

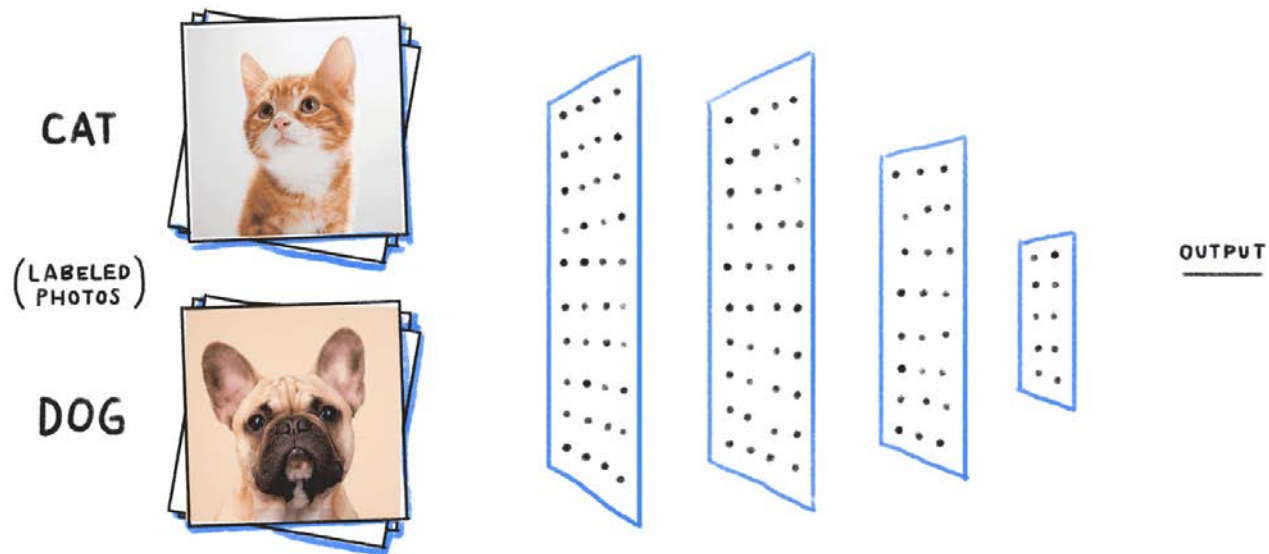


Neural Network Architectures for Time-Series

Industrial AI Lab.
Prof. Seungchul Lee

So Far

- Regression, Classification, Dimension Reduction,
- Based on snapshot-type data



Robocup 2011



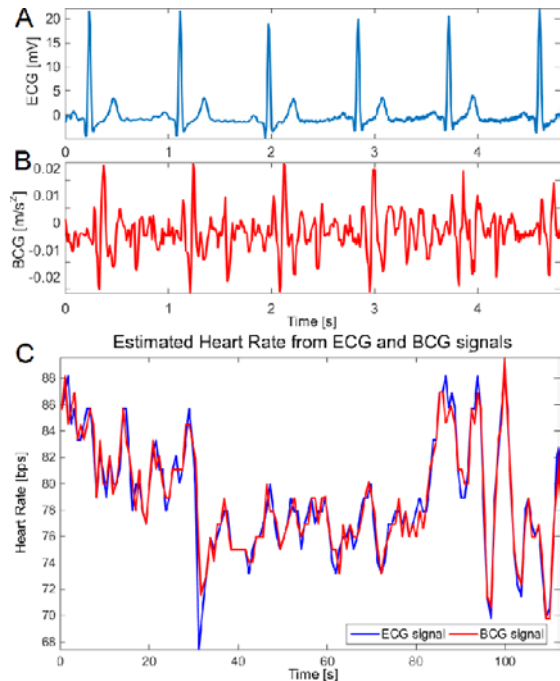
Sequence Matters



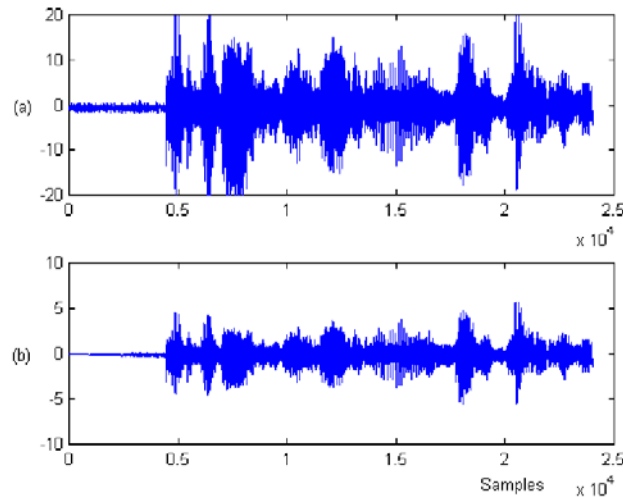
What is a Sequence ?

- Sentence
 - “This morning I took the dog for a walk.”

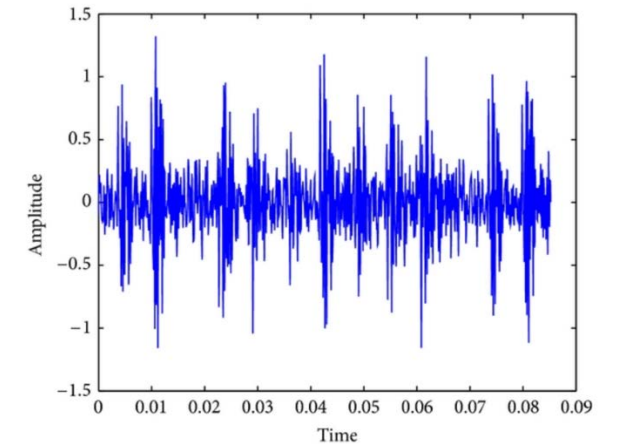
- Medical signals



- Speech waveform



- Vibration measurement



Sequence Modeling

- Most of the real-world data is time-series
- There are important bits to be considered
 - Past events
 - Relationship between events
 - Causality
 - Credit assignment
 - Learning the structure and hierarchy
- Use the past and present observations to predict the future



(Deterministic) Time Series Data

- For example

$$y[0] = 1, \quad y[1] = \frac{1}{2}, \quad y[2] = \frac{1}{4}, \quad \dots$$

- Closed-form

$$y[n] = \left(\frac{1}{2}\right)^n, \quad n \geq 0$$

- Linear difference equation (LDE) and initial condition

$$y[n] = \frac{1}{2}y[n-1], \quad y[0] = 1$$

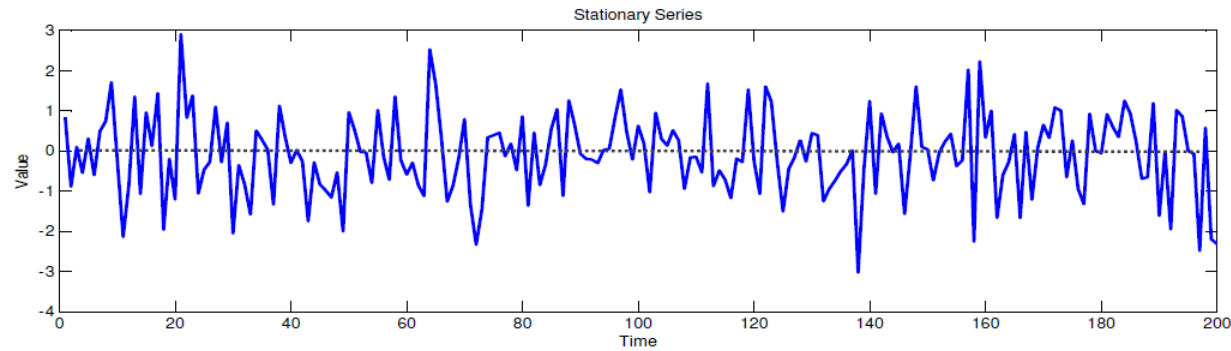
- High order LDEs

$$y[n] = \alpha_1 y[n-1] + \alpha_2 y[n-2]$$

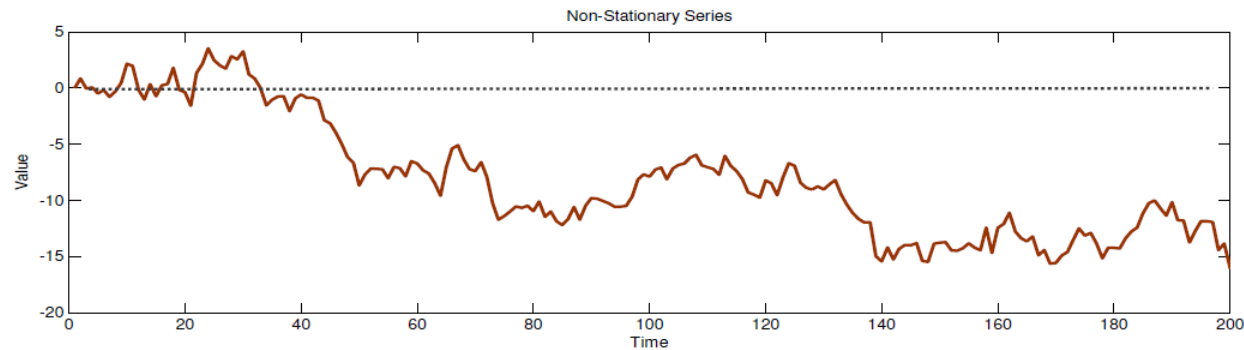
$$y[n] = \alpha_1 y[n-1] + \alpha_2 y[n-2] + \dots + \alpha_k y[n-k]$$

(Stochastic) Time Series Data

- Stationary

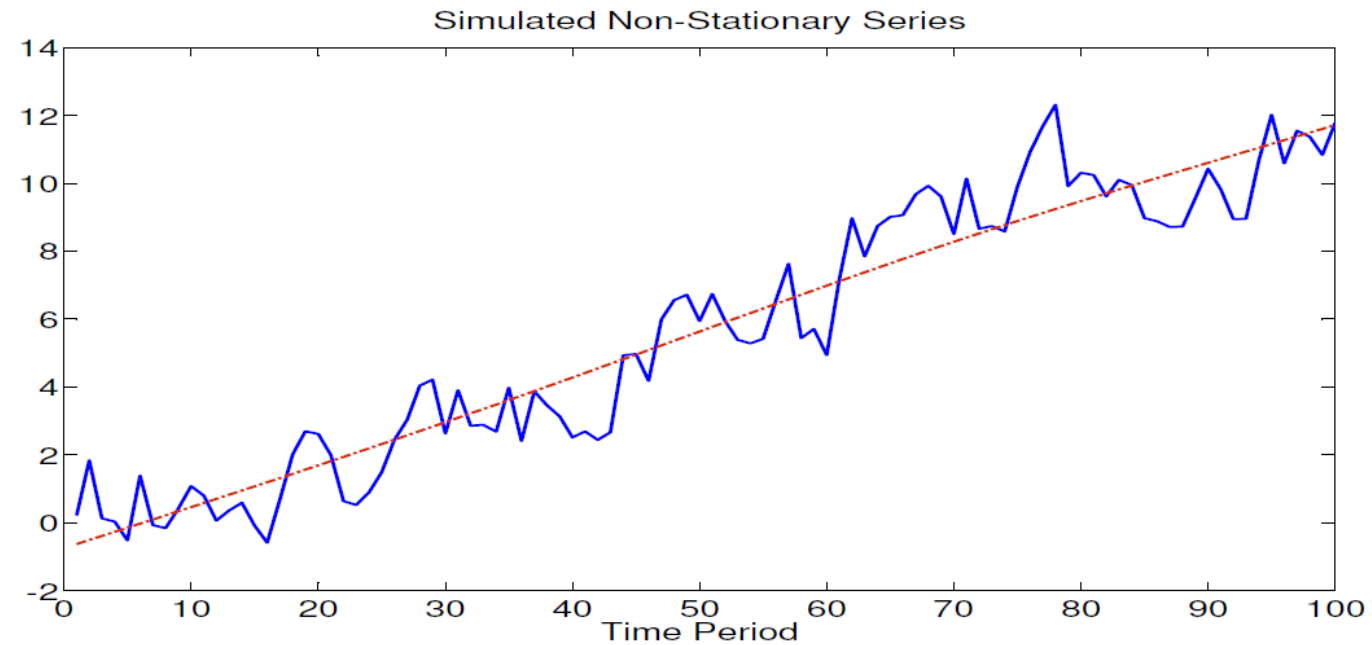


- Non-stationary
 - Mean and variance change over time



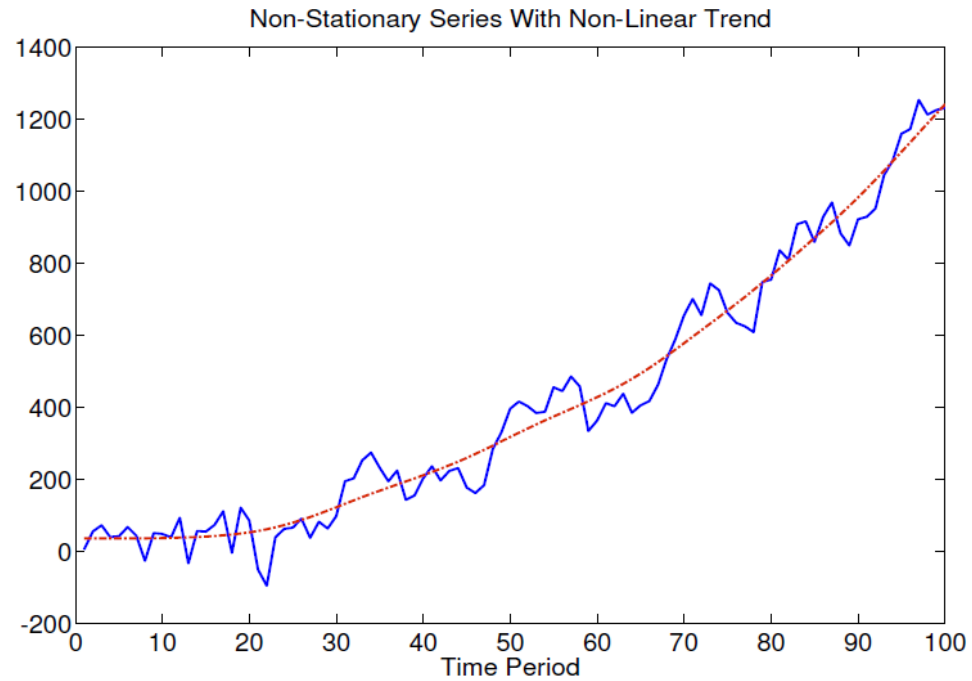
Dealing with Non-Stationarity

- Linear trends



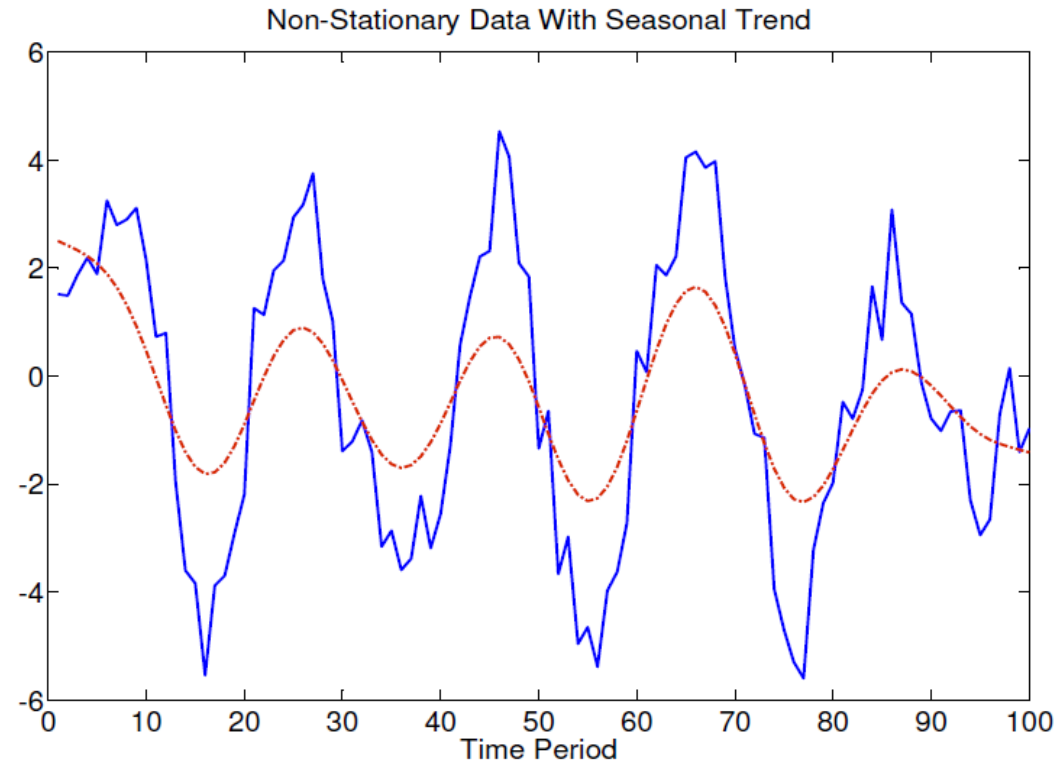
Dealing with Non-Stationarity

- Non-linear trends



Dealing with Non-Stationarity

- Seasonal trends



Dealing with Non-Stationarity

- Model assumption

$$\begin{aligned} Y_t = & \beta_1 + \beta_2 Y_{t-1} \\ & + \beta_3 t + \beta_4 t^{\beta_5} \\ & + \beta_6 \sin \frac{2\pi}{s} t + \beta_7 \cos \frac{2\pi}{s} t \\ & + u_t \end{aligned}$$

Markov Process

Sequential Processes

- Most classifiers ignored the **sequential** aspects of data
- Consider a system which can occupy one of N discrete states or categories

$$q_t \in \{S_1, S_2, \dots, S_N\}$$

- We are interested in **stochastic** systems, in which state evolution is **random**
- Any **joint** distribution can be factored into a series of **conditional** distributions

$$p(q_0, q_1, \dots, q_T) = p(q_0) p(q_1 \mid q_0) p(q_2 \mid q_1 q_0) p(q_3 \mid q_2 q_1 q_0) \dots$$

Almost impossible to compute !

Markov Chain

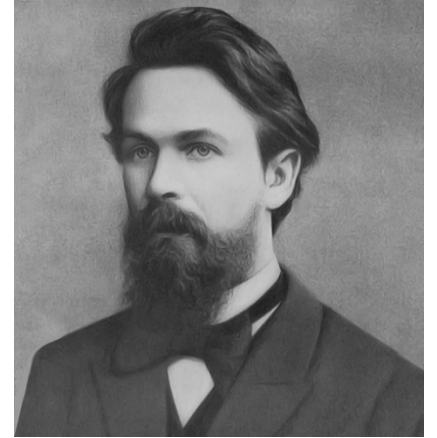
- Markovian property (assumption)
 - Information state: sufficient statistic of history

$$p(q_{t+1} \mid q_t, \dots, q_0) = p(q_{t+1} \mid q_t)$$

- Tractable in computation of joint distribution

$$\begin{aligned} p(q_0, q_1, \dots, q_T) &= p(q_0) p(q_1 \mid q_0) p(q_2 \mid q_1, q_0) p(q_3 \mid q_2, q_1, q_0) \cdots \\ &= p(q_0) p(q_1 \mid q_0) p(q_2 \mid q_1) p(q_3 \mid q_2) \cdots \end{aligned}$$

- Future is independent of past given present
 - Rain \rightarrow snow \rightarrow sunny \rightarrow sunny \rightarrow sunny \rightarrow rain \rightarrow snow \rightarrow ??
 - ~~– Rain \rightarrow snow \rightarrow sunny \rightarrow sunny \rightarrow sunny \rightarrow rain \rightarrow snow \rightarrow ??~~



State Transition Matrix

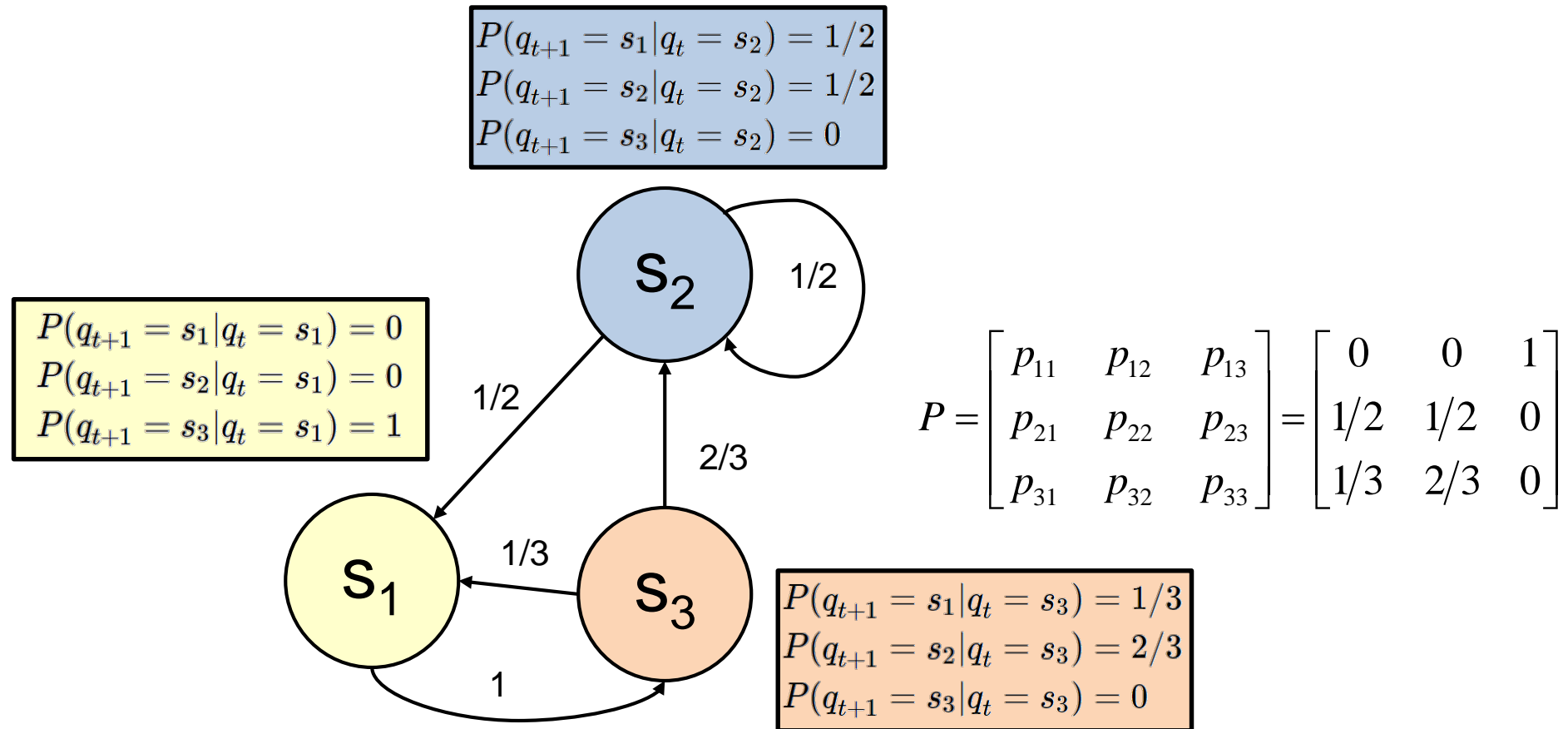
- For a Markov state s and successor state s' , the state transition probability is defined by

$$\mathcal{P}_{ss'} = \mathbb{P} [S_{t+1} = s' \mid S_t = s]$$

- State transition matrix P defines transition probabilities from all states s to all successor states s' ,

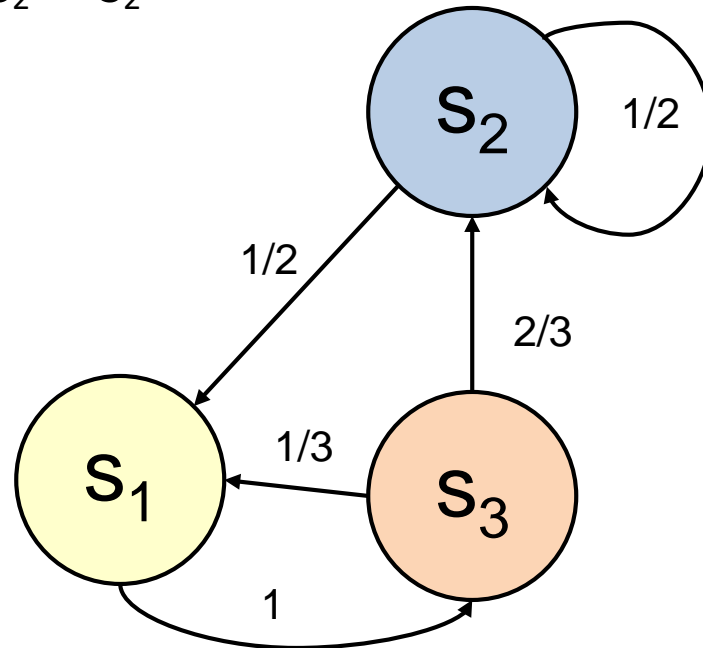
$$\mathcal{P} = \begin{matrix} & \text{to} \\ \text{from} & \begin{bmatrix} \mathcal{P}_{11} & \dots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{n1} & \dots & \mathcal{P}_{nn} \end{bmatrix} \end{matrix}$$

State Transition Matrix



Example: MC Episodes

- Sample episodes starting from S_1
 - $S_1 \rightarrow S_3 \rightarrow S_2 \rightarrow S_2 \rightarrow S_1 \rightarrow \dots$
 - $S_1 \rightarrow S_3 \rightarrow S_1 \rightarrow S_3 \rightarrow S_2 \rightarrow \dots$
 - $S_1 \rightarrow S_3 \rightarrow S_2 \rightarrow S_2 \rightarrow S_2 \rightarrow \dots$

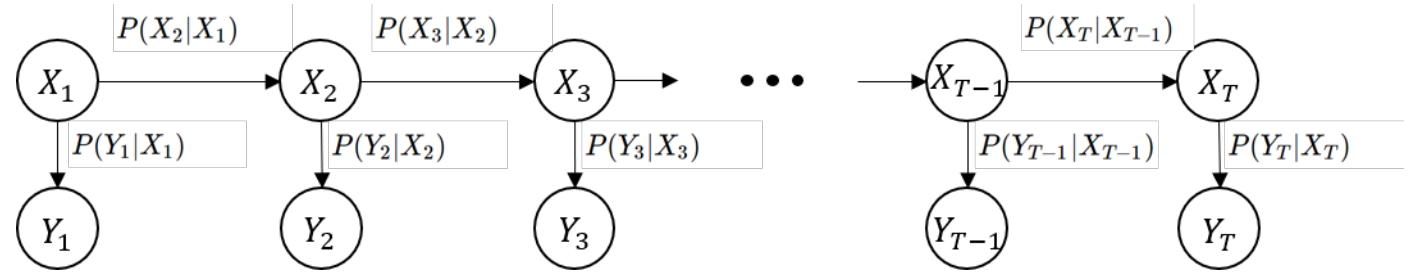


- Generate passive stochastic sequence

Hidden Markov Model

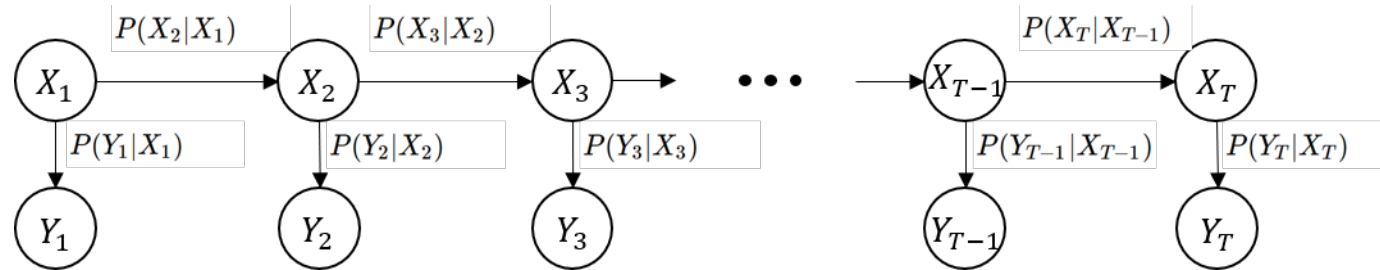
Hidden Markov Models

- Discrete state-space model
 - Used in speech recognition
 - **State** representation is simple
 - Hard to scale-up the training
- Assumption
 - We can observe something that's affected by the true state
 - Natural way of thinking
- Limited sensors (incomplete state information)
 - But still partially related
- Noisy sensors
 - Unreliable



Hidden Markov Model (HMM)

- True state (or hidden variable) follows Markov chain
- Observation emitted from state
 - Y_t is noisily determined depending on the current state X_t



- Forward: sequence of observations can be generated
- Question: state estimation

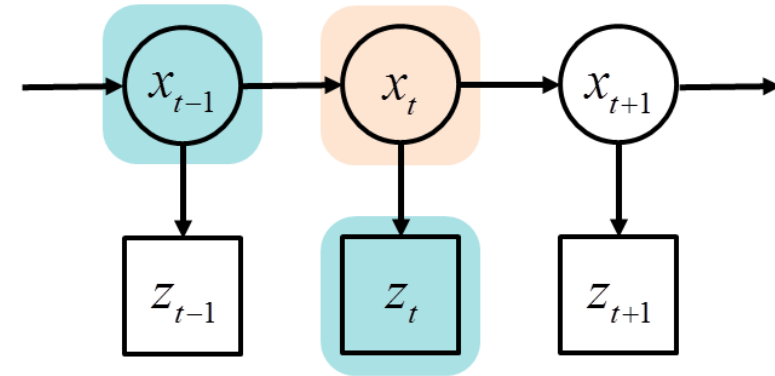
$$P(X_T = s_i \mid Y_1 Y_2 \cdots Y_T)$$

- HMM can do this, but with many difficulties

Kalman Filter

Kalman Filter

- Linear dynamical system of motion
- A, B, C ?
- Continuous State space model
 - For filtering and control applications
 - Linear-Gaussian state space model
 - Widely used in many applications:
 - GPS, weather systems, etc.
- Weakness
 - Linear state space model assumed
 - Difficult to apply to highly non-linear domains



$$\begin{aligned}x_{t+1} &= Ax_t + Bu_t \\ z_t &= Cx_t\end{aligned}$$

Kalman Filter

