MAB Demo1: 어떻게 동작하는가? (ϵ-greedy)

문제 정의

- 3개의 arm이 주어지고 arm을 당길 수 있는 총 기회는 **100회**
- 각 arm에서 주어지는 reward는 **0이나 1**
- 1st arm에서 reward가 1이 나올 확률: **0.8**
- 2nd arm에서 reward가 1이 나올 확률: **0.6**
- 3rd arm에서 reward가 1이 나올 확률: **0.5**
- **3개의 arm 중에서 가장 높은 reward를 주는 arm을 찾아라!**

가정

- 각 arm은 서로 독립적으로 동일한 확률 분포로 매 시점마다 reward의 분포가 변경됨.
    - 즉, reward는 매 시점에 의존하지 않는 i.i.d(independent identically distributed) 분포임.

예시

- $\epsilon = 0.1$ 이라면?
- 매 시점마다 앞면이 나올 확률이 **90%**이고, 뒷면이 나올 확률이 **10%**인 동전을 던짐.
    - 앞면이면 지금까지의 평균 보상값이 가장 높은 arm을 선택
    - 뒷면이면 평균 보상값을 무시하고 랜덤하게 arm을 선택

패키지 로드

In [1]:

```
import os, sys
module_path = os.path.abspath(os.path.join('..'))
if module_path not in sys.path:
    sys.path.append(module_path)
from mab import algorithm as bd
from mab import arm
from mab import scorer as sc
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
np.set_printoptions(precision = 2)
```

파라메터 설정

In [2]:

```
num_draws = 100
print('total number of draws: {}'.format(num_draws))

arms = [
    arm.BernoulliArm(0.8),
    arm.BernoulliArm(0.6),
    arm.BernoulliArm(0.5)
]
num_arms = len(arms)
print('number of arms: {}'.format(num_arms))
algorithm = bd.EpsilonGreedyAlgorithm(num_arms, 0.1)
print('algorithm: ' + str(algorithm))

scorers = [
    sc.AverageRewardScorer(),
    sc.BestArmSelectedScorer(arms),
    sc.CumulativeRewardScorer()
]
```

```
total number of draws: 100
number of arms: 3
algorithm: EpsilonGreedy(epsilon=0.1)
```

알고리즘

In [3]:

```
avg_score, best_score, cum_score = 0.0, 0.0, 0.0
for i in range(num_draws):
    selected_arm = algorithm.select_arm()
    reward = arms[selected_arm].draw()
    algorithm.update(selected_arm, reward)

    print('{0:d}, selected_arm: {1}, reward_of_selected_arm: {2}, '
          'avg_reward: {3}'.format(i + 1, selected_arm, reward, algorithm.ave
rages))
    #input()
    draw = i + 1
    avg_score = scorers[0].update_score(draw, selected_arm, reward)
    best_score = scorers[1].update_score(draw, selected_arm, reward)
    cum_score = scorers[2].update_score(draw, selected_arm, reward)

print('avg_reward: {}, best_selected: {}, cum_reward: {}'.format(avg_score, b
est_score, cum_score))
```

```
1, selected_arm: 2, reward_of_selected_arm: 0, avg_reward: [ 0.  0
.  0.]
2, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 1.  0
.  0.]
3, selected_arm: 1, reward_of_selected_arm: 0, avg_reward: [ 1.  0
.  0.]
4, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 1.  0
.  0.]
```

```
5, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 1.  0
.  0.]
6, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 1.  0
.  0.]
7, selected_arm: 2, reward_of_selected_arm: 0, avg_reward: [ 1.  0
.  0.]
8, selected_arm: 0, reward_of_selected_arm: 0, avg_reward: [ 0.8
0.   0. ]
9, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.83
0.    0.  ]
10, selected_arm: 0, reward_of_selected_arm: 0, avg_reward: [ 0.71
0.    0.  ]
11, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.75
0.    0.  ]
12, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.78
0.    0.  ]
13, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.8
0.   0. ]
14, selected_arm: 0, reward_of_selected_arm: 0, avg_reward: [ 0.73
0.    0.  ]
15, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.75
0.    0.  ]
16, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.77
0.    0.  ]
17, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.79
0.    0.  ]
18, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.8
0.   0. ]
19, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.81
0.    0.  ]
20, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.82
0.    0.  ]
21, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.83
0.    0.  ]
22, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.84
0.    0.  ]
23, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.85
0.    0.  ]
24, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.86
0.    0.  ]
25, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.86
0.    0.  ]
26, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.87
0.    0.  ]
27, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.87
0.    0.  ]
28, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.88
0.    0.  ]
29, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.88
0.    0.  ]
30, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.89
0.    0.  ]
31, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.89
0.    0.  ]
32, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.9
0.   0. ]
33, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.9
0.   0. ]
34, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.9
0.   0. ]
35, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.91
0.    0.  ]
36, selected_arm: 2, reward_of_selected_arm: 1, avg_reward: [ 0.91
0.    0.33]
37, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.91
0.    0.33]
38, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.91
0.    0.33]
39, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.91
0.    0.33]
40, selected_arm: 0, reward_of_selected_arm: 0, avg_reward: [ 0.89
0.    0.33]
41, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.89
0.    0.33]
42, selected_arm: 1, reward_of_selected_arm: 0, avg_reward: [ 0.89
0.    0.33]
43, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.89
0.    0.33]
44, selected_arm: 0, reward_of_selected_arm: 0, avg_reward: [ 0.87
0.    0.33]
45, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.87
0.    0.33]
46, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.88
0.    0.33]
47, selected_arm: 0, reward_of_selected_arm: 0, avg_reward: [ 0.86
0.    0.33]
48, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.86
0.    0.33]
49, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.86
0.    0.33]
50, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.87
0.    0.33]
51, selected_arm: 0, reward_of_selected_arm: 0, avg_reward: [ 0.85
0.    0.33]
52, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.85
0.    0.33]
53, selected_arm: 0, reward_of_selected_arm: 0, avg_reward: [ 0.83
0.    0.33]
54, selected_arm: 0, reward_of_selected_arm: 0, avg_reward: [ 0.82
0.    0.33]
55, selected_arm: 0, reward_of_selected_arm: 0, avg_reward: [ 0.8
0.    0.33]
56, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.8
0.    0.33]
57, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.81
0.    0.33]
58, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.81
0.    0.33]
59, selected_arm: 2, reward_of_selected_arm: 1, avg_reward: [ 0.81
0.    0.5 ]
60, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.81
0.    0.5 ]
61, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.82
0.    0.5 ]
```

```
62, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.82
0.    0.5 ]
63, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.82
0.    0.5 ]
64, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.83
0.    0.5 ]
65, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.83
0.    0.5 ]
66, selected_arm: 0, reward_of_selected_arm: 0, avg_reward: [ 0.82
0.    0.5 ]
67, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.82
0.    0.5 ]
68, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.82
0.    0.5 ]
69, selected_arm: 0, reward_of_selected_arm: 0, avg_reward: [ 0.81
0.    0.5 ]
70, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.81
0.    0.5 ]
71, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.82
0.    0.5 ]
72, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.82
0.    0.5 ]
73, selected_arm: 1, reward_of_selected_arm: 1, avg_reward: [ 0.82
0.33  0.5 ]
74, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.82
0.33  0.5 ]
75, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.82
0.33  0.5 ]
76, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.83
0.33  0.5 ]
77, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.83
0.33  0.5 ]
78, selected_arm: 2, reward_of_selected_arm: 0, avg_reward: [ 0.83
0.33  0.4 ]
79, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.83
0.33  0.4 ]
80, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.83
0.33  0.4 ]
81, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.84
0.33  0.4 ]
82, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.84
0.33  0.4 ]
83, selected_arm: 0, reward_of_selected_arm: 0, avg_reward: [ 0.83
0.33  0.4 ]
84, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.83
0.33  0.4 ]
85, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.83
0.33  0.4 ]
86, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.83
0.33  0.4 ]
87, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.84
0.33  0.4 ]
88, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.84
0.33  0.4 ]
89, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.84
0.33  0.4 ]
90, selected_arm: 0, reward_of_selected_arm: 0, avg_reward: [ 0.83
0.33  0.4 ]
91, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.83
0.33  0.4 ]
92, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.83
0.33  0.4 ]
93, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.84
0.33  0.4 ]
94, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.84
0.33  0.4 ]
95, selected_arm: 0, reward_of_selected_arm: 0, avg_reward: [ 0.83
0.33  0.4 ]
96, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.83
0.33  0.4 ]
97, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.83
0.33  0.4 ]
98, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.83
0.33  0.4 ]
99, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.84
0.33  0.4 ]
100, selected_arm: 0, reward_of_selected_arm: 1, avg_reward: [ 0.8
4  0.33  0.4 ]
avg_reward: 0.8, best_selected: 0.92, cum_reward: 80.0
```

실험 결과 (매 실험마다 결과는 바뀜)

- 1, 선택된 arm: **1**, 선택된 arm의 보상: **0**, 평균 보상: **[0. 0. 0.]**
- 2, 선택된 arm: **0**, 선택된 arm의 보상: **1**, 평균 보상: **[1. 0. 0.]**
- 3, 선택된 arm: **0**, 선택된 arm의 보상: **1**, 평균 보상: **[1. 0. 0.]**
- 4, 선택된 arm: **2**, 선택된 arm의 보상: **1**, 평균 보상: **[1. 0. 1.]**
- 5, 선택된 arm: **2**, 선택된 arm의 보상: **0**, 평균 보상: **[1. 0. 0.5]**
- ... (11, 12 주목!)
- 11, 선택된 arm: **0**, 선택된 arm의 보상: **1**, 평균 보상: **[1. 0. 0.5.]**
- 12, 선택된 arm: **2**, 선택된 arm의 보상: **1**, 평균 보상: **[1. 0. 0.67]** **(동전이 뒷면이 나왔군요!)**
- ...
- 100, 선택된 arm: **0**, 선택된 arm의 보상: **1**, 평균 보상: **[0.81 0. 0.67]**
- 평균 보상: **0.79**, 최적 arm이 선택될 확률: **0.93**, 누적 보상: **79.0**