|T}'$ 
   $\hat{\mathbf{z}}_{t-1|T} = \hat{\mathbf{z}}_{t-1|t-1} + J_{t-1} (\hat{\mathbf{z}}_{t|T} - A \hat{\mathbf{z}}_{t-1|t-1})$ 
   $P_{t-1|T} = P_{t-1|t-1} + J_{t-1} (P_{t|T} - P_{t|t-1}) J_{t-1}'$ 
end for

```

## Linear Dynamical Systems

- Q2: Given the observation sequence and the model, compute the probability of hidden state  $z_T$ , where  $T > t$ . (**Kalman Smoothing**)

$$\mathbb{E}[z_t | \mathbf{y}]$$

Kalman smoothing is a **backward recursive** algorithm, as follows:

```

for t = T-1 → 1 do
   $M_{t|T} = P_{t|T} + \hat{\mathbf{z}}_{t|T} \hat{\mathbf{z}}_{t|T}'$ 
   $J_{t-1} = P_{t-1|t-1} A' (P_{t|t-1})^{-1}$ 
   $P_{t,t-1|T} = P_{t|t} J_{t-1}' + J_t (P_{t+1,t|T} - A P_{t|t}) J_{t-1}'$ 
   $M_{t,t-1|T} = P_{t,t-1|T} + \hat{\mathbf{z}}_{t|T} \hat{\mathbf{z}}_{t-1|T}'$ 
   $\hat{\mathbf{z}}_{t-1|T} = \hat{\mathbf{z}}_{t-1|t-1} + J_{t-1} (\hat{\mathbf{z}}_{t|T} - A \hat{\mathbf{z}}_{t-1|t-1})$ 
   $P_{t-1|T} = P_{t-1|t-1} + J_{t-1} (P_{t|T} - P_{t|t-1}) J_{t-1}'$ 
end for

```

Posteriori estimate up to time t-1 (points to  $\hat{\mathbf{z}}_{t-1|T}$ )  
Weight (points to  $J_{t-1}$ )  
Error (points to  $(\hat{\mathbf{z}}_{t|T} - A \hat{\mathbf{z}}_{t-1|t-1})$ )

## Linear Dynamical Systems

---

- Q3: Given the model, compute the expected values at future timestamps. (**Prediction**)

$$\hat{y}_{t'}, \text{ where } t' > T \quad \begin{array}{ll} z_t = Az_{t-1} + \zeta_t & \zeta_t \sim \mathcal{N}(0, Q) \\ y_t = Cz_t + \epsilon_t & \epsilon_t \sim \mathcal{N}(0, R) \\ & z_1 \sim \mathcal{N}(\xi, \Psi) \end{array}$$

$$\hat{y}_{T+1} = C\hat{z}_{T+1} = CA\hat{z}_T$$

$$\hat{y}_{T+2} = C\hat{z}_{T+2} = CA\hat{z}_{T+1} = CA^2\hat{z}_T$$

$$\vdots$$

$$\hat{y}_{t'} = C\hat{z}_{t'} = \dots = CA^{t'-T}\hat{z}_T$$

## Linear Dynamical Systems

---

- Q3: Given the model, compute the expected values at future timestamps. (**Prediction**)

$$\hat{y}_{t'}, \text{ where } t' > T \quad \begin{array}{ll} z_t = Az_{t-1} + \zeta_t & \zeta_t \sim \mathcal{N}(0, Q) \\ y_t = Cz_t + \epsilon_t & \epsilon_t \sim \mathcal{N}(0, R) \\ & z_1 \sim \mathcal{N}(\xi, \Psi) \end{array}$$

$$\hat{y}_{T+1} = C\hat{z}_{T+1} = CA\hat{z}_T$$

$$\hat{y}_{T+2} = C\hat{z}_{T+2} = CA\hat{z}_{T+1} = CA^2\hat{z}_T$$

$$\vdots$$

$$\hat{y}_{t'} = C\hat{z}_{t'} = \dots = CA^{t'-T}\hat{z}_T$$

**How can we get  $\hat{z}_T$  ?**

## Linear Dynamical Systems

---

- Q3: Given the model, compute the expected values at future timestamps. (**Prediction**)

$$\hat{y}_{t'}, \text{ where } t' > T \quad \begin{array}{ll} z_t = Az_{t-1} + \zeta_t & \zeta_t \sim \mathcal{N}(0, Q) \\ y_t = Cz_t + \epsilon_t & \epsilon_t \sim \mathcal{N}(0, R) \\ & z_1 \sim \mathcal{N}(\xi, \Psi) \end{array}$$

$$\begin{aligned} \hat{y}_{T+1} &= C\hat{z}_{T+1} = CA\hat{z}_T \\ \hat{y}_{T+2} &= C\hat{z}_{T+2} = CA\hat{z}_{T+1} = CA^2\hat{z}_T \\ &\vdots \\ \hat{y}_{t'} &= C\hat{z}_{t'} = \dots = CA^{t'-T}\hat{z}_T \end{aligned}$$

**How can we get  $\hat{z}_T$  ?**  
**Kalman filtering !**

## Linear Dynamical Systems

---

- Q4: Learning of parameters of LDS (ML parameter estimate),  $\Omega$ .  
**The EM Algorithm (M-step).**

$$\text{maximize } Q = \mathbb{E}_{\mathbf{z}}[\log p(\mathbf{y}, \mathbf{z})] \quad \begin{array}{ll} z_t = Az_{t-1} + \zeta_t & \zeta_t \sim \mathcal{N}(0, Q) \\ y_t = Cz_t + \epsilon_t & \epsilon_t \sim \mathcal{N}(0, R) \\ & z_1 \sim \mathcal{N}(\xi, \Psi) \end{array}$$

$$\log p(\mathbf{y}, \mathbf{z}) = \log p(z_1) + \sum_{t=2}^T \log p(z_t | z_{t-1}) + \sum_{t=1}^T \log p(y_t | z_t)$$

$$p(y_t | z_t) = \exp\left\{-\frac{1}{2}(y_t - Cz_t)'R^{-1}(y_t - Cz_t)\right\} (2\pi)^{-n/2} |R|^{-1/2}$$

$$p(z_t | z_{t-1}) = \exp\left\{-\frac{1}{2}(z_t - Az_{t-1})'Q^{-1}(z_t - Az_{t-1})\right\} (2\pi)^{-d/2} |Q|^{-1/2}$$

$$p(z_1) = \exp\left\{-\frac{1}{2}(z_1 - \xi)' \Psi^{-1}(z_1 - \xi)\right\} (2\pi)^{-d/2} |\Psi|^{-1/2}$$

## Linear Dynamical Systems

- Q4: Learning of parameters of LDS (ML parameter estimate),  $\Omega$ .

### The EM Algorithm (M-step).

$$\begin{aligned} \text{maximize } Q &= \mathbb{E}_{\mathbf{z}}[\log p(\mathbf{y}, \mathbf{z})] \\ z_t &= Az_{t-1} + \zeta_t \quad \zeta_t \sim \mathcal{N}(0, Q) \\ y_t &= Cz_t + \epsilon_t \quad \epsilon_t \sim \mathcal{N}(0, R) \\ z_1 &\sim \mathcal{N}(\xi, \Psi) \end{aligned}$$

$$\begin{aligned} \log p(\mathbf{y}, \mathbf{z}) &= \log p(z_1) + \sum_{t=2}^T \log p(z_t | z_{t-1}) + \sum_{t=1}^T \log p(y_t | z_t) \\ &= -\sum_{t=1}^T \left( \frac{1}{2} (y_t - Cz_t)' R^{-1} (y_t - Cz_t) \right) - \frac{T}{2} \log |R| \\ &\quad - \sum_{t=2}^T \left( \frac{1}{2} (z_t - Az_{t-1})' Q^{-1} (z_t - Az_{t-1}) \right) - \frac{T-1}{2} \log |Q| \\ &\quad - \frac{1}{2} (z_1 - \xi)' \Psi^{-1} (z_1 - \xi) - \frac{1}{2} \log |\Psi| - \frac{T(n+d)}{2} \log 2\pi \end{aligned}$$

## Linear Dynamical Systems

- Q4: Learning of parameters of LDS (ML parameter estimate),  $\Omega$ .

### The EM Algorithm (M-step).

$$\begin{aligned} \text{maximize } Q &= \mathbb{E}_{\mathbf{z}}[\log p(\mathbf{y}, \mathbf{z})] \\ \log p(\mathbf{y}, \mathbf{z}) &= -\sum_{t=1}^T \left( \frac{1}{2} (y_t - Cz_t)' R^{-1} (y_t - Cz_t) \right) - \frac{T}{2} \log |R| \\ &\quad - \sum_{t=2}^T \left( \frac{1}{2} (z_t - Az_{t-1})' Q^{-1} (z_t - Az_{t-1}) \right) - \frac{T-1}{2} \log |Q| \\ &\quad - \frac{1}{2} (z_1 - \xi)' \Psi^{-1} (z_1 - \xi) - \frac{1}{2} \log |\Psi| - \frac{T(n+d)}{2} \log 2\pi \end{aligned}$$

Update rules:

$$C = \left( \sum_{t=1}^T y_t \hat{z}_t \right) \left( \sum_{t=1}^T M_t \right)^{-1} \quad R = \frac{1}{T} \left( \sum_{t=1}^T y_t y_t' - C \hat{z}_t y_t' \right)$$

$$A = \left( \sum_{t=2}^T M_{t,t-1} \right) \left( \sum_{t=2}^T M_{t-1} \right)^{-1} \quad Q = \frac{1}{T-1} \left( \sum_{t=2}^T M_t - A \sum_{t=2}^T M_{t-1} \right)$$

$$\xi = \hat{z}_1 \quad \Psi = M_1 - \hat{z}_1 \hat{z}_1'$$

Notations:

- $\hat{z}_t \equiv \mathbb{E}[z_t | \mathbf{y}]$ .
- $M_t \equiv \mathbb{E}[z_t z_t' | \mathbf{y}]$ .
- $M_{t,t-1} \equiv \mathbb{E}[z_t z_{t-1}' | \mathbf{y}]$ .

## Linear Dynamical Systems

---

- Q4: Learning of parameters of LDS (ML parameter estimate),  $\Omega$ .  
**The EM Algorithm (E-step).**

In the E-step, we will compute the following sufficient statistics, which are used in M-step.

- $\hat{z}_t \equiv \mathbb{E}[z_t | \mathbf{y}]$ .
- $M_t \equiv \mathbb{E}[z_t z_t' | \mathbf{y}]$ .
- $M_{t,t-1} \equiv \mathbb{E}[z_t z_{t-1}' | \mathbf{y}]$ .

$\hat{z}_t, M_t, M_{t,t-1}$  can be computed **recursively** by **Kalman smoothing**, for  $t = 1, 2, \dots, T$ .

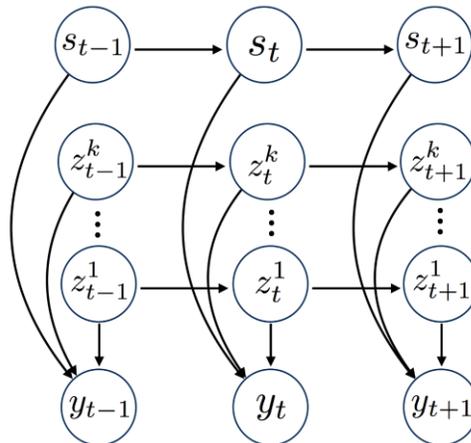
## Agenda

- Introduction
- Markov Models
- Hidden Markov Models
  - Inference
  - Learning
- Linear Dynamical Systems
  - Inference
  - Learning
- Recent Work

## Recent work

---

- Factorial HMM and Switching LDS. [ML'97] [NIPS'09] [JMLR'06]



## Recent work

---

- Spectral learning for HMM and LDS. [NIPS'07] [CSC'12] [MLKDD'11]

$$\begin{bmatrix} y_1 & y_2 & y_3 & \cdots & y_\tau \\ y_2 & y_3 & y_4 & \cdots & y_{\tau+1} \\ y_3 & y_4 & y_5 & \cdots & y_{\tau+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_d & y_{d+1} & y_{d+2} & \cdots & y_{d+\tau-1} \end{bmatrix}_{m \times \tau}$$

Hankel matrix

- Identification of hidden state space dimensionality. [ANN APPL STAT'13] [AISTAT'10] [ICML'10]

## Reference

---

- [1] Bishop, Christopher M. *Pattern recognition and machine learning*. Vol. 1. New York: springer, 2006.
- [2] Ghahramani, Zoubin, and Geoffrey E. Hinton. *Parameter estimation for linear dynamical systems*. Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science, 1996.
- [3] Ghahramani, Zoubin, and Michael I. Jordan. "Factorial hidden Markov models." *Machine learning* 29.2-3 (1997): 245-273.
- [4] Willsky, Alan S., et al. "Nonparametric Bayesian learning of switching linear dynamical systems." *Advances in Neural Information Processing Systems*. 2009.
- [5] Barber, David. "Expectation correction for smoothed inference in switching linear dynamical systems." *The Journal of Machine Learning Research* 7 (2006): 2515-2540.
- [6] Hsu, Daniel, Sham M. Kakade, and Tong Zhang. "A spectral algorithm for learning hidden Markov models." *Journal of Computer and System Sciences* 78.5 (2012): 1460-1480.
- [7] Balle, Borja, Ariadna Quattoni, and Xavier Carreras. "A spectral learning algorithm for finite state transducers." *Machine Learning and Knowledge Discovery in Databases*. Springer Berlin Heidelberg, 2011. 156-171.
- [8] Städler, Nicolas, and Sach Mukherjee. "Penalized estimation in high-dimensional hidden Markov models with state-specific graphical models." *The Annals of Applied Statistics* 7.4 (2013): 2157-2179.
- [9] Siddiqi, Sajid M., Byron Boots, and Geoffrey J. Gordon. "Reduced-Rank Hidden Markov Models." *International Conference on Artificial Intelligence and Statistics*. 2010.
- [10] Song, Le, et al. "Hilbert space embeddings of hidden Markov models." *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010.

## Q&A

---

Thank you

Oct. 09 2014