# HW3

```
Introdunction to AI @ NCTU, Spring 2018
0411276 陳奕安
測試環境:
    OS: Ubuntu Linux with cpython 3.6.5
    CPU: Intel i7-4790 (4C8T)
    RAM: 16GB.
```

## Achievements

**Basic**

- 根據給予的資料建立搜尋樹(CART)
- 實作 Tree bagging, Attribute Bagging
- 實作 Random Forests
- 利用 Cross validation 手法進行驗證

**Extra Effort**

- 提升驗證結果的穩定程度，以及確定驗證資料在總資料集中涵蓋的完整性
- 將訓練模型以及驗證過程使用平行化運算加速
- 給予一定的資料點範圍，程式能透過普查的方式找出最適合的訓練參數

## 實驗結果

驗證方法

所有的資料會被分成兩種，驗證用以及訓練用的資料，且實際驗證時的資料組是由驗證資料中隨機抽取特定筆數來進行驗證。必須先澄清，訓練模型用的資料跟驗證用的資料是完全不重疊的，以此方法比較能夠確定訓練出來的模型有學習到真正模型的情況。 再者，在每一輪的測試中，都盡量保證驗證資料必須包含到所有可能label的資料集，如此一來能減少不會造成驗證資料對母群代表性不佳的機率發生。

效能評估

| 資料集 | 最高平均模型正確率 |
|---|---|
| cross200 | 76.5% |
| iris | 99.68% |
| optical-digits | 83.64% |

### 資料集: cross200

- 資料筆數: 200

- 可能Label數量: 2
- 不同label資料量的比例是否平均: 是

獲得最佳參數範圍: + 分類樹資料量:5 + attribute bagging 子樹數量:3~7 + forests 子樹數量: 40~80


```
{"avg_rate": 66.32, "avg_td_rate": 73.0, "forest_size": 40, "bag_size": 5,
"validate_pool_size": 20, "model_training_size": 5}
```


使用不同資料進行準確性測試

| 驗證資料 | 訓練資料 |
| --- | --- |
| 66.32% | 73.0% |

## 資料集: iris

- 資料筆數:150
- 可能Label數量: 3
- 不同label資料量的比例是否平均:是

獲得最佳參數範圍: + 分類樹資料量: 25~100 + attribute bagging 子樹數量:4~16 + forests 子樹數量: 1~30


```
{"avg_rate": 92.95, "avg_td_rate": 95.08, "forest_size": 30, "bag_size":
10, "validate_pool_size": 30, "model_training_size": 100}
```


使用不同資料進行準確性測試

| 驗證資料 | 訓練資料 |
| --- | --- |
| 92.95% | 95.08 |

## 資料集: optical-digits

- 資料筆數: 3823
- 可能Label數量: 10
- 不同label資料量的比例是否平均: 是

獲得最佳參數範圍: + 分類樹資料量: 40 + attribute bagging 子樹數量: 3 + forests 子樹數量: 175


```
{"avg_rate": 84.96, "avg_td_rate": 86.57, "forest_size": 150, "bag_size":
3, "validate_pool_size": 100, "model_training_size": 40}
```


使用不同資料進行準確性測試

| 驗證資料 | 訓練資料 |
| --- | --- |

| 驗證資料 | 訓練資料 |
| --- | --- |
| 84.96% | 86.57% |

# 實作簡述

驗證資料

最一開始並沒有注意到驗證資料的涵蓋性問題，也就是因為簡單隨機抽樣的關係，導致驗證用的資料拿取到具有高度同類性，於是我對驗證用資料的拿取改採取部落抽樣，降低隨機性同時也提昇對母體的代表性。

資料選取

最一開始的實做方法是將資料穩定分成n份，其中一份拿來驗證，n-1份拿來進行不同模型的訓練。但是後來發現因為給予的資料量並沒有多到可以產生足夠信服的模型來投票產生平均結果，故思考了另外一種資料的取法。最終我將資料只分成訓練跟驗證用的兩類，兩者分別有數量T,V的資料數，每一次驗證都會用簡單隨機方法抽取數量v的驗證資料，而每筆隨機產生的驗證資料都會對應到n個產生出來的模型，每個模型的資料便是由T中隨機抽樣出t筆資料進行訓練，用這種方法就能夠從固定的訓練資料中產生出極為大量的決策樹組，解決資料量不夠的問題。

# 實驗心得

第一次寫有關機器學習類別的程式碼，感覺還蠻開心的。不過使用CART跟Random Forest實做分類器時能發現這種實做模型對應到調校參數的方法並不直觀，調整時並不具備太多專業知識能帶來的邏輯關係，雖然說正確率仍在可以接受的範圍內，但是這種方法所產生的智慧跟人類一直以來追求的通用性人工智慧仍詹不太上邊，更直接的說 - 缺乏推論與創造的元素。