ECE276B: Planning & Learning in Robotics
Lecture 1: Markov Chains

Instructor:
Nikolay Atanasov: natanasov@ucsd.edu

Teaching Assistants:
Zhichao Li: zhl355@eng.ucsd.edu
Ehsan Zobeidi: ezobeidi@eng.ucsd.edu
Ibrahim Akbar: iakbar@eng.ucsd.edu

UC San Diego
JACOBS SCHOOL OF ENGINEERING
Electrical and Computer Engineering

What is this class about?

- **ECE276A**: sensing and state estimation:
  - how to model a robot's motion and observations
  - how to estimate (a distribution of) the robot state x_t from the history of observations $z_{0:t}$ and control inputs $u_{0:t-1}$
- **ECE276B**: planning and decision making:
  - how to select control $u_{0:t-1}$ to achieve safe navigation or maximize rewards
- **References** (not required):
  - Dynamic Programming and Optimal Control: Bertsekas
  - Planning Algorithms: LaValle (http://planning.cs.uiuc.edu)
  - Reinforcement Learning: Sutton & Barto (http://incompleteideas.net/book/the-book.html)
  - Calculus of Variations and Optimal Control Theory: Liberzon (http://liberzon.csl.illinois.edu/teaching/cvoc.pdf)

Logistics

- Course website: `https://natanaso.github.io/ece276b`
- Includes links to (sign up!):
  - **Piazza**: discussion – it is your responsibility to check Piazza regularly because class announcements, updates, etc., will be posted there
  - **GradeScope**: homework submissions and grades
- Four assignments:
  - Project 1: Dynamic Programming (20% of final grade)
  - Project 2: Motion Planning (25% of final grade)
  - Project 3: Control & Reinforcement Learning (25% of final grade)
  - Final Exam (30% of final grade)
- Each project includes:
  - theoretical homework
  - programming assignment(s) in **python**
  - project report
- Grades:
  - assigned based on the class performance, i.e., there will be a "curve"
  - **no late policy**: Work submitted past the deadline will receive 0 credit

Prerequisites

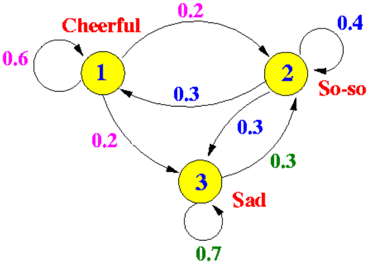
- **Probability theory**: random vectors, probability density functions, expectation, covariance, total probability, conditioning, Bayes rule
- **Linear algebra/systems**: eigenvalues, positive definiteness, linear systems of ODEs, matrix exponential
- **Optimization**: gradient descent, linear constraints, convex functions
- **Programming**: experience with at least one language (python/C++/Matlab), classes/objects, data structures (e.g., queue, list), data input/output, plotting
- It is up to you to judge if you are ready for this course!
  - Consult with your classmates who took ECE276A
  - Take a look at the material from last year: `https://natanaso.github.io/ece276b2018`
  - If the first assignment in ECE276B seems hard, the rest will be hard as well

Syllabus (Winter 2018)

Date	Lecture	Materials	Assignments
Jan 09	Introduction		
Jan 11	Markov Chains	Grinstead-Snell-Ch11	
Jan 16	Markov Decision Processes	Bertsekas 1.1-1.2	HW1
Jan 18	Dynamic Programming	Bertsekas 1.3-1.4	
Jan 23	Deterministic Shortest Path	Bertsekas 2.1-2.3	
Jan 25	Catch-up		
Jan 30	Configuration Space	LaValle 4.3, 6.2-6.3	HW2
Feb 01	Search-based Planning	LaValle 2.1-2.3, JPS	
Feb 06	Anytime Incremental Search	RTAA*, ARA*, AD*, Journal Paper	
Feb 08	Catch-up		
Feb 13	Sampling-based Planning	LaValle 5.5-5.6	
Feb 15	Stochastic Shortest Path	Bertsekas 7.1-7.3	
Feb 20	Bellman Equations I	Sutton-Barto 4.1-4.4	HW3
Feb 22	Bellman Equations II	Sutton-Barto 4.5-4.8	
Feb 27	Continuous-time Optimal Control	Bertsekas 3.1-3.2	
Mar 01	Pontryagin's Minimum Principle	Bertsekas 3.3-3.4, Liberzon Ch. 2.4 and Ch. 4	
Mar 06	Catch-up		
Mar 08	Linear Quadratic Control	Bertsekas 4.1	HW4
Mar 13	Model-free Prediction	Sutton-Barto 6.1-6.3	
Mar 15	Model-free Control	Sutton-Barto 6.4-6.7	

Markov Chain

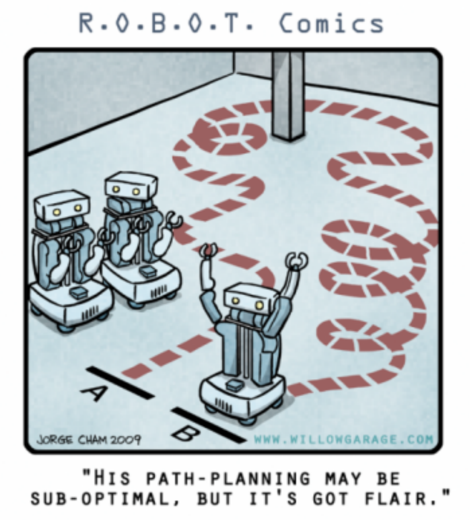
- A **Markov Chain** is a probabilistic model used to represent the evolution of a robot system
- The state $x_t \in \{1, 2, 3\}$ is fully observed (unlike HMM and Bayes filtering settings)
- The transitions are random, determined by a transition kernel but uncontrolled (just like in the HMM and Bayes filtering settings, the control input is known)
- A **Markov Decision Process** (MDP) is a Markov chain, whose transitions are controlled



$$P = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.3 & 0.4 & 0.3 \\ 0.0 & 0.3 & 0.7 \end{bmatrix}$$

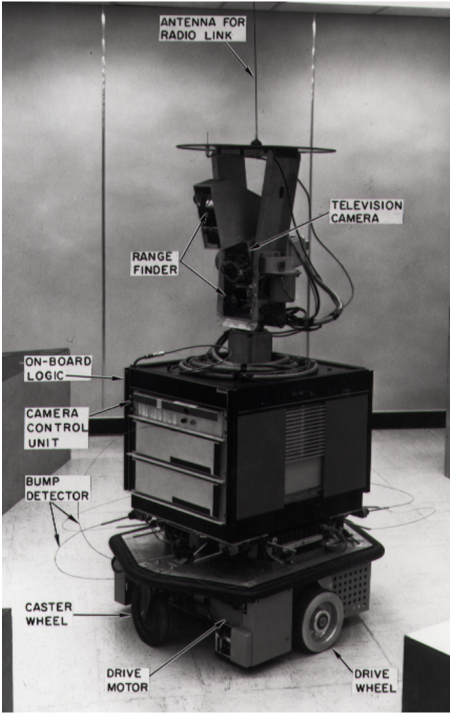
$$P_{ij} = \mathbb{P}(x_{t+1} = j \mid x_t = i)$$

Motion Planning

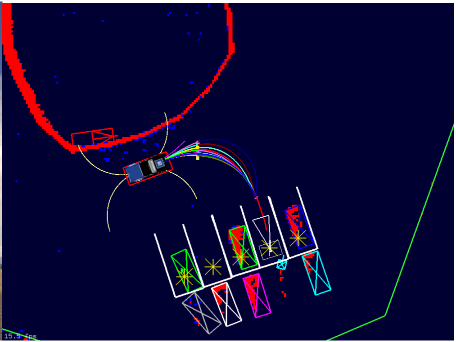


A* Search

- Invented by Hart, Nilsson and Raphael of Stanford Research Institute in 1968 for the Shakey robot
- Video: `https://youtu.be/qXdn6ynwpiI?t=3m55s`

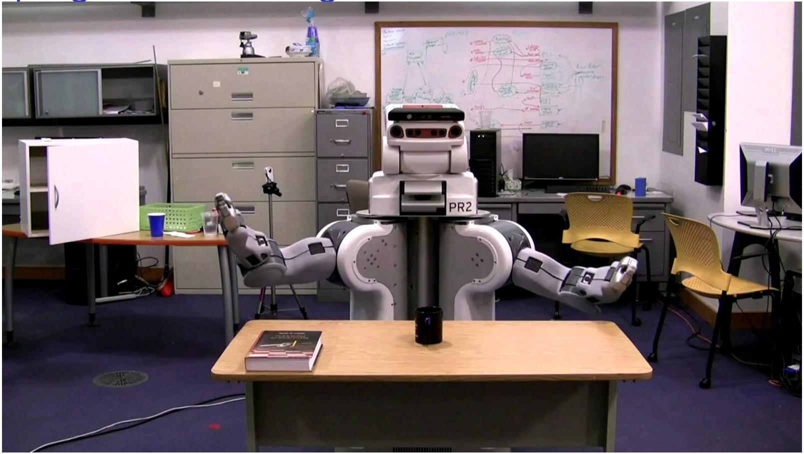


Search-based Planning



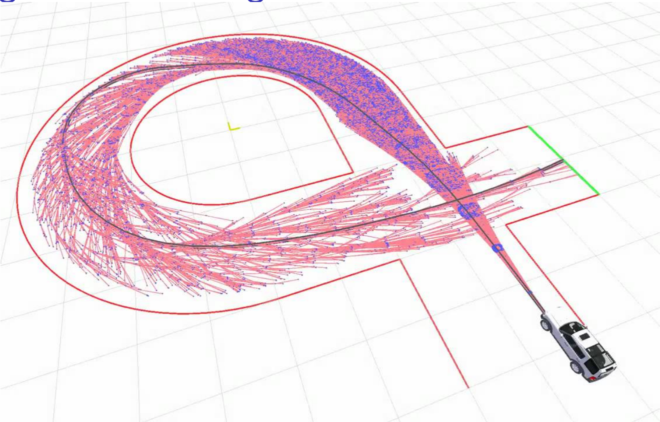
- CMU's autonomous car used search-based planning in the DARPA Urban Challenge in 2007
- Likhachev and Ferguson, "Planning Long Dynamically Feasible Maneuvers for Autonomous Vehicles," IJRR'09
- Video: `https://www.youtube.com/watch?v=4hFhl0Oi8KI`
- Video: `https://www.youtube.com/watch?v=qXZt-B7iUyw`
- Paper: `http://journals.sagepub.com/doi/pdf/10.1177/0278364909340445`

Sampling-based Planning



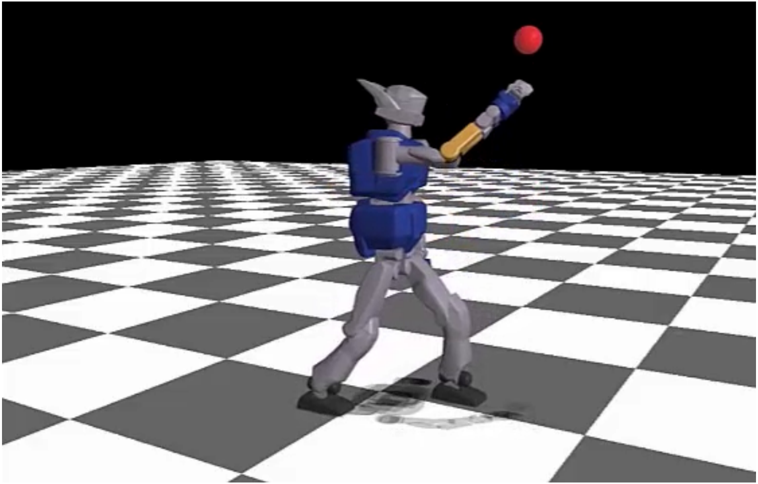
- RRT algorithm on the PR2 – planning with both arms (12 DOF)
- Karaman and Frazzoli, “Sampling-based algorithms for optimal motion planning,” IJRR’11
- Video: `https://www.youtube.com/watch?v=vW74bC-Ygb4`
- Paper: `http://journals.sagepub.com/doi/pdf/10.1177/0278364911406761`

Sampling-based Planning



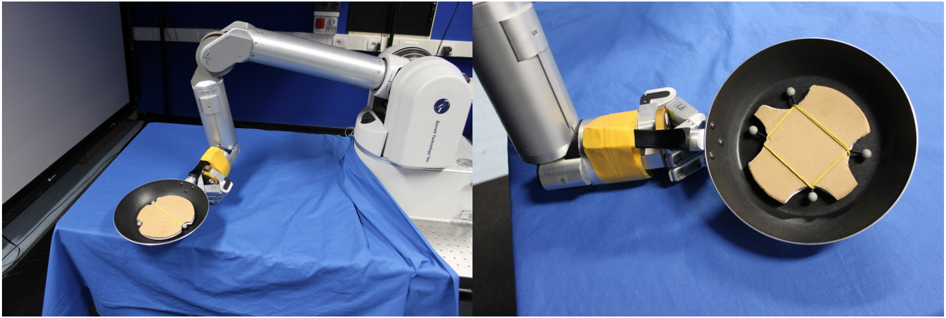
- RRT* algorithm on a race car – 270 degree turn
- Karaman and Frazzoli, "Sampling-based algorithms for optimal motion planning," IJRR'11
- Video: `https://www.youtube.com/watch?v=p3nZHnOWhrg`
- Video: `https://www.youtube.com/watch?v=LKL5qRBiJaM`
- Paper: `http://journals.sagepub.com/doi/pdf/10.1177/0278364911406761`

Dynamic Programming and Optimal Control



- Tassa, Mansard and Todorov, "Control-limited Differential Dynamic Programming," ICRA'14
- Video: `https://www.youtube.com/watch?v=tCQSSkBH2NI`
- Paper: `http://ieeexplore.ieee.org/document/6907001/`

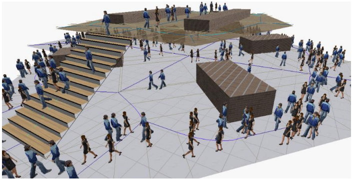
Model-free Reinforcement Learning



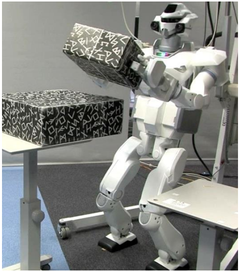
- Robot learns to flip pancakes
- Kormushev, Calinon and Caldwell, "Robot Motor Skill Coordination with EM-based Reinforcement Learning," IROS'10
- Video: `https://www.youtube.com/watch?v=W_gxLKSsSIE`
- Paper: `http://www.dx.doi.org/10.1109/IROS.2010.5649089`

Applications of Optimal Control & Reinforcement Learning

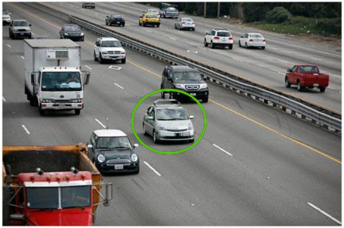
(a) Games



(b) Character Animation

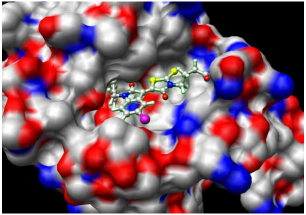


(c) Robotics



(d) Autonomous Driving

(e) Marketing



(f) Computational Biology

Problem Formulation

- **Motion model**: specifies how a dynamical system evolves

$$x_{t+1} = f(x_t, u_t, w_t) \sim p_f(\cdot \mid x_t, u_t), \quad t = 0, \ldots, T-1$$

  - discrete time $t \in \{0, \ldots, T\}$
  - state $x_t \in \mathcal{X}$
  - control $u_t \in \mathcal{U}(x_t)$ and $\mathcal{U} := \bigcup_{x \in \mathcal{X}} \mathcal{U}(x)$
  - motion noise w_t (random vector) with known probability density function (pdf) and assumed conditionally independent of other disturbances w_τ for $\tau \neq t$ for given x_t and u_t
  - the motion model is specified by the nonlinear function f or equivalently by the pdf p_f of x_{t+1} conditioned on x_t and u_t

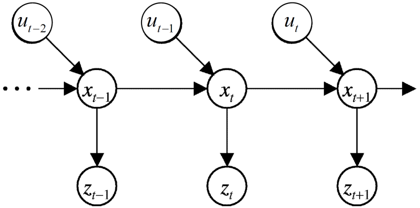
- **Observation model**: the state x_t might not be observable but perceived through measurements:

$$z_t = h(x_t, v_t) \sim p_h(\cdot \mid x_t), \quad t = 0, \ldots, T$$

  - measurement noise v_t (random vector) with known pdf and conditionally independent of other disturbances v_τ for $\tau \neq t$ for given x_t and w_t for all t
  - the observation model is specified by the nonlinear function h or equivalently by the pdf p_h of z_t conditioned on x_t

Problem Formulation

- **Markov Assumptions**
  - The state x_{t+1} only depends on the previous time input u_t and state x_t
  - The observation z_t only depends on the state x_t



- **Joint distribution**:

$$p(x_{0:T}, z_{0:T}, u_{0:T-1}) = \underbrace{p_{0|0}(x_0)}_{\text{prior}} \prod_{t=0}^{T} \underbrace{p_h(z_t \mid x_t)}_{\text{observation model}} \prod_{t=1}^{T} \underbrace{p_f(x_t \mid x_{t-1}, u_{t-1})}_{\text{motion model}}$$

- **The Problem of Acting Optimally**: Given a model p_f of the system evolution and direct observations of its state x_t (or prior pdf $p_{0|0}$ and observation model p_h) determine control inputs $u_{0:T-1}$ to minimize (maximize) a scalar-valued additive cost (reward) function:

$$V_0^{u_{0:T-1}}(x_0) := \mathbb{E}_{x_{1:T}} \left[\underbrace{\mathfrak{q}(x_T)}_{\text{terminal cost}} + \sum_{t=0}^{T-1} \underbrace{\ell(x_t, u_t)}_{\text{stage cost}} \;\middle|\; x_0, u_{0:T-1} \right]$$

Problem Solution: Control Policy

- The problem of acting optimally is called:
  - **Optimal Control** (OC): when the models p_f, p_h are known
  - **Reinforcement Learning** (RL): when the models are unknown but samples can be obtained from them
  - **Inverse RL/OC**: when the cost (reward) functions ℓ are unknown

- The solution to an OC/RL problem is a **policy** π
  - Let $\pi_t(x_t)$ map a state $x_t \in \mathcal{X}$ to a feasible control input $u_t \in \mathcal{U}(x_t)$
  - The sequence $\pi := \{\pi_0(\cdot), \pi_1(\cdot), \ldots, \pi_{T-1}(\cdot)\} = \pi_{0:T-1}$ of functions π_t is called an **admissible control policy**
  - The cost (reward) of a policy $\pi \in \Pi$ (set of all admissible policies) is:

$$V_0^\pi(x_0) := \mathbb{E}_{x_{1:T}}\left[\mathfrak{q}(x_T) + \sum_{t=0}^{T-1} \ell(x_t, \pi_t(x_t)) \,\middle|\, x_0\right]$$

  - a policy $\pi^* \in \Pi$ is an **optimal policy** if $V_0^{\pi^*}(x_0) \leq V_0^\pi(x_0)$ for all $\pi \in \Pi$ and its cost will be denoted $V_0^*(x_0) := V_0^{\pi^*}(x_0)$

- Conventions differ in optimal control and reinforcement learning:
  - **OC**: minimization, cost, state x, control u, policy μ
  - **RL**: maximization, reward, state s, action a, policy π
  - **ECE276B**: minimization, cost, state x, control u, policy π

Further Observations

- Goal: select controls to minimize long-term cumulative costs
  - Controls may have long-term consequences, e.g., delayed reward
  - It may be better to sacrifice immediate reward to gain long-term rewards:
    - A financial investment may take months to mature
    - Refueling a helicopter might prevent a crash in several hours
    - Blocking opponent moves might help winning chances many moves from now

- **Information state**: a sequence (history) of observations and control inputs $i_t := z_0, u_0, \dots, z_{t-1}, u_{t-1}, z_t$ used in the partially observable setting to estimate the (pdf of the) state x_t

- A policy fully defines the behavior of the robot/agent by specifying, at any given point in time, which controls to apply. Policies can be:
  - **stationary** ($\pi \equiv \pi_0 \equiv \pi_1 \equiv \cdots$) $\subseteq$ **non-stationary** (time-dependent)
  - **deterministic** ($u_t = \pi_t(x_t)$) $\subseteq$ **stochastic** ($u_t \sim \pi_t(\cdot \mid x_t)$)
  - **open-loop** (a sequence $u_{0:T-1}$ regardless of x_t or i_t) $\subseteq$ **closed-loop** (π_t depends on x_t or i_t)

Problem Variations

- **deterministic** (no noise v_t, w_t) vs **stochastic**
- **fully observable** (no noise v_t and $z_t = x_t$) vs **partially observable**
  - **fully observable**: Markov Decision Process (MDP)
  - **partially observable**: Partially Observable Markov Decision Process (POMDP)
- **stationary** vs **nonstationary** (time-dependent $p_{f,t}$, $p_{h,t}$, ℓ_t)
- **finite** vs **continuous** state space $\mathcal{X}$
  - tabular approach vs function approximation (linear, SVM, neural nets,...)
- **finite** vs **continuous** control space $\mathcal{U}$:
  - tabular approach vs optimization problem to select next-best control
- **discrete** vs **continuous** time:
  - finite-horizon discrete time: dynamic programming
  - infinite-horizon ($T \rightarrow \infty$) discrete time: Bellman equation (first-exit vs discounted vs average-reward)
  - continuous time: Hamilton-Jacobi-Bellman (HJB) Partial Differential Equation (PDE)
- reinforcement learning (p_f, p_h are unknown) variants:
  - **Model-based RL**: explicitly approximate models from experience and use optimal control algorithms
  - **Model-free RL**: directly learn a control policy without approximating the motion/observation models

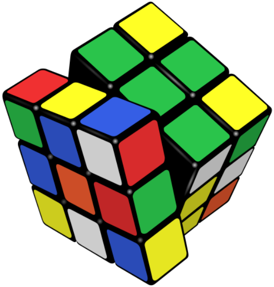
Example: Inventory Control

- Consider the problem of keeping an item stocked in a warehouse:
  - If there is too little, we will run out of it soon (not preferred).
  - If there is too much, the storage cost will be high (not preferred).

- We can model this scenario as a discrete-time system:
  - $x_t \in \mathbb{R}$: stock available in the warehouse at the beginning of the t-th time period
  - $u_t \in \mathbb{R}_{\geq 0}$: stock ordered and immediately delivered at the beginning of the t-th time period (supply)
  - w_t: (random) demand during the t-th time period with known pdf. Note that excess demand is back-logged, i.e., corresponds to negative stock x_t
  - **Motion model**: $x_{t+1} = x_t + u_t - w_t$
  - **Cost function**: $\mathbb{E}\left[R(x_T) + \sum_{t=0}^{T-1} \left(r(x_t) + cu_t - pw_t\right)\right]$ where
    - pw_t: revenue
    - cu_t: cost of items
    - $r(x_t)$: penalizes too much stock or negative stock
    - $R(x_T)$: remaining items we cannot sell or demand that we cannot meet

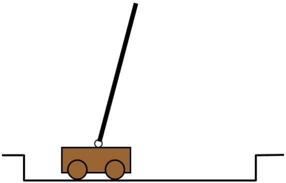
Example: Rubik's Cube

- Invented in 1974 by Ernõ Rubik
- Formalization
  - State space: $\sim 4.33 \times 10^{19}$
  - Actions: 12
  - Reward: -1 for each time step
  - Deterministic, Fully Observable
- The cube can be solved in 20 or fewer moves



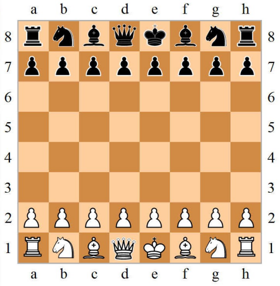
Example: Pole Balancing

- Move the cart left and right in order to keep the pole balanced
- Formalization
  - State space: 4-D continuous $(x, \dot{x}, \theta, \dot{\theta})$
  - Actions: $\{-N, N\}$
  - Reward:
    - 0 when in the goal region
    - -1 when outside the goal region
    - -100 when outside the feasible region
  - Deterministic, Fully Observable



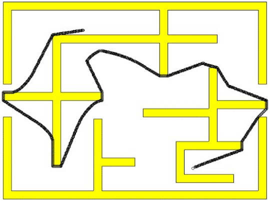
Example: Chess

- Formalization
  - State space: $\sim 10^{47}$
  - Actions: from 0 to 218
  - Reward: 0 each step, $\{-1, 0, 1\}$ at the end of the game
  - Deterministic, opponent-dependent state transitions (can be modeled as a game)
- The size of the game tree is 10^{123}



Example: Grid World Navigation

- Navigate to a goal without crashing into obstacles
- Formalization
  - State space: robot pose, e.g., 2-D position
  - Actions: allowable robot movement, e.g., $\{left, right, up, down\}$
  - Reward: -1 until the goal is reached; $-\infty$ if an obstacles is hit
  - Can be deterministic or stochastic; fully or partially observable



Definition of Markov Chain

- **Stochastic process**: an indexed collection of random variables $\{x_0, x_1, \ldots\}$ on a measurable space $(\mathcal{X}, \mathcal{F})$
  - example: time series of weekly demands for a product
- A temporally homogeneous **Markov chain** is a stochastic process $\{x_0, x_1, \ldots\}$ of $(\mathcal{X}, \mathcal{F})$-valued random variables such that:
  - $x_0 \sim p_{0|0}(\cdot)$ for a prior probability density function on $(\mathcal{X}, \mathcal{F})$
  - $\mathbb{P}(x_{t+1} \in A \mid x_{0:t}) = \mathbb{P}(x_{t+1} \in A \mid x_t) = \int_A p_f(x \mid x_t) dx$ for $A \in \mathcal{F}$ and a conditional pdf $p_f(\cdot \mid x_t)$ on $(\mathcal{X}, \mathcal{F})$
- Intuitive definition:
  - In a Markov Chain the distribution of $x_{t+1} \mid x_{0:t}$ depends only on x_t (a memoryless stochastic process)
  - The state captures all information about the history, i.e., once the state is known, the history may be thrown away
  - "The future is independent of the past given the present" (**Markov Assumption**)

Formal Definition of Markov Chain

- A measurable space $(\mathcal{X}, \mathcal{F})$ is called **nice** (or standard Borel space) if it is **isomorphic** to a compact metric space with the Borel σ-algebra (i.e., there exists a one-to-one map ϕ from $\mathcal{X}$ into $\mathbb{R}^n$ such that both ϕ and ϕ^{-1} are measurable)

- A **Markov transition kernel** is a function $\mathbb{P}_f : (\mathcal{X}, \mathcal{F}) \to [0, 1]$ on a nice space $(\mathcal{X}, \mathcal{F})$ such that:
  - $\mathbb{P}_f(x, \cdot)$ is a probability measure on $(\mathcal{X}, \mathcal{F})$ for all $x \in \mathcal{X}$
  - $\mathbb{P}_f(\cdot, A)$ is measurable for all $A \in \mathcal{F}$

- A temporally homogeneous **Markov chain** is a sequence $\{x_0, x_1, \ldots\}$ of $(\mathcal{X}, \mathcal{F})$-valued random variables such that:
  - $x_0 \sim \mathbb{P}_{0|0}(\cdot)$ for a prior probability measure on $(\mathcal{X}, \mathcal{F})$
  - $x_{t+1} \mid x_{0:t} \sim \mathbb{P}_f(x_t, \cdot)$ for a Markov transition kernel $\mathbb{P}_f$ on $(\mathcal{X}, \mathcal{F})$, i.e., the distribution of $x_{t+1} \mid x_{0:t}$ depends only on x_t so that:

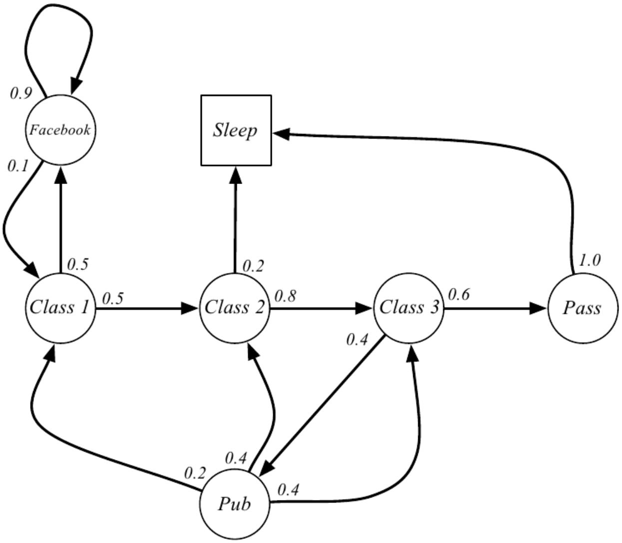
"the future is conditionally independent of the past, given the present"

Markov Chain

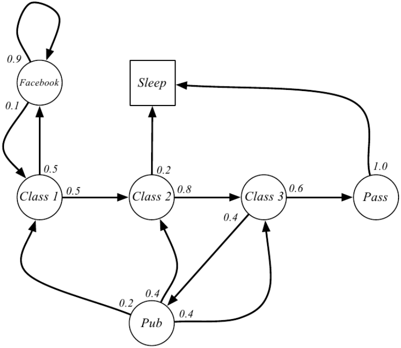
A **Markov Chain** is a stochastic process defined by a tuple $(\mathcal{X}, p_{0|0}, p_f)$:

- $\mathcal{X}$ is discrete/continuous set of states
- $p_{0|0}$ is a prior pmf/pdf defined on $\mathcal{X}$
- $p_f(\cdot \mid x_t)$ is a conditional pmf/pdf defined on $\mathcal{X}$ for given $x_t \in \mathcal{X}$ that specifies the stochastic process transitions. In the finite-dimensional case, the transition pmf is summarized by a matrix $P_{ij} := \mathbb{P}(x_{t+1} = j \mid x_t = i) = p_f(j \mid x_t = i)$

Example: Student Markov Chain



Example: Student Markov Chain



- Sample paths:
  - C1 C2 C3 Pass Sleep
  - C1 FB FB C1 C2 Sleep
  - C1 C2 C3 Pub C2 C3 Pass Sleep
  - C1 FB FB C1 C2 C3 Pub C1 FB FB FB C1 C2 Sleep
- Transition matrix:

$$P = \begin{array}{c} FB \\ C1 \\ C2 \\ C3 \\ Pub \\ Pass \\ Sleep \end{array} \begin{bmatrix} 0.9 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0 & 0 & 0.2 \\ 0 & 0 & 0 & 0 & 0.4 & 0.6 & 0 \\ 0 & 0.2 & 0.4 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Chapman-Kolmogorov Equation

- ***n*-step transition probabilities** of a time-homogeneous Markov chain on $\mathcal{X} = \{1, \ldots, N\}$

$$P_{ij}^{(n)} := \mathbb{P}(X_{t+n} = j \mid X_t = i) = \mathbb{P}(X_n = j \mid X_0 = i)$$

- **Chapman-Kolmogorov**: the *n*-step transition probabilities can be obtained recursively from the 1-step transition probabilities:

$$P_{ij}^{(n)} = \sum_{k=1}^{N} P_{ik}^{(m)} P_{kj}^{(n-m)}, \qquad \forall i,\ j,\ n,\ 0 \le m \le n$$

$$P^{(n)} = \underbrace{P \cdots P}_{n \text{ times}} = P^n$$

- Given the transition matrix P and a vector $p_{0|0}$ of prior probabilities, the vector of probabilities after t steps is:

$$\boxed{p_{t|t}^T = p_{0|0}^T P^t}$$

Example: Student Markov Chain

$$P = \begin{array}{r} FB \\ C1 \\ C2 \\ C3 \\ Pub \\ Pass \\ Sleep \end{array} \begin{bmatrix} 0.9 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0 & 0 & 0.2 \\ 0 & 0 & 0 & 0 & 0.4 & 0.6 & 0 \\ 0 & 0.2 & 0.4 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P^2 = \begin{array}{r} FB \\ C1 \\ C2 \\ C3 \\ Pub \\ Pass \\ Sleep \end{array} \begin{bmatrix} 0.86 & 0.09 & 0.05 & 0 & 0 & 0 & 0 \\ 0.45 & 0.05 & 0 & 0.4 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 0 & 0.32 & 0.48 & 0.2 \\ 0 & 0.08 & 0.16 & 0.16 & 0 & 0 & 0.6 \\ 0.1 & 0 & 0.1 & 0.32 & 0.16 & 0.24 & 0.08 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P^{100} = \begin{array}{r} FB \\ C1 \\ C2 \\ C3 \\ Pub \\ Pass \\ Sleep \end{array} \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0 & 0 & 0.99 \\ 0.01 & 0 & 0 & 0 & 0 & 0 & 0.99 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

First Passage Time

- **First Passage Time**: the number of transitions necessary to go from x_0 to state j for the first time (random variable $\tau_j := \inf\{t \geq 1 \mid x_t = j\}$)
- **Recurrence Time**: the first passage time to go from $x_0 = i$ to $j = i$
- **Probability of first passage in n steps**: $\rho_{ij}^{(n)} := \mathbb{P}(\tau_j = n \mid x_0 = i)$

$$
\begin{aligned}
\rho_{ij}^{(1)} &= P_{ij} \\
\rho_{ij}^{(2)} &= [P^2]_{ij} - \rho_{ij}^{(1)} P_{jj} \qquad \text{(first time we visit } j \text{ should not be 1!)} \\
&\vdots \\
\rho_{ij}^{(n)} &= [P^n]_{ij} - \rho_{ij}^{(1)}[P^{n-1}]_{jj} - \rho_{ij}^{(2)}[P^{n-2}]_{jj} - \cdots - \rho_{ij}^{(n-1)} P_{jj}
\end{aligned}
$$

- **Probability of first passage**: $\rho_{ij} := \mathbb{P}(\tau_j < \infty \mid x_0 = i) = \sum_{n=1}^{\infty} \rho_{ij}^{(n)}$
- **Number of visits to j up to time n**:

$$
v_j^{(n)} := \sum_{t=0}^{n} \mathbb{1}\{x_t = j\} \qquad v_j := \lim_{n \to \infty} v_j^{(n)}
$$

Recurrence and Transience

- **Absorbing state**: a state j such that $P_{jj} = 1$
- **Transient state**: a state j such that $\rho_{jj} < 1$
- **Recurrent state**: a state j such that $\rho_{jj} = 1$
- **Positive recurrent state**: a recurrent state j with $\mathbb{E}\left[\tau_j \mid x_0 = j\right] < \infty$
- **Null recurrent state**: a recurrent state j with $\mathbb{E}\left[\tau_j \mid x_0 = j\right] = \infty$
- **Periodic state**: can only be visited at integer multiples of t
- **Ergodic state**: a positive recurrent state that is aperiodic

Recurrence and Transience

Total Number of Visits Lemma

$$\mathbb{P}(v_j \geq k+1 \mid x_0 = j) = \rho_{jj}^k \text{ for all } k \geq 0$$

Proof: By the (strong) Markov property and induction ($\mathbb{P}(v_j \geq k+1 \mid x_0 = j) = \rho_{jj}\mathbb{P}(v_j \geq k \mid x_0 = j)$).

$0-1$ Law for Total Number of Visits

j is recurrent iff $\mathbb{E}[v_j \mid x_0 = j] = \infty$

Proof: Since v_j is discrete, we can write $v_j = \sum_{k=0}^{\infty} \mathbb{1}\{v_j > k\}$ and

$$\mathbb{E}[v_j \mid x_0 = j] = \sum_{k=0}^{\infty} \mathbb{P}(v_j \geq k+1 \mid x_0 = j) = \sum_{k=0}^{\infty} \rho_{jj}^k = \frac{\rho_{jj}}{1-\rho_{jj}}$$

Theorem: Recurrence is contagious

i is recurrent and $\rho_{ij} > 0 \quad \Rightarrow \quad j$ is recurrent and $\rho_{ji} = 1$

Classification of Markov Chains

- **Absorbing Markov Chain**: contains at least one absorbing state that can be reached from every other state (not necessarily in one step)
- **Irreducible Markov Chain**: it is possible to go from every state to every state (not necessarily in one step)
- **Ergodic Markov Chain**: an aperiodic, irreducible and positive recurrent Markov chain
- **Stationary distribution**: a vector $w \in \{p \in [0,1]^N \mid \mathbf{1}^T p = 1\}$ such that $w^T P = w^T$
  - Absorbing chains have stationary distributions with nonzero elements only in absorbing states
  - Ergodic chains have a unique stationary distribution (Perron-Frobenius Theorem)
  - Some periodic chains only satisfy a weaker condition, where $w_j > 0$ only for recurrent states and w_j is the frequency $\frac{v_j^{(n)}}{n+1}$ of being in state j as $n \to \infty$

Absorbing Markov Chains

- Interesting questions:
  - Q1: On average, how mant times is the process in state j?
  - Q2: What is the probability that the state will eventually be absorbed?
  - Q3: What is the expected absorption time?
  - Q4: What is the probability of being absorbed by j given that we started in i?

Absorbing Markov Chains

- **Canonical form**: reorder the states so that the transient ones come first: $P = \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix}$

- One can show that $P^n = \begin{bmatrix} Q^n & * \\ 0 & I \end{bmatrix}$ and $Q^n \to 0$ as $n \to \infty$

  Proof: If j is transient, then $\rho_{ij} < \infty$ and from the 0-1 Law:

$$\infty > \mathbb{E}[v_j \mid x_0 = i] = \mathbb{E}\left[\sum_{n=0}^{\infty} \mathbb{1}\{x_n = j\} \,\middle|\, x_0 = i\right] = \sum_{n=0}^{\infty} [P^n]_{ij}$$

- **Fundamental matrix**: $Z^A = (I - Q)^{-1} = \sum_{n=0}^{\infty} Q^n$ exists for an absorbing Markov chain
  - Expected number of times the chain is in state j: $Z^A_{ij} = \mathbb{E}[v_j \mid x_0 = i]$
  - Expected absorption time when starting from state i: $\sum_j Z^A_{ij}$
  - Let $B = Z^A R$. The probability of reaching absorbing state j starting from state i is B_{ij}

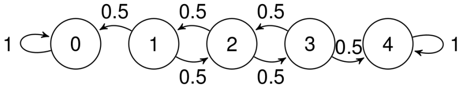
Example: Drunkard's Walk

- Transition matrix:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- Canonical form:

$$P = \begin{bmatrix} 0 & 0.5 & 0 & 0.5 & 0 \\ 0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



- Fundamental matrix:

$$Z^A = (I - Q)^{-1} = \begin{bmatrix} 1.5 & 1 & 0.5 \\ 1 & 2 & 1 \\ 0.5 & 1 & 1.5 \end{bmatrix}$$

Perron-Frobenius Theorem

Theorem

Let P be the transition matrix of an irreducible, aperiodic, finite, time-homogeneous Markov chain with stationary distribution w. Then

- 1 is the eigenvalue of max modulus, i.e., $|\lambda| < 1$ for all other eigenvalues
- 1 is a simple eigenvalue, i.e., the associated eigenspace and left-eigenspace have dimension 1
- The eigenvector is $\mathbf{1}^T$, the unique left eigenvector w is nonnegative and

$$\lim_{n \to \infty} P^n = \mathbf{1} w^T$$

Hence, w is the unique stationary distribution for the Markov chain and any initial distribution converges to it.

Fundamental Matrix for Ergodic Chains

- We can try to get a fundamental matrix as in the absorbing case but $(I-P)^{-1}$ does not exist because $P\mathbf{1}=\mathbf{1}$ (Perron-Frobenius)
- $I+Q+Q^2+\ldots=(I-Q)^{-1}$ converges because $Q^n \to 0$
- Try $I+(P-\mathbf{1}w^T)+(P^2-\mathbf{1}w^T)+\ldots$ because $P^n \to \mathbf{1}w^T$ (Perron-Frobenius)
- Note that $P\mathbf{1}w^T=\mathbf{1}w^T$ and $(\mathbf{1}w^T)^2=\mathbf{1}w^T\mathbf{1}w^T=\mathbf{1}w^T$

$$(P-\mathbf{1}w^T)^n=\sum_{i=0}^{n}(-1)^i\binom{n}{i}P^{n-i}(\mathbf{1}w^T)^i=P^n+\sum_{i=1}^{n}(-1)^i\binom{n}{i}(\mathbf{1}w^T)^i$$

$$=P^n+\underbrace{\left[\sum_{i=1}^{n}(-1)^i\binom{n}{i}\right]}_{(1-1)^n-1}(\mathbf{1}w^T)=P^n-\mathbf{1}w^T$$

- Thus, the following inverse exists:

$$I+\sum_{n=1}^{\infty}(P^n-\mathbf{1}w^T)=I+\sum_{n=1}^{\infty}(P-\mathbf{1}w^T)^n=(I-P+\mathbf{1}w^T)^{-1}$$

Fundamental Matrix for Ergodic Chains

- **Fundamental matrix**: $Z^E := (I - P + \mathbf{1}w^T)^{-1}$ where P is the transition matrix and w is the stationary distribution.
- **Properties**: $w^T Z^E = w^T$, $Z^E \mathbf{1} = \mathbf{1}$, and $Z^E(I - P) = I - \mathbf{1}w^T$
- **Mean first passage time**: $m_{ij} := \mathbb{E}\left[\tau_j \mid x_0 = i\right] = \frac{Z^E_{jj} - Z^E_{ij}}{w_j}$

Example: Land of Oz

- Transition matrix:

$$P = \begin{bmatrix} 0.5 & 0.25 & 0.25 \\ 0.5 & 0 & 0.5 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}$$

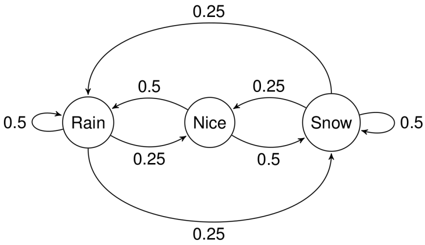
- Stationary distribution:

$$w^T = \begin{bmatrix} 0.4 & 0.2 & 0.4 \end{bmatrix}$$

- Fundamental matrix:

$$I - P + \mathbf{1}w^T = \begin{bmatrix} 0.9 & -0.05 & 0.15 \\ -0.1 & 1.2 & -0.1 \\ 0.15 & -0.05 & 0.9 \end{bmatrix}$$

$$Z^E = \begin{bmatrix} 1.147 & 0.04 & -0.187 \\ 0.08 & 0.84 & 0.08 \\ -0.187 & 0.04 & 1.147 \end{bmatrix}$$



- Mean first passage time:

$$m_{12} = \frac{Z^E_{22} - Z^E_{12}}{w_2} = \frac{0.84 - 0.04}{0.2} = 4$$