CS533 Information Retrieval Systems

Assignment - 3

Tunç GÜLTEKİN

21203122

1)

[1] Cluster hypothesis states that; documents, those are in the same cluster, behave similarly with respect to relevance to information needs. If there is a document from a cluster which is relevant to a search query, then it is likely that other documents from the same cluster are also relevant to query. This is logical because in clustering operation similar documents are put into same clusters and similar documents are also relevant to similar queries.

2)

Clustering methods are groups of clustering techniques or algorithm type, such as; single pass, multi pass hierarchical or graph theoretical. However clustering algorithms are implementations of clustering methods. For example, single link clustering algorithm is an implementation of hierarchical clustering method.[2]

3)

Similarity measures such as Dice, Cosine and Euclidian distance are produce numerical values with respect to similarities of different documents. Results of these similarity measures are different from each other but correlated. Thus choosing a most accurate or distinctive similarity measure for an IR application is a difficult problem. In this study [7] they have implemented all of these measures in a structured way, and have done retrieval experiments on them to show which features yield good retrieval behavior in a variety of retrieval environments. They demonstrated that it is surprisingly difficult to identify which techniques work best, and comment on the experimental methodology required to support any claims as to the superiority of one method over another. The main difficulty behind that, comparing of combination of similarity measures. There are many similarity measures in the literature and comparing different combinations of them becomes an exhaustive search problem and takes too much time. Also different similarity measures produces different results for different IR environments for these reasons selecting the most proper similarity measure for specific IR algorithm environment are a difficult thing.

4)

$$D = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$t_1 \rightarrow d_1$
$t_2 \rightarrow d_1, d_2, d_4$
$t_3 \rightarrow d_2$
$t_4 \rightarrow d_1$
$t_5 \rightarrow d_3, d_4, d_5$
$t_6 \rightarrow d_5$

For d_1, it contains t_1, t_2, t_4

Due t_4

S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
x	0	0	0	0

Due t_2

S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
x	1	0	1	0

Due t_1

S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
x	1	0	1	0

⇨ S_{12}, S_{14} were calculated.

For d_2, it contains t_2, t_3

Due t_3;

S_{21}	S_{22}	S_{23}	S_{24}	S_{25}
0	x	0	0	0

Due t_2;

S_{21}	S_{22}	S_{23}	S_{24}	S_{25}

1	x	0	1	0

⇨ $S_{24,}$ was calculated. (S_{21} has already been calculated)

For d_3, it contains only t_5

Due $t_{5\,;}$

S_{31}	S_{32}	S_{33}	S_{34}	S_{35}
0	0	x	1	1

⇨ $S_{34\,,}$ S_{35} were calculated.

For d_4, it contains $t_{2,}$ t_5

Due $t_{5\,;}$

S_{41}	S_{42}	S_{43}	S_{44}	S_{45}
0	0	1	x	1

Due $t_{2;}$

S_{41}	S_{42}	S_{43}	S_{44}	S_{45}
1	1	1	x	1

⇨ S_{45} was calculated. (S_{41} $S_{42,}$ S_{43} have already been calculated)

For d_5, it contains $t_{5,}$ t_6

Due $t_{6;}$

S_{51}	S_{52}	S_{53}	S_{54}	S_{55}
0	0	0	0	x

Due $t_{5;}$

S_{51}	S_{52}	S_{53}	S_{54}	S_{55}
0	0	1	1	x

⇨ There was no computation ($S_{53,}$ $S_{54,}$ have already been calculated)

Common term counts of documents (this table is obtained from the results of previous calculations);

	D_1	D_2	D_3	D_4	D_5
D_1	x	1	0	1	0
D_2	1	x	0	1	0
D_3	0	0	x	1	1
D_4	1	1	1	x	1
D_5	0	0	1	1	x

Dice Coefficients

S_{12} = (2*1)/(3+2) = 0.4 S_{23} = (2*0)/(2+1) = 0 S_{34} = (2*1)/(1+2) = 0.67 S_{45} = (2*1)/(2+2) =0.5
S_{13}= (2*0)/(3+1) = 0 S_{24} = (2*1)/(2+2) = 0.5 S_{35} = (2*1)/(1+2) =0.67
S_{14} = (2*1)/(3+2) = 0.4 S_{25} = (2*0)/(3+3) = 0
S_{15} = (2*0)/(3+2) = 0

$$S = \begin{bmatrix} 1 & 0.4 & 0 & 0.4 & 0 \\ - & 1 & 0 & 0.5 & 0 \\ - & - & 1 & 0.67 & 0.67 \\ - & - & - & 1 & 0.5 \\ - & - & - & - & 1 \end{bmatrix}$$

a) For single link dendrogram, document pairs sorted with respect to their pair similarity values;

D_3, D_4	0.67
D_3, D_5	0.67
D_2, D_4	0.5
D_4, D_5	0.5
D_1, D_2	0.4
D_1, D_4	0.4
D_1, D_3	0
D_1, D_5	0
D_2, D_3	0
D_2, D_5	0

Corresponding single link dendrogram is below;

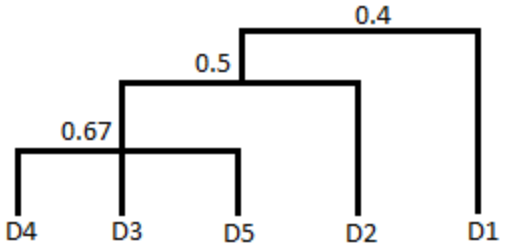


Figure – 1: Single Link Dendrogram

For this dengrogram, we can obtain 3 cluster, by cutting 2 different similarity values (0.5 and 0.4)
Cluster – 1: D4, D3, D5
Cluster – 2: D2
Cluster – 3: D1

b) For complete link dendrogram, document pairs sorted with respect to their pair similarity values;

Step	Pair Similarity	Pairs Covered
1	D_3, D_4 → 0.67	**(D_3, D_4)**
2	D_3, D_5 → 0.67	**(D_3, D_4)** (D_3, D_5)
3	D_2, D_4 → 0.5	**(D_3, D_4)** (D_3, D_5) (D_2, D_4)
4	D_4, D_5 → 0.5	**(D_3, D_4) (D_3, D_5)**(D_2, D_4)**(D_4, D_5)**
5	D_1, D_2 → 0.4	**(D_3, D_4)** (D_3, D_5) (D_2, D_4)(D_4, D_5)**(D_1, D_2)**
6	D_1, D_4 → 0.4	**(D_3, D_4) (D_3, D_5) (D_2, D_4)(D_4, D_5)(D_1, D_2)(D_1, D_4)**
7	D_1, D_3 → 0	
8	D_1, D_5 → 0	
9	D_2, D_3 → 0	
10	D_2, D_5 → 0	

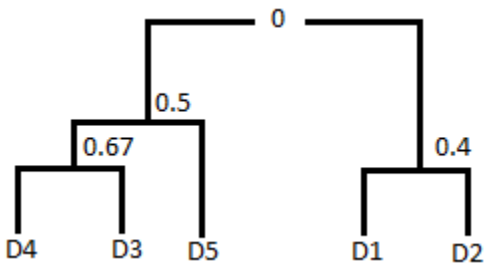


Figure – 2: Complete Link Dendrogram

Initially we have 2 different clusters (D_1,D_2) and (D_3,D_4,D_5). Because their similarity value is 0 and if we use previous similarity thresholds, we can obtain these clusters;

Cluster – 1: D4, D3
Cluster – 2: D5
Cluster – 3: D1, D2

c) Similarity matrix is below;

$$S = \begin{bmatrix} 1 & 0.4 & 0 & 0.4 & 0 \\ - & 1 & 0 & 0.5 & 0 \\ - & - & 1 & 0.67 & 0.67 \\ - & - & - & 1 & 0.5 \\ - & - & - & - & 1 \end{bmatrix}$$

The similarity matrix implied by the complete link dendrogram is below;

$$S'' = \begin{bmatrix} 1 & 0.4 & 0 & 0 & 0 \\ - & 1 & 0 & 0 & 0 \\ - & - & 1 & 0.67 & 0.5 \\ - & - & - & 1 & 0.5 \\ - & - & - & - & 1 \end{bmatrix}$$

$$r(X,Y) = \frac{cov(X,Y)}{\sqrt{Var(X).Var(Y)}}$$

By using Matlab, product moment was calculated as follows;

```
>> x=[1 0.4 0 0.4 0;0.4 1 0 0.5 0;0 0 1 0.67 0.67;0.4 0.5 0.67 1 0.5; 0 0 0.67 0.5 1]

x =

    1.0000    0.4000         0    0.4000         0
    0.4000    1.0000         0    0.5000         0
         0         0    1.0000    0.6700    0.6700
    0.4000    0.5000    0.6700    1.0000    0.5000
         0         0    0.6700    0.5000    1.0000

>> y = [1 0.4 0 0 0;0.4 1 0 0 0;0 0 1 0.67 0.5;0 0 0.67 1 0.5; 0 0 0.5 0.5 1]

y =

    1.0000    0.4000         0         0         0
    0.4000    1.0000         0         0         0
         0         0    1.0000    0.6700    0.5000
         0         0    0.6700    1.0000    0.5000
         0         0    0.5000    0.5000    1.0000
```

```
meanx = mean(x);     %Compute mean value of X
meany = mean(y);     %Compute mean value of Y

%Compute the arguments that go into the mathematical formula of R
sx2   = sum((x-meanx).^2);
sy2   = sum((y-meany).^2);
sxy   = sum((x-meanx).*(y-meany));

% Mathematical definition of Pearson's product moment correlation
coefficient

r = 0.9063
```

d) [3] Monte Carlo analysis can approximate an unknown distribution if an experimental sampling procedure can be programmed on a computer that simulates the process being studied. To decide whether the correlation coefficient value is significant (sufficiently large) or not;

- Firstly, implied similarity matrices of different dendrograms are created.
- For each implied similarity matrix, correlation coefficients are calculated (m different value.)
- An integer k is selected and k/m = alpha = level of significance value (such as 0.05 or 0.01).
- If the investigated correlation coefficient value is the k^{th} largest of the m values

the index is significantly large and the null hypothesis can be rejected.
The probability that the Monte Carlo test would reject the null hypothesis is the probability that no more that k-1 of the samples from this distribution exceed investigated correlation coefficient or;

$$P(k,\ p) = \sum_{r=0}^{k-1} \binom{m-1}{r} p^{m-r-1} q^r$$

5)

If we insert (D2,D4) before (D3,D4) the resulting dendrogram would be below and this one is different than previous complete link dendrogram.

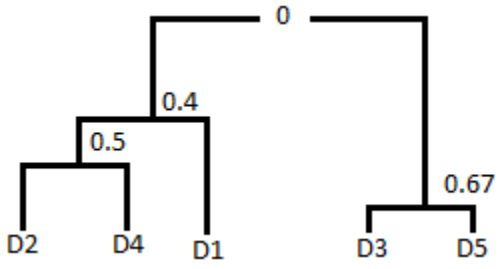


Figure – 3: Different Complete Link Dendrogram

Single link dendrograms are not order dependent because, [3] they have a continuity property. If the ties are broken in the proximities by adding or subtracting a small amount from the tied proximities, the resulting single link dendrograms will merge smoothly into the same dendrogram as the added amount tends to zero, no matter how the ties are broken.

6)

a)

Divided by row sums;

$$S = \begin{bmatrix} 0.33 & 0.33 & 0 & 0.33 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0.5 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 \end{bmatrix}$$

Divided by column sums;

$$S'' = \begin{bmatrix} 1 & 0.33 & 0 & 1 & 0 & 0 \\ 0 & 0.33 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.33 & 0 \\ 0 & 0.33 & 0 & 0 & 0.33 & 0 \\ 0 & 0 & 0 & 0 & 0.33 & 1 \end{bmatrix}$$

$C = S \cdot (S'')^T$

$$C = \begin{bmatrix} 0.7689 & 0.1089 & 0 & 0.1089 & 0 \\ 0.1650 & 0.6650 & 0 & 0.1650 & 0 \\ 0 & 0 & 0.3300 & 0.3300 & 0.3300 \\ 0.1650 & 0.1650 & 0.1650 & 0.3300 & 0.1650 \\ 0 & 0 & 0.1650 & 0.1650 & 0.1650 \end{bmatrix}$$

b)

Number of clusters is equal to the summation of diagonal elements;
0.7689 + 0.6650 + 0.3300 + 0.3300 + 0.6650 = **2.7589 ~= 3**

So we can say there are 3 clusters.

c)

Cluster seed power; $p_i = \delta_i . \psi_i . x_{di}$
where;

$\delta_i \rightarrow C_{ii}$
$\psi_i \rightarrow (1- C_{ii})$
$x_{di} \rightarrow$ number of terms in d_i

Seed powers;
$P_1 = (0.7689).(1-0.7689).3 = 0.5331$
$P_2 = (0.6650).(1-0.6650).2 = 0.4456$
$P_3 = (0.3300).(1-0.3300).1 = 0.2211$
$P_4 = (0.3300).(1-0.3300).2 = 0.4422$
$P_5 = (0.1650).(1-0.1650).2 = 0.2756$

d)

Since, $P_1 > P_2 > P_4 > P_5 > P_3$ first three terms are selected as seeds for three clusters.
Cluster seeds are; D_1, D_2 and D_4

e)

The inverted index as follows;
$t_1 \rightarrow$ **<d_1,1>**
$t_2 \rightarrow$ **<d_1,1> <d_2,1> <d_4,1>**
$t_3 \rightarrow$ **<d_2,1>**
$t_4 \rightarrow$ **<d_1,1>**
$t_5 \rightarrow$ **<d_4,1>**
$t_6 \rightarrow$

f)

To cluster D_3;

For t_5

$C_{31} = C_{31} + \alpha_3(d_{35} * \beta_5 * d_{15}) = 0 + 1*(1*0.33*0) = 0$

$C_{32} = C_{32} + \alpha_3(d_{35} * \beta_5 * d_{25}) = 0 + 1*(1*0.33*0) = 0$

$C_{34} = C_{34} + \alpha_3(d_{35} * \beta_5 * d_{45}) = 0 + 1*(1*0.33*1) =$ **0.33**

D_3 can be put into D_4's group because it has the biggest c value with respect to D_3

g)

By looking C matrix, for document 5, cluster 4's C value (C_{54}) is bigger than C_{51} and C_{52} so document 5 belongs to document 4's cluster. So the overall clustering structure is

D_1	**D_2**	**D_4**
		D_3
		D_5

h)

In efficient implementation;
$m.X_d + n_c.X_d + (m - n_c).X_d.t_{gs}$ matrix entry should have to be calculated.
Where;

$X_d \rightarrow$ Average number of terms per document
$m \rightarrow$ Total document count
$n_c \rightarrow$ Number of clusters
$t_{gs} \rightarrow$ Average posting list in IISD

For that d matrix, total number of calculations;
$m = 5$, $X_d = (3+2+1+2+2) / 5 = 2$, $n_c = 3$, $t_{gs} = (1+3+1+1+1+0)/6 = 1.17 \sim= 1$

$5*2 + 3*2 + (5-3)*2*1 =$ **20**

7)

For the obtained S'' and S matrices above;

$C' = (S'')^T * S$

$$C' = \begin{bmatrix} 0.3300 & 0.3300 & 0 & 0.3300 & 0 & 0 \\ 0.1089 & 0.4389 & 0.1650 & 0.1089 & 0.1650 & 0 \\ 0 & 0.5000 & 0.5000 & 0 & 0 & 0 \\ 0.3300 & 0.3300 & 0 & 0.3300 & 0 & 0 \\ 0 & 0.1650 & 0 & 0 & 0.6600 & 0.1650 \\ 0 & 0 & 0 & 0 & 0.5000 & 0.5000 \end{bmatrix}$$

n_c' = 0.33+0.4389+0.5+0.33+0.66+0.5 = **2.7589**
n_c = **2.7589** (obtained in previous question) so they are the same.

Because, documents are group or clustered according to the their term similarities and if we create an inverted index, we can group terms according to documents that are contains them. So this is a bi directional relationship. Thus the cluster sizes from these different viewpoints (document or term) must be equal.

8)

$n_c = \frac{m*n}{t}$ $\frac{m}{t} = \frac{1}{X_d}$ $\frac{n}{t} = \frac{1}{t_g}$

$n_c = \frac{n}{X_d}$ $n_c = \frac{m}{t_g}$

Where;

X_d → Average number of terms per document
m → Total document count
n_c → Number of clusters
t_g → Average number of docs per term

For that D matrix;

m = 5, X_d = (3+2+1+2+2) / 5 = 2, tg = (1+3+1+1+3+1)/6 = 1.67

$n_c = \frac{6}{2}$ = **3** $n_c = \frac{5}{1.67}$ = 2.99 ~= **3**

So the cluster count is 3 with respect to indexing-clustering relationships. This result also matches with C^3M' s number of clusters result = **3**

By using clustering-indexing concept, we can practically calculate number of clusters without calculating C matrices and their diagonal element sums. This enables us to interpret distribution of documents according to similarities and if know document distribution of a set we can create more balanced sets (document count according to subjects). This property is very useful for

making algorithm test sets because for test sets, document counts according to their subjects should be nearly the same to evaluate IR algorithms better. If we know cluster count of test set we would have general idea about that set and we would use it or throw it for IR algorithm evaluation.

9)

[4] Cluster maintenance is a handling process for modification of clustering structures due to the addition of new documents or deletion of old documents or both. Most of the clustering algorithms are unsuitable for cluster maintenance. There are very few maintenance algorithms and they developed for growing databases. C^3M algorithm is one of the algorithms that are suitable for cluster maintenance. Incremental version of C^3M, which is C^2ICM, is suitable for cluster maintenance. It starts with m documents and cluster them using C^3M algorithm. After that clusters are updated due to newcomers and deletions autonomously. Since all clustering operations are done on single C matrix, new points would be added or existing points would be deleted. Outline C^2ICM of is as follows;

- Compute number of clusters and cluster seed powers of updated dataset.
- Determine the set of documents to be clustered and cluster them to cover seeds.
- If there were documents not covered by any seed, then group those together in a ragbag.
- Apply the above steps for each dataset update.

10)

a)

[5] When the large datasets are considered. Speed of clustering algorithms is very important. However calculation speed of an clustering algorithm is not sufficient alone because there are natural limits of computational complexity for example $m^2(\log m)^2$. So it is harder to improve computational complexity and these clustering algorithms take too much time. This problem would be solved by paralleling clustering algorithms. By parallelization whole clustering task is divided into small tasks and these tasks are handled by different machines or hardware's. Thus a large dataset would be clustered quickly owing to parallelization. Parallelization is also useful for the lack of memory. When the main memory is insufficient to keep whole dataset, clustering algorithm could not be used. Again by dividing dataset into small sets and handling them on different machines would be a solution.

b)

[5]Data mining is extraction process of novel, meaningful and valuable information from large datasets. It can be applied to relational, transaction and spatial databases, as well as large stores of unstructured data. Data mining, like clustering, is an exploratory activity, so clustering methods are suitable for data mining. Clustering is one of the important initial step of various data mining approaches. Because the nature of data mining finding self similar new patterns from large datasets and this operation describes clustering process. Some of the data mining approaches which use clustering are database segmentation, predictive modeling and visualization of large databases.

c)

I think there is two reasons for that paper, being the most frequently cited papers in computer science. First one is; this paper is an survey paper and it mentions various applications

of computer science such as image segmentation, information retrieval, data mining, object recognition. Second reason is; the paper mentions all of these applications on the basis of clustering. Clustering is very frequently used techniques in various subjects. Therefore it is logical that, for that paper, being one of the most cited papers in computer science.

d)

I selected “Distinctive Image Features from Scale-Invariant Keypoints (2003) by David G. Lowe “ it cited 3096 times. It presents a method for extracting distinctive invariant features from images, which can be used to perform reliable matching between different images of an object or scene. These features are scale and rotation invariant and they can be used in real time image matching applications. Because the success and robustness of method, this paper is very frequently cited and became a baseline for new papers.

11)

a)

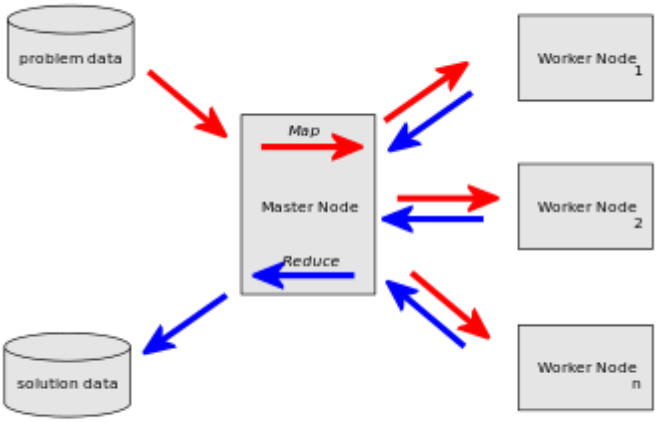
K-means is an multi-pass clustering algorithm and it uses squared error criterion. It starts with a random initial points (cluster centers) and assigns each points to its closest cluster center after that it recomputes the cluster centers using current cluster memberships. These procedures are repeated until reaching a specific iteration count or sum of square error value. It is a very popular algorithms because it is easy to implement and its time complexity is O(n). Major problem of that algorithm is, sensitivity of initial point selection.

b)

Apache Hadoop is a very popular implementation of map reduce. It divides single big problem into small sub problems and makes them solved in different machines merges their results and construct general solution. To implement K-Means on Hadoop; we follow these steps [6];

The dataset is divided into sub vectors, each vector represents a part of all dataset. Cluster centers are subvectors of these ones.

- In the map step
    - Read the cluster centers into memory from a sequence file
    - Iterate over each cluster center for each input key/value pair.
    - Measure the distances and save the nearest center which has the lowest distance to the vector
    - Write the cluster center with its vector to the file system.
- In the reduce step (we get associated vectors for each center)
    - Iterate over each value vector and calculate the average vector. (Sum each vector and divide each part by the number of vectors we received).
    - This is the new center, save it into a Sequence File.
    - Check the convergence between the cluster center that is stored in the key object and the new center.
    - If it they are not equal, increment an update counter
- Run this whole thing until nothing was updated anymore



12)

a)

m = 400 n_c = 20 k = 5

Assuming that each cluster has the same size, expected number of relevant documents;
Yao's formula;

$$= n.\left[1 - \prod_{i=1}^{k} \frac{m - \frac{m}{n} - i + 1}{m - i + 1}\right]$$

$$= 20.\left[1 - \left(\frac{(400-20-1+1)}{400-1+1}\right).\left(\frac{(400-20-2+1)}{400-2+1}\right).\left(\frac{(400-20-3+1)}{400-3+1}\right).\left(\frac{(400-20-4+1)}{400-4+1}\right).\left(\frac{(400-20-5+1)}{400-5+1}\right)\right]$$

= 20*0,2272 =4.5440 ~= **5 clusters to be accessed.**

b)

Similar paper for that; "Block Access Estimation for Clustered Data Using a Finite LRU Buffer (1993) By Fabio Grandi and Maria Rita Scalas from IEEE Transactions on Software Engineering"

References

[1] http://nlp.stanford.edu/IR-book/html/htmledition/clustering-in-information-retrieval-1.html

[2] http://www.dcs.gla.ac.uk/Keith/Chapter.3/Ch.3.html

[3] http://www.cse.msu.edu/~jain/Clustering_Jain_Dubes.pdf

[4] Refer to Can, F. Incremental Clustering for Dynamic Information Processing, ACM Trans. On Information Systems. Vol. 11, No. 2 (April 1993), pp. 143-164

[5] Jain, A. K., M., Murty, N., Flynn, P. J. Data clustering: A review. ACM Computing Surveys

[6] http://codingwiththomas.blogspot.com/2011/05/k-means-clustering-with-mapreduce.html

[7] Exploring the Similarity Space by Justin Zobel Ali Stair Moffat