

## 二創短片推薦優化系統—以動畫為研究對象

專題組員:何仁翔、彭嘉勳、寇舒婷、林祥彬

專題編號: PRJ-NTPUCSIE-108-001

執行期間:2019 年 09 月 至 2020 年 05 月

### 1. 摘要

影音平台日漸興盛，影片種類繁多，影片內容也因此分眾化，然而現今各大影音平台對於分眾化影片的推薦模式並不完善，多數都以協同過濾（Collaborative Filtering）的方式推薦影片。是以，本專題將以 bilibili 影音平台上的 MAD（音樂影片動畫）、AMV（動畫音樂錄影帶）此二類動畫為研究案例，在影片的畫面、音訊外，亦將影片的彈幕文字及影片型態標籤納入推薦的參考依據，以使用者協同過濾（User-Based Collaborative Filtering）和情境感知（Context Aware）技術的概念為基礎，達到 MAD、AMV 的推薦優化。為驗證本專題成果效用，我們收集 300 部 MAD、AMV 短片，並邀請 30 人進行實驗，結果顯示，在非個人化推薦方式的效果提升 34.7%，而個人化提升 7%。

關鍵字：影片推薦、協同過濾、情境感知

### 2. 簡介

隨著網路的發展，在家娛樂興起，影音平台的影片種類也更加多元，除了時間較長的電影外，現在更多創作者製作出短影片的內容，還有類型較獨特的二次創作。影片內容可分為音訊和影像，故現今各大影音平

台上的影片推薦方式多數是以使用影音中的音訊和影像為特徵，進行影片推薦，或是閱聽人瀏覽過或喜歡的內容作為基礎資訊後再行推薦。以 YouTube 為例，YouTube 主要使用搜尋紀錄、觀看紀錄與閱聽人的統計資料，以協同過濾（Collaborative Filtering）的方式推薦影片[1]。然而，在最大的中文二創影音平台 bilibili 上，影片內容除了音訊、影像，還包含閱聽人在影片播放時即時回應影片的彈幕：影片的字幕與劇情的發展或角色的性格與行為較為相關，因此，字幕可說是延伸自影片本身，與閱聽人較無關係；影片留言區的內容，通常是閱聽人在影片結束後對影片整體的評論。然而，彈幕是閱聽人觀看影片後，將當下立即的感受轉換成這項回饋。因此，與影片的字幕和留言相比，彈幕在直觀上較台詞更能體現閱聽人的特徵。另一方面，在過往推薦系統的相關研究中，類別也是常見的參考特徵，但影片種類繁多，我們因此採用閱聽人所發的彈幕替影片進行標籤分類，將記載於彈幕的文字中閱聽人的感受回饋的概念轉化並融入到標籤類別的層次。而我們透過建構推薦模型的方式，進行實驗與資料分析，從科學與理性的層面探討彈幕之於閱聽人所具備的影響力與背後隱藏的含意為何。我們將探討在

既有的推薦系統上，加入彈幕文字與自彈幕衍生出的標籤作為特徵，對於推薦結果所能帶來的影響。我們預期，加入此元素後，會比以往的影片推薦模式還來得完善。故我們將從影片本身的音訊和影像元素，並加入彈幕，設計一套推薦優化系統。

### 3. 專題進行方式

以 bilibili 平台的 MAD、AMV 為研究案例，我們將從影片的音訊、影像畫面、彈幕文字等三方面進行影片的特徵擷取，並利用機器學習連結彈幕文字與影片型態標籤間的關係，以獲得標籤特徵。

此後，我們考量非個人化、個人化兩種推薦情境，差異在於使用者是否登入網頁：非個人化（Content-Based）推薦技術實踐：以音訊、影像、彈幕文字、標籤綜合計算閱聽人正在觀看的影片與其他影片之 Cosine 相似度，以相似度高者進行推薦；個人化推薦則是基於使用協同過濾（Collaborative Filtering）和情境感知（Context Aware）的概念進行影片推薦。

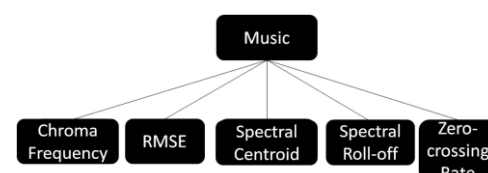
最後，我們將三種推薦方法：Content-Based Recommendation、User-Based Collaborative Filtering、Context Aware User-Based Collaborative Filtering，搭配四種影片特徵：音訊、影像、彈幕、標籤實踐於本專題設立之影片推薦系統，讓使用者可以直接體驗 MAD、AMV 影片推薦優化。

#### 資料擷取與處理：

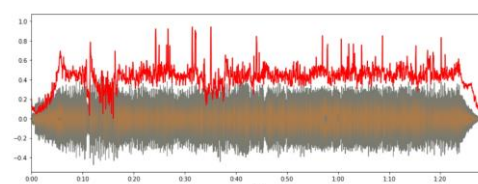
##### A. 音訊特徵的擷取：

我們使用 Python 的附加套件

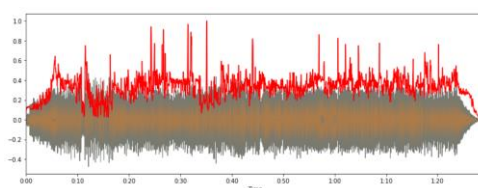
Librosa 分析音訊內容：將影片音訊擷取特徵後，取出 Chroma Frequencies（將音高進行傅立葉轉換）、RMSE（方均根差：估計量與  $\theta$  之差期望值的平方根）、Spectral Centroid（聲音的質心位置：計算方式為聲音中存在的頻率的加權平均值）、Spectral Rolloff（訊號衰減率：低於特定百分比的總頻譜能量之頻率）、Zero-Crossing Rate（訊號變化率）[2]。接著將上述特徵組合成一個長度為 5 的向量，對每個向量之間的元素正規化。



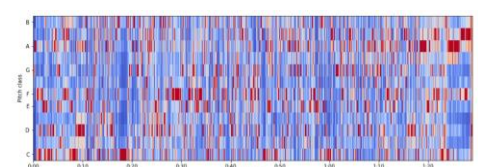
（圖一：音訊擷取的五個特徵）



（圖二：Spectral Centroid）



（圖三：Spectral Rolloff）

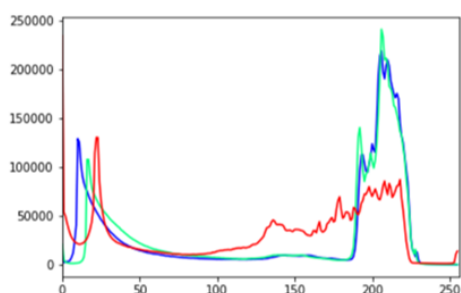


（圖四：Chroma Frequencies）

##### B. 影像特徵的擷取：

我們使用 Open-CV 分析影像內容：從一部影片中以 Open-CV 擷取出 keyframe，並提取 keyframe 中的

顏色直方圖，透過直方圖來表示該影像中的色彩分佈，分別是 RGB 三原色，取值範圍從 0~255。並將各張影像該色彩的像素加總，以此表示一部影像裡紅色佔了多少比例，綠色佔了多少比例，藍色佔了多少比例。最後，統計 RGB 從 0~255 出現的 pixel 數並存成 img.csv 檔 [3]，如圖五。接著將上述特徵（RGB 從 0~255 出現的 pixel 數）組成一個長度為  $256 \times 3$  的向量，對每個向量之間的元素正規化後，將上述標籤向量計算 Cosine 相似度。



（圖五：影像顏色分布）

### C. 彈幕特徵的擷取：

我們使用 jieba 和 TF-IDF 分析彈幕內容：將影片彈幕擷取特徵後，使用 jieba 進行中文斷詞，並且清除停用詞 [4]。接著以 TF-IDF 提取關鍵詞 [5]。完成後，將處理好的斷詞結果組成一個包含詞頻與逆向文件頻率的向量，對每個向量之間的元素正規化。

### D. 標籤的擷取：

由於平台使用者上傳影片時所附的標籤種類數量龐大，影片標籤也不一定全數與影片內容相關，因此我們請組員中兩位對二次創作領域較為熟悉的組員，選取較具有代表性的 8 種標籤類型：回憶向、校

園向、治癒向、萌向、唯美向、戀愛向、戰鬥向、獵奇向，如圖六。



（圖六：標籤所表示的影片種類）  
定義完成後，我們手動將各類影片找出 3 部影片，並人工加上標籤，一共人工標註 24 部影片。人工標註完成後，我們將代表該標籤的影片彈幕進行斷詞，並計算各類型彈幕斷詞的詞頻，以此詞頻組成該標籤的特徵向量。另外的 276 部影片使用已經完成的標籤特徵向量計算 Cosine 相似度，電腦將自動歸類該影片為最相似的標籤，並完成所有影片標註。而所有標籤的格式皆為長度為 8 的向量。

### 推薦方法：

#### A. 非個人化 (Content-Based)：

a. 使用影片音訊、彈幕、影像、標籤推薦：

將每一部影片依照特徵擷取的結果組成標籤向量，有該標籤則值為 1，無該標籤則值為 0。對每個向量之間的元素正規化後，將上述標籤向量計算 Cosine 相似度，根據 Cosine 相似度的結果推薦前三名

的影片。

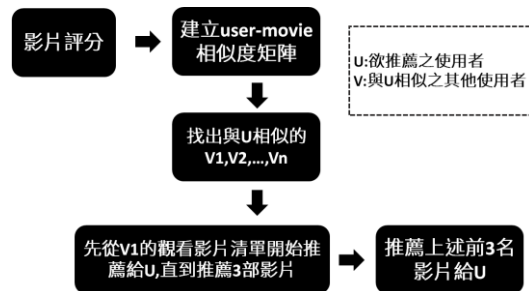
b. 綜合影片音訊、彈幕、影像的推薦：

將影片、音訊、彈幕、標籤的特徵依照 1:1:1:1 的權重組成一個向量，對每個向量之間的元素正規化，根據 Cosine 相似度的結果，根據結果推薦前三名的影片。

## B. 個人化：

a. 使用者協同過濾 (User-Based Collaborative Filtering) [6]：

閱聽人會對影片送出滿意度評分 (0~5 分)，我們根據閱聽人送出的評分，建立 user-movie 相似度矩陣，並且每隔 30 秒更新一次相似度矩陣。再根據 user-movie 相似度矩陣，以 Cosine 相似度計算該閱聽人與其他閱聽人的相似程度並進行相似排名，再從其他閱聽人的觀看影片清單開始推薦影片，推薦到滿 3 部為止，如圖七所示。

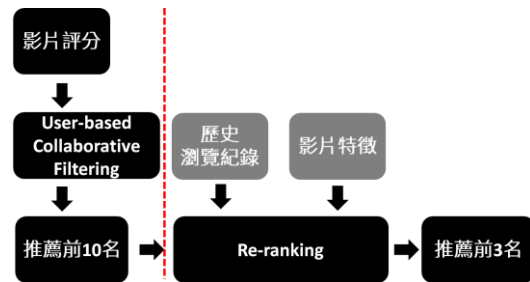


(圖七：使用者協同過濾推薦流程)

b. 情境感知推薦 (Context Aware User-Based Collaborative Filtering)：

根據前述閱聽人進行 User-Based Collaborative Filtering，先產生前 10 名的推薦影片，並根據閱聽人的歷史瀏覽紀錄選取最近 5 部瀏覽過且評分為 4 或 5 分的影片，將前 10 名推薦影片與其之相似度重新排序

(re-ranking)，以歷史瀏覽影片之特徵計算 cosine 相似度前 3 名的影片進行推薦，如圖八所示。

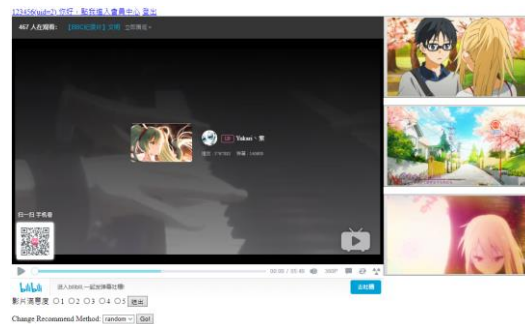


(圖八：情境感知推薦流程)

## 推薦系統環境及介面功能：

由於影片類型眾多，我們將以 MAD、AMV 為例，並以 bilibili 平台的影片為基礎，自行設立一個影片推薦系統為目標。系統環境使用 Windows 10，網頁伺服器使用 apache，後端使用 php，資料庫使用 MySQL，而影音內容分析程式使用 python 撰寫。

閱聽人在網頁端瀏覽推薦影片後，可點選網頁介面依照對推薦影片的喜好程度進行評分並回傳給後端的 MySQL 所建立之資料庫，以利於本次實驗計算推薦結果之 precision。



(圖九：推薦系統介面預覽)

## 4. 主要成果與評估

影片樣本數量為 300 部影片，使用者樣本數量為 30 位使用者，年齡分布為 17-26 歲，平均年齡為 22.4 歲，男女比例為 75%:25%，其中 26.6%為資工系，20%為高中生，13.3%為資管系，10%為社會人士，大學其他科系佔 30%。

我們評估模型的方式是以使用者在評分系統中於觀看後為影片評分，滿意度評分從 0~5 分，實驗將滿意度達一半以上（4、5 分）的評分認定為相關的（relevant）評分，表示推薦成功，其中 True Positive（真正，TP）：將正類預測為正類數、True Negative（真負，TN）：將負類預測為負類數、False Positive（假正，FP）：將負類預測為正類數、False Negative（假負，FN）：將正類預測為負類數，並以此計算出推薦方法的

$\text{precision} = TP / TP + FP$  [7]

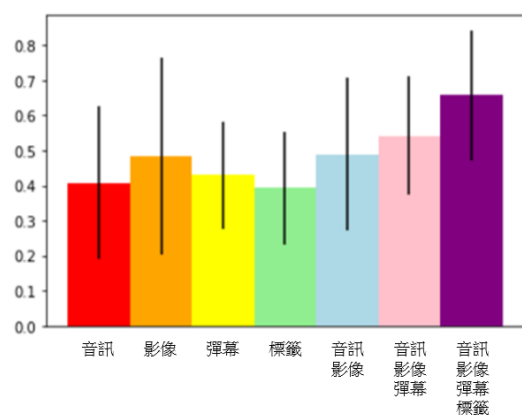
$$P = \frac{TP}{TP + FP}$$

（式一：precision 計算方式）

目前得出的效果如下：

#### a. 非個人化推薦：

如下圖十所示，其中每個柱狀的直線表示變異數，y 軸表示 precision。



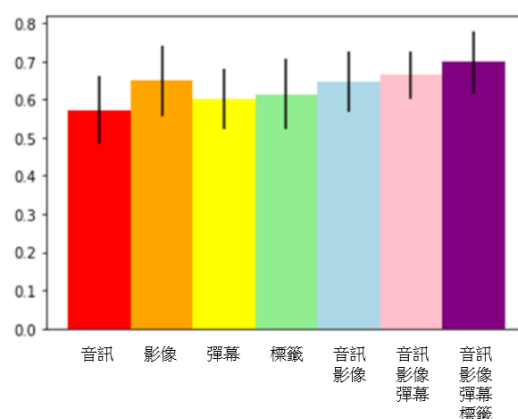
（圖十：非個人化推薦結果）

經由圖十，影像特徵有最高的 precision，同時變異數也較大，與之相比，音訊的 precision 較低，而且它的變異數較小，實驗結果顯示，畫面是一部影片呈現在閱聽人面前時最直觀使大腦做出直接反應的要素，所以推薦結果是否受到閱聽人喜愛也較

為明顯，故 precision 較高；然而，影像的變異數也較大，表示平均而言影像的變化幅度較其他特徵來的高。而標籤是自彈幕衍生出的結果，由於標籤是將記載於彈幕的文字中閱聽人的感受回饋的概念轉化並融入到標籤類別的層次，故 precision 比起彈幕較低，變異數也稍微提高。

我們將影像與音訊以 1:1 權重推薦，與影像、音訊、彈幕以 1:1:1 銓重進行比較，可發現彈幕提升了推薦的 precision，幅度為 11%，並且變異數也有一定程度的下降。再加上標籤，並以權重 1:1:1:1 進行比較，發現比起使用三個特徵，四個特徵的 precision 又提高了 21%，這說明了標籤在經過彈幕的壓縮後，更容易判斷閱聽人的喜好。對於一般的僅採用影像與音訊的推薦模式總共提高了 34.7%，這說明彈幕及彈幕歸納的標籤對於推薦有顯著的提升，也暗示其他閱聽人對於影片當下的感受有助於讓閱聽人形成共感，才會推薦成功。

#### b. 個人化推薦：



（圖十一：個人化推薦結果）

經由圖十一，可以看到 precision 比起非個人化推薦高上不少，這是由於經過 User-Based Collaborative Filtering 處理的結果：閱聽人基於喜

好程度送出的評分結果會在此推薦模式中的協同過濾步驟時互相影響。並且，與非個人化推薦有類似的結果：影像的 precision 最高，變異數也最大。而標籤的變異數雖然與非個人化推薦的情形相似，但標籤是將彈幕第二次淬鍊的結果，所以在個人化推薦時，標籤對使用者的代表性比彈幕更好，故標籤的 precision 在此處比起彈幕得來的稍高。

與前述相同，將影像與音訊以 1:1 權重推薦，與影像、音訊、彈幕以 1:1:1 銓重進行比較，可發現彈幕提升了推薦的 precision，幅度為 2%，並且變異數也有一定程度的下降。再加上標籤，並以權重 1:1:1:1 進行比較，發現比起使用三個特徵，四個特徵的 precision 提高了 5%。對於一般的僅採用影像與音訊的推薦模式總共提高了 7%，這說明無論是個人化或非個人化的推薦，加入彈幕與標籤後，對於推薦效果都有一定程度的提升。

## 5. 結語與展望

本次專題鎖定了平時不易被注意及重視的二次元主題相關影片為實作範圍，針對影像色彩分布、音訊特徵與彈幕特徵相似度去分析。雖然仍無法達到一拿到新增樣本即馬上分類的層級，但已能在特定範圍內推薦出適當的結果。若使用者平時就有接觸相關影片作為日常消遣，想必看到這些成果也能更加享受自己喜好的事物。最後期望未來能有人更加重視特定族群的影片分類、分析及推薦，不單單只是把二次元圈與一般社會流行區隔開，而是深入進行研究探討。

## 6. 銘謝

特別感謝專題指導教授在各方面提供我們研究方向的指點與技術的提供，進度落後時也會點出我們根本上的問題讓我們進行修正。

感謝所有資料分析在畫面、音訊、字詞方面的所有拓荒者。

## 7. 參考文獻

[1] Paul Covington and Jay Adams and Emre Sargin. (2016). Deep Neural Networks for YouTube Recommendations. New York, NY, USA: Association for Computing Machinery.

<https://research.google/pubs/pub45530/>

[2] Parul Pandey. (2018). Music Genre Classification with Python.

<https://towardsdatascience.com/music-genre-classification-with-python-c714d032f0d8>

[3] opencv 提取图像中的颜色直方图 (RGB、HSV)

[https://blog.csdn.net/Jaster\\_wisdom/article/details/83585378](https://blog.csdn.net/Jaster_wisdom/article/details/83585378)

[4] 用 Python 进行简单的文本相似度分析

<https://blog.csdn.net/xiexfl89/article/details/79092629>

[5] TF-IDF 与余弦相似度

<https://zhuanlan.zhihu.com/p/32826433>

[6] 基于用户的协同过滤算法的电影推荐系统

<https://blog.csdn.net/u012050154/article/details/52268057>

[7] 机器学习：准确率 (Precision)、召回率 (Recall)、F 值 (F-Measure)、ROC 曲线、PR 曲线

[https://blog.csdn.net/quiet\\_girl  
/article/details/70830796](https://blog.csdn.net/quiet_girl/article/details/70830796)