1 input *ci* ∈ ***M***

2 for *i* ∀ ***M*** :

3 if ( *c*i == 1 ) :

4 ***o*** = *ci* ∪ ***o***

5 if ( *y* == 1 ) :

6 ***p*** = 加入與***o***相似的景點

7 ***o*** = ***p*** ∪ ***o***

8 while ( len(***o***) > 0 ) :

9 for *i* in range( 0 , len(***o***) ) :

10 *probi* = normalization(***o***,*mi*,*ti*,*si*)

11 *k* =

12 ***x*** = ***x*** ∪ *rk*

13 Remove *rk* from ***o***

14 ***x =***

*ci*：使用者所選的景點

***M***：系統推薦或使用者輸入的景點集合

***o***：使用者選中的景點集合

*y*：使用者是否同意系統推薦相似的景點，是的話為1，否為0

***p***：推薦與 ***o*** 集合中相似的景點

*probi*：第 *i* 個景點被系統推薦的機率

*mi*：第 *i* 個景點的人潮流量

*ti*：第 *i* 個景點的營業時間

si：第 *i* 個景點的評分

*k*：***prob*** 集合中推薦機率最高的索引值

***x***：存放最後景點推薦順序的向量

*rk*：***o*** 集合中推薦機率最高的景點

* Filter-first（1-7行）

首先，由系統推薦或使用者自行輸入的景點作為M集合，之後將被選擇的景點放入***o***集合，再根據使用者是否同意系統推薦相似的景點，可以得到新的***o***集合。

* tour-second（8-14行）

將***o***集合中的每個景點，透過classifier放入參數***o***、*mi*、*ti*、*di*來產生每個景點被推薦的機率，之後找出推薦機率最大的景點放入***x***集合，當全部的行程都被放入***x***集合後，最後輸出***x***作為我們替使用者安排的最佳路線。