Clustering

Rich Caruana
Cornell University

Supervised Learning

- Linear regression
- Logistic regression
- Artificial neural nets
- Decision trees
- K-nearest neighbor
- Support vectors
- ...

Supervised Learning

Train Set:

57,M,195,0,125,95,39,25,0,1,0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0 0
78,M,160,1,130,100,37,40,1,0,0,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0 1
69,F,180,0,115,85,40,22,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0 0
18,M,165,0,110,80,41,30,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 0
54,F,135,0,115,95,39,35,1,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,0 1
84,F,210,1,135,105,39,24,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0 0
89,F,135,0,120,95,36,28,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,0,0 1
49,M,195,0,115,85,39,32,0,0,0,1,1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0 0
40,M,205,0,115,90,37,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 0
74,M,250,1,130,100,38,26,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 0
77,F,140,0,125,100,40,30,1,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,1 1
...

Test Set:

71,M,160,1,130,105,38,20,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0 ?

Supervised Learning

Well Defined Goal:

Learn function y = f(x) that is a good approximation from the training sample D

Know How to Measure Error:

Accuracy, Squared Error, ...

Clustering

≠

Supervised Learning

Clustering

=

Unsupervised Learning

Supervised Learning

Train Set:

57,M,195,0,125,95,39,25,0,1,0,0,0,1,0,0,0,0,0,1,1,0,0,0,0,0,0,0	0
78,M,160,1,130,100,37,40,1,0,0,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0	1
69,F,180,0,115,85,40,22,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0	0
18,M,165,0,110,80,41,30,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	0
54,F,135,0,115,95,39,35,1,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0	1
84,F,210,1,135,105,39,24,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0	0
89,F,135,0,120,95,36,28,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,1,0,0	1
49,M,195,0,115,85,39,32,0,0,0,1,1,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0	0
40,M,205,0,115,90,37,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	0
74,M,250,1,130,100,38,26,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0	0
77,F,140,0,125,100,40,30,1,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,1	1
…	

Test Set:

71,M,160,1,130,105,38,20,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0	?

Un-Supervised Learning

Train Set:

57,M,195,0,125,95,39,25,0,1,0,0,0,1,0,0,0,0,0,1,1,0,0,0,0,0,0,0	0
78,M,160,1,130,100,37,40,1,0,0,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0	1
69,F,180,0,115,85,40,22,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0	0
18,M,165,0,110,80,41,30,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	0
54,F,135,0,115,95,39,35,1,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0	1
84,F,210,1,135,105,39,24,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0	0
89,F,135,0,120,95,36,28,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,1,0,0	1
49,M,195,0,115,85,39,32,0,0,0,1,1,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0	0
40,M,205,0,115,90,37,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	0
74,M,250,1,130,100,38,26,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0	0
77,F,140,0,125,100,40,30,1,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,1	1
…	

Test Set:

71,M,160,1,130,105,38,20,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0	?

Un-Supervised Learning

Train Set:

57,M,195,0,125,95,39,25,0,1,0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0	0
78,M,160,1,130,100,37,40,1,0,0,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0	1
69,F,180,0,115,85,40,22,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0	0
18,M,165,0,110,80,41,30,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	0
54,F,135,0,115,95,39,35,1,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0	0
84,F,210,1,135,105,39,24,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0	1
89,F,135,0,120,95,36,28,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,1,0,0	1
49,M,195,0,115,85,39,32,0,0,0,1,1,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0	0
40,M,205,0,115,90,37,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	0
74,M,250,1,130,100,38,26,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0	0
77,F,140,0,125,100,40,30,1,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,1	1
…	

Test Set:

71,M,160,1,130,105,38,26,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0	?

Un-Supervised Learning

Data Set:

57,M,195,0,125,95,39,25,0,1,0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0
78,M,160,1,130,100,37,40,1,0,0,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0
69,F,180,0,115,85,40,22,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0
18,M,165,0,110,80,41,30,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
54,F,135,0,115,95,39,35,1,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0
84,F,210,1,135,105,39,24,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
89,F,135,0,120,95,36,28,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,1,0,0
49,M,195,0,115,85,39,32,0,0,0,1,1,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0
40,M,205,0,115,90,37,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
74,M,250,1,130,100,38,26,1,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
77,F,140,0,125,100,40,30,1,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,1
…

Clustering

- Goal:
  - Find groups in data
  - Group similar items together
  - Separate different kinds of objects
  - Find patterns in data
  - Organize/understand complex data

Clustering

- Given:
  - Data Set D (training set)
  - Similarity/distance metric/information
- Find:
  - Partitioning of data
  - Groups of similar/close items

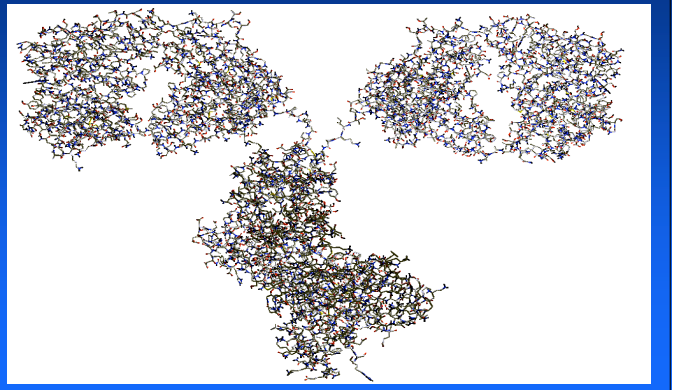
Similarity?

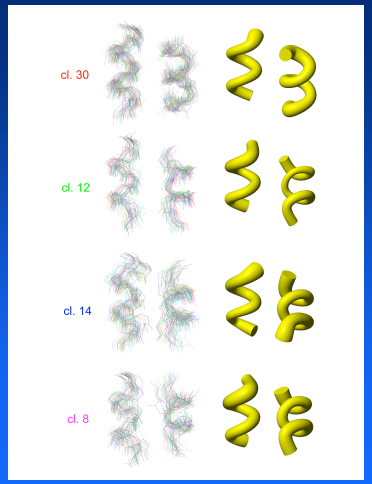
- Groups of similar customers
  - Similar demographics
  - Similar buying behavior
  - Similar health
- Similar products
  - Similar cost
  - Similar function
  - Similar store
  - …
- Similarity usually is domain/problem specific

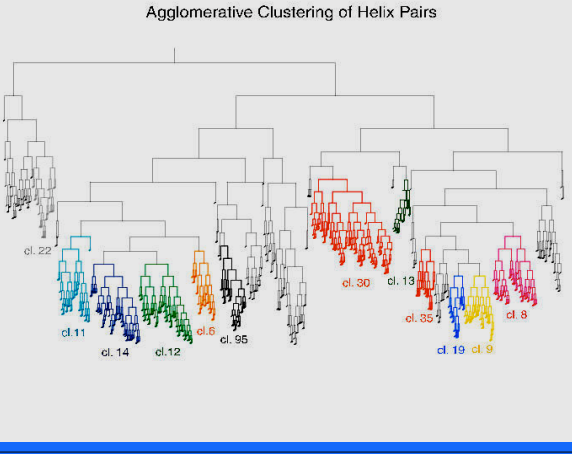
Types of Clustering

- Hierarchical
  - Divisive clustering (top down)
  - Agglomerative clustering (bottom up)
- Partitioning
  - K-means clustering
  - K-medoids clustering
  - EM (expectation maximization) clustering
- Density-Based Methods
  - Regions of dense points separated by sparser regions of relatively low density

Clustering Proteins







Two Types of Data/Distance Info

- N-dim vector space representation and distance metric

D1: 57,M,195,0,125,95,39,25,0,1,0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0

D2: 78,M,160,1,130,100,37,40,1,0,0,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0

...

Dn: 18,M,165,0,110,80,41,30,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

Distance (D1,D2) = distance_function

- Pairwise distances between points (no N-dim space)
  - + Similarity/dissimilarity matrix (upper or lower diagonal)

  - + Distance: 0 = near, ∞ = far
  - + Similarity: 0 = far, ∞ = near

--	1	2	3	4	5	6	7	8	9	10
1	-	d	d	d	d	d	d	d	d	d
2		-	d	d	d	d	d	d	d	d
3			-	d	d	d	d	d	d	d
4				-	d	d	d	d	d	d
5					-	d	d	d	d	d
6						-	d	d	d	d
7							-	d	d	d
8								-	d	d
9									-	d

Agglomerative Clustering

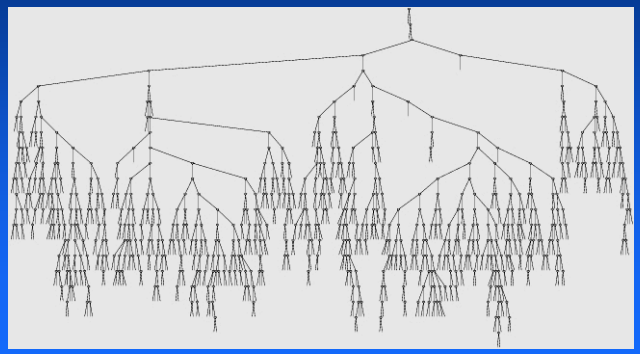
- Put each item in its own cluster (N singletons)

- Calculate pairwise distances between clusters
- Merge the two *closest* clusters
- Repeat until all points in one cluster

- Hierarchical clustering
- Yields a clustering with each possible # of clusters
- Greedy clustering: not optimal for any cluster size

Do Agglomerative Clustering Demo

Agglomerative Clustering of Proteins



Merging: Closest Clusters

- Nearest neighbors (shortest link)
- Nearest average distance (average link)
- Smallest greatest distance (maximum link)
- Domain specific similarity measure
  - word frequency, TFIDF, KL-divergence, ...
- Merge clusters that optimize criterion after merge
  - minimum mean_internal_distance

Mean Distance Between Clusters

$$Mean\_Dist\ (c_1, c_2) = \frac{\sum_{i \in c_1} \sum_{j \in c_2} Dist\,(i, j)}{\sum_{i \in c_1} \sum_{j \in c_2} 1}$$

Minimum Distance Between Clusters

$$Min\_Dist\ (c_1, c_2) = \underset{i \in c_1, j \in c_2}{MIN}\ (Dist\,(i, j))$$

Mean Internal Distance in Cluster

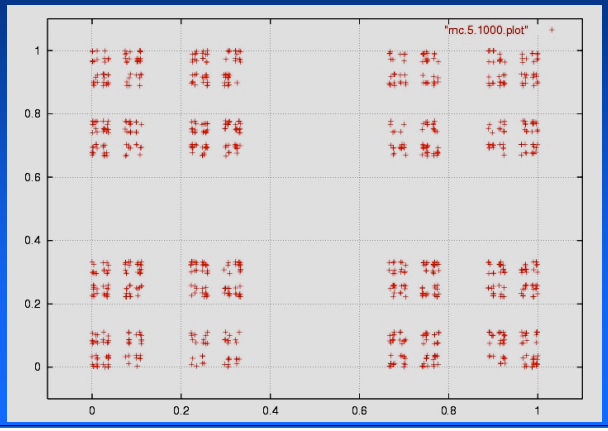
$$Mean\_Internal\_Dist(c) = \frac{\sum_{i \in c} \sum_{j \in c, i \neq j} Dist(i,j)}{\sum_{i \in c} \sum_{j \in c, i \neq j} 1}$$

Average Cluster Compactness

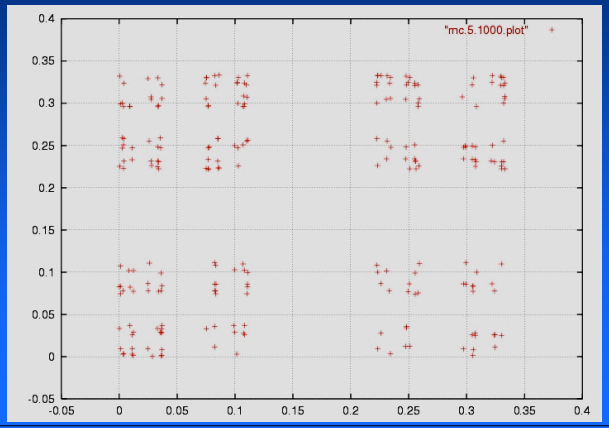
$$\delta_{ij} = \begin{cases} 1 & when\ cluster(i) = cluster(j) \\ 0 & when\ cluster(i) \neq cluster(j) \end{cases}$$

$$Avg\_Compactness = \frac{\sum_{i} \sum_{j \neq i} \delta_{ij} \bullet Dist(i,j)}{\sum_{i} \sum_{j \neq i} \delta_{ij}}$$

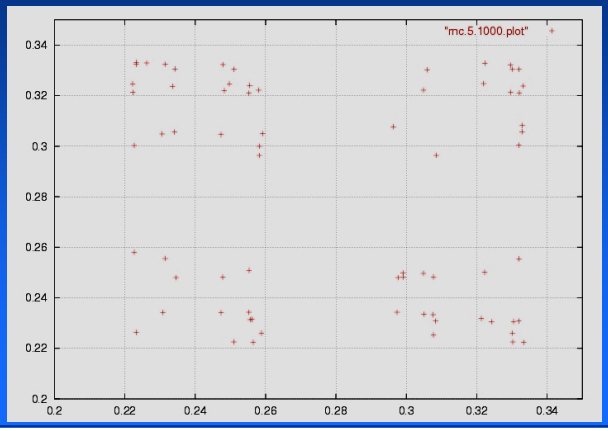
Recursive Clusters



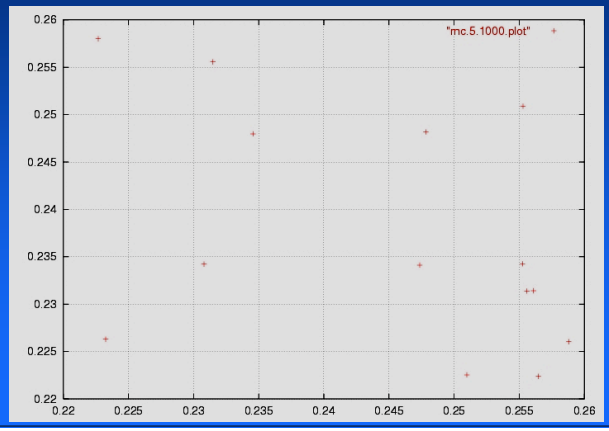
Recursive Clusters



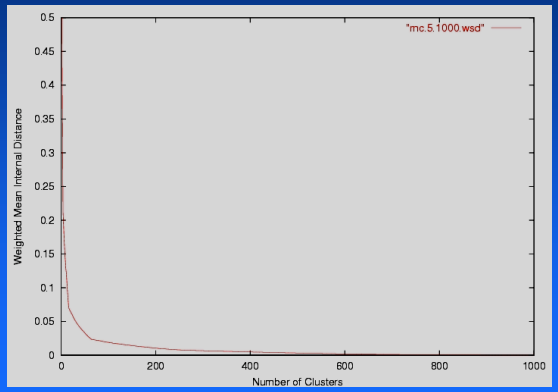
Recursive Clusters



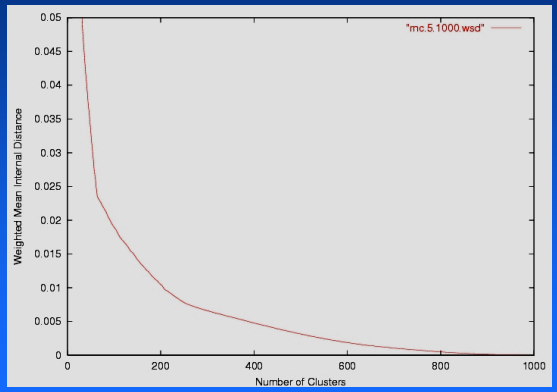
Recursive Clusters



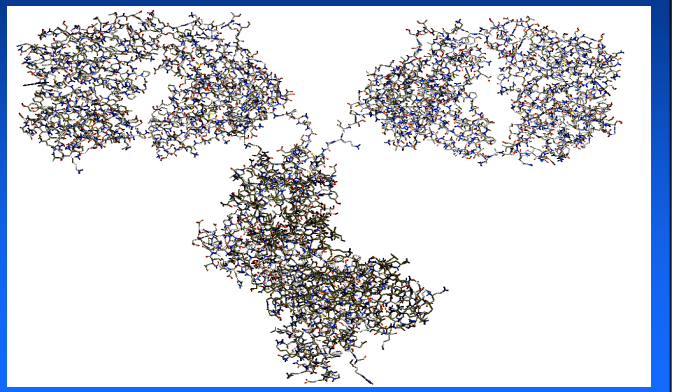
Weighted Mean Internal Distance

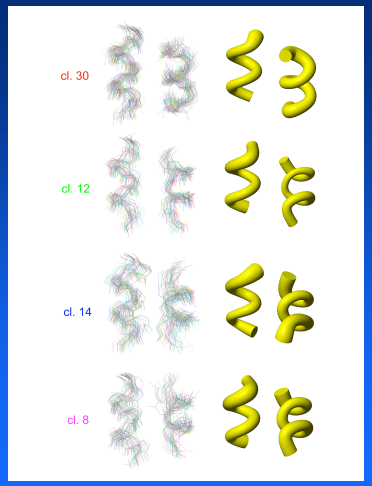


Weighted Mean Internal Distance



Clustering Proteins

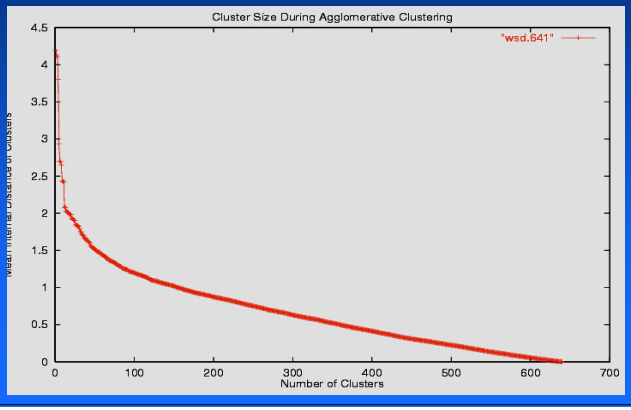




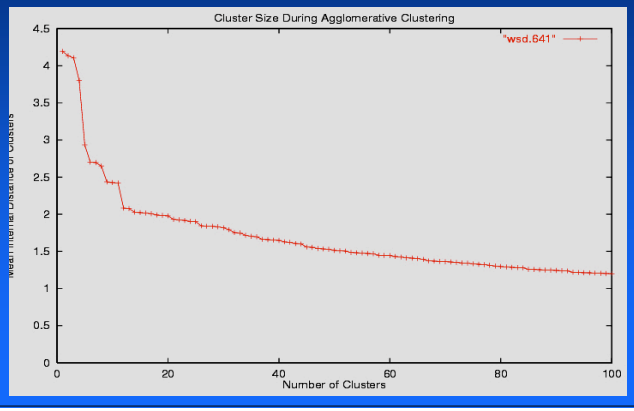
Distance Between Helices

- Vector representation of protein data in 3-D space that gives x,y,z coordinates of each atom in helix
- Use a program developed by chemists (fortran) to convert 3-D atom coordinates into average atomic distances in angstroms between aligned helices
- 641 helices = 641 * 640 / 2
  = 205,120 pairwise distances

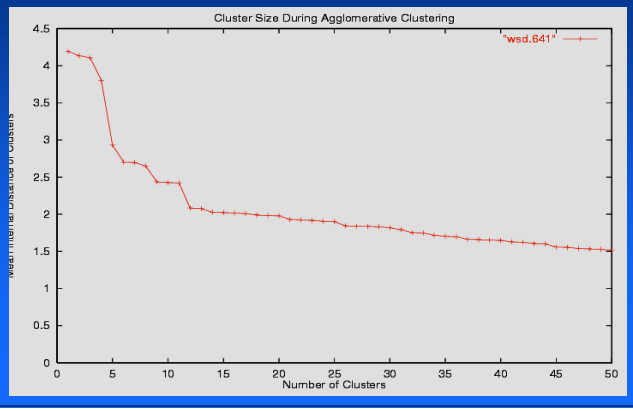
Agglomerative Clustering of Proteins



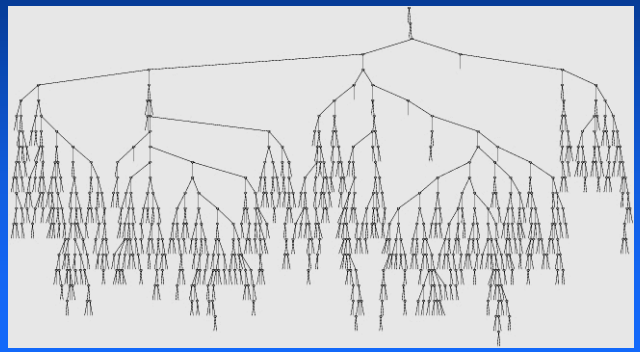
Agglomerative Clustering of Proteins



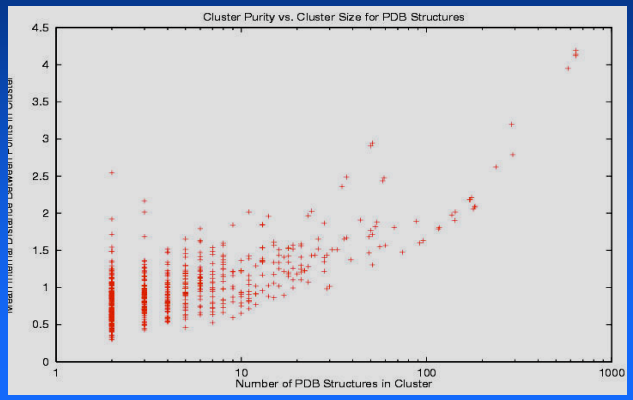
Agglomerative Clustering of Proteins

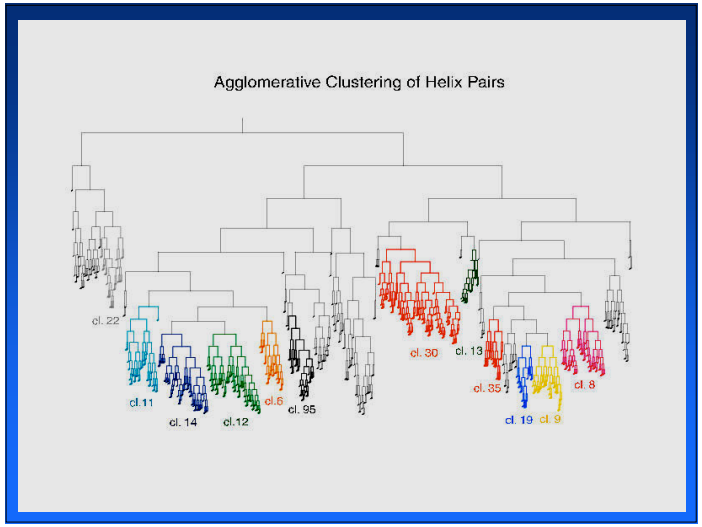


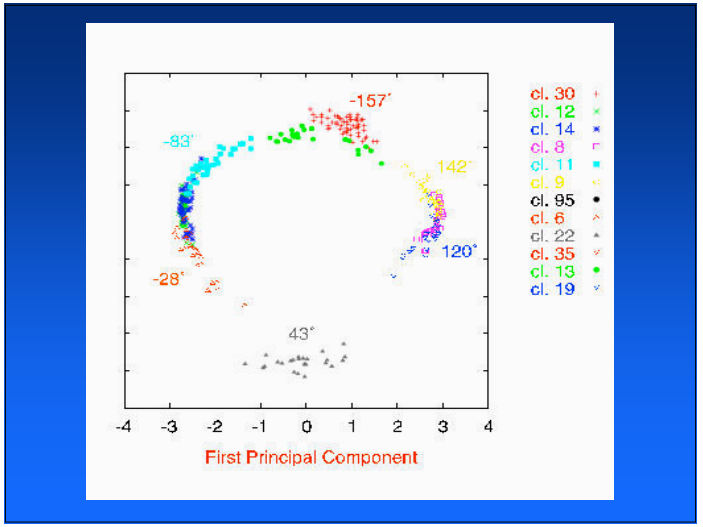
Agglomerative Clustering of Proteins

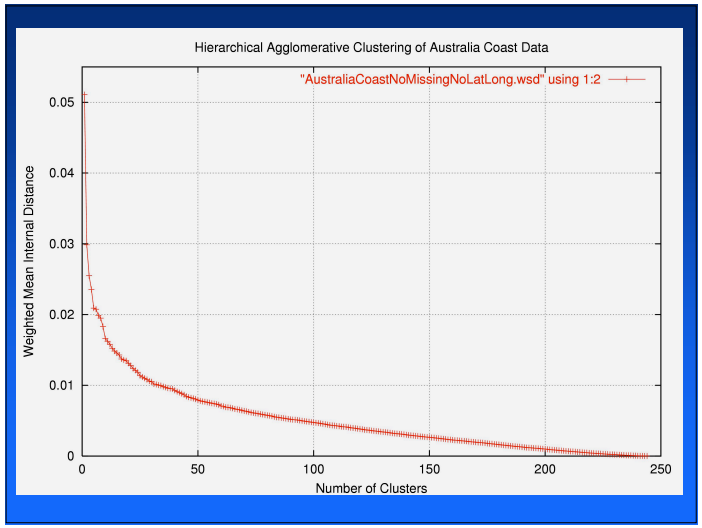


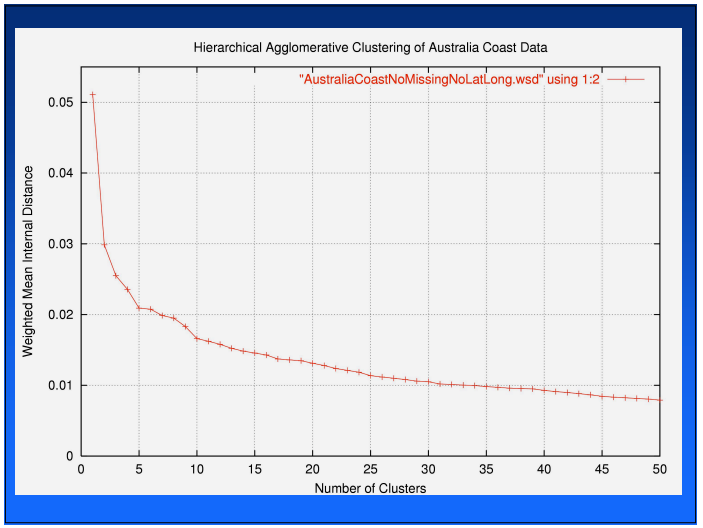
Agglomerative Clustering of Proteins

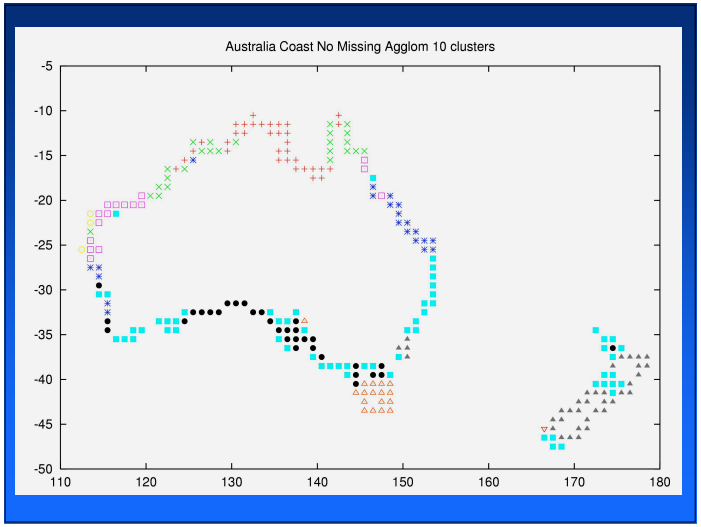



Agglomerative Clustering of Helix Pairs
cl. 22
cl. 11
cl. 14
cl. 12
cl. 6
cl. 95
cl. 30
cl. 13
cl. 35
cl. 19
cl. 9
cl. 8


-157°
-83°
142°
120°
-28°
43°
cl. 30
cl. 12
cl. 14
cl. 8
cl. 11
cl. 9
cl. 95
cl. 6
cl. 22
cl. 35
cl. 13
cl. 19
-4
-3
-2
-1
0
1
2
3
4
First Principal Component

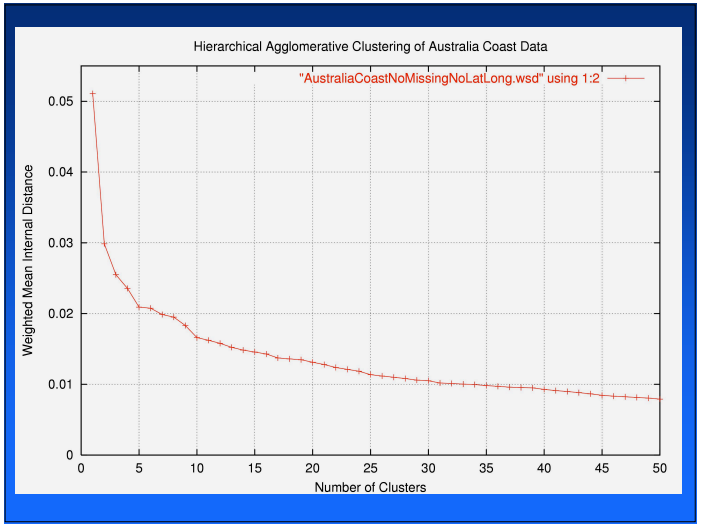

Hierarchical Agglomerative Clustering of Australia Coast Data
"AustraliaCoastNoMissingNoLatLong.wsd" using 1:2
Weighted Mean Internal Distance
0
0.01
0.02
0.03
0.04
0.05
0
50
100
150
200
250
Number of Clusters

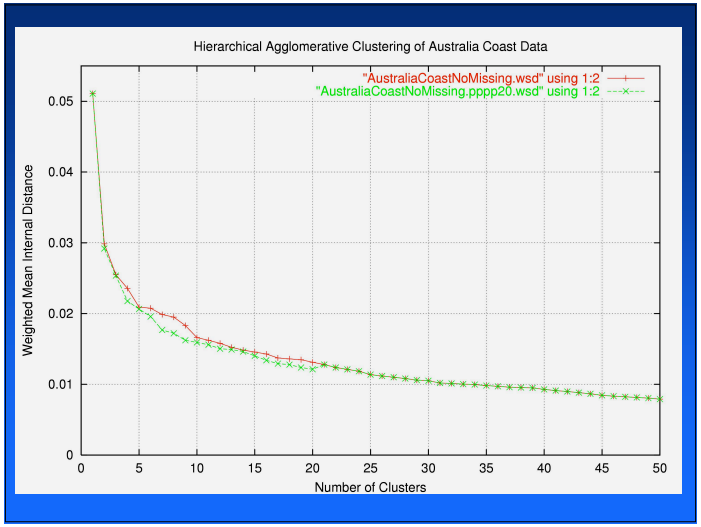

Hierarchical Agglomerative Clustering of Australia Coast Data
"AustraliaCoastNoMissingNoLatLong.wsd" using 1:2
Weighted Mean Internal Distance
0
0.01
0.02
0.03
0.04
0.05
0
5
10
15
20
25
30
35
40
45
50
Number of Clusters

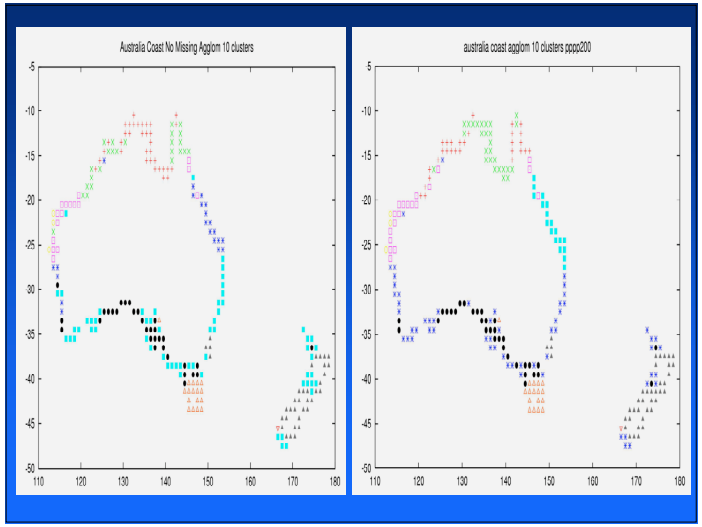


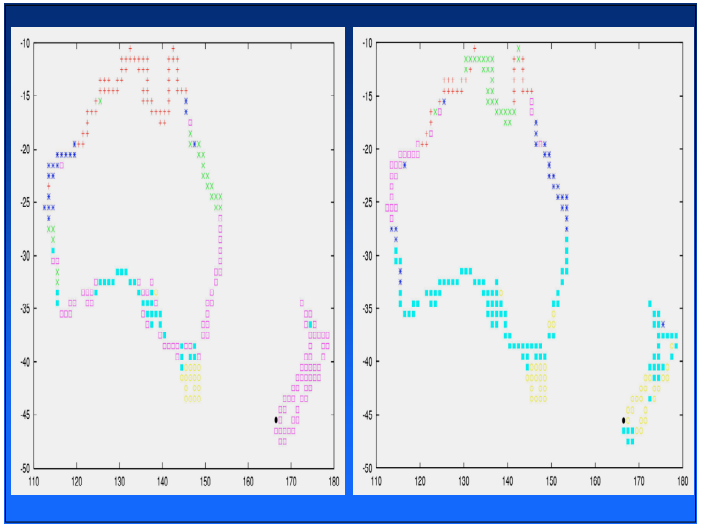
Agglomerative Clustering

- Greedy clustering
  - once points are merged, never separated
  - suboptimal w.r.t. clustering criterion
- Combine greedy with iterative refinement
  - post processing
  - interleaved refinement



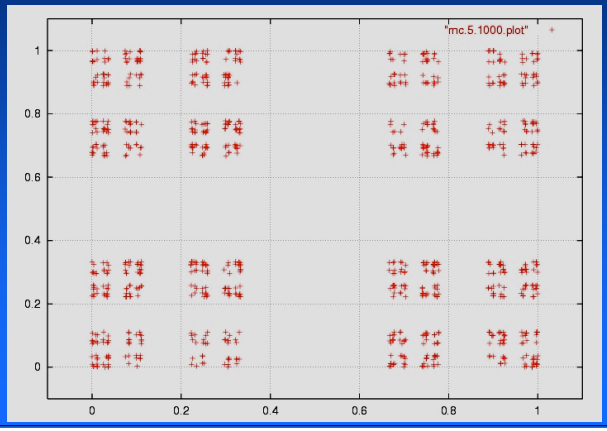




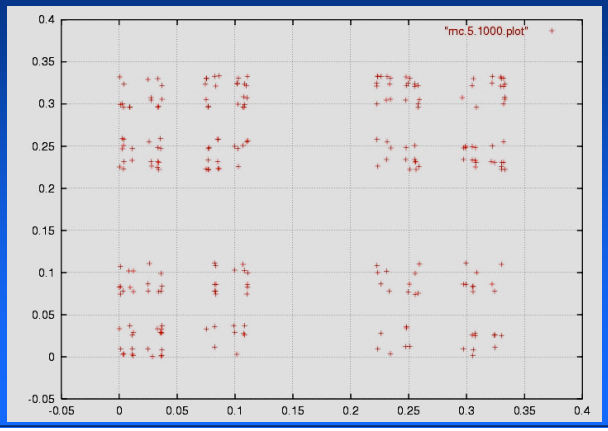


How do you tell if clustering is finding ***REAL*** structure in data?

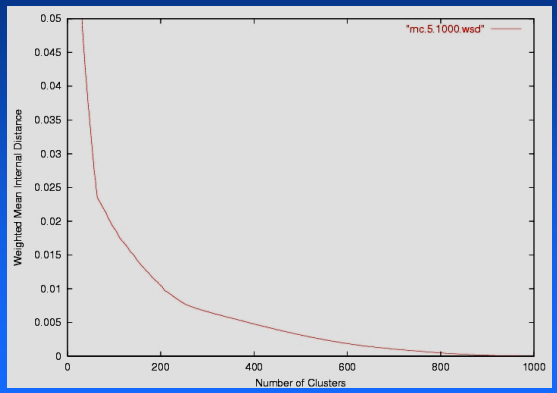
Recursive Clusters



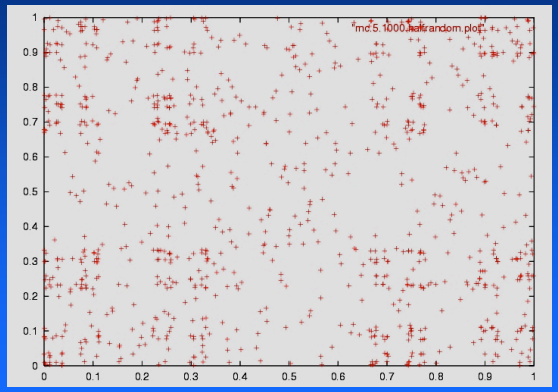
Recursive Clusters



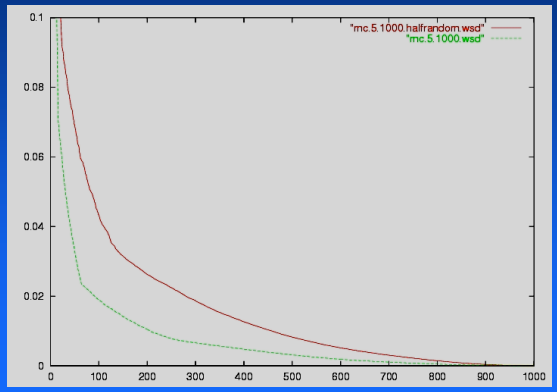
Weighted Mean Internal Distance



Recursive Clusters + Random Noise

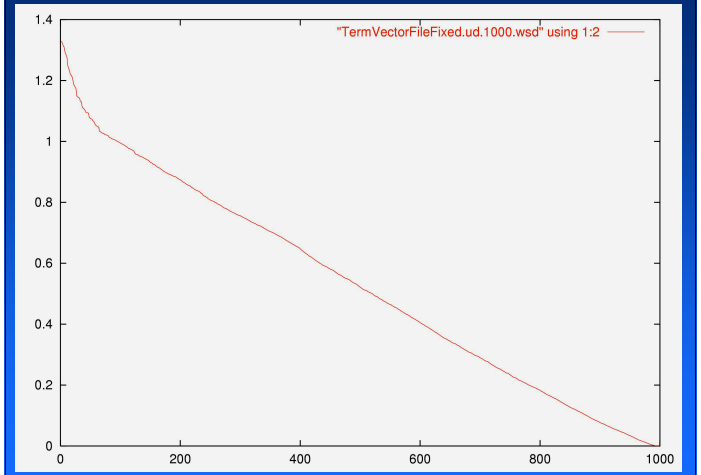


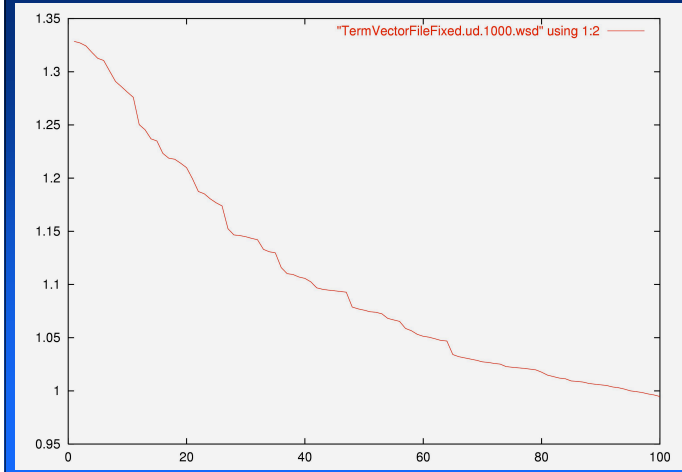
Recursive Clusters + Random Noise



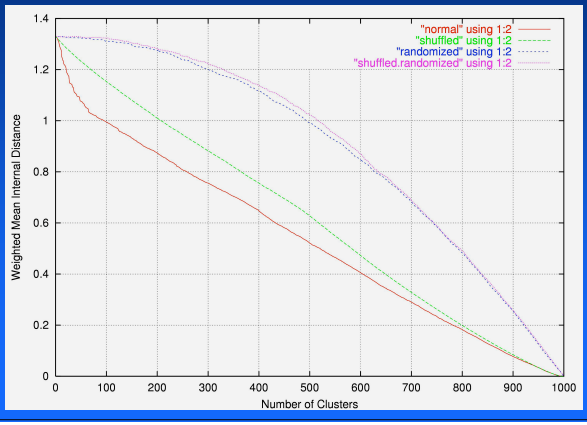
Bergmark Data

- 35,000 web documents
  - results from 26 focused web crawls
  - have crawl label of each document (know "class")
- Bags-of-words
  - how many times each word occurs in each document
- Cosine distance
  - good metric for sparse vectors
  - computationally efficient
    - Euclidean distance on normalized vectors ~ cosine distance
    - dot product of normalized vectors = cosine distance

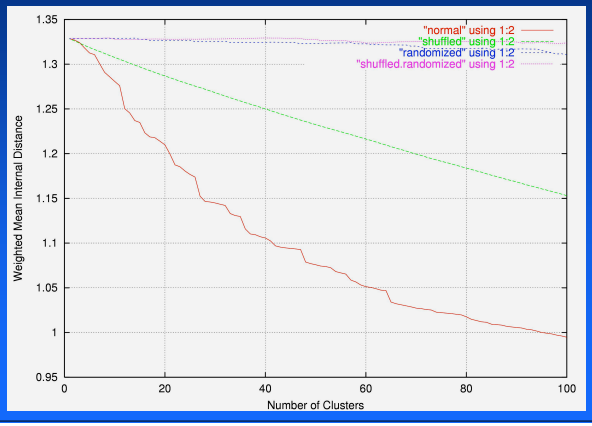




Agglomerative on Bergmark



Agglomerative on Bergmark



Agglomerative Clustering

- Computational Cost
  - $O(N^2)$ just to read/calculate pairwise distances
  - N-1 merges to build complete hierarchy
    - + scan pairwise distances to find closest
    - + calculate pairwise distances between clusters
    - + fewer clusters to scan as clusters get larger
  - Overall $O(N^3)$ for simple implementations
- Improvements
  - sampling
  - dynamic sampling: add new points while merging
  - Kruskal's algorithm for minimum spanning tree (shortest link)
  - tricks for updating pairwise distances

K-Means Clustering

- Inputs: data set and k (number of clusters)
- Output: each point assigned to one of k clusters

- K-Means Algorithm:
  - `Initialize the k-means`
    - + `assign from randomly selected points`
    - + `randomly or equally distributed in space`
  - `Assign each point to nearest mean`
  - `Update means from assigned points`
  - `Repeat until convergence`

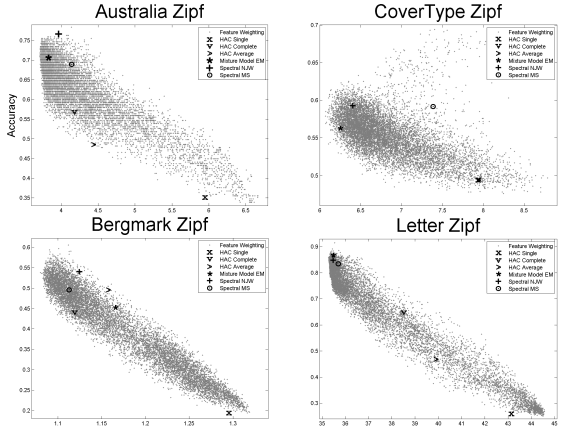
K-Means Clustering: Convergence

- Squared-Error Criterion

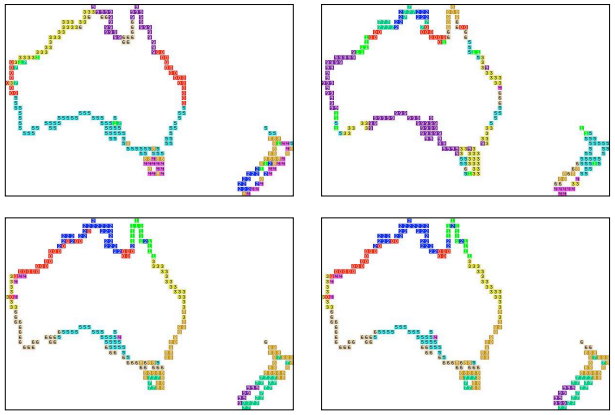
$$Squared\_Error = \sum_{c} \sum_{i \in c} (Dist(i, mean(c)))^2$$

- Converged when SE criterion stops changing
- Increasing K reduces SE - can't determine K by finding minimum SE
- Instead, plot SE as function of K

Clustering Methods Other Than K-Means?



4 Different Clusterings of Australia Coastline



K-Means Clustering

- Efficient
  - K << N, so assigning points is O(K*N) < O(N^2)
  - updating means can be done during assignment
  - usually # of iterations << N
  - Overall O(N*K*iterations) closer to O(N) than O(N^2)
- Gets stuck in local minima
  - Sensitive to initialization
- Number of clusters must be pre-specified
- Requires vector space date to calculate means

Soft K-Means Clustering

- Instance of EM (Expectation Maximization)
- Like K-Means, except each point is assigned to each cluster with a probability
- Cluster means updated using weighted average
- Generalizes to Standard_Deviation/Covariance
- Works well if cluster models are known

Soft K-Means Clustering (EM)

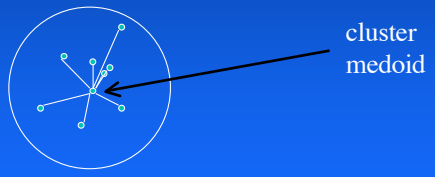
- Initialize model parameters:
  - means
  - std_devs
  - ...
- Assign points probabilistically to each cluster
- Update cluster parameters from weighted points
- Repeat until convergence to local minimum

What do we do if we can't calculate cluster means?

```
-- 1 2 3 4 5 6 7 8 9 10
1  - d d d d d d d d d
2    - d d d d d d d d
3      - d d d d d d d
4        - d d d d d d
5          - d d d d d
6            - d d d d
7              - d d d
8                - d d
9                  - d
```

K-Medoids Clustering

$$Medoid\ (c) = pt \in c\ s.t.\ MIN(\sum_{i \in c} Dist(i, pt))$$



K-Medoids Clustering

- Inputs: data set and k (number of clusters)
- Output: each point assigned to one of k clusters
-
- Initialize k medoids
  - pick points randomly
- Pick medoid and non-medoid point at random
- Evaluate quality of swap
  - Mean point happiness
- Accept random swap if it improves cluster quality

Comparison: Cost of K-Means

- n cases; d dimensions; k centers; i iterations
- compute distance each point to each center: O(n*d*k)
- assign each of n cases to closest center: O(n*k)
- update centers (means) from assigned points: O(n*d*k)
- repeat i times until convergence
- overall: O(n*d*k*i)
- much better than $O(n^2)$-$O(n^3)$ for HAC
- sensitive to initialization - run many times
- usually don't know k - run many times with different k
- requires many passes through data set

Clustering Summary

- Goal
  - Find groups of similar objects in complex data
- Agglomerative clustering
  - Finds a hierarchy
  - Can cut clustering tree at any number of clusters
  - Works with both vector and similarity data
  - Expensive: O(N2) cost
- K-Means clustering
  - Must be told how many clusters to find
  - Usually requires vector data
  - Faster with large data sets
  - Gets stuck in local minima --- run multiple times and pick best

Thank You!