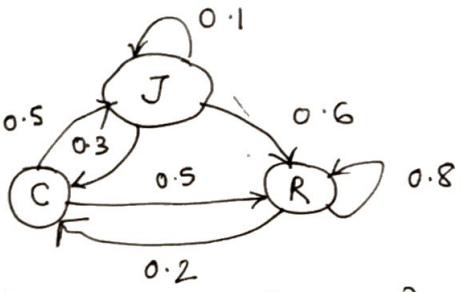
P1.1 Time homogeneous Markov chain $(X_t)_{t=0}^{\infty}$

Starts with Jazz
so initially $t = 0$
it's at J with prob 1



State space $S = \{Jazz, Rock, Country\}$

$$\mu_0 = [1, 0, 0]$$

Transition matrix P in order J, R, C

$R1 \rightarrow P_{J\to J} = 0.1, \; P_{J\to R} = 0.6, \; P_{J\to C} = 0.3$

$R2 \rightarrow P_{R\to J} = 0.0, \; P_{R\to R} = 0.8, \; P_{R\to C} = 0.2$

$R3 \rightarrow P_{C\to J} = 0.5, \; P_{C\to R} = 0.5, \; P_{C\to C} = 0.0$

$$P = \begin{bmatrix} 0.1 & 0.6 & 0.3 \\ 0.0 & 0.8 & 0.2 \\ 0.5 & 0.5 & 0.0 \end{bmatrix}$$

Answer

P1.2 (a) (J, C, C, R):

$P(X_0 = J) = 1, \quad P(X_1 = C \mid X_0 = J) = 0.3, \quad P(X_2 = C \mid X_1 = C) = 0.0$

$P(X_3 = R \mid X_2 = C) = 0.5$

$\therefore$ Prob $= 1 \times 0.3 \times 0.0 \times 0.5 = 0$

P1.2 (b) (J, R, R, C, J)

$P(X_0 = J) = 1, \quad P(X_1 = R \mid X_0 = J) = 0.6, \quad P(X_2 = R \mid X_1 = R) = 0.8$

$P(X_3 = C \mid X_2 = R) = 0.2, \quad P(X_4 = J \mid X_3 = C) = 0.5$

$\therefore$ Prob $= 1 \times 0.6 \times 0.8 \times 0.2 \times 0.5 = 0.048$

P1.2 (c) $(J, R, C, R, C, \ldots, R, C)$

$X_0 \; X_1 \; X_2 \; X_3 \; X_4 \quad X_{2k-1} \; X_{2k}$

$X_0 = J, \quad X_{2k-1} = R, \quad X_{2k} = C$

Total $2m+1$ states $\quad k \in \{1, \ldots m\}$

$$Prob = \prod_{t=0}^{2m-1} P\left(X_{t+1} \mid X_t\right)$$

$X_0 = J$, $X_1 = R$. $\quad P(X_1 = R \mid X_0 = J) = 0.6$

$X_1 = R$, $X_2 = C$, $X_3 = R$, $X_4 = C \ldots X_{2m-1} = R$, $X_{2m} = C$

$\therefore$ m times $R \to C \quad P[C \mid R) = 0.2$

$\therefore$ (m times $C \to R \quad P(R \mid C] = 0.5$
$\therefore$ (m −1)

$$\therefore Prob = 0.6 \times (0.2)^m \times (0.5)^{m-1}$$

$$= 0.6 \times 0.2 \times (0.2 \times 0.5)^{m-1} = 0.12 \times (0.1)^{m-1}$$

Ans

P1.2

(d) $(J, R, C, R, C \ldots)$

$X_0 = J$, $X_{2k-1} = R$, $X_{2k} = C$ $\quad k \in \mathbb{N} = \{1, 2, \ldots\}$

as $m \to \infty$

$$P(m \to \infty) = \lim_{m \to \infty} \left[0.12 \times (0.1)^{m-1}\right] = 0.12 \times 0 = 0$$

as $m \to \infty \quad (0.1)^{m-1} \to 0$

$\therefore P(m \to \infty) \to 0.$

P1.3 We have $\mu_t = \mu_0 P^t$ here $t = 2$ $\therefore \mu_2 = \mu_0 P^2$

$$\therefore P^2 = P \times P = \begin{bmatrix} 0.1 & 0.6 & 0.3 \\ 0.0 & 0.8 & 0.2 \\ 0.5 & 0.5 & 0.0 \end{bmatrix} \begin{bmatrix} 0.1 & 0.6 & 0.3 \\ 0.0 & 0.8 & 0.2 \\ 0.5 & 0.5 & 0.0 \end{bmatrix}$$

$$R_1 = [0.1 \times 0.1 + 0.6 \times 0 + 0.3 \times 0.5 \quad 0.1 \times 0.6 + 0.6 \times 0.8 + 0.3 \times 0.5 \quad 0.1 \times 0.3 + 0.6 \times 0.2 + 0.3 \times 0]$$

$$R_2 = [0 + 0 + 0.2 \times 0.5 \quad 0 + 0.8 \times 0.8 + 0.2 \times 0.5 \quad 0 + 0.8 \times 0.2 + 0]$$

$$R_3 = [0.5 \times 0.1 + 0 + 0 \quad 0.5 \times 0.6 + 0.5 \times 0.8 + 0 \quad 0.5 \times 0.3 + 0.5 \times 0.2 + 0]$$

$$P^2 = \begin{bmatrix} 0.01+0+0.15 & 0.06+0.48+0.15 & 0.03+0.12+0 \\ 0+0+0.1 & 0+0.64+0.1 & 0+0.16+0 \\ 0.05+0+0 & 0.3+0.4+0 & 0.15+0.1+0 \end{bmatrix}$$

$$P^2 = \begin{bmatrix} 0.16 & 0.69 & 0.15 \\ 0.10 & 0.74 & 0.16 \\ 0.05 & 0.70 & 0.25 \end{bmatrix}$$

$$\mu_2 = \underset{1\times3}{[1, 0, 0]} \quad \underset{3\times3}{\begin{bmatrix} 0.16 & 0.69 & 0.15 \\ 0.10 & 0.74 & 0.16 \\ 0.05 & 0.70 & 0.25 \end{bmatrix}} = [0.16$$

μ_0 P^2

$\therefore P(X_2 = J) = 0.16$

$P(X_2 = R) = 0.69$

$P(X_2 = C) = 0.15$

$\therefore$ P of third song being Jazz, Rock, Country are 0.16, 0.69 & 0.15 respectively (Ans)

P.1.4
Answer → See code & answer in Markdown MKD.

P1.5
Ans → See code & answer in MkD

P1.6
Ans → See code & answer MkD

P2.1 Original space $\{1,2\}$

Original 2nd order MM transition prob are, $t \geq 2$

$P(X_t = 1 \mid X_{t-2} = 1, X_{t-1} = 1) = 0.8 \therefore P(X_t = 2 \mid X_{t-2} = 1, X_{t-1} = 1) = 0.2$

$P(X_t = 1 \mid X_{t-2} = 1, X_{t-1} = 2) = 0.1 \therefore P(X_t = 2 \mid X_{t-2} = 1, X_{t-1} = 2) = 0.9$

$P(X_t = 1 \mid X_{t-2} = 2, X_{t-1} = 1) = 0.3 \therefore P(X_t = 2 \mid X_{t-2} = 2, X_{t-1} = 1) = 0.7$

$P(X_t = 1 \mid X_{t-2} = 2, X_{t-1} = 2) = 0.7 \quad P(X_t = 2 \mid X_{t-2} = 2, X_{t-1} = 2) = 0.3$

$P(X_1 = 1 \mid X_0 = 1) = 0.2 \quad \therefore P(X_1 = 2 \mid X_0 = 1) = 0.8$

$P(X_1 = 1 \mid X_0 = 2) = 0.4 \quad \therefore P(X_1 = 2 \mid X_0 = 2) = 0.6$

$t = 1$, $P(X_0 = 1) = 0.5$, $P(X_0 = 2) = 0.5$ also

equivalent 1st order processes:

Let $Y_t = (X_{t-1}, X_t)$, $t \geq 1 \Rightarrow Y_{t+1} = (X_t, X_{t+1})$

All possible pairs of original state $\{1,2\}$ $S' = \{(1,1), (1,2), (2,1), (2,2)\}$

① $(1,1) \to (1,1)$ & $(1,1) \to (1,2)$

$P(X_{t+1} = 1 \mid (1,1)) = 0.8$, $P(X_{t+1} = 2 \mid (1,1)) = 0.2$ Given,

$X_{t-1} = 1$, $X_t = 1$, $Y_{t+1} = (1, Y_{t+1})$

if $X_{t+1} = 1$, new $(1,1)$

$X_{t+1} = 2$, new $(1,2)$

$\therefore$ Transition prob $\begin{cases} (1,1) \to (1,1) \text{ is } 0.8 \\ (1,1) \to (1,2) \text{ is } 0.2 \end{cases}$

Next,

② $X_{t-1} = 1$, $X_t = 2$, $Y_{t+1} = (2, X_{t+1})$

$\therefore$ if $X_{t+1} = 1$, new state $(2,1)$

if $X_{t+1} = 2$, " " $(2,2)$

$P(X_{t+1} = 1 \mid (1,2)) = 0.1$, $P(X_{t+1} = 2 \mid (1,2)) = 0.9$

$\therefore$ transition prob $\begin{cases} (1,2) \to (2,1) \text{ is } 0.1 \\ (1,2) \to (2,2) \text{ is } 0.9 \end{cases}$

(3) Next

$X_{t-1} = 2, X_t = 1 \Rightarrow Y_{t+1} = (1, X_{t+1})$ ∴ if $X_{t+1} = 1$, new state $(1,1)$

& $X_{t+1} = 2$, new state $(1,2)$

$\mathbb{P}(X_{t+1} = 1 \mid (2,1)) = 0.3$, $\mathbb{P}(X_{t+1} = 2 \mid (2,1)) = 0.7$

∴ transition prob $(2,1) \to (1,1)$ is 0.3
$(2,1) \to (1,2)$ is 0.7

(4) Next

$X_{t-1} = 2, X_t = 2 \Rightarrow Y_{t+1} = (2, X_{t+1})$

So $X_{t+1} = 1$, new state $(2,1)$

$X_{t+1} = 2$, " " $(2,2)$

$\mathbb{P}(X_{t+1} = 1 \mid (2,2)) = 0.7$, $\mathbb{P}(X_{t+1} = 2 \mid (2,2)) = 0.3$

∴ transition prob $(2,2) \to (2,1)$ is 0.7
$(2,2) \to (2,2)$ is 0.3

So in order $P = \begin{bmatrix} 0.8 & 0.2 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.1 & 0.9 \\ 0.3 & 0.7 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.7 & 0.3 \end{bmatrix}$ Ans

Now initial state distribution is $\pi = [0.1, 0.4, 0.2, 0.3]$ Ans

For $(1,1)$ $X_0 = 1, X_1 = 1$

$\mathbb{P}(1,1) = \mathbb{P}(X_0 = 1) \cdot \mathbb{P}(X_1 = 1 \mid X_0 = 1) = 0.5 \times 0.2 = 0.1$

For $(1,2)$, $X_0 = 1, X_1 = 2$

$\mathbb{P}(1,2) = \mathbb{P}(X_0 = 1) \cdot \mathbb{P}(X_1 = 2 \mid X_0 = 1) = 0.5 \times 0.8 = 0.4$

For $(2,1)$, $X_0 = 2, X_1 = 1$

$\mathbb{P}(2,1) = \mathbb{P}(X_0 = 2) \cdot \mathbb{P}(X_1 = 1 \mid X_0 = 2) = 0.5 \times 0.4 = 0.2$

For $(2,2)$ $X_0 = 2, X_1 = 2$

$\mathbb{P}(2,2) = \mathbb{P}(X_0 = 2)\, \mathbb{P}(X_1 = 2 \mid X_0 = 2) = 0.5 \times 0.6 = 0.3$

P2.2 Task convert k order MC to 1st order MC.

We define new state Y_t, this is a vector of last k states from original s.t $Y_t = (X_{t-k+1}, X_{t-k+2}, \dots X_{t-1}, X_t)$ State space should be $|S|^k$ where S is original state space.

$Y_t = (X_{t-k+1}, \dots, X_{t-1}, X_t) \rightarrow Y_{t+1} = (X_{t-k+2}, \dots, X_t, X_{t+1})$ allowed transitions.

with transition probabilities

$$\mathbb{P}\left(Y_{t+1} = (X_{t-k+2}, \dots, X_t, X_{t+1}) \mid Y_t = (X_{t-k+1}, \dots, X_t)\right) = \mathbb{P}\left(X_{t+1} = x_{t+1} \mid X_{t-k+1}, \dots, X_t\right)$$

3 $X_5 \mid X_0 \dots X_4$

$X_5 \mid X_2, X_3, X_4$

Initial distribution of Y_t comes from joint distribution.

$$\mathbb{P}\left(Y_{k-1} = (x_0, x_1, \dots, x_{k-1})\right) = \mathbb{P}(X_0 = x_0, X_1 = x_1, \dots X_{k-1} = x_{k-1})$$

P.2.2 k-th order Markov Chain → simple Markov chain

$$\mathbb{P}(X_t \mid X_0, X_1, \dots X_{t-1}) = \mathbb{P}(X_t \mid X_{t-k}, X_{t-k+1}, \dots X_{t-1}) \quad t > k$$

① we convert to 1st order Markov chain.

Define new, $Y_t = (X_{t-k+1}, X_{t-k+2}, \dots, X_t)$ with state space S^k where S is the original state space.

if Y_t is current block, next block is Y_{t+1}

② Transitions allowed, from states $Y_t \rightarrow Y_{t+1}$

ie. $(X_{t-k+1}, \dots X_t) \rightarrow (X_{t-k+2}, \dots X_t, X_{t+1})$ will have probabilities.

$$\mathbb{P}\left(X_{t+1} = x_{t+1} \mid X_{t-k+1} = x_{t-k+1}, \dots, X_t = x_t\right)$$ with all other transition probabilities will be zero.

③ Next we calculate initial distribution Y_t using joint prob chain rule

ie. $\mathbb{P}\left(Y_{k-1} = (x_0 \dots, x_{k-1})\right) = \mathbb{P}\left(X_0 = x_0 \dots, X_{k-1} = x_{k-1}\right)$

So this way every order-k MC can be represented as 1st order MC

P3.1 By definition $\mu = (S, \mu_0, A, P, R, \gamma)$

State space $S = \{s_A, s_H, s_C\}$ for Action, Horror & Comedy.

Initial state distribution $\mu_0 = [1/3, 1/3, 1/3]$ because $\mu_0[A] = \frac{1}{3}$, $\mu_0[H] = \frac{1}{3}$, $\mu_0[C] = 1/3$

Action space $A = [$ Movie A, Movie B, Movie C, Movie D $]$ list of available movies.

Transition function, $P(s'|s,a)$ $P: S \times A \to A(S)$

This depends on if recommendation match user's current genre.

e.g. if no match to desired genres, next state probability 1 & if movie belong/match desired genre of user is chosen uniform randomly from two other genres → This is as per problem so each of the other two genres have probability 0.5

$\therefore s \in S$ & $a \in A$ also let $I(a,s) = 1$ if match & $I(a,s) = 0$ otherwise.

$$P(s'|s,a) = \begin{cases} \end{cases}$$

$P(s'|s,a) = 0.5$ if $I(a,s) = 1$ & $s' \neq s$

$= 0$ if $I(a,s) = 1$ & $s' = s$

$= 1$ if $I(a,s) = 0$ & $s' = s$

$= 0$ if $I(a,s) = 0$ & $s' \neq s$

Last Reward fn $R(s,a) = +1$ if if $I(a,s) = 1$

$R(s,a) = -1$ if $I(a,s) = 0$.

P3.2 The policy is stationary → i.e action selection depends on current state & not on time step or history.

e.g. here if user wants to choose 'Action' the system always will use same current prob distribution i.e. 0.4 for MovA, 0.4 for MovB, 0.2 for MovD without any dependence on time step or previous historical interaction. (Ans)

The policy is nondeterministic & randomized (stochastic)

A policy is stochastic if it assign non zero probabilities to multiple actions for at least one state. (e.g. here

Action: system chooses MovA, MovB or MovD with 0.4, 0.4 & 0.4 respectively.

Horror: MovA, MovB, MovC have equal 0.25 probabilities

Comedy: MovB or MovC with 0.1 & 0.9 probabilities respectively.

So since multiple actions have non-zero probabilities for each state, the policy is randomized or stochastic.

(Ans)

P3.3 Same as before for the Markov chain in part 2

State space $S = \{S_A, S_H, S_C\}$ each state corresponds to the user's desired genre Action, horror or comedy respectively

$\mu_0 = [1/3, 1/3, 1/3]$ in order of Action, Horror & Comedy

$P = P(s): S \to \Delta(S)$ where this can be described using transition matrix

rule is stationary policy $P^{\pi}(s'|s) = \sum_a \pi(a|s)\, P(s'|s,a)$

Policy: if S = Action: π(Movie A) = 0.4, π(Movie B) = 0.4, π(Movie D) = 0.2

if S = Horror: π(Movie A) = 0.25, = π(Movie B) = π(Movie C) = π(Movie D)

if S = Comedy, π(Movie B) = 0.1, π(Movie C) = 0.9

transition matrix is computed (See Code)

$$P^{\pi} = \begin{bmatrix} 0.2 & 0.4 & 0.4 \\ 0.375 & 0.25 & 0.375 \\ 0.45 & 0.45 & 0.10 \end{bmatrix}$$

The reward fn as before $R(s) = \begin{cases} +1 & \text{if } I(a,s) = 1 \\ -1 & I(a,s) = 0 \end{cases}$

expected reward $r^{\pi} = (0.60, 0.50, 0.80) \rightarrow$ (See Code for Calc).
Ac Hor Com

P3.4 See code & answer

$V^\pi \in \mathbb{R}^{|S|}$ value fr.

$S \in \mathcal{S}$ Collection of vertices
$I \in \mathbb{R}^{|S| \times |S|}$
$P^\pi \in \mathbb{R}^{|S| \times |S|}$
$R^\pi \in \mathbb{R}^{|S|}$

P5 1. Given $V^\pi = (I - \gamma P^\pi)^{-1} R^\pi$

Since P^π is row stochastic i.e. all entries are non-negative & sum of each row is 1 & all eigenvalues of P^π $|\lambda|_{eigen} \leq 1$
Let λ be any eigenvalue of matrix P^π. The corresponding eigenvalue of matrix $I - \gamma P^\pi$ is $1 - \gamma\lambda$ as shown below

i.e $P^\pi v = \lambda v$ $\quad v \neq 0$, non zero eigenvector.

$$(I - \gamma P^\pi) v = Iv - \gamma P^\pi v = v - \gamma(\lambda v) = \underline{(1 - \gamma\lambda) v}$$

To prove invertibility we show eigenvalue is non zero for λ. i.e $1 - \gamma\lambda \neq 0$.

Since $|\lambda| \leq 1$ & $0 \leq \gamma < 1$, we can list two conditions

$\lambda = 1$, then $1 - \gamma > 0$ since $\gamma < 1$

& when $\lambda \neq 1$ $\therefore |\lambda| \leq 1$ $\therefore |\gamma\lambda| \leq \gamma \cdot |\lambda| \leq \gamma < 1$

$\therefore \gamma\lambda \neq 1$ $\therefore$ $1 - \gamma\lambda \neq 0$

So no eigenvalue of $(I - \gamma P^\pi)$ is zero so $\det(I - \gamma P^\pi) \neq 0$

So $I - \gamma P^\pi$ has all non zero eigenvalues.
$\therefore$ matrix is invertible. Q.E.D

P5.2 Beginning with Bellman's eq: stochastic stationary!

$$V^\pi(s) = \mathbb{E}_{a \sim \pi(s)} \Big[\underbrace{R(s,a)}_{\text{expected reward}} + \gamma \underbrace{\mathbb{E}_{s' \sim P(\cdot|s,a)} \big[V^\pi(s') \big]}_{\text{discounted expected future value.}} \Big] \quad \text{—(1)}$$

Since, $\mathbb{E}_{a \sim \pi(\cdot|s)} [f(a)] = \sum_{a \in A} \pi(a|s) f(a)$

expanding outer expt.

$$\therefore V^\pi(s) = \sum_{a \in A} \pi(a|s) \Big[R(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \big[V^\pi(s') \big] \Big] \quad \text{—(2)}$$

next expanding the inner expectation ← this guy

i.e. $\mathbb{E}_{s' \sim P(\cdot|s,a)} [V^\pi(s')] = \sum_{s' \in S} P(s'|s,a) V^\pi(s')$ —(3)

add ② & ③ into ① and substitute

$$v^{\pi}(s) = \sum_{a} \pi(a|s) \left[R(s,a) + \gamma \sum_{s' \in S} \mathbb{P}(s'|s,a)\, v^{\pi}(s') \right] \quad \forall s \in S \qquad \text{—④}$$

new variable $R^{\pi}(s) = \sum_{a} \pi(a|s)\, R(s,a)$ } first term reward vector —⑤

& $P^{\pi}(s,s') = \sum_{a} \pi(a|s)\, \mathbb{P}(s'|s,a)$ transition matrix —⑥ reward & transprob

So ④ becomes

$$v^{\pi}(s) = R^{\pi}(s) + \gamma \sum_{s' \in S} \left[P^{\pi}(s,s')\, v^{\pi}(s') \right] \qquad \text{—⑦}$$

Next we write in vector form, for discrete state space

$S = \{s_1, s_2, \ldots, s_n\}$ state values

$$\vec{v}^{\pi} = \begin{bmatrix} V^{\pi}(s_1) \\ V^{\pi}(s_2) \\ \vdots \\ V^{\pi}(s_n) \end{bmatrix}_{|S| \times 1} \qquad \vec{R}^{\pi}(s) = \begin{bmatrix} R^{\pi}(s_1) \\ \vdots \\ R^{\pi}(s_n) \end{bmatrix}_{|S| \times 1} \leftarrow \text{its } R^{\pi}(s_n)$$

expected reward

transition matrix

$$\& \quad P^{\pi} = \begin{bmatrix} P^{\pi}(s_1,s_1') & \cdots & P^{\pi}(s_1,s_n') \\ \vdots & & \\ P^{\pi}(s_n,s_1') & \cdots & P^{\pi}(s_n,s_n') \end{bmatrix}_{|S| \times |S|}$$

Generalizing the equation

$$\vec{v}^{\pi} = \vec{R}^{\pi}(s) + \gamma \langle P^{\pi}(s,s'), \vec{v}^{\pi} \rangle$$

$$\begin{bmatrix} \square \end{bmatrix} = \begin{bmatrix} \square \end{bmatrix} + \gamma \begin{bmatrix} \square\square \end{bmatrix} \begin{bmatrix} \square \end{bmatrix}$$

$$\vec{v}^{\pi} = \vec{R}^{\pi} + \gamma P^{\pi} \vec{v}^{\pi}$$

$$\Rightarrow \quad \therefore \quad (I - \gamma P^{\pi}) \vec{v}^{\pi} = \vec{R}^{\pi}$$

$$\vec{v}^{\pi} = (I - \gamma P^{\pi})^{-1} \vec{R}^{\pi} \qquad \text{—⑧}$$

1) here $\vec{R}^{\pi}$ is col^{n} vector $|S| \times 1$, can be calculated using eq^{n} ⑤ if dependent on (s,a)

2) $P^{\pi}(s,s')$ is $|S| \times |S|$ square matrix calculated using ⑥ eq^{n}.

Then we solve linear system ⑧ [Answer]

P5.3

Reward fⁿ stochastic

$$R: S \times A \to \Delta(\mathbb{R})$$

value fⁿ under policy π defined as,

$$v^{\pi}(s) = \mathop{\mathbb{E}}_{\pi, P, R}\left[\sum_{t=0}^{\infty} \gamma^t R(S_t, A_t) \,|\, S_0 = s\right] \quad \text{—— (1)}$$

Since $\sum \gamma^t R(S_t, a_t)\big|_{S_0 = s} = R(S_0, a_0) + \gamma \mathop{\mathbb{E}}_{t=1}^{\infty} \gamma^{t-1} R(S_t, a_t) \,|\, S_0 = s$

($v^{\pi}(S_0')$ value fⁿ next step.)

$$=$$

$$\therefore (1) \text{ becomes } \quad v^{\pi}(s) = \mathop{\mathbb{E}}_{\pi, P, R}\left[R(S_0, a_0) + \gamma v^{\pi}(S_0') \,|\, S_0 = s\right] \quad \text{—— (2)}$$

$$v^{\pi}(s) = \mathop{\mathbb{E}}$$

$$= \sum_{a \in A} \pi(a|s) \left[\sum_{r \in R} \sum_{s' \in S} p(s', r|s, a) \cdot [r + \gamma v^{\pi}(s')]\right]$$

$$\checkmark\ v^{\pi}(s) = \sum_{a \in A} \pi(a|s) \left(\sum_{r \in R} \sum_{s' \in S} p(s', r|s, a) \cdot r + \sum_{r \in R} \sum_{s' \in S} p(s', r|s, a) \cdot \gamma v^{\pi}(s')\right) \quad \text{—— (2a)}$$

$$R^{\pi}(s) = \sum_{a \in A} \pi(a|s) \left[\sum_{s' \in S} \sum_{r \in R} p(s', r|s, a) \cdot r\right] \quad \text{—— (3)}$$

$$\sum_{s' \in S} p(s', r|s, a) = p(r|s, a) \quad \text{—— (4)}$$

(3) becomes $\quad R^{\pi}(s) = \sum_{a \in A} \pi(a|s) \sum_{r \in R} p(r|s, a)\, r \quad \checkmark \quad \text{—— (5)}$

↓ column vector $|S| \times 1$

$$p^{\pi}(s, s') = \sum_{a \in A} \pi(a|s) \left[\sum_{r \in R} p(s', r|s, a)\right]$$

↓ $p(s'|s, a)$

transition prob $s \to s'$ with π

$$\therefore \quad p^{\pi}(s, s') = \sum_{a \in A} \pi(a|s) \cdot p(s'|s, a) \cdot \quad \checkmark \quad \text{—— (6)}$$

↓ $|S| \times |S|$ matrix as before

∴ Substituting ⑤ & ⑥ in 2a

$$v^{\pi}(s) = R^{\pi}(s) + \gamma \sum_{s' \in S} P^{\pi}(s, s')\, v^{\pi}(s')$$

Same as part I answer

$$v^{\pi} = R^{\pi} + \gamma P^{\pi} v^{\pi} \quad \text{matrix \& vector notation.}$$

$$\text{or } (I - \gamma P^{\pi})\, v^{\pi} = R^{\pi}$$

$$\text{or } V^{\pi} = (I - \gamma P^{\pi})^{-1} R^{\pi}$$

Answer

R^{π} & V^{π} can be determined from expressions derived earlier page.

P4.1

Given, $M_1 = (S, \mu_0, A, P, R, \gamma)$ $M_2 = (S, \mu_0, A, P, R', \gamma)$

infinite-horizon discounted setting.

$$R'(s,a) = \alpha R(s,a) + \beta \quad , \alpha > 0 \quad \forall s \in S \text{ \& } a \in A \qquad \text{—— ①}$$

V_1^{π} & V_2^{π} are value fn of M_1 & M_2

Val fn for infinite hoz discount setting from lecture 6

$$V^{\pi}(s) = \mathop{\mathbb{E}}_{\substack{A_t \sim \pi \\ S_{t+1} \sim P(\cdot|S_t, A_t)}} \left[\sum_{t=0}^{\infty} \gamma^t R(S_t, a_t) \,|\, S_0 = s \right] \qquad \forall s \in S$$

as question asks, we start with value fn of M_2

$$V_2^{\pi}(s) = \mathop{\mathbb{E}}_{\substack{A_t \sim \pi \\ S_{t+1} \sim \ldots}} \left[\sum_{t=0}^{\infty} \gamma^t R'(s_t, a_t) \,|\, S_0 = s \right] \qquad \text{—— ②}$$

R' into ② from ① we get

$$V_2^{\pi}(s) = \mathop{\mathbb{E}}_{\substack{A_t \sim \pi \\ S_{t+1} \sim P(\cdot|S_t, A_t)}} \left[\sum_{t=0}^{\infty} \gamma^t \left(\alpha R(S_t, a_t) + \beta \right) \,|\, S_0 = s \right]$$

$$= \mathop{\mathbb{E}}_{\substack{A_t \sim \pi \\ S_{t+1} \ldots}} \left[\sum_{t=0}^{\infty} \gamma^t \alpha R(S_t, a_t) \,|\, S_0 = s \right] + \mathop{\mathbb{E}}_{\substack{A_t \sim \pi \\ S_{t+1} \ldots}} \left[\sum_{t=0}^{\infty} \gamma^t \beta \,|\, S_0 = s \right] \qquad \text{—— ③}$$

since $\mathbb{E}[C_1 X_1 + C_2 X_2]$ linearity exists. $= C_1 \mathbb{E}[X_1] + C_2 \mathbb{E}[X_2]$

$$\therefore V_2^{\pi}(s) = \alpha \mathop{\mathbb{E}}_{\substack{A_t \sim \pi(\tau) \\ s_{t+1} \sim P(\cdot|s_t, A_t)}} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \,\middle|\, s_0 = s \right] + \left[\beta \mathop{\mathbb{E}}_{\substack{A_t \sim \pi(\tau) \\ s_{t+1} \sim P(\cdot|s_t, A_t)}} \left[\sum_{0}^{\infty} \gamma^t \right] \right] \quad (13)$$

(term 1) $\rightarrow V_1^{\pi}(s)$; (term 2) $\rightarrow$ infinite series

term 2 is geometric series $\sum_{0}^{\infty} \gamma^t = 1 + \gamma + \gamma^2 + \cdots$

let sum $S_n = 1 + \gamma + \gamma^2 + \cdots + \gamma^{n-1}$ $\therefore \gamma S_n = \gamma + \gamma^2 + \cdots + \gamma^n$

$\therefore S_n(1-\gamma) = 1 - \gamma^n$ $\therefore S_n = \frac{1-\gamma^n}{1-\gamma}$ (saw wikipedia)

Since $0 \le \gamma < 1$

$$\lim_{n \to \infty} \frac{1-\gamma^n}{1-\gamma} = \frac{1-0}{1-\gamma} = \frac{1}{1-\gamma}$$

$$\therefore V_2^{\pi}(s) = \alpha V_1^{\pi}(s) + \beta \cdot \left(\frac{1}{1-\gamma} \right) = \alpha V_1^{\pi}(s) + \frac{\beta}{1-\gamma}$$

Answer

RE2

P4.2 Optimal policy $V_2^*(s) = \max_{\pi} V_2^{\pi}(s) = \max_{\pi} \left(\alpha V_1^{\pi}(s) + \frac{\beta}{1-\gamma} \right)$

$= \alpha V_1^*(s) + \frac{\beta}{1-\gamma}$ $\forall \alpha > 0$

$$Q_2^*(s,a) = R'(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_2^*(s')$$

$$= (\alpha R(s,a) + \beta) + \gamma \sum P$$

P4.2 Optimal, deterministic policy → let $\pi_1^*(s)$ & $\pi_2^*(s)$ be that for M_1 & M_2 respectively.

↓ Sets of optimal stationary policies.

By definition π_1^* being optimal policy of M_1, $V_1^{\pi_1^*}(s) \ge V_1^{\pi}(s)$ $\forall s \in S$

Since $\alpha > 0$, $\alpha V_1^{\pi_1^*}(s) \ge \alpha V_1^{\pi}(s)$ ($\times \alpha$ both side)

or $\alpha V_1^{\pi_1^*}(s) + \frac{\beta}{1-\gamma} \ge \alpha V_1^{\pi}(s) + \frac{\beta}{1-\gamma}$ ($+ \frac{\beta}{1-\gamma}$ both side)

from P4.1 answer we substitute expression for $V_2^{\pi}(s)$

$\therefore V_2^{\pi_1^*}(s) \geq V_2^{\pi}(s)$ this is definition of optimal policy of M_2.

$\therefore$ optimal policy π_1^* for M_1 also optimal policy for M_2

Same way, $V_2^{\pi_2^*}(s) \geq V_2^{\pi}(s) \quad \forall s \in S$

$\therefore \alpha V_1^{\pi_2^*}(s) + \frac{\beta}{1-\gamma} \geq \alpha V_2^{\pi}(s) + \frac{\beta}{1-\gamma}$

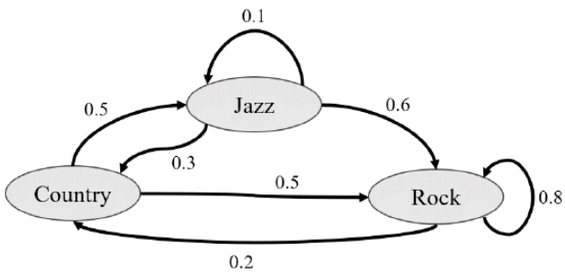
Subst: same way, $V_1^{\pi_2^*}(s) \geq V_1^{\pi}(s)$ (by removing/div constant terms & factors)

$\therefore$ same way π_2^* for M_2 is also an optimal policy for M_1

So we see for M_1 & M_2, any optimal policy of M_1 is also optimal for M_2 so set of policies π_1^* & π_2^* are identical.

—x—

Problem 1: Consider a jukebox that plays songs from three genres of music: "Jazz," "Rock," and "Country." Once started, it plays a Jazz song and then, switches between the genres according to the time-homogeneous Markov chain $(X_t)_{t=0}^{\infty}$, depicted below.



Question P1.3 With what probabilities will the third song (at t = 2) belong to each of these three genres ?

Answer:

In [ ]:
```
states = ["Jazz", "Rock", "Country"] # State Space
import numpy as np

P_matrix = np.array(
    [[0.1, 0.6, 0.3],
    [0.0, 0.8, 0.2],
    [0.5, 0.5, 0.0],])

mu0 = np.array([1.0, 0.0, 0.0])
mu2 = mu0 @ np.linalg.matrix_power(P_matrix, 2)

print("mu_2 (t=2) distribution is [Jazz, Rock, Country]:", mu2)
```

```
mu_2 (t=2) distribution is [Jazz, Rock, Country]: [0.16 0.69 0.15]
```

Question P1.4 This Markov chain is ergodic and hence, has a unique stationary (steady-state) distribution $\overline{\mu}$. Compute $\overline{\mu}$ by hand, calculator, or code; show the steps you use in the computation

Answer:

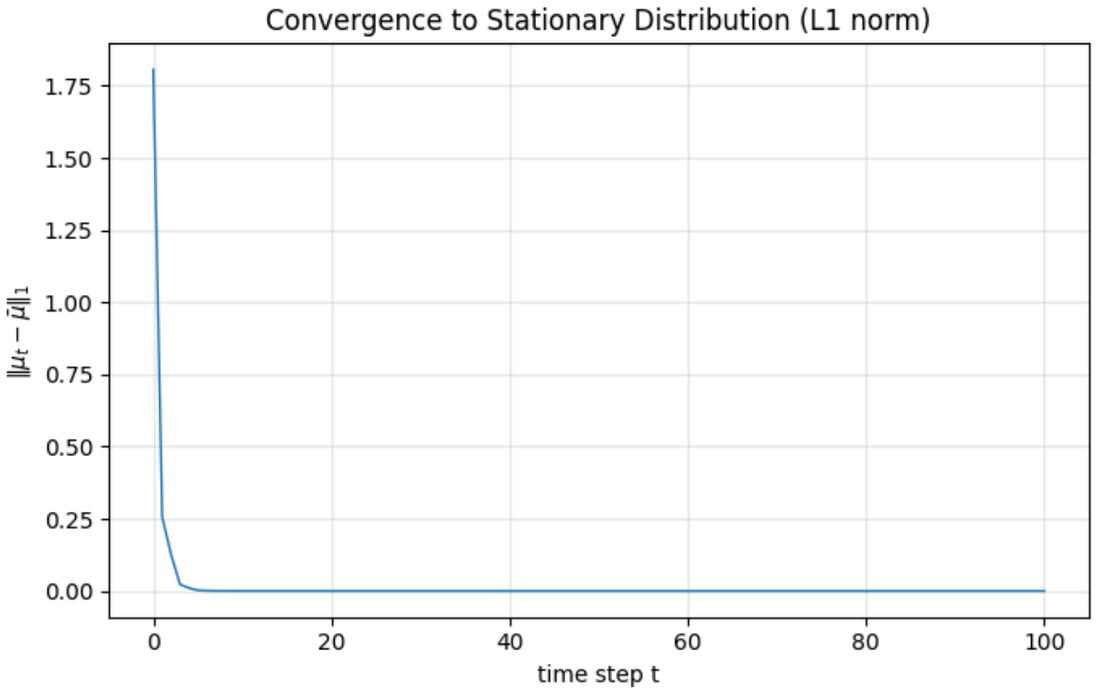
In [ ]:
```
import numpy as np
n = P_matrix.shape[0]

# Build linear system for stationary μ: We solve (P^T - I) μ^T = 0
M = P_matrix.T - np.eye(n) # Transpose of P minus identity matrix
M[-1, :] = 1.0 # Replace last row with [1, 1, 1] for normalization
b = np.zeros(n) # Right-hand side vector
b[-1] = 1.0 # Set last element to 1 for sum = 1

# Solve
mu_bar_T = np.linalg.solve(M, b)
mu_bar = mu_bar_T.T

# Display
print("Stationary Distribution (mu_bar)")
print(mu_bar)

genres = ["Jazz", "Rock", "Country"]
print("Long-Term Probabilities")
for genre, prob in zip(genres, mu_bar):
    print(f" {genre:<8}: {prob:.1%}")
```

```
Stationary Distribution (mu_bar)
[0.09708738 0.72815534 0.17475728]
Long-Term Probabilities
 Jazz    : 9.7%
 Rock    : 72.8%
 Country : 17.5%
```

Question P1.5 Consider the consecutive update of the state distribution μt from t = 0 to t = 100. Plot the sequence of differences between the state distribution μ_t and the stationary distribution $\bar{\mu}$, measured using $L1$ norm, with respect to the time steps.

Answer:

In [ ]:
```
import matplotlib.pyplot as plt

mu = mu0.copy()
ts = list(range(0, 101))   # timesteps
l1_vals = []
# Compute L1 distance at each t and update mu_{t+1} = mu_t P
for t in ts:
  l1_vals.append(np.linalg.norm(mu - mu_bar, 1))
  mu = mu @ P_matrix # Markov update (row vector times P)

# Plot
plt.figure(figsize=(7, 4.5))
plt.plot(ts, l1_vals, linewidth=1)
plt.xlabel("time step t")
plt.ylabel(r"$\| \mu_t - \bar{\mu} \|_1$")
plt.title("Convergence to Stationary Distribution (L1 norm)")
plt.grid(True, alpha=0.3)
plt.tight_layout()
```



Problem P1.6 If we start the jukebox and let it play for a very long time, which genre do you expect to be played most often? Explain your reasoning.

Answer The genre expected to be played most often in the long run is **Rock**. Since the Markov chain is ergodic, it has a unique stationary distribution $\bar{\mu}$ that gives the long-term proportion of time spent in each state (genre), independent of the starting state. The stationary distribution is obtained by solving $\bar{\mu}P = \bar{\mu}$ with the normalization $\bar{\mu}_J + \bar{\mu}_R + \bar{\mu}_C = 1$ which was implemented in prob1.4

Jazz: ≈ 0.097 Rock: ≈ 0.728 Country: ≈ 0.175

So, Rock has the highest stationary probability ($\approx 72.8\%$) and will be played most often.

Problem 3: Consider a movie recommendation system, interacting with a human user, that aims to suggest movies that the user would like to watch. In each step of the interaction, the human communicates its current desired movie genre to the system, which may be *Action*, *Horror*, or *Comedy*. Upon receiving this communication, the system selects a movie among *Movie A*, *Movie B*, *Movie C*, and *Movie D*. The genre of these movies is listed below, where 1 shows that the movie is categorized under that genre while 0 shows that it is not categorized under that genre.

	Action	Horror	Comedy
Movie A	1	0	1
Movie B	1	1	0
Movie C	0	1	1
Movie D	0	1	0

Upon receiving the recommendation, the user watches the movie and provides a *like* if the movie belongs to the desired genre and provides a *dislike* otherwise. If the movie belongs to the desired genre, in the next interaction, the user's desired movie genre will be selected uniformly at random from the other two genres. For instance, if the user wanted Horror in this interaction and Movie B, C, or D was suggested, the user will want Action or Comedy in the next interaction with equal probability. However, if the movie does not belong to the desired genre, in the next interaction, the user's desired movie genre will remain the same. At the beginning of the interaction, the user may select any of the genres uniformly at random.

Consider the following recommendation strategy by the system:

- If the user asks for Action movies $\longrightarrow$ The system will recommend Movies A or B, each with probability of 0.4, or Movie D with probability of 0.2.
- If the user asks for Horror movies $\longrightarrow$ The system will recommend any of the four movies with probability 0.25.
- If the user asks for Comedy movies $\longrightarrow$ The system will recommend Movie B with probability of 0.1, or Movie C with probability of 0.9.

Problem 3.3 Define the Markov chain that is induced by the policy given in Part 2 over the MDP you introduced in Part 1; in particular, specify all elements of the Markov chain.

Answer: Please refer to the handwritten homework pages 8 and 9 and below computation for matrices as combined answer.

In [ ]:
```
import numpy as np

# Define the policy pi(a|s): States: 0= Action, 1= Horrror, 2= Comedy and Actions:
pi = np.zeros((3, 4))

# Policy declaration Pi(a|s)
pi[0, 0] = 0.4  # Movie_A for Action
pi[0, 1] = 0.4  # Movie_B for Action
pi[0, 3] = 0.2  # Movie_D for Action
pi[1, :] = 0.25  # uniform for Horror
pi[2, 1] = 0.1  # Movie_B for Comedy
pi[2, 2] = 0.9  # Movie_C for Comedy
# print("Policy matrix:\n", pi)

# Genre matching I(s, a): 1 if movie a matches genre s, and if doesn't match
# its 0 otherwise. Provided.
I = np.array([
    [1, 0, 1],  # Movie_A: Action=1, Horror=0, Comedy=1
    [1, 1, 0],  # Movie_B: Action=1, Horror=1, Comedy=0
    [0, 1, 1],  # Movie_C: Action=0, Horror=1, Comedy=1
    [0, 1, 0]   # Movie_D: Action=0, Horror=1, Comedy=0
]).T

# Transition matrix P^pi (3x3)
P_pi = np.zeros((3, 3))
for s in range(3):
  for a in range(4):
    if pi[s, a] == 0:
      continue
    if I[s, a] == 1:  # Match: switch (prob 0.5)
      for sp in range(3):
        if sp != s:
          P_pi[s, sp] += pi[s, a] * 0.5
    else:  # No match: stay in s (prob 1)
```

```
      P_pi[s, s] += pi[s, a] * 1.0

print("Transition Matrix P^pi:")
print(P_pi)

# Expected rewards R^pi(s): sum pi(a|s) * r(s,a), where r= +1 if match, -1 else
R_pi = np.zeros(3)
for s in range(3):
  r_sa = 2 * I[s, :] - 1
  R_pi[s] = np.sum(pi[s, :] * r_sa)

print("\nExpected Rewards R^pi:")
print(R_pi)
```

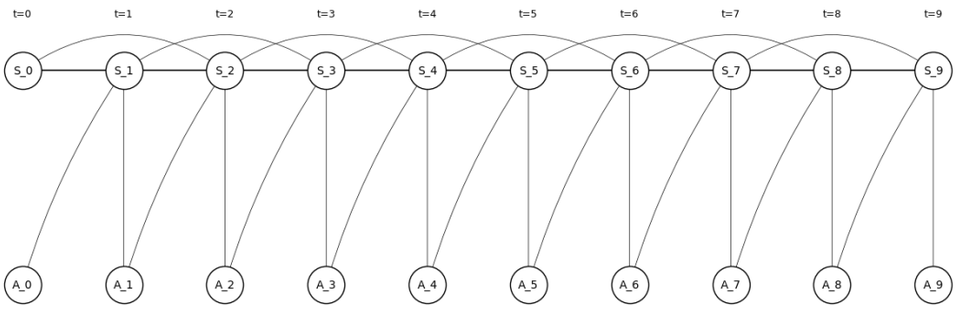
```
Transition Matrix P^pi:
[[0.2   0.4   0.4  ]
 [0.375 0.25  0.375]
 [0.45  0.45  0.1  ]]

Expected Rewards R^pi:
[0.6 0.5 0.8]
```

Problem 3.4. Now, suppose the user's next desired movie genre was dependent not only on its last desired genre (and the recommended movie) but also the desired genre before that. Show a graphical representation of the temporal evolution of the model in this case.

Answer: Second-Order MDP Temporal Evolution See graph below.

In [ ]:
```
import matplotlib.pyplot as plt
import networkx as nx

T =9
G = nx.DiGraph()

x_gap = 2.0
y_s, y_a = 0.8, -0.2

pos = {}
for t in range(T+1):
  pos[f"S_{t}"] = (t*x_gap, y_s); G.add_node(f"S_{t}")
  pos[f"A_{t}"] = (t*x_gap, y_a); G.add_node(f"A_{t}")


policy_edges = []                 # S_t -> A_t
env_edges_first = []              # S_{k-1} -> S_k
env_edges_second_top_red = []     # S_{k-2} -> S_k
act_to_state_edges = []           # A_{k-1} -> S_k

# policy edges
for t in range(1, T+1):
  policy_edges.append((f"S_{t}", f"A_{t}"))
for k in range(1, T+1):
  env_edges_first.append((f"S_{k-1}", f"S_{k}"))
  if k >= 2:
      env_edges_second_top_red.append((f"S_{k-2}", f"S_{k}"))
  act_to_state_edges.append((f"A_{k-1}", f"S_{k}"))

plt.figure(figsize=(14, 4))

# nodes + labels
nx.draw_networkx_nodes(G, pos, node_color='white', edgecolors='black',
                       node_size=1100, linewidths=1.0)
nx.draw_networkx_labels(G, pos, font_size=10)

# S_{k-1} -> S_k
nx.draw_networkx_edges(
    G, pos, edgelist=env_edges_first, arrows=True, arrowstyle='->',
    width=1.2, connectionstyle='arc3,rad=0.0', edge_color='black'
)

nx.draw_networkx_edges(
    G, pos, edgelist=env_edges_second_top_red, arrows=True, arrowstyle='->',
```

```
    width=0.5, connectionstyle='arc3,rad=-0.35', edge_color='black', alpha=0.95
)

# A_{k-1} -> S_k
nx.draw_networkx_edges(
    G, pos, edgelist=act_to_state_edges, arrows=True, arrowstyle='->',
    width=0.5, connectionstyle='arc3,rad=-0.08', edge_color='black'
)

# S_t -> A_t
nx.draw_networkx_edges(
    G, pos, edgelist=policy_edges, arrows=True, arrowstyle='->',
    width=0.5, connectionstyle='arc3,rad=0.0', edge_color='black'
)

# time labels
for t in range(T+1):
    plt.text(t*x_gap, 1.05, f"t={t}", ha="center", fontsize=9)
plt.axis('off'); plt.tight_layout(); plt.show()
```



In [2]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [4]:
```
def preprocess(dir):
    # To deal with the error when there is [] in the path
    dir.replace('[','[[]')
    dir.replace(']','[]]')
    return dir

########################        YOUR CODE        ############################

your_ipynb_file_dir = '/content/drive/MyDrive/Colab_Notebooks/ECE595RL/ECE 59500: R

########################      END YOUR CODE      ############################

!jupyter nbconvert --to html '{preprocess(your_ipynb_file_dir)}'
```

```
[NbConvertApp] Converting notebook /content/drive/MyDrive/Colab_Notebooks/ECE595RL/E
CE 59500: Reinforcement Learning HW2.ipynb to html
[NbConvertApp] WARNING | Alternative text is missing on 2 image(s).
[NbConvertApp] Writing 762142 bytes to /content/drive/MyDrive/Colab_Notebooks/ECE595
RL/ECE 59500: Reinforcement Learning HW2.html
```