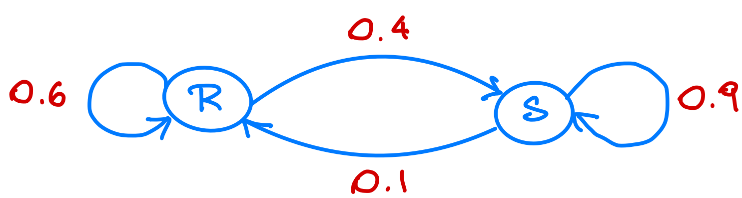
Example. Simple weather model

In some city:

- if some day is rainy the then next day is rainy with probability 0.6 and sunny with probability 0.4;
- if some day is sunny the then next day is rainy with probability 0.1 and sunny with probability 0.9.

Diagram notation:



Matrix notation

$$P = \begin{array}{cc} & \begin{array}{cc} R & \;\; S \end{array} \leftarrow \text{today} \\ \begin{array}{c} R \\ S \end{array} & \begin{bmatrix} 0.6 & 0.1 \\ 0.4 & 0.9 \end{bmatrix} \end{array}$$

next day

Interesting questions:

1) Assume that there is 0.3 probability of rain today. What is the probability that it will rain 2 days from now?

2) In general, what percentage of days is rainy/sunny in this city?

3) If it is sunny one day, how many days on average one will need to wait for a rainy day?

Solution to question 1:

$$X_0 = \begin{array}{c} \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \end{bmatrix} \end{array}\begin{array}{c} R \\ S \end{array}$$ – vector of probabilities that today will be rainy / sunny

$$X_n = \begin{bmatrix} x_1^{(n)} \\ x_2^{(n)} \end{bmatrix}\begin{array}{c} R \\ S \end{array}$$ – vector of probabilities that it will be rainy / sunny n days from now

Assume: $X_0 = \begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix}\begin{array}{c} R \\ S \end{array}$

Tomorrow:

$$X_1 = \begin{array}{c} R \\ S \end{array}\underbrace{\begin{bmatrix} \overset{R}{0.3} \cdot \overset{R\to R}{0.6} + \overset{S}{0.7} \cdot \overset{S\to R}{0.1} \\ \underset{R}{0.3} \cdot \underset{R\to S}{0.4} + \underset{S}{0.7} \cdot \underset{S\to S}{0.9} \end{bmatrix}}_{\begin{bmatrix} 0.6 & 0.1 \\ 0.4 & 0.9 \end{bmatrix} \cdot \begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix} = P \cdot X_0} = \begin{bmatrix} 0.25 \\ 0.75 \end{bmatrix}\begin{array}{c} R \\ S \end{array}$$

We obtain: $X_1 = PX_0$

In the same way:

$$X_2 = PX_1 = P(PX_0) = P^2 X_0$$
$$X_3 = PX_2 = P(P^2 X_0) = P^3 X_0$$
$$\vdots$$
$$X_n = PX_{n-1} = P^n X_0$$

Definition

A *Markov chain* consists of:

- *states* of the Markov chain: $S_1, S_2, \ldots, S_N$;
- the *transition matrix*:

$$P = \begin{bmatrix} p_{11} & \cdots & p_{1N} \\ \vdots & & \vdots \\ p_{N1} & \cdots & p_{NN} \end{bmatrix}$$

  where $p_{i,j}$ = the probability that the system will transition from the state S_j to the state S_i in one step.
- for each $n = 0, 1, \ldots$ the *state vector*

$$X_n = \begin{bmatrix} x_1^{(n)} \\ \vdots \\ x_N^{(n)} \end{bmatrix}$$

  where $x_i^{(n)}$ = the probability that the system is in the state S_i after n steps.

We have:

$$X_1 = PX_0$$
$$X_2 = PX_1 = P^2X_0$$
$$X_3 = PX_2 = P^3X_0$$
$$\vdots$$
$$X_n = PX_{n-1} = P^nX_0$$

Notation:

X_0 = the initial state vector

P^n = the n-step transition matrix (gives probabilities that the system will transition from state S_i to S_j in n steps).

Definition

A *stochastic matrix* is a square matrix

$$P = \begin{bmatrix} p_{11} & \cdots & p_{1N} \\ \vdots & & \vdots \\ p_{N1} & \cdots & p_{NN} \end{bmatrix}$$

such that $p_{ij} \geq 0$ for all i, j and the sum of each column of P is equal to 1.

A *probability vector* is a vector

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$$

such that $x_i \geq 0$ for all i and the sum of all entries of X is equal to 1.

Example. Gambling model.

A gambler repeatedly plays the same game. Each time he plays he will win \$1 with probability p and loose \$1 with probability $1 - p$. The gambler stops playing when the amount of money he has is either \$0 or \$$N$.

Interesting questions. Assume that the gambler has \$$M$ when he starts playing.

1) What are the chances that the gambler will loose all his money?

2) How many games will the gambler play on the average before he stops?

Example. Toy collecting.

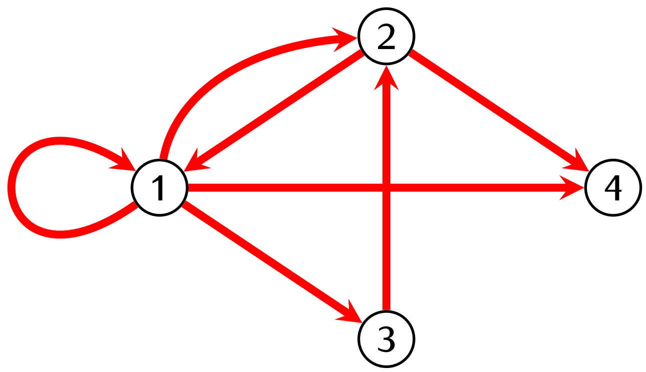
A store adds a free toy to each purchase. There are N different toys, and the type of the toy is selected randomly with probability $\frac{1}{N}$.

Interesting question:

> How many purchases a customer will need to make on the average in order to collect all types of toys?

Example. Random walks on networks.

Directed network:



Random walker: travels from a node to a node. At each step selects an edge outgoing from the current node at random and moves to the end of this edge. If the current node has no outgoing edges, the walker stays at that node.

Interesting questions.

1) In a long run, how often will the walker visit each node?

2) If a walker starts at a node i, how many steps it will take on the average before the walker gets to a node j?