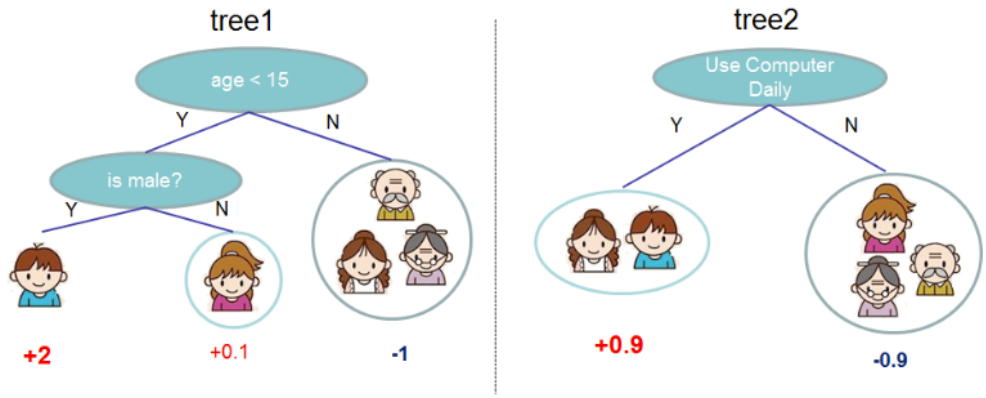
XGboost (eXtreme Gradient Boosting) 極限梯度提升

一、簡介

XGBoost 即是數顆「分類與回歸樹(Classification and Regression Tree；CART)」組合成一個高準確率的模型。每加入一次新的函數至原模型中，修正上一顆樹的錯誤，不斷的迭代以提升整體的模型。如圖一所示，Tree1 根據年紀與性別分類，藍衣小男孩所得的葉子分數為+2；Tree2 為每日使用電腦習慣，小男孩所得的葉子分數為+0.9，因此最後的模型結果，小男孩總共獲得 2.9 分，反之，阿公的分數為-1.9，而這個累加結果則會作為預測結果。除此之外，模型訓練過程是隨機抽取特徵，每棵樹的生成僅與前棵樹部分相關，也就是 Boosting，此方法比 Bagging（每棵樹互相獨立）計算的模型更加精準。另外，XGB 的主要運用場景為回歸與分類，但對於時間序列的數據則不擅長。



圖一、數集成模型（Chen and Guestrin, 2016）。

二、比較

以下 XGB 與 GBDT（Gradient Boosting Decistion Tree；梯度提升決策樹）的比較。

	XGB	GBDT
樹的種類	回歸樹與決策樹	回歸樹
運算速度	較快，GPU 平行運算	較慢，模型是串接生成

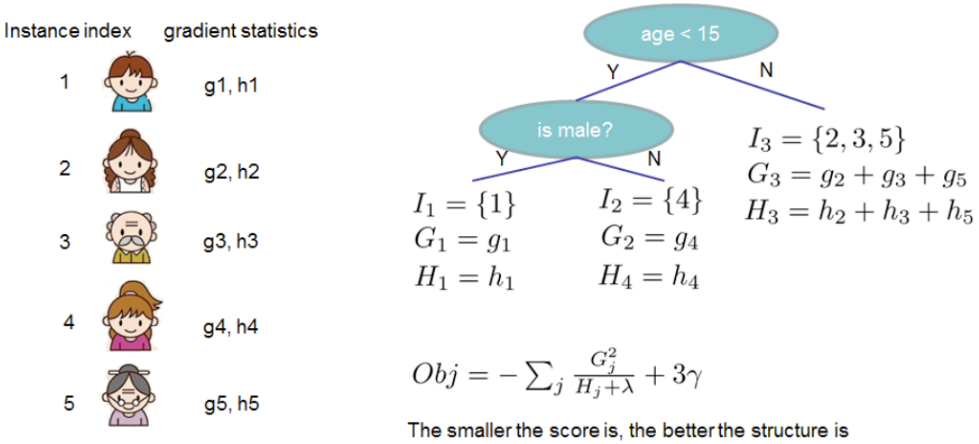
Loss function	二階導數	一階導數
正則項	有，使 loss function 更加平滑	無
缺失值	僅先利用有值的數據尋找特徵建立模型，再嘗試將缺失值劃入對應分支，選擇損失最優的值作為分裂點	會先手動對缺失值填充（沒有任何依據的狀況），當作有值的特徵處理
抽樣	借用了隨機森林的欄抽樣（每一次分裂只使用被抽取的特徵）	

三、模型的劃分方法

下述為 XGB 模型中的損失函數、樹的劃分方法。每一次的迭代會藉由計算下式的損失函數（Loss Function）來得知真實值及預測值之間的差異，以獲得最佳模型（圖二）：

$$-\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda}+\gamma T$$

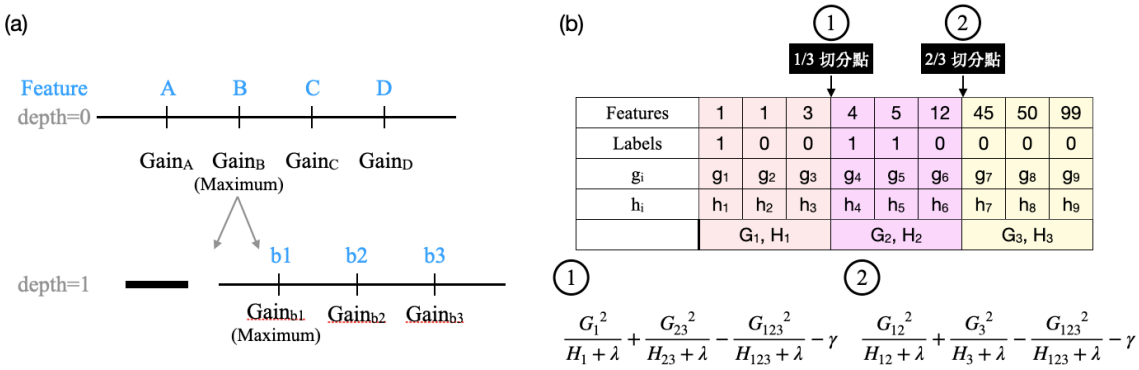
其方程式用於判斷模型的優劣，H和γ則分別與後續參數 min_child_weight、gamma 有關。



圖二、樹結構計算範例（Chen and Guestrin, 2016）。

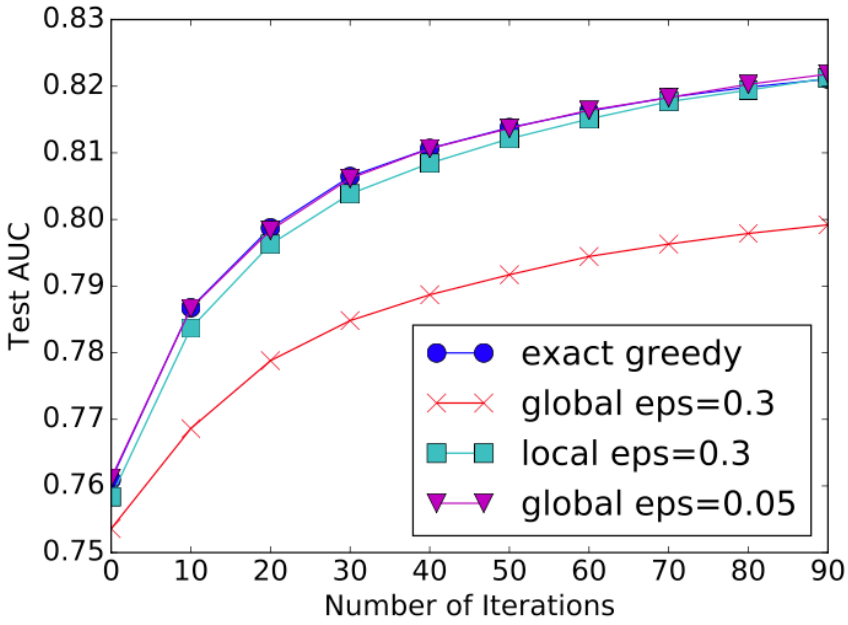
樹的劃分方法則與數據量的多寡有關—— Exact Greedy Algorithm 與 Approximate Algorithm，前者較適用於數據量小的狀況，如圖三 a，該算法會在該深度將「所有的特徵列出且劃分」，計算「劃分後的左葉與右葉分數和」及「劃分前的分數」差（式一），即為 Gain，當 **Gain 越大時代表 Loss function 下降的越多，進而找出最好的劃分點**，然後再往下個深度，再將所有特徵列出計算 Gain，來建立最佳的模型；後者則適用於數據量大的狀況，先根據 k 個特徵選擇分位點（Quartile），將這些分位點視為要觀察的劃分點，只要考慮這些劃分點的結果代入式一，即可決定分位點應選擇哪一個（圖三 b）。

$$\frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda}+\frac{G_R^2}{H_R+\lambda}-\frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right]-\gamma \qquad (1)$$



圖三、兩種劃分方法建立分位點方式。（a）為 Exact Greedy Algorithm（b）為 Approximate Algorithm。

而圖三 b 的分點選取時間點，可分為（1）「學習每棵樹前」，提出候選的分位點（在此稱之 Global），與（2）「每次分裂前」，重新提出候選的分位點（在此稱之 Local）。直覺上來說，Local 需要較多計算步驟、Global 需要提出較多的候選分位點。從圖三來看，global 的分位點夠多（1/0.05=200）時，與 Exact Greedy Algorithm 的效能一致，同時，local 的分位點少量（1/0.3=33.3）即可達到一樣的效果。



圖四、以 Approximate Algorithm 建立樹模型的分位點方式（Chen and Guestrin, 2016）。eps 的倒數為分位點數量。

三、XGB 參數設定（詳見 *XGBoost_iris.ipynb*）

1. `n_estimators`: 樹的數量。[default= 10]
2. `booster`: 模型種類。[default= gbtree]
   - `gbtree`: tree-based models
   - `gblinear`: linear models
3. `max_depth`: 每棵樹的最大深度，越大越能學到更局部的樣本，但也有可能 over fitting。[default= 6]
4. `objective`: 定義最小化 loss function 類型（詳見官方手冊）。[default= reg:linear]
   - `reg:squarederror`: squared loss 的迴歸
   - `binary:logistic`: 邏輯斯回歸，return 值為預測機率
   - `multi:softmax`: 邏輯斯回歸的延伸，每個 class 都有機率分佈，return 值為 class

◦ multi:softprob: 與 softmax 相同，但 return 的是每個資料點在每個 class 的機率

5. eta/learning_rate: 學習速率，完成一次迭代後會把葉子權重乘上此係數，削弱前棵樹的影響，讓後面的樹有更大的空間可學習/修正。[default= 0.3]
6. tree_method: 樹的劃分方法。[default= “auto”]
   ◦ auto：對於小數據集--> “exact”; 對於大數據集-->”approx”
   ◦ exact：Exact Greedy Algorithm
   ◦ approx：Approximate Algorithm
   ◦ hist：更快的直方圖優化法（based on Exact Greedy Algorithm ）
   ◦ gpu_hist：加入 GPU 的 hist 計算
7. min_child_weight: 葉子的最小樣本數，與式一的H_j（二階導數項的和）有關。假設 h 在 0.01 附近，min_child_weight 為 1，代表葉子最少需要 100 個樣本。此值愈小愈容易 overfitting。[default= 1]
8. gamma: 即「γ」，越大越保守。[default= 0]
9. lambda: L2 正則項參數，用於控制模型複雜度，使 loss function 更加平滑，參數越大越不容易 overfitting。[default= 1]
10. alpha: L1 正則項參數，用於控制模型複雜度，使 loss function 更加平滑，參數越大越不容易 overfitting。[default= 0]
11. max_leaves: 不與 tree_method=”exact”同用。[default= 0]
12. colsample_bytree: 生成樹時所進行的 column 採樣，通常設置為 0.5-1。[default= 1]
13. seed: 隨機種子，用於產生可重現的結果。[default= 1000]
14. max_delta_step: 取正值使迭代步驟更加保守，若該值為 0 則沒有約束，但當使用 logtstics regression 時調整此參數可能有效果。[default= 0]
15. subsample: 若該值為 0.6，每次迭代隨機抽取六成的訓練數據，以防止 overfitting。[default= 0]

16. scale_pos_weight: 大於 0（設定值）的值可以處理類別不平衡狀況，幫助模型更快收斂。[default=0]
17. eval_metric: 驗證數據所需要的衡量方法（詳見官方手冊）。[default= according to objective]
    - rms: root mean square error (for regression)
    - auc: area under the curve for ranking evaluation
    - error: binary classification error rate (for classification)

四、參考文章與網址

Chen, T., & Guestrin, C. (2016). XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.2939785

https://blog.csdn.net/sb19931201/article/details/52557382

https://medium.com/chung-yi/xgboost%E4%BB%8B%E7%B4%B9-b31f7ec8295e

https://www.hrwhisper.me/machine-learning-xgboost/

https://zhuanlan.zhihu.com/p/145041410

https://www.gushiciku.cn/pl/p8al/zh-tw

https://ithelp.ithome.com.tw/articles/10268984

https://ithelp.ithome.com.tw/articles/10301273