台灣人工智慧學校

機器學習基礎與演算法

Chapter 7　集成學習 (Ensemble learning)
Chapter 8　Practical concerns

[講師投影片Chapter7]
[講師投影片Chapter8]
[課程投影片]
[資料與程式碼]
[播放清單]

「版權聲明頁」

本投影片已經獲得作者授權台灣人工智慧學校得以使用於教學用途，如需取得重製權以及公開傳輸權需要透過台灣人工智慧學校取得著作人同意；如果需要修改本投影片著作，則需要取得改作權；另外，如果有需要以光碟或紙本等實體的方式傳播，則需要取得人工智慧學校散佈權。

– 台灣人工智慧學校

課程內容

7. 集成學習 (Ensemble learning)

-Ensemble learning
-Bagging and random forest
-AdaBoost
-Gradient boosting
-Stacking
[實作] 隨機森林 (Random Forest)
[實作] 梯度提升機
(Gradient Boosting Machine)
[實作] XGBoost

8. Practical concerns

[實作] 調整參數

Code 放在Hub中的course內

- 為維護課程資料，courses中的檔案皆為read-only，如需修改請cp至自身環境中
- 打開terminal，輸入

cp -r courses-tpe/Machine_Learning <存放至本機的名稱>

Chapter 7　集成學習 (Ensemble learning)

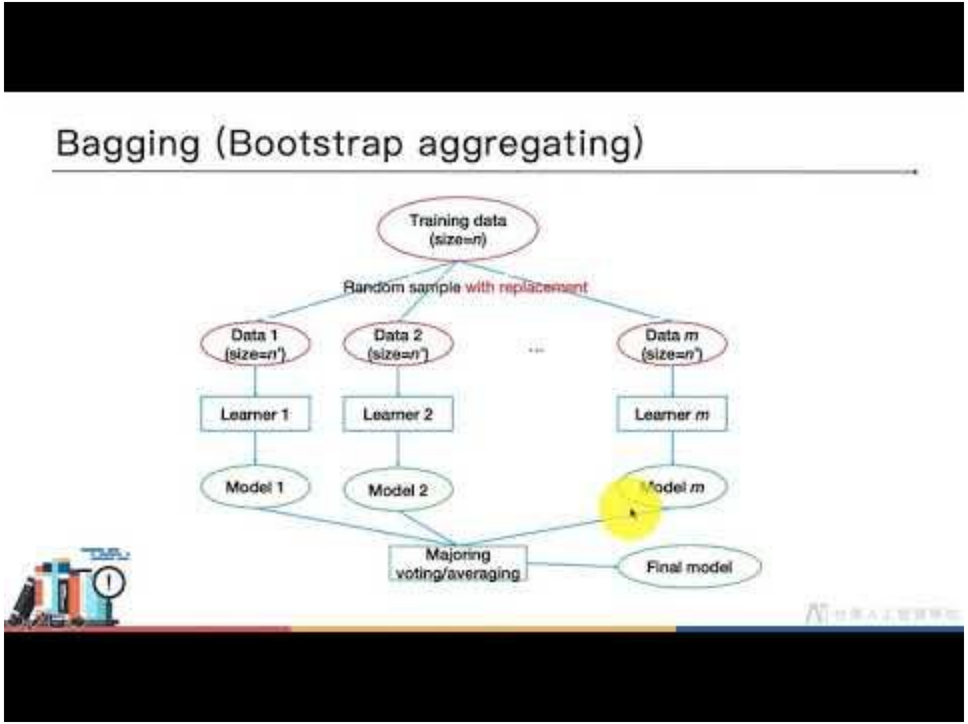
- 範例程式(example)的檔名會以**藍色字體**顯示且旁邊附上
- 練習(exercise)的檔案以**紅色字體**顯示且旁邊附上

07-1: Ensemble learning

Ensemble methods

- Bagging: resample training data
  - Random forest
- Boosting: reweight training data
  - AdaBoost
  - Gradient Boosting
- Stacking: blending weak learners

07-2: Bagging and random forest

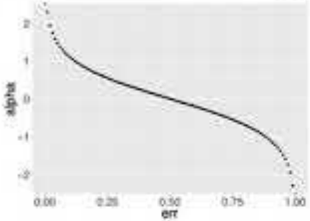


07-3: AdaBoost

AdaBoost – training

- Set the weight of f_k based on weighted error

$$\alpha_k = 0.5 \cdot \log\left(\frac{1 - err^{(k)}}{err^{(k)}}\right)$$



$$err^{(k)} = \sum_i w_i^{(k-1)} \cdot \xi_k(f_k(x_i), y_i)$$

- Set the weight of each instance based on ensemble prediction

$$w_i^{(k)} = \frac{w_i^{(k-1)} \exp(-\alpha_k y_i \hat{y}_i)}{z^{(k)}}$$

$z^{(k)}$ is the normalization term such that $\sum_i w_i^{(k)}$ sums to one

07-4: Gradient boosting

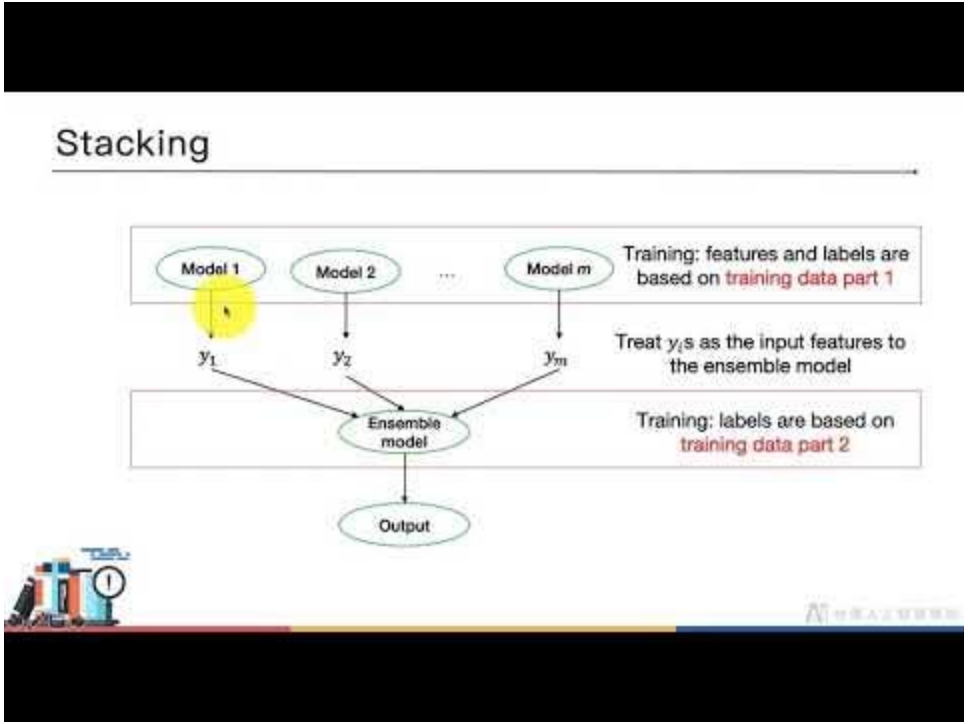
How is this related to gradient descent? (1/2)

- Loss function: $J = \frac{1}{2}\sum_i (y_i - F(x_i))^2$
- Gradient of J to $F(\boldsymbol{x}_j)$
  - $\frac{\partial J}{\partial F(\boldsymbol{x}_j)} = (y_j - F(\boldsymbol{x}_j))(-1) = F(\boldsymbol{x}_j) - y_j$

- Algorithm
  - Initially, $F(x_j) = f_1(x_j)$
  - Update by gd until termination condition is met:

$$F^{(k+1)}(x_j) = F^{(k)}(x_j) - \frac{\partial J}{\partial F(\boldsymbol{x}_j)}$$
$$= F^{(k)}(\boldsymbol{x}_j) + \left(y_j - F^{(k)}(\boldsymbol{x}_j)\right)$$
$$= f_1(\boldsymbol{x}_j) + \cdots + f_k(\boldsymbol{x}_j) + f_{k+1}(\boldsymbol{x}_j)$$

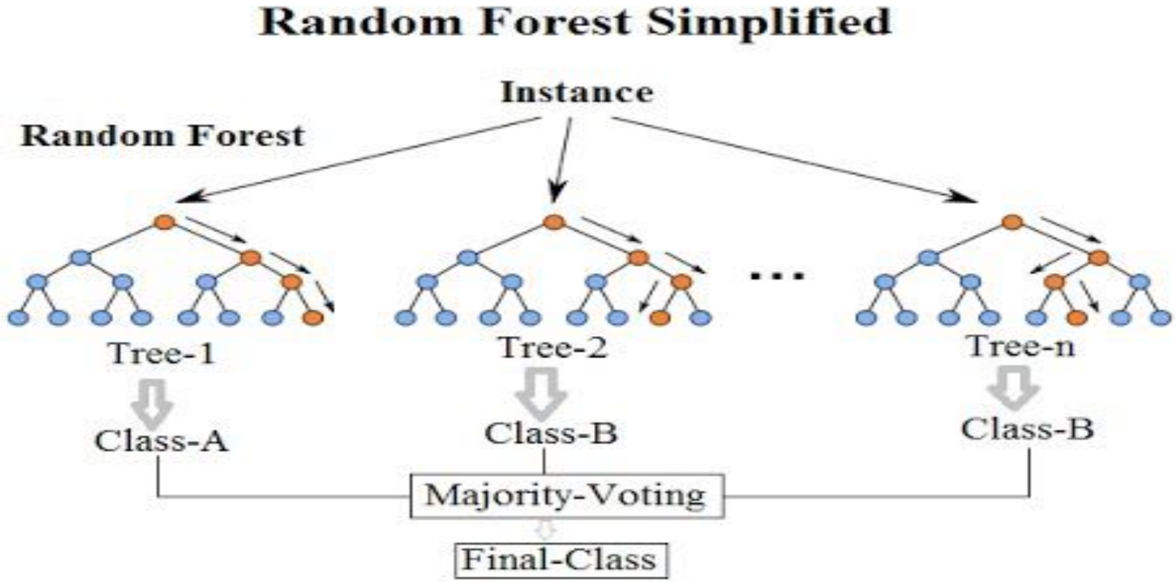
07-5: Stacking



[實作課程] 隨機森林 (Random Forest)

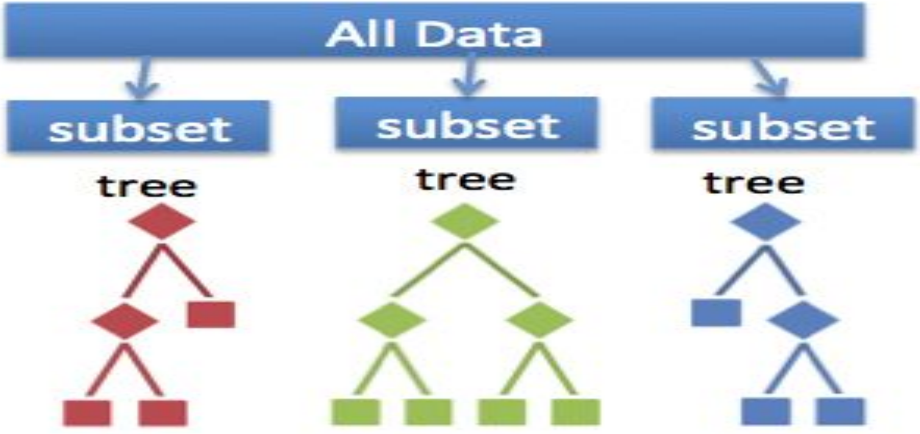
隨機森林 (Random Forest, RF)

- 決策樹非常容易 Overfitting (why?)
- 那如果多種幾棵樹，把樹變成森林會怎麼樣?...



Why Random?

- 每一棵樹在生成過程中，都可能用到不同訓練資料及不同的 features
- 會用到哪些訓練資料及 features 則是隨機 (random) 決定!



決策樹系列：隨機森林 (Random Forest) (續)

Why better than Decision Tree?

- Random forest 使用了我們稱作「Ensemble」的方式。從model 的 import 就能看出

```
from sklearn.ensemble import RandomForestClassifier
```

- 因為每棵樹可能是由不同樣本、不同 features 所生成，當使用集成方法，可將所有樹的結果做平均，緩解決策樹 Overfitting 的情形，使得預測結果更為穩定。

隨機森林 in Scikit-learn

```python
from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import RandomForestRegressor

clf = RandomForestRegressor()
```

隨機森林中的常見參數

```
from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier(
        n_estimators=100, #number of trees
        criterion="gini",
        max_features="auto", #sqrt(features)
        max_depth=10,
        min_samples_split=2,
        min_samples_leaf=1
)
```

練習 random_forest_exercises.ipynb

- 請使用 Radom forest 來執行 Iris dataset，比較 Random forest 的模型結果是否比 Decision tree 來得好
- 請使用 digits dataset，並比較如果樹的數量多寡 (n_estimators)，對結果是否會有改善？

Write a Random forest from Scratch (optional)

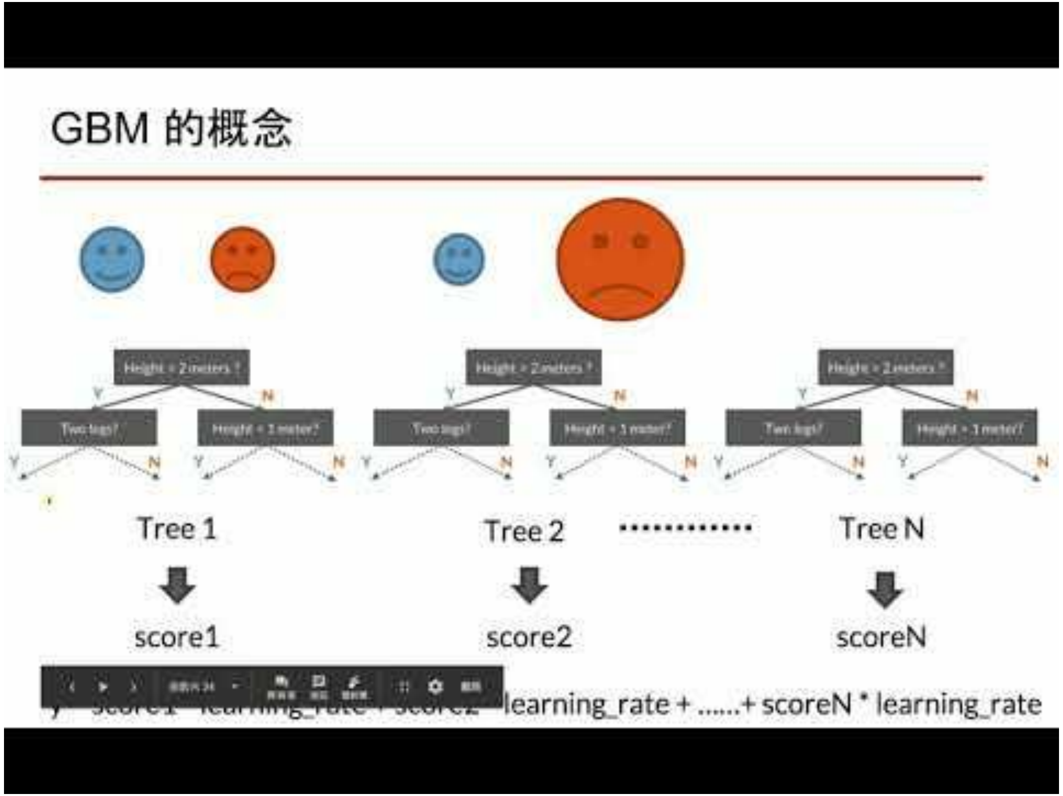
補充閱讀

- 如果前面助教講的影片你都聽不懂，肯定是因為助教講的不夠清楚，只好幫各位找一些寫的不錯的文章，給大家參考
  - 隨機森林 (random forest) - 中文
  - how random forest works - 英文

思考問題

- Random Forest 中的每一棵樹，是希望能夠
  1. 盡量的生長 (讓樹生成很深，比較複雜)
  2. 不要過度生長，避免 Overfitting ?
- 假設資料總共有 N 筆 samples (N is large)，每棵樹用**取後放回**的方式抽了總共 N 筆資料來生成一棵樹，請問這棵樹大約使用了多少 % 的 unique 原資料生成 (不重複)?
  - hint: google 0.632 bootstrap

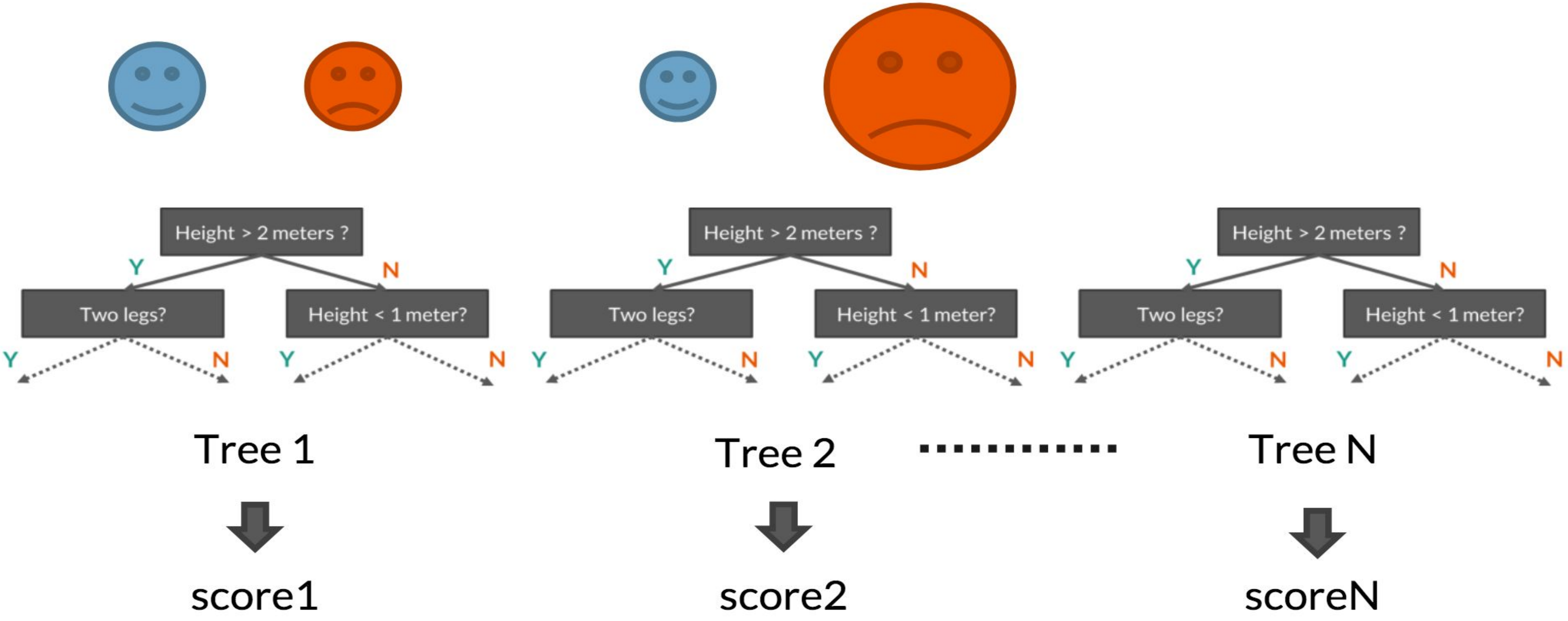
[實作課程] 梯度提升機 (Gradient Boosting Machine)



Boosting? Gradient?

- 前面我們學到的方法稱為 Bagging (Bootstrap aggregating), 用抽樣的資料與 features 生成每一棵樹, 最後再取平均
- Boosting 則是希望能夠由後面生成的樹, 來修正前面樹學的不好的地方
- 要怎麼修正前面學錯的地方呢？計算 Gradient! (先想像 Gradient 就是一個能教我們修正錯誤的東西）

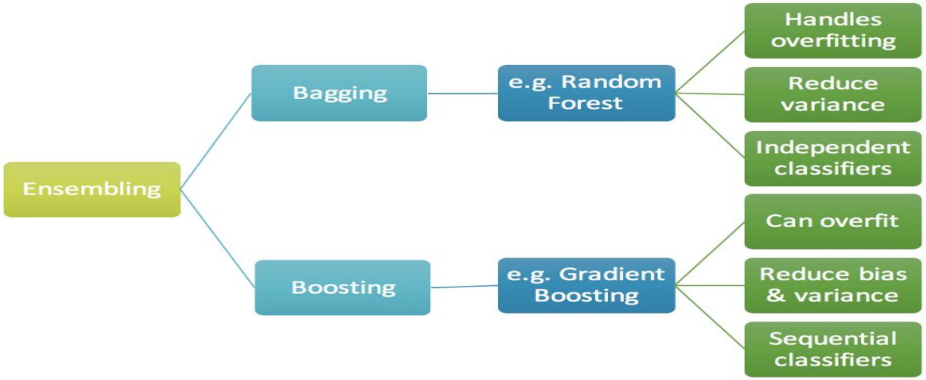
Adaboost 的概念



y = score1 * learning_rate + score2 * learning_rate + ……+ scoreN * learning_rate

Bagging vs. Boosting

- Bagging: 透過抽樣的方式生成樹，每棵樹彼此獨立
- Boosting: 透過序列 (additive) 的方式生成樹，後面生成的樹會與前面的樹有關
- 一般來說，Boosting 的模型會比 Bagging 來的準確



[實作課程] Kaggle 大師帶你理解 Gradient boosting (連結)

[實作課程] 梯度提升機 (Gradient Boosting Machine) (續)

GBM 常見參數設定

```
from sklearn.ensemble import GradientBoostingClassifier

clf = GradientBoostingClassifier(
      n_estimators=100, #number of trees
      learning_rate=0.1, # shrinkage of prediction
      max_features="None",
      max_depth=3
)
```

GBM in Scikit-learn

```python
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import GradientBoostingRegressor
```

```python
clf = GradientBoostingClassifier()
```

GBM 常見參數設定

```
from sklearn.ensemble import GradientBoostingClassifier

clf = GradientBoostingClassifier(
      n_estimators=100, #number of trees
      learning_rate=0.1, # shrinkage of prediction
      max_features=”None”,
      max_depth=3
)
```

練習

- 請改用 **Gradient boosting (gradient_boosting_example .ipynb) 的模型來執行 Iris / digits dataset**，並試著增加樹的數量 (n_estimators), 比較是否會影響結果
- 如果單純增加樹的數量, 沒有對 learning_rate 做調整, 是否會影響結果?

這麼難的模型還是想要手刻?! (optional)

- 沒問題! 單純用 Python 實現 Gradient Boosting Machine

補充閱讀

- 如果前面助教講的影片你都聽不懂，肯定是因為助教講的不夠清楚，只好幫各位找一些寫的不錯的文章，給大家參考
  - GBM 簡介 - 中文
  - intro to gradient boosting - 英文

[實作課程] XGBoost

一段 Kaggle 冠軍的訪談

Interview from Kaggle winner (What have you taken away from this competition?)

- With a good computer, R can process “big data” too
- Always write data processing code with scalability in mind
- **When in doubt, use XGBoost**

What's XGBoost? (1/2)

- 全名為 eXtreme Gradient Boosting
- XGBoost is an implementation of gradient boosting machine but add some features

XGBoost

What's XGBoost? (2/2)

- Additive model (與 GBM 類似）
- Features sampling (與 Random forest 類似)
- Add regularization in objective function
- Use 1st and 2nd derivative to help training

XGBoost vs. GBM

- 阿里巴巴的面試題目: 請問 XGBoost 與 GBM (Gradient boosting machine) 有什麼差異？
  - objective function 加上 regularization，避免 Overfitting
  - 用上一階及二階導數來生成下一棵樹
  - feature / data sampling。與 RF 相同，每棵樹生長時用到不同的資料與 features

XGBoost 安裝

- XGBoost 是由華盛頓大學博士班學生陳天奇所開發，是目前 Kaggle 比賽中最常見到的算法！
- Hub 環境上已經幫各位安裝完成

```
from xgb import XGBClassifier, XGBRegressor
```

- 若希望在自己的本機上安裝，請參考
  - Windows: [install XGBoost on windows]
  - Mac / linux: `pip install XGBoost`

XGBoost model

```
from xgb import XGBClassifier, XGBRegressor

clf = XGBClassifier()

clf.fit()

...
```

XGBoost 常見參數設定

- XGBoost 需設定的參數大概是目前我們學習到所有模型中最多的
- 要學會如何設定參數，需要先瞭解參數的意義
    - booster [default=gbtree]: (gbtree, gblinear)

XGBoost 常見參數設定 - 樹參數設定

n_estimators [100]: number of trees

learning_rate [0.1]: shrinkage

max_depth [3]: too large → overfitting

gamma [0]: L2 loss regularization, too small → overfitting

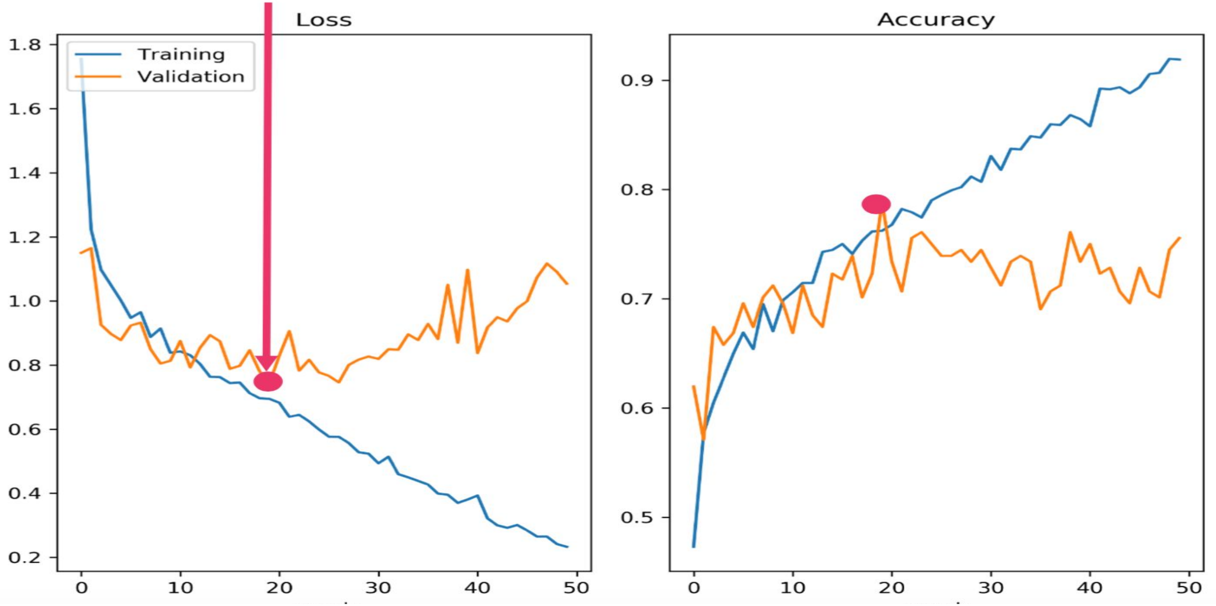
lambda [0]: L1 loss regularization, too small → overfitting

scale_pos_weight [1]: use for imbalance data

*方括 [ ] 內為該參數預設值

Early stop in XGBoost (1/2)

- XGBoost model 非常強大，但也容易 Overfitting，Early stop 幫助我們在 Overfitting 前提早停下來



Early stop in XGBoost (2/2)

```
# eval_metrics = rmse, logloss, error, auc, merror, mlogloss, custom
eval_set =  [(X_test, y_test)]
model = XGBClassifier()
model.fit(X_train, y_train, early_stopping_rounds=10, eval_metric="auc",
          eval_set=eval_set, verbose=True)
```

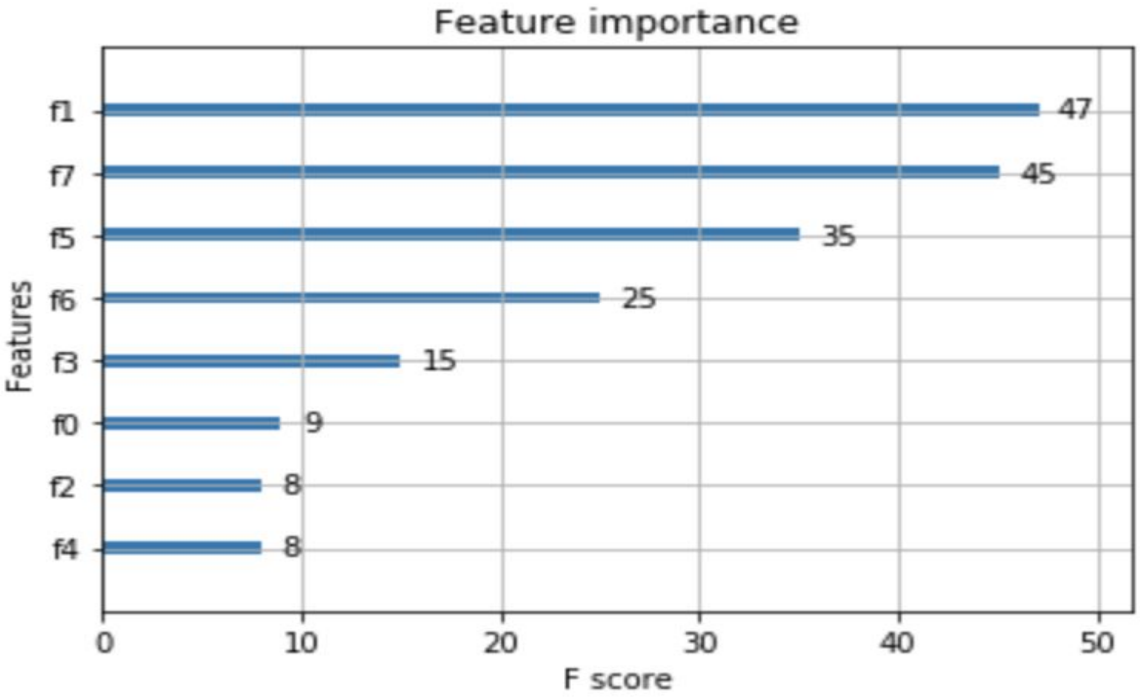
```
[0]	validation_0-auc:0.817834
Will train until validation_0-auc hasn't improved in 10 rounds.
```

- 將 testing data 放進 eval_set，如果 validation 的結果 10 次沒有進步，就提前結束 training
- 也可以改放 training data，觀看 training loss 下降的感覺 (文字一樣會顯示 validation_0)

Feature Importance in XGBoost

XGBoost 內建功能

```
from xgboost import plot_importance
plot_importance(model)
plt.show()
```



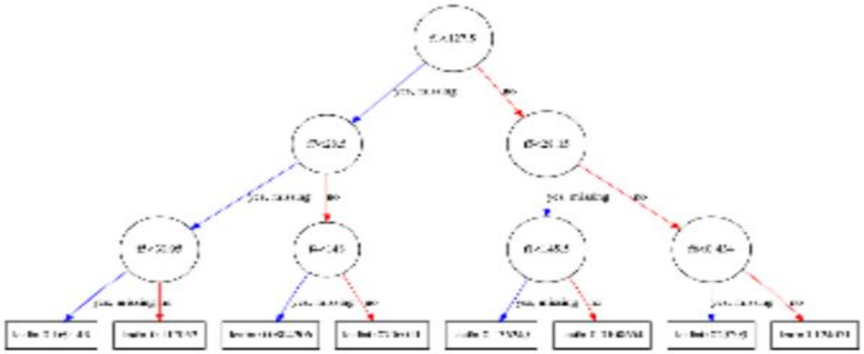
XGBoost 視覺化

- 若執行這段 code 有 error，代表環境還沒有安裝好 graphviz，請重開 Server

```
In [31]: from xgboost import plot_tree
         from matplotlib.pylab import rcParams

         plot_tree(model, num_trees=1)
         # plt.title("max_depth = 100, with gamma = 10")
         # plt.savefig("tree_with_max_depth_gamma", dpi = 700)

Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x7f416570b6a0>
```



練習

- 請使用 example 中的 diabetes dataset，**使用 XGBoost 進行訓練**，試著更改如 n_estimators、max_depth 甚至是 scale_pos_weight (平衡 imbalance data，如果 `類別 0 數量 : 類別 1 數量 = 5 : 1`，則可設置 5)
- 與 Decision Tree, Random Forest, Gradient Boosting Machine 進行比較，XGBoost 真的有比較厲害? (記得使用同一份 testing set)
- 不設定 early stop，把 n_estimators 調高 (500~1000)，就可以體驗看看甚麼叫做 Overfitting

XGBoost 作者講解並推導原理

補充閱讀

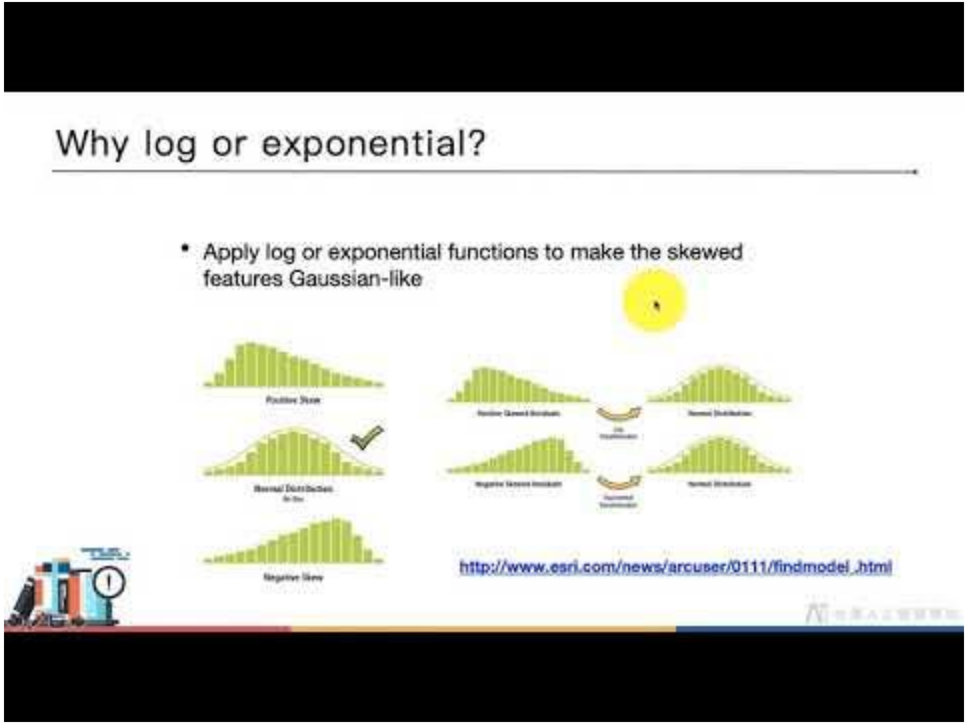
- 如果前面助教講的影片你都聽不懂，肯定是因為助教講的不夠清楚，只好幫各位找一些寫的不錯的文章，給大家參考
  - XGBoost 詳解 - 中文
  - XGBoost parameter tuning - 英文
  - slides from XGBoost author - 英文

思考問題

- 同樣的 dataset 若存在兩個完全一模一樣的 feature (feature1, feature2)，這兩個 feature 的 importance，在 XGBoost 與 Random Forest 的模型結果中，會一樣嗎?
- XGBoost 中，row_sample 代表對資料筆數抽樣 (row)，col_sample 代表對 features 抽樣，若這兩個都設置成 1 (代表不抽樣，全部使用)，每次訓練後的樹會長的一模一樣嗎?

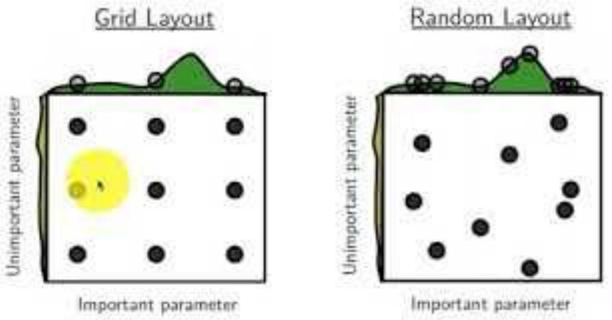
Chapter 8　參數選取 (Parameter Selection)

08-1: Data preprocessing



08-2: Selecting hyper-parameters

Random search is surprisingly good



Bergstra and Bengio. "Random search for hyper-parameter optimization." JMLR 2012

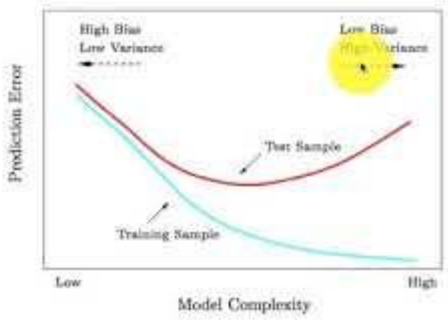
08-3: Multi-class classification

One–vs.–one

- Training: for a *k*-nary classification problem, one trains *C*(*k*, 2) classifiers
- Test:
  - Feed the test instance to all *C*(*k*, 2) classifiers
  - The class receiving the most "+1" predictions is the predicted class

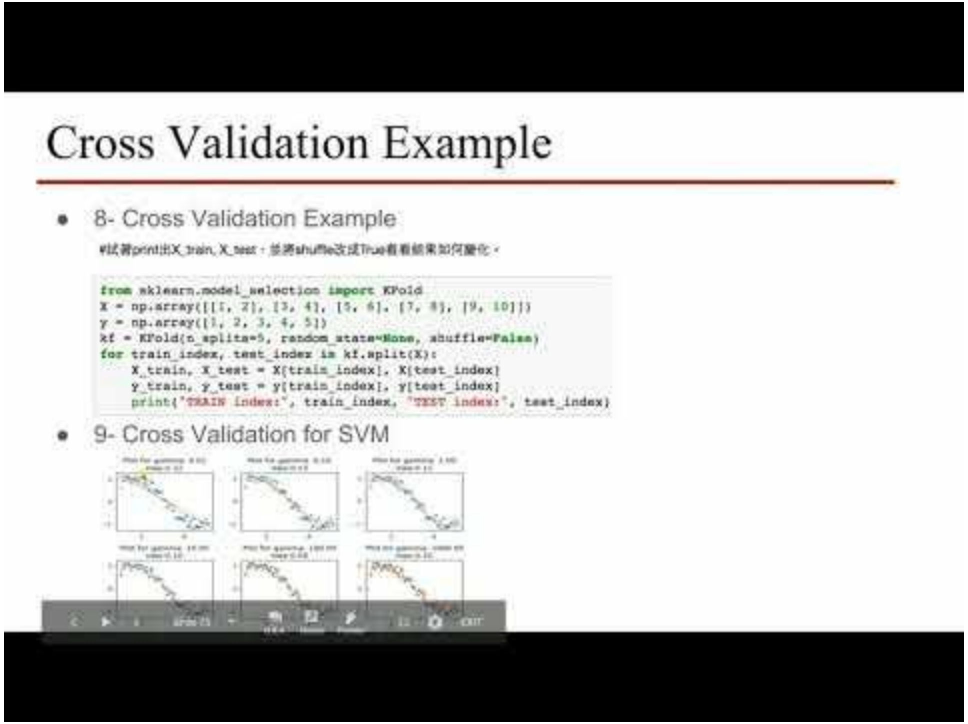
08-4: Model selection

Model complexity vs error



Hastie et al., "The Elements of Statistical Learning," 2001.

[實作課程] 調整參數



Cross Validation Example
8- Cross Validation Example
#試著print出X_train, X_test，並將shuffle改成True看看結果如何變化。
from sklearn.model_selection import KFold
X = np.array([[1, 2], [3, 4], [5, 6], [7, 8], [9, 10]])
y = np.array([1, 2, 3, 4, 5])
kf = KFold(n_splits=5, random_state=None, shuffle=False)
for train_index, test_index in kf.split(X):
X_train, X_test = X[train_index], X[test_index]
y_train, y_test = y[train_index], y[test_index]
print("TRAIN index:", train_index, "TEST index:", test_index)
9- Cross Validation for SVM

選擇模型參數 - 評估模型好壞

- k-fold Cross Validation：將其中一份validation set從training data抽出，剩下資料拿來做訓練，如此重複k次，並將結果平均，來得到更robust的評估結果。

EX: 5- fold CV

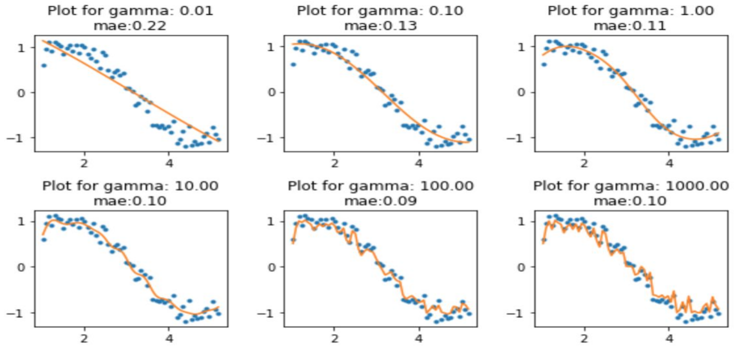
Cross Validation Example

- **5- Cross Validation Example**

#試著print出X_train, X_test，並將shuffle改成True看看結果如何變化。

```
from sklearn.model_selection import KFold
X = np.array([[1, 2], [3, 4], [5, 6], [7, 8], [9, 10]])
y = np.array([1, 2, 3, 4, 5])
kf = KFold(n_splits=5, random_state=None, shuffle=False)
for train_index, test_index in kf.split(X):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]
    print("TRAIN index:", train_index, "TEST index:", test_index)
```

- **6- Cross Validation for SVM**



[實作課程] Grid Search Cross Validation

選擇模型參數 -Grid Search CV

10- Cross Validation Example---iris：比較選擇參數前後的準確度及output最佳參數。

Grid Search for SVM Parameters

```
from sklearn.model_selection import GridSearchCV
parameters= {'kernel':['linear', 'rbf'], 'C':[0.01,0.1,1,10], 'gamma':[0.01,0.1,1,10]}
model = svm.SVC()
best_model = GridSearchCV(model, parameters, cv=5, scoring='accuracy')
best_model.fit(X, y)
```

- GridSearchCV input:
  - 欲運行的模型
  - 欲scan的參數並將之存成dict的形式
  - cv為做cross validation的folds
  - scoring可指定sklearn內建的eval metrics

C

Gamma

Gamma/C	0.01	0.1	1	10
0.01				
0.1				
1				
10				

http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

選擇模型參數 -Grid Search CV

7- Cross Validation Example---iris：比較選擇參數前後的準確度及 output最佳參數。

Grid Search for SVM Parameters

```python
from sklearn.model_selection import GridSearchCV
parameters= {'kernel':['linear', 'rbf'], 'C':[0.01,0.1,1,10], 'gamma':[0.01,0.1,1,10]}
model = svm.SVC()
best_model = GridSearchCV(model, parameters, cv=5, scoring='accuracy')
best_model.fit(X, y)
```

- GridSearchCV input:
  - 欲運行的模型
  - 欲scan的參數並將之存成dict的形式
  - cv為做cross validation的folds
  - scoring可指定sklearn內建的[eval metrics]()

C

Gamma

Gamma/C	0.01	0.1	1	10
0.01				
0.1				
1				
10				

http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Machine Learning Practice

- Wine Quality
  - Task1: Classify red/ white wine（目標:Accuracy）
  - Task2: Predict wine quality（目標:F1_Score）

- German Credit Data（目標:F1_Score）
  - Datasets

Note:
- 點連結即可進到題目 說明網址
- 第一題datasets已經先幫大家整理好放在資料夾內，第二題請自行從Datasets連結導入。

Hint: 試著畫圖了解資料

from UCI Machine Learning Repository