RLearning:
Short guides to reinforcement learning

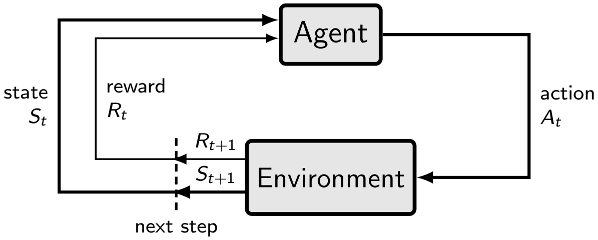
Unit 2-1: Markov Processes

Davud Rostam-Afschar (Uni Mannheim)

How to predict transitions?

Markov Chains

Unrolling the Problem



Goal: Learn to choose actions that maximize rewards

Unrolling the Problem

- Modeling environment dynamics
- Unrolling the control loop leads to a sequence of states, actions and rewards:

$$s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, \ldots$$

- This sequence forms a stochastic process (due to some uncertainty in the dynamics of the process)

Common Properties

- Processes are rarely arbitrary
- They often exhibit some structure
  - Laws of the process do not change
  - Short history sufficient to predict future

Common Properties

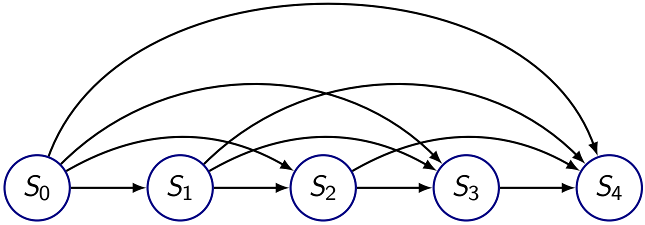
- Processes are rarely arbitrary
- They often exhibit some structure
  - Laws of the process do not change
  - Short history sufficient to predict future
- **Example:** weather prediction
  - Same model can be used everyday to predict weather
  - Weather measurements of past few days sufficient to predict weather

Common Properties

- Processes are rarely arbitrary
- They often exhibit some structure
  - Laws of the process do not change
  - Short history sufficient to predict future
- **Example:** weather prediction
  - Same model can be used everyday to predict weather
  - Weather measurements of past few days sufficient to predict weather
- **Example:** text prediction
  - Same model can be used in every conversation to predict next utterance
  - letter sequences of past texts sufficient to predict new sentences

Markovian and Stationary Processes

- Consider the sequence of states only
- Definition
  - Set of States: S
  - Stochastic dynamics: $\mathbb{P}(s_t | s_{t-1}, \ldots, s_0)$



Stochastic Process

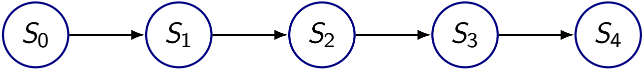
- Problem:
  - Infinitely large conditional distributions
- Solutions:
  - **Stationary process:**
    Dynamics do not change over time
  - **Markov assumption:**
    Current state depends only on a finite history of past states
  - **?**, Section 15.1

K-Order Markov Process

- Assumption: last k states sufficient

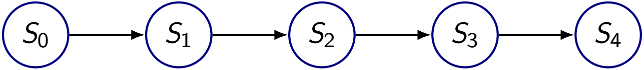
K-Order Markov Process

- Assumption: last *k* states sufficient
- First-order Markov Process
  - $\mathbb{P}(s_t|s_{t-1}, \ldots, s_0) = \mathbb{P}(s_t|s_{t-1})$

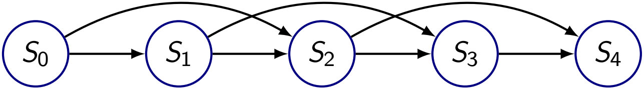


K-Order Markov Process

- Assumption: last k states sufficient
- First-order Markov Process
  - $\mathbb{P}(s_t | s_{t-1}, \dots, s_0) = \mathbb{P}(s_t | s_{t-1})$



- Second-order Markov Process
  - $\mathbb{P}(s_t | s_{t-1}, \dots, s_0) = \mathbb{P}(s_t | s_{t-1}, s_{t-2})$



Markov Process

- Commonly, a Markov Process refers to a
  - First-order process

$$\mathbb{P}\left(s_t \mid s_{t-1}, s_{t-2}, \ldots, s_0\right) = \mathbb{P}\left(s_t \mid s_{t-1}\right) \forall t$$

  - Stationary process

$$\mathbb{P}\left(s_t \mid s_{t-1}\right) = \mathbb{P}\left(s_{t'} \mid s_{t'-1}\right) \forall t'$$

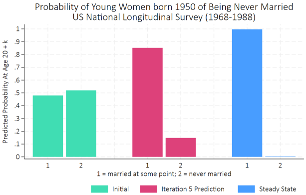
- **Advantage:**
  can specify the entire process with a single concise conditional distribution

$$\mathbb{P}\left(s' \mid s\right)$$

Examples

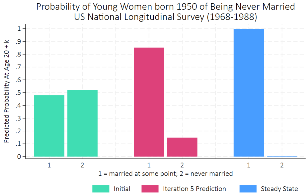
Examples

- Marrying decision of young women
  - **States:** relationship history
  - **Dynamics:** age

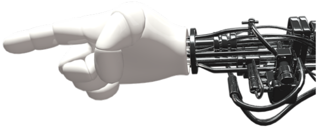


Examples

- Marrying decision of young women
  - **States:** relationship history
  - **Dynamics:** age

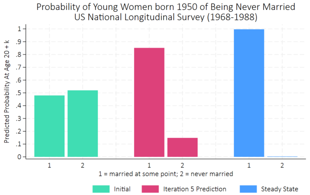


- Robotic control
  - **States:** $\langle x, y, z, \theta \rangle$ coordinates of joints
  - **Dynamics:** constant motion

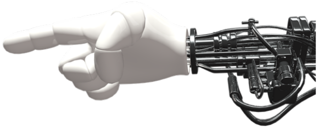


Examples

- Marrying decision of young women
  - **States:** relationship history
  - **Dynamics:** age



- Robotic control
  - **States:** $\langle x, y, z, \theta \rangle$ coordinates of joints
  - **Dynamics:** constant motion



- Inventory management
  - **States:** inventory level
  - **Dynamics:** constant (stochastic) demand

- Common task is prediction: $\mathbb{P}\left(s_{t+k} \mid s_t\right)$
- Computation:

$$\mathbb{P}\left(s_{t+k} \mid s_t\right) = \sum s_{t+k} \ldots s_{t+k-1} \prod_{i=1}^{k} \mathbb{P}\left(s_{t+i} \mid s_{t+i-1}\right)$$

- Discrete states (matrix operations):
  - Let T be a $|S| \times |S|$ matrix representing $\mathbb{P}\left(s_{t+k} \mid s_t\right)$
  - Then $\mathbb{P}\left(s_{t+k} \mid s_t\right) = T^k$
  - Complexity: $\mathcal{O}\left(k|S|^3\right)$

Example: Marrying as 2-State Markov Process

Setup: Initial distribution $p_t = \left[\, 0.5_{\text{never married}} \;\; 0.5_{\text{married}} \,\right]$

Example: Marrying as 2-State Markov Process

Setup: Initial distribution $p_t = \left[\, 0.5_{\text{never married}} \;\; 0.5_{\text{married}} \,\right]$

$T =$

	never married	married
never married	0.5	0.5
married	0	1

Example: Marrying as 2-State Markov Process

Setup: Initial distribution $p_t = \left[\, 0.5_{\text{never married}} \;\; 0.5_{\text{married}} \,\right]$

$T =$

	never married	married
never married	0.5	0.5
married	0	1

$\rightarrow \quad T^2 = \begin{pmatrix} 0.25 & 0.75 \\ 0 & 1 \end{pmatrix}, \ldots$

Example: Marrying as 2-State Markov Process

Setup: Initial distribution $p_t = \left[\, 0.5_{\text{never married}} \;\; 0.5_{\text{married}} \,\right]$

$T =$

	never married	married
never married	0.5	0.5
married	0	1

$\rightarrow \quad T^2 = \begin{pmatrix} 0.25 & 0.75 \\ 0 & 1 \end{pmatrix}, \ldots$

Predicted Distributions:

Year k	$p_{t+k} = p_t \, T^k$

Example: Marrying as 2-State Markov Process

Setup: Initial distribution $p_t = \left[\, 0.5_{\text{never married}} \;\; 0.5_{\text{married}} \,\right]$

$T =$

	never married	married
never married	0.5	0.5
married	0	1

$\rightarrow \quad T^2 = \begin{pmatrix} 0.25 & 0.75 \\ 0 & 1 \end{pmatrix}, \ldots$

Predicted Distributions:

Year k	$p_{t+k} = p_t\, T^k$
1	$[\,0.250000 \quad 0.750000\,]$

Example: Marrying as 2-State Markov Process

Setup: Initial distribution $p_t = \left[\, 0.5_{\text{never married}} \;\; 0.5_{\text{married}} \,\right]$

$T =$

	never married	married
never married	0.5	0.5
married	0	1

$\rightarrow \quad T^2 = \begin{pmatrix} 0.25 & 0.75 \\ 0 & 1 \end{pmatrix}, \ldots$

Predicted Distributions:

Year k	$p_{t+k} = p_t \, T^k$
1	$[\,0.250000 \quad 0.750000\,]$
2	$[\,0.125000 \quad 0.875000\,]$

Example: Marrying as 2-State Markov Process

Setup: Initial distribution $p_t = \begin{bmatrix} 0.5_{\text{never married}} & 0.5_{\text{married}} \end{bmatrix}$

$T =$

	never married	married
never married	0.5	0.5
married	0	1

$\rightarrow \quad T^2 = \begin{pmatrix} 0.25 & 0.75 \\ 0 & 1 \end{pmatrix}, \ldots$

Predicted Distributions:

Year k	$p_{t+k} = p_t \, T^k$
1	$[\,0.250000 \quad 0.750000\,]$
2	$[\,0.125000 \quad 0.875000\,]$
3	$[\,0.062500 \quad 0.937500\,]$

Example: Marrying as 2-State Markov Process

Setup: Initial distribution $p_t = \left[\, 0.5_{\text{never married}} \;\; 0.5_{\text{married}} \,\right]$

$T =$

	never married	married
never married	0.5	0.5
married	0	1

$\rightarrow \quad T^2 = \begin{pmatrix} 0.25 & 0.75 \\ 0 & 1 \end{pmatrix}, \ldots$

Predicted Distributions:

Year k	$p_{t+k} = p_t\, T^k$
1	[0.250000 0.750000]
2	[0.125000 0.875000]
3	[0.062500 0.937500]
4	[0.031250 0.968750]

Example: Marrying as 2-State Markov Process

Setup: Initial distribution $p_t = \left[\, 0.5_{\text{never married}} \;\; 0.5_{\text{married}} \,\right]$

$T =$

	never married	married
never married	0.5	0.5
married	0	1

$\rightarrow \quad T^2 = \begin{pmatrix} 0.25 & 0.75 \\ 0 & 1 \end{pmatrix}, \ldots$

Predicted Distributions:

Year k	$p_{t+k} = p_t \, T^k$
1	[0.250000 0.750000]
2	[0.125000 0.875000]
3	[0.062500 0.937500]
4	[0.031250 0.968750]
5	[0.015625 0.984375]

Example: Marrying as 2-State Markov Process

Setup: Initial distribution $p_t = \left[\, 0.5_{\text{never married}} \;\; 0.5_{\text{married}} \,\right]$

$T =$

	never married	married
never married	0.5	0.5
married	0	1

$\rightarrow \quad T^2 = \begin{pmatrix} 0.25 & 0.75 \\ 0 & 1 \end{pmatrix}, \ldots$

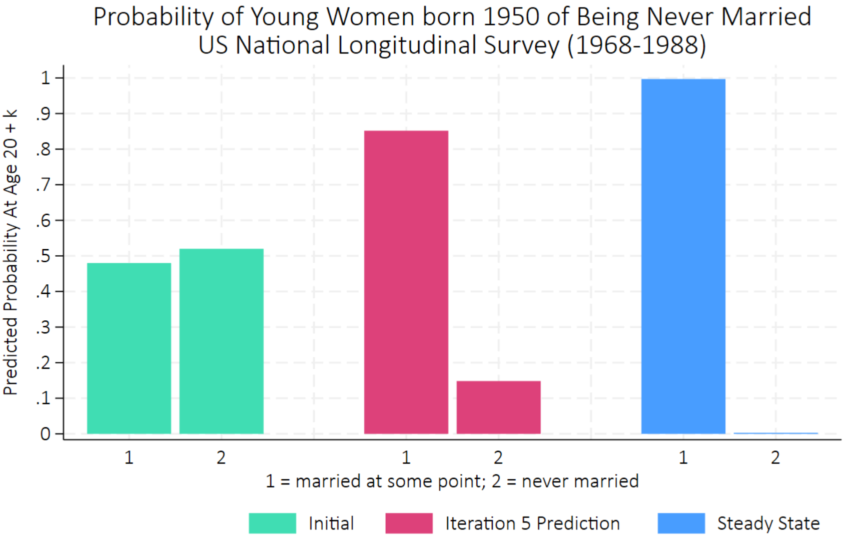
Predicted Distributions:

Year k	$p_{t+k} = p_t\, T^k$
1	$[\,0.250000 \quad 0.750000\,]$
2	$[\,0.125000 \quad 0.875000\,]$
3	$[\,0.062500 \quad 0.937500\,]$
4	$[\,0.031250 \quad 0.968750\,]$
5	$[\,0.015625 \quad 0.984375\,]$

Long Run:

$$\boldsymbol{\pi} = \lim_{k \to \infty} p_{t+k} = [\,0 \quad 1\,] \quad \text{(everyone eventually marries)}$$

How Quickly Get Young Women Married?




```
xtsteadystate nev_mar if birth_yr ==50, tw 3dists ini ss pred twowayopt(.)
```

Non-Markovian and/or Non-Stationary Processes

- What if the process is not Markovian and/or not stationary?
- Solution: add new state components until dynamics are Markovian and stationary

Non-Markovian and/or Non-Stationary Processes

- What if the process is not Markovian and/or not stationary?
- Solution: add new state components until dynamics are Markovian and stationary
  - Marrying: probability of marrying may depend on: How long a woman has been single, her past relationship history, norms in 1970 vs. 1980
  - Add time since last relationship, number of prior marriages, cohort, ...

Non-Markovian and/or Non-Stationary Processes

- What if the process is not Markovian and/or not stationary?
- Solution: add new state components until dynamics are Markovian and stationary
  - Marrying: probability of marrying may depend on: How long a woman has been single, her past relationship history, norms in 1970 vs. 1980
  - Add time since last relationship, number of prior marriages, cohort, ...
  - Where do we stop?

Markovian Stationary Process

- **Problem:** adding components to the state description to force a process to be Markovian and stationary may significantly increase computational complexity
- **Solution:** try to find the smallest state description that is self-sufficient (i.e., Markovian and stationary)

Decision Making

- Predictions by themselves are useless
- They are only useful when they will influence future decisions
- Hence the ultimate task is decision making
- How can we influence the process to visit desirable states?
  - Model: Markov Decision Process

References I

Russell, S. J., and P. Norvig (2016): *Artificial intelligence: a modern approach*. Pearson.

Takeaways

How Can we Use Markov Processes to Predict Future States?

- Model sequences of states with probabilistic transitions
- First-order Markov and stationarity assumptions simplify prediction
- Adding state components can restore Markovian/stationary properties—at a computational cost
- Prediction relies on transition matrices
- Real goal: use predictions for decision-making

  $\rightarrow$ Markov Decision Processes

Appendix

Prediction and Steady State via Eigendecomposition

Objective: Predict future state distributions $\mathbb{P}(s_{t+k} \mid s_t)$ and compute the steady-state distribution using eigendecomposition

Inputs:

- Initial distribution: p_t
- Transition matrix: T where $T_{ij} = \mathbb{P}(s_{t+1} = j \mid s_t = i)$
- Horizon: k (number of steps ahead)

Procedure:

1. Eigendecompose: $T = U \Lambda U^{-1}$
2. Compute predicted distribution:
$$p_{t+k} = T^k p_t = U \Lambda^k U^{-1} p_t$$
3. Steady state distribution:
$$\boldsymbol{\pi} = \lim_{k \to \infty} p_{t+k}$$