Clustering
Part 5

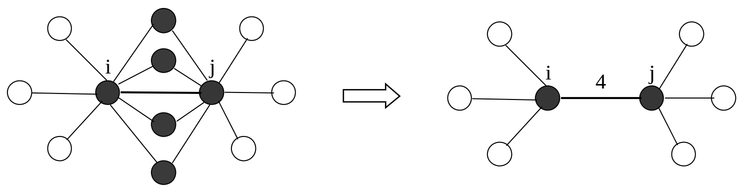
Dr. Sanjay Ranka
Professor
Computer and Information Science and Engineering
University of Florida, Gainesville

SNN Approach to Clustering

- Ordinary distance measures have problems
  - Euclidean distance is less appropriate in high dimensions
    - Presences are more important than absences
  - Cosine and Jaccard measure take in to account presences, but do not satisfy the triangle inequality
- SNN distance is more appropriate in these cases

Shared Near Neighbor Graph

- In the SNN graph, the strength of a link is the number of shared neighbors between documents given that the documents are connected

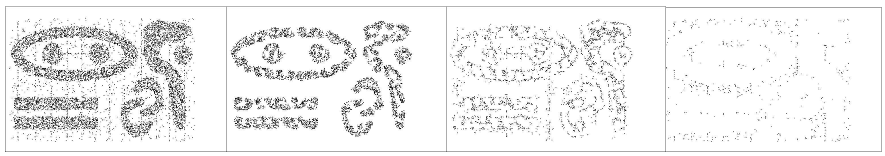


SNN Approach: Density

- Ordinary density measures have problems
  - Typical Euclidean density is number of points per unit volume
  - As dimensionality increases, density goes to 0
- Can estimate the relative density, i.e., probability density, in a region
  - Look at the distance to the k^{th} nearest neighbor, or
  - Look at the number of points within a fixed radius
  - However, since distances become uniform in high dimensions, this does not work well either
- If we use SNN similarity then we can obtain a more robust definition of density
  - Relatively insensitive to variations in normal density
  - Relatively insensitive to high dimensionality
  - Uniform regions are dense, gradients are not

SNN Density can identify Core, Border and Noise points

- Assume a DBSCAN definition of density
  - Number of points within *Eps*
- Example



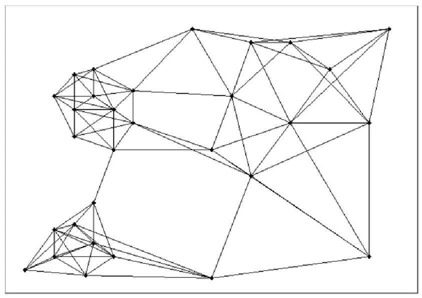
a) All Points Density b) High SNN Density c) Medium SNN Density d) Low SNN Density

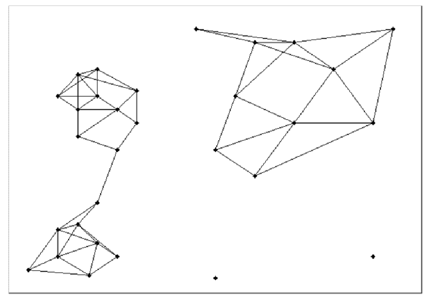
ROCK

- ROCK (RObust Clustering using linKs)
  - Clustering algorithm for data with categorical and boolean attributes
    - It redefines the distances between points to be the number of shared neighbors whose strength is greater than a given threshold
    - Then uses a hierarchical clustering scheme to cluster the data

1. Obtain a sample of points from the data set
2. Compute the link value for each set of points, i.e., transform the original similarities (computed by the Jaccard coefficient) into similarities that reflect the number of shared neighbors between points
3. Perform an agglomerative hierarchical clustering on the data using the “number of shared neighbors” similarities and the “maximize the shared neighbors” objective function
4. Assign the remaining points to the clusters that have been found

Creating the SNN Graph





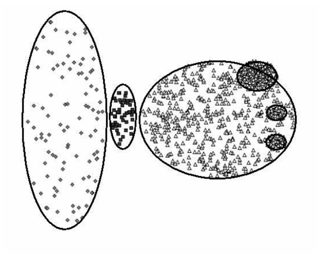
5 Near neighbor graph

Shared near neighbor graph

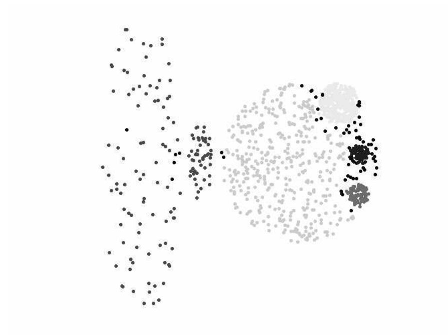
Jarvis Patrick Clustering

- First, the k-nearest neighbors of all points are found
  - In graph terms this can be regarded as breaking all but the *k* strongest links from a point to other points in the proximity graph
- A pair of points is put in the same cluster if
  - any two points share more than *T* neighbors and
  - the two points are in each others *k* nearest neighbor list
- For instance, we might choose a nearest neighbor list of size 20 and put points in the same cluster if they share more than 10 near neighbors
- JP is too brittle

When Jarvis Patrick Works Reasonably Well



Original Points



Jarvis Patrick Clustering

6 shared neighbors out of 20

When Jarvis Patrick Does NOT Work Well

Smallest threshold, T, that does not merge clusters.

Threshold of T - 1

SNN Clustering Algorithm

1. **Compute the similarity matrix**

   This corresponds to a similarity graph with data points for nodes and edges whose weights are the similarities between data points.

2. **Sparsify the similarity matrix by keeping only the *k* most similar neighbors**

   This corresponds to only keeping the *k* strongest links of the similarity graph.

SNN Clustering Algorithm …

3. **Construct the shared nearest neighbor graph from the sparsified similarity matrix**

   At this point, we could apply a similarity threshold and find the connected components to obtain the clusters (Jarvis-Patrick algorithm)

4. **Find the SNN density of each point**

   Using a user specified parameter, *Eps*, find the number points that have an SNN similarity of *Eps* or greater to each point. This is the SNN density of the point.

SNN Clustering Algorithm …

5. **Find the core points**

   Using user specified parameter, *MinPts*, find the core points, i.e., all points that have an SNN density greater than *MinPts*.

6. **Form clusters from the core points**

   If two core points are within a radius, *Eps*, of each other they are placed in the same cluster.

SNN Clustering Algorithm …

7. **Discard all noise points**

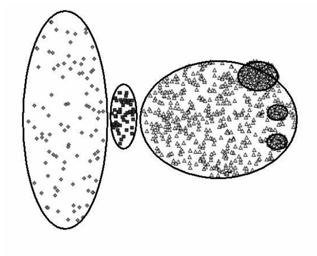
   All non-core points that are not within a radius of *Eps* of a core point are discarded.

8. **Assign all non-noise, non-core points to clusters**

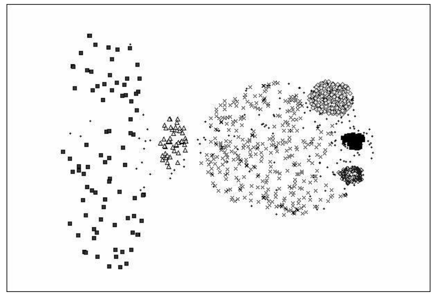
   This can be done by assigning such points to the nearest core point

Note that steps 4 – 8 are DBSCAN

SNN Clustering Can Handle Differing Densities

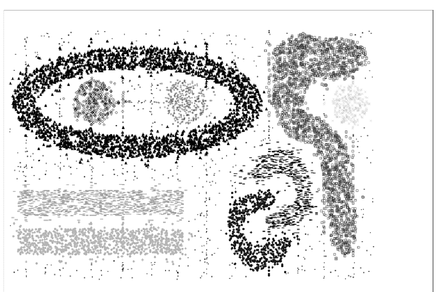


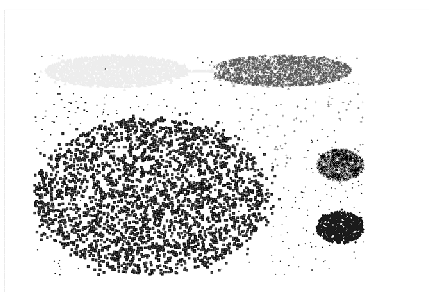
Original Points



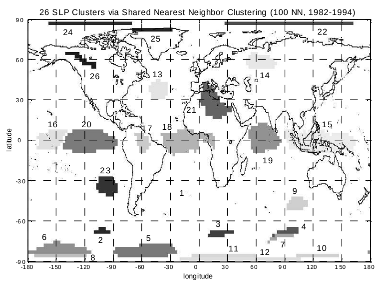
SNN Clustering

SNN Can Handle Other Difficult Situations

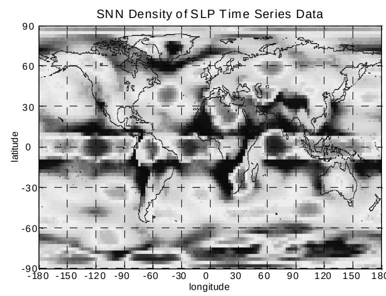




Finding Clusters of Time Series in Spatio-Temporal Data

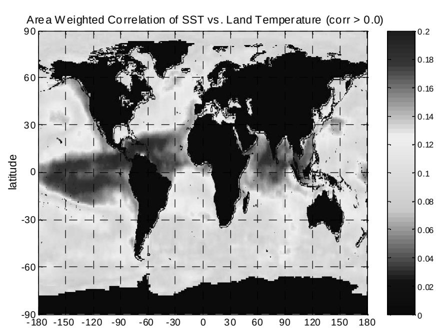


SNN Clusters of SLP

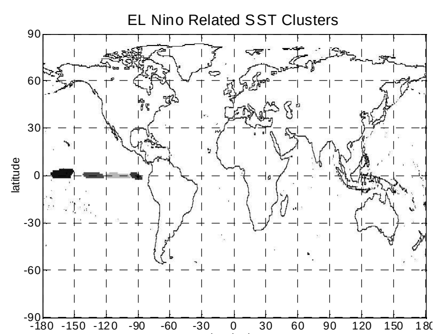


SNN Density of Points on the Globe

Finding Clusters of Time Series in Spatio-Temporal Data



Area Weighted Correlation of SST to Land temperature



Four SNN Clusters of SST

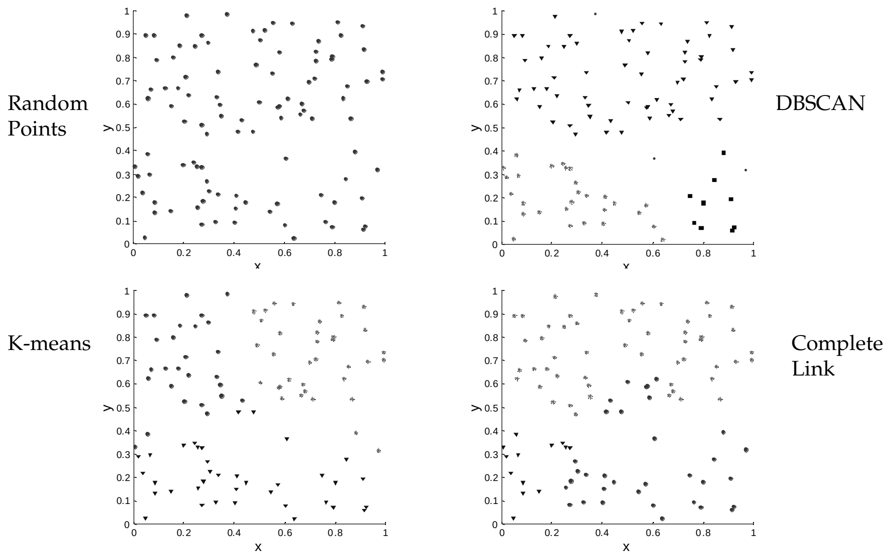
Features and Limitations of SNN Clustering

- Does not cluster all the points
  - Points can be added back in
- Complexity of SNN Clustering is high
  - O(n * time to find numbers of neighbor within *Eps*)
  - In worst case, this is $O(n^2)$
  - For lower dimensions, there are more efficient ways to find the nearest neighbors
    - R* Tree
    - k-d Trees

Cluster Validity

- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall
- For cluster analysis, the analogous question is how to evaluate the “goodness” of a the resulting clusters?
- However, if “clusters are in the eye of the beholder” then why should we want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two sets of clusters
  - To compare two clusters

Clusters Found in Random Data



Different Aspects of Cluster Validation

1. Determining the **clustering tendency** of a set of data, i.e., distinguishing whether non-random structure actually exists in the data
2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information
   - Use only the data
4. Comparing the results of two different sets of cluster analyses to determine which is better
5. Determining the ‘correct’ number of clusters

For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

Measures of Cluster Validity

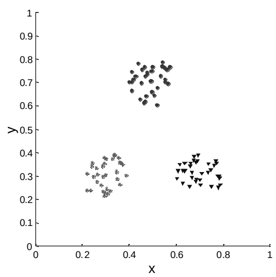
- The numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types:
  - External Index: Used to measure the extent to which cluster labels match externally supplied class labels
    - Entropy
  - Internal Index: Used to measure the goodness of a of clustering structure *without* respect to external information
    - Sum of Squared Error (SSE)
  - Relative Index: Used to compare two different clusterings or clusters
    - Often an external or internal index is used for this function, e.g., SSE or entropy
- Sometimes these are referred to as criteria instead of indices
  - However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion

Measuring Cluster Validity via Correlation

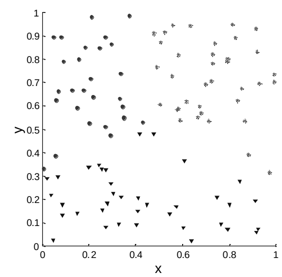
- Two matrices
  - Proximity Matrix
  - "Incidence" Matrix
    - One row and one column for each data point
    - An entry is 1 if the associated pair of points belong to the same cluster
    - An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between the two matrices
  - Since the matrices are symmetric, only the correlation between n(n-1) / 2 entries needs to be calculated
- High correlation indicates that points that belong to the same cluster are close to each other
- Not a good measure for some density or contiguity based clusters

Measuring Cluster Validity via Correlation

- Correlation of incidence and proximity matrices for the K-means clusterings of the following two data sets



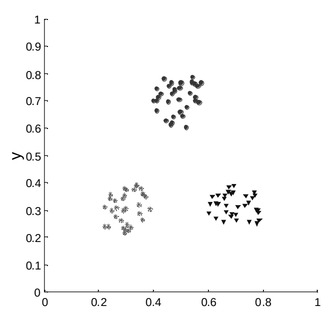
Correlation = -0.9235

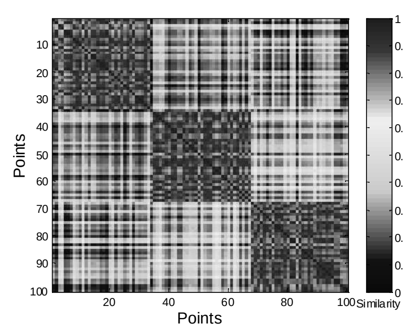


Correlation = -0.5810

Using Similarity Matrix for Cluster Validation

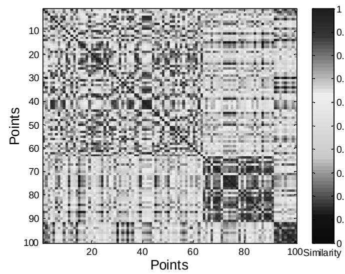
- Order the similarity matrix with respect to cluster labels and inspect visually

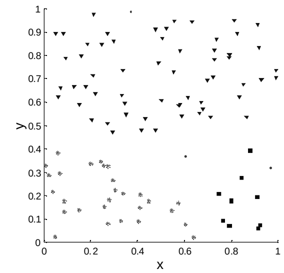




Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

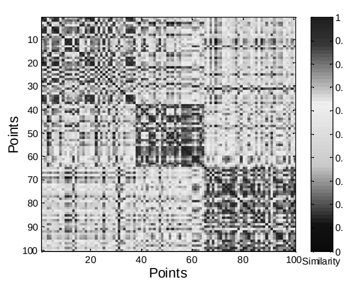


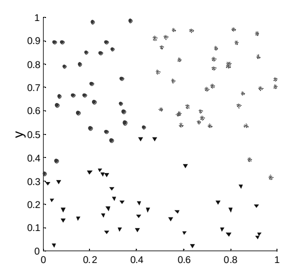


DBSCAN

Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

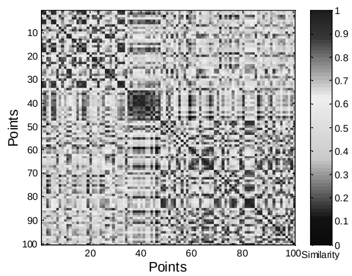


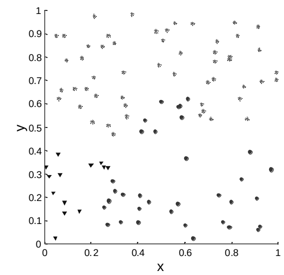


K-means

Using Similarity Matrix for Cluster Validation

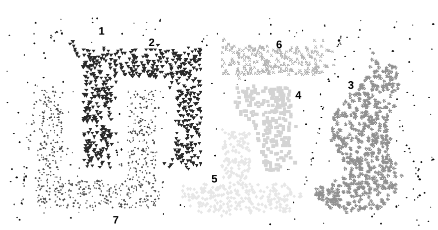
- Clusters in random data are not so crisp

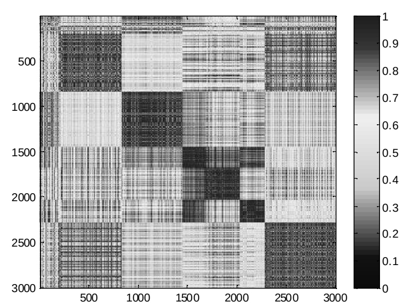




Complete Link

Using Similarity Matrix for Cluster Validation

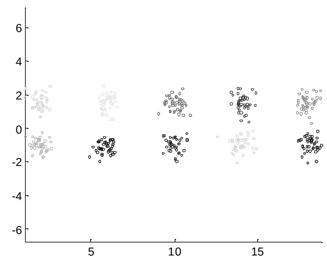


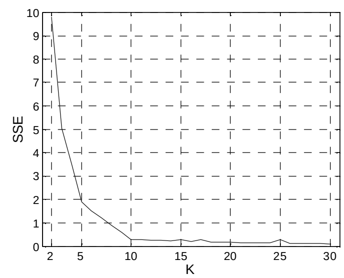


DBSCAN

Internal Measures for Cluster Validity: SSE

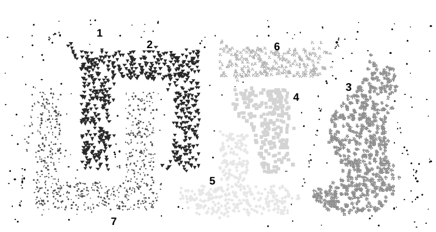
- Clusters in more complicated figures aren't well separated
- Internal Index: Used to measure the goodness of a of clustering structure without respect to external information.
  - SSE
- SSE is good for comparing two clusterings or two clusters (average SSE)
- Can also be used to estimate the number of clusters

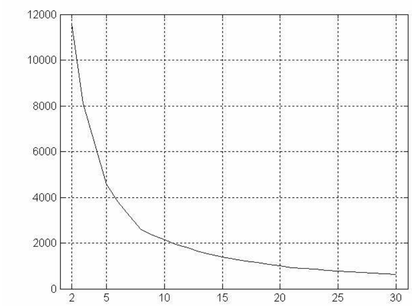




Internal Measures for Cluster Validity: SSE

- SSE curve for a more complicated data set





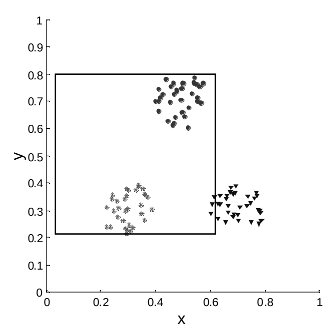
SSE of clusters found using K-means

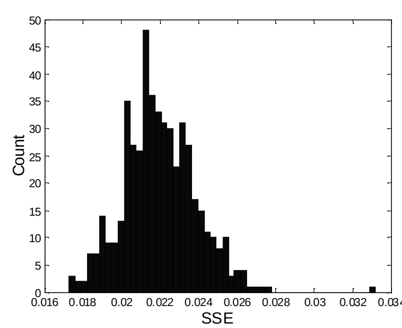
Framework for Cluster Validity

- Need a framework to interpret any measure
  - For example, if our measure of evaluation has the value, 10, is does that good, fair, or poor?
- Statistics can provide a framework
  - The more atypical a clustering result is, the more likely it represents valid structure in the data
  - Can compare the values of an index that result from random data or clusterings to those of a clustering result
    - If the value of the index is unlikely, then the cluster results are valid
  - These approaches are more complicated and hard to understand
- For comparing the results of two different sets of cluster analyses, a framework is less necessary
  - However, there is the question of whether the difference between two index values is significant

Statistical Framework for SSE

- If you want a more absolute measure
  - Compare SSE of 0.005 against three clusters in random data
  - Histogram shows SSE of three clusters in 500 sets of random data points of size 100 distributed over the range 0.2 – 0.8 for x and y values

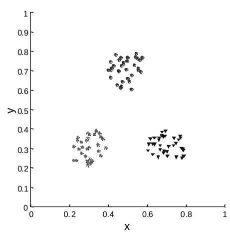




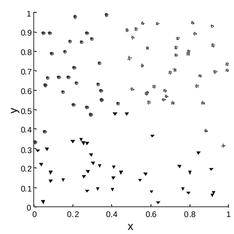
- In general, it can be hard to have representative data to generate statistics

Statistical Framework for Correlation

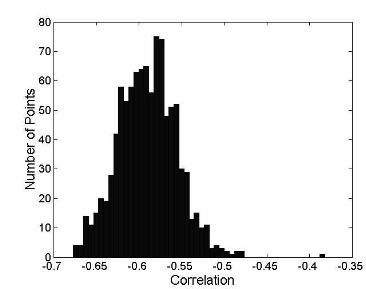
- Correlation of incidence and proximity matrices for the K-means clusterings of the following two data sets.



Correlation = -0.9235



Correlation = -0.5810



Internal Measures of Cluster Validity: Cohesion and Separation

- Cluster Cohesion: Measure of how closely the objects in a cluster are related. E.g. SSE
- Cluster Separation: Measure of how well-separated or distinct a cluster is from other clusters
- Example: Squared Error
  - Cohesion is measure by the within cluster sum of squares (SSE)

$$WSS = \sum_{i} \sum_{x \in C_i} (x - m_i)^2$$

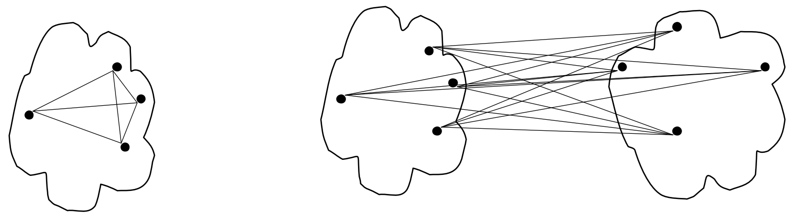
  - Separation is measured by the between cluster sum of squares

$$BSS = \sum_{i} (m - m_i)^2$$

  - BSS + WSS = constant

Internal Measures of Cluster Validity: Cohesion and Separation

- A proximity graph based approach can also be used for cohesion and separation
  - Cluster cohesion is the sum of the weight of all links within a cluster
  - Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster

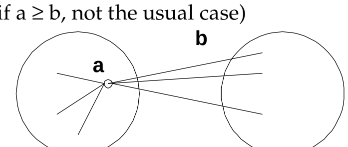


cohesion separation

Internal Measures of Cluster Validity: Silhouette Coefficient

- Silhouette Coefficient combine ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings
- For an individual point, *i*
  - Calculate ***a*** = average distance of *i* to the points in its cluster
  - Calculate ***b*** = min (average distance of *i* to points in another cluster)
  - The silhouette coefficient for a point is then given by

    $s = 1 - a/b$ if $a < b$, (or $s = b/a - 1$ if $a \geq b$, not the usual case)



  - Typically between 0 and 1
  - The closer to 1 the better
- Can calculate the Average Silhouette width for a cluster or a clustering

External Measures of Cluster Validity: Entropy and Purity

Table 5.9. K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

entropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij}, the 'probability' that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^{L} p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{i=1}^{K} \frac{m_i}{m} e_j$, where m_j is the size of cluster j, K is the number of clusters, and m is the total number of data points.

purity Using the terminology derived for entropy, the purity of cluster j, is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{i=1}^{K} \frac{m_i}{m} purity_j$.

Final Comment on Cluster Validity

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

Algorithms for Clustering Data, Jain and Dubes

Issues

- Scalability
- Independence of the order of input
- Effective means of detecting and dealing with noise or outlying points
- Effective means of evaluating the validity of clusters that are produced.
- Easy interpretability of results

Issues …

- The ability to find clusters in subspaces of the original space
- The ability to handle distances in high dimensional spaces properly
- Robustness in the presence of different underlying data and cluster characteristics
- An ability to estimate any parameters
- An ability to function in an incremental manner

Issues: Handling Different Types of Data

- Robustness in the presence of different underlying data and cluster characteristics
  - Dimensionality
  - Noise and Outliers
  - Statistical Distribution
  - Cluster Shape
  - Cluster Size
  - Cluster Density
  - Cluster Separation
  - Type of data space, e.g., Euclidean or non-Euclidean
  - Many and Mixed Attribute Types

Other Clustering Approaches

- Modeling clusters as a “mixture” of Multivariate Normal Distributions. (Raftery and Fraley)

- Bayesian Approaches (*AutoClass*, Cheeseman)

- Neural Network Approaches (*SOM*, Kohonen)

- Many, many other variations and combinations of approaches