GENERAL ASSEMBLY

INTRO TO DATA SCIENCE
LECTURE 11: CLUSTERING

Paul Burkard

12/02/2015

LAST TIME:

- SUPPORT VECTOR MACHINES
- MAXIMUM MARGIN HYPERPLANE
- SVM CLASSIFICATION
- SOFT-MARGIN SVM
- NONLINEAR SVM

QUESTIONS?

I. CLUSTER ANALYSIS
II. K-MEANS CLUSTERING
III. OTHER CLUSTERING ALGORITHMS
IV. CLUSTER EVALUATION
HANDS-ON: CLUSTERING

I. CLUSTER ANALYSIS

	continuous	*categorical*
supervised	*regression*	*classification*
unsupervised	*dim reduction*	*clustering*

supervised	*making predictions*
unsupervised	*discovering patterns*

supervised	*labeled examples*
unsupervised	*no labeled examples*

Q: *So what is cluster analysis?*

A: **Unsupervised learning algorithms** *that seek to discover patterns in data by grouping unlabeled observations into coherent subsets.*

recall:

unsupervised – searching for patterns, no labels on target variable

Q: *So what is cluster analysis?*

A: **Unsupervised learning algorithms** *that seek to discover patterns in data by grouping unlabeled observations into coherent subsets.*

Clustering provides a layer of abstraction from individual data points.

The goal is to enhance the natural structure of the data (not to impose arbitrary structure!)

Q: *When should we use cluster analysis?*
A: *Clustering is often useful in the* **data exploration stage** *of the data analysis pipeline to get a better feel for your data.*

Does it have inherent groups of observations?

Do these groups have different behaviors for building further models?

Can I build better models by taking these groups into consideration?

Q: *What is a cluster?*

A: *A group of similar data points.*

The concept of similarity is central to the definition of a cluster, and therefore to cluster analysis.

Q: *How do you solve a clustering problem?*

A: *Think of a cluster as a "potential class"; then the solution to a clustering problem is to programmatically determine these classes.*

The real purpose of clustering is data exploration, so a solution is anything that contributes to your understanding.

II. K-MEANS CLUSTERING

Q: *What is K-Means Clustering?*

A: *Probably the most famous* **clustering algorithm**, *a* **greedy** *learner that* **partitions** *a data set into k clusters.*

greedy - only makes locally optimal decisions
partitions - each point belongs to one cluster (usually)

Q: *How are K-Means partitions determined?*

A: *Each point is assigned to the cluster with the nearest* **centroid**.

centroid – the mean of the data points in a cluster

→ *requires continuous (vector-like) features*

→ *highlights iterative nature of algorithm*

One important point to keep in mind is that partitions are not scale-invariant!

This means that the same data can yield very different clustering results depending on the scale and the units used.

Therefore it's important to think about your data representation before applying a clustering algorithm.

1) *choose k initial centroids (note that k is an input)*

2) *for each point:*
   - *find distance to each centroid*
   - *assign point to nearest centroid*

3) *recalculate centroid positions*
4) *repeat steps 2-3 until stopping criteria met*

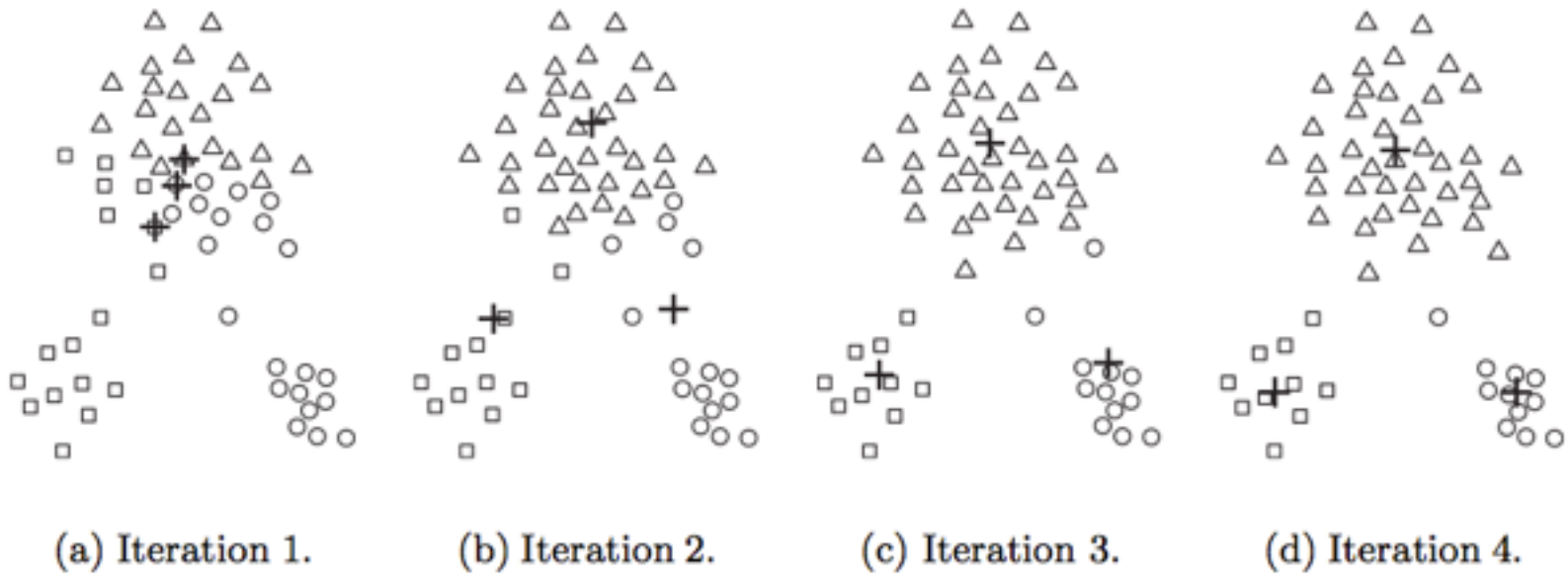


Figure 8.3. Using the K-means algorithm to find three clusters in sample data.

source: http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf

K-means is algorithmically pretty efficient (time & space complexity is linear in number of records).

Requires user to specify the starting centroids and the number of clusters.

So how do we choose the starting centroids?

Q: How do you choose the initial centroid positions?

A: There are several options:

- randomly (but may yield divergent behavior)

- perform alternative clustering task, use resulting centroids as initial k-means centroids

Q: How do you determine which centroid is the nearest?

The "nearness" criterion is determined by the similarity/distance measure we discussed earlier.

This measure makes quantitative inference possible.

There are a number of different similarity measures to choose from, and in general the right choice depends on the problem.

For data that takes values in $\mathbb{R}^n$, the typical choice is the **Euclidean distance***:*

$$d(x, y) = \sqrt{\sum (x_i - y_i)^2}$$

We can express different semantics about our data through the choice of metric.

The matrix whose entries D_{ij} contain the values $d(x, y)$ for all x and y is called the **distance matrix**.

The distance matrix contains all of the information we know about the dataset.

For this reason, it's really the features and choice of metric that determines the definition of a cluster.

Q: How do we recompute the positions of the centroids at each iteration of the algorithm?

A: By optimizing an **objective function** *that tells us how "good" the clustering is.*

The iterative part of the algorithm (recomputing centroids and reassigning points to clusters) explicitly tries to minimize this objective function.

Ex: Using the Euclidean distance measure, one typical objective function is the **sum of squared errors** *from each point x to its centroid c_i:*

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} d(x, c_i)^2$$

Given two clusterings, we will prefer the one with the lower SSE since this means the centroids have converged to better locations (a better local optimum).

We iterate until some stopping criteria are met; in general, suitable convergence is achieved in a small number of steps.

III. OTHER CLUSTERING ALGORITHMS

There are as many as hundreds of different clustering algorithms.

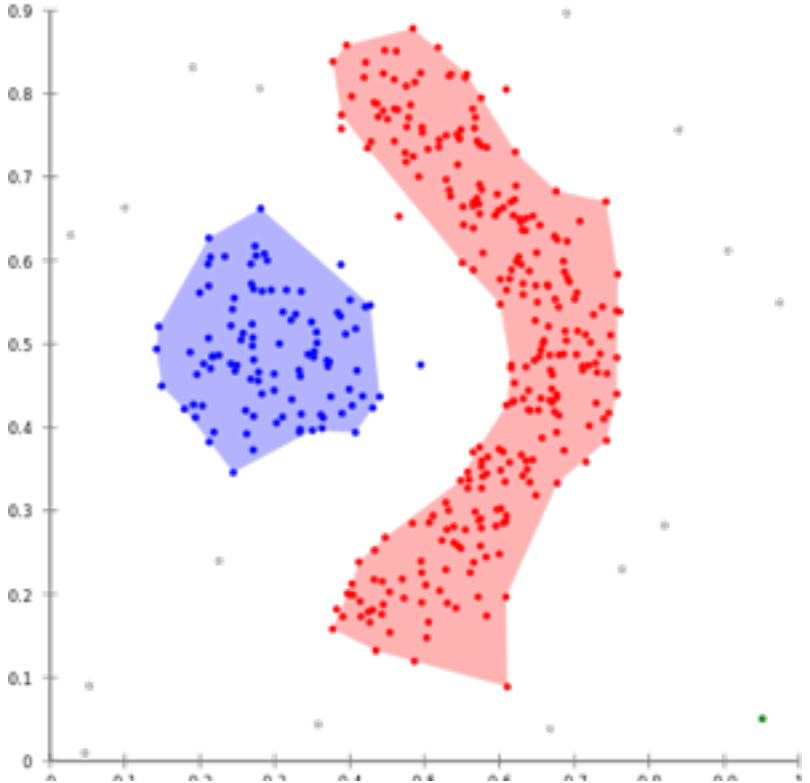
They generally fall into a handful of classes:

- *Density-Based Clustering*
- *Hierarchical Clustering (Connective Models)*
- *Distribution-Based Clustering*
- *Graphical Models*

In **density-based clustering**, *clusters are defined as areas of higher density than the remainder of the data set.*

Objects in these sparse areas - that are required to separate clusters - are usually considered to be noise and border points.

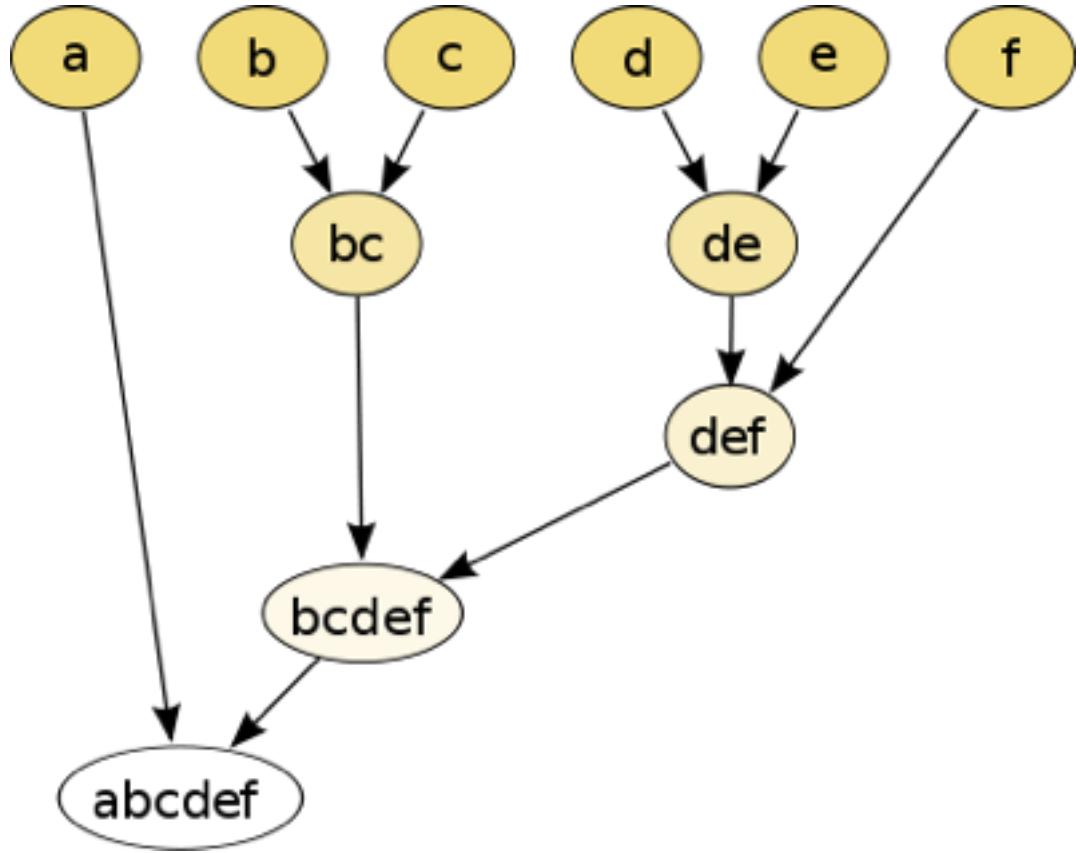
Popular density-based algorithms include **DBSCAN** *and* **OPTICS**


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

Hierarchical Clustering *is a method of cluster analysis which seeks to build a hierarchy of clusters.*

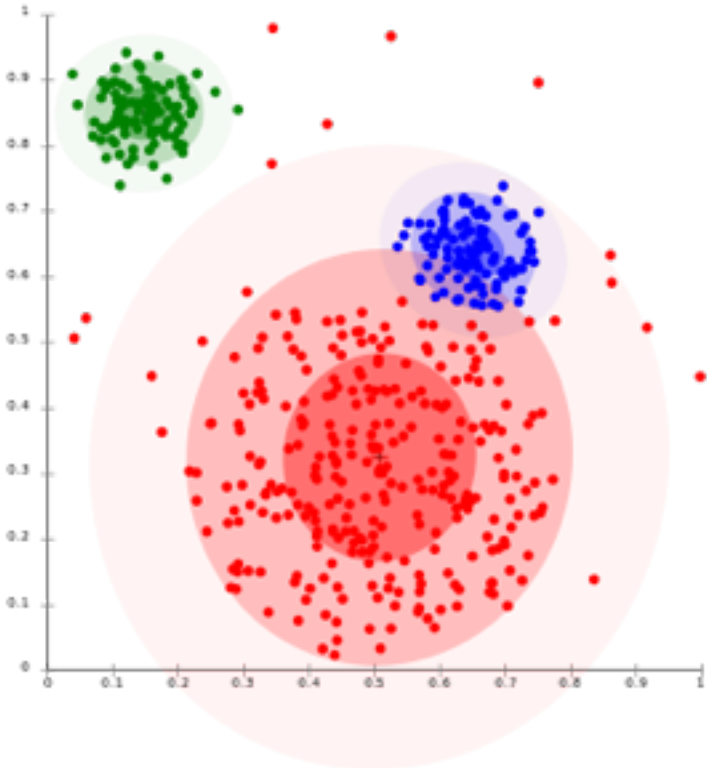
Strategies for hierarchical clustering generally fall into two types:

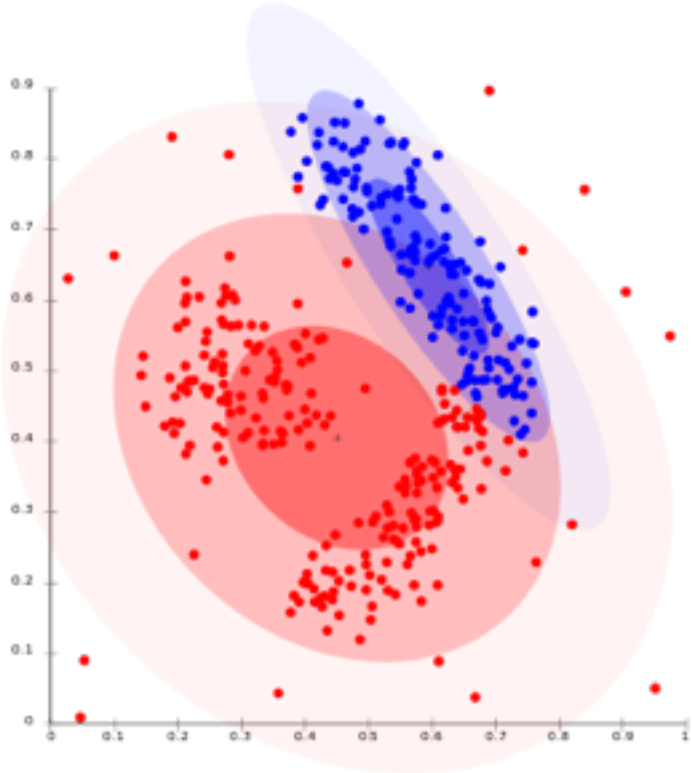
- **Agglomerative:** *This is a "bottom up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.*
- **Divisive:** *This is a "top down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.*


a
b
c
d
e
f
bc
de
def
bcdef
abcdef

In **Distribution models,** *clusters can then easily be defined as objects belonging most likely to the same distribution.*

One prominent method is known as Gaussian mixture models, in which the data set is usually modeled with a fixed (to avoid overfitting) number of Gaussian distributions that are initialized randomly and whose parameters are iteratively optimized to fit better to the data set.


0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

IV. CLUSTER EVALUATION

In general, k-means will converge to a solution and return a partition of k clusters, even if no natural clusters exist in the data.

How do we evaluate the usefulness or performance of our resulting clusters?

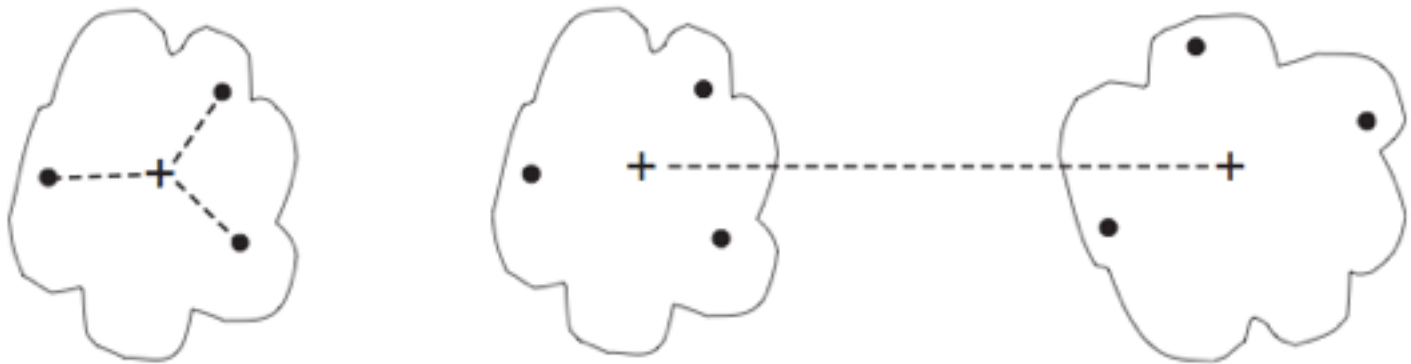
We will look at two validation metrics useful for partitional clustering: **cohesion** *and* **separation**.

Cohesion *measures clustering effectiveness within a cluster.*

$$\hat{C}(C_i) = \sum_{x \in C_i} d(x, c_i)$$

Separation *measures clustering effectiveness between clusters.*

$$\hat{S}(C_i, C_j) = d(c_i, c_j)$$



(a) Cohesion. (b) Separation.

Figure 8.28. Prototype-based view of cluster cohesion and separation.

source: http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf

We can turn these values into overall measures of clustering validity by taking a weighted sum over clusters:

$$\hat{V}_{total} = \sum_{1}^{K} w_i \hat{V}(C_i)$$

Here V can be cohesion, separation, or some function of both.

The weights can all be set to 1 (best for k-means), or proportional to the cluster masses (the number of points they contain).

Cluster evaluation measures can be used to identify clusters that should be split or merged, or to identify individual points with disproportionate effect on the overall clustering.

One useful measure than combines the ideas of cohesion and separation is the **silhouette coefficient**. *For point x_i, this is given by:*

$$SC_i = \frac{b_i - a_i}{max(a_i, b_i)}$$

such that:

a_i = *average in-cluster distance to* x_i

b_{ij} = *average between-cluster distance to* x_i

$b_i = min_j(b_{ij})$

The silhouette coefficient can take values between -1 and 1.

In general, we want separation to be high and cohesion to be low. This corresponds to a value of SC *close to +1.*

A negative silhouette coefficient means the cluster radius is larger than the space between clusters, and thus clusters overlap.

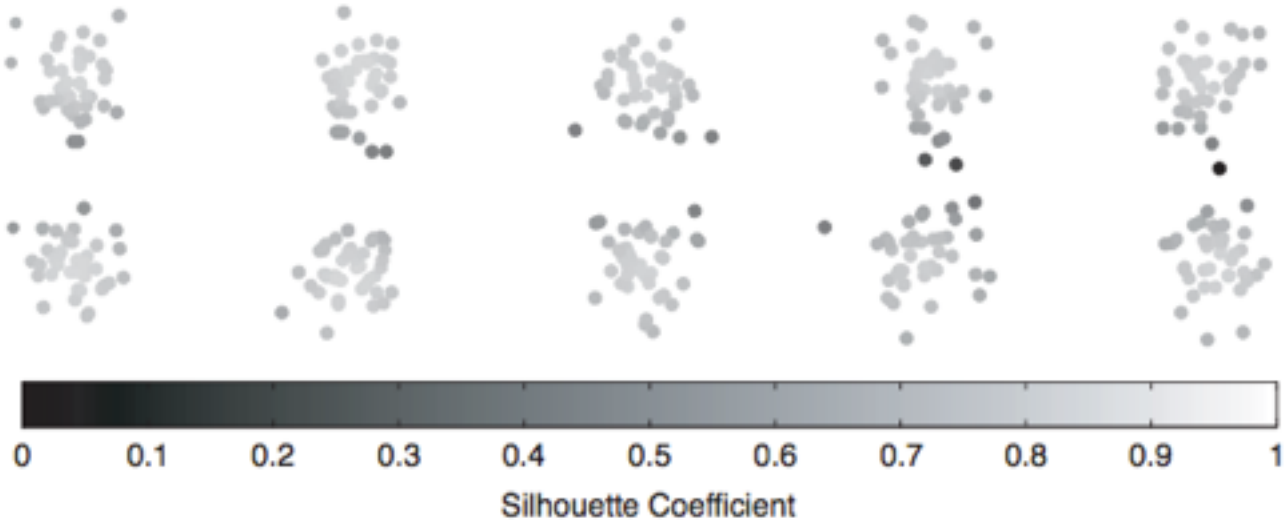


Figure 8.29. Silhouette coefficients for points in ten clusters.

source: http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf

The silhouette coefficient for the cluster C_i is given by the average silhouette coefficient across all points in C_i:

$$SC(C_i) = \frac{1}{m_i} \sum_{x \in C_i} SC_i$$

The overall silhouette coefficient is given by the average silhouette coefficient across all points:

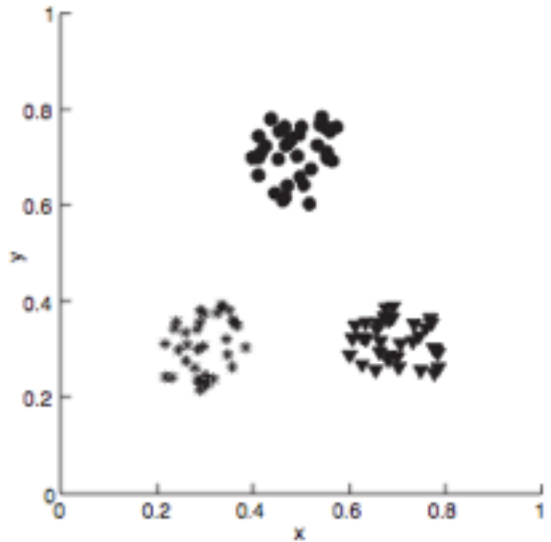
$$SC_{total} = \frac{1}{k} \sum_{1}^{k} SC(C_i)$$

NOTE

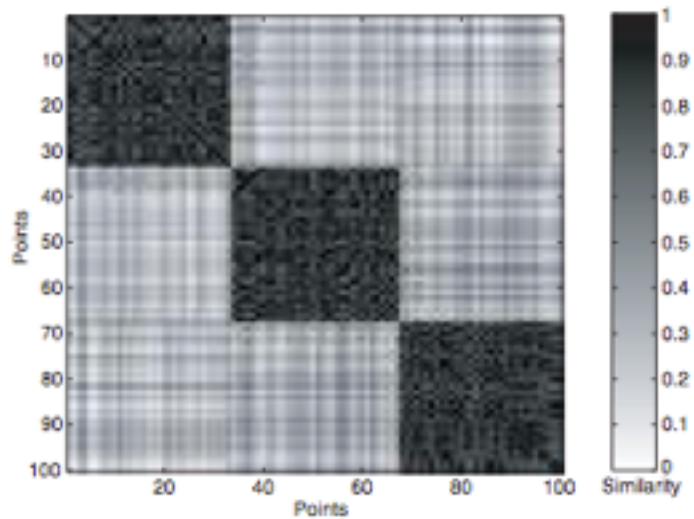
This gives a summary measure of the overall clustering quality.

An alternative validation scheme is given by comparing the similarity matrix with an idealized (0/1) similarity matrix that represents the same clustering configuration.

This can be done either graphically or using correlations.



(a) Well-separated clusters.



(b) Similarity matrix sorted by K-means cluster labels.

source: http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf

One useful application of cluster validation is to determine the best number of clusters for your dataset.

Q: *How would you do this?*

A: *By computing the overall SSE or SC for different values of k.*

Then treat k as a model parameter and **cross-validate***!*

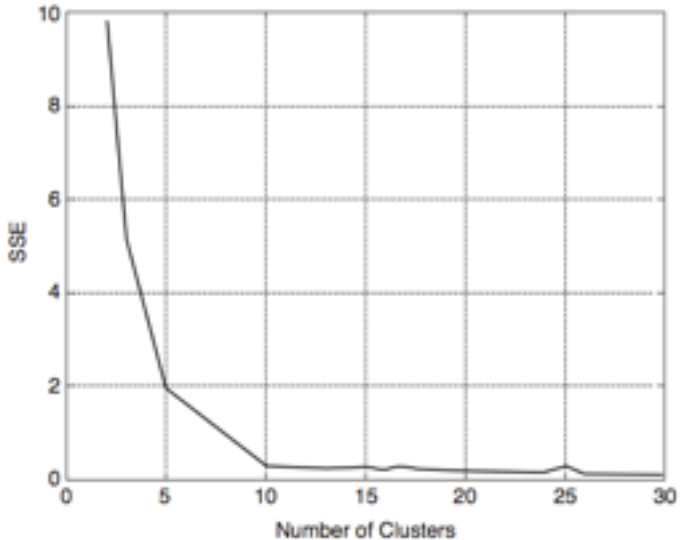


Figure 8.32. SSE versus number of clusters for the data of Figure 8.29.

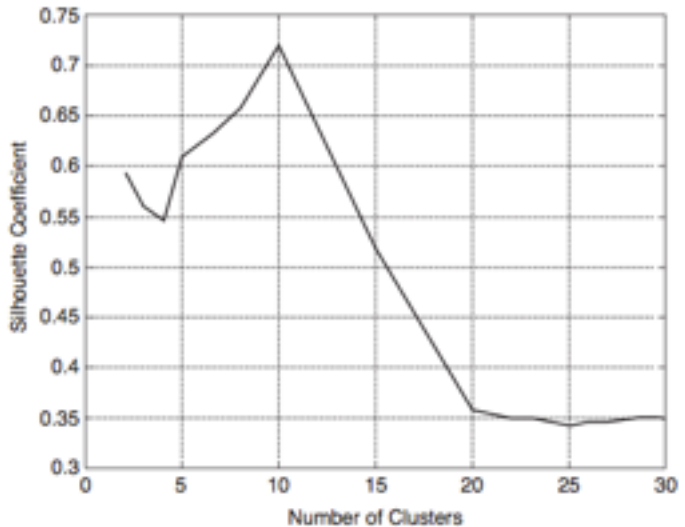


Figure 8.33. Average silhouette coefficient versus number of clusters for the data of Figure 8.29.

source: http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf

Ultimately, cluster validation and clustering in general are suggestive techniques that rely on human interpretation to be meaningful.

HANDS-ON: CLUSTERING