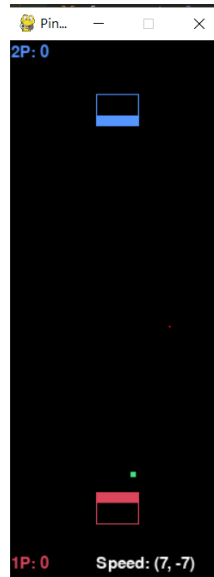


HW2

```
screen=(0,0,200,500)
```

```
'ball_speed': (7, -7),  
'platform_1P': (0, 420),  
'platform_2P': (0, 50),  
'blocker': (50, 240)
```



程式執行

```
python MLGame.py -i ml_play.py -i ml_play.py pingpong HARD
```

Rule base

1. 根據scenes_info提供的ball_speed去預測落點
2. 由於HARD模式會有移動式block擋在中間,必須根據scenes_info提供的block位置去預測當球進入到block的y範圍的時候是否會撞到block觸發反彈



如果忽略block計算當speed>15的時候會沒有辦法跟上預測的速度(只有在不
會打到block的部分預測的點是正確的,其他地方預測的都會打到block導致預測
錯誤)

3. 麻煩的點是由於球的速度會越來越快但是block保持不變,所以每一偵會越來越難計算



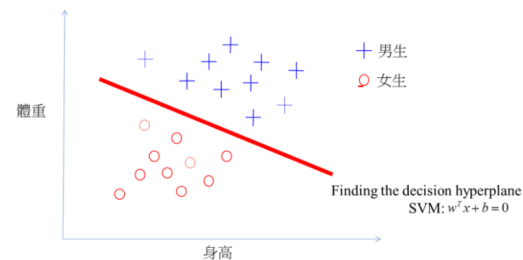
解決方法:把每一偵記錄起來在預測的時候適當加上速度就可以跟現實情況一樣了,再根據題目給的物理規則去預測反彈後的落點可以輕鬆通過speed>15

4 speed>20有點困難雖然有時候可以過但是都在17~23浮動,因為我超過block不進行預測因為沒辦法得知另一位player的block位置(不固定),所以就算反彈後可以成功預測落點但因為速度過快而導致板子來不及接到

Modal Predict

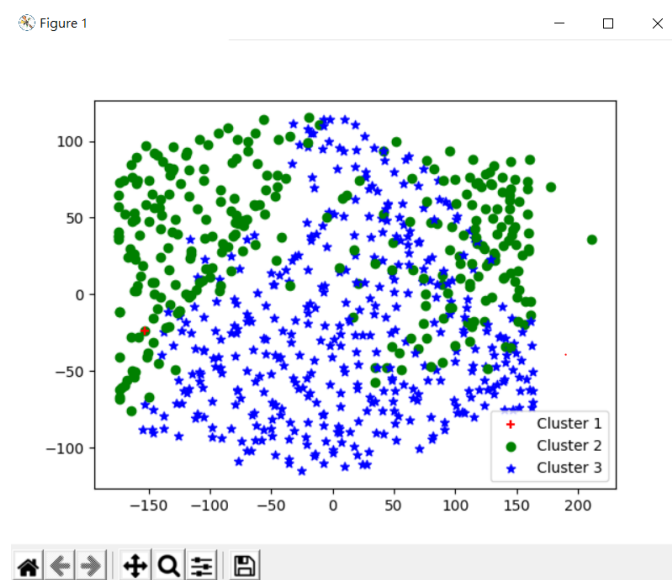
使用svm進行modal training

svm會將data很好的區分開來達到predict的目的



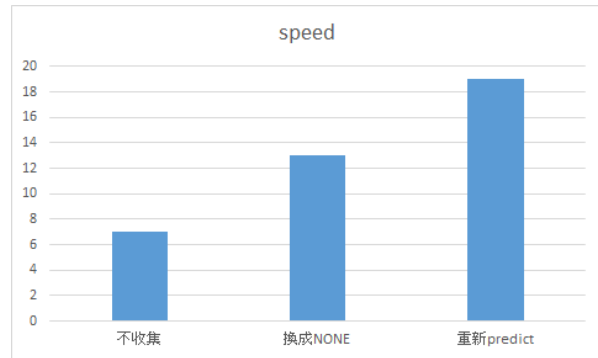
將data進行切塊區分

下圖為1個pickle檔案的範例,pickle太多程式畫圖跑不出來了



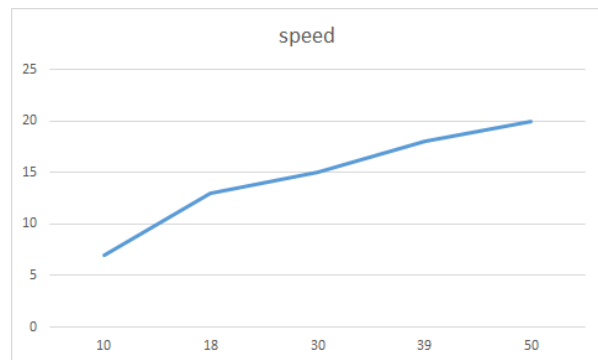
Problem

- 可能是data資料太少,加上scene_info紀錄速度太快,所以有很多資料command是"NONE"
- 這裡我在train.py再判斷一次斜率,修正那些錯誤的command
- 一開始是直接不採用錯誤的command,但修正command data結束時的速度越快



訓練data多寡

- 大約利用50個pickle檔進行訓練
- 由於block的起始位置是隨機的所以不用改變任何參數直接收集data
- 右圖可以看到訓練data越多最終速度會越快



Pickle特徵提取

- 如果只提取ball_y和platform_x predict score會偏低
- 加上ball_x可以大幅增加predict score
- 加上ball_speed以及block位置分數最高
- 提取參數為ball_x,ball_y,ball_speed_x,ball_speed_y,platform_x,block_x

