H4016 Text Mining and Information Retrieval

Unit 7: Mining the Prepared Text Data: part I - Clustering

Ref:
1. Weiss et al. Text Mining: predictive methods for analysing unstructured information chpts 4, 5
2. Tan et al, Introduction to Data Mining, chpts 8 & 9

Context & Recap

This Unit is on steps 4 and 5, looking at clustering algorithms.

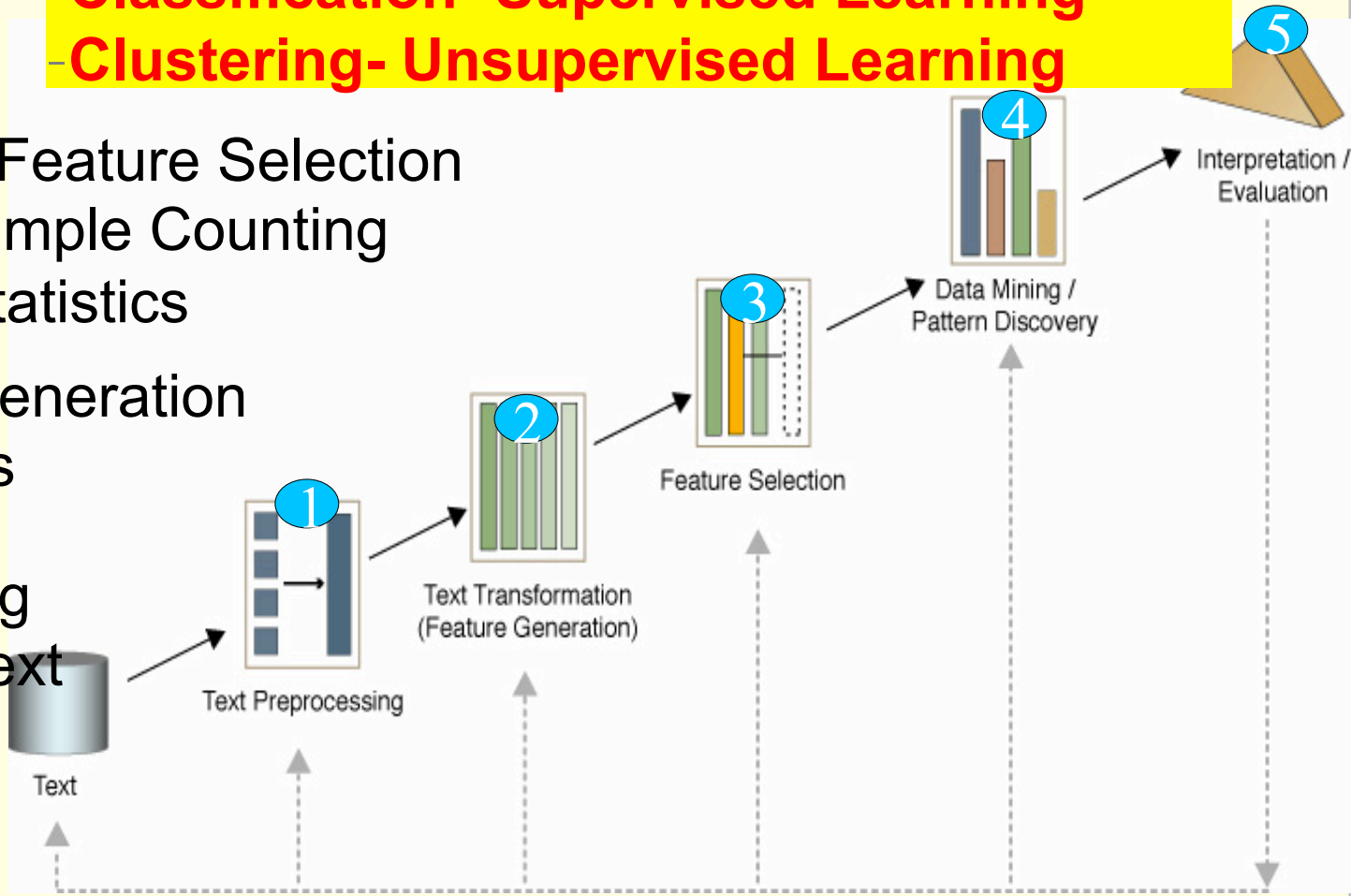
5. Analyzing Results

4. Text/Data Mining
- **Classification- Supervised Learning**
- **Clustering- Unsupervised Learning**

3. Feature Selection
- Simple Counting
- Statistics

2. Features Generation
- Bag of Words

1. Text Preprocessing
- Syntactic/Semantic Text Analysis



Overview

1. Similarity and Distance Measures
2. Clustering Algorithms
3. Objective and Subjective analysis of clusters

Properties

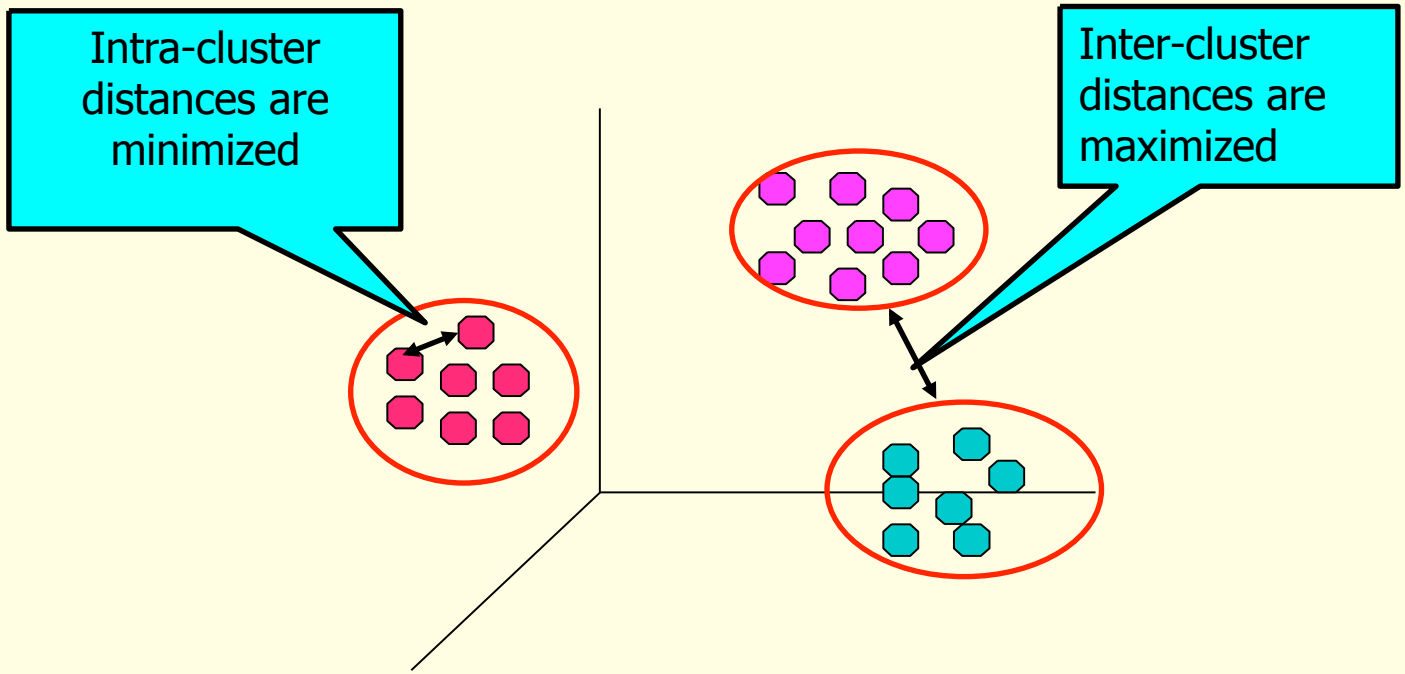
- Following on from data preparation (Units 2 & 3) we now have a document vector for each document in the data set:

Words	applications	binary	computer	computer system	engineering	eps	.....
D1	1	0	1	0	0	0	
D2	0	0	1	1	0	0	
D3	0	0	0	0	0	1	
D4	0	0	0	0	1	1	
D5	0	1	0	0	0	0	
D6	0	0	0	0	0	0	
D7	0	0	0	0	0	0	

- Data entries can be:
  - **Binary**
  - Three-values
  - **Occurrence** frequencies
  - **Term Frequencies**
  - IDF or **TFIDF**
  - Latent semantic indexing
- Note: the data set may include other columns as well.

Introduction to Clustering

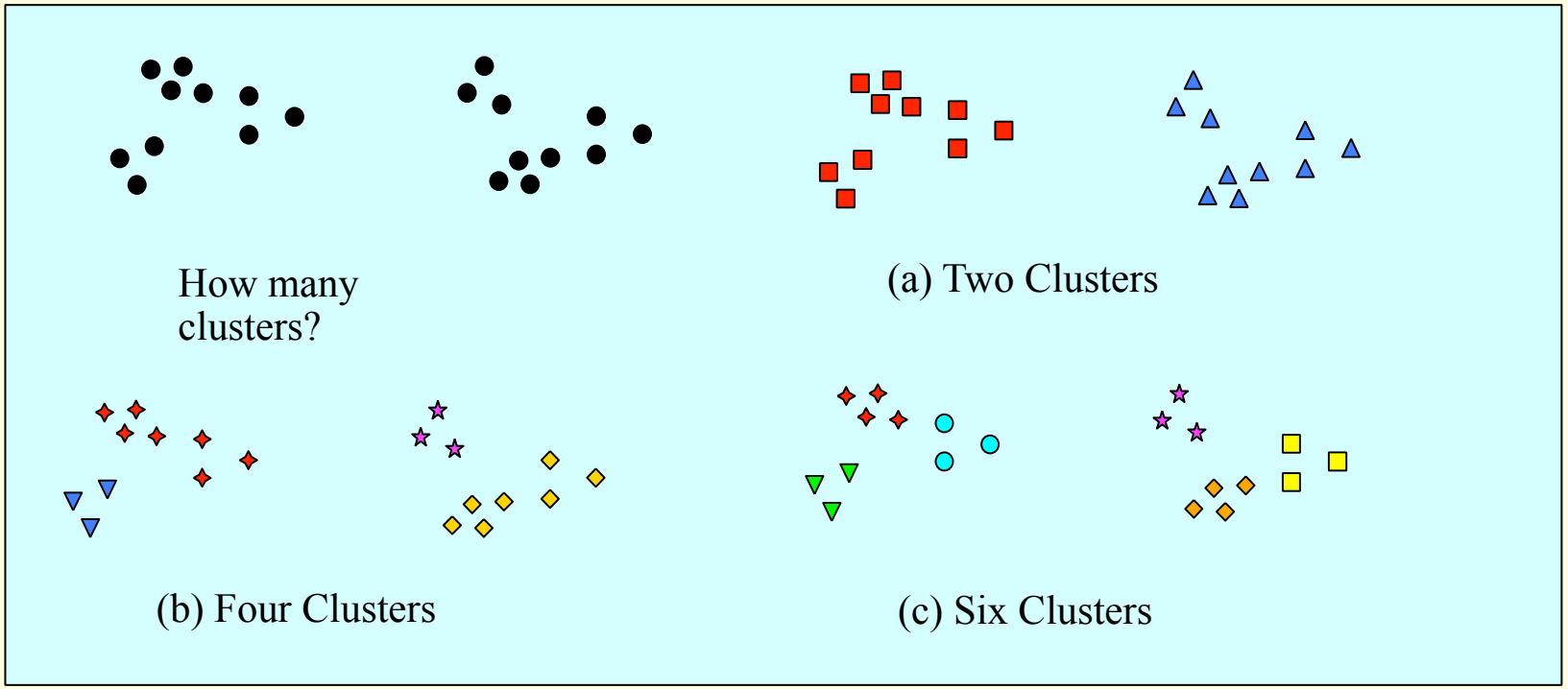
- The objective in clustering (or unsupervised learning) is to group objects (documents) such that objects within a cluster have a high similarity in comparison with one another, but are dissimilar to objects in other clusters.



- Clustering algorithms differ in how they compare objects to assess how similar they are.

Introduction to Clustering

- In many cases, the notion of a cluster is not well defined.
- The diagram below shows 20 points (document vectors), and three different ways of dividing them into clusters. The shapes of the markers indicate cluster membership.



Similarity & Distance Measures

Distance Measures

Key to the performance of a clustering algorithm is how distance is measured between rows of data.

Some terminology:

- **Similarity** between two objects is a numeric measure of the degree to which the two objects are alike.
  - Similarities are usually non-negative, and are generally in the range **[0,1]** where 0 is no similarity and 1 is complete similarity.
- Conversely, **dissimilarity** is a numerical measure of the degree to which objects are different.
  - Dissimilarities sometime fall in the interval [0,1] but more more commonly are measures in the range from **[0,∞]**, where 0 is no dissimilarity (i.e. the objects are the same).
- The term **Distance** is often used to mean dissimilarity.

Symmetric & Asymmetric attributes

Standard data set

	Salary	Married	No. Children	Home Owner	Rent / Mortgage	Age
Person A	70,000	Y	3	Y	900	50
Person B	40,000	Y	1	Y	800	35
Person C	37,000	N	0	N	300	28
Person D	32,000	N	0	N	400	26

- Which rows are similar?

- Is ‘number of children = 0’ relevant when comparing person C and D?

Symmetric & Asymmetric attributes

Document vector

	Hospital	Married	Children	Prison	Bail	Kofi
Doc A	10	3	3	0	0	0
Doc B	0	3	0	3	2	0
Doc C	0	0	0	7	3	0
Doc D	5	0	5	0	0	0

- Which documents are similar?

- Is (Kofi = 0) in all documents useful in determining similarities?
- Is (married = 0) in docs C and D useful in determining their similarity?

Symmetric & Asymmetric attributes

- In symmetric attributes, the absence of a value or a zero value is as significant as the presence of a non zero value.
- An asymmetric attribute is one where only the presence of a non-zero value is considered important. The absence of such a value is not important.

Words	applications	binary	computer	computer system	graph
D5	0	1	0	0	0
D6	0	0	0	0	1

- A symmetric measure would regard the documents D5 and D6 as having the same value in three attributes: applications, computer and computer system, i.e. 3 of the 5 attributes are the same.
- An asymmetric measure would regard documents D5 and D6 as having no attributes the same.

Distance Measures

- Many of the more popular distance measures are symmetric measures, and so are not suitable for document vectors, such as:
  - Euclidean distance
  - Manhattan distance

- For text mining, and mining high-dimensional, sparse datasets, measure need to be asymmetric. We will look at the following, **asymmetric similarity** measures:
  - Shared word count
  - Shared word count weighted by DF
  - Cosine Measure

Asymmetric similarity measures

1. **Shared Word Count**. The simplest asymmetric similarity measure is based on a count of the words that each document has in common.
   - This can be a count based on all words in the dictionary, or a count based on preselected words from the dictionary.
   - Preselecting words influences how the clusters will be formed.

- Take the following three sample document vectors:

	applications	binary	computer	graph
DocA	1	1	1	1
DocB	0	1	1	0
DocC	1	0	0	0

- similarity(docA, docB) = 2
- similarity(docA, docC) = 1
- similarity(docB, docC) = 0

This result is normalised by dividing the result by the number of terms (i.e. 4) giving:

Sim(A, B) = 0.5
Sim(A, C) = 0.25
Sim(B, C) = 0

Asymmetric similarity measures

1. Shared Word Count continued . . .

- Formula:

$$\text{Similarity} = \sum_{j=1}^{K} w(j)$$

$$w(j) = \begin{cases} 1 \text{ if term is in both documents} \\ 0 \text{ otherwise} \end{cases}$$

The result would be divided by K to normalise it.

where K is the number of dimensions (terms)

Advantages & Disadvantages:

- ✓ If the dictionary used has true key words, with weakly predictive words removed, then this method produces reasonably accurate results.
- ✓ Results are clearly intuitive
- ✓ Computationally easy
- ✓ Can use any type of document vector (binary, occurrences, frequency, IDF)
- ✗ Performance degrades with large dictionaries, and non specific words.

Asymmetric similarity measures

2. Shared Word Count Weighted by Document Frequency

- The last measure, shared word count, works best if weakly predictive terms are removed. However, it may be difficult to decide before hand which terms will be good for defining clusters.
- This measure weights the shared word count by how many documents that word occurs in, which improves performance for large dimensions.

Formula:
df = number of documents that term appears in

$$\text{Similarity} = \sum_{j=1}^{K} w(j)$$

$$w(j) = \begin{cases} 1 + \mathbf{1/df} & \text{if term is in both documents} \\ 0 & \text{otherwise} \end{cases}$$

As before, you count the number of words two documents have in common. An additional bonus is also added depending on the number of documents this term appears in. Preference is given to terms that appear in less document.

Asymmetric similarity measures

2. Shared Word Count Weighted by Document Frequency

- Example: Taking the same three documents again, the table below includes a document frequency for each term, i.e. how many documents each term appears in.

	applications	binary	computer	graph
DocA	1	1	1	1
DocB	0	1	1	0
DocC	1	0	0	0
. . . .	. . .	. . .	. . .	. . .
Total number of documents this term appears in:	2	40	35	10

- According to the table, the first term, *applications*, appears in just two documents; the second term, *binary,* appears in 40 documents. This changes the similarity measures as follows:
  - similarity(docA, docB) = (1+1/40) + (1 + 1/35) = 2.05
  - similarity(docA, docC) = (1 + 1/2) = 1.5
  - similarity(docB, docC) = 0

Asymmetric similarity measures

2. Shared Word Count Weighted by Document Frequency

- This formula puts the similarity of A and C much closer to the similarity of A and B, even though A and C have only one term in common. This is because
  - Documents A and B are linked by two weekly predictive terms - both terms appear in many documents.
  - Documents A and C are linked by potentially a strongly predictive term which only occurs in these two documents.

Advantages and disadvantages:

✓ Is more accurate than a simple word count

✓ Results are still clearly intuitive

✓ Still computationally inexpensive.

✗ Performance still not as good as the final, and most popular measure we will look at, the **Cosine Measure**.

Asymmetric similarity measures

3. Cosine Measure

- The cosine measure is based on a document vector which uses word frequencies rather than a binary entry for each term.

$$\cos(Doc1,Doc2) = \frac{\sum(w_{doc1}(i) * w_{doc2}(i))}{||x||\ ||y||}$$

Formula:

i = a term in the document vector

w(i) = tf(i) * log(N/n(i))

Same as IDF formula, Unit 3, making this **TFIDF**

where :

tf(i) = term frequency for *i*, i.e. term count / number of word in the document

N = total number of documents

n(i) is the number of documents containing term i.

- $||x|| = \sqrt{\sum w_{doc1}(i)^2}$
- $||y|| = \sqrt{\sum w_{doc2}(i)^2}$

Cosine similarity is in fact a measure of the cosine of the angle between the two document vectors

Cosine measures

Step 1: Calculate w(i) = tf(i) * log(N/n(i))

Wi (or TF-IDF)

	applications	*binary*	*computer*	*computer system*	*engineering*	*graph*	*human*	*interface*
W_i for Doc 1	1.16	0.05	0.00	1.38	0.00	0.41	0.00	0.00
W_i for Doc 2	1.16	0.00	0.00	0.00	0.00	0.62	0.00	1.66

Step 2: Calculation for **Top row:**

Wi(Doc1) * Wi(Doc2)	1.35	0	0	0	0	0.26	0	0	1.6
(calculated as follows:)	1.16*1.16	0.05 * 0	0*0	1.38*0	0*0	etc.			

∑(Wi(Doc1) * Wi(Doc2)) = 1.35 + 0.26 = 1.6

Step 3: Calculation for **Bottom row**

||x|| = √(1.16*1.16) + (0.05*0.05) + 0*0 + 1.38*1.38 + 0*0 + 0.14*0.14 + 0*0 + 0*0 = √3.44 = 1.85

similarly ||y|| = √4.49 = 2.12

Cosine measures

Step 4: Final Calculation:

$$\cos(Doc1,Doc2) = \frac{\sum(w_{doc1}(i) * w_{doc2}(i))}{||x|| \, ||y||}$$

cos(Doc1,Doc2) = 1.6 / (1.85 * 2.12)
= 0.41

Question: Is this a similarity or a distance measure?

All calculations:

	applications	binary	computer	computer system	engineering	graph	human	interface	*Total number of terms*
Occurrences Doc 1	3	2	0	5	0	2	0	0	*12*
Occurrences Doc 2	1	0	0	0	0	1	0	2	*4*
TF for Doc 1	0.25	0.17	0	0.42	0	0.17	0	0	*divide each count by 12*
TF for Doc 2	0.25	0	0	0	0	0.25	0	0.5	*divide each count by **4***
Number of documents this term appears in:	2	40	35	5	5	9	5	5	
Log (N/n_i)	4.64	0.32	0.51	3.32	3.32	2.47	3.32	3.32	
Wi (or TFIDF)									
W_i for Doc 1	1.16	0.05	0	1.38	0	0.41	0	0	
W_i for Doc 2	1.16	0	0	0	0	0.62	0	1.66	

Top Row:

Wi(Doc1) * Wi(Doc2)	1.35	0	0	0	0	0.26	0	0	1.6

Bottom row:

||x||	1.35	0	0	1.92	0	0.17	0	0	1.85
||y||	1.35	0	0	0	0	0.38	0	2.76	2.12

Cosine (Doc1, Doc2):	0.41

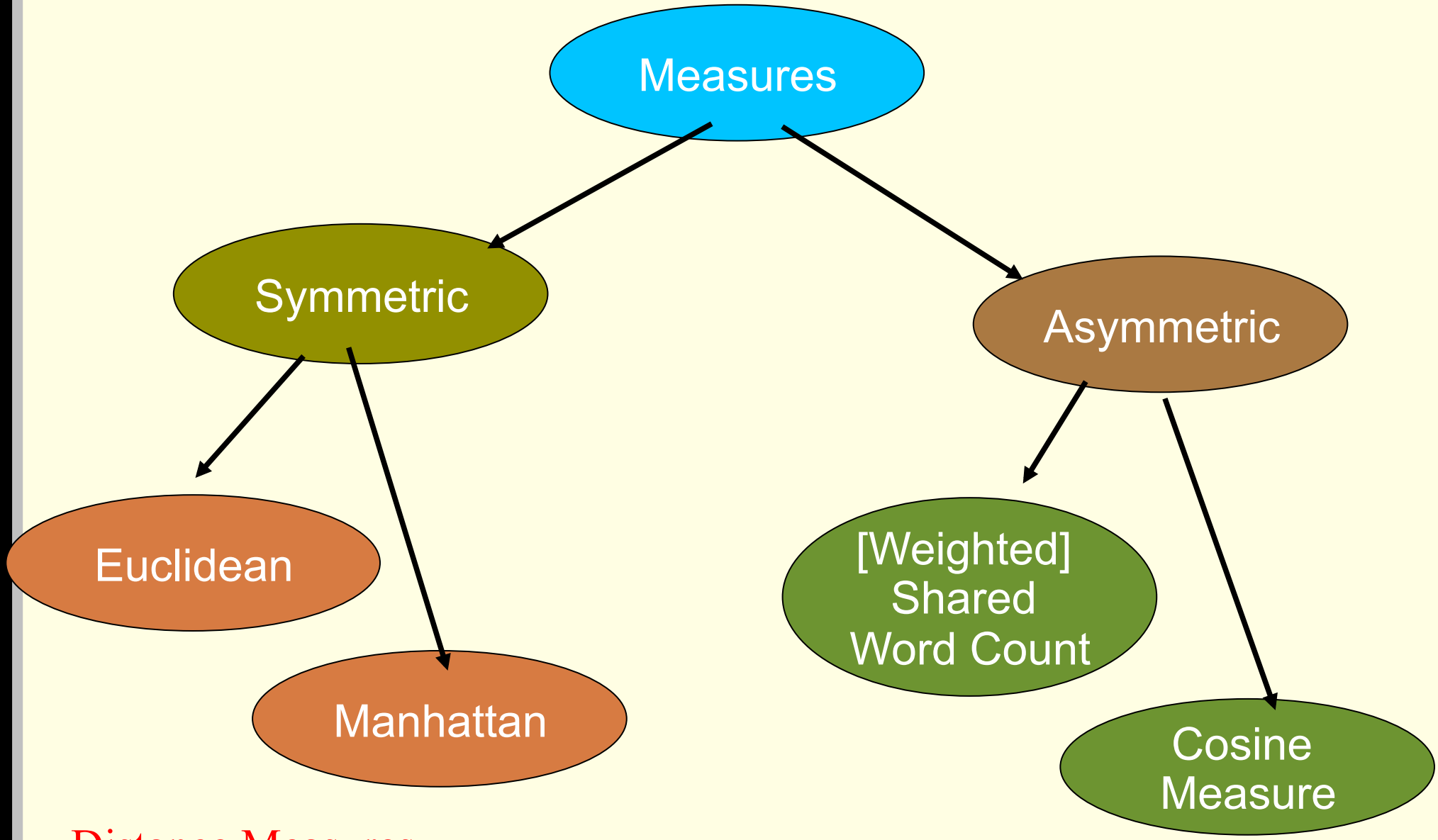
Asymmetric similarity measures

3. Cosine measure contd. . .

Advantages & Disadvantages

- ✓ The cosine measure is the default computation for document similarity, and is the benchmark against which other measures are tested.
- ✓ Is well tried and tested, with consistently good results.
- ✗ Result is not as intuitive.
- ✗ Values can cover a wide range.
- ✗ Computationally more expensive.

Summary of measures



Distance Measures

Similarity Measures

Past exam questions

Question 4. *Summer 2013*

a) Discuss the difference between *symmetric* and *asymmetric* variables, and the relevance of this when choosing the most appropriate similarity measures for clustering text documents. Illustrate your answer with reference to the three document vectors given below:

	Design	Police	Control	Salary
Document 1	1	1	1	0
Document 2	0	1	0	0
Document 3	0	0	0	1

(8 marks)

Marking scheme:
Define symmetric: 2 marks
Define asymmetric: 2 marks
Understand why docs need asymmetric, with reference to example given: 4 marks.

Clustering Algorithms

Clustering Algorithms

- There are a large variety of clustering algorithms.
- It is advisable to use several algorithms to see what the data might disclose.
- The three most common categories of clustering algorithms are as follows:
  1. **Partitioning methods**– you specify the number of clusters in advance, and an algorithm organises objects into the specified number of clusters. Each object is allocated to exactly one cluster (complete clustering), and so clusters are non-overlapping.
  2. **Hierarchical methods**– clusters are organized into a tree structure, where children represent the optimal slit of the parent group. Each parent is a union of its child sub-clusters. The Root node is all objects, (in this case objects are documents) and frequently each leaf nodes contains just one document. At any one level in the tree, each document is allocated to exactly one cluster, and so clusters are non-overlapping. As every document is allocated to exactly one cluster, it is also complete clustering.

Clustering Algorithms

3. **Grid-based methods**– organised objects into a grid structure such that clusters at opposite ends of the grid have the largest distance measurement. Those left in the centre do not naturally belong to any cluster (partial clustering). This is useful for identifying outliers, i.e. objects that do not easily fall into a cluster.

- We will now look at these in more detail. . .

1. K-Means (Partitioning method)

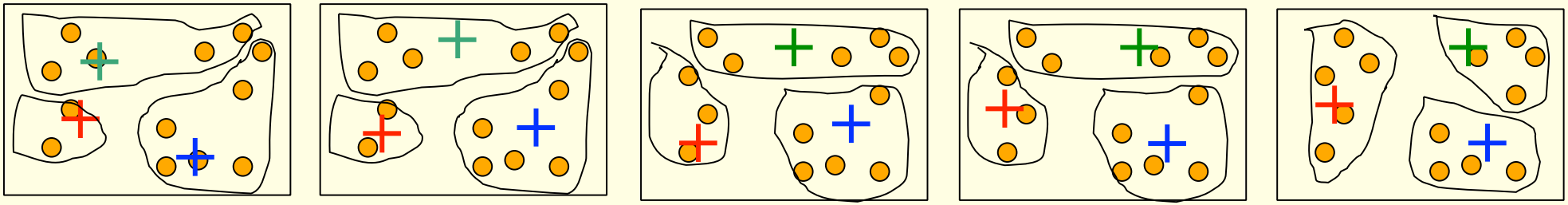
- Given a database of n tuples (rows), a partitioning method constructs k partitions of the database, such that $k \leq n$.
- Each group must contain at least one row of data, and each row of data must belong to exactly one group.
- **The number of groups, *k*, is specified in advance.**
- A good partitioning will result in objects from the same cluster being 'close' or related to each other, while objects from different clusters are "far apart" or very different.

- The most popular partitioning algorithm is the **k-means** algorithm which works as follows:

1. K-Means (Partitioning method)

1. *k* rows of the data set are selected at random, each of which initially represents a cluster mean or centre
2. For each remaining object, the object is assigned to the cluster to which it is most similar, based on the similarity between it and the cluster's mean.
3. Once all objects are allocated to a cluster, a new mean is calculated for each cluster.
4. All tuples in the database are now compared to this new mean, and again each object is allocated to the cluster to which it is most similar.
5. Once all objects a re-allocated, a new mean is again calculated for the cluster and the process is done again.
6. This process iterates until no document has moved to a new bin, or the distances between documents and their cluster centre converges.

1. K-Means (Partitioning method)



- Diagram 1: The first three rows representing the initial cluster centres (marked with green red and blue crosses respectively) are chosen at random. Three clusters are formed based on the proximity of other points to these three.
- Diagram 2: A new mean is calculated for each cluster, indicated by the new positions of the crosses in diagram 2 above.
- Diagram 3: Each point is now reallocated to a cluster based on its proximity to the new cluster centres. This changes the composition of each of the three clusters as shown.
- Diagram 4: The centre point of these new clusters is calculated, indicated by the new positions of the crosses in diagram 4 above.
- Diagram 5: Each point is again reallocated to a cluster based on its proximity to each new cluster centre. This again changes the composition of each of the three clusters.

1. K-Means (Partitioning method)

- See examples 1 & 2 in 'worked examples.pdf' for examples of the algorithm at work with the following five documents, and k set to 2.
- The entries in the data set represent actual term frequencies.

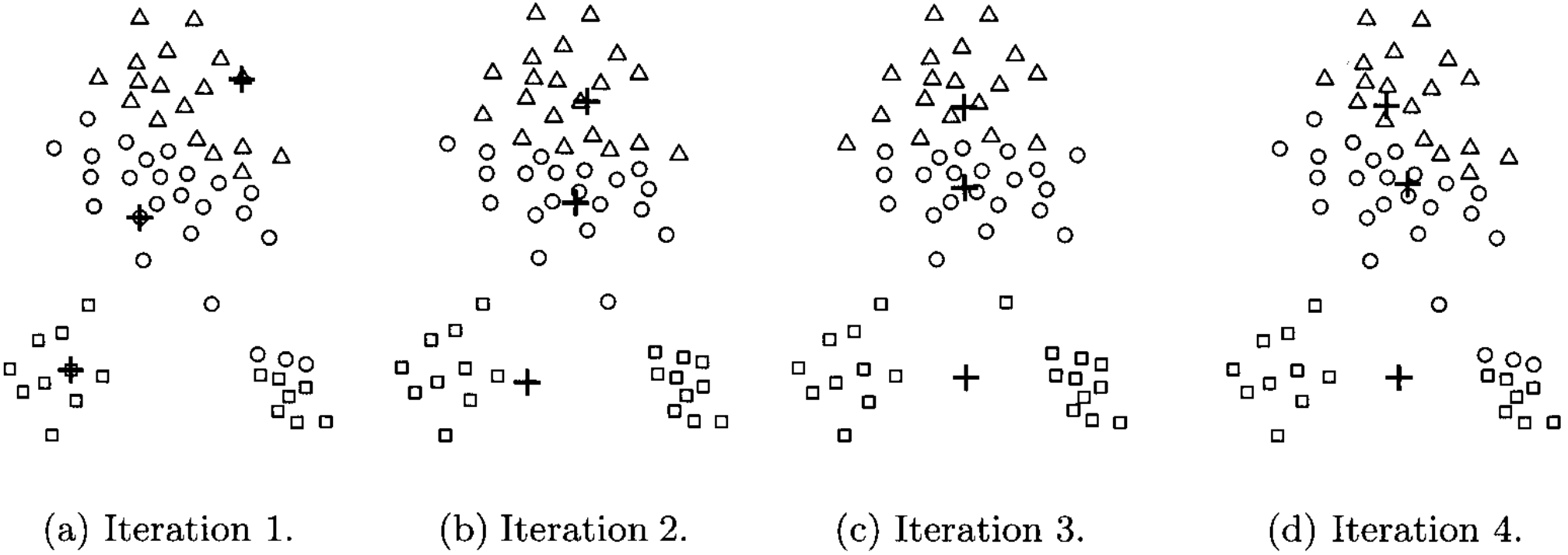
	applications	binary	computer	graph
Doc1	1	4	1	0
Doc2	2	2	2	0
Doc3	2	1	0	4
Doc4	3	1	1	3
Doc5	4	0	0	6

- In the first example, the initial two cluster centres are set to Doc1 and Doc5.
- In the second example, the initial two cluster centres are set to Doc1 and Doc2.

1. K-Means (Partitioning method)

- Strengths and Weaknesses
  - ✓ K-means is the most widely used method for statistical clustering.
  - ✓ It can be readily adapted to document vectors
  - ✓ It is computationally inexpensive
  - ✗ However you need to specify K in advance. The algorithm does not perform well if K does not match the actual number of clusters in the data set
  - ✗ It is also adversely effected by outliers, or documents that do not readily fit in a cluster.
  - ✗ It also performs poorly if clusters are not distinct (i.e. documents that could potentially belong to more than one cluster causes a problem.) (see illustrations on slide 39)
  - ✗ The outcome of the algorithm depends on the initial cluster centres chosen, which is random (see illustration, next slide)
    - Modifications to k-means include selecting initial cluster points that are distant from each other.

1. K-Means (Partitioning method)

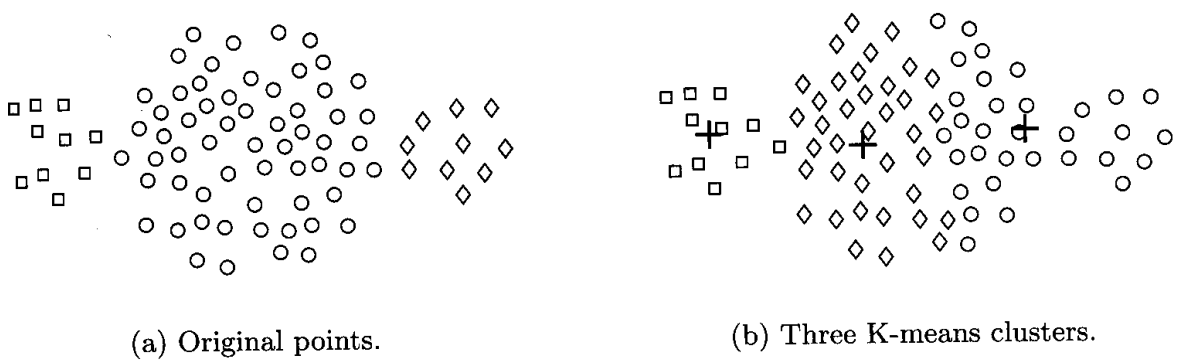


(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

Illustration of poor starting centroids (from Tan/Steinbach/Kumar 'Introduction to Data Mining', pg 503)

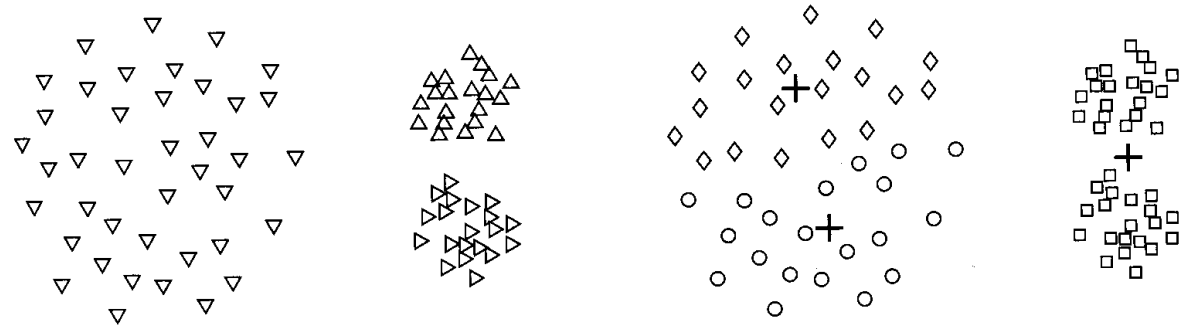
Limitations of K-Means

(from Tan/Steinbach/Kumar 'Introduction to Data Mining', pg 511)



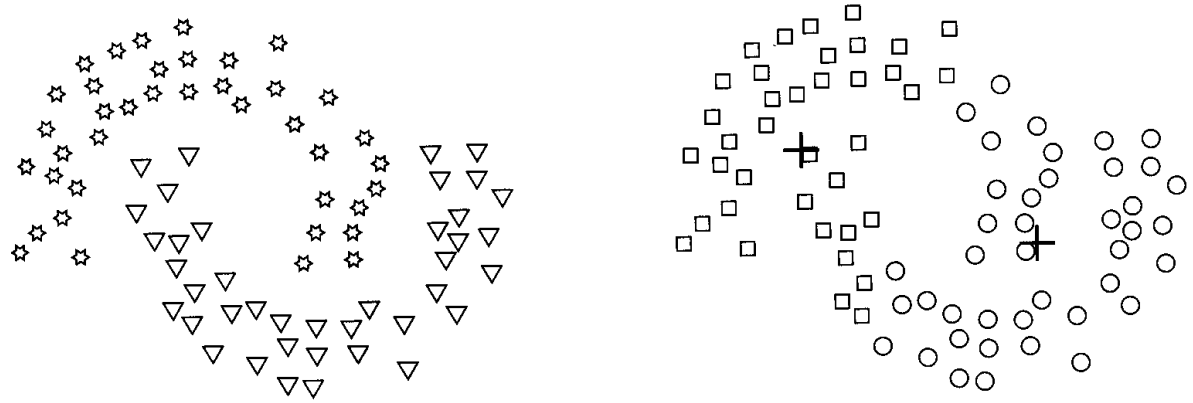
(a) Original points. (b) Three K-means clusters.

Figure 8.9. K-means with clusters of different size.



(a) Original points. (b) Three K-means clusters.

Figure 8.10. K-means with clusters of different density.

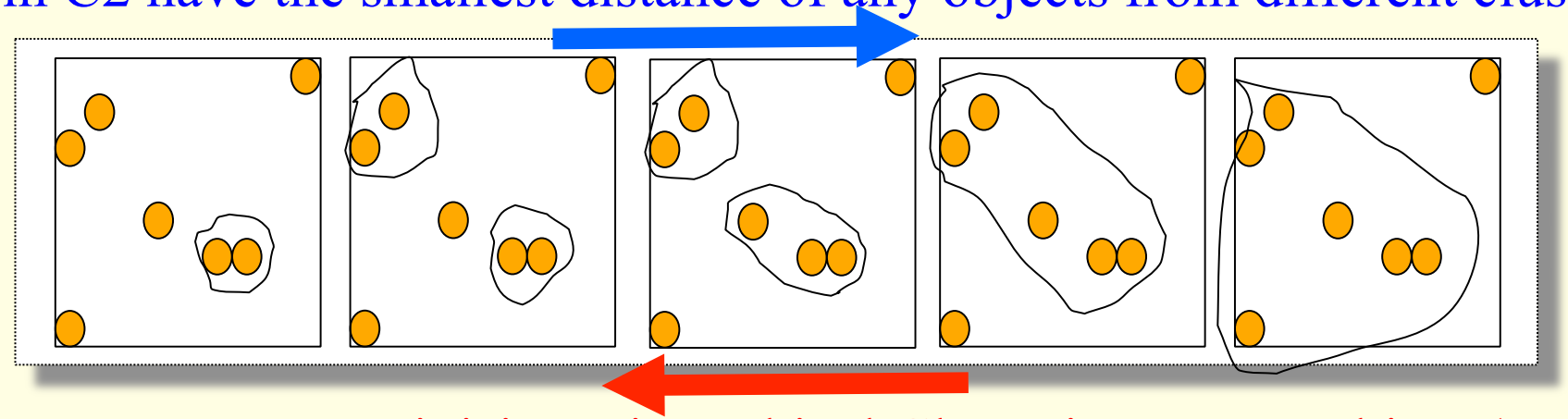


(a) Original points. (b) Two K-means clusters.

Figure 8.11. K-means with non-globular clusters.

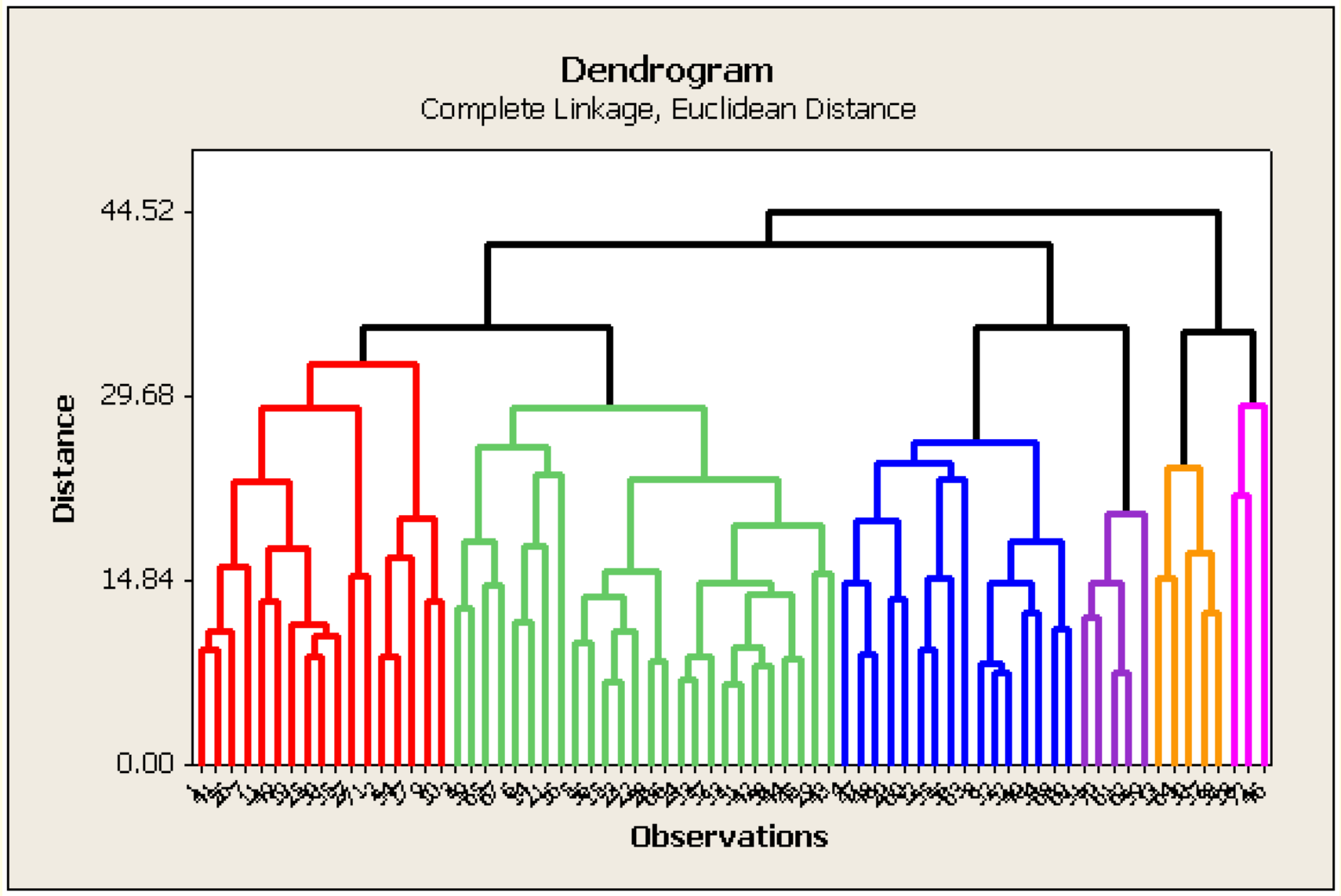
2. Hierarchical Clustering

- A hierarchical clustering method works by grouping data objects into a tree of clusters. There are two approaches:
  - Bottom Up - Agglomerative Hierarchical Clustering. Every object is given its own cluster to start with. Clusters are then merged until all objects are in a single cluster, or some terminating condition is met. Cluster C1 and cluster C2 are merged if an object in C1 and an object in C2 have the smallest distance of any objects from different clusters.



  - or Top Down - Divisive Hierarchical Clustering. Every object (or tuple) is put into one cluster to start with. Clusters are then subdivided into sub-clusters until every object has its own cluster, or some terminating condition is met. Clusters are generally split at the point where there is the maximum distance between two neighbours.

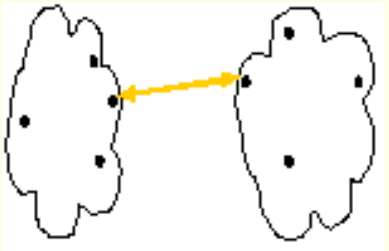
Output: dendogram



Merge / Split points

The method used to select merge or split points has the biggest impact on cluster definition. Methods commonly used include:

1. MIN / **single linkage**: the distance between two clusters is the distance between the closest points that are in different clusters. In other words, the similarity of the clusters is evaluated based on their closest points.

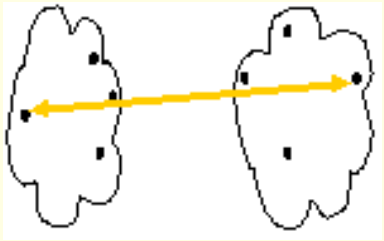


This is a Local Decision, which ignores the rest of the cluster.

It is a good method for non-elliptic shapes, however an outlier in a cluster can have an impact on how clusters are merged.

Merge / Split points

2. MAX / **complete linkage**. This distance between two clusters is the distance between the farthest points that are in different clusters. In other words, their similarity is measured in terms of their LEAST similar points.



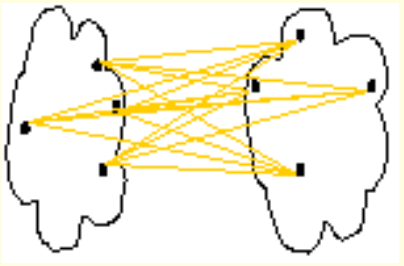
This is a NON-LOCAL decision, where the entire structure of the cluster can influence the decision.

a) This method preferences compact clusters with small diameters, and so is biased towards globular clusters.
b) Large clusters tend to be split.
c) A single point far from the centre may impact on how clusters are merged, making this sensitive to noise and outliers. (There is some disagreement on this point.)

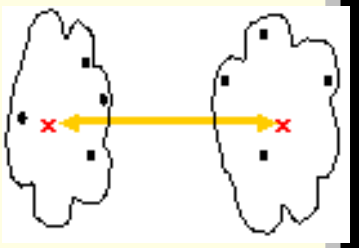
Globular clusters are clusters which are very roughly spherical or elliptical in shape (convex)

Merge / Split points

3. Average / **average linkage**. Cluster proximity is the average pair-wise proximities of all pairs of points from different clusters. This method is not as sensitive to outliers/noise, but is again biased towards globular clusters.

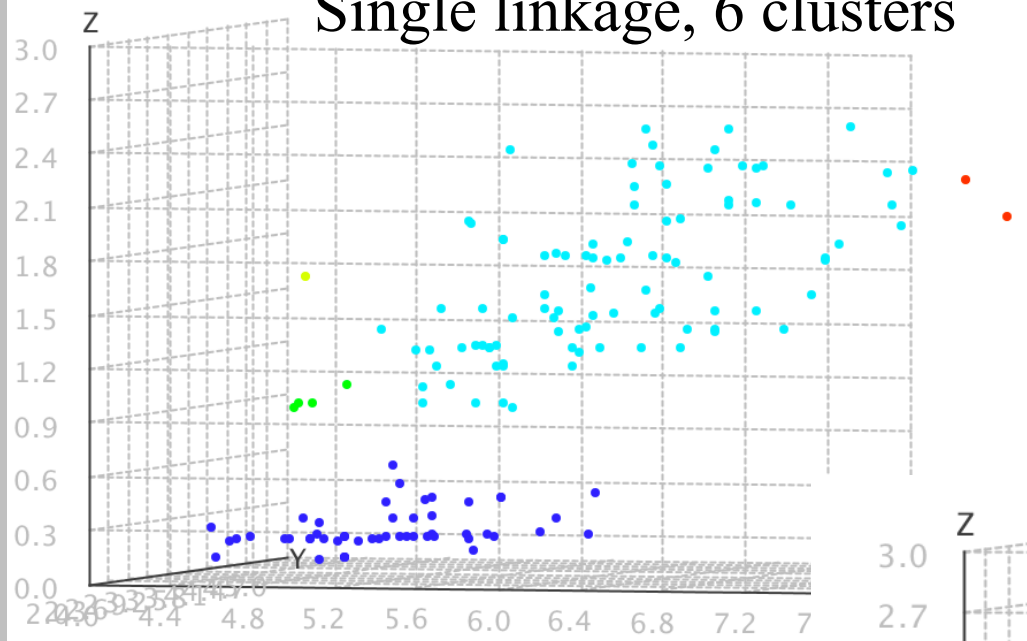


4. Mean distance / **centroid linkage**: Cluster proximity is the distance between the centroid of the two clusters, if the cluster has a centre. Like the average distance above, this method is not sensitive to outliers/noise, but is biased towards globular clusters.

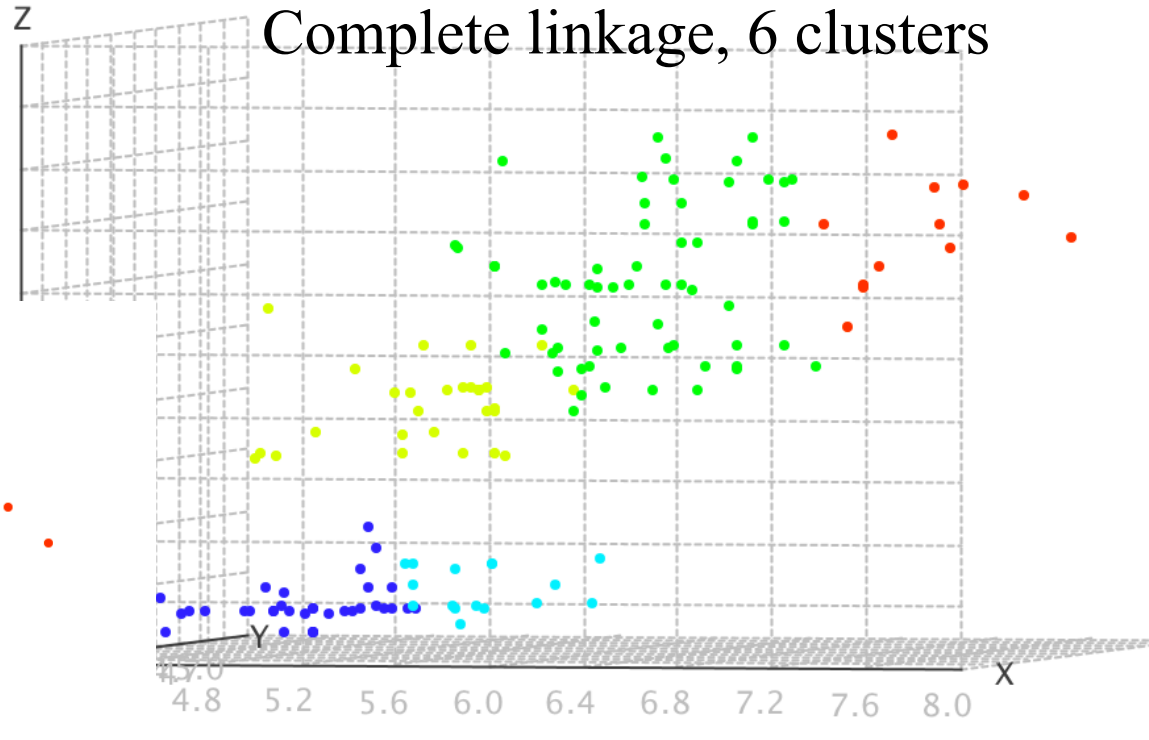


Single V Complete linkage (iris dataset)

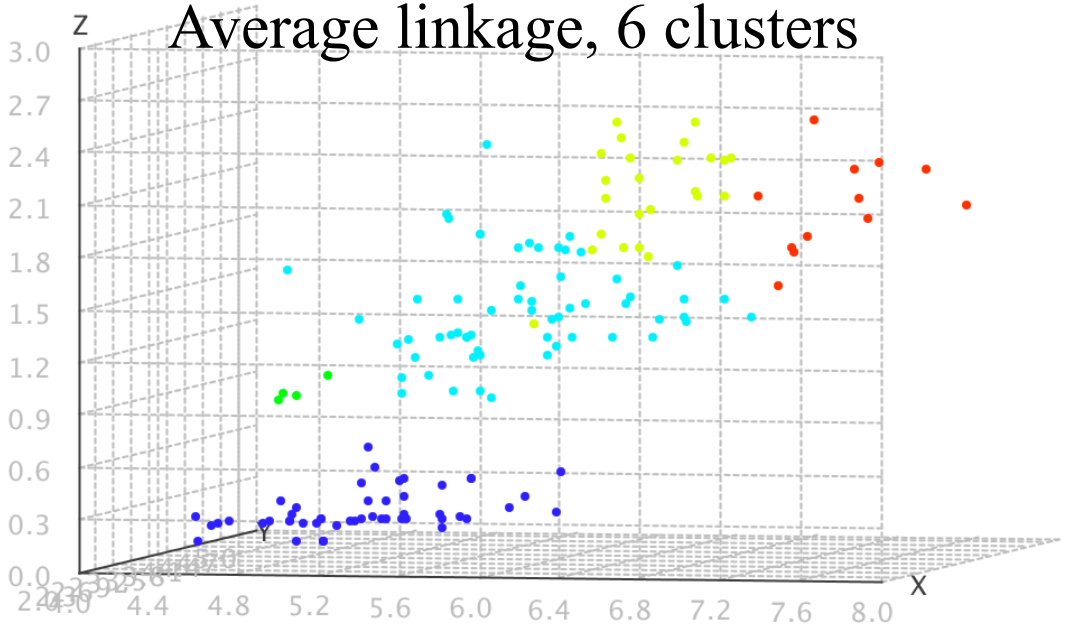
Single linkage, 6 clusters



Complete linkage, 6 clusters



Average linkage, 6 clusters



2. Hierarchical Clustering

- See example 3 in 'worked examples.pdf' which uses Agglomerative Hierarchical Clustering to cluster the following five documents.
  - The cosine measure is used to calculate the distance between documents
  - single linkage is used to pick the closest clusters.

	applications	binary	computer	graph
Doc1	1	4	1	0
Doc2	2	2	2	0
Doc3	2	1	0	4
Doc4	3	1	1	3
Doc5	4	0	0	6

2. Hierarchical Clustering

Strengths & Weakness.

- ✓ Studies suggest that the clusters generated by agglomerative hierarchical clustering are better than those generated by other methods. There are two issues that arise with this method though:
- ✗ a) Once a group of objects is merged/split that cannot be 'undone' in future iterations. This causes a problem when the data is very noisy, or high-dimensional.
- ✗ b) It is computationally expensive.

One approach to addressing these two issues is to partially cluster the data using another technique first.

View results of agglomerative clustering in Rapidminer, using both Complete linkage and Single linkage

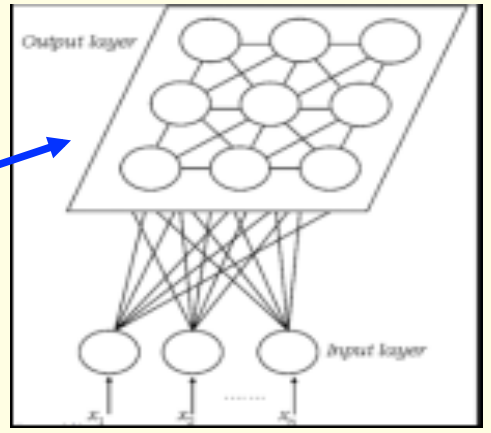
AgglomerativeClustering gives the full tree.
FlatAgglomerativeClustering returns a point on that tree which represents the number of clusters required

See: lab7-SingleComplete.rmp

3. Grid based methods

A grid based clustering approach uses a grid data structure. It is based on Neural Networks, and called a **Kohonen Net** or **Self Organising Map**.

Suppose you specify a 3 x 3, 2-D grid. This produces 9 output layer neurons, organised in a two-dimensional output grid of processing units.

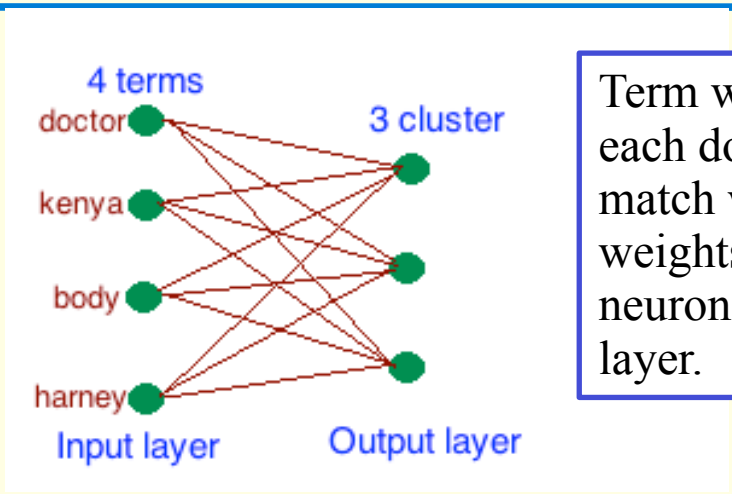


- During training, each unit competes with all of the others to "win" each row or document vector. When a unit wins a document vector, it's weights (along with those of other nearby units, collectively referred to as a **neighbourhood**) are adjusted to better match the pattern of predictor values for that record. As training proceeds, the weights on the grid units are adjusted so that they form a two-dimensional "map" of the clusters. (Hence the term **self-organizing map**.)

This results in the distance between clusters in neighbouring grids being relatively small, where as the distance between cluster at opposite ends of the grid table is relatively large

3. Grid based methods

Implementing a Neural Networks as a Self Organising Map:
There is an input neuron for each term in a document vector.
There is an cluster (output) neuron for each grid location.



Term weights from each document are match with the input weights for each neuron in the cluster layer.

Rows of data (documents) are read in one by one. For each document:

1. The cluster neuron whose weights are closest to the term frequencies for that document become the BMU (best matching unit) for that document.
2. Weights of the BMU are adjusted as follows:

$$w_j^{(k+1)} = w_j^{(k)} + \lambda(tf_i - w_j^{(k)})$$

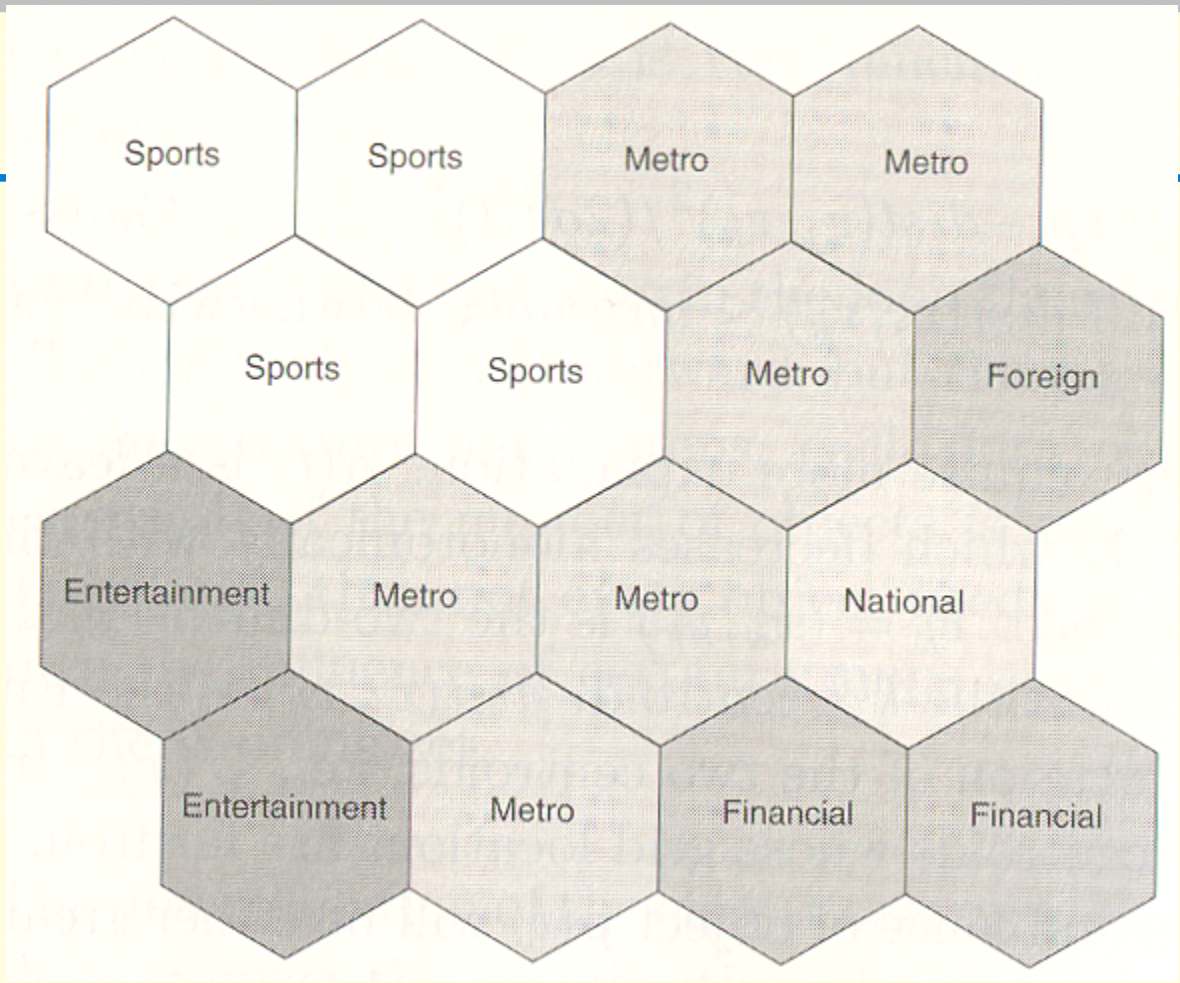
i.e. the new weight equals the old weight, plus a measure of the difference between the neurons weight and the term frequency for that row of data, multiplied by the learning rate λ. The learning weight is a value between 0 and 1, which influences the size of the change in weights. The closer to 1 it is, the larger the change in weights at each iteration.

3. Adjust weights of neighbouring neurons by a smaller amount.

3. Grid based methods

- In a self organising map, **λ** is adjusted separately for each output node relative to the distance between the current data point and the cluster centroid (the output neuron).
- If the distance between the current row, and a particular output neuron is **small**, the value of λ is **increased**, meaning this input will have a **significant** impact on how weights are adjusted;
- If the distance between the current row, and a particular output neuron is **large**, the value of λ is **decreased**, meaning this input will have an **insignificant** impact on how weights are adjusted;

- Initial weights can be set at random, or can be set to represent a cross section of examples from the data set.

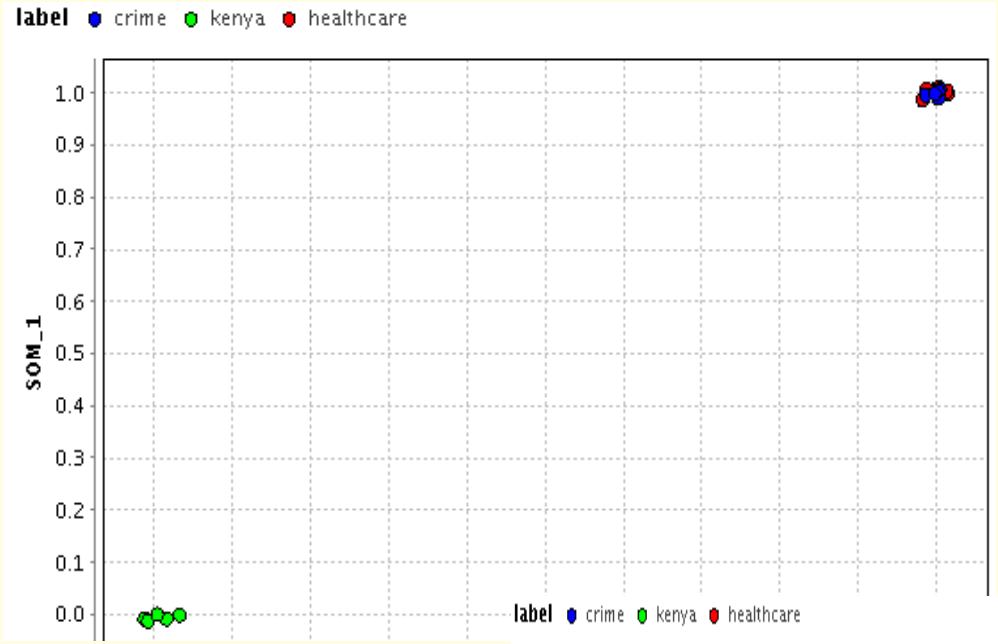


- The diagram above is the result of organising 3204 news paper articles from the *Los Angelos Times* using a 4 x 4 hexagonal grid, allowing each output neuron to have 6 neighbours rather than 4.
- Surprisingly, Sports and Finance articles had the least number of terms in common; Metro articles contained stories relevant to all other sections.

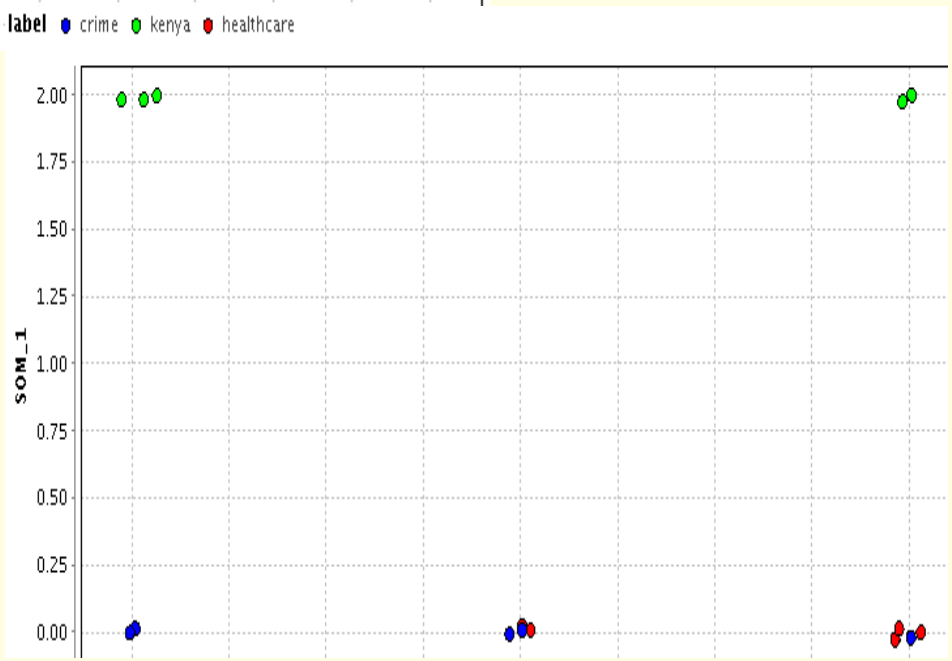
Ref: Tan et al, Introduction to Data Mining, pages 597, 598

SOM on the 15 lab docs

- Net size 2 x 2:



- Net size 3 x 3



This is a scatter plot of SOM_1 against SOM_0, with LABEL as the colour column

3. Grid based method

Strengths and Weaknesses of Kohonen model

- ✓ Excellent tool for visualising the relationships between clusters in the data set.
- ✓ Performance is comparable to other clustering algorithms
- ✓ Scalable to large volumes of data
- ✓ Good for identifying outliers
- ✘ Forces a topology on clusters, which may or may not be applicable
- ✘ As with all neural network algorithms, it works best with numeric data that has a good range of values.

4. Genetic Algorithms

- Genetic algorithms (GA) are an alternative approach to organising documents such that documents that are similar to each other are located close together.

- Genetic Algorithms, like neural networks, have their basis in nature, and are well suited to problems that have a huge number of possible solutions that **can not** be exhaustively searched.

Genetic Algorithms in general:

1. Possible solutions are represented as x-bit chromosome
2. Each solution is given a fitness score measuring it's merit as a solution
3. New solutions are found by combining two existing solutions, picking from those with the highest fitness score
4. Occasional mutation occurs by randomly changing a solution in some way.

4. Genetic Algorithms

GA's can be applied to clustering documents as follows:

Suppose we view the problem as follows:

Arrange all documents in a circle such that documents that are similar to each other are located close together.

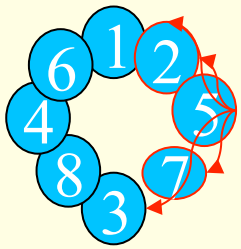
The huge number of possible solutions now becomes all possible permutations of how to arrange the documents in a circle. The x-bit chromosome is a list of documents in a specific order.

Steps in the algorithm:

1. Initially, a subset of possible solutions are generated by randomly arranging documents in a circle.
2. A fitness measure is calculated based on the similarity of each document to the documents at either side. It can be one document at either side, or a few documents at either side. If 'n' is the total number of documents, and 'r' is the number of documents to be considered as a close neighbour, the fitness measure is:

$$\sum_{i=1}^{n} \sum_{j=1}^{r} sim(i, i-j) + sim(i, i+j)$$

Fitness measure: $\sum^{n}_{i=1} \sum^{r}_{j=1}$ sim(i, i-j) + sim(i, i+j)



In this example illustrated to the left, n=8 as there are 8 documents in total. Lets set r=2. This means when calculating a fitness score:

$\sum^{n}_{i=1}$ means you start at the first document and work through all 8 documents

$\sum^{r}_{j=1}$ sim(i, i-j) + sim(i, i+j) means that for a particular document calculate it's similarity to the two documents on either side of it.

In other words, working your way around the circle, calculate the similarity of each document to its neighbouring documents, and add up all the similarity scores.

4. Genetic Algorithms

3. Two solutions are merged as follows:
   a. Pick two solutions that have a high similarity measure (fitness measure is above a set threshold).
   b. Pick any document that is common to both
   c. To the left of the document put the sequence from one parent.
   d. To the right of the document put the sequence from the other parent.
   e. Continue to do this until you get to a document that has already been added.
   f. Add any remaining documents randomly to the end of the child solution.

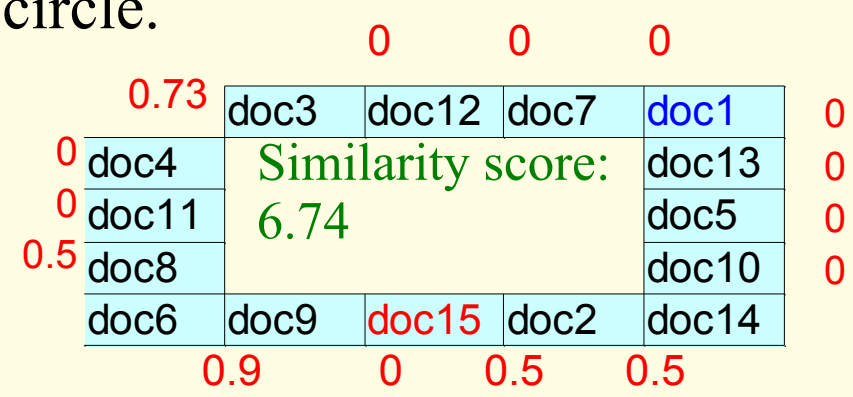
4. Mutation is implemented by swapping the position of two documents in the child node.

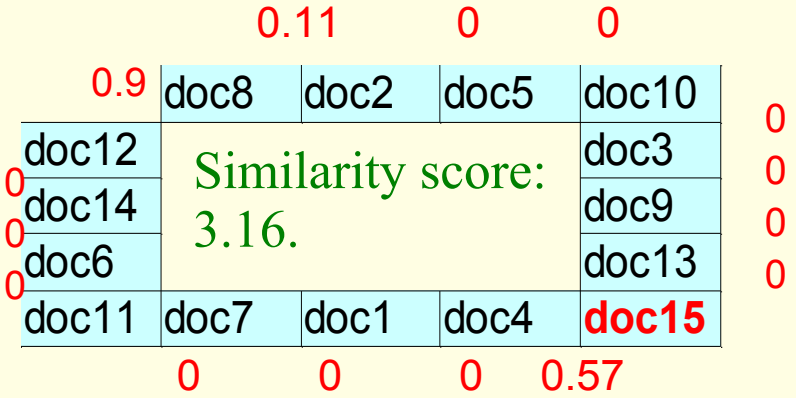
4. Genetic Algorithms

Worked example based on the 15 news extracts (healthcare, kenya, crime). Similarities are calculated using the cosine measure:

Step 1: Create a number of solutions by randomly ordering documents in a circle. Calculate a fitness score for each.

The diagram below shows two possible solutions. The distance measure between each document an its adjacent document is shown outside the circle.





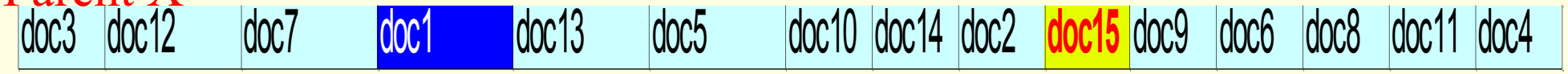
Step 2: Letting r=1, the fitness measure is calculated by adding up the similarity scores between each document and its neighbouring document (one before and one after). This means adding each measure above twice.

4 Genetic Algorithm

Step 3: Crossover: derive a new solution by combining two existing solutions.

Taking the same two solutions as on the previous slide, randomly pick a document common to both. In this case we pick doc15:

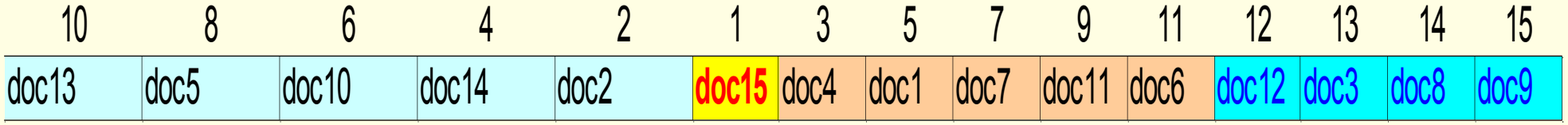
Parent X



Parent Y

Starting at document 15, alternatively add docs to the left and right. The left side comes from parent X, the right side comes from parent Y. Continue until you reach a document that is already added:

In this case, documents can be added until doc1 is reached on parent X, as this has already been added from parent Y. At this point, all remaining documents are added at random, giving:



The number above each document shows the order in which they were added.

4 Genetic Algorithm

- This new solution has a similarity measure of 4.42, which is an improvement on parent Y, but not on parent X. Generally the solution is accepted if it represents an improvement on the existing pool of solutions.
- While the effectiveness of this crossover method is dependant on the somewhat random position of the initial document chosen (i.e. doc15), it has been proven in research to work well.

- Step 4: Mutation is implemented by randomly swapping the position of two documents in a solution. For example, suppose we swap documents 7 and 14 in the solution to the last crossover. This would give a new similarity measure of 5.14, which is an improvement on the original solution:

Step 4: Mutation on above, randomly swop the position of two documents:

	doc13	doc5	doc10	doc7	doc2	doc15	doc4	doc1	doc14	doc11	doc6	doc12	doc3	doc8	doc12	simila
similarit	0	0	0.86	0	0.5	0	0.57	0	0	0	0	0	0	0.19	0.45	5.14

Research has shown that in text mining, mutation is as effective as cross over in generating improved solutions.

4 Genetic Algorithm

Strengths and Weaknesses of GA model

- ✓ Good tool for visualising the relationships between clusters in the data set.
- ✓ Scalable to large volumes of data
- ✘ Forces a circular ordering on clusters, which may or may not be applicable.

Evaluation of clustering algorithms

Exercise: Fill in table below

Algorithm	Scalable	Handles outliers /noise	Specify number of clusters in advance	Finds arbitrary shapes.
K-means				
Hierarchical				
Kohonen				
Genetic Algorithm				

Evaluating Clusters

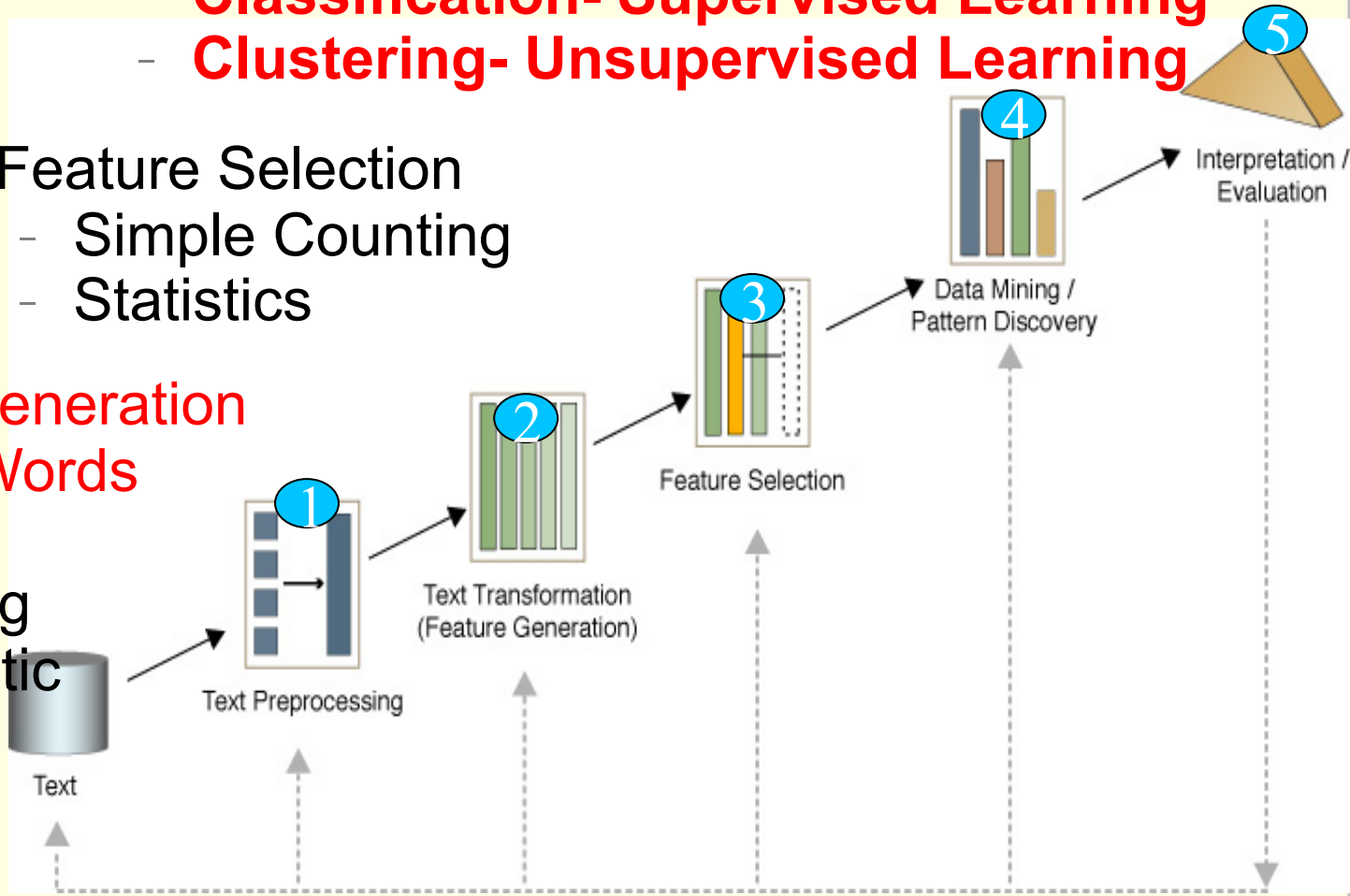
5. Analyzing Results

4. **Text/Data Mining**
   - **Classification- Supervised Learning**
   - **Clustering- Unsupervised Learning**

3. Feature Selection
   - Simple Counting
   - Statistics

2. Features Generation
   - Bag of Words

1. Text Preprocessing
   - Syntactic/Semantic Text Analysis

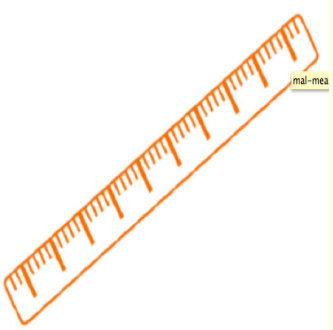


Cluster Evaluation

Cluster Evaluation can be

1. Subjective: Investigate the make up of each cluster, and make a judgement call on how well the data was segmented into clusters. This is time consuming, but benefits from the input of a domain expert, and so results will be relevant to the business objective.

2. Objective: use metrics to measure the performance of the clustering algorithm. Metrics would be based on inter- and intra- cluster distances, and will not take the business context into account.



Usually need to do both subjective and objective evaluation

Cluster Evaluation

- Apart from the algorithms themselves, the formation of a cluster can be influenced by:
  - The similarity measure used (Cosine, Euclidean etc.)
  - The linking method used (hierarchical: single, complete)
  - The indexing method used in the document vector (occurrences, term frequency, TF-IDF etc.)
- A change in any of these can impact the results of the algorithm.
- In addition, a number of algorithms have a random component which has an impact on the final results (initial clusters picked in k-means; initial weightings in Neural Network; initial set of solutions in a GA)

Subjective evaluation

- Subjective evaluation entails examining the results of a clustering algorithm to see what makes the most sense from a business context.
- Visualisation tools can help with this
- The following slides will look at the output from agglomerative clustering and SOM on the 15 documents from lab sheets.
- Before looking at the output from the clustering algorithms, we will first look at the dataset itself, and the similarity measures between the 15 documents.

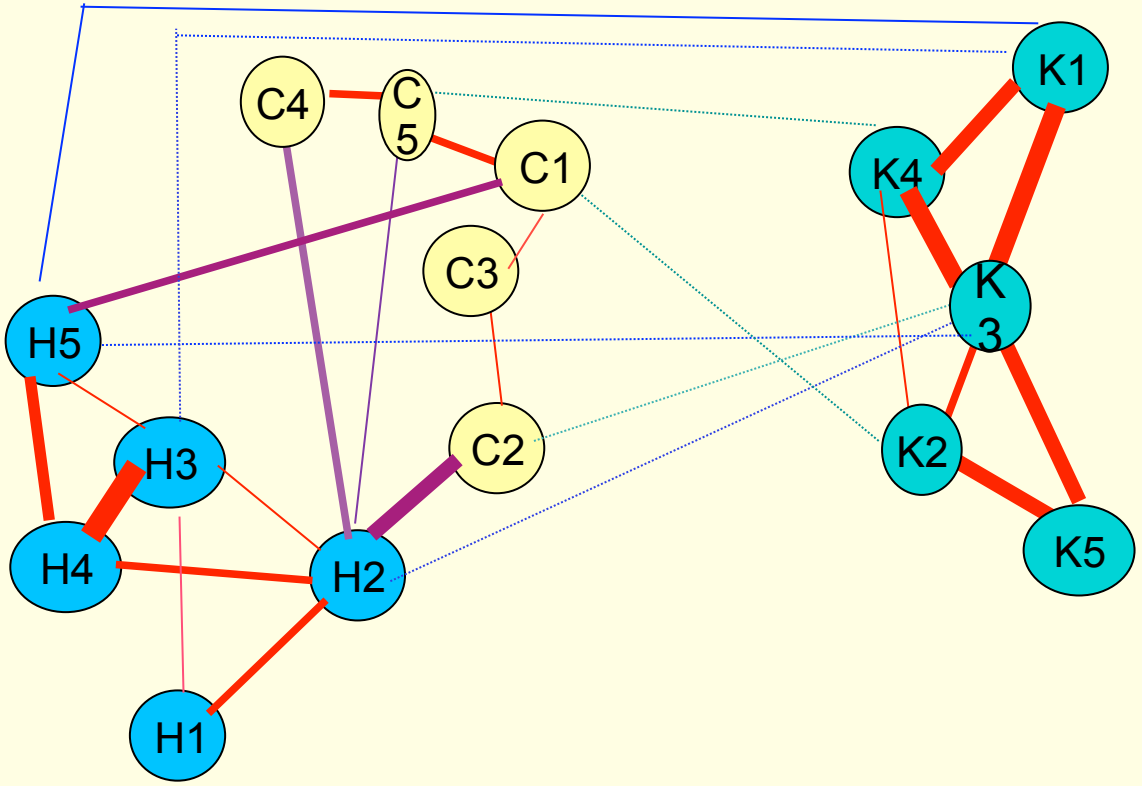
Similarity measures for 15 docs

SIMILARITY	Colum														
Row:Doc	C1	C2	C3	C4	C5	K1	K2	K3	K4	K5	H1	H2	H3	H4	H5
C1	1.0	.1	.2		.4		.1								.3
C2	.1	1.0	.2	.2				.1				.6		.1	
C3	.2	.2	1.0										.2		
C4		.2		1.0	.4							.3			
C5	.4			.4	1.0				.1			.2			
K1						1.0		.6	.4				.1		
K2	.1						1.0	.3	.2	.6					
K3		.1				.6	.3	1.0	.6	.4		.1			.2
K4					.1	.4	.2	.6	1.0						
K5							.6	.4		1.0					
H1											1.0	.3	.2		
H2		.6		.3	.2			.1			.3	1.0	.3	.2	
H3			.2			.1					.2	.3	1.0	.7	.1
H4		.1										.2	.7	1.0	.4
H5	.3							.2					.1	.4	
(blank)															
Grand Total	2.1	2.3	1.5	1.8	2.1	2.1	2.2	3.2	2.3	2.1	1.5	2.9	2.5	2.4	1.0

In RM, use Data to Similarity Data & Pivot operators

See lab7-SimilarityMatrix

Visualising the similarity measures



Similarity ≤ 0.2

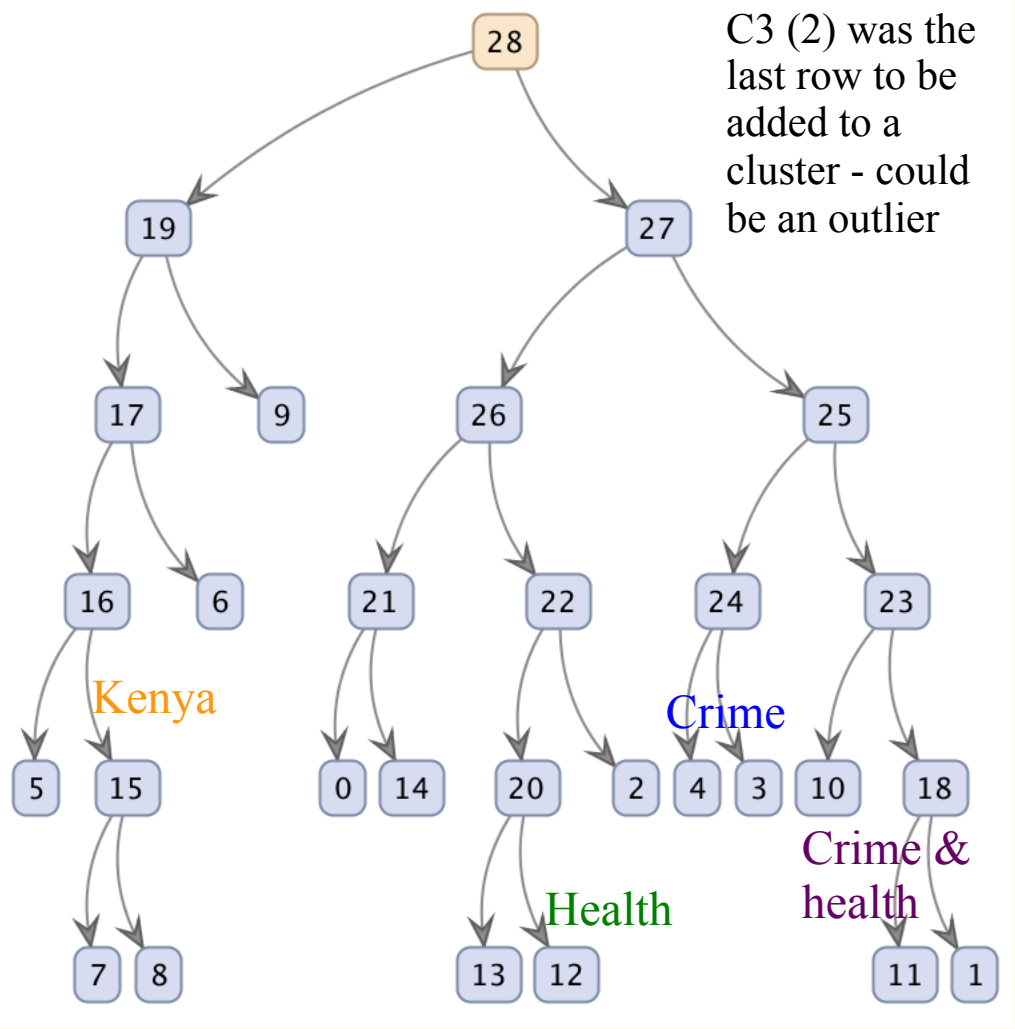
Similarity > 0.2 and < 0.5

Similarity > 0.5

How many clusters are the dataset?

Tree from Hierarchical Clustering

Look at output from agglomerative clustering using cosine similarity



C3 (2) was the last row to be added to a cluster - could be an outlier

0	C1
1	C2
2	C3
3	C4
4	C5
5	K1
6	K2
7	K3
8	K4
9	K5
10	H1
11	H2
12	H3
13	H4
14	H5

Number on leaf node matches the document ID. See table for corresponding document.

Number on parent nodes indicates the order in which merging was done, starting with node 15 which merged H3 with H4

Sampling the Hierarchical tree using flatten clustering

Run lab7-clustering-flatten agglomerative.rmp with different values for number of clusters in 'flatten cluster' operator with averageLinkage:

2 clusters

label	m...	cluster
crime	C1.	cluster_0
crime	C2.	cluster_0
crime	C3.	cluster_0
crime	C4.	cluster_0
crime	C5.	cluster_0
healthcare	H1	cluster_0
healthcare	H2	cluster_0
healthcare	H3	cluster_0
healthcare	H4	cluster_0
healthcare	H5	cluster_0
kenya	K1.	cluster_1
kenya	K2.	cluster_1
kenya	K3.	cluster_1
kenya	K4.	cluster_1
kenya	K5.	cluster_1

3 clusters

label	m...	cluster
crime	C1.	cluster_0
crime	C2.	cluster_0
crime	C4.	cluster_0
crime	C5.	cluster_0
healthcare	H1.	cluster_0
healthcare	H2.	cluster_0
healthcare	H3.	cluster_0
healthcare	H4.	cluster_0
healthcare	H5.	cluster_0
kenya	K1.	cluster_1
kenya	K2.	cluster_1
kenya	K3.	cluster_1
kenya	K4.	cluster_1
kenya	K5.	cluster_1
crime	C3.	cluster_2

4 clusters

label	m...	cluster
crime	C2.	cluster_0
healthcare	H1.	cluster_0
healthcare	H2.	cluster_0
crime	C1.	cluster_1
crime	C4.	cluster_1
crime	C5.	cluster_1
healthcare	H3.	cluster_1
healthcare	H4.	cluster_1
healthcare	H5.	cluster_1
kenya	K1.	cluster_2
kenya	K2.	cluster_2
kenya	K3.	cluster_2
kenya	K4.	cluster_2
kenya	K5.	cluster_2
crime	C3.	cluster_3

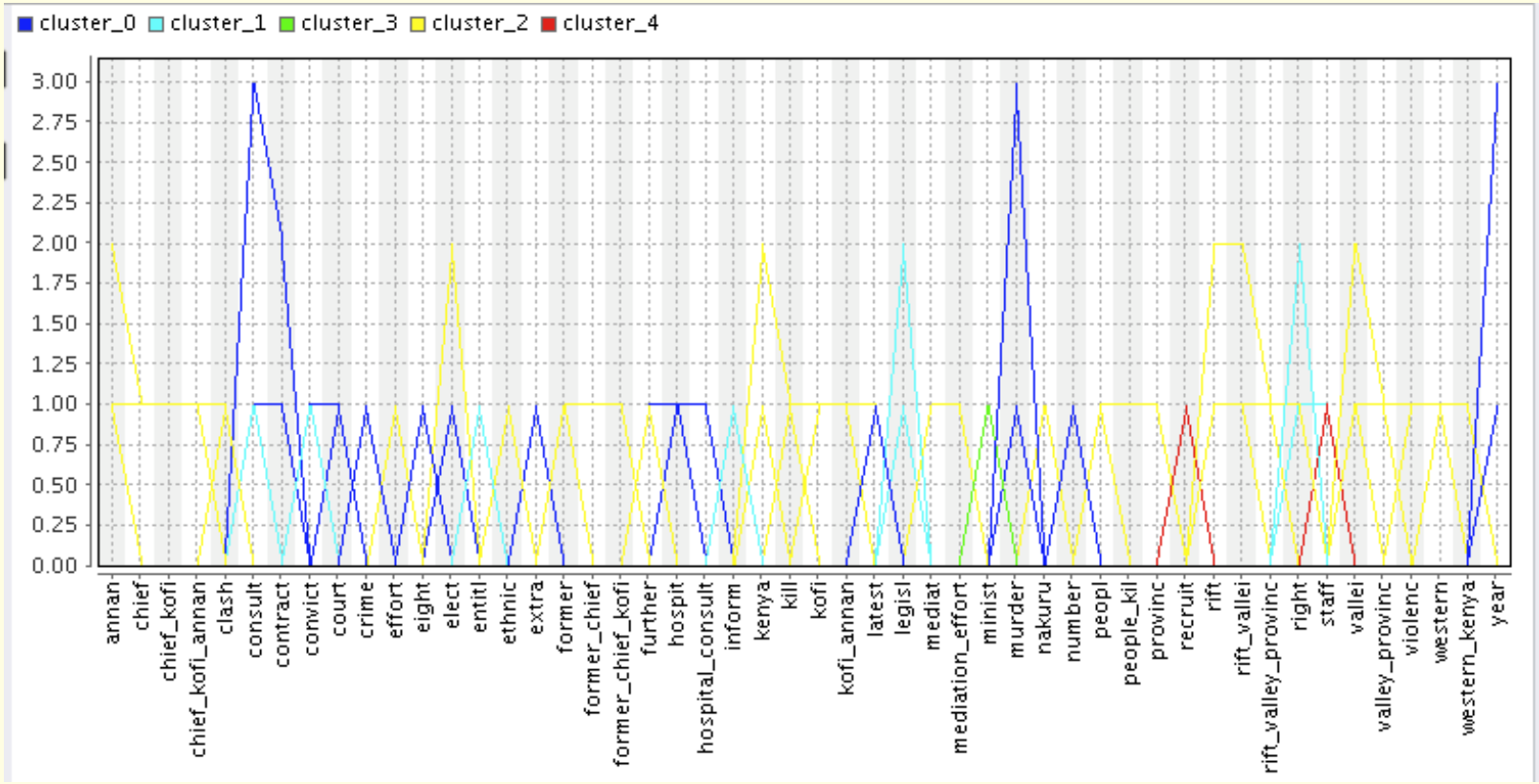
5 clusters

label	m...	cluster
crime	C1.	cluster_0
crime	C4.	cluster_0
crime	C5.	cluster_0
healthcare	H3.	cluster_0
healthcare	H4.	cluster_0
healthcare	H5.	cluster_0
crime	C2.	cluster_1
healthcare	H2.	cluster_1
kenya	K1.	cluster_2
kenya	K2.	cluster_2
kenya	K3.	cluster_2
kenya	K4.	cluster_2
kenya	K5.	cluster_2
crime	C3.	cluster_3
healthcare	H1.	cluster_4

SingleLinkage, and a low number of clusters helps to identify outliers / problem documents.

Using a parallel plot to intrepret cluster membership

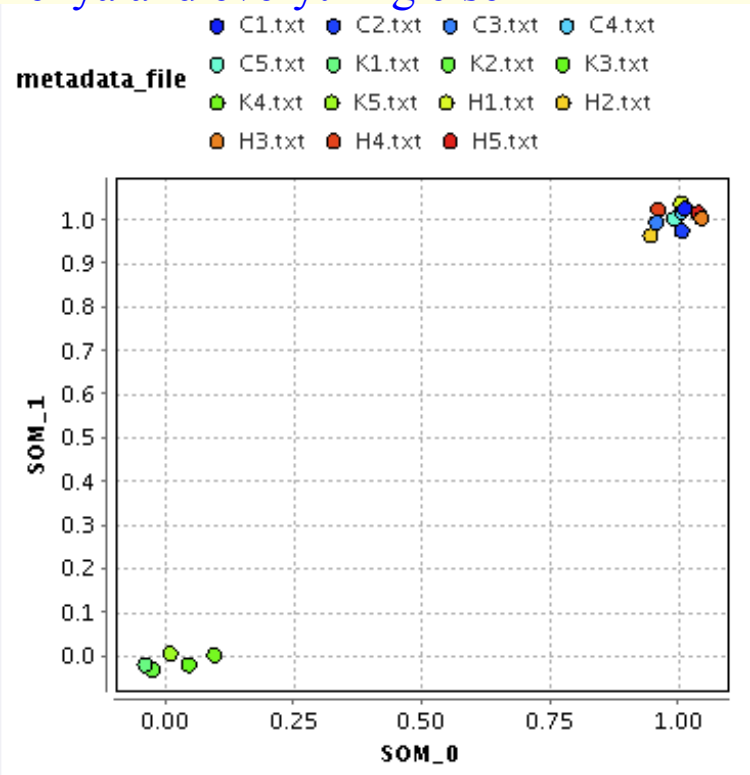
- With a larger dataset, visualisation tools can be used to investigate cluster membership. One such plot is a parallel plot, which illustrates the terms that appear in each cluster. Vertical lines represent the counts for each term. The coloured lines in the plot represent which terms occur in each of the four clusters below.



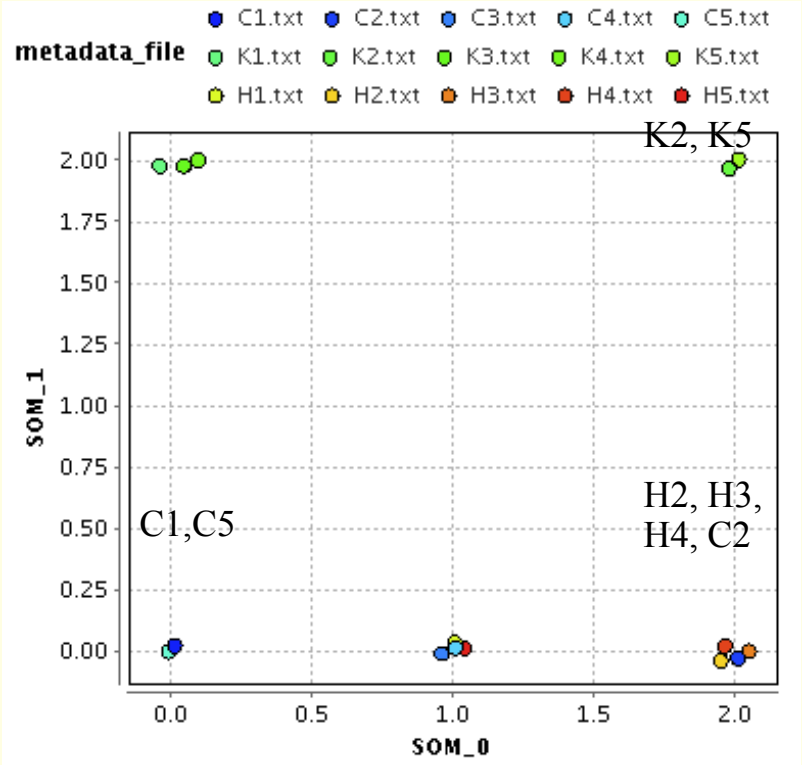
Subjective analysis: Output from a SOM – using a Neural Network

Run clustering-SOM.rmp with different grid sizes. Look at the dataset in the plot view, plotting SOM1 with SOM2, and colour coding by metadata_file:

A 2x2 grid identified two clusters, Kenya and everything else

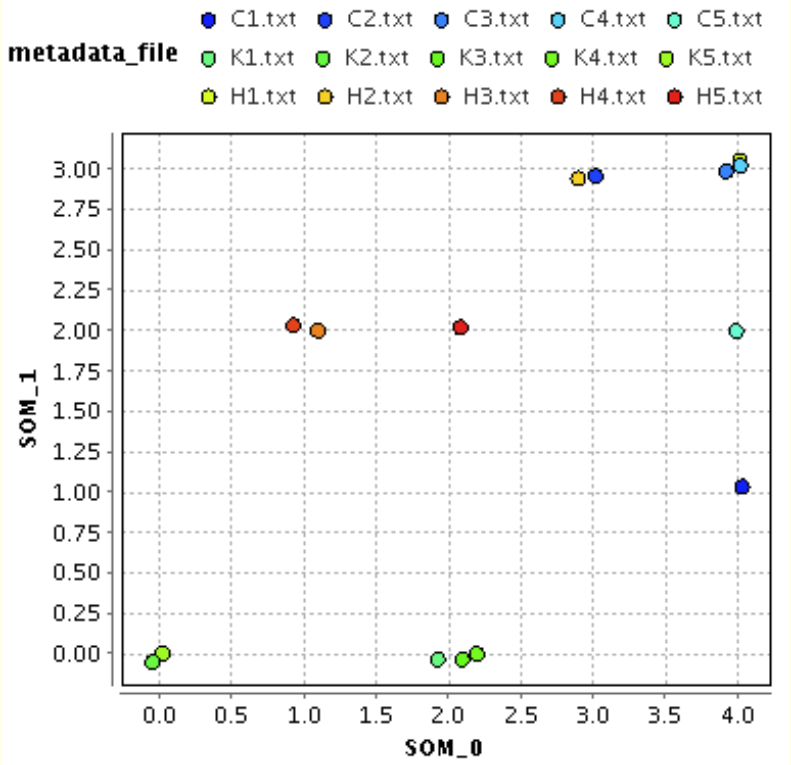


A 3x3 grid identified four clusters



A bigger grid can give more useful results in terms of neighbouring subgroups . . .

A 5x5 grid identified five clusters and some docs with weak connections



Still not as robust / dependable as agglomerative clustering

Exercise: Does attribute pruning improve performance?

prune below absolute	3
prune above absolute	5

Cluster Evaluation - Objective Measures

Clustering algorithms attempt to minimise intra-cluster distance, and maximise inter-cluster distances.

Measures to evaluate clusters generally involve calculating a ratio between these two distances.

We will look at two such ratios:

- Dunn index
- Davies-Bouldin index

Dunn Index

The Dunn index defines the ratio between the smallest inter-cluster distance – d_{min}, and the largest intra-cluster distance - d_{max}, and is defined as follows:

$$D = \frac{d_{min}}{d_{max}}$$

i.e. d_{min} is the smallest distance between two objects from DIFFERENT clusters, and d_{max} is the largest distance between two objects from the SAME cluster.

The result will be a number in the interval [0,∞]. The larger the number, the better the cluster definition.

This is a simplest formula, but sensitive to noise.

Dunn Index

- Applying this formula to our 15 documents:
  - Note: to do this accurately, you would need to use latent semantic indexing to avoid the number of similarity scores = 0 for rows in the same cluster.
  - Using a distance measure of 1-similarity, cosine similarity gives:
  - For k = 2
    - Largest intra cluster distance: K1 to K5. Distance = 1-0 = 1
    - Smallest inter cluster distance: H5 to K3. Distance = 1-0.2 = 0.8
    - Dunn index = 0.8 / 1 = **0.8**
  - For k = 3
    - Largest intra cluster distance: K1 to K5. Distance = 1-0 = 1
    - Smallest inter cluster distance: C3 to H3. Distance = 1-0.2 = 0.8
    - Dunn index = 0.8 / 1 = 0.8
  - For k = 4
    - Largest intra cluster distance: K1 to K5. Distance = 1-0 = 1
    - Smallest inter cluster distance: H3 to H3. Distance = 1-0.3 = 0.7
    - Dunn index = 0.7 / 1 = 0.7

Davies Bouldin Index

The formula for Davies-Bouldin is as follows:

$$DB = \frac{1}{n} \sum_{i=1, i \neq j}^{n} max \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right)$$

where:
n is the number of clusters,

δ_i is the average distance of all objects to the centre of their cluster i,

δ_j is the average distance of all objects to the centre of their cluster j, and
$d(c_i, c_j)$ is the distance between the cluster centres c_i and c_j.

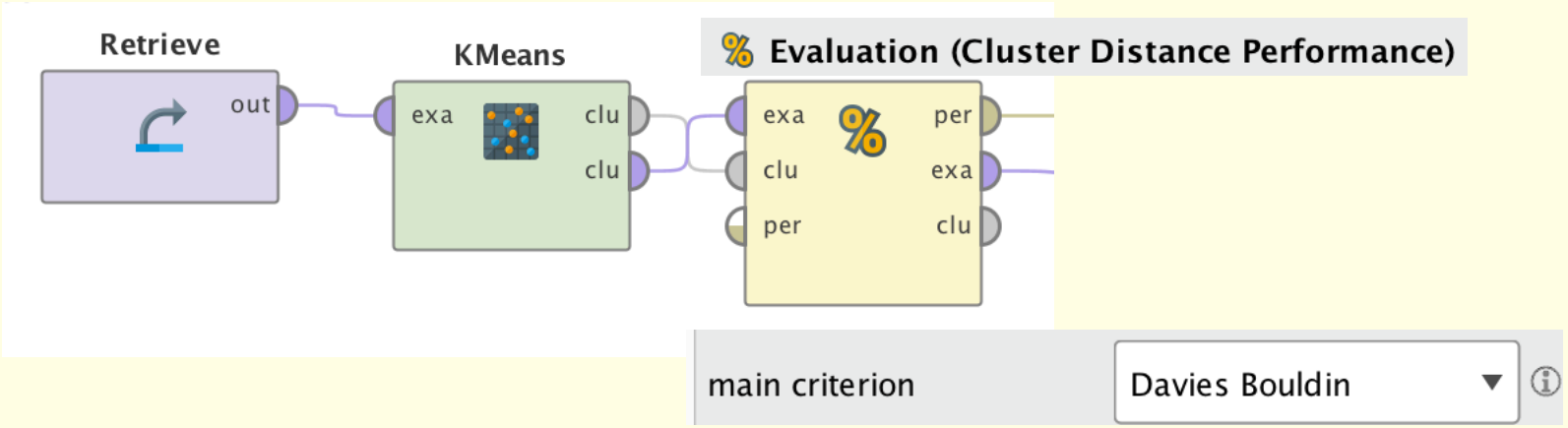
So, working through each cluster, find the max ratio of: the average distance within clusters over distance between clusters. Add these max ratios together, and divide by the the number of clusters.

A **small** DB refers to **compact clusters** that are **far away** from each other, so the cluster configuration that minimises DB is optimal.

To use this formula, a cluster centre must be available.

Davies Bouldin in Rapidminer

The operator Cluster distance performance in Rapidminer calculates Davies Bouldin for a dataset provided the cluster model used defines a cluster center, e.g. k-Means. The screen shot below is taken from the process samples/processes/07_custering/08_kMeansWithEvaluation in the samples Rapidminer repository.



Result for Iris dataset:

Davies Bouldin

Davies Bouldin: -0.666

Summary

- Document vectors, by their nature, tend to have large numbers of dimensions and sparsely populated datasets, making patterns hard to find.
- When preparing the data for clustering, a miner should attempt to minimise the number of concepts used to define clusters.

- Key to the results of a clustering algorithm is how distance is measured. Asymmetric measures must be used with documents vectors such as:
  - Shared word count
  - Shared word count weighted by document frequency
  - Cosine measure

- There are a number of categories of clustering algorithms:
  - Partitioning methods for which you need to specify the number of clusters in advance.
  - Hierarchical clustering methods which are good at identifying the optimal number of clusters.
  - Grid based methods which are good for visualising clusters and highlighting outliers.
  - Genetic Algorithms which are also good for visualising clusters, and are scalable