巨量資料導論期中報告

--K-means與fuzzy c-means實作

一、組員:

資工109級B053040012羅紹瑄 資工109級B053040014陳鵬宇

二、簡介:

* K-means:

K-means 集群分析(又稱c-means Clustering，中文: k-平均演算法，基本上Clustering的方法大都是非監督式學習(Unsupervised learning)，K-means也是非監督式學習。

* Fuzzy c-means:

將K-means依據模糊理論所衍生的演算法，利用membership function計算出各資料對各群的隸屬值，不強制歸為某一類，能將資料的特性保留地更完整。

三、運作概念步驟:

* K-means

1. 我們先設定好要分成多少(k)群。

2. 然後在feature space隨機給k個群心。

3. 每個資料都會所有k個群心算歐式距離(

4. 將每筆資料分類判給距離最近的那個群心。

5. 每個群心內都會有被分類過來的資料，用這些資料更新一次新的群心。

6. 一直重複3–5，直到所有群心不再有太大的變動(收斂)，結束。

* Fuzzy c-means:

1. 同樣，先設定好要分成多少(c)群。

2. 並且一樣在feature space隨機給c個群心。

3. 每個資料都會所有c個群心算歐式距離

4. 計算每筆資料與各群心的隸屬度。

5. 利用所有資料的隸屬值，更新群心座標。

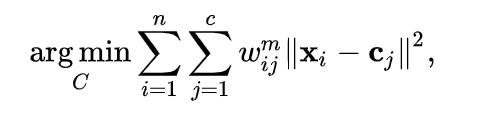
6. 一直重複3–5，直到所有群心不再有太大的變動(收斂)，結束。

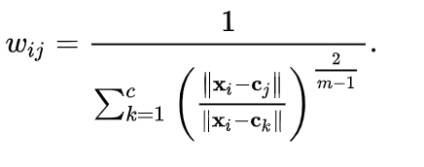
四、更新公式

* K-means:

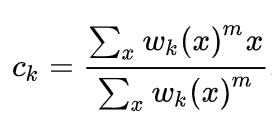
K-means clustering目的在最小化群內的資料和群心的誤差平方和，數學公式如下:

* Fuzzy c-means:

 FCM欲最小化的objcective function:



更新資料隸屬度的公式:



更新資料群心座標的公式:

五、使用資料集

* Scikit-learn的iris dataset
* Abalone dataset

六、實作方法:

用python輔以numpy,pandas等library來實作:

* K-means:

第一步-Data Preprocessing:

把資料分成data\_x與data\_y ，data\_x是要訓練的資料feature，格式可為pandas.DataFrame或numpy.array ; data\_y 則是資料的label,計算準確率用,格式須為numpy.array。

另外還需將資料做額外預處理，我們用的方法是scikit-learn裡的preprocessing方法，主要是preprocessing.scale這個function，把資料標準化，即mean= 0 , variance = 1。

第二步-把k-means寫成class:

把k-means寫成class的好處是只要將資料做好預處理，便可丟入產生結果，提供的主要函數有:

1. run() : 執行k-means，步驟如(五) 。
2. calculate\_acc() : 計算準確率，以驗證演算法實作的結果；大致的步驟是:分別計算分類完後各群的重心與原本各label的重心、把最相近的重心判斷為同一類以重新命名、再與原本的label相比以計算準確率。若有給定Simulate\_time (欲模擬的次數)，則會把每一次的準確率做平均計算。
3. print\_cent() : 印出該次做完k-means後的群心座標(若超過一次則只印出最後一次) 。
4. print\_result() : 印出k-means結果、最後一次的sum of square error、最後一次迭代的變動點個數、以及迭代次數。
5. reault.acc : 此為變數，紀錄平均準確率。

第三步-印出結果:

印出群心座標與平均準確率以察看結果。

第四步-與scikit-learn的cluster.KMeans做比較:

將我們的結果與cluster.KMeans做比較，但由於cluster.KMeans並無提供計算準確率的函式，所以依然需要做relabeling後才能比較。

* FCM :

大致過程與K-means相同，但如(五)所述，要計算membership values與更新群心的過程需要引入數學公式，較為複雜；另外為了做，準確度的比較，我們挑了每筆資料的隸屬度中，最高的那筆作為其所屬的cluster。

七、實作結果

對於iris dataset:

我們的K-means約為0.87

FCM 約為0.88

Scikit-learn 約為0.89

對於abalone dataset:

用11個rings當作分界分兩類:

我們的K-means約為0.663

FCM 約為0.668

Scikit-learn 約為0.665

八、遇到的困難

1. Python dataframe 處理

Pandas的DataFrame是個好用的資料格式，有很多函式可以處理資料，但也因為函示很多，所以在不夠熟練的情況下，反而操作起來有些費時，我們花了不少時間在搞懂他。

2. abalone dataset

Abalone dataset的年齡有28類，加上不同的年齡並無那麼多的差異，個體的差異性或許會更大些，所以分出來的效果很差，為此，我們參考了網路上別人處理的方式與同學的建議，決定用11歲當分界，才有較好一點的準確率。

3.FCM的實作

利用公式更新並非難事，但我們花了些時間才搞懂，FCM中所謂的收斂，是指objective function 的value的變化，而由於FCM較花時間，所以對於確認結果來說，也比較費神。

九、結論:

1.Iris 似乎不做正規化準確率比較高

2.Iris, Abalone dataset 用FCM的分群效果較佳