歌聲轉譜競賽 - Report

隊名: **為什麼不能叫哭哭**

方法

Preprocess - zcr

- 想法：由於zcr與換氣時機具有一定相關性，必定也會與音符出現時機有關
- 觀察：將原始音檔與ground truth比對後，發現當zcr出現大幅下降時，多半為某個音符的開始
- 目標：希望能正確地判斷音符的起始位置
- 實作：定義high threshold & low threshold，若前一個音符的zcr > high threshold 且這一個音符的zcr < low threshold 則將資料分段

```
if data(i-1) > threshold_h && data(i) <= threshold_l)
        sub(num_sub, 2) = i - 1; %subset的結束
        sub(num_sub + 1, 1) = i; %subset的開始
        num_sub = num_sub + 1;
end
```

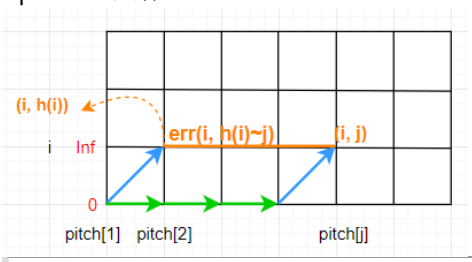
- 調整：多次嘗試後，high threshold = 0.2, low threshold = 0.1 效果最佳
- 補足：由於滿足zero crossing rate threshold條件的資料點較少，需要做進一步的資料分割

Preprocess - pitch

- 用zero crossing rate進行大略分段後，再透過分析音高的分布做更細的切割
- 若遇到pitch = 0的資料點，則將資料分段
- 理想狀況下，data經此function處理後只剩下有唱歌的部分

dp (Dynamic Programming)

- 把 preprocess 分完的段落，一段一段進行 dp
- dp table 實作



dp table: Dis[N][N]，column 代表第 j 個 pitch
h(i): 第 i 段的開頭，其中 Dis(i, h(i)) = Dis(i-1, h(i)-1)
err(i, h(i)~j): 找到第 i 段，從第 h(i) 到第 j 個 pitch 的中位數，計算誤差的總和

```
Dis[1][1] = 0
Dis[j+1][j] = Inf /*j 個 pitch 不可能分成 j+1 段*/
Dis[i][j] = min(Dis[i-1][j-1], Dis[i, h(i)] + err(i, h(i)-j)) /*j 大於 1*/
```

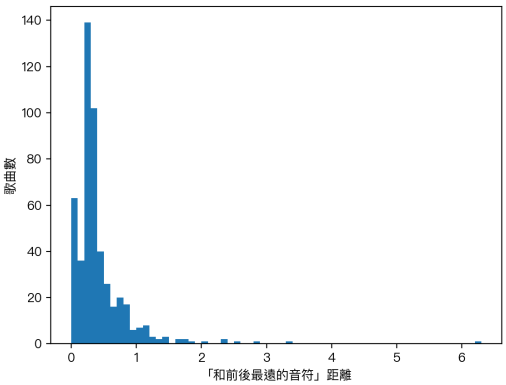
- 從最後面的 node 往前找 distance 最小的路徑
  - 因為將 pitch 一個一個切段，一定會使的誤差最小

    - 因此在找最小路徑時，會設定一個誤差門檻，當誤差沒有超過此門檻，就不必切段 (但最後此門檻接近0時表現最佳)
- 最後切完段落之後，取每一個段落的中位數作為此段對應的 note

dp 之後的調整

1. 當某一段的 pitch 數量少於設定門檻時，就將與他和相鄰兩段中，對應的 note 較接近的那一段合併
2. 當某一段和他的相鄰段對應的 note 是一樣的，且時間差(前一段的結束與後一段的開始)小於設定門檻，就將他們合併

最後調整

- 觀察：
    - 比對了ground truth後我們發現，主辦方給的feature.json裡面有時會出現「人明明沒有唱歌，卻有vocal pitch」的情形，推測是抽取人聲時抽不乾淨導致的。
    - 進一步來看，這樣的狀況大多發生在前奏、間奏、尾奏：
- 目標：希望能去除因為原始feature錯誤而選出的錯誤音符
- 作法：
    - 基本假設：歌詞是連續的一句話，因此音符之間不會間隔太遠
    - 以「該音符和前後音符的距離之最小值$\min(t_{i,start} - t_{i-1,end}\ ,\ t_{i+1,start} - t_{i,end})$」作為指標，可以選出「和前後距離太遠」的音符，也就是前奏、間奏、尾奏中錯誤抽出vocal pitch的音符
    - 先跑一遍ground truth，找出500首歌中，每首歌「和前後距離最遠的音符」的距離，前六名為 `'6.29', '3.36', '2.83', '2.55', '2.40', '2.39'` ，且只有35首 `>1` ，7首 `>2`
    - 

- 調整：根據上述ground truth給予的資訊，調整不同的threshold，發現distance = `1.05` 時有最好的表現，大約可以提升 `0.00025`

結論

- 在評估參數調整的過程中，有時會發生結果與預期不同的狀況
    - 例如zcr的high threshold，原本預期調高一些準確率會較高，但最後得到較好結果的是threshold相對不嚴苛的嘗試
    - dp實作中，尋找最短路徑的誤差門檻，在接近0的時候表現最好；但如此一來便與沒有設門檻差距不大
    - 以上狀況發生的原因可能為訊息量的缺乏，因為每次進行的嘗試都是獨立的，無法從過去的評估有效推測參數應該如何調整；若想改善，可以套用機器學習方法，也許能更準確地找出最佳參數