**Analisi RandIndexCalculator**

**Problema**

Un data set artificiale è salvato in un file di testo seguendo il formato

-81.5726317891524,65.893764821124 …

dove le prime k righe, per un data set da k punti ed n cluster, fanno riferimento al primo cluster; le righe da k+1 a 2k fanno riferimento al secondo cluster e così via.

L’output dell’implementazione BrizziB è salvato su file nella forma

-80.914590830844 66.5111812711113

-80.32974400996152 63.67437888508667

dove alla prima riga è stampato il centroide, le altre sono i punti che fanno parte del cluster rappresentato dal centroide stesso. È possibile che un file contenga più di un cluster, può quindi accadere che alla riga k sia stampato un centroide e dalla k+2 in poi i punti che fanno riferimento ad esso. La riga k+1 sarà vuota (\t\n).

Riconosciamo la presenza di un nuovo cluster dalla assenza del carattere di tabulazione.

L’output dell’implementazione Jgalilee è salvato su file di testo, nella forma

1 -47.55515667108383 72.89179626563877 2 -80.91459083084465 66.51118127111107 3 -95.80069117827341 -84.07661573696988 …

Ad ogni riga corrisponde un centroide (o rappresentante) di cluster preceduto dal suo codice identificativo (a partire da 1). Sarà quindi necessario considerare il file di input di testo contenente tutti i punti del data set, nel seguente formato

-81.2622959409307 65.84777093920698 -80.32974400996152 63.67437888508667 -83.16982883084457 69.21286832653041 …

E associare ognuno di essi al cluster corretto, definito dal centroide più vicino considerata la distanza euclidea; al fine di ricostruire il data set risultante dall’operazione di clustering.

L’output dell’implementazione Yhfyhf è salvato su file di testo nel seguente formato

0 -46.4452 71.7803 0 -46.596603 73.60119 1 -49.32123 70.934906 …

Dove il primo numero identifica il cluster di appartenenza (a partire da 0). I rappresentanti di questo cluster sono stampati a video seguendo il formato:

[-47.555946 72.89613, -80.918594 66.511505, -95.79116 -84.07881]

Questi vengono trasferiti su file di testo composto da una singola riga coincidente con il formato sopra mostrato.

È necessario considerare la ground truth del data set Indian Pines scaricabile all’indirizzo <http://www.di.uniba.it/~appice/software/S2TEC/index.htm>.

È inoltre necessario tenere conto che i punti, in tutti i casi visti, possono essere espressi in notazione scientifica.

Considerata la coppia (data set artificiale, implementazione) con implementazione ∈ {BrizziB, Jgalilee, Yhfyhf}, data set artificiale ∈ {Indian Pines, 2d3c, 2d5c}, si vuole calcolare l’indice di Rand.

Nota: data set artificiale è sempre un file di testo nel formato csv.

**Soluzione**

Per risolvere il problema è necessario ottenere una rappresentazione, per ogni punto del data set in analisi, nella forma:

-80.32974400996152,63.67437888508667,1,2

Dove i primi due numeri reali, sono il punto bidimensionale considerato o, nel caso del data set Indian Pines, la posizione x, y del pixel nell’immagine (comune all’output dell’implementazione scelta e al dataset artificiale scelto) e i due valori interi successivi sono rispettivamente: il codice identificativo della classe di appartenenza del punto nel data set artificiale e il codice identificativo del cluster di appartenenza del punto nell’output dell’implementazione scelta.

In definitiva, la soluzione, per l’istanza del problema (d,i), consiste dei seguenti passi:

1. Il caricamento in memoria del data set d e del data set i come insieme di coppie chiave, valore dove, considerato un punto, la chiave è univoca ed è una stringa contenente gli attributi spaziali. Il valore è il codice identificativo della classe, per il data set d, il codice identificativo del cluster, per il data set i, di appartenenza del punto.
2. La fusione dei due data set d ed i in un unico data set rappresentato da una matrice bidimensionale, le cui righe rappresentano esempi, le cui colonne sono la classe ed il cluster di appartenenza del punto.
3. Il calcolo dell’indice di rand partendo dal data set risultante dall’operazione al punto 2.

L’algoritmo RAND-INDEX utilizzato per il calcolo dell’indice di Rand prende in input la matrice M calcolata al punto 2:

RAND-INDEX(M)

a,d = 0

**for** i = 1 **to** M.length

class1 = M[i][1]

cluster1 = M[i][2]

**for** j = 1 **to** i-1

class2 = M[j][1]

cluster2 = M[j][2]

**if** class1 == cluster1 **and** class2 == cluster2

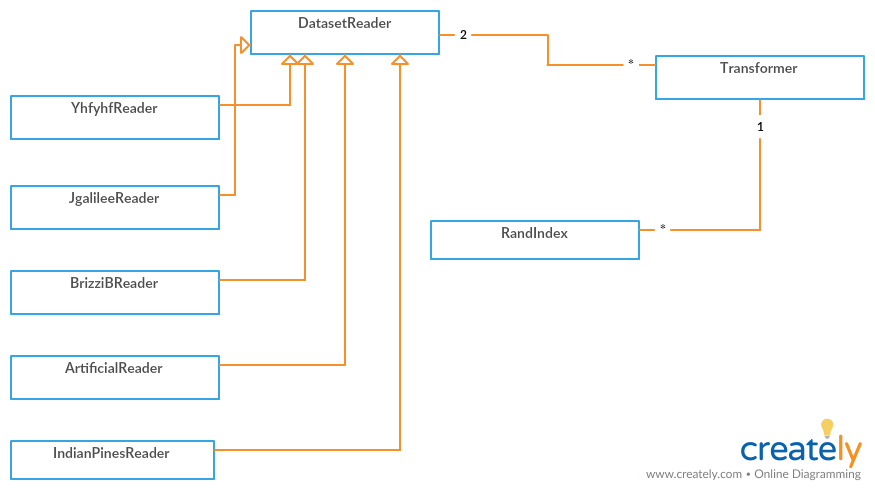
a = a + 1

**else if** class1 != cluster1 **and** class2 != cluster2

d = d + 1

**return** (a + d) / ((length(M) \* (length(M) - 1)) / 2)

La complessità in tempo dell’algoritmo è O(n2), ciò è facilmente deducibile dal fatto che, i due cicli for iterano sull’insieme di tutte le possibili combinazioni degli n (pari al numero di osservazioni o punti) indici, senza ripetizioni, e la cardinalità di questo insieme è n\*(n-1)/2, ovvero n2/2 – n/2 = O(n2).

Come diagramma delle classi UML ad alto livello (o kite level) consideriamo il seguente:

dove le responsabilità sono così distribuite:

* DatasetReader legge i punti da file assumendo che le prime due componenti di ognuno di essi identificano la posizione del punto nell’immagine
* YhfyhfReader legge il data set ottenuto dall’operazione di clustering effettuata dall’implementazione Yhfyhf.
* JgalileeReader legge il data set ottenuto dall’operazione di clustering effettuata dall’implementazione Jgalilee.
* BrizziBReader legge il data set ottenuto dall’operazione di clustering effettuata dall’implementazione BrizziB.
* ArtificialReader legge i dataset artificiali bidimensionali generati dal programma DatasetGenerator.
* IndianPinesReader: legge il data set Indian Pines per come presentato nel pacchetto del software S2Tec.
* Transformer fonde due data set contenti gli stessi punti in un unico data set.
* RandIndex calcola l’indice di Rand per mezzo dell’algoritmo RAND-INDEX.