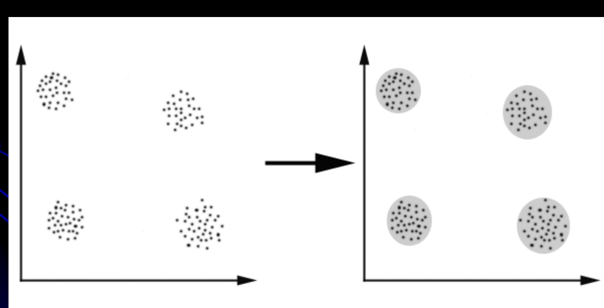
Clustering and You

Evan Clark

April 5, 2005

What is Clustering?

Identification of related samples within a sample space



Some Uses for Clustering

- Ontology generation
  - Disease classification
  - Life science taxonomies
- Data indexing
  - Library layout
  - Web document grouping
- Data mining
  - Political vote analysis
  - Customer classification
  - Marketing surveys

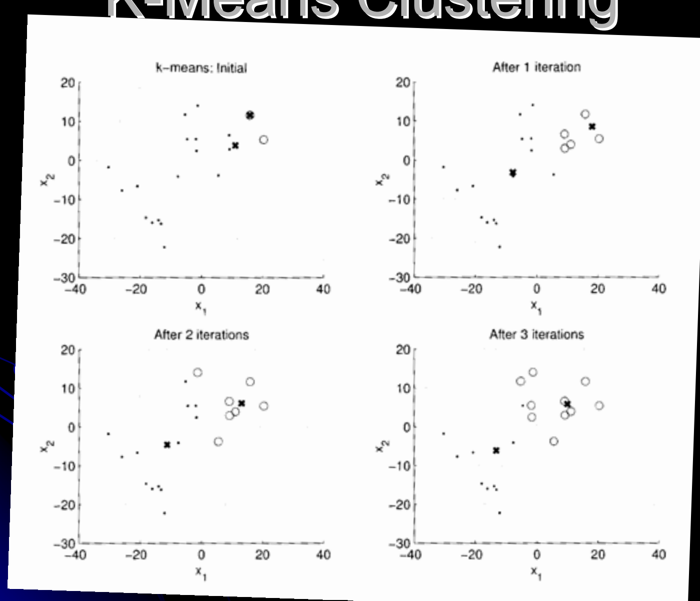
Clustering vs Classification

- In classification we had *n* samples, each with a label. The object was to learn these samples so we could label new samples. In clustering, you not only learn how to classify new samples, but you learn the labels as well

K-Means Clustering

1. Choose *k* points from the sample space
2. For each point, identify the samples that are closer to that point than to any other. These samples are a cluster
3. Move each point to the centroid of the corresponding cluster
4. Go back to 2. Repeat until the clusters stabilize

K-Means Clustering



K-Means Clustering – Thought Exercises

- How dependent is cluster selection on the initial choices of the *k* centroids?
- Is it possible to end up with an empty cluster?
- What are some reasonable choices for starting values of the *k* centroids?

K-Means Clustering – Choosing Initial Centroids

- Randomly select *k* samples
- Use *k* small, random offsets from the center of the sample space
- Place them evenly distributed in the the sample space

K-Means Clustering – Vector quantization

We can use k-means clustering to digitize color images or to compress existing digital images. Suppose you had a high fidelity image you wanted to store in much smaller bit map.

- Each pixel in your source image has 32 bits of color—or 4.3 billion color choices
- Each pixel in your target representation has 8 pixels—or 256 color choices

How do you choose which 256 colors to use? Ideally, you want each pixel in your target to be as close as possible to the corresponding pixel in the source.

Ideas?

K-Means Clustering – Vector quantization

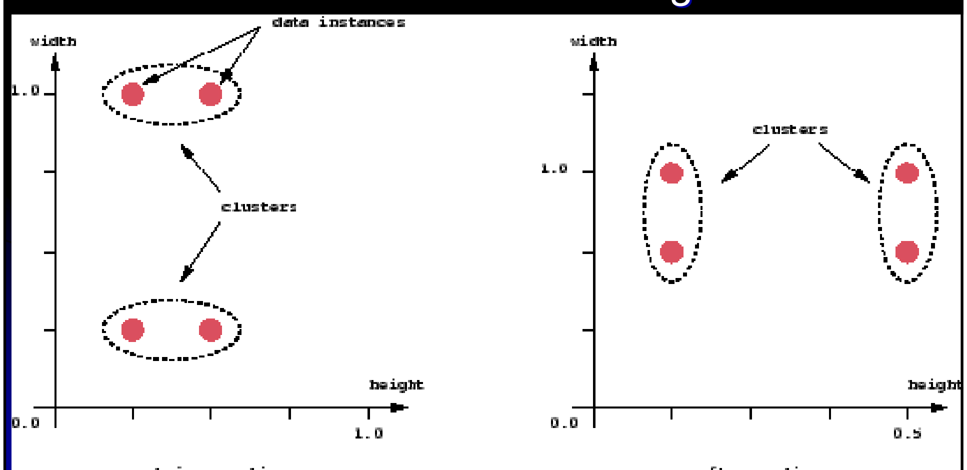
- The source color palette is our sample space
- Each pixel in the source image is a point in our sample space
- K is 256. We'll have 1 cluster per color in our target representation
- After we apply the k-means algorithm, we'll have 256 points in the sample space
- We'll end up with 256 32 bit colors which give (hopefully) a near optimal compression

Types of Clustering Algorithms

- Exclusive (k-means)
- Overlapping (fuzzy c-means)
- Hierarchical
- Probabilistic

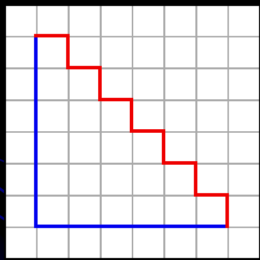
Distance

Scale matters a lot in determining clusters

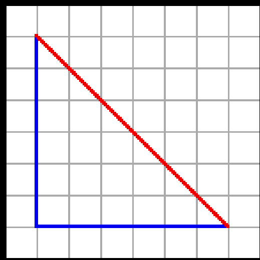


Scaling, Minkowski Distance

$$d = \left[\sum (v_i - v_i')^r\right]^{1/r}$$



r=1.0 Manhattan distance



r=2.0 Euclidean distance

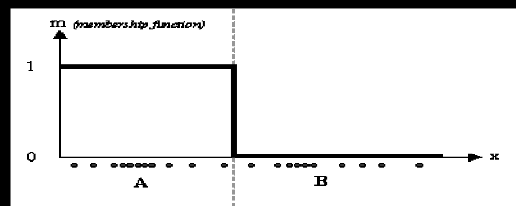
Fuzzy C-Means Clustering

1. Choose random cluster centers
2. For each data point, assign it full membership to the closest cluster center
3. Update the cluster center by using the (weighted) average location of all the member
4. For each data point, assign it membership to each cluster inversely proportional to the distance to the cluster
5. Repeat steps 3 and 4 until the iterative improvement in any membership is below some threshold
6. This algorithm converges to a local minimum

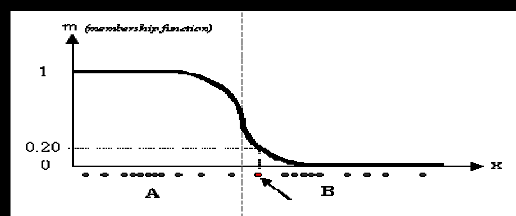
Fuzzy C-Means Clustering

1D data set

k-means result

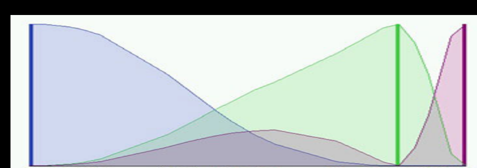


fuzzy c-means result

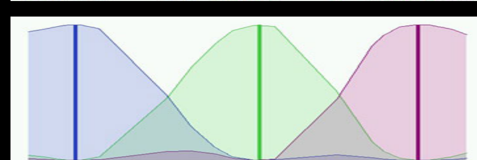


Fuzzy C-Means Clustering

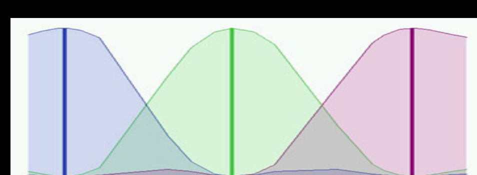
Initial step



After 8 iterations



After 37 iterations



Hierarchical Clustering - Agglomerative

1. Assign each data point to its own cluster
2. Find the closest 2 clusters and merge them
3. Calculate the distance between each cluster pair
4. Repeat steps 2 and 3 until the entire set is in one cluster

Hierarchical Clustering – variations on step 3

- *Single-linkage* – use shortest distance from any member to any member
- *Average-linkage* – use the average distance of all members to all members
- *Complete-linkage* – use the greatest distance from any member to any member

Hierarchical Clustering

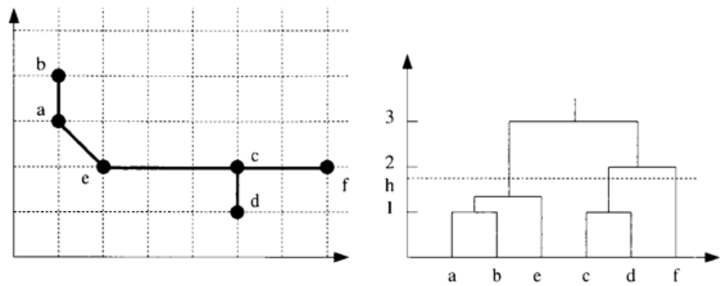


Figure 7.5 A two-dimensional dataset and the dendrogram showing the result of single-link clustering is shown. Note that leaves of the tree are ordered so that no branches cross. The tree is then intersected at a desired value of h to get the clusters.

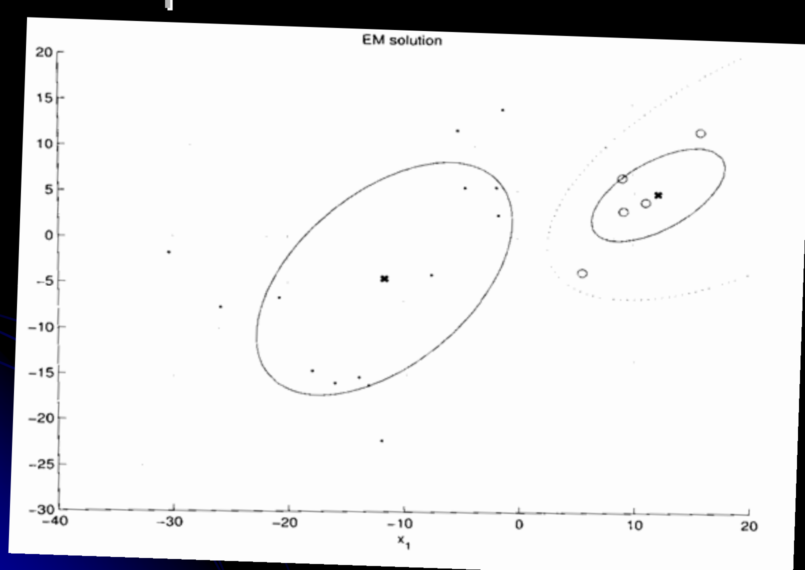
Mixture Model Clustering

- Each cluster is described by a “mixture” of probability distributions such as the Gaussian or Poisson
- The set of distribution parameter vectors fully defines the cluster

Expectation Maximization Algorithm

1. Start by doing several iterations of the k-means algorithm
2. For each cluster, choose random values for our expectation parameter vector, Φ.
3. Use Φ to calculate our 'soft' labels
4. Incrementally improve our choice of Φ by choosing a Φ that maximizes the likelihood of our labels being correct
5. Return to 3. Repeat until Φ stabilizes.

Expectation Maximization



Expectation Maximization Algorithm comparison

- K-means clustering is a special case of Expectation Maximization
- K-means clustering is based on circular areas around centroids (since it uses distance). Expectation maximization uses ellipses of arbitrary shape (since it uses a covariance matrix)

Expectation Maximization Algorithm comparison

- K-means clustering assumes that each point is independent. Expectation maximization allows samples to be probabilistically related via hidden variables
- In k-means clustering, labels are a ‘hard’ 0 or 1. In expectation maximization, labels are based on the probability of being in a cluster. These labels are ‘soft’ values between 0 and 1.

References

- http://locutus.ucr.edu/~kevin/minkowski.html
- http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/
- Introduction to Machine Learning – Ethem Alpaydin