Number Recommendation System using K-Arm Bandit

Idea

- model recommends a number between 0 and 9 (10 numbers)
- recommending each number is an action -> K = 10, 10 actions
- rewarding system is binary : {1 is user likes the number and 0 if the number is disliked}

Alogrithm

- Set K to 10
- Intialize arrays of estimated rewards, confidence intervals and pulls
- Recommendation :
    - choose a random number at P(E) = 0.1 (Exploring) else choose maximum reward (Exploiting)
- Learning:
    - pass the recommended number along side the user reward (like / dislike)
    - update the pull, estimated reward and confidence interval of the recommended number

In [1]:
```
import numpy as np
```

In [2]:
```
K = 10
```

In [3]:
```
estimated_rewards = np.zeros(K)
confidence_intervals = np.ones(K) * float('inf')
```

In [4]:
```
pulls = np.zeros(K, dtype=int)
```

In [5]:
```
def recommend_item():
    if np.random.rand() < 0.1:
        return np.random.randint(K)
    else:
        return np.argmax(estimated_rewards + confidence_intervals)
```

In [19]:
```
def update_estimates(item_idx, reward):
    pulls[item_idx] += 1
    estimated_rewards[item_idx] += (reward - estimated_rewards[item_i
    confidence_intervals[item_idx] = np.sqrt(2 * np.log(pulls.sum())
    return estimated_rewards, confidence_intervals
```

In [25]:
```
# lets try recommending 1000 numbers
NUMS = 1000
```

In [26]:

```
# storing random rewards for 1000 cases, 1 meaning number recommended
# these are user inputs but for the purpose of an example rewards are
rewards = list()
for i in range(NUMS):
  rewards.append(np.random.randint(2))
```

In [27]:

```
rewards
```

Out[27]:

```
[0,
 0,
 0,
 1,
 0,
 0,
 1,
 1,
 0,
 1,
 1,
 0,
 1,
 0,
 1,
 1,
 0,
 0,
 0,
 0
```

In [28]:

```
recommended_nums = list()
for i in range(NUMS):
    item_idx = recommend_item()
    recommended_nums.append(item_idx)
    # reward = input("Do you like the number:")
    reward = rewards[i]
    ER, CI = update_estimates(item_idx, reward)
```

In [35]:

```
pulls
```

Out[35]:

```
array([154,  77, 112,  88, 119, 105, 106,  77, 150, 113])
```

In [30]:

```
ER
```

Out[30]:

```
array([0.52597403, 0.38961039, 0.47321429, 0.42045455, 0.48739496,
       0.45714286, 0.46226415, 0.4025974 , 0.52      , 0.49557522])
```

In [31]:

```
CI
```

Out[31]:

```
array([0.30106187, 0.42641162, 0.35326066, 0.39861057, 0.34296065,
       0.36477369, 0.36326437, 0.42442843, 0.30535259, 0.35208542])
```

```
In [32]: recommended_nums
```

```
Out[32]: [5,
          7,
          7,
          4,
          4,
          6,
          8,
          8,
          8,
          7,
          9,
          9,
          4,
          1,
          3,
          3,
          3,
          5,
          4,
```

In [43]:

```
for i in range(10):
  print(i)
  print(f'number of times {i} was recommended is : {pulls[i]}')
  print(f'estimated reward of {i} is : {ER[i]}')
  print(f'confidence interval of {i} is : {CI[i]}')
  print(f'chance of {i} being chosen is : {pulls[i]/NUMS * 100}%')
  print("---------------------------------------------------------")
```

```
0
number of times 0 was recommended is : 154
estimated reward of 0 is : 0.5259740259740264
confidence interval of 0 is : 0.30106187125833284
chance of 0 being chosen is : 15.4%
----------------------------------------------------------
1
number of times 1 was recommended is : 77
estimated reward of 1 is : 0.38961038961038963
confidence interval of 1 is : 0.4264116193281533
chance of 1 being chosen is : 7.7%
----------------------------------------------------------
2
number of times 2 was recommended is : 112
estimated reward of 2 is : 0.47321428571428575
confidence interval of 2 is : 0.35326065806385376
chance of 2 being chosen is : 11.200000000000001%
----------------------------------------------------------
3
number of times 3 was recommended is : 88
estimated reward of 3 is : 0.4204545454545455
confidence interval of 3 is : 0.398610566591265
chance of 3 being chosen is : 8.799999999999999%
----------------------------------------------------------
4
number of times 4 was recommended is : 119
estimated reward of 4 is : 0.4873949579831932
confidence interval of 4 is : 0.3429606548776809
chance of 4 being chosen is : 11.899999999999999%
----------------------------------------------------------
5
number of times 5 was recommended is : 105
estimated reward of 5 is : 0.4571428571428571
confidence interval of 5 is : 0.3647736889435815
chance of 5 being chosen is : 10.5%
----------------------------------------------------------
6
number of times 6 was recommended is : 106
estimated reward of 6 is : 0.46226415094339646
confidence interval of 6 is : 0.36326436589970257
chance of 6 being chosen is : 10.6%
----------------------------------------------------------
7
number of times 7 was recommended is : 77
estimated reward of 7 is : 0.4025974025974025
confidence interval of 7 is : 0.42442842989635904
chance of 7 being chosen is : 7.7%
----------------------------------------------------------
8
number of times 8 was recommended is : 150
estimated reward of 8 is : 0.5200000000000002
confidence interval of 8 is : 0.3053525857743899
chance of 8 being chosen is : 15.0%
----------------------------------------------------------
9
number of times 9 was recommended is : 113
estimated reward of 9 is : 0.4955752212389382
confidence interval of 9 is : 0.3520854214715383
chance of 9 being chosen is : 11.3%
----------------------------------------------------------
```

```
In [ ]:
```