W4995 Applied Machine Learning

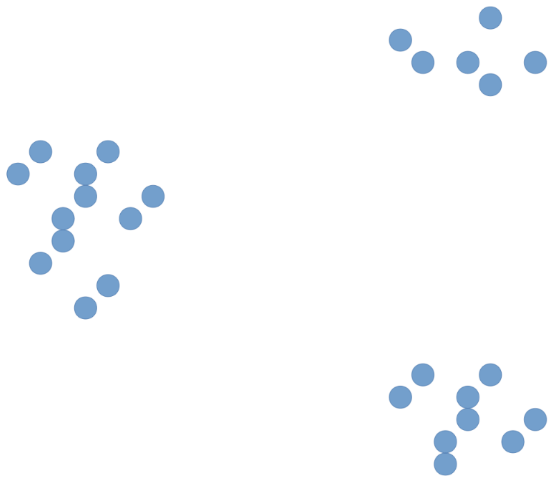
Clustering

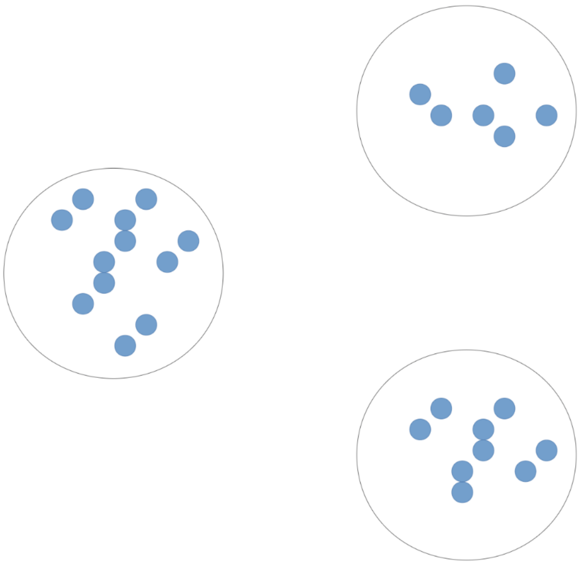
03/27/19

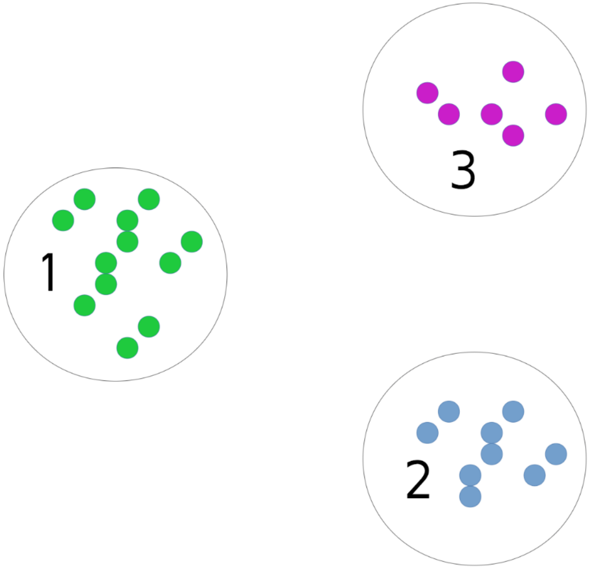
Andreas C. Müller

(Adapted and modified for CC 6021236 @ PCC/Ciencias/UCV by Eugenio Scalise, October 2019)

Clustering

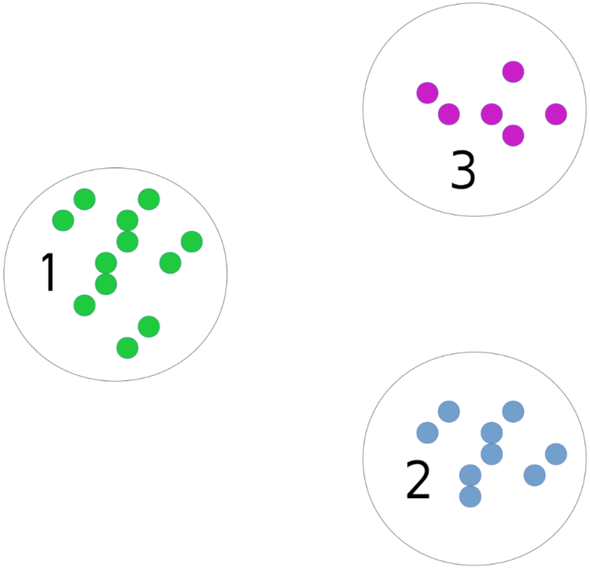


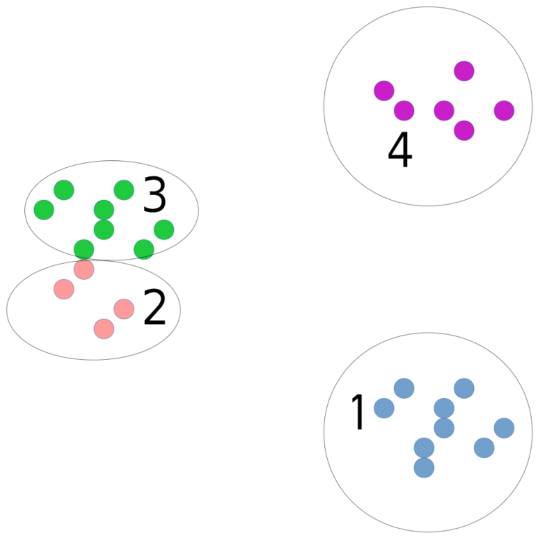



3
1
2

Clustering

- Partition data into groups (clusters)
- Points within a cluster should be “similar”.
- Points in different cluster should be “different”.


1
2
3


4
3
2
1

Goals of Clustering

- Data Exploration
  - Are there coherent groups ?
  - How many groups are there ?

Goals of Clustering

- Data Exploration
  - Are there coherent groups ?
  - How many groups are there ?
- Data Partitioning
  - Divide data by group before further processing

Goals of Clustering

- Data Exploration
  - Are there coherent groups ?
  - How many groups are there ?
- Data Partitioning
  - Divide data by group before further processing
- Unsupervised feature extraction
  - Derive features from clusters or cluster distances

Clustering Algorithms

K-Means

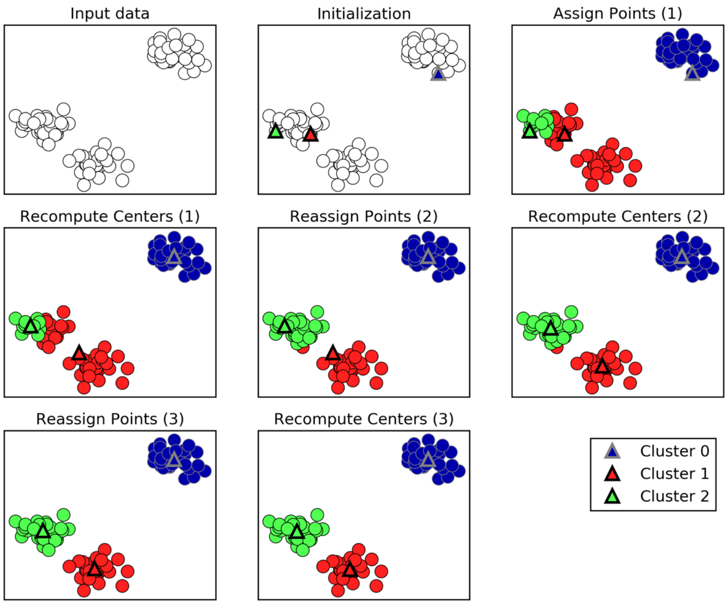
Objective function for K-Means

$$\min_{\mathbf{c}_j \in \mathbf{R}^p, j=1,..,k} \sum_i ||\mathbf{x}_i - \mathbf{c}_{x_i}||^2$$

$\mathbf{c}_{x_i}$ is the cluster center c_j closest to x_i

- This is an NP hard problem, so we can't really hope to solve it exactly (k-means algorithm provide an aproximation).

K-Means algorithm



- Pick number of clusters k.
- Pick k random points as “cluster center”
- While cluster centers change:
  1. Assign each data point to it’s closest cluster center
  2. Recompute cluster centers as the mean of the assigned points.

K-Means API

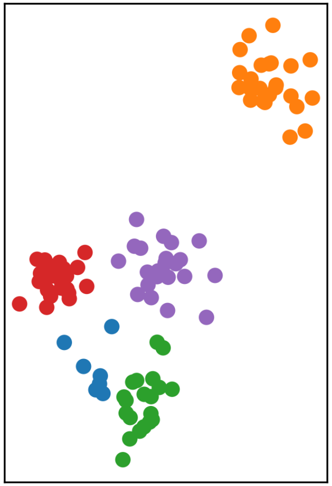
```
X, y = make_blobs(centers=4, random_state=1)

km = KMeans(n_clusters=5, random_state=0)
km.fit(X)
print(km.cluster_centers_.shape)
print(km.labels_.shape)
```

```
(5, 2)
(100,)
```

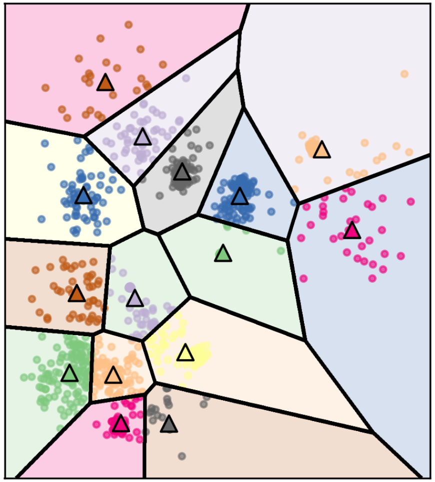
```
# predict is the same as labels_ on training data
# but can be applied to new data
print(km.predict(X).shape)
```

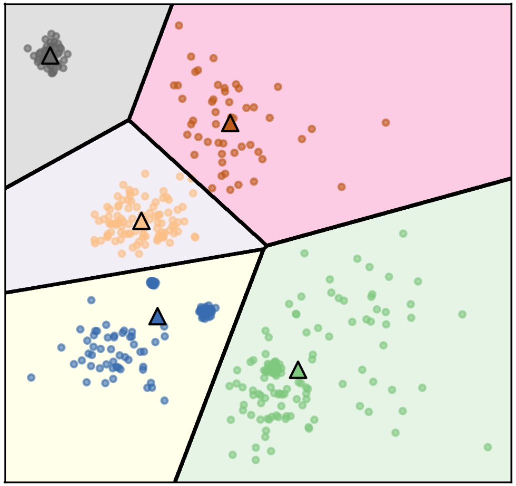
```
(100,)
```

```
plt.scatter(X[:, 0], X[:, 1], c=km.labels_)
```



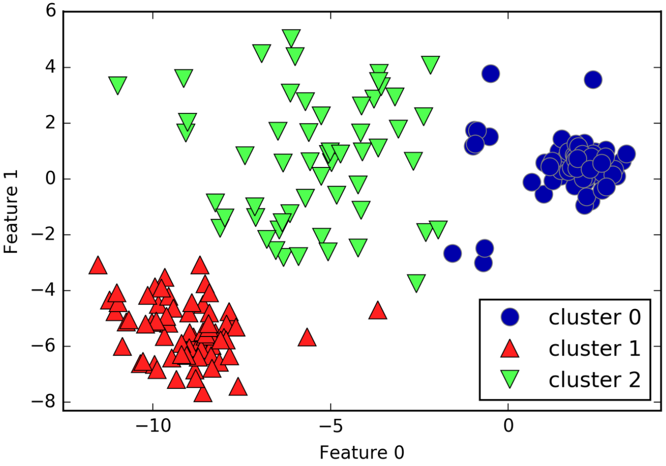
Restriction of Cluster Shapes





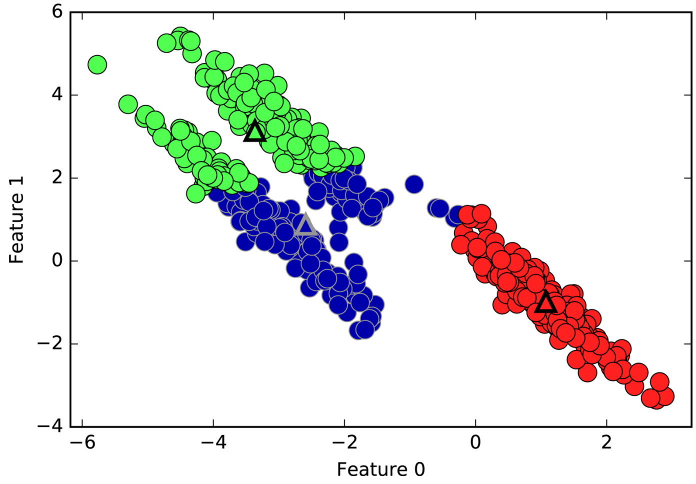
Clusters are Voronoi-diagrams of centers (convex sets)

Limitations of K-Means



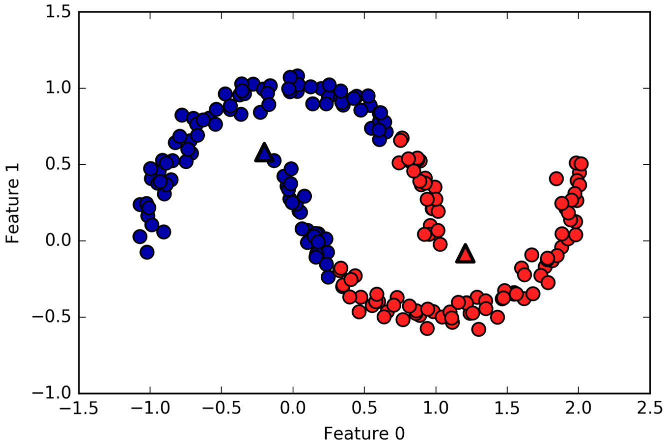
- Cluster boundaries equidistant to centers

Limitations of K-Means



- Can’t model covariances well

Limitations of K-Means

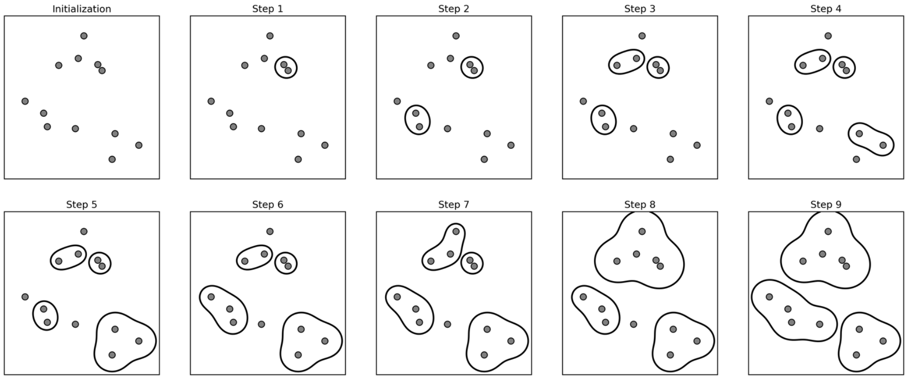


- Only simple cluster shapes

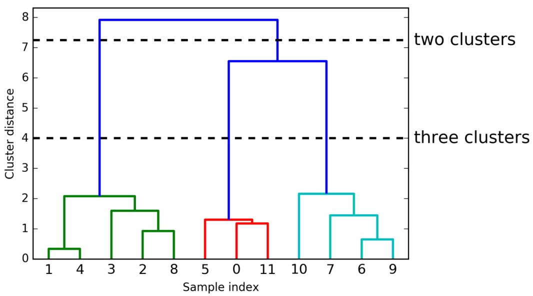
Agglomerative Clustering

Agglomerative Clustering

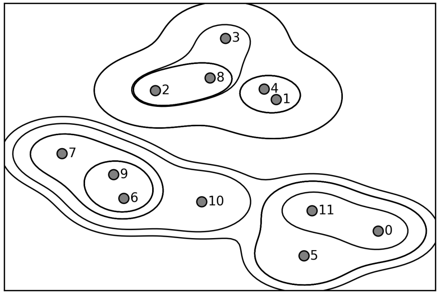
- Start with all points in their own cluster.
- Greedily merge the two most similar clusters.



Dendograms

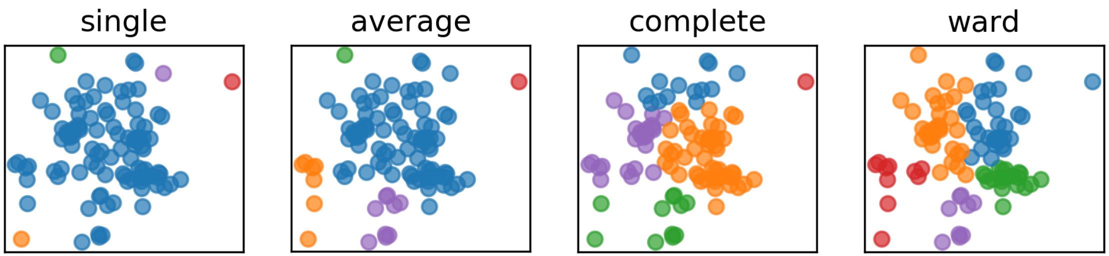


two clusters
three clusters
Cluster distance
Sample index
1 4 3 2 8 5 0 11 10 7 6 9



3
2
8
4
1
7
9
6
10
11
0
5

Linkage Criteria



Cluster sizes:

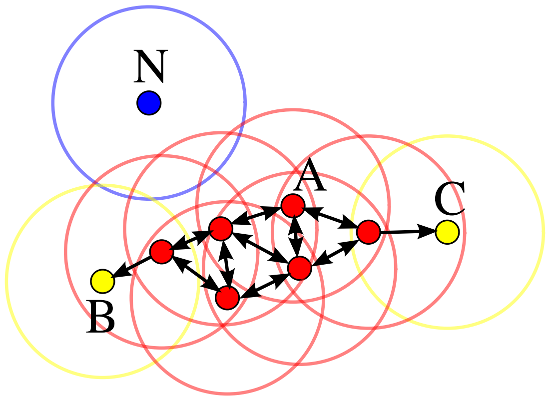
```
single :    [96  1  1  1  1]
average :   [82  9  7  1  1]
complete :  [50 24 14 11  1]
ward :      [31 30 20 10  9]
```

- Single Linkage
  - Smallest minimum distance
- Average Linkage
  - Smallest average distance between all pairs in the clusters
- Complete Linkage
  - Smallest maximum distance
- Ward (default in sklearn)
  - Smallest increase in within-cluster variance
  - Leads to more equally sized clusters.

DBSCAN

(density- based spatial clustering of applications with noise)

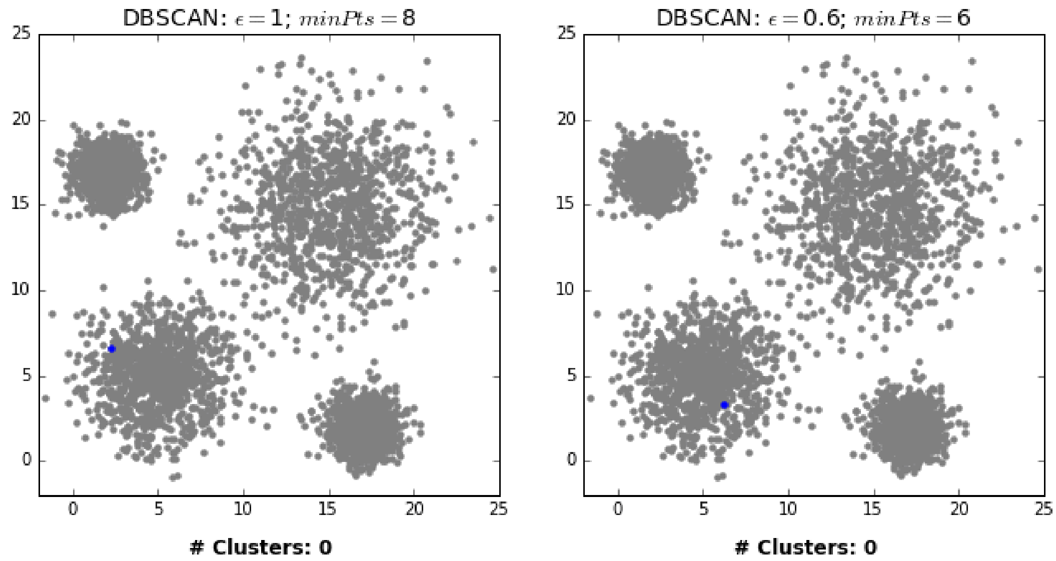
Algorithm



- eps: neighborhood radius
- min_samples: 4

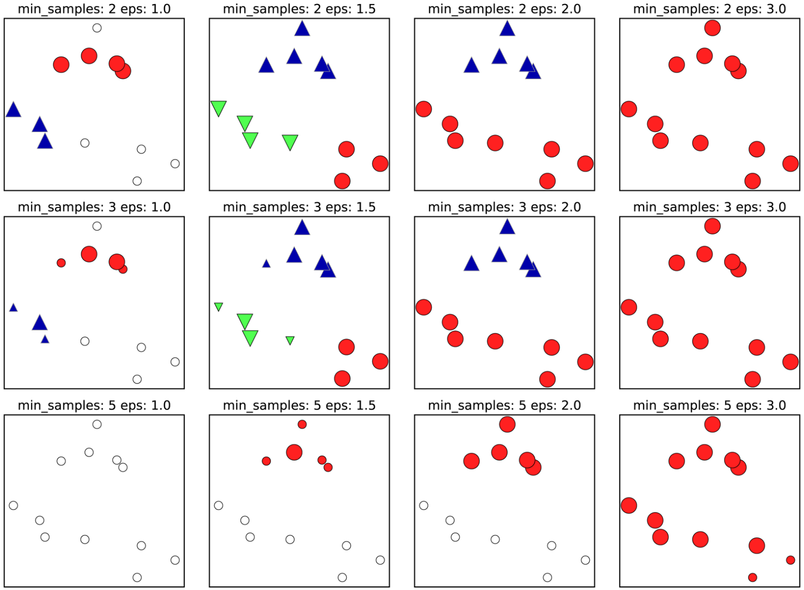
- A: Core
- B, C: not core
- N: noise

Core: sample that has more than (min_sample - 1) other samples in its epsilon neighborhood.



by David Sheehan dashee87.github.io

Illustration of Parameters

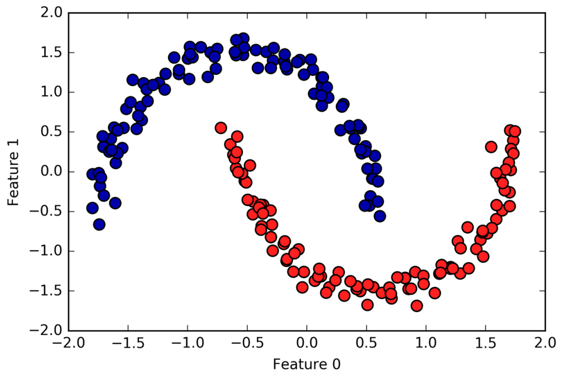


Parameters

- The parameter eps determines what it means for points to be “close”.
- Setting eps to be very small will mean that no points are core samples, and may lead to all points being labeled as noise.
- Setting eps to be very large will result in all points forming a single cluster.
- The min_samples setting determines whether points in less dense regions will be labeled as outliers or as their own clusters.
- min_samples determines the minimum cluster size.

Pros and Cons

- Pro: Can learn arbitrary cluster shapes
- Pro: Can detect outliers
- Con: Needs two (non-obvious?) parameters to adjust



Summary

- KMeans
  Classic, simple. Only convex cluster shapes, determined by cluster centers.

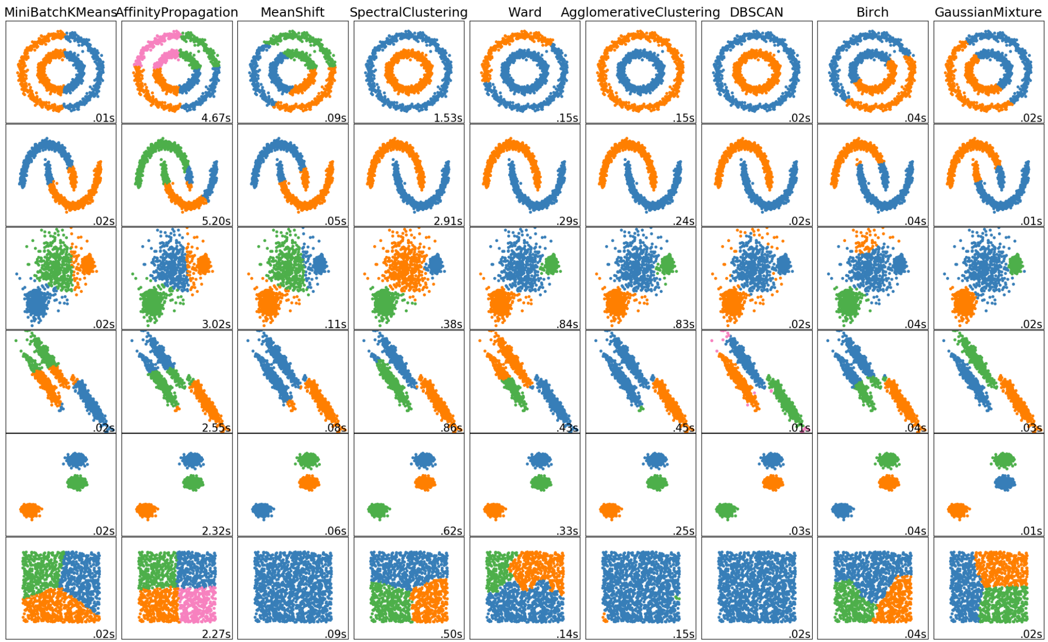
Summary

- KMeans
  Classic, simple. Only convex cluster shapes, determined by cluster centers.

- Agglomerative
  Can take input topology into account, can produce hierarchy.

Summary

- KMeans
  Classic, simple. Only convex cluster shapes, determined by cluster centers.
- Agglomerative
  Can take input topology into account, can produce hierarchy.
- DBSCAN
  Arbitrary cluster shapes, can detect outliers, often very different sized clusters.



http://scikit-learn.org/dev/auto_examples/cluster/plot_cluster_comparison.html

Questions ?