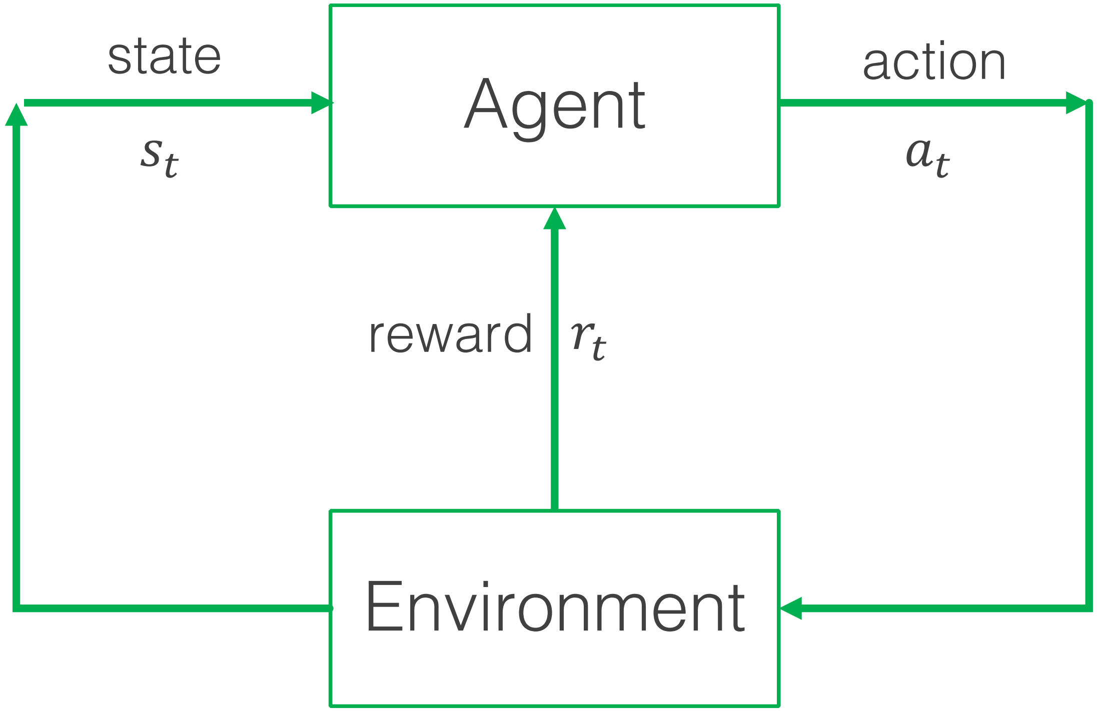
Reinforcement Learning II

Lecture 20

RL Components



Policy (agent behavior), $\pi(s)$

- Determines action given current state
- Agent's way of behaving at a given time

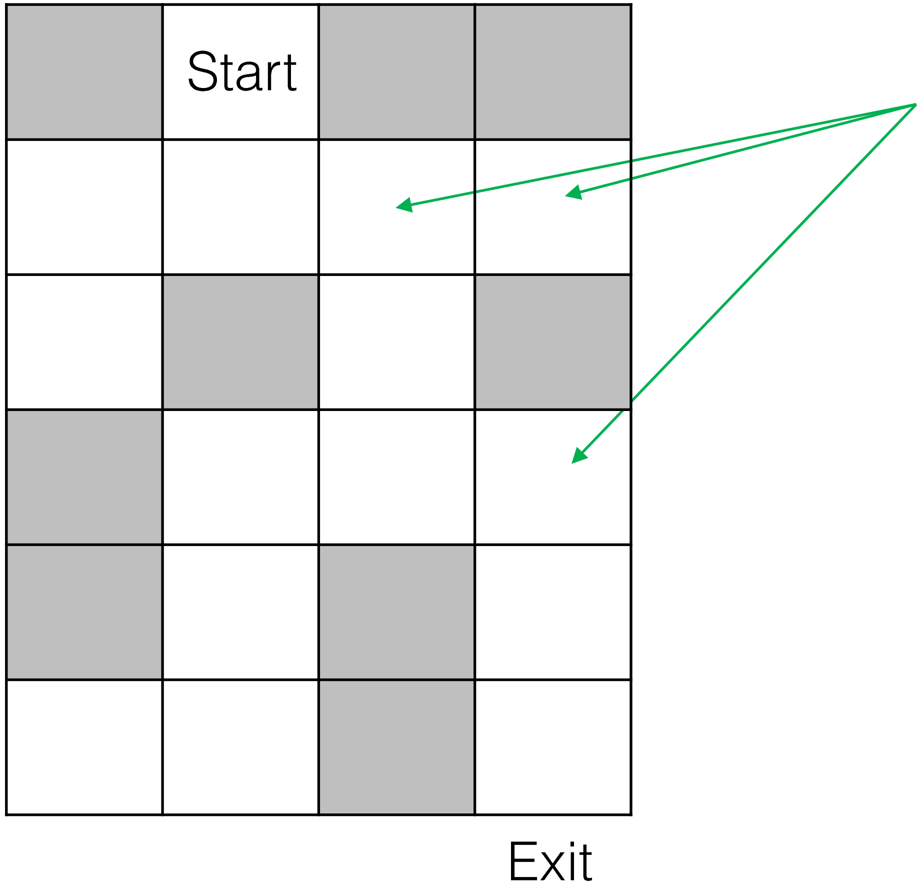
Reward function (the goal to max), r_t

- Maps state of the environment to a reward that describes the state desirability
- Objective is to **maximize total rewards**

Value functions (state reward), $v_\pi(s)$, $q_\pi(s,a)$

- Total expected reward from a state if we follow the policy
- How "good" is each state

Maze Example: Policy, Value, and Reward



Each location in the maze represents a **state**

The **reward** is -1 for each step the agent is in the maze

Available **actions**: move ↑,↓,←,→ (as long as that path is not blocked)

Adapted from David Silver, 2015

Policy $\pi(s)$

(which actions to take in each state)

Start

	↓		
→	→	↓	←
↑		↓	
	→	→	↓
	↑		↓
→	↑		↓

Exit

Reward r_t

(rewards are received as you transition OUT OF the state)

Start

	-1		
-1	-1	-1	-1
-1		-1	
	-1	-1	-1
	-1		-1
-1	-1		-1

Exit

State Value $v_\pi(s)$

(expected cumulative rewards starting from current state **if** we follow the policy)

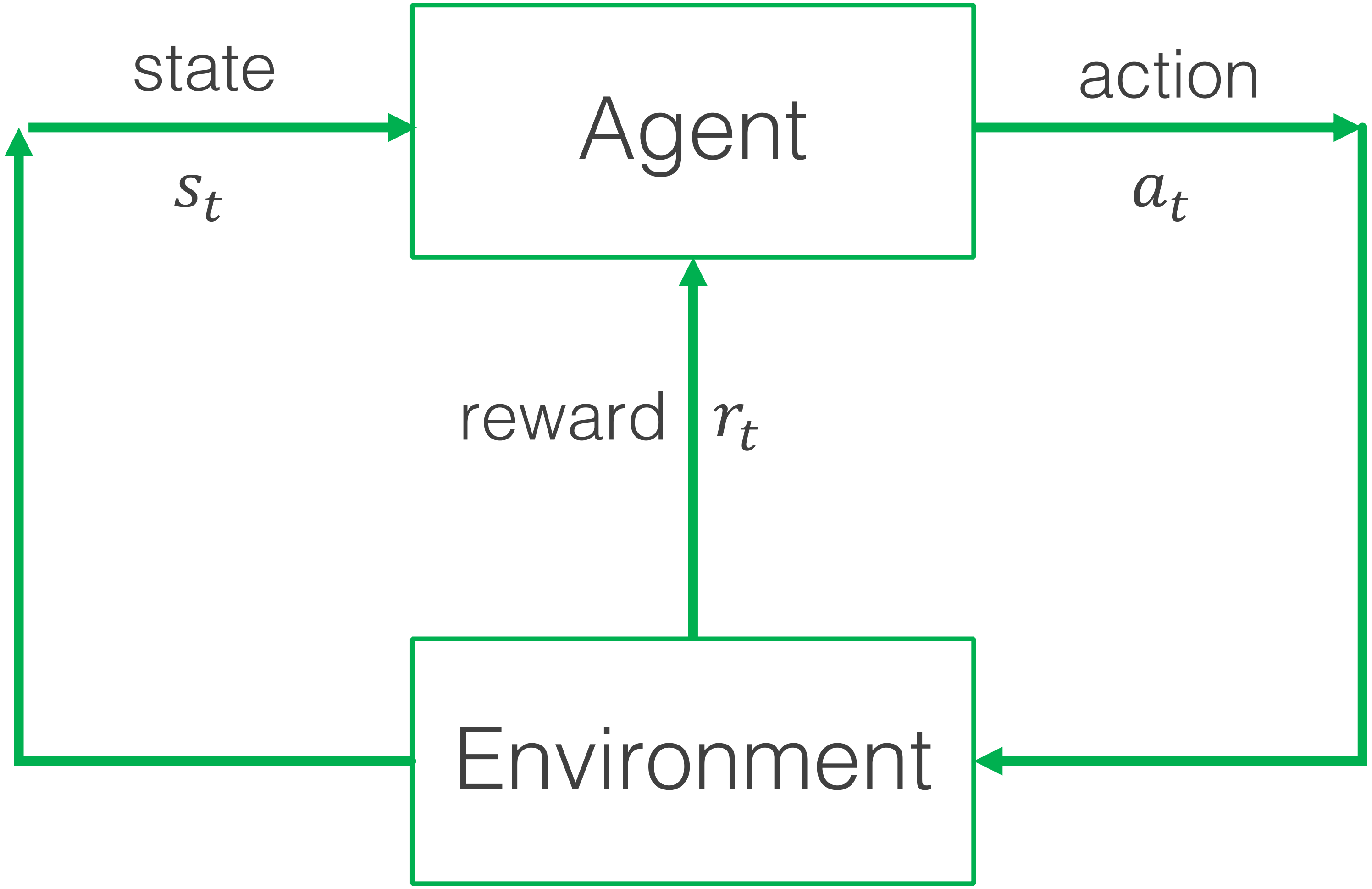
Start

	-8		
-8	-7	-6	-7
-9		-5	
	-5	-4	-3
	-6		-2
-8	-7		-1

Exit

Adapted from David Silver, 2015

Environment



Markov Decision Process

(assumed form for most RL problems)

Goal

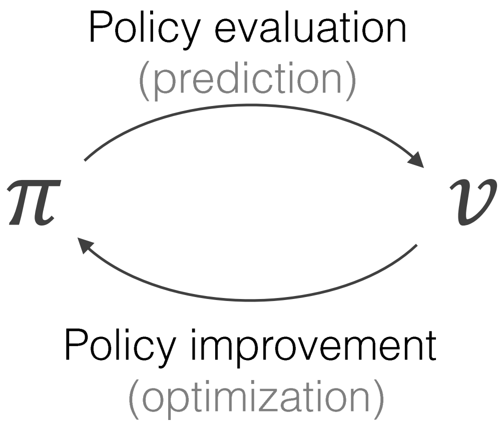
Find the best policy to guide our actions in an environment

Here, environment = Markov Decision Process

Reinforcement Learning

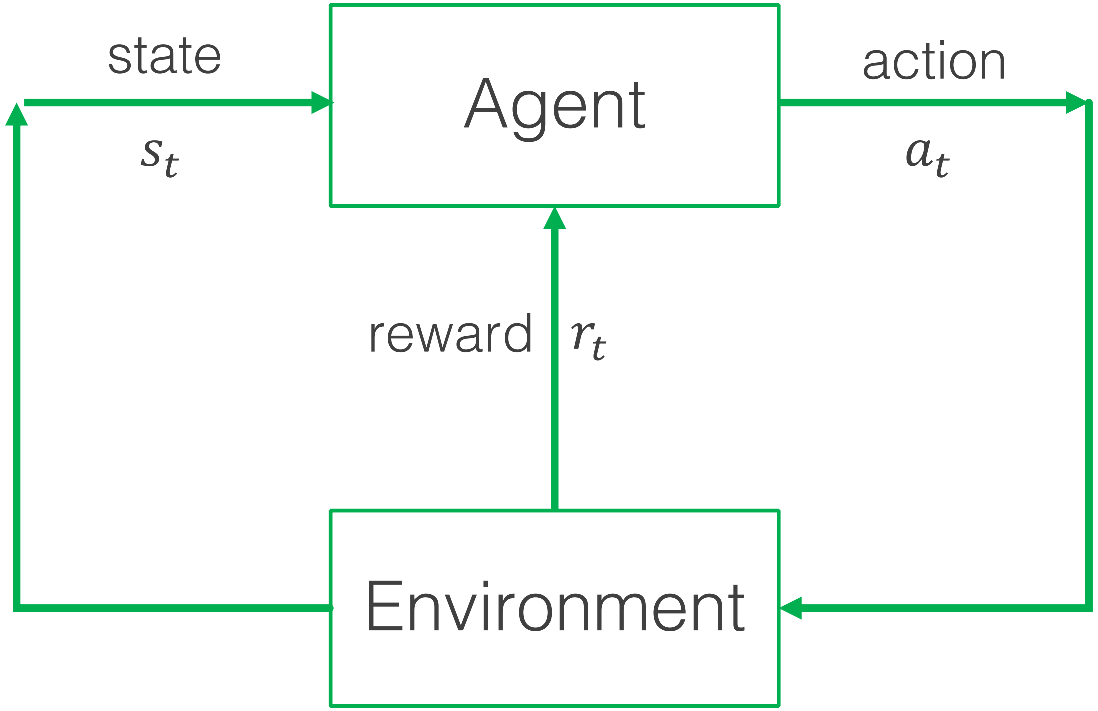
Learning strategy

Knowledge of **Environment**

	Model-based (planning)	Model-free
No knowledge Must learn from experience		
Perfect knowledge Known MDP	**Dynamic Programming** Policy iteration Value iteration  Policy evaluation (prediction) π v Policy improvement (optimization)	

History

The record of all that has happened in this system



Step 0: s_0, a_0

Step 1: r_1, s_1, a_1

Step 2: r_2, s_2, a_2

$\vdots$

Step T: r_t, s_t, a_t

History at time t : $H_t = \{s_t, a_t, r_{t-1}, s_{t-1}, a_{t-1}, \dots r_1, s_1, a_1, s_0, a_0\}$

Markov property

Instead of needing the full history:

$$H_t = \{s_t, a_t, r_{t-1}, s_{t-1}, a_{t-1}, \dots r_1, s_1, a_1, s_0, a_0\}$$

We can summarize everything in the current state

$$H_t = \{s_t, a_t\}$$

The future is independent of the past given the present

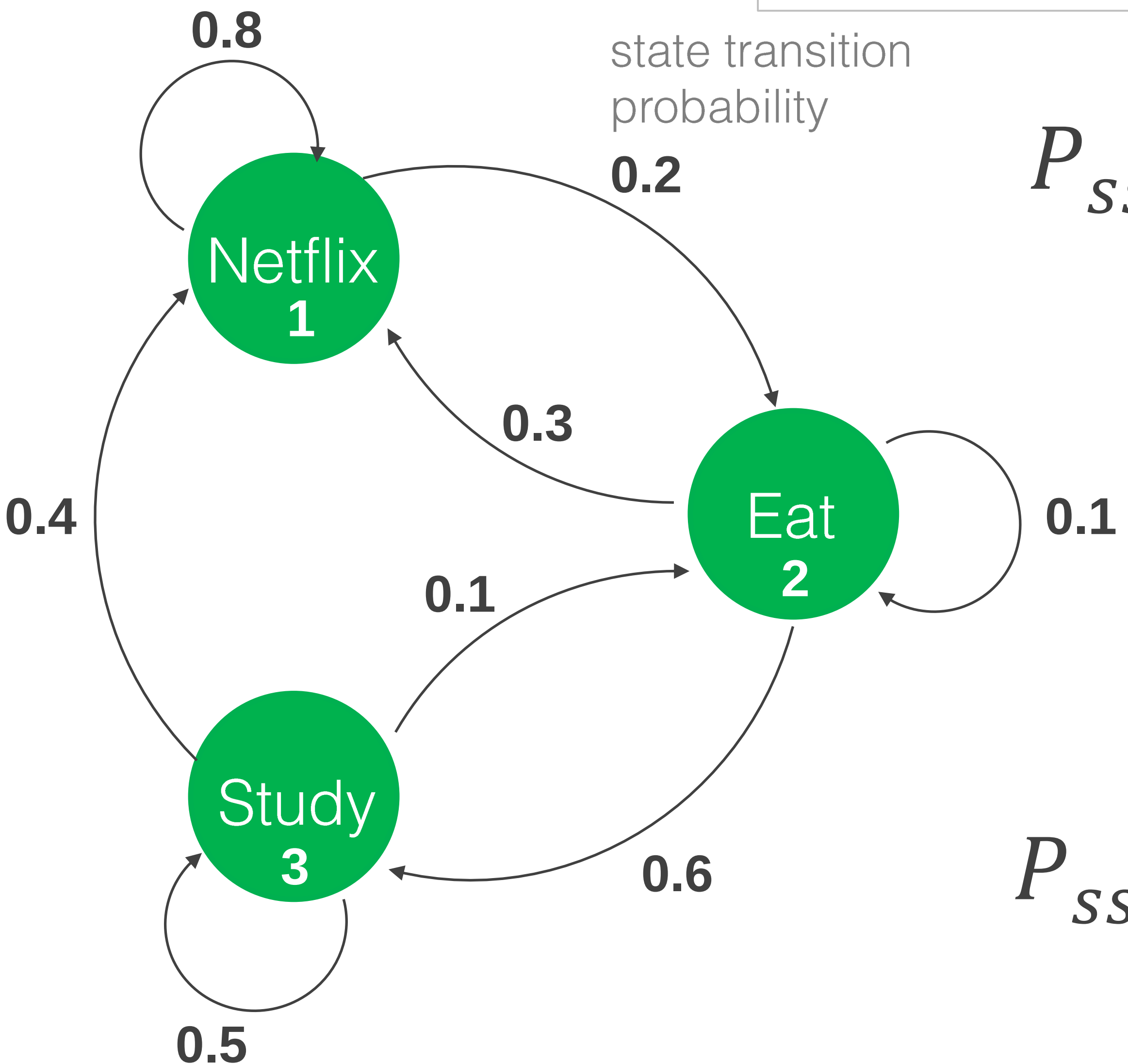
Another way of saying this is:

$$P(s_{t+1}|s_t) = P(s_{t+1}|s_t, s_{t-1}, \dots, s_1, s_0)$$

Markov Chains

Example: student life

Two components: $\{S, P\}$
State space, S
Transition matrix, P



state transition probability

State-space representation

State transition probabilities

To state

$$P_{ss'} = \begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \end{array}$$

From state

Transitions out of each state sum to 1

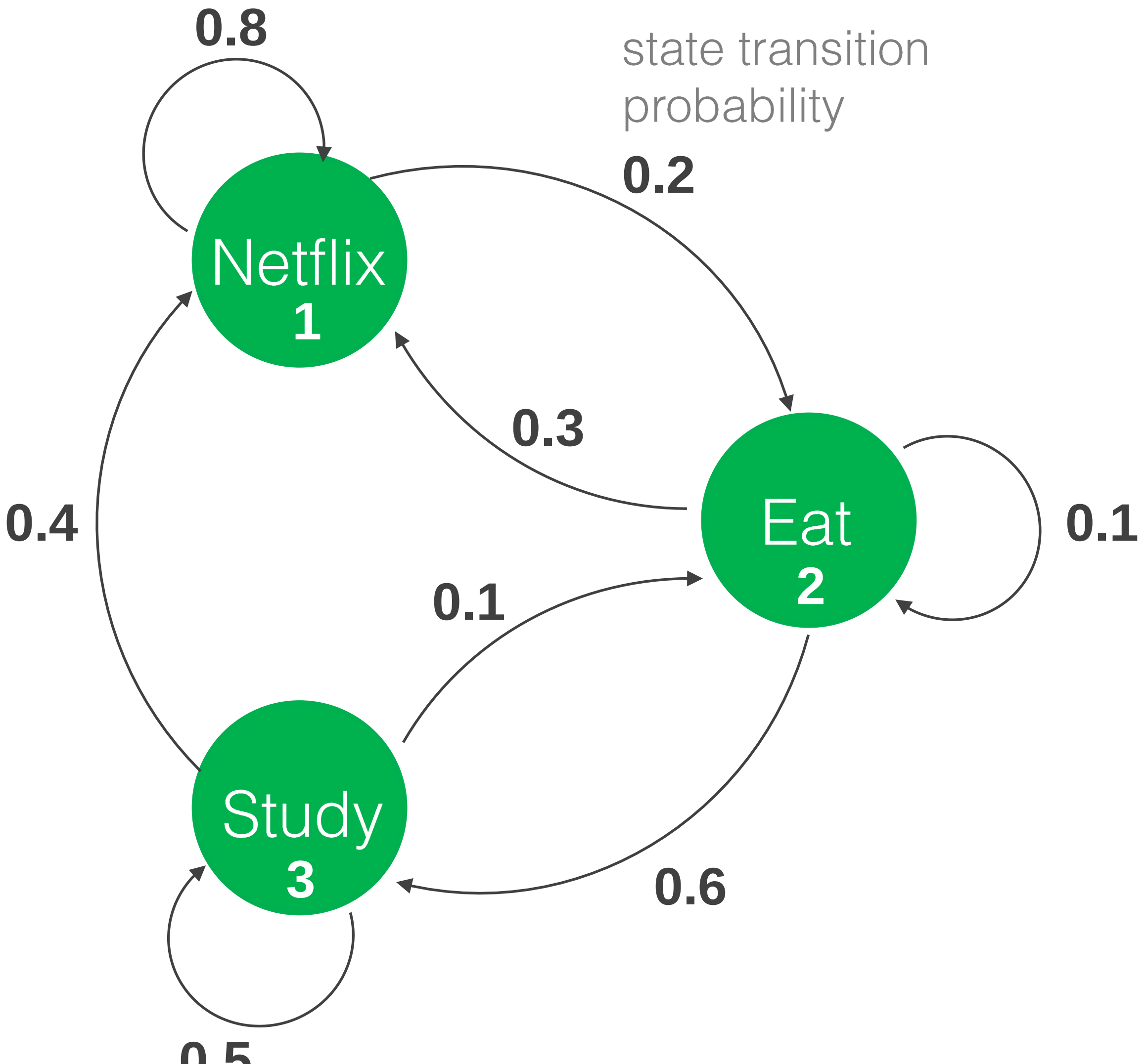
To state

$$P_{ss'} = \begin{array}{c c} & \begin{array}{ccc} \text{Netflix} & \text{Eat} & \text{Study} \end{array} \\ \begin{array}{c} \text{Netflix} \\ \text{Eat} \\ \text{Study} \end{array} & \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.3 & 0.1 & 0.6 \\ 0.4 & 0.1 & 0.5 \end{bmatrix} \end{array}$$

From state

Markov Chains

Example: student life



State-space representation

If we start in state 1, what's the probability we'll be in each state after one step?

$$\begin{array}{cccc} & 1 & 2 & 3 \\ P_1 = & [0.8 & 0.2 & 0] \end{array}$$

This is the first row of the state transition probability matrix

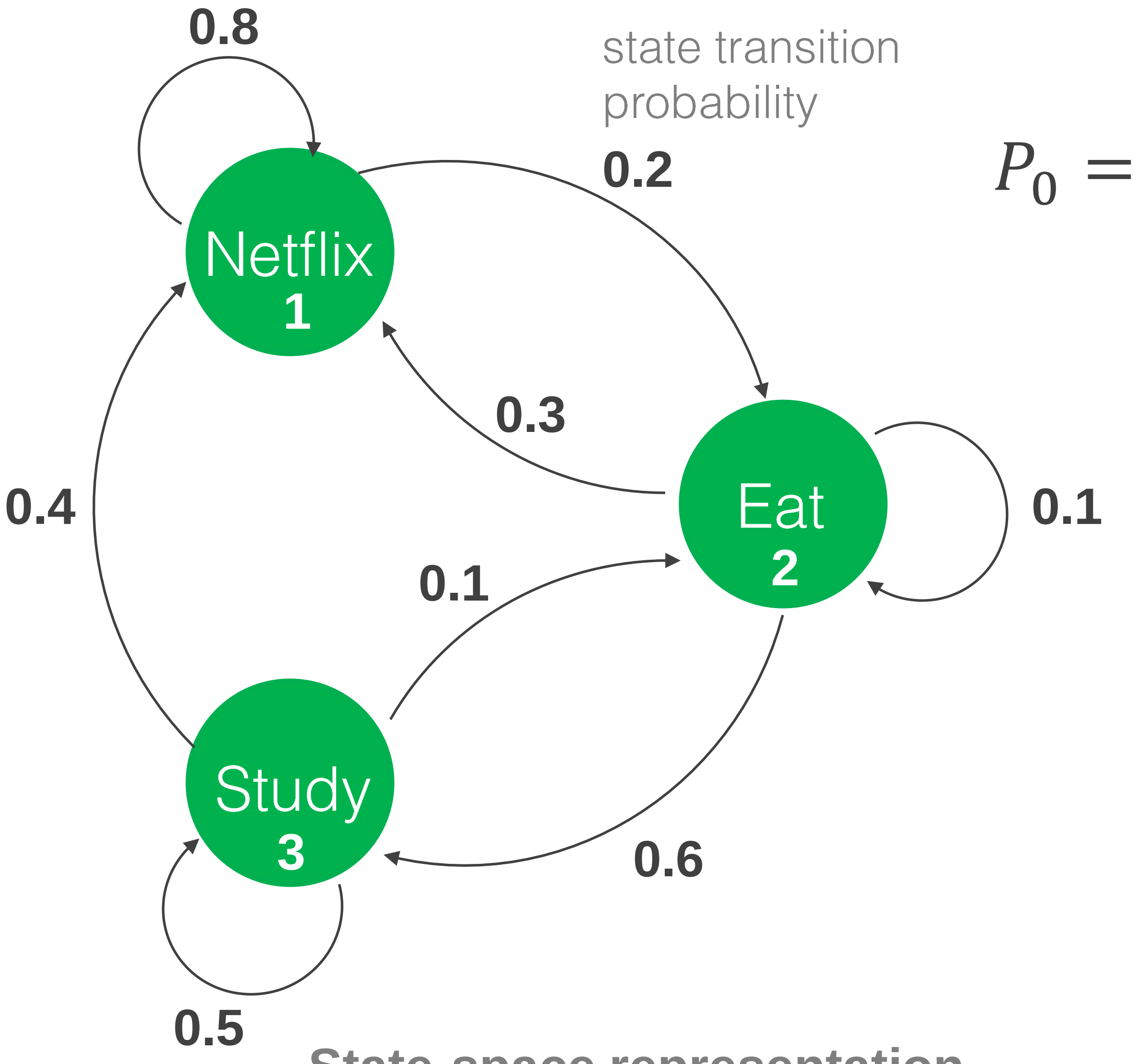
To state

From state

$$P_{ss'} = \begin{array}{c|ccc} & \text{Netflix} & \text{Eat} & \text{Study} \\ \hline \text{Netflix} & 0.8 & 0.2 & 0 \\ \text{Eat} & 0.3 & 0.1 & 0.6 \\ \text{Study} & 0.4 & 0.1 & 0.5 \end{array}$$

Markov Chains

Example: student life



State-space representation

If we started in state 1, we can calculate the probabilities of being in each state at step 1 as:

$$P_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^T$$

$$P_1 = P_0 P_{ss'}$$

$$P_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.3 & 0.1 & 0.6 \\ 0.4 & 0.1 & 0.5 \end{bmatrix}$$

$$P_1 = \begin{bmatrix} 0.8 & 0.2 & 0 \end{bmatrix}$$

To state

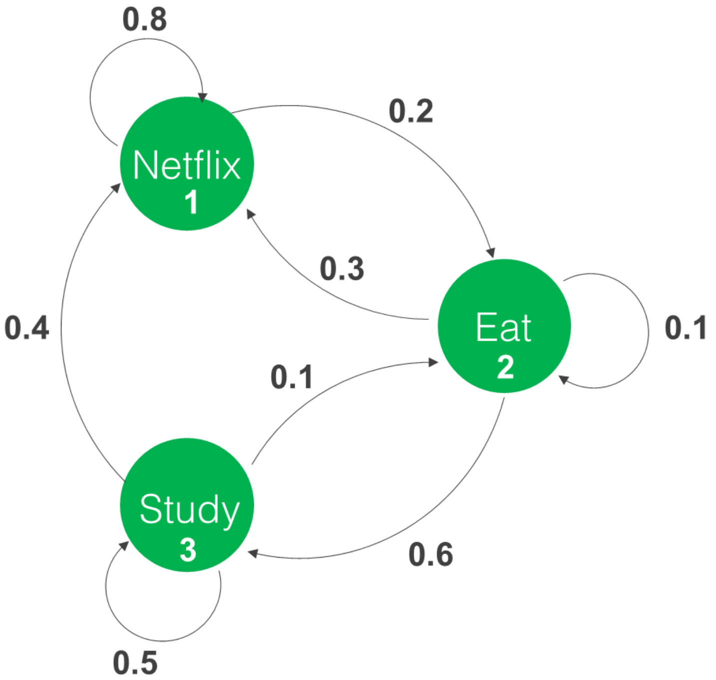
$$P_{ss'} = \begin{array}{c|c} & \begin{array}{ccc} \text{Netflix} & \text{Eat} & \text{Study} \end{array} \\ \text{From state} \begin{array}{c} \text{Netflix} \\ \text{Eat} \\ \text{Study} \end{array} & \begin{bmatrix} \boxed{0.8 \quad 0.2 \quad 0} \\ 0.3 \quad 0.1 \quad 0.6 \\ 0.4 \quad 0.1 \quad 0.5 \end{bmatrix} \end{array}$$

① $P_1 = P_0 P_{ss'}$

$$P_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.3 & 0.1 & 0.6 \\ 0.4 & 0.1 & 0.5 \end{bmatrix}$$

$$P_1 = \begin{bmatrix} 0.8 & 0.2 & 0 \end{bmatrix}$$

$$P_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^T$$



② $P_2 = P_1 P_{ss'} = P_0 P_{ss'} P_{ss'} = P_0 P_{ss'}^2$

$$P_2 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.3 & 0.1 & 0.6 \\ 0.4 & 0.1 & 0.5 \end{bmatrix} \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.3 & 0.1 & 0.6 \\ 0.4 & 0.1 & 0.5 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} 0.7 & 0.18 & 0.12 \end{bmatrix}$$

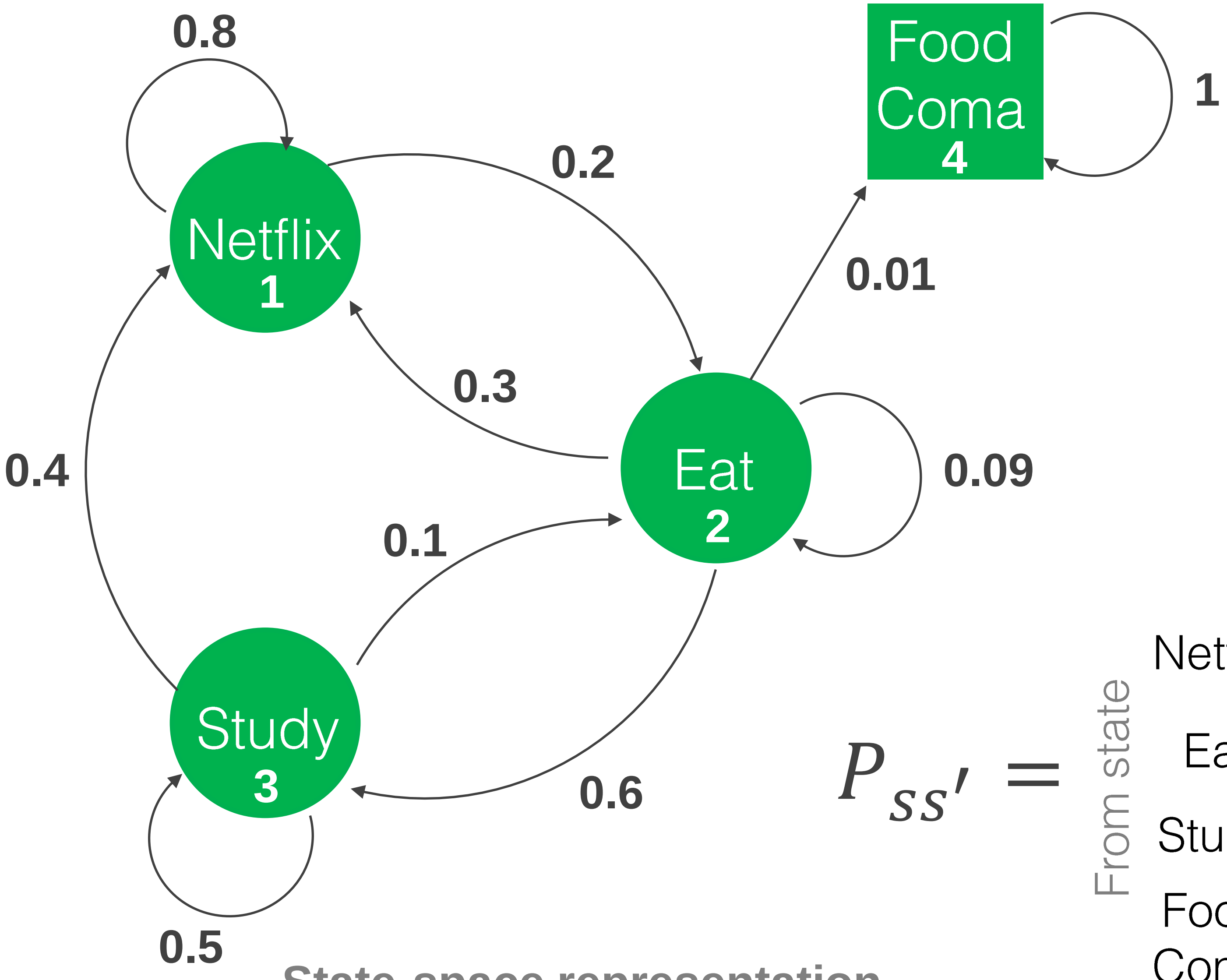
ⓝ $P_n = P_0 P_{ss'}^n$

As $n \rightarrow \infty$, we identify our steady state probabilities

$$P_\infty = \begin{bmatrix} 0.64 & 0.16 & 0.20 \end{bmatrix}$$

Markov Chains with absorbing state

Example: student life



State-space representation

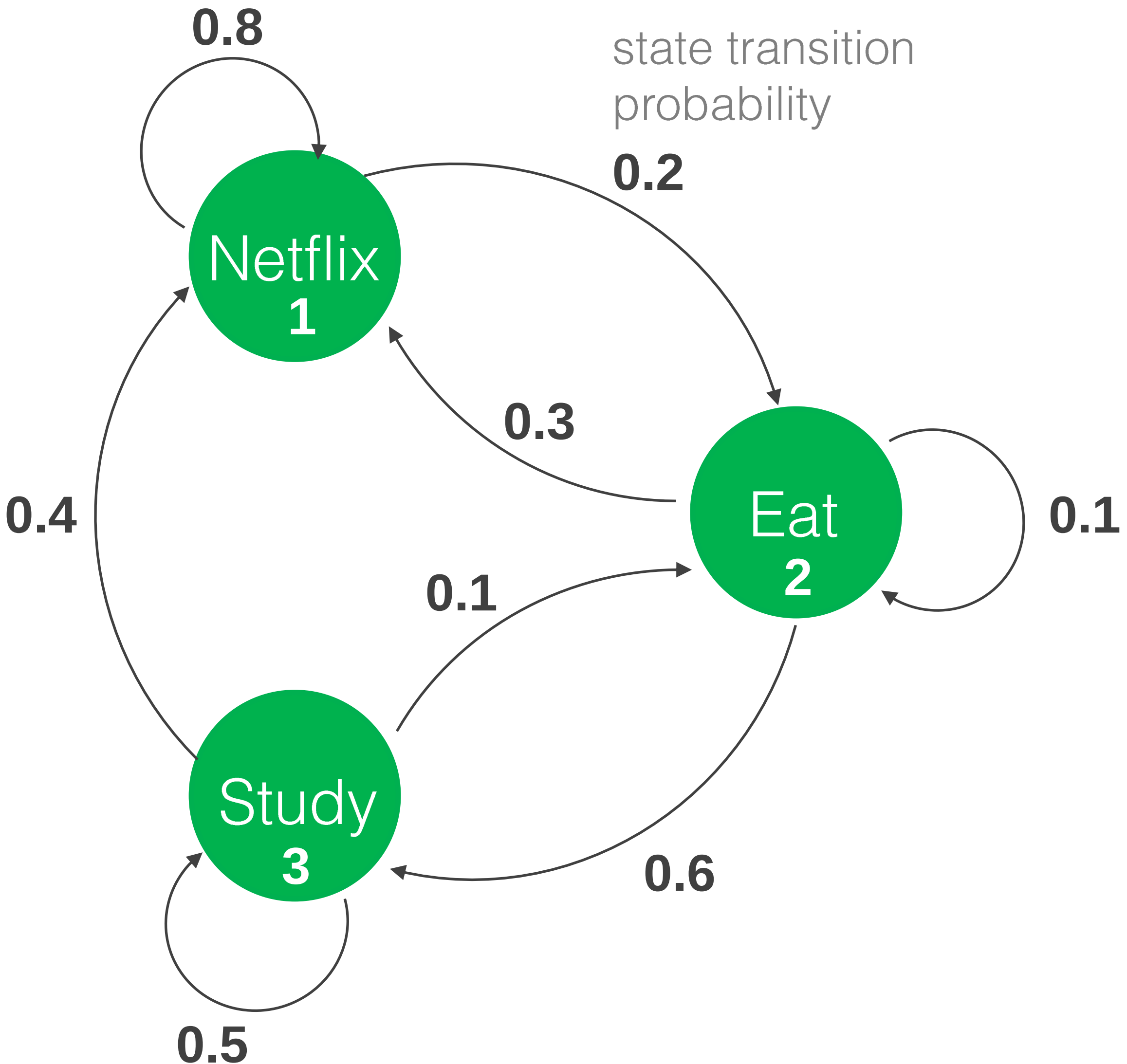
To state

From state

$$P_{ss'} = \left[\begin{array}{c|cccc} & \text{Netflix} & \text{Eat} & \text{Study} & \text{Food Coma} \\ \hline \text{Netflix} & 0.8 & 0.2 & 0 & 0 \\ \text{Eat} & 0.3 & 0.09 & 0.6 & 0.01 \\ \text{Study} & 0.4 & 0.1 & 0.5 & 0 \\ \text{Food Coma} & 0 & 0 & 0 & 1 \end{array}\right]$$

Markov Chains

Example: student life



State-space representation

Markov chains can be used to represent sequential discrete-time data

Can estimate long-term state probabilities

Can simulate state sequences based on the model

Markov property applies

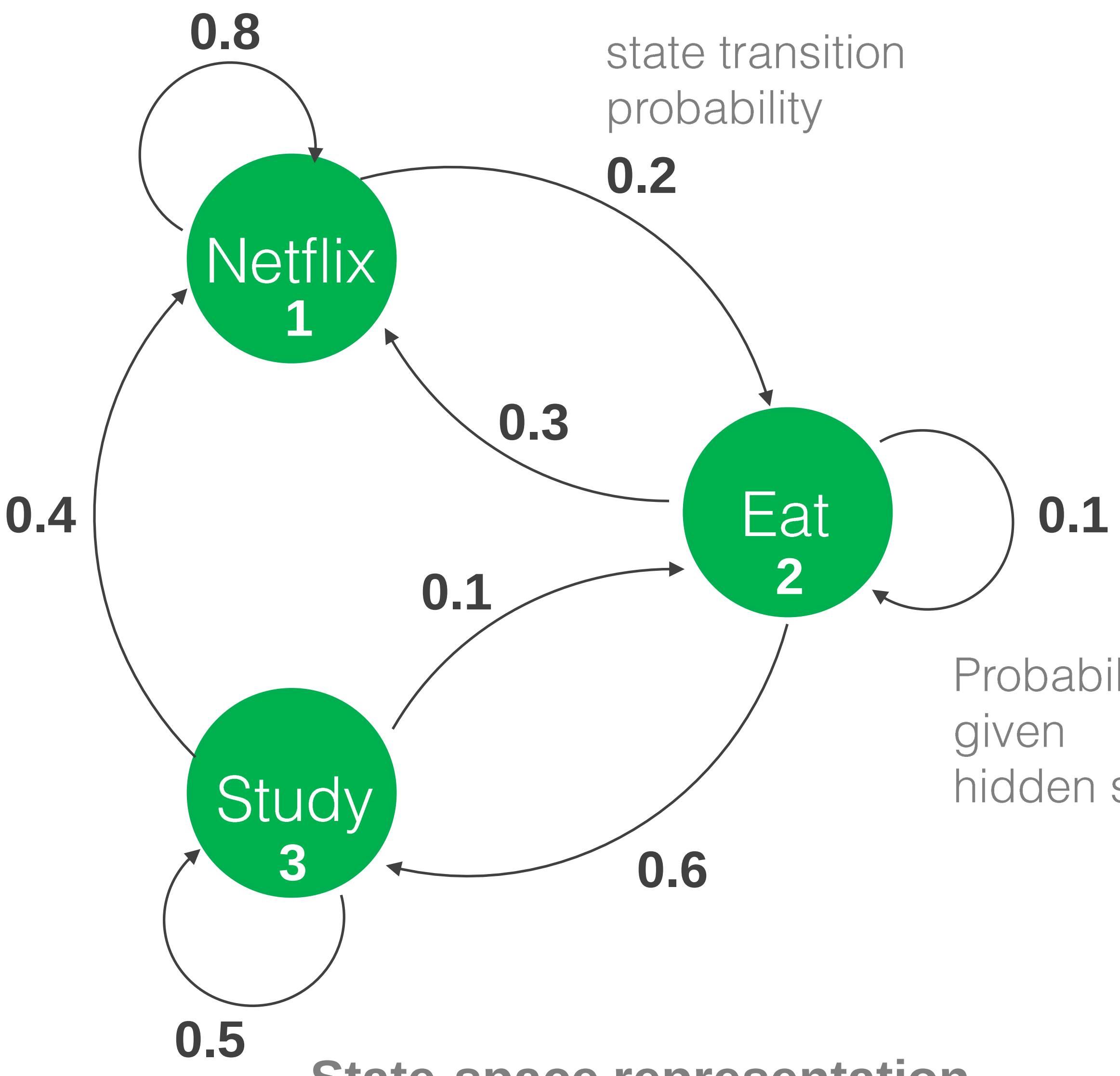
(current state gives you all the information you need about future states)

$$P(s_{t+1}|s_t) = P(s_{t+1}|s_t, s_{t-1}, \dots, s_1, s_0)$$

Valid if the system is **autonomous** and the states are **fully observable**

Hidden Markov Models

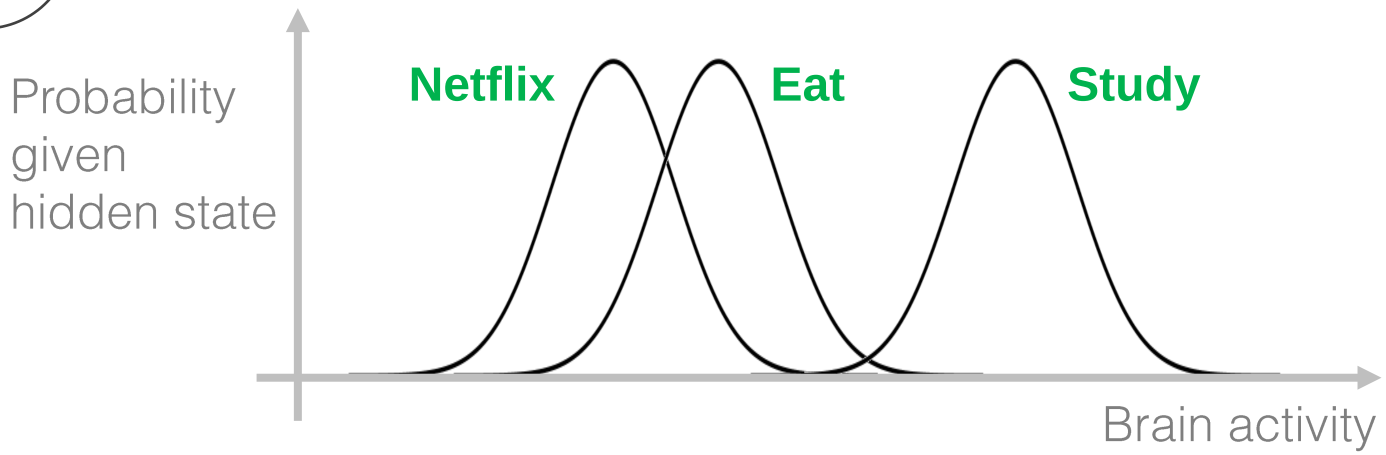
Example: student life



State-space representation

What if we **don't directly observe what state** the system is in, but instead observe a quantity that depends on the state?

Example: the student wears an EEG monitor, and we see readings of brain activity.



States are **hidden** or **latent variables**

Markov Models

	States are **Fully Observable**	States are **Partially Observable**
Autonomous (no actions; make predictions)	Markov Chain & Markov Reward Process	Hidden Markov Model (HMM)
Controlled (can take actions)	Markov Decision Process (MDP)	Partially Observable Markov Decision Process (POMDP)

Applications

HMMs: time series ML, e.g. speech + handwriting recognition, bioinformatics

MDPs: used extensively for reinforcement learning

Building blocks for the full RL problem

1	Markov Chain	{state space S, transition probabilities P}
2	Markov Reward Process (MRP)	{S, P, + rewards R, discount rate γ} adds rewards (and values)
3	Markov Decision Process (MDP)	{S, P, R, γ, + actions A} adds decisions (i.e. the ability to control)

MDPs form the basis for most reinforcement learning environments

Adapted from David Silver, 2015