【智慧化企業整合】

Project #3

利用 RL 輔助消費決策

109034533 李珩慈

指導教授：邱銘傳 教授

一、背景介紹

1. 研究背景

忙碌的現代人常至超市購買平常日常所需的用品或食材，面對玲琅滿目的超市貨架，我們往往無法在多種選擇之下找出最適當的購買組合，因此本次專案將以強化學習 (Reinforcement Learning) 來探討如何在機器學習的輔助之下幫助消費者達成更佳的購物體驗。

2. 5W1H 分析法

本專案透過 5W1H 幫助我們進一步了解欲解決的問題並思考解決方法，以下將以事情 (what)、時間 (when)、人物 (who)、地點 (where)、原因 (why)、方法 (how) 對問題進行綜合分析。

What?	建立產品組合推薦系統
When?	準備做出消費行為時
Who?	精打細算的消費者
Where?	實體、線上超市
Why?	提升購物決策的品質
How?	建立強化學習模型，協助消費者迅速判斷

二、資料和模型架構

1. 資料建構

本專案資料參考公開資料，預計將分為五個欄位，分別為 Ingredient、Product、QMerged_label、Real_Cost、V_0，可參考下面兩表，以 Ingredient 來說，可以將其設想爲製造一道菜所需的各種食材種類，本次專案考慮了 4 種食材；另外 Product 則是每項食材的產品種類，舉例來說，若食材為白米，考慮的產品種類即可能有池上米或是泰國米等選擇，本次專案在四種食材底下分別設定了兩到三種產品種類，共計有九項產品，並以 Real_Cost 記錄每項產品的實際價格；最後，本專案建構 QMerged_label 和 V_0 兩個欄位以利進一步的模型建置，此兩個欄位的功能將於後續說明。

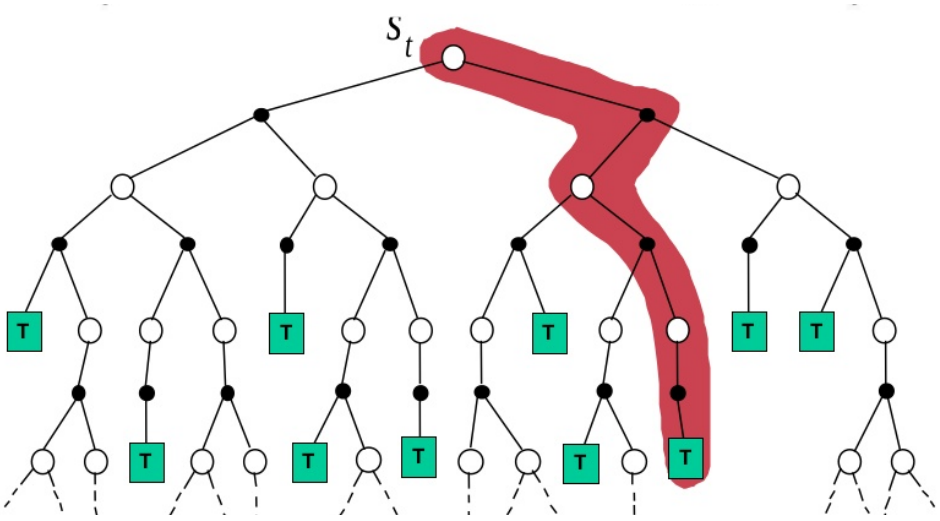
欄位名稱	資料型態	資料說明
Ingredient	類別	食材種類，以不同數字標號
Product	類別	每項食材的產品種類，以不同數字標號
QMerged_label	類別	食材標號和產品標號的合併 （以利後續的模型建立）
Real_Cost	數值	每項產品的實際價格
V_0	數值	每項產品的初始 V 值 （以利後續的模型建立）

Ingredient	Product	QMerged_label	Real_Cost	V_0
1	1	11	10	0
1	2	12	6	0
2	1	21	8	0
2	2	22	11	0
3	1	31	3	0
3	2	32	7	0
4	1	41	8	0
4	2	42	5	0
4	3	43	1	0

2. 模型建置

(1) 蒙地卡羅學習 (Monte Carlo Learning)

本專案將使用機器學習中其中一個常見的演算法 - 蒙地卡羅學習法，該演算法依據經驗求解最佳策略，透過函式 $V(a) \leftarrow V(a) + \alpha \times \big(G - V(a)\big)$ 獲得 Reward 並依樣本平均回報解決強化學習問題。如下圖所示，透過大量的抵達 Terminal 點時獲得的 Reward 估算最佳策略。



(2) 環境設定

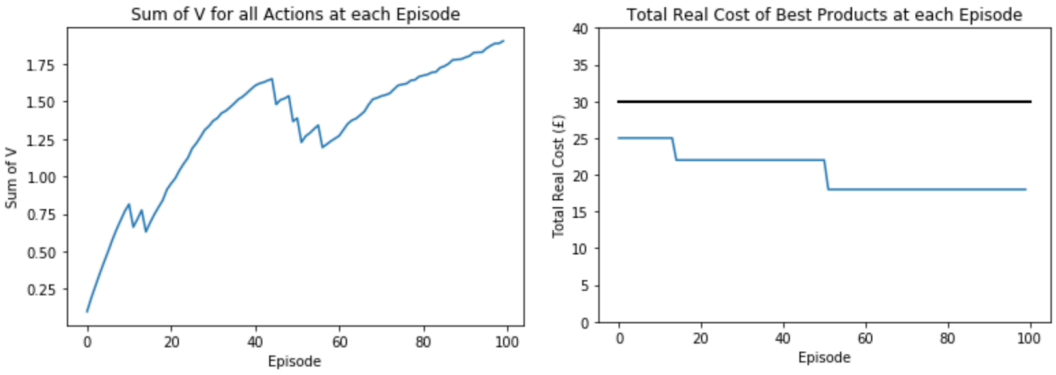
在環境中有幾個重要的元素，State 代表在特定時間點 agent 身處的狀態；Action 代表 agent 進行的動作；Reward 代表 environment 給予 agent 所做 action 的獎勵或懲罰。本專案將這幾個環境變數設定如以下：

- State（狀態）: 每一項食材
- Action（動作）: 選擇食材的特定品牌產品
- Reward（獎勵）: 是否有超過預算，無 R_T +1；有則 R_T - 1

將環境設定完成後，本專案的虛擬程式碼如下圖所示，若回合數內的產品組合超過預算 Reward 會得到－１，若無則得到１。

```
if(budget >= episode2['Real_Cost'].sum()):
        Return = 1
else:
        Return = -1
```

在撰寫程式碼後我們可以發現，在參數設定為學習率 0.1、Epsilon 0.5、Episode 為 100、預算為 30 元的情況下，V 值總和隨著回合數逐漸上升，每個回合數的產品組合的價錢也同時下降。

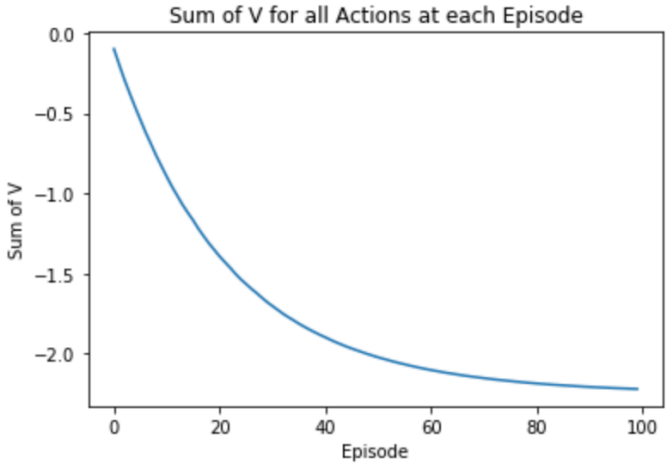


三、參數調整

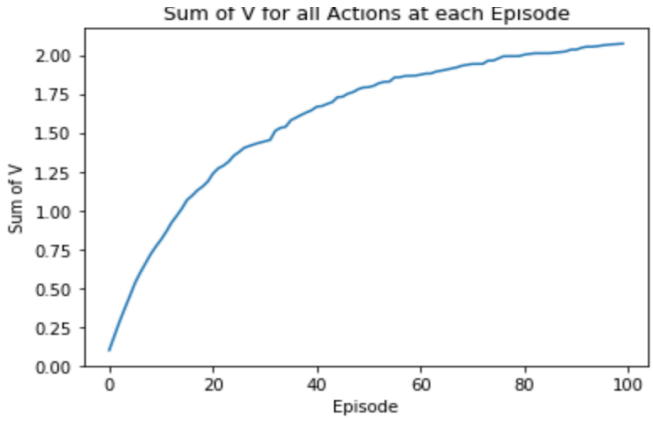
在參數調整中，本專案依據上述建置的模型分別調整預算、學習率、Epsilon 值、回合數四個參數，並依調整的結果再進一步延伸探討。

1. 預算 (Budget)

對商品組合的預算上限，在原先的模型中，預算的限制為 30 元，本專案將藉由調整預算大小觀察 V 值總和的變化程度。首先，我們將預算上限設定為非常低的數值 - 5 元，並重新建置原先的模型，下圖 V 值總和非常快速地即趨近於負值，透過觀察資料我們可以發現最便宜的產品組合也無法滿足此限制，因此 V 值總和必定會不斷下降。



接下來，將預算上限設定為非常高的數值 - 100 元，由下圖可以發現 V 值總和皆為正值。我們可以由資料了解到最貴的商品組合也不會超過 100 元，因此 V 值總和當然將為正數。



在觀察預算的調整後，我們得知預算的高低設定將嚴重影響到模型的適用性。另外，下圖為將預算設定為 23 元的最便宜組合的 V 值總和（綠線）與其他產品組合的 V 值總和（橘線），在回合數為 100 的情況下，最便宜的商品組合將大幅超越其他的產品組合。

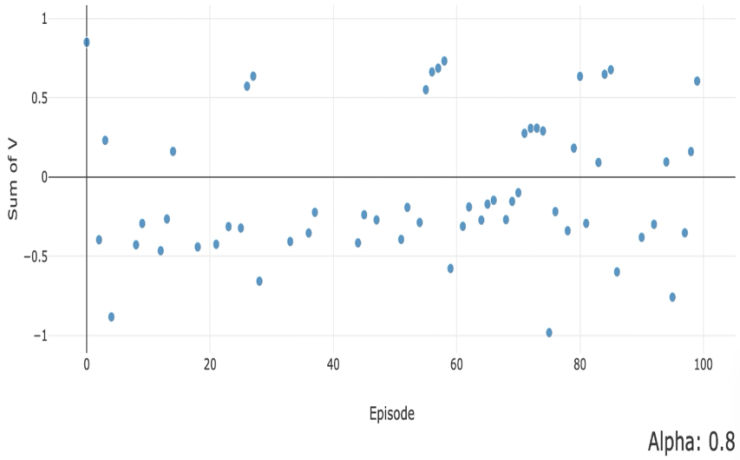
2. 學習率 (Alpha)

在原先的模型中，本專案將學習率設定為 0.1，在此章節分別上調和下修學習率並以 plotly 此 API 繪製動態圖表（完整圖表請詳見程式碼），協助我們更加了解調整間的 V 值總和變化，最後本專案選擇將學習率設定為 0.05。

(1) 將學習率設定為 1~0.1

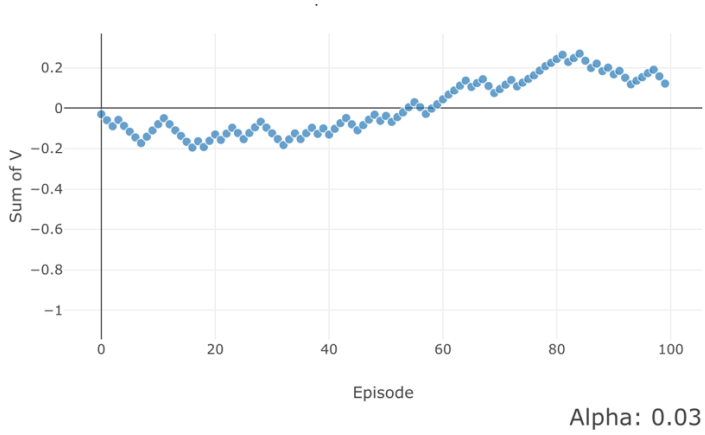
由下動態圖表可以發現 V 值總和曲線會隨著學習率的提升而變得更加平

滑，但當學習率為 0.1 時，V 值總和曲線還是有陡曲的現象。



(2) 將學習率設定為 0.1~0.01

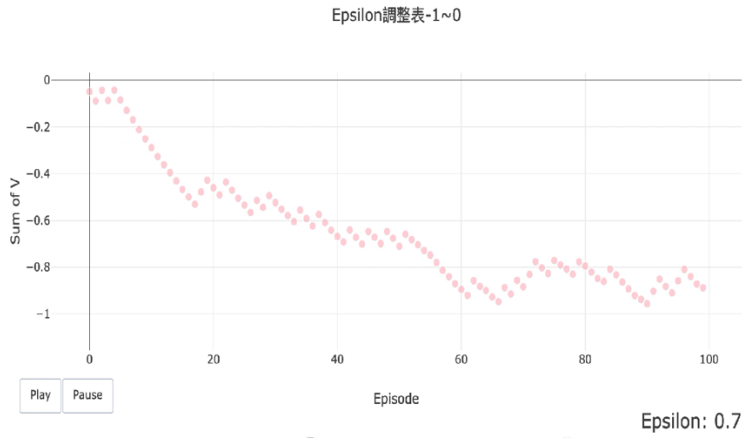
由下動態圖表可以發現平滑的現象隨著學習率的提升而變得更加明顯，但在學習率下降的同時，模型變得無法在 100 次的回合中收斂，且每個學習度的結果也不盡相同。



3. Epsilon 值

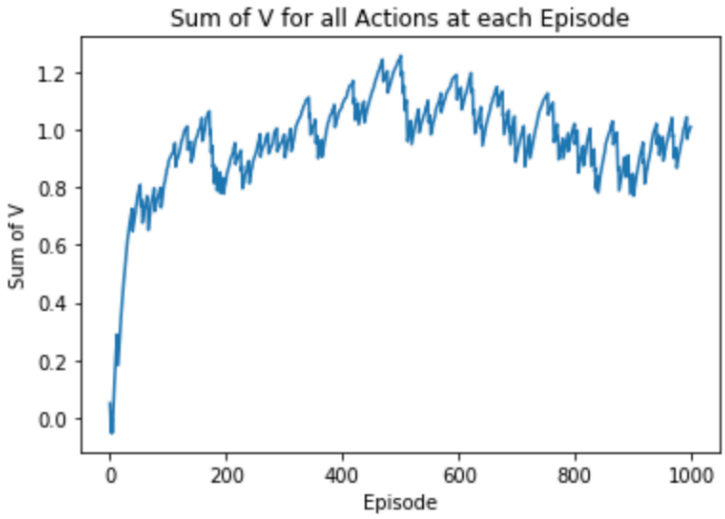
接下來將調整 Epsilon 值於 1~0.1 以觀察以不同程度的貪心算法在此模型中帶來的影響。另外將學習率設定為前面發現最佳值 0.05。由下面動態圖表 (完整圖表請詳見程式碼)可以觀察到當 Epsilon 值越大時代表有越大機率隨機選擇行為，因此模型中使得 V 值總和分布零星，而在模型中 Epsilon 值越小，

代表有越大機會選擇選擇 V 值最大的行為，因此 V 值總和反而越平滑，但我們也需同時避免落入局部最佳解，因此對本模型來說 Epsilon 最佳值為 0.2。



4. 回合數 (number of episodes)

在前面對學習率、Epilson 值的調整後，我們發現最佳參數組合為學習率為 0.05 與 Epilson 為 0.2，因調降學習率，此模型需增加運行的回合數以達到較良好的學習，因此我們將回合數設定為 1000 次，並以 V 值總和觀察模型的學習情形，可以發現到雖然模型在後期達到不錯的結果，但因本模型為尋找符合預算內的產品組合，可能有多種結果的選擇，因此曲線尚有崎嶇的現象。



四、模型延伸

1. 模型二 - 尋求最便宜的產品組合

上述模型尋求的結果可能為各種不同符合預算限制的產品組合，因此在實際情況下，我們可能需要尋求一個單一的最佳解，因此本專案延伸原先的模型以最便宜的產品組合作為目標，將此命名為模型二。若將預算設定為 23 元將會只有最便宜的產品組合可以符合，因此模型二中每回合的獎勵變成了預算減上每回合得出的商品組合的實際花費（如下圖虛擬碼所示），如此一來模型將會更快速的找出最便宜的產品組合。由下圖可以發現最便宜組合的 V 值總和（綠線）完全地超越了其他組合（橘線）。

```
if(budget >= episode2['Real_Cost'].sum()):
        Return = (budget - episode2['Real_Cost'].sum())
else:
        Return = (budget - episode2['Real_Cost'].sum())
```

2. 模型三 - 引入個人偏好設定

另外，消費者在現實情境中個人對品牌的偏好也往往影響到決策過程中，因此本專案進一步延伸原模型，加入個人偏好的元素，將其命名為模型三。在模型三中也同時考量了預算限制，並且為每個產品皆設立了 Reward 值（如下方虛擬碼所示），在此模型本專案將 a2 和 b2 的 Reward 值設為 0.8，可以發現索求出結果雖 a2 和 b2 皆不是該食材中最便宜的產品，但卻入選了最終產品組合。最後我們可以看到下圖其他組合（橘線）的 V 值總和在後段的回合中超越了最便宜組合（綠線），我們可以將其解讀為在符合預算內，最便宜產品組合並不是我們的個人偏好選擇。

```
if(budget >= episode2['Real_Cost'].sum()):
        Return = 1 + (episode2['Reward'].sum())/len(Ingredients)
else:
        Return = -1 + (episode2['Reward'].sum())/len(Ingredients)
```

五、結果與未來展望

1. 結論

本專案成功建置三個強化學習模型，分別為依預算推薦模型、最便宜產品組合模型、符合預算下的個人偏好模型來解決我們日常採買可能會遇到的問題。對於這些問題，我們往往會直觀地選擇以線性規劃來解決，但在本專案建構的過程中，發現以模型一、三來說使用強化學習相對線性規劃來說更為快速，得出結果也相當不錯，因此可能更適合使用快速或需要大量資料的問題中。

2. 未來展望

成功建置三個強化學習模型後，發現到這些模型在未來有非常大的潛力發展在其他產業間，以下為一些實際的範例。

- 零售業：蒐集顧客的消費歷史資料建立個人化的產品偏好，並以本模型即時給予消費者產品組合的推薦。
- 公司採購自動化：產業間採購部門往往花很多的時間在重複性高的採購訂單，本專案未來可結合 RPA 自動且智慧化選擇廠商的減少流程浪費。
- 金融商品組合：投資人在投資金融商品時常難以做出有效率的獲利性和風險性的評估，本專案模型可結合大數據應用和爬蟲應用，即時的為投資人依決策目標推薦資金配置組合。

強化學習模型可對於這些產業間的問題有更廣泛的應用，但在實際解決上本專案也期待可以加入其他更有效率的演算法來解決大量資料的問題。

六、參考資料

- 台大李宏毅老師線上教學

  https://www.youtube.com/watch?v=o_g9JUMw1Oc

- Coursera 吳恩達 - What is Machine Learning

  https://www.coursera.org/learn/ai-for-everyone

- Terasoft
  https://www.terasoft.com.tw/support/tech_articles/reinforcement_learning_a_brief_guide.asp