基於 KNN 演算法之新聞分類

天主教輔仁大學資訊工程學系
資工碩一 405226215 廖宜楷
242 新北市新莊區中正路 510 號
Mail: kobe821125@gmail.com

天主教輔仁大學資訊工程學系
資工碩一 405226227 許登傑
242 新北市新莊區中正路 510 號
Mail: gt810034@gmail.com

天主教輔仁大學資訊工程學系
資工碩一 405226057 顏嘉村
242 新北市新莊區中正路 510 號
Mail: a82585578@gmail.com

I. INTRODUCTION

在文字探勘的領域中，自動文章摘要是一個重要的研究議題，其作用是將一篇文章如新聞，自動擷取比較重要的文字來代表這篇新聞，但這對我們目前來說有點困難，所以我們從比較簡單的新聞內文分類開始做起。

在搜集資料上，我們在聯合新聞網（udn.com）上抓了三百三十篇的新聞，分類分別為體育、財經以及世界新聞各一百一十篇。在機器學習的演算法上，我們選的是 KNN，將三百篇新聞當作訓練資料，三十篇新聞當作測試資料，得到一個可以判斷文章為哪個分類的系統。

II. PREPARE DATA

我們的資料來源是來自網路新聞，在準備資料的時候，需要注意下面兩件事情，讓我們後續建立文字模型能夠順利。

A. 篩選新聞來源

我們是利用 python 做大量自動擷取網頁內容，所以我們必須找到，網頁內容結構較清晰的新聞來源，以利後面的資料處理能夠更為順利。

為此，我們找了許多不同的新聞來源，如蘋果新聞網、聯合新聞網、中時電子報以及自由時報電子報等，其中，頻果新聞網的內容中，夾雜了許多不同的 html 標籤，屬於較難處理的類型，而聯合新聞網，排版優美，內文整齊，且已斷好行，屬於非常優良的資料來源。因此我們選用聯合新聞網。

B. 內文前處理

將新聞擷取下來後，將所有內文中的英文轉成小寫，為了避免有些英文字，字首是大寫，導致明明是一樣的意思，卻為不同的兩個字，會造成訓練上維度不必要的增加。

C. 斷詞

對於文字探勘，斷詞是一個非常重要且必要的步驟，我們這裡選用開源的中文斷詞系統－結巴（Jieba），原因是他是基於 python 所開發的斷詞工具，而且支援繁體中文，斷詞結果雖然不是完美的，但讓人滿意，使用起來也相當方便、快速。

III. 建立文字模型

對於文章，每個字在每篇文章中的重要度並不相同，因此需要加入權重來凸顯文章中的重要詞彙，最後將文章內容轉成向量。

A. 權重

我們使用 TF-IDF(Term-Frequency-Inverse-Document-Frequency)做為我們文字在文章中的權重，參考圖一，TF 為這個詞語在某篇文章中的出現頻率，IDF 為這個詞語出現在所有文章中的出現頻率之倒數，然而每個研究對於頻率的定義都有所不同，我們在這是採取最通用的公式。

$$\underbrace{\frac{\text{該詞在該文章出現的次數}}{\text{該文章中的總詞數}}}_{\text{TF}} * \underbrace{\log\left(\frac{\text{文章總數}}{\text{該詞出現在幾篇文章中}}\right)}_{\text{IDF}}$$

圖 一 TF-IDF

B. 向量空間模型

首先計算，在三百篇新聞中，總共有多少個不一樣的字，以確定維度。參考圖二，依序將每篇文章的每個字，在那個字的位置上，會加上那個字的權重值，最後會得到訓練資料共三百個向量。

D1	他 中餐 吃 便當
D2	我 中餐 吃 土

計算TF-IDF

$\frac{1}{4} * log_2(\frac{2}{1}) = 0.25$ 他

$\frac{1}{4} * log_2(\frac{2}{2}) = 0$ 中餐

$\frac{1}{4} * log_2(\frac{2}{2}) = 0$ 吃

$\frac{1}{4} * log_2(\frac{2}{1}) = 0.25$ 便當

$\frac{1}{4} * log_2(\frac{2}{1}) = 0.25$ 我

$\frac{1}{4} * log_2(\frac{2}{1}) = 0.25$ 土

將文字轉成向量

	他	我	中餐	吃	便當	土
D1	0.25	0	0	0	0.25	0
D2	0	0.25	0	0	0	0.25

圖 二 文字轉向量範例

IV. K 個最相近鄰居演算法

再經過前一章節的 TF-IDF 權重計算以及文字向量轉換後，我們使用了 K-Nearest Neighbor(KNN)分群方式作為判別輸入文章屬於哪一項文章分類的演算法。

A. K 值

KNN 分類演算法簡單來說就是要找和新數據最近的 K 個鄰居，進而決定新數據屬於哪個分群。以圖三來說，原先給定的樣本有兩種分類族群，紫色圓點代表 ClassB, 黃色圓點代表 ClassA 紅色星星則為需要定義分群的新數據。在 K 設定為三的時候，離紅色星星距離最近的三個數據點有兩個代表 ClassB 的紫色圓點及一個代表 ClassA 的黃色圓點，因此此時新數據會被歸類為 ClassB，而在 K 設定為六時，距離最近的六個數據點有四個代表 ClassA 的黃色圓點以及兩個代表 ClassB 的紫色圓點，因此新數據會被歸類在 ClassA。

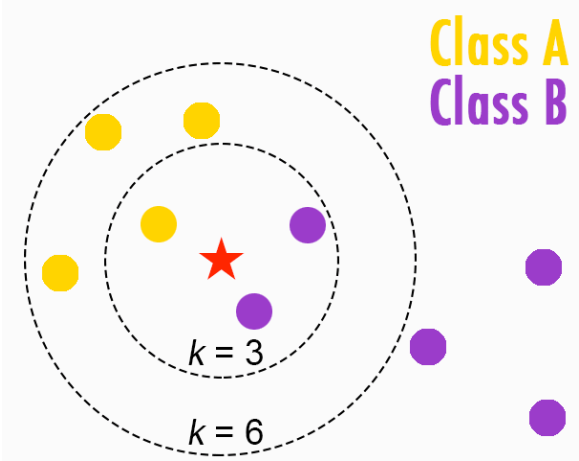

圖 三 KNN 示意圖

B. 相似度計算

在圖四的例子當中，當數據點以二維空間表示時，我們能使用最常見的歐式距離公式計算數據點之間的距離，但大多數實際案例中的資料並無法使用簡單的二為座標呈現，以我們的主題為例，將文章中的文字轉成向量的表現的方式，並以 Cosine similarity 的公式計算兩項向量之間的夾角已得到一個介於-1 與 1 之間的數值代表它們之間的相似度程度，值越高則代表彼此的距離越相近。

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|_2 \|\mathbf{B}\|_2} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

圖 四 Cosine similarity

V. 實驗方法

利用上述三百筆新聞當作訓練資料，利用 KNN 作為訓練的演算法，預測另外三十筆新聞資料所屬的分類。由於在 KNN 裡，一開使需要輸入 K 值，參考附近的 K 個鄰居，在這我們嘗試了 K= 1, 3, 5, 7。以及在輸入測試資料時，文字有無加入權重，皆為我們的考量之一。參考圖五為準確率的定義，後面評估的部分透過模糊矩陣來計算準確率。

$$Accuracy = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN}$$

圖 五 Accuracy 定義

VI. 實驗評估

由於有四個 K 值，有無權重，會有八種結果，這邊模糊矩陣只顯示出有權重的最好 K 值和無權重的最好 K 值。

參考圖六為 K=3，沒權重的部份，Accuracy 為 0.67，我們可以發現，Sport 的 Recall 為 1，要是目的是要預測 Sport，我們可以很準確的預測出來，而 Precision 為 0.59，不算太低。

		Predict class		
		Finance	Sport	Global
Actual class	Finance	7	3	0
	Sport	0	10	0
	Global	3	4	3

圖 六 K = 3, 無權重之模糊矩陣

參考圖七為 K=3，有權重的部分，Accuracy 為 0.53，整體的準確率不算很高，雖然 Sport 的 Recall 依然為 1，但 Precision 為 0.43，相對來說結果不太好。

		Predict class		
		Finance	Sport	Global
Actual class	Finance	4	6	0
	Sport	0	10	0
	Global	1	7	2

圖 七 K = 3, 無權重之模糊矩陣

VII. 結論

在有無權重的部分，我們尚未找到實際原因，加了權重反而準確率下降，我們的推測是訓練的資料不夠多，在文字向量一萬多維的情況下，我們只有三百筆的訓練資料。未來如果有要繼續往這個方向發展的話，首先要累積大量的資料。在 K 值的部分，我們測試出來的結果都是 K=3 時，會有最好的準確率。在結果的部分，新聞分類的最好準確率為 0.67，而如果是預測 Sport 的話，我們可以百分之百的預測出來。

參考資料

[1]KNN HTTP://ENGINEBAI.LOGDOWN.COM/POSTS/241676/KNN

[2]INTRODUCTION TO DATA MINING HTTP://WWW-USERS.CS.UMN.EDU/~KUMAR/DMBOOK/INDEX.PHP

[3] Quick Tour of Text Mining https://www.slideshare.net/tw_dsconf/ss-73708487