1) Basic concepts

Melih Kandemir

University of Southern Denmark
Department of Mathematics and Computer Science (IMADA)

February 6, 2025

Supervised learning

- *Setup:* Observation space $\mathcal{X}$ and label space $\mathcal{Y}$ and a map $f : \mathcal{X} \to \mathcal{Y}$ called a labeling function:

$$\underbrace{x}_{\text{observation}} \to \underbrace{f(\cdot)}_{\text{labeling function}} \to \underbrace{y}_{\text{label}}$$

- *Data:* $\mathcal{D}_n = \{(x_i, y_i) : i = 1, \ldots, n\}$ called a training set.
- *Problem:* Devise an algorithm $\mathbb{A}(\mathcal{D}_n)$ that returns a *predictor* $\hat{f} : \mathcal{X} \to \mathcal{Y}$ such that the generalization error $\mathbb{E}_x[\ell(\hat{f}(x), f(x))]$ is minimum for some loss function ℓ suitable to the output space.
- *Dilemma:* Bias (finding abstractions from individual observations) versus variance (accurately predicting individual observations)

Reinforcement learning: What?

- *Setup:* State space $\mathcal{S}$ and action space $\mathcal{A}$

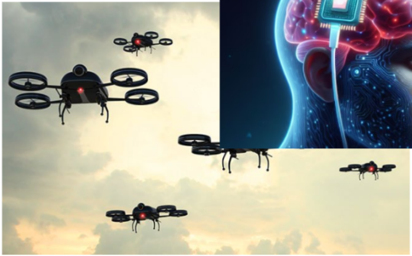
$$\underbrace{s}_{\text{state}}, \underbrace{a}_{\text{action}} \rightarrow \text{environment} \rightarrow \underbrace{r}_{\text{reward}}, \underbrace{s'}_{\text{next state}}$$

- *Data:* $\mathcal{D}_n = \{(s_i, a_i, r_i, s'_i) : i = 1, \ldots, n\}$ called a *replay buffer*.
- *Problem:* Devise an algorithm $\mathbb{A}(\mathcal{D}_n)$ that returns a *policy (agent)* $\pi : \mathcal{S} \rightarrow \mathcal{A}$ such that the total observed reward $\mathbb{E}_s[\sum_{t=1}^{\infty} r_t]$ is maximum.
- *Dilemma:* Exploration (knowing the environment better) versus exploitation (getting maximum reward with the current environmental knowledge).
- *Synonyms:* Approximate Dynamic Programming (ADP), Neuro-Dynamic Programming (NDP)

Reinforcement Learning: Why? (user view)

We already have intelligent data processors. Next step is to have intelligent *agents*.




Reinforcement learning: Why? (expert view)

The reward hypothesis: All machine learning setups can be described as a reward-based learning scheme.

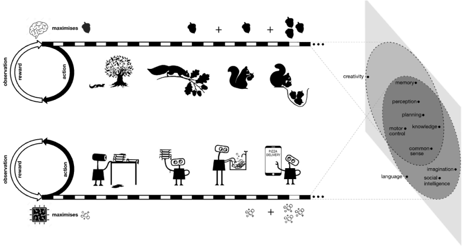
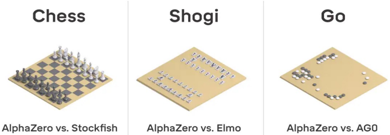


Fig. 1. The *reward-is-enough* hypothesis postulates that intelligence, and its associated abilities, can be understood as subserving the maximisation of reward by an agent acting in its environment. For example, a squirrel acts so as to maximise its consumption of food (top, reward depicted by acorn symbol), or a kitchen robot acts to maximise cleanliness (bottom, reward depicted by bubble symbol). To achieve these goals, complex behaviours are required that exhibit a wide variety of abilities associated with intelligence (depicted on the right as a projection from an agent's stream of experience onto a set of abilities expressed within that experience).

Figure: D. Silver et al., Reward is enough, Artif. Intl., 2021



History

- 1962: Checkers at human level (Arthur Samuel)
- 1992: Backgammon at super-human level (Gerald Tesauro). Uses neural networks for temporal difference learning. The model invented new openings adopted by grandmaster later.
- 1996: Chess at super-human level (IBM, Deep Blue), but NOT with RL.
- 2015: Go at super-human level (DeepMind, AlphaGo), reusing Tesauro's ideas on improved hardware.
- 2022: Continual improvement of large language models from human feedback (OpenAI, RLHF)
- 2025: Large language models with reasoning capabilities (DeepSeek, GRPO)

Discrete Dynamic Systems

$$s_{t+1} = f_t(s_t, a_t), \qquad t = 0, 1, \ldots, T-1$$

where

- t is the time index
- $s_t \in \mathcal{S}_t$ is the state at time t and $\mathcal{S}_t$ is the set of possible states at time t
- $a_t \in \mathcal{A}_t$ is the action (control variable) at time t and $\mathcal{A}_t$ is the set of possible actions at time t
- $f_t : \mathcal{S}_t \times \mathcal{A}_t \rightarrow \mathcal{S}_{t+1}$ is the state transition function that characterizes the environment dynamics.
- $T > 0$ is the time horizon of the system.

Example: Linear dynamic systems

$$s_{t+1} = As_t + Ba_t \tag{1}$$

where $s_t \in \mathbb{R}^n$, $a_t \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times n}$, and $A \in \mathbb{R}^{n \times m}$.

Basic concepts

- The above system is *time-varying* because $f_t, \mathcal{S}_t, \mathcal{A}_t$ depend on t. A *time-invariant* system would look as below:

$$s_{t+1} = f(s_t, a_t), \qquad t = 0, 1, \ldots, T-1$$

  where $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$.
- The state-action space is *finite (a.k.a. discrete or tabular)* if

$$\mathcal{S}_t = \{1, 2, \ldots, n_t\} \quad \text{and} \quad \mathcal{A}_t(s) = \{1, 2, \ldots, m_t(s)\}, s \in \mathcal{S}_t$$

  for all time steps t.
- A sequence $h_T = (s_0, a_0, \ldots, s_{T-1}, a_{T-1}, s_T)$ such that $a_t \in \mathcal{A}_t$ and $s_{t+1} = f_t(s_t, a_t)$ for $t \in \{0, \ldots, T-1\}$ is called a *feasible path.*

Feasible path

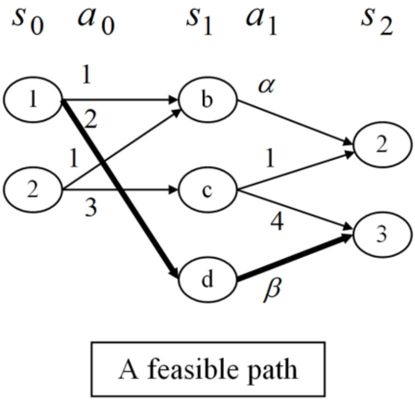


Figure is taken from Mannor et al.

Finite horizon decision problem

Total reward: (a.k.a. cumulative reward) Given a feasible path h_t

$$V_t(h_t) := \sum_{t=0}^{T-1} r_t(s_t, a_t) + r_T(s_T)$$

where

- $r_t(s_t, a_t)$ is the *instantaneous (single-stage)* reward at stage t,
- r_T is the *terminal* reward.

T-stage finite horizon problem: Find a feasible h_T^* such that

$$h_T^* = \arg\max_{h_T} V_T(h_T).$$

Such h_T^* is called an *optimal path* from s_0.

Policy

A *deterministic* control policy is

- *History-dependent* if $a_t = \pi_t(h_t)$ such that $\pi_t \in \Pi_{HD}$
- *Markov* if $a_t = \pi_t(s_t)$ such that $\pi_t \in \Pi_{MD}$
- *Stationary* if $a_t = \pi(s_t)$ such that $\pi \in \Pi_{SD}$
- Note that $\Pi_{HD} \supset \Pi_{MD} \supset \Pi_{SD}$.

A *stochastic* (a.k.a. randomized, probabilistic) control policy is

- *History-dependent* if $P(a_t = a|h_t) = \pi_t(a|h_t)$ such that $\pi_t \in \Pi_{HS}$
- *Markov* if $P(a_t = a|s_t) = \pi_t(a|s_t)$ such that $\pi_t \in \Pi_{MS}$
- *Stationary* if $P(a_t = a|s_t) = \pi(a|s_t)$ such that $\pi \in \Pi_{SS}$
- Note that $\Pi_{HS} \supset \Pi_{MS} \supset \Pi_{SS}$.

where $P(\cdot|\cdot)$ defines conditional probability.

Policy versus a feasible path

- Policy specifies an action for each state.
- Path specifies an action only for the states on the path.
- *Induced path* of a policy π_t is a path h_T^π such that $a_t = \pi_t(h_t)$ for all $t \in [T]$.

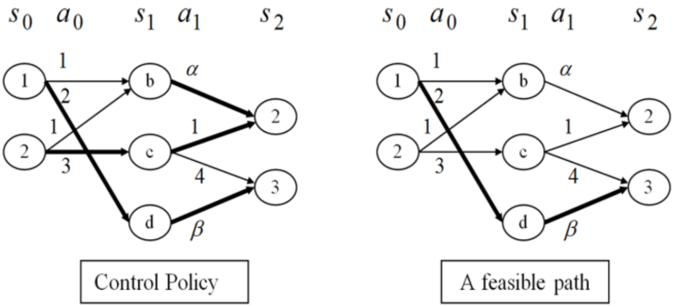


Figure is taken from Mannor et al.
Notation: $[T] = \{0, 1, 2, \ldots, T-1\}$.

Reduction between policy classes

Define state-action probability of time step t induced by a policy π as

$$\begin{aligned}\rho_t^\pi(s,a) &:= P(a_t = a, s_t = s | h_{t-1}^\pi) \\ &= \mathbb{E}_{h_{t-1}^\pi}[\mathbb{I}(s_t = s, a_t = a) | h_{t-1}^\pi]\end{aligned}$$

and plug into the definition of total reward

$$\begin{aligned}\mathbb{E}[V_T(h_T^\pi)] &= \sum_{t=0}^{T-1} \sum_{(s,a)\in\mathcal{S}_t\times\mathcal{A}_t} r_t(s,a)\rho_t^\pi(s,a) \\ &=: \mathbb{E}[V^\pi(s_0)]\end{aligned}$$

where $V^\pi(s_0)$ is the *reward-to-go* for state s_0 at time 0 when policy π is executed. Hence, the expected total reward of two policies π and π' will be equal if and only if $\rho_t^\pi(s,a) = \rho_t^{\pi'}(s,a)$.
Notation: $\mathbb{I}(\cdot)$ is the indicator function and $\mathbb{E}[\cdot]$ is the expectation.

From Π_{HS} to Π_{MS}

Theorem

For any stochastic history-dependent policy $\pi \in \Pi_{HS}$ *there exists a stochastic Markov policy* $\pi' \in \Pi_{MS}$ *such that* $\rho_t^{\pi}(s,a) = \rho_t^{\pi'}(s,a)$ *for all* $(s,a) \in \mathcal{S}_t \times \mathcal{A}_t$*, which implies*

$$\mathbb{E}[V^{\pi}(s_0)] = \mathbb{E}[V^{\pi'}(s_0)]$$

Proof sketch. Choose

$$\pi_t'(a|s) = \frac{\rho_t^{\pi}(s,a)}{\sum_{a' \in \mathcal{A}_t} \rho_t^{\pi}(s,a')},$$

define $\rho_t^{\pi}(s_0) := P(s_t = s | h_{t-1}^{\pi})$ and apply induction.

From Π_{MS} to Π_{MD}

Theorem

For any stochastic Markov policy $\pi \in \Pi_{MS}$ *there exists a better deterministic Markov policy* $\pi' \in \Pi_{MD}$ *in the sense that*

$$\mathbb{E}[V^{\pi}(s_0)] \leq \mathbb{E}[V^{\pi'}(s_0)].$$

Proof sketch. Backward induction by claim: *For any policy* $\pi \in \Pi_{MS}$ *which is deterministic in* $[t+1, T]$ *there is a policy* $\pi' \in \Pi_{MS}$ *which is deterministic in* $[t, T]$ *and* $\mathbb{E}[V^{\pi}(s_0)] \leq \mathbb{E}[V^{\pi'}(s_0)]$. Base case $t = T$ holds trivially. In the induction step do

$$\pi'_t(s_t) = \arg\max_{a \in \mathcal{A}_t} r_t(s_t, a) + V^{\pi}(f_t(s_t, a))$$

which would satisfy

$$\begin{aligned} \mathbb{E}[V^{\pi}(s_t)] &= \mathbb{E}_{h_t^{\pi}}[\mathbb{E}_{a_t \sim \pi}[r_t(s_t, a_t) + V^{\pi}(f_t(s_t, a_t))]] \\ &\leq \mathbb{E}_{h_t^{\pi}}\left[\max_{a_t \in \mathcal{A}_t} r_t(s_t, a_t) + V^{\pi}(f_t(s_t, a_t))\right] = \mathbb{E}[V^{\pi'}(s_t)]. \end{aligned}$$

Optimal control policies

Definition

A control policy $\pi \in \Pi_{MD}$ is called *optimal* if for each $s_0 \in \mathcal{S}_0$ it holds that $V^{\pi}(s_0) \geq V^{\pi'}(s_0)$ for any other $\pi' \in \Pi_{MD}$.

T**-stage finite-horizon planning problem:** Find the optimal π for a T-stage deterministic dynamical system.

Brute-force search: Assume $|\mathcal{S}_t| = n$ and $|\mathcal{A}_t(s)| = m$. Then we need to consider m^{nT} policies. When $T = n = m = 10$, this amounts to 10^{100} policies! Dynamic programming will speed up the search.

Finite horizon dynamic programming

- Dynamic Programming (DP) breaks down the T-stage problem into T sequential single-stage optimization problems.
- DP builds on *Bellman's Principle of Optimality:*

> The tail of an optimal policy is optimal for the tail problem.

- The same principle does not hold for the head problem!
- The essence of DP is to apply this principle recursively from the last stage backwards.

Remark: Tail problem is defined with respect to a single starting state.

The DP algorithm

Algorithm Finite-horizon Dynamic Programming

$V_T(s) = r_T(s)$ for all $s \in \mathcal{S}_T$
for all $t = T-1, \ldots, 0$ **do**
    Compute $V_t(s) = \max_{a \in \mathcal{A}_t} \{r_t(s,a) + V_{t+1}(f_t(s,a))\}$ for all $s \in \mathcal{S}_t$
end for
return $\pi_t^*(s) \in \arg\max_{a \in \mathcal{A}_t}\{r_t(s,a) + V_{t+1}(f_t(s,a))\}$ for $t \in [T]$.

In the algorithm above

- $V_t : \mathcal{S}_t \to \mathbb{R}$ are called the *value functions* that are calculated recursively.
- Thanks to the value functions, the algorithm visits each state exactly once!
- The step $V_t(s) = \max_{a \in \mathcal{A}_t} \{r_t(s,a) + V_{t+1}(f_t(s,a))\}$ is called the *Bellman equation*.
- It is guaranteed that (π_t^*) is optimal and $V_0(s) = \max_\pi V^\pi(s), \forall s \in \mathcal{S}_0$.

Example: Run the algorithm on this decision graph

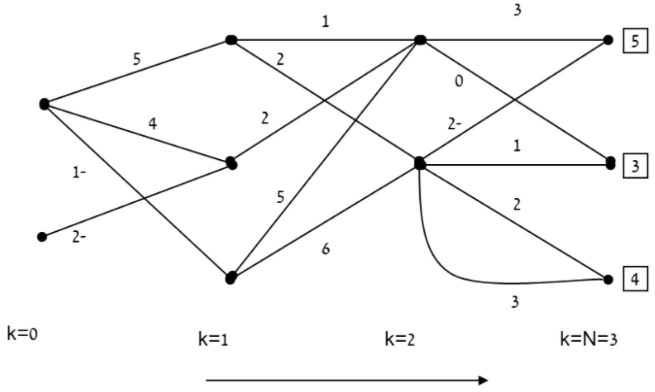


Figure taken from Mannor et al.

Average reward criteria

The aim is to maximize the expectation of:

$$R_{avg} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} r_t(s_t, a_t).$$

Any finite prefix has no influence on the final average reward. For any DDP, the optimal average reward is reached by a policy that cycles around a simple cycle. The maximum average reward is the total reward of this simple cycle.

Linear quadratic regulator (LQR)

This is a continuous optimal control method that assumes that dynamics are linearly and costs are quadratically dependent on the states and actions:

$$
\begin{aligned}
&\min_{a_0,\ldots,a_T} \sum_{t=0}^{T} c_t(s_t, a_t), \\
&\text{s.t.} \quad s_{t+1} = A_t s_t + B_t a_t, \\
&\qquad\;\; c_t = s_t^\top Q_t s_t + a_t^\top R_t a_t, \quad \forall t = 0, \ldots, T-1, \\
&\qquad\;\; c_T = s_T^\top Q_T s_T.
\end{aligned}
$$

where s_0 is given, $Q_t = Q_t^\top \geq 0$ is a symmetric non-negative definite state-cost matrix (i.e. $v^\top Q_t v \geq 0, \forall v \in \mathbb{R}^n$), and $R_t = R_t^\top > 0$ is a symmetric positive definite control-cost matrix (i.e. $v^\top R_t v > 0, \forall v \in \mathbb{R}^m$). Let $V_t(s)$ denote the value function of a state at time t, that is,

$$
V_t(s) = \min_{a_t,\ldots,a_T} \sum_{t'=t}^{T} c_{t'}(s_{t'}, a_{t'}) \quad \text{s.t.} \quad s_t = s.
$$

DP solution to LQR

Theorem

The value function has a quadratic form: $V_t(s) = s^\top P_t s$*, and* $P_t = P_t^\top$.

Proof.

For $t = T$, by definition, as $V_T(s) = s^\top Q_T s$. Assume $V_{t+1}(s) = s^\top P_{t+1} s$, then

$$\begin{aligned}
V_t(s) &= \min_{a_t} s^\top Q_t s + a_t^\top R_t a_t + V_{t+1}(A_t s + B_t a_t) \\
&= \min_{a_t} s^\top Q_t s + a_t^\top R_t a_t + (A_t s + B_t a_t)^\top P_{t+1}(A_t s + B_t a_t) \\
&= s^\top Q_t s + (A_t s)^\top P_{t+1}(A_t s) \\
&\qquad + \min_{a_t} a_t^\top (R_t + B_t^\top P_{t+1} B_t) a_t + 2(A_t s)^\top P_{t+1}(B_t a_t)
\end{aligned}$$

Solving the minimization gives $a_t^* = -(R_t + B_t^\top P_{t+1} B_t)^{-1} B_t^\top P_{t+1} A_t s$.
Substituting back a_t^* into $V_t(s)$ gives a quadratic expression in s. □

Markov Chains

Definition

A Markov chain $\{X_t : t \in \mathbb{N}^+\}$, with $X_t \in \mathcal{X}$, is a discrete-time stochastic, process, over a finite or countable state-space $\mathcal{X}$, that satisfies the following Markov property:

$$P(X_{t+1} = j | X_t = i, X_{t-1}, \dots X_0) = P(X_{t+1} = j | X_t = i)$$

We focus on time-homogeneous Markov chains, where:

$$P(X_{t+1} = j | X_t = i) = P(X_1 = j | X_0 = i) \overset{\Delta}{=} p_{ij}.$$

Define $p_{ij}^{(m)} = P(X_m = j | X_0 = i)$, the m-step transition probabilities, then

$$p_{ij}^{(m)} = [P^m]_{ij}$$

where P^m is the m-th power of the matrix P.

Some definitions

Definition

State j is **accessible** (or reachable) from i (denoted by $i \to j$) if $p_{ij}^{(m)} > 0$ for some $m \geq 1$.

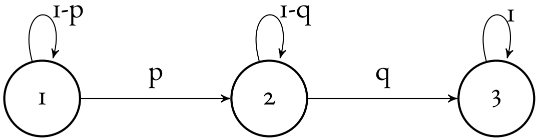
Construct a directed graph $G(X, E)$ where $E = \{(i, j) : p_{ij} > 0\}$ and find a directed path from i to j. Remarks:

- The relation is transitive. If $i \to j$ and $j \to k$ then $i \to k$.
- If $i \to j$ then $\exists m_1$ s.t. $p_{ij}^{(m_1)} > 0$. When $j \to k$ where $\exists m_2$ s.t. $p_{jk}^{(m_2)} > 0$, we also have $p_{ik}^{(m_1+m_2)} \geq p_{ij}^{(m_1)} p_{jk}^{(m_2)} > 0$.

States i and j are **communicating** states if $i \to j$ and $j \to i$.

Example

Transition diagram:



Adjacency matrix:

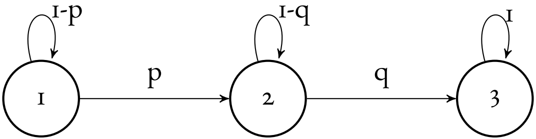
$$P = \begin{pmatrix} 1-p & p & 0 \\ 0 & 1-q & q \\ 0 & 0 & 1 \end{pmatrix}$$

where $P(s_{t+1} = j | s_t = i) = [P]_{ij} =: p_{ij}$.

$$\begin{aligned} P(s_{t+2} = j | s_t = i) &= \sum_k P(s_{t+2} = j | s_{t+1} = k) P(s_{t+1} = k | s_t = i) \\ &= p_{ik} p_{kj} \end{aligned}$$

Then the matrix of $P(s_{t+2} = j | s_t = i)$'s can be expressed as $P \cdot P = P^2$.

P^2



$$P^2 = \begin{pmatrix} (1-p)^2 & p(2-p-q) & pq \\ 0 & (1-q)^2 & q(2-q) \\ 0 & 0 & 1 \end{pmatrix}$$

hence

- 3 is accessible from 1 as $pq > 0$.
- As the matrix is triangular, no pair of states are communicating.

Notation: $p_{ij}^{(2)} := [P_{ij}^2]$.

Some definitions

Definition

A **communicating class** (or just class) is a maximal collection of states that communicate.

Definition

The Markov chain is **irreducible** if all states belong to a single class (i.e., all states communicate with each other).

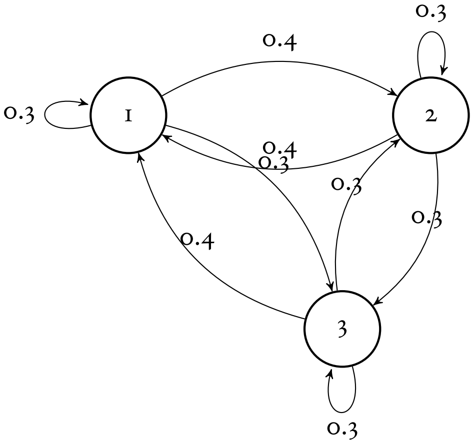
Definition

State i is **recurrent** if $\exists m$ such that $p_{ii}^{(m)} = 1$. Otherwise, state i is **transient**.

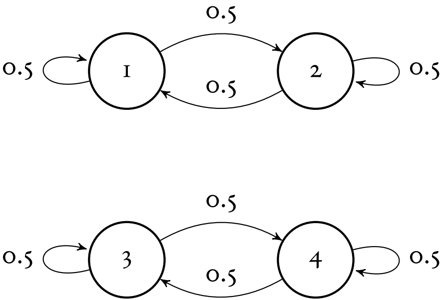
Definition

State i has a **period** $d_i = \mathrm{GCD}\{m \geq 1 : p_{ii}^{(m)} > 0\}$, where GCD is the greatest common divisor. A state is **aperiodic** if $d_i = 1$.

Irreducible aperiodic

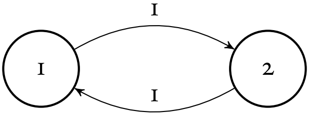


Non-irreducible

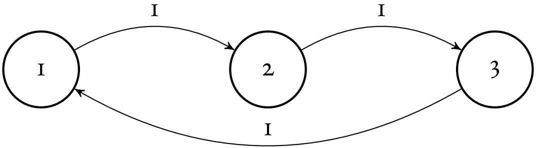


Periodic

Period 2



Period 3



Some results

Theorem

State i is transient if and only if $\sum_{m=1}^{\infty} p_{ii}^{(m)} < \infty$ *and recurrent if and only if* $\sum_{m=1}^{\infty} p_{ii}^{(m)} = \infty$.

Theorem

Recurrence is a class property in the sense that if i and j are communicating and i is recurrent, then j is also recurrent.

Theorem

If i and j are in the same recurrent class, then $\exists m$ such that $p_{ij}^{(m)} = 1$.

Theorem

For any two states i and j with periods d_i and d_j, in the same communicating class, we have $d_i = d_j$.

Stationary distribution

The probability vector $\mu = (\mu_i)$ is an *invariant distribution* (or *stationary distribution* or *steady-state distribution*) for the Markov chain if $\mu^\top P = \mu^\top$:

$$\mu_j = \sum_i \mu_i p_{ij}, \quad \forall j.$$

In this case, if $X_t \sim \mu$ then $X_{t+1} \sim \mu$. If $X_0 \sim \mu$, then the Markov chain (X_t) is a stationary stochastic process.

Theorem

Let (X_t) be an irreducible and aperiodic Markov chain over a finite state space $\mathcal{X}$ with transition matrix P. Then there is a unique distribution μ such that $\mu^\top P = \mu^\top > 0$.

Convergence of visitation counts

We define the average fraction that a state $j \in X$ occurs, given that we start with an initial state distribution x_0, as follows:

$$\pi_j^{(m)} = \frac{1}{m} \sum_{t=1}^{m} \mathbb{I}(X_t = j). \tag{2}$$

Theorem

Let (X_t) be an irreducible and aperiodic Markov chain over a finite state space X with transition matrix P. Let μ be the stationary distribution of P. Then, for any $j \in X$ we have,

$$\mu_j = \lim_{m \to \infty} \mathbb{E}[\pi_j^{(m)}] = \frac{1}{\mathbb{E}[T_j]}. \tag{3}$$

where T_i is the return time to state i (number of steps until its next visit).

Note that if i is a recurrent state, then $T_i < \infty$ with probability (w.p.) 1.

Key properties of finite Markov chains

Theorem

Let (X_t) be an irreducible, aperiodic Markov chain over a finite state space X. Then the following properties hold:

1. *All states are positive recurrent, i.e.,* $\mathbb{E}[T_i] < \infty, \forall i \in \mathcal{S}$.
2. *There exists a unique stationary distribution* μ, *where* $\mu(i) = 1/\mathbb{E}[T_i]$.
3. *Convergence to the stationary distribution:* $\lim_{t\to\infty} P[X_t = j] = \mu_j \; (\forall j)$
4. *Ergodicity: For any finite* f*:* $\lim_{t\to\infty} \frac{1}{t} \sum_{s=0}^{t-1} f(X_s) = \sum_i \mu_i f(i) \stackrel{\Delta}{=} \pi \cdot f$.

Remark: A state i with $\mathbb{E}[T_i] = \infty$ is called *null recurrent*.

Reversible Markov chains

Suppose there exists a probability vector $\mu = (\mu_i)$ so that

$$\mu_i p_{ij} = \mu_j p_{ji}, \quad i, j \in X.$$

These equations are called the **detailed balance equations**. It is then easy to verify by direct summation that μ is an invariant distribution for the Markov chain defined by $(p_{i,j})$. This follows since

$$\sum_i \mu_i p_{ij} = \sum_i p_{ji} \mu_j = \mu_j$$

A Markov chain that satisfies these equations is called **reversible**.

Mixing time

The mixing time measures how fast the Markov chain converges to the steady state distribution. We first define the *Total Variation (TV) distance* between distributions D_1 and D_2 as:

$$\|D_1 - D_2\|_{TV} = \max_{B \subseteq \mathcal{X}} \{D_1(B) - D_2(B)\} = \frac{1}{2} \sum_{x \in \mathcal{X}} |D_1(x) - D_2(x)|$$

The mixing time τ is defined as the time to reach a total variation of at most $1/4$:

$$\|s_0 P^\tau - \mu\|_{TV} = \|p^{(\tau)} - \mu\|_{TV} \leq \frac{1}{4} \|s_0 - \mu\|_{TV}$$

where μ is the steady state distribution and $p^{(\tau)}$ is the state distribution after τ steps starting with an initial state distribution s_0.
Note that after 2τ time steps we have

$$\|s_0 P^{2\tau} - \mu\|_{TV} = \|p^{(\tau)} P^\tau - \mu\|_{TV} \leq \frac{1}{4} \|p^{(\tau)} - \mu\|_{TV} \leq \frac{1}{4^2} \|s_0 - \mu\|_{TV}.$$

Mixing time

After $k\tau$ time steps we have

$$\begin{aligned}\|s_0 P^{k\tau} - \mu\|_{TV} &= \|p^{((k-1)\tau)} P^{\tau} - \mu\|_{TV} \\ &\leq \frac{1}{4}\|p^{((k-1)\tau)} - \mu\|_{TV} \\ &\leq \frac{1}{4^k}\|s_0 - \mu\|_{TV}.\end{aligned}$$

where the formal proof is by induction on $k \geq 1$.