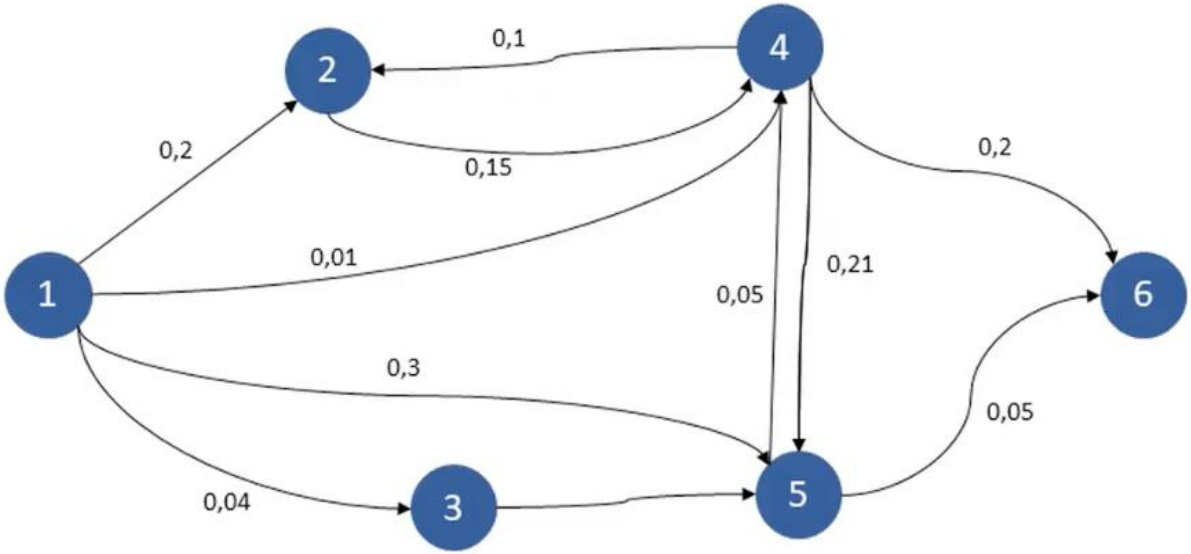
Learning Influence Probabilities

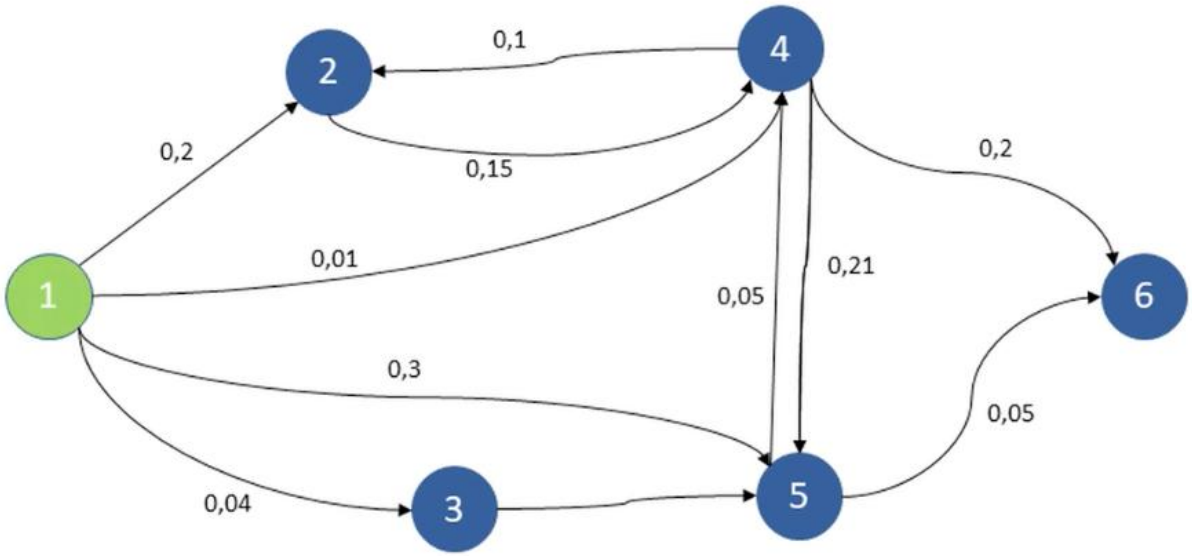
Exercise Lecture

Contents

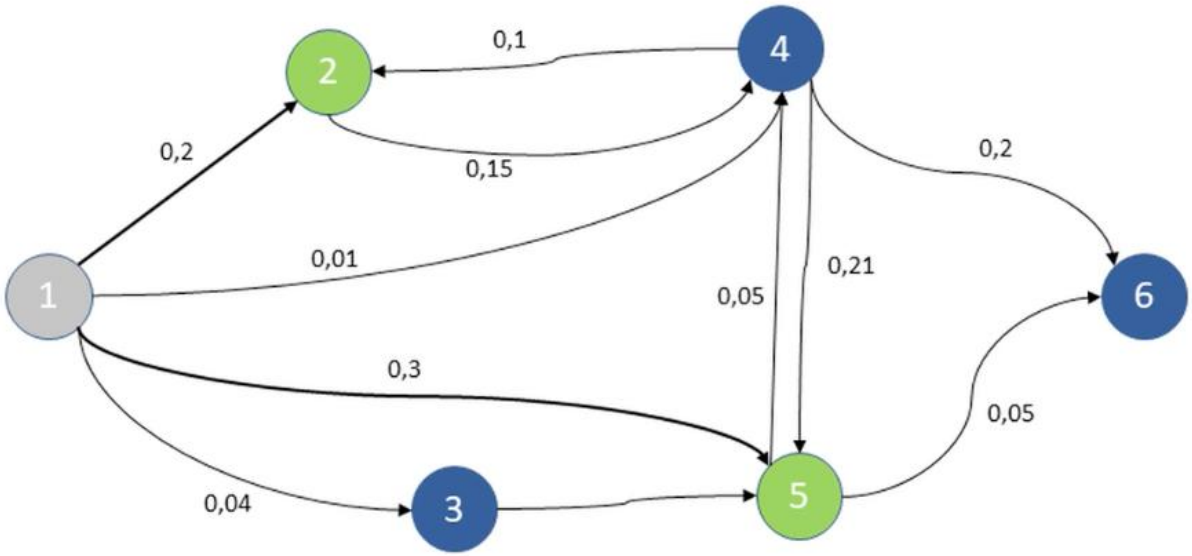
Scenario



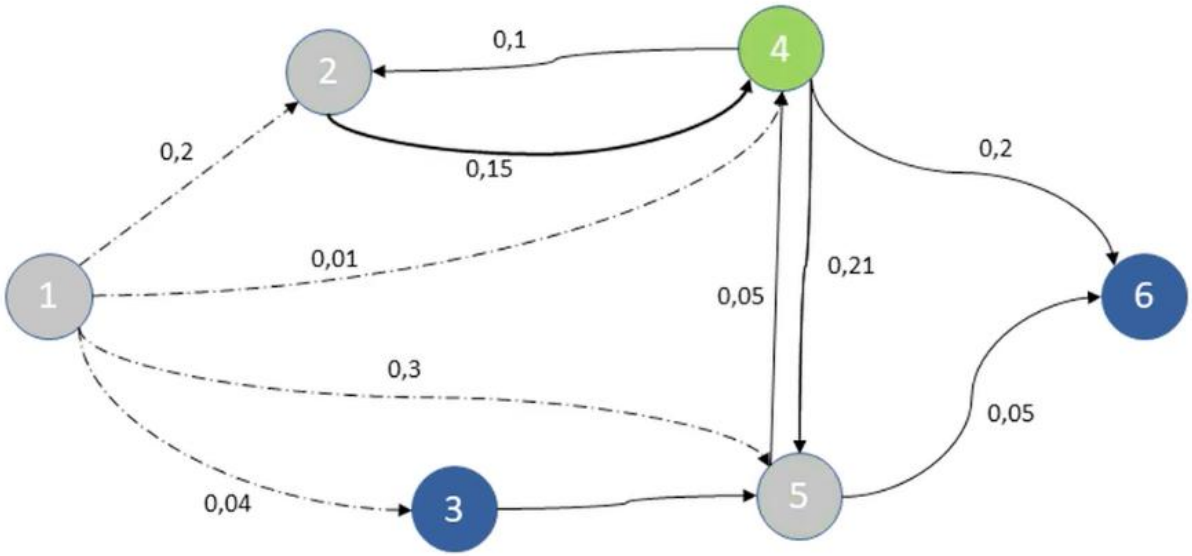
Scenario



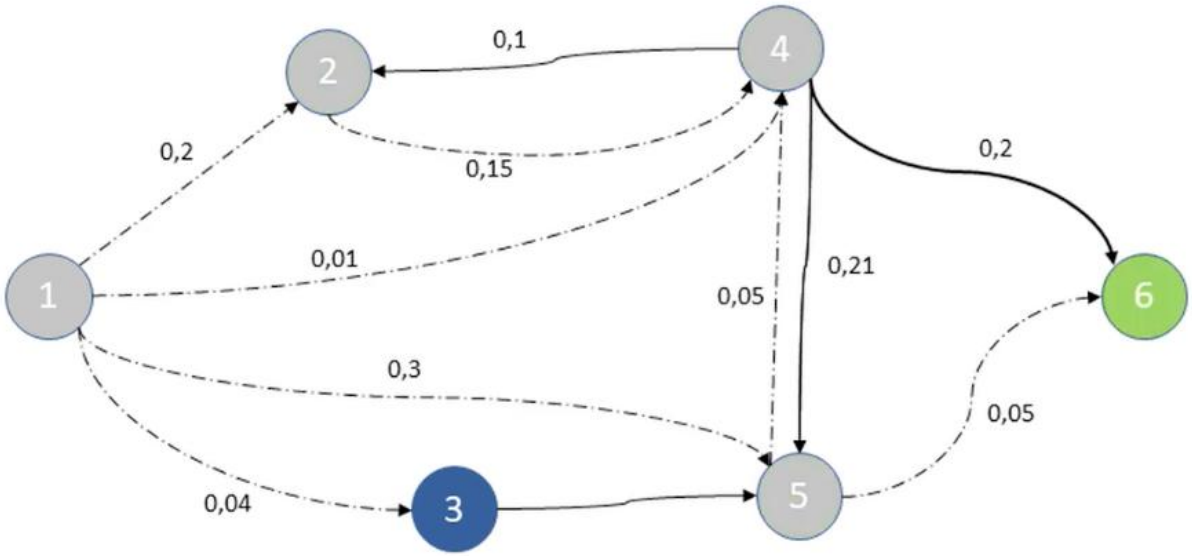
Scenario



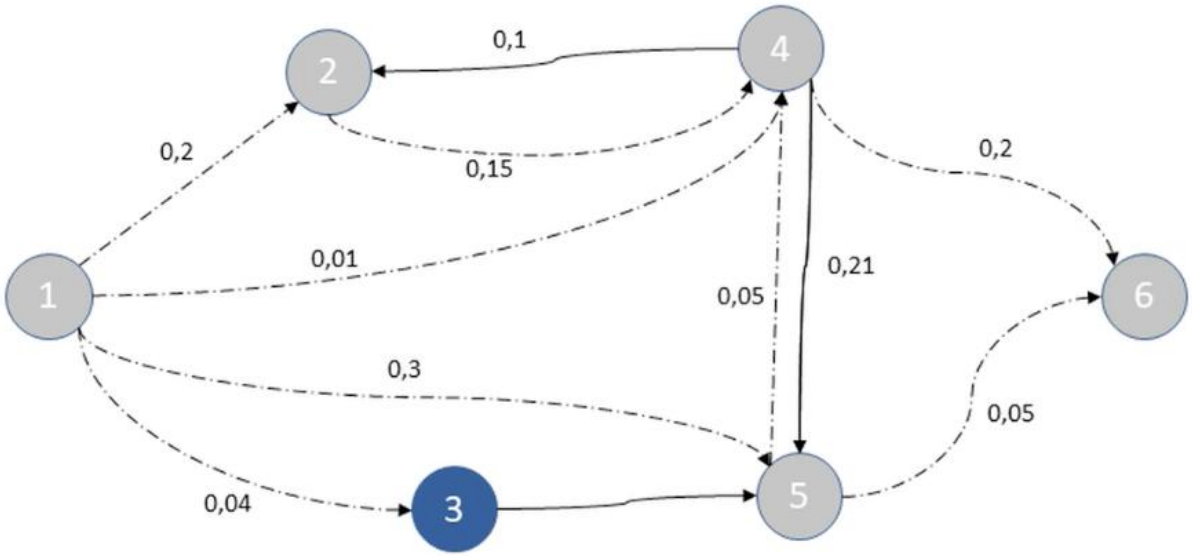
Scenario



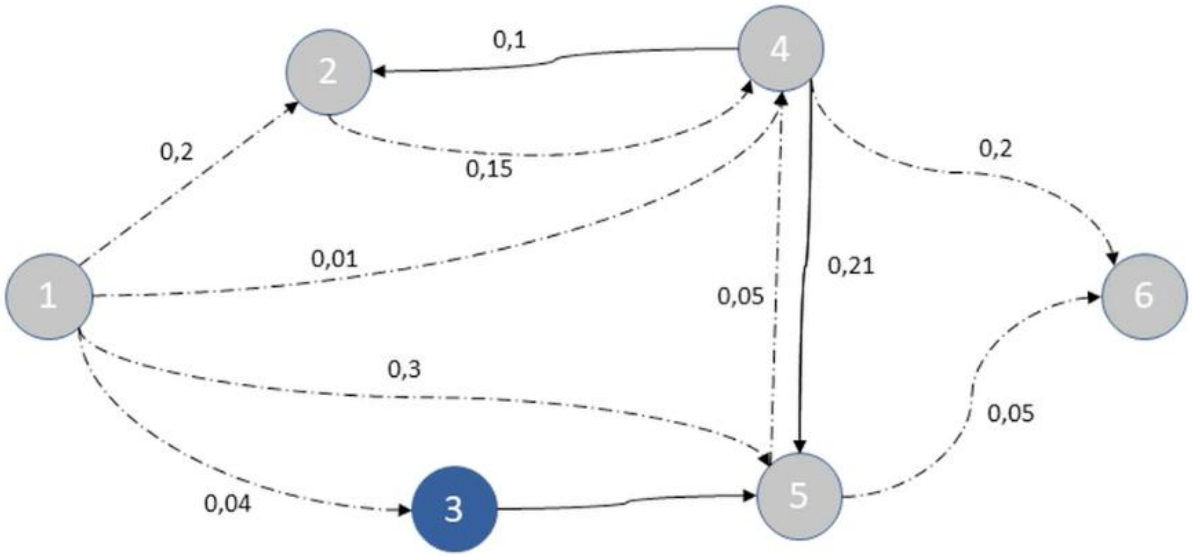
Scenario



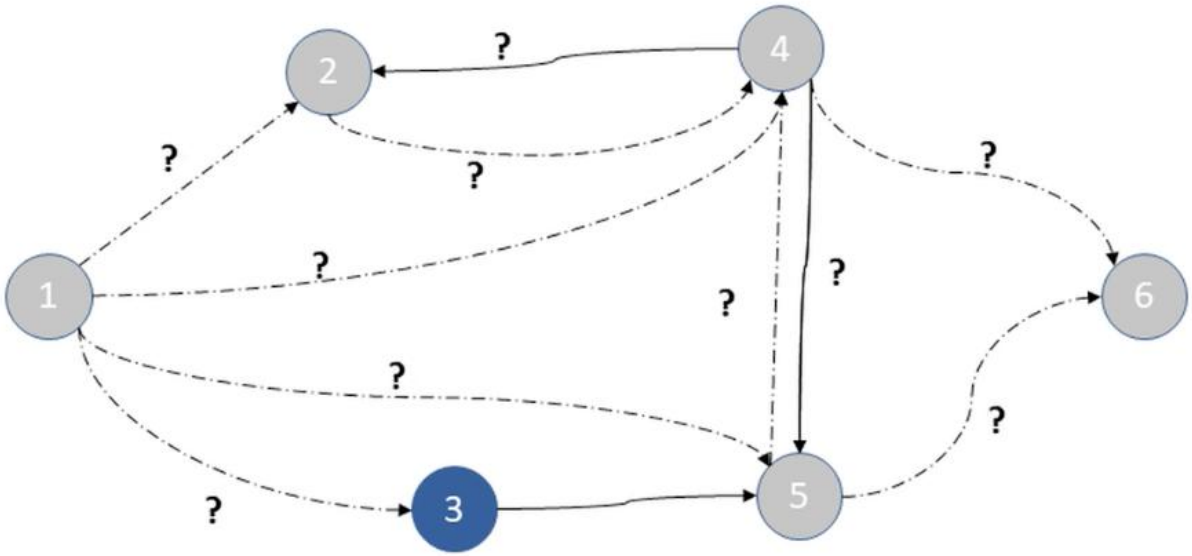
Scenario



Scenario

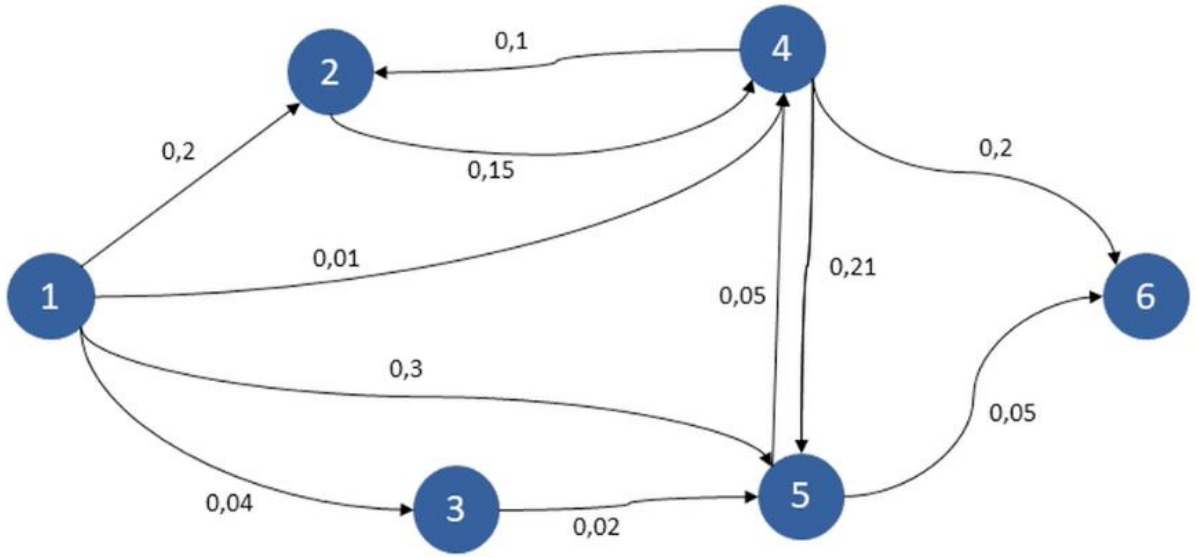


Scenario

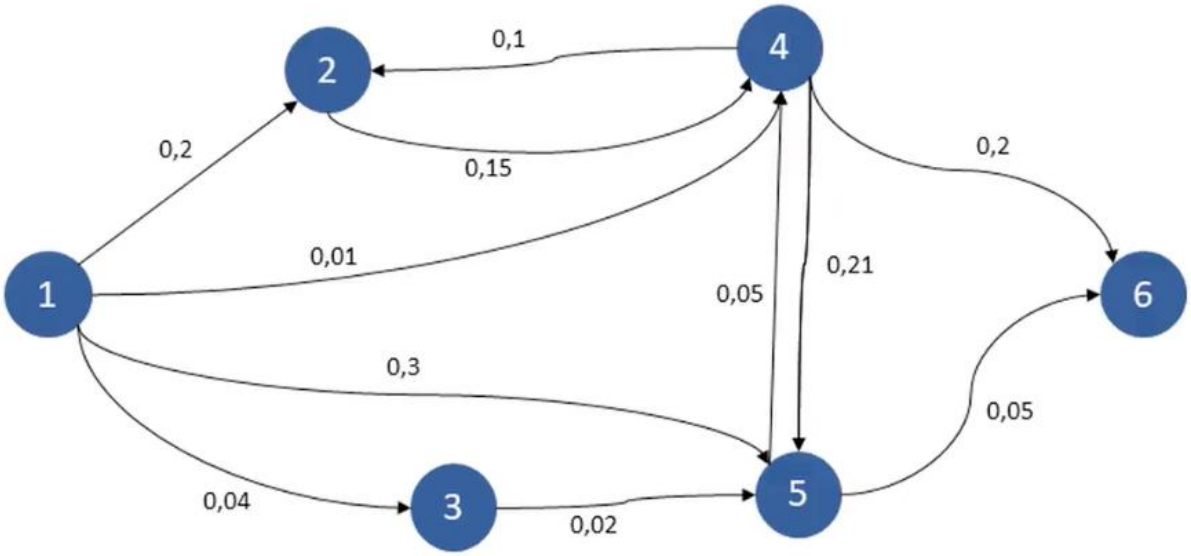


Example

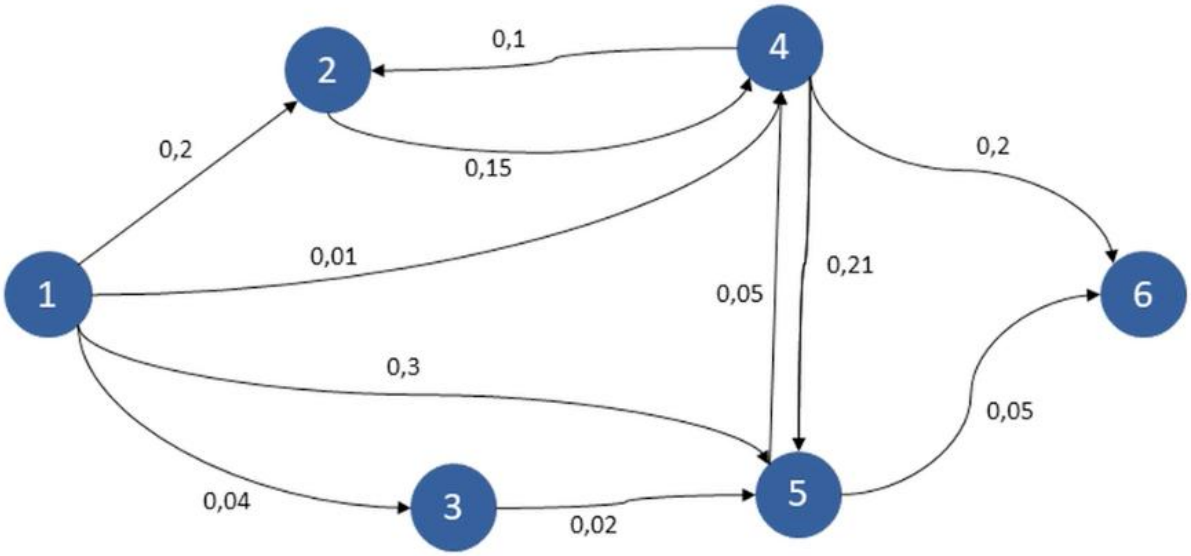
- Simulate an influence diffusion episode and store the values of activated nodes at each time step



Example

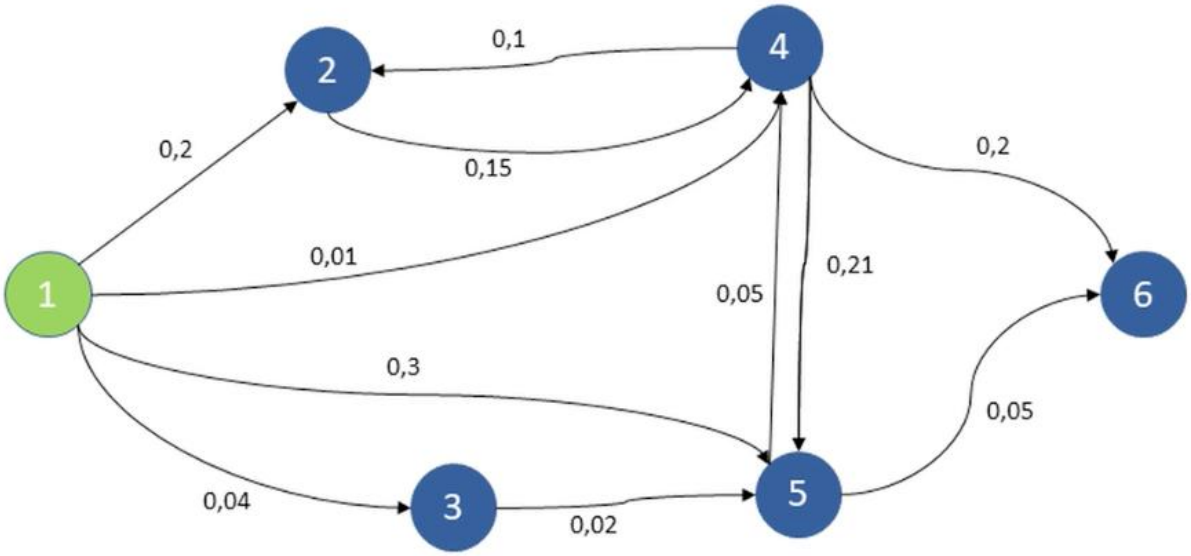


Example: Step 1



	u1	u2	u3	u4	u5	u6
u1	0	0,2	0,04	0,01	0,3	0
u2	0	0	0	0,15	0	0
u3	0	0	0	0	0,02	0
u4	0	0,1	0	0	0,21	0,2
u5	0	0	0	0,05	0	0,05
u6	0	0	0	0	0	0

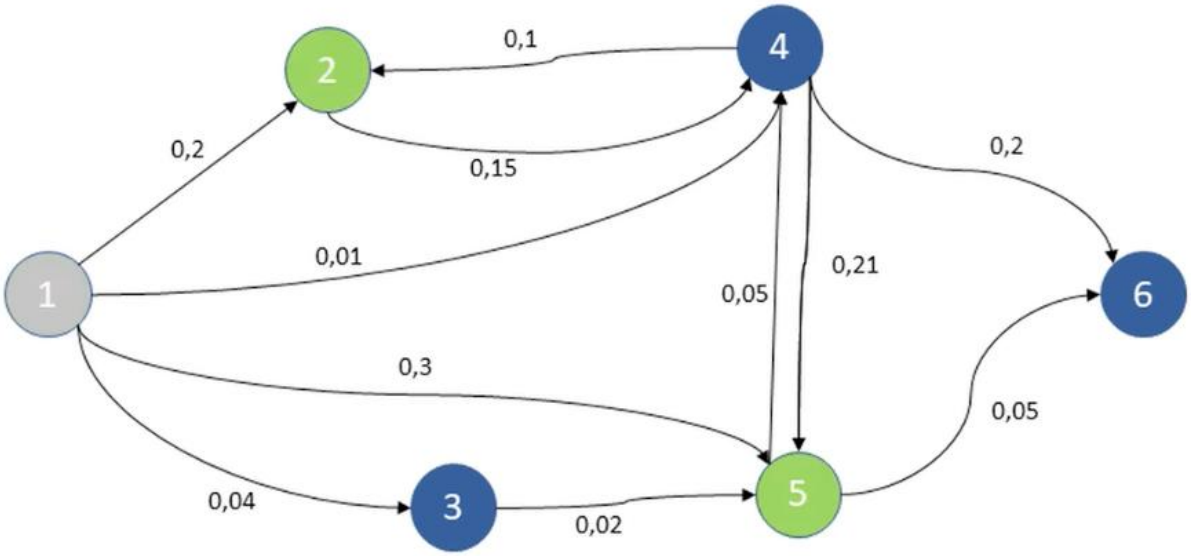
Example: Step 2



u1	u2	u3	u4	u5	u6
1	0	0	0	0	0

	u1	u2	u3	u4	u5	u6
t0	1	0	0	0	0	0
t1						
t2						
t3						

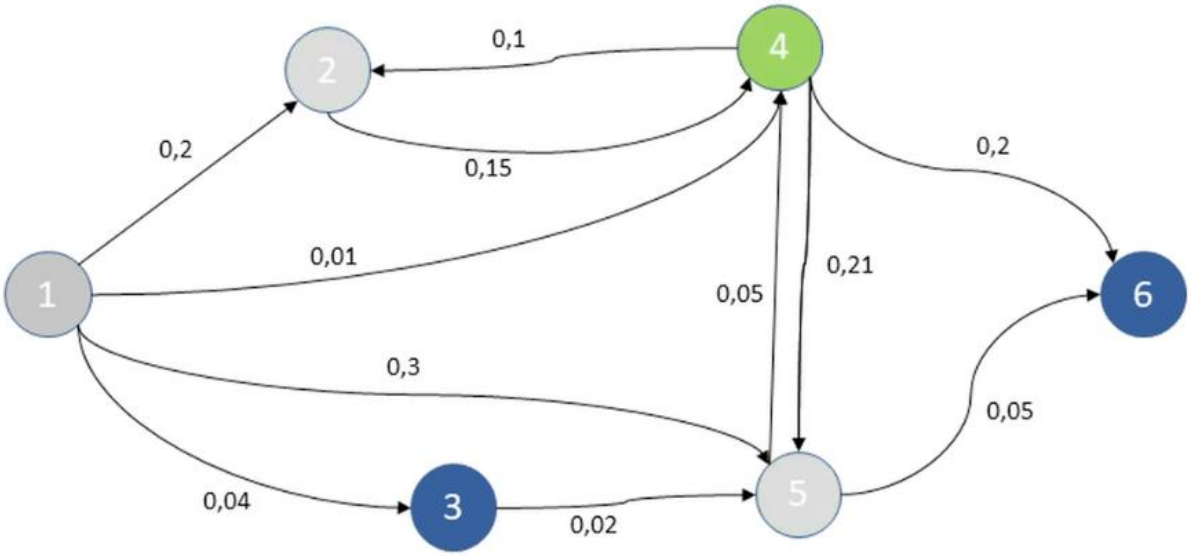
Example: Step 3



	u1	u2	u3	u4	u5	u6
u1	0	0	0	0	0	0
u2	0	0	0	0,15	0	0
u3	0	0	0	0	0,02	0
u4	0	0,1	0	0	0,21	0,2
u5	0	0	0	0,05	0	0,05
u6	0	0	0	0	0	0

	u1	u2	u3	u4	u5	u6
t0	1	0	0	0	0	0
t1	0	1	0	0	1	0
t2						
t3						

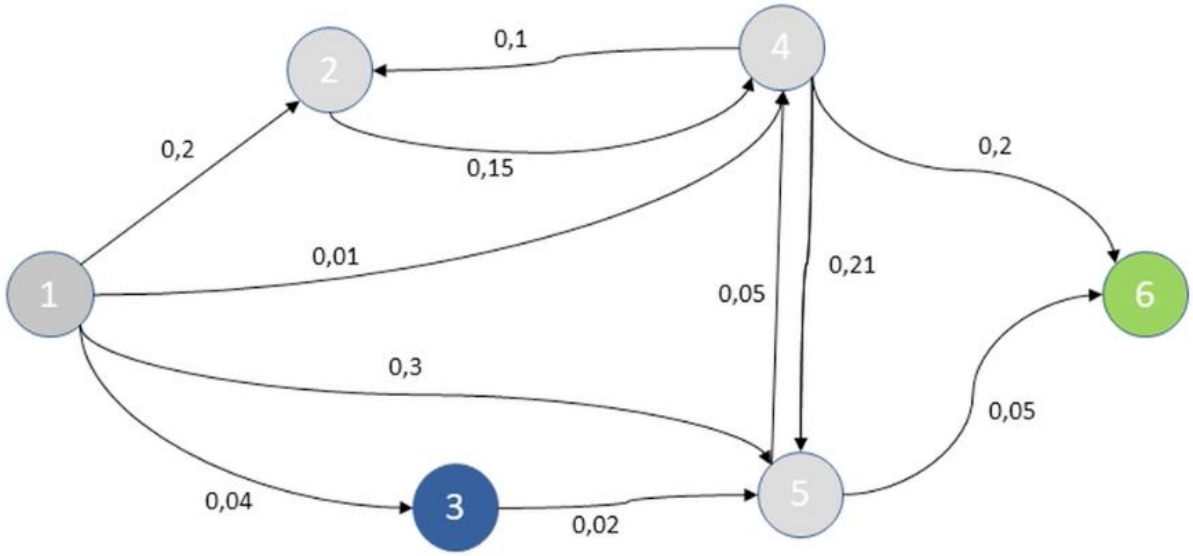
Example: Step 3



	u1	u2	u3	u4	u5	u6
u1	0	0	0	0	0	0
u2	0	0	0	0	0	0
u3	0	0	0	0	0,02	0
u4	0	0,1	0	0	0,21	0,2
u5	0	0	0	0	0	0
u6	0	0	0	0	0	0

	u1	u2	u3	u4	u5	u6
t0	1	0	0	0	0	0
t1	0	1	0	0	1	0
t2	0	0	0	1	0	0
t3						

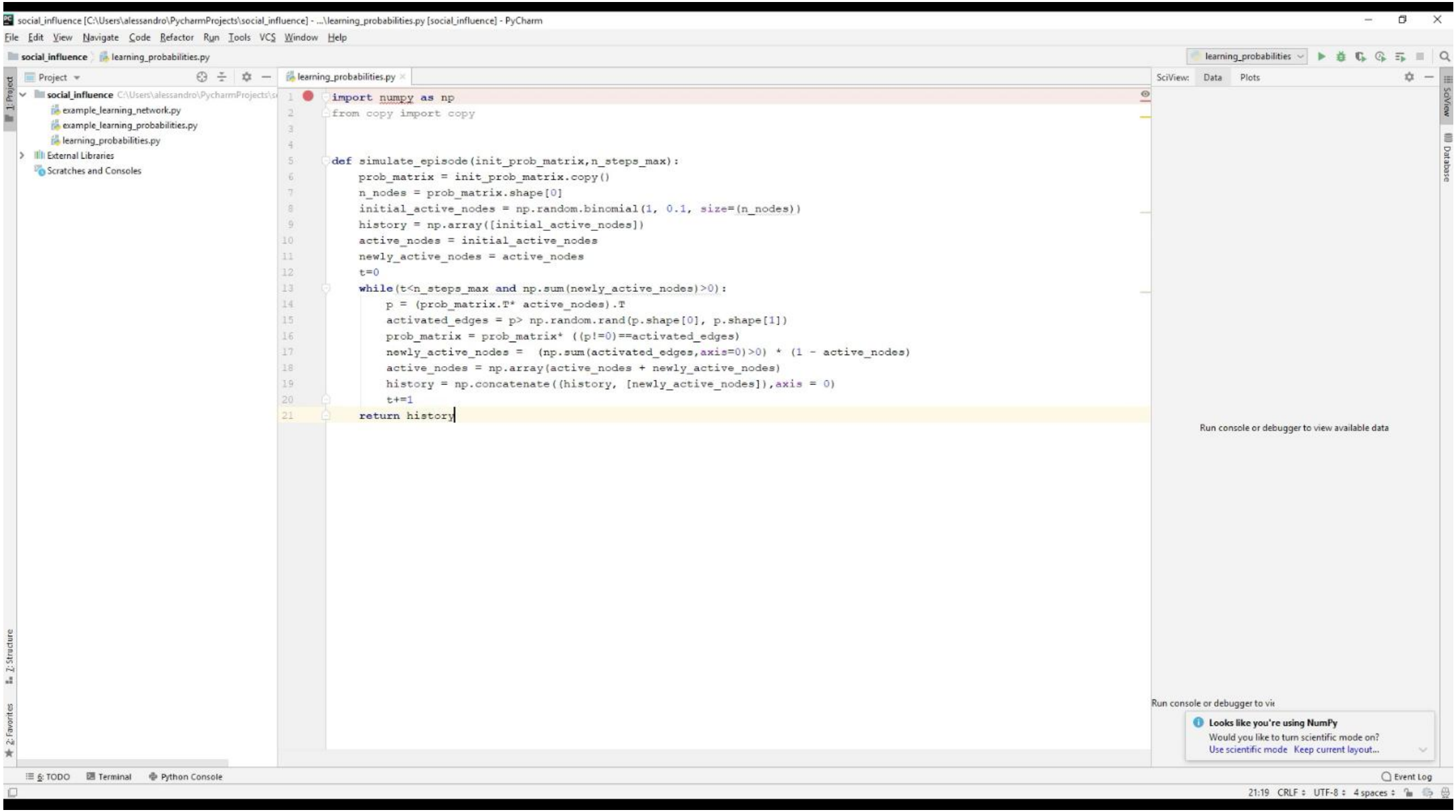
Example: Step 3



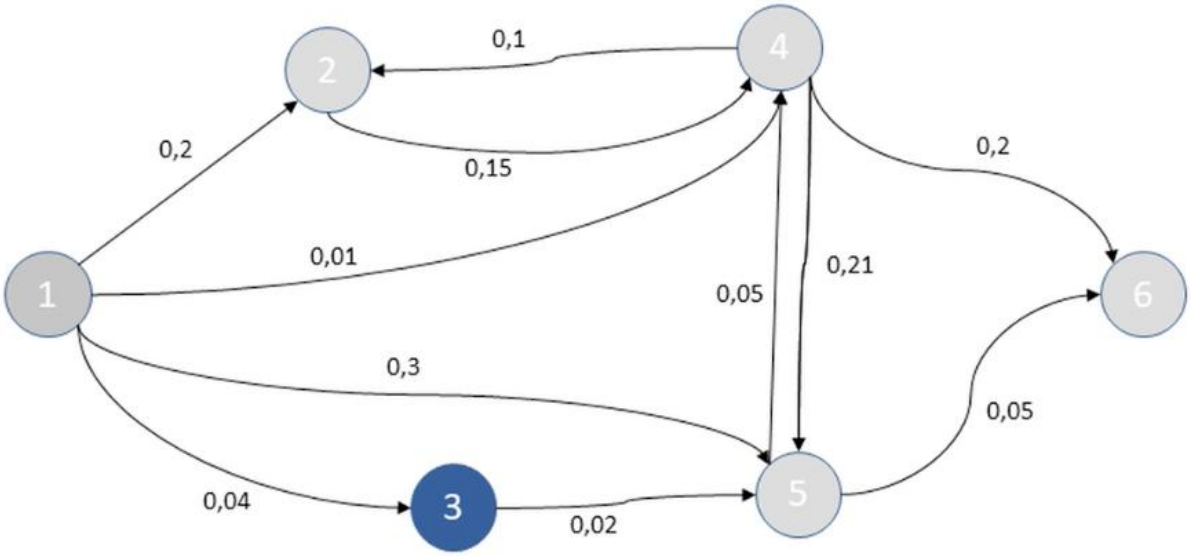
	u1	u2	u3	u4	u5	u6
u1	0	0	0	0	0	0
u2	0	0	0	0	0	0
u3	0	0	0	0	0,02	0
u4	0	0	0	0	0	0
u5	0	0	0	0	0	0
u6	0	0	0	0	0	0

	u1	u2	u3	u4	u5	u6
t0	1	0	0	0	0	0
t1	0	1	0	0	1	0
t2	0	0	0	1	0	0
t3	0	0	0	0	0	1

Let’s implement it!

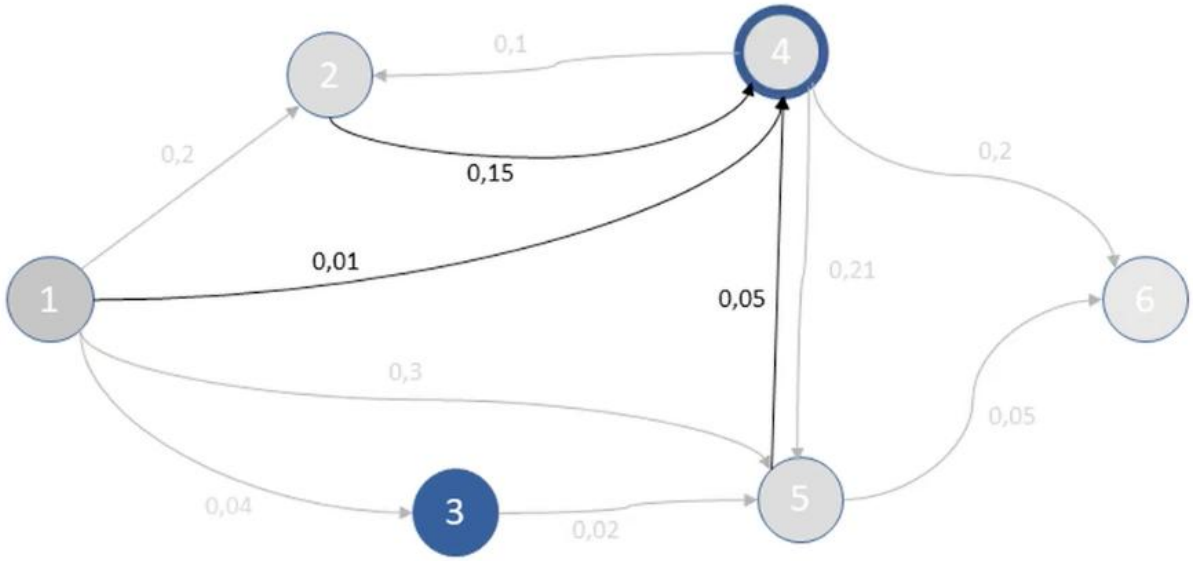

social_influence [C:\Users\alessandro\PycharmProjects\social_influence] - ...\learning_probabilities.py [social_influence] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
social_influence learning_probabilities.py
learning_probabilities
Project
social_influence C:\Users\alessandro\PycharmProjects\s
example_learning_network.py
example_learning_probabilities.py
learning_probabilities.py
External Libraries
Scratches and Consoles
learning_probabilities.py
import numpy as np
from copy import copy
def simulate_episode(init_prob_matrix,n_steps_max):
prob_matrix = init_prob_matrix.copy()
n_nodes = prob_matrix.shape[0]
initial_active_nodes = np.random.binomial(1, 0.1, size=(n_nodes))
history = np.array([initial_active_nodes])
active_nodes = initial_active_nodes
newly_active_nodes = active_nodes
t=0
while(t<n_steps_max and np.sum(newly_active_nodes)>0):
p = (prob_matrix.T* active_nodes).T
activated_edges = p> np.random.rand(p.shape[0], p.shape[1])
prob_matrix = prob_matrix* ((p!=0)==activated_edges)
newly_active_nodes = (np.sum(activated_edges,axis=0)>0) * (1 - active_nodes)
active_nodes = np.array(active_nodes + newly_active_nodes)
history = np.concatenate((history, [newly_active_nodes]),axis = 0)
t+=1
return history
SciView: Data Plots
Run console or debugger to view available data
Run console or debugger to vi
Looks like you're using NumPy
Would you like to turn scientific mode on?
Use scientific mode Keep current layout...
6: TODO Terminal Python Console
Event Log
21:19 CRLF UTF-8 4 spaces

Credit Assignment



	u1	u2	u3	u4	u5	u6
t0	1	0	0	0	0	0
t1	0	1	0	0	1	0
t2	0	0	0	1	0	0
t3	0	0	0	0	0	1

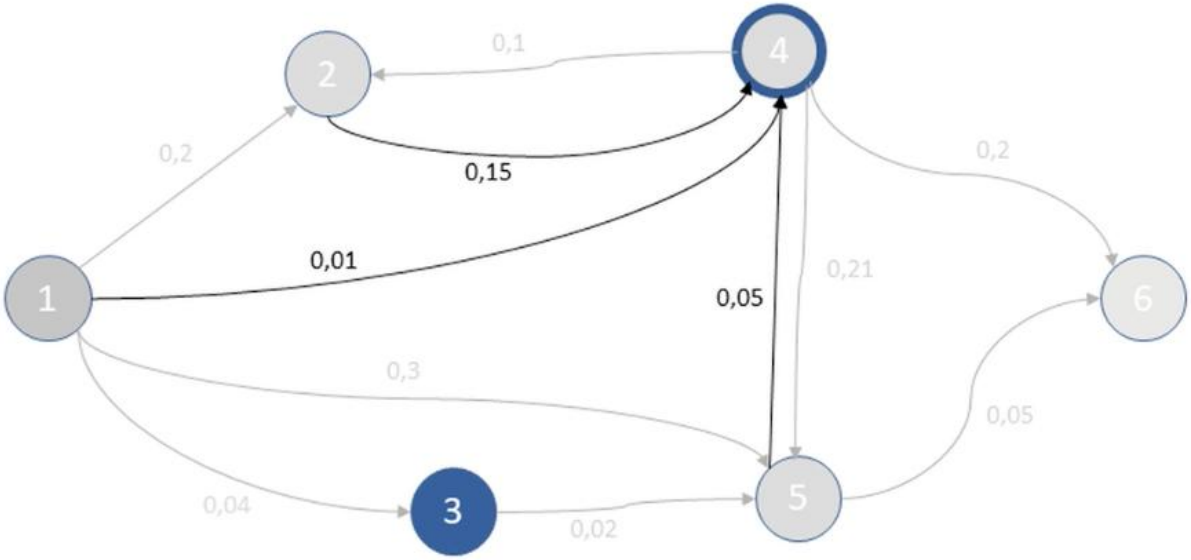
Credit Assignment



$$p_{vw} = \frac{\sum credit_{uv}}{A_v}$$

$$credit_{uv} = \frac{1}{\sum_{w \in S} I(t_w = t_v - 1)}$$

Credit Assignment

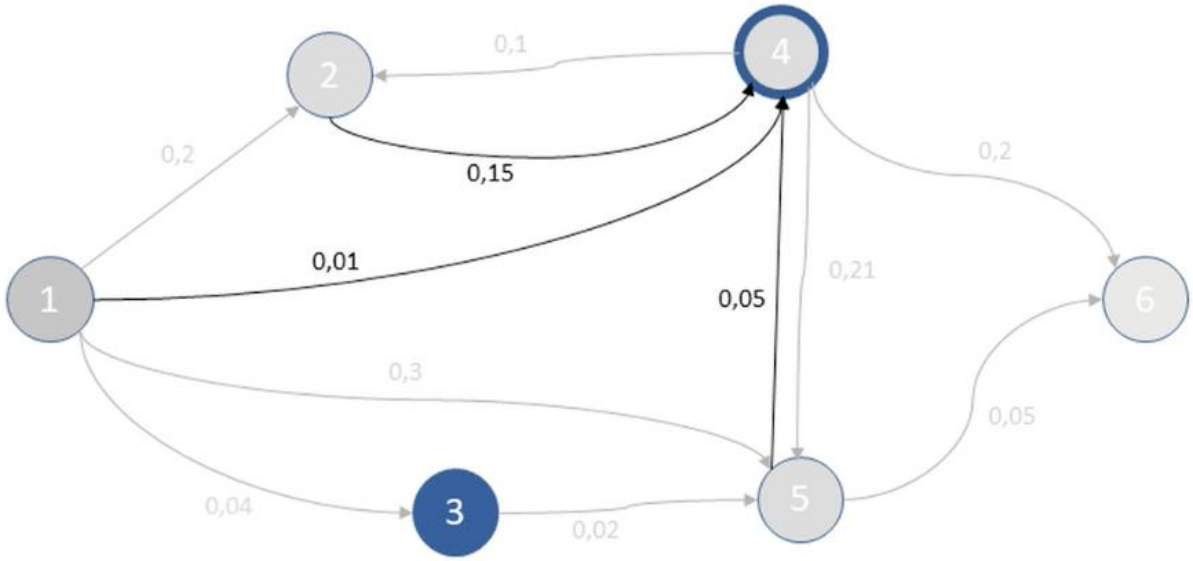


$$p_{vw} = \frac{\sum credit_{uv}}{A_v}$$

$$credit_{uv} = \frac{1}{\sum_{w \in S} I(t_w = t_v - 1)}$$

	u1	u2	u3	u4	u5	u6
t0	1	0	0	0	0	0
t1	0	1	0	0	1	0
t2	0	0	0	1	0	0
t3	0	0	0	0	0	1

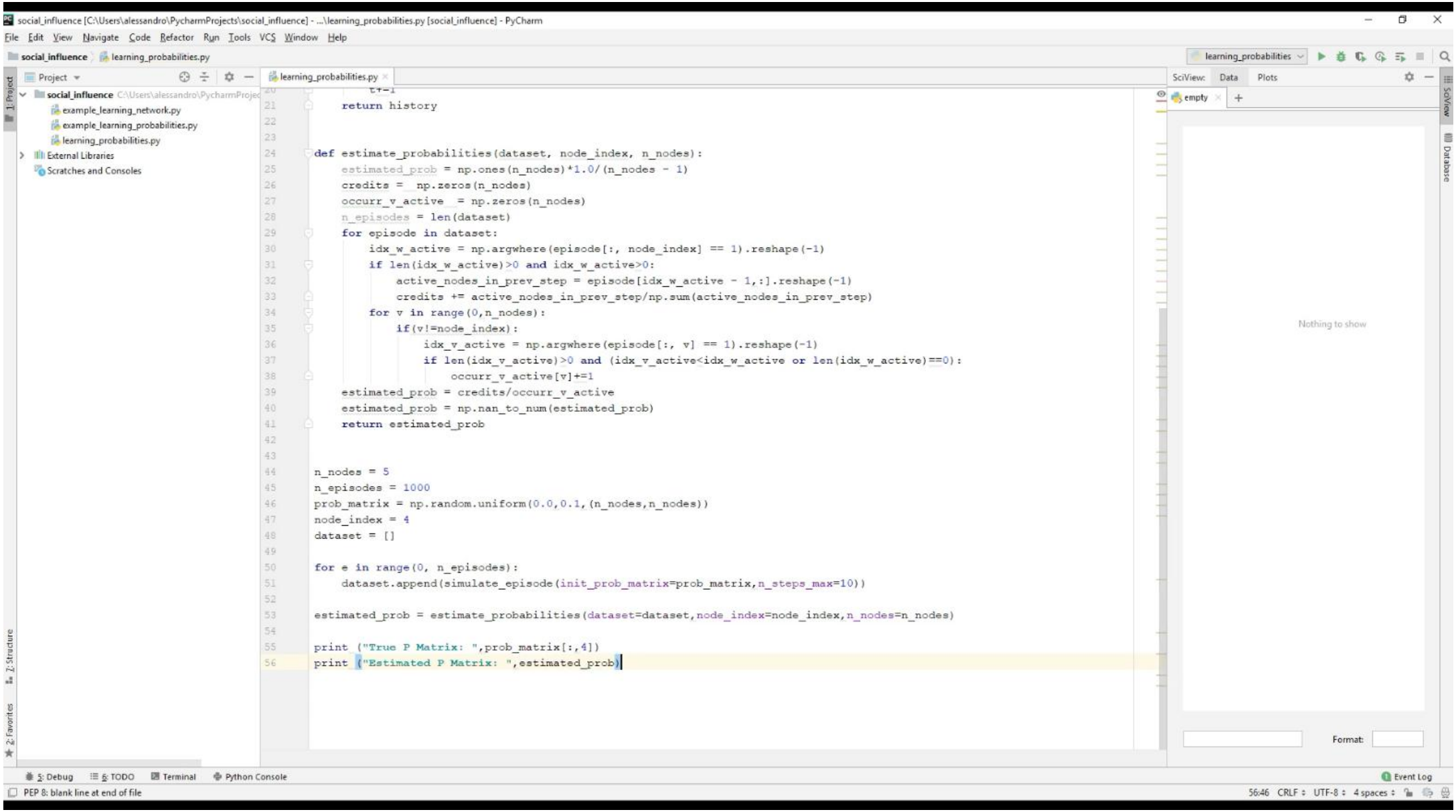
Credit Assignment



$$p_{vw} = \frac{\sum credit_{uv}}{A_v}$$

$credit_{24} = 1/2$ $credit_{54} = 1/2$

	u1	u2	u3	u4	u5	u6
t0	1	0	0	0	0	0
t1	0	**1**	0	0	**1**	0
t2	0	0	0	1	0	0
t3	0	0	0	0	0	1


social_influence [C:\Users\alessandro\PycharmProjects\social_influence] - ...\learning_probabilities.py [social_influence] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
social_influence learning_probabilities.py
learning_probabilities
Project
social_influence C:\Users\alessandro\PycharmProjec
example_learning_network.py
example_learning_probabilities.py
learning_probabilities.py
External Libraries
Scratches and Consoles
learning_probabilities.py
t+=1
return history
def estimate_probabilities(dataset, node_index, n_nodes):
estimated_prob = np.ones(n_nodes)*1.0/(n_nodes - 1)
credits = np.zeros(n_nodes)
occurr_v_active = np.zeros(n_nodes)
n_episodes = len(dataset)
for episode in dataset:
idx_w_active = np.argwhere(episode[:, node_index] == 1).reshape(-1)
if len(idx_w_active)>0 and idx_w_active>0:
active_nodes_in_prev_step = episode[idx_w_active - 1,:].reshape(-1)
credits += active_nodes_in_prev_step/np.sum(active_nodes_in_prev_step)
for v in range(0,n_nodes):
if(v!=node_index):
idx_v_active = np.argwhere(episode[:, v] == 1).reshape(-1)
if len(idx_v_active)>0 and (idx_v_active<idx_w_active or len(idx_w_active)==0):
occurr_v_active[v]+=1
estimated_prob = credits/occurr_v_active
estimated_prob = np.nan_to_num(estimated_prob)
return estimated_prob
n_nodes = 5
n_episodes = 1000
prob_matrix = np.random.uniform(0.0,0.1,(n_nodes,n_nodes))
node_index = 4
dataset = []
for e in range(0, n_episodes):
dataset.append(simulate_episode(init_prob_matrix=prob_matrix,n_steps_max=10))
estimated_prob = estimate_probabilities(dataset=dataset,node_index=node_index,n_nodes=n_nodes)
print ("True P Matrix: ",prob_matrix[:,4])
print ("Estimated P Matrix: ",estimated_prob)
SciView: Data Plots
empty
Nothing to show
Format:
5: Debug
6: TODO
Terminal
Python Console
Event Log
PEP 8: blank line at end of file
56:46 CRLF UTF-8 4 spaces