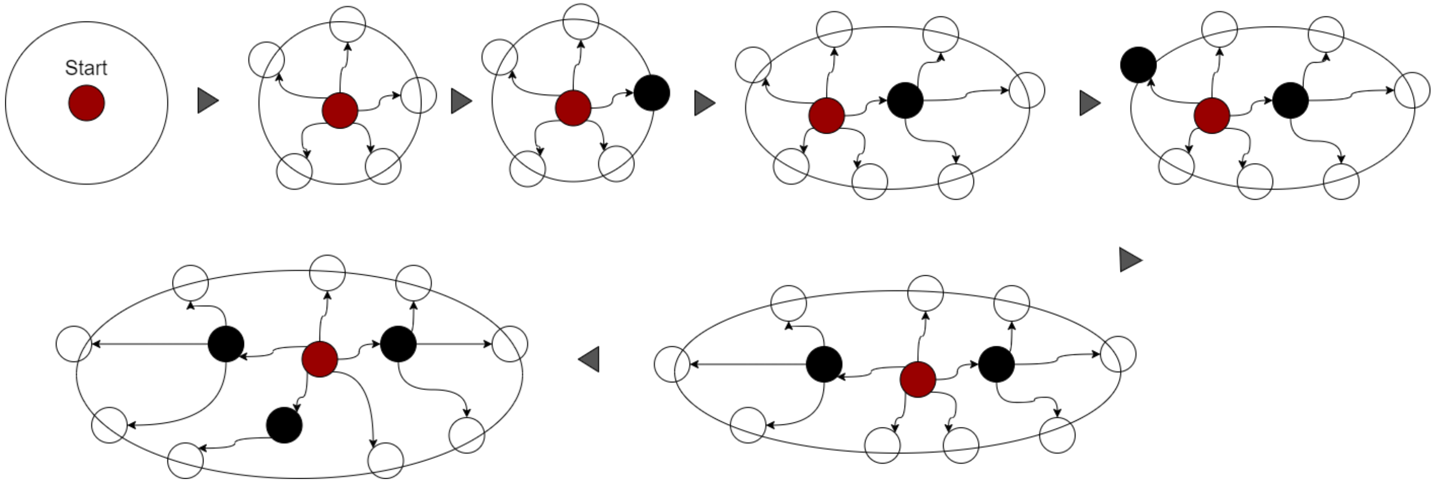
人工智慧作業報告簡報

信息工程學院 2021 級 干皓丞 2101212850

Performance Effectiveness of Multimedia Information Search Using the Epsilon-Greedy Algorithm
使用 Epsilon-Greedy 演算法進行多媒體資訊搜尋的效能表現
(研究者想知道 Epsilon-Greedy 演算法在此運用會不會比較棒)
https://ieeexplore.ieee.org/document/8999097
Published in: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)

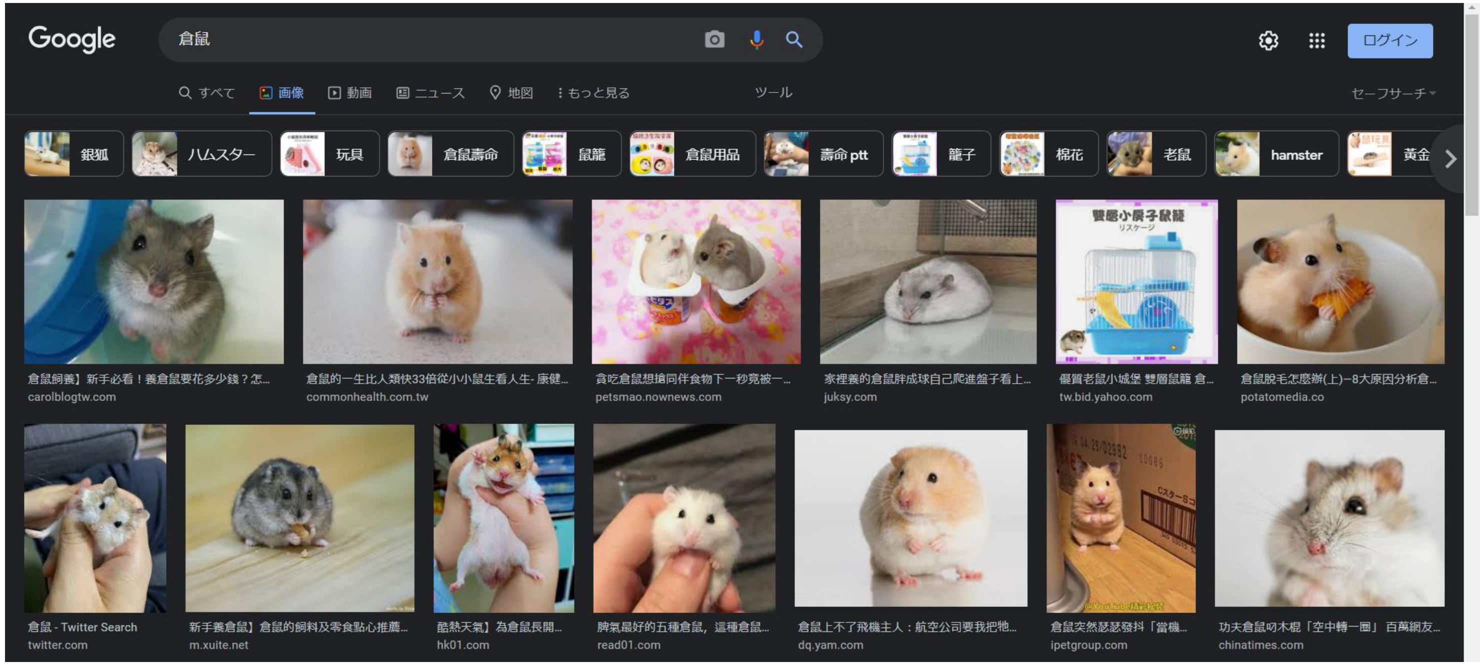
Motivation 动机

貪婪演算法來處理多媒體搜尋的問題，也就是所謂的多媒體資訊檢索 (MIR; multimedia information retrieval)。



研究團隊想要的解決過往都用手工作業的問題 ...

同時也想找出搜尋系統中不容易找到的對象。



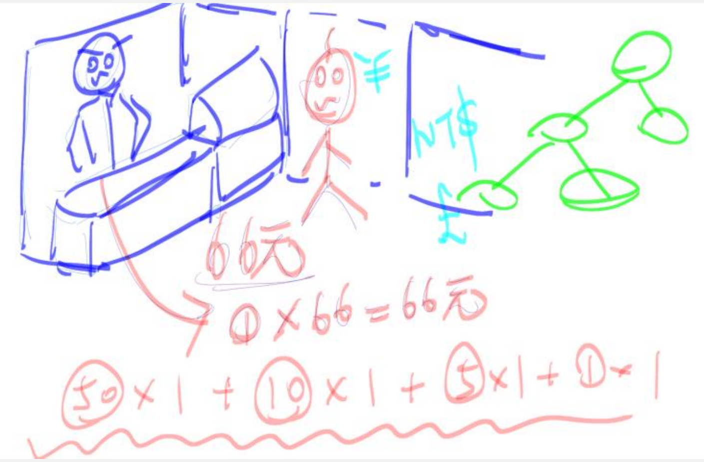
Motivation 动机

生活上，我們 Greedy 演算法會用到的 ...

超商找錢

O：
50 元 x 1 ＋ 10 元 x 1 ＋ 5 元 x 1 ＋ 1 元 x 1

X：
~~1 元 x 66 = 66 元~~



補充

Motivation 动机

C++ 進行實踐

Windows PowerShell

```
(base) PS D:\USERDATA\Desktop\t> g++ -o owo owo.cpp
(base) PS D:\USERDATA\Desktop\t> ./owo
1567
1567=1000*1 500*1 50*1 10*1 5*1 1*2
```

D:\USERDATA\Desktop\t\owo.cpp - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

owo.cpp

```cpp
#include<iostream>
using namespace std;

int main()
{
    int cash[8];
    cash[0] = 2000;
    cash[1] = 1000;
    cash[2] = 500;
    cash[3] = 100;
    cash[4] = 50;
    cash[5] = 10;
    cash[6] = 5;
    cash[7] = 1;

    int n, i;

    while( cin >> n )
    {
        cout << n << "=";
        for( i = 0 ; i <= 7 ; i++ )
        {
            if( n >= cash[i] )
            {
                cout << cash[i] << "*" << n/cash[i] << " ";
                n = n%cash[i];
            }
        }
        cout << endl;
    }
    return 0;
}
```

補充

Motivation 动机

生活上，我們 Epsilon-Greedy 演算法會用到的 ...

留學顧問諮詢

也就是所謂多臂吃角子老虎問題 ...

學生來了不用指定顧問，由演算法來決定哪位顧問進行諮詢：

步驟 1: 學生 user = 1...T 逐一過來

步驟 2: 給學生推薦顧問，學生接受則留下(reward=1)，拒絕則離開(reward=0)

步驟 3: 記錄選擇接受的學生總數 total_reward += reward

當學生到來時 :

Epsilon-greedy 的機率來選擇探索 (Exploration) ，从 N 個顧問中隨機選擇(Epsilon/N)一個讓學生被諮詢，根據學生回饋好顧問的機率是 { q1, q2, q3, ... qN}

以 1 - Epsilon 的機率選擇利用 (Exploitation)，從 N 個顧問 { q1, q2, q3, ... qN} 中選擇機率最好的顧問推薦給學生。

Motivation 动机

補充

Python 進行實踐

```
Windows PowerShell
(base) PS D:\USERDATA\Desktop\test-code\epsilon-greedy> python .\eps-greedy.py
total_reward=9243
(base) PS D:\USERDATA\Desktop\test-code\epsilon-greedy>
```

eps-greedy.py

```python
import numpy as np
T = 10000 # 學生數量 T
N = 20 # 顧問數 N
true_rewards = np.random.uniform( low = 0, high = 1, size = N) # 被評好顧問的機率
estimated_rewards = np.zeros(N)
number_of_trials = np.zeros(N)
total_reward = 0
def alpha_greedy( N, alpha = 0.1):
    item = 0
    if np.random.random() < alpha:
        item = np.random.randint( low = 0, high = N)
    else:
        item = np.argmax(estimated_rewards)
    reward = np.random.binomial( n = 1, p = true_rewards[item])
    return item, reward
for t in range(1, T): # T個學生逐一進入中心諮詢
   # 從 N 個顧問中推薦一位, reward = 1 表示學生接受, reward = 0 表示學拒絕並離開。
   item, reward = alpha_greedy(N)
   total_reward += reward # 一共有多少學生接受(也就是願意請顧問申請學校)

   # 更新成功機率
   number_of_trials[item] += 1
   estimated_rewards[item] = ((number_of_trials[item] - 1) * estimated_rewards[item] + reward) / number_of_trials[item]

print("total_reward=" + str(total_reward))
```

補充

Framework 框架

Epsilon-Greedy Algorithm
Epsilon 貪婪演算法

提出 EGSE-A & EGSE-B 兩個變體

+

The initial indexing
搜尋系統中的初值索引

過去搜尋的對象是根據索引分數排名，初始值大的就會主導呈現的結果 ...

即當給定的隨機對象 Z 已包含在先前的 M-list

EGSE-A 可以被重新選擇

EGSE-B 將被排除

兩者代表對容錯與效率的重視

Framework 框架

研究團隊在 Reinforcement Learning 框架下，使用Epsilon-Greedy 演算法進行研究。

Algorithm 1 EGSE-A: Search Space Exploration with Constant Probability

Require: epsilon E, length of result list *M*, query max counter *C*

```
Initialize terminating condition Δ ← False,
Initialize query counter Θ ← 0,
Initialize exploration proportion R ← E × M,
Initialize exploitation proportion K ← (1 − E) × M,
while Δ == False do
    Retrieve and parse new user query Q
    Determine S_1 = {O_i | objects with the highest relevant scores}_{i=1}^{k}, where |S_1| = K
    Determine S_2 = {O_j | O_j ∈ S_1^∁}_{j=1}^{R}, where |S_2| = R
    Present M-list := S_1 ∪ S_2 to user
    Capture object click information from user
    Increment the score of clicked objects
    Θ ← Θ + 1
    if Θ == C then
        Δ ← True
```

Algorithm 2 EGSE-B: Search Space Exploration with Variable Probability

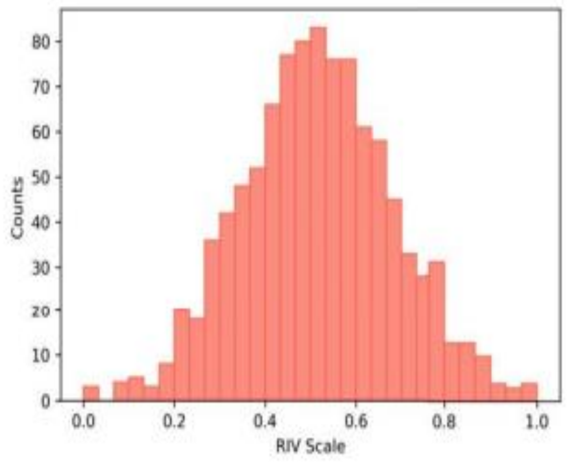
Require: epsilon E, length of result list *M*, query max counter *C*

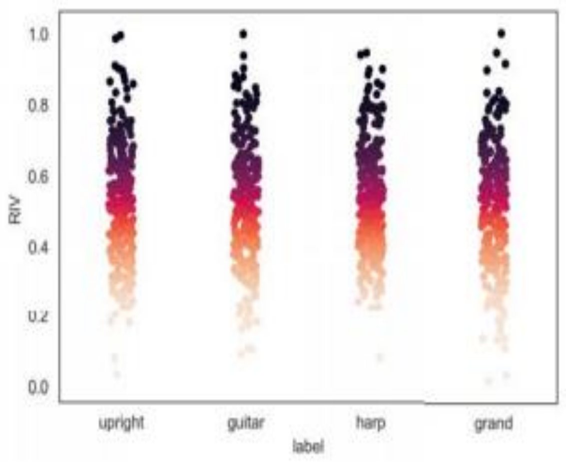
```
Initialize terminating condition Δ ← False,
Initialize query counter Θ ← 0,
Initialize exploration proportion R ← E × M,
Initialize exploitation proportion K ← (1 − E) × M,
Initialize previously presented M-list for Query Q_i as S_i ← ∅, for all possible i
while Δ == False do
    Retrieve and parse new user query Q_i
    Determine S_1 = {O_l | objects with the highest relevant scores}_{l=1}^{k}, where |S_1| = K
    if |(S_1 ∪ S_i)^∁| ≥ R then
        Determine S_2 = {O_j | O_j ∈ (S_1 ∪ S_i)^∁}_{j=1}^{R}, where |S_2| = R
    else
        Determine S_2 = (S_1 ∪ S_i)^∁, where |S_2| = |(S_1 ∪ S_i)^∁|
    Present M-list := S_1 ∪ S_2 for query Q_i to user
    S_i ← S_i ∪ S_1 ∪ S_2
    Capture object click information from user
    Increment the score of clicked objects
    Θ ← Θ + 1
    if Θ == C or (S_1 ∪ S_i)^∁ == ∅ then
        Δ ← True
```

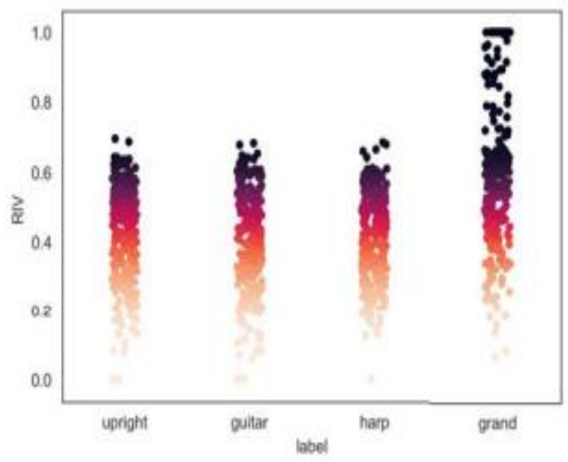
Intuition 直觉

當中的 RIV 為相關索引值初始化

測試的每張圖像屬於分為四類各 25 %：三角鋼琴 - grand piano、立式鋼琴 - upright piano、古典吉他 - classical guitar、豎琴 - harp







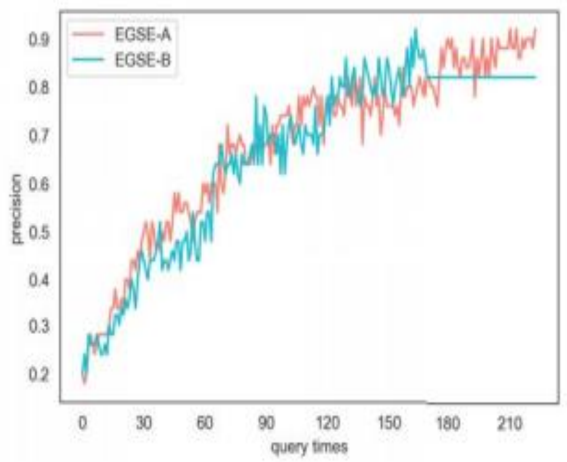


Fig. 3. Distribution of Initial RIV Scores. Fig. 4. Distribution of Initial RIV Scores for Each Category (EGSE-B). Fig. 5. Distribution of RIV Scores for Each Category When Hidden Object X is Discovered (EGSE-B). Fig. 6. Evolution of Query Precisions against Query Times. Settings: $N = 1000, M = 50, \epsilon = 0.1$.

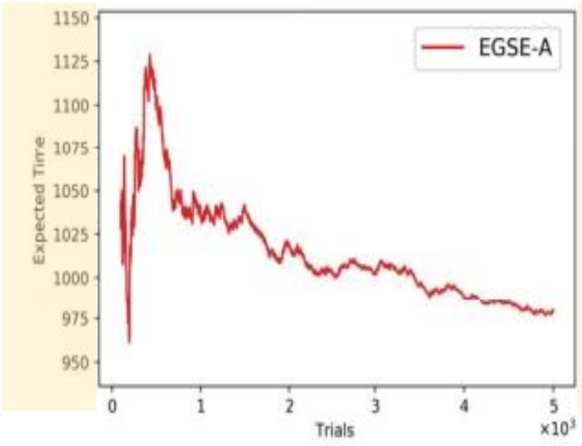
EGSE-B 初始 RIV

找到隱藏對象的 RIV 分數分布

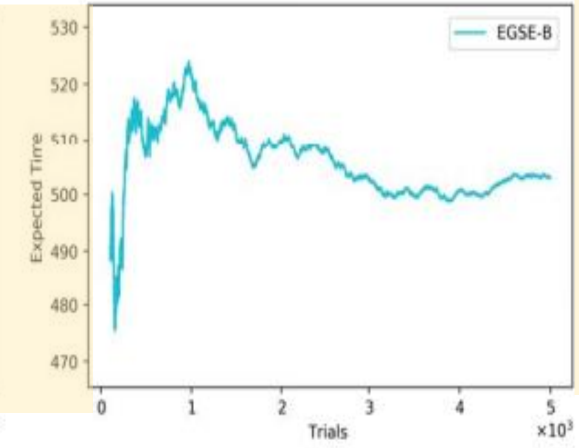
查詢的精確度隨時間越來越好

Justification 理由

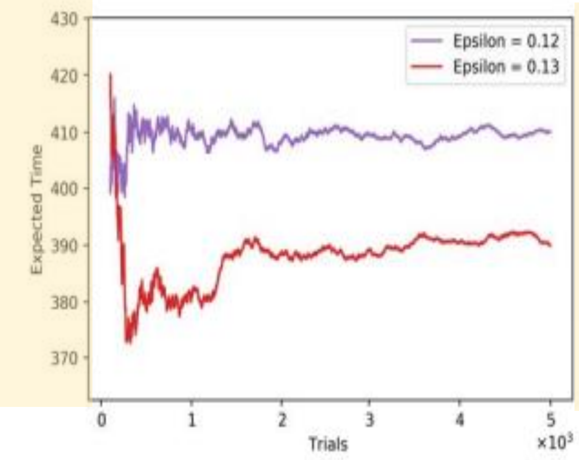
EGSE-A發現時間



EGSE-B發現時間



EGSE-B Epsilon 在 0.12 跟 0.13 的 發現時間



時間限制條件下，EGSE-B發現相關對象的機率

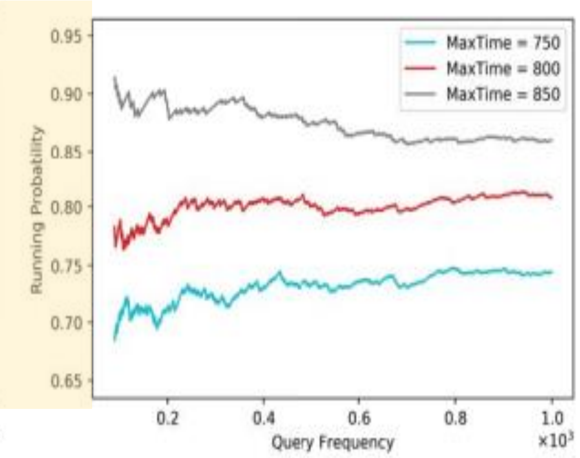


Fig. 7. Expected Discovery Time of EGSE-A. Fig. 8. Expected Discovery Time of EGSE-B. Settings: N = 10000, M = 100, ϵ = 0.1. Fig. 9. Expected Discovery Time of EGSE-B with $\epsilon = \{0.12, 0.13\}$. Fig. 10. Probability of Discovering the Most Relevant Object in EGSE-B with Time Constraints.

Result 结果

三角鋼琴 - grand piano

Fig. 1. Sample Images from Dataset (Size = 50). Fig. 2. Final Returned M-list using EGSE-A. Settings: $N = 1000$, $M = 50$, $\epsilon = 0.1$.

Reference

https://www.csie.ntu.edu.tw/~b98902112/cpp_and_algo/cpp02/greedy.html
https://ithelp.ithome.com.tw/articles/10205466

https://jamesmccaffrey.wordpress.com/2017/11/30/the-epsilon-greedy-algorithm/
https://www.baeldung.com/cs/epsilon-greedy-q-learning
https://blog.csdn.net/lafengxiaoyu/article/details/102634543
https://zhuanlan.zhihu.com/p/32335683
https://www.geeksforgeeks.org/epsilon-greedy-algorithm-in-reinforcement-learning/
https://www.jianshu.com/p/590d98967a93