Hierarchical clustering

Ludwig Krippahl

Hierarchical clustering

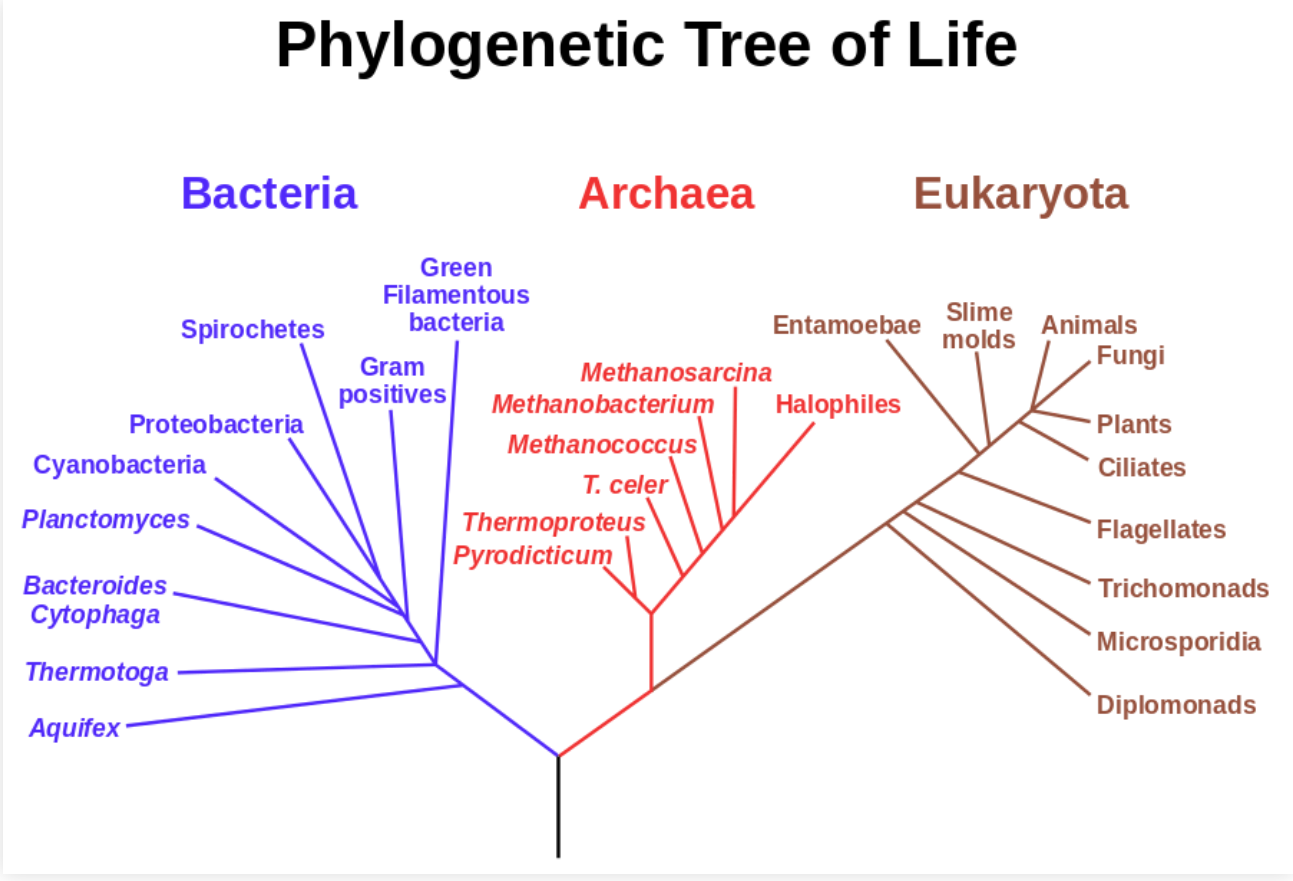
Summary

- Hierarchical Clustering
  - Agglomerative Clustering
- Divisive Clustering
- Clustering Features

Hierarchical clustering

Hierarchical clustering

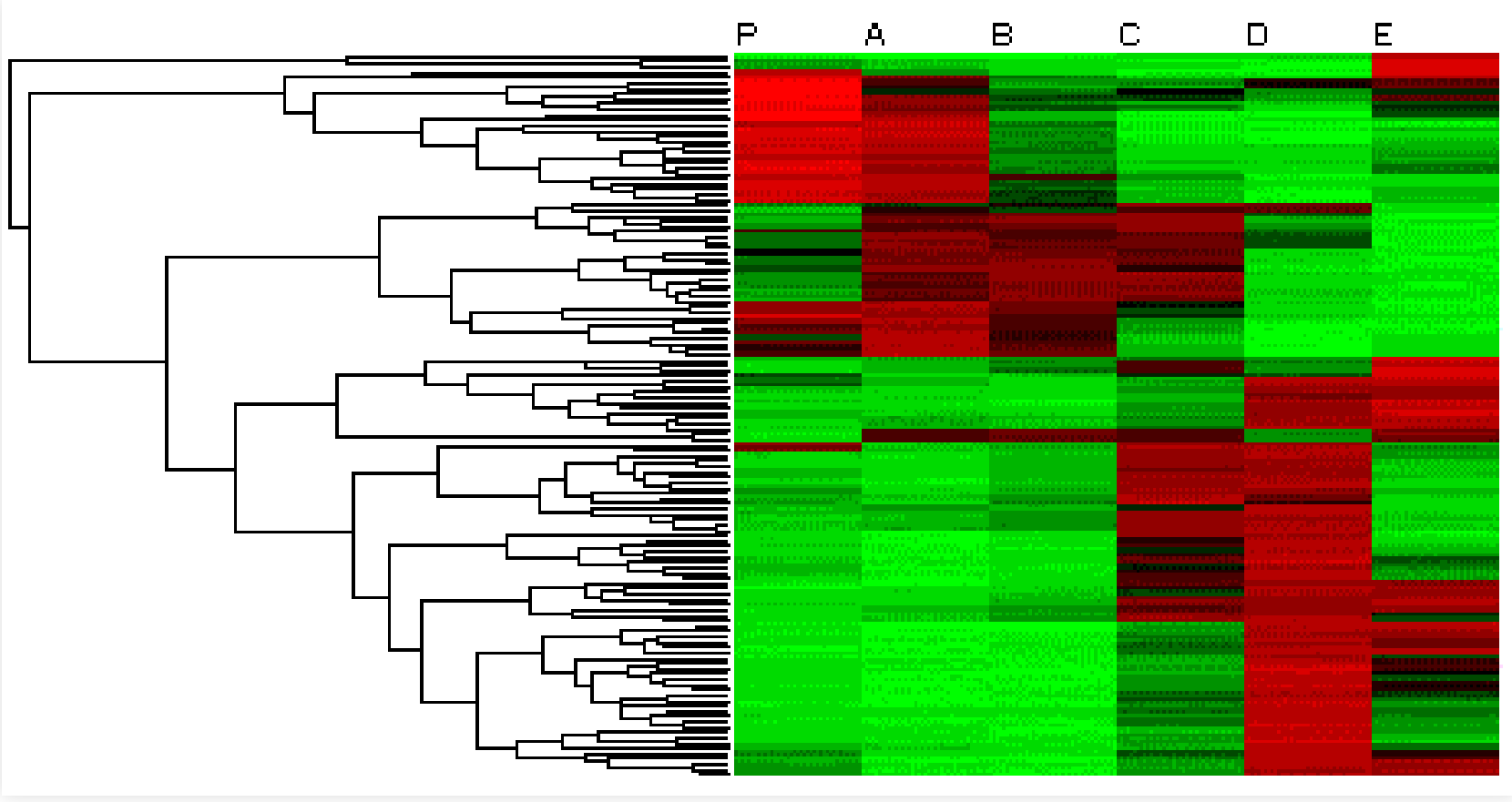
- Grouping groups of groups (...)



Source: Wikipedia

Hierarchical clustering

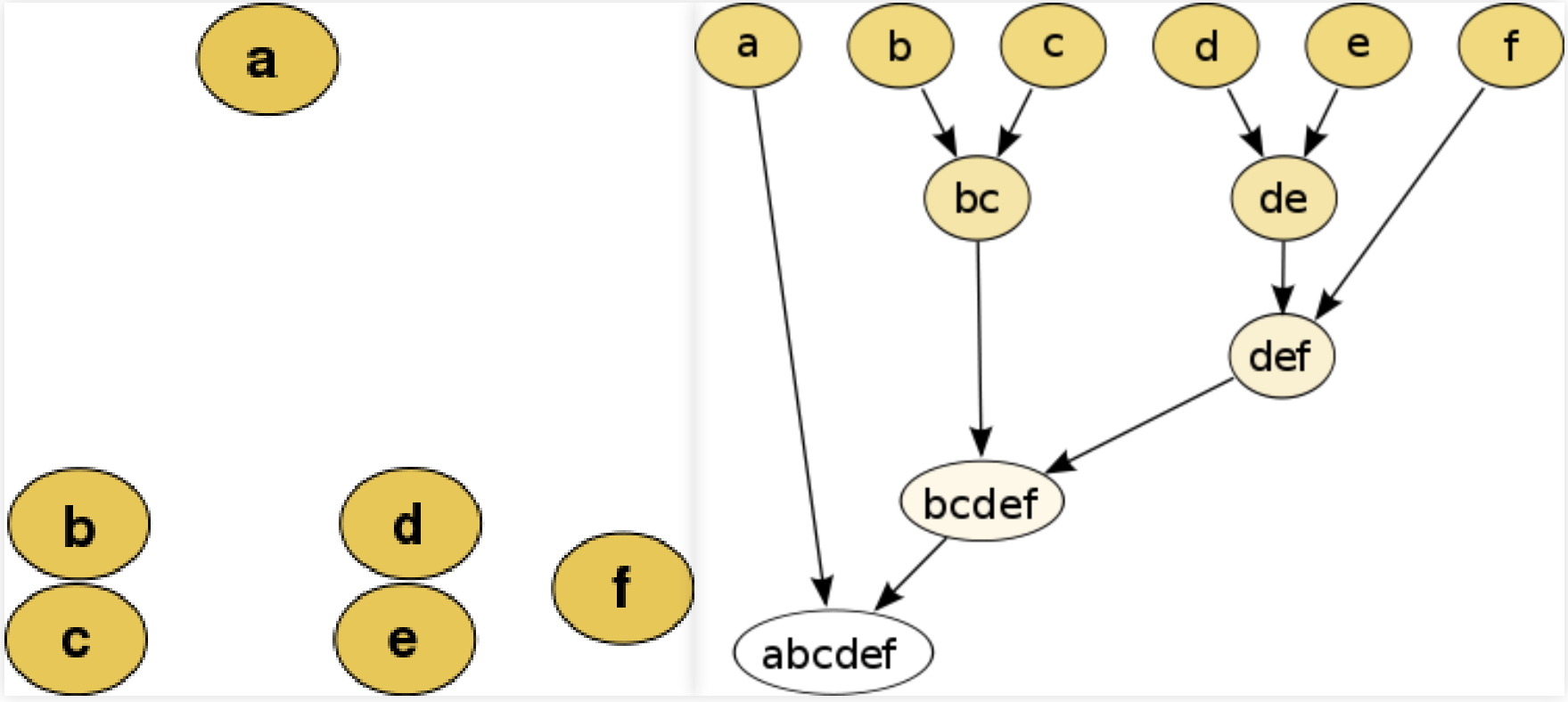
- Clustering gene activity patterns



Source: Wikipedia

Hierarchical clustering

- Can be represented as dendrogram



Source: Wikipedia

Hierarchical clustering

Need to measure how alike examples are:

- Proximity : Generic term for "likeness"
- Similarity : Measure of how alike, generally $\in [0, 1]$
- Dissimilarity : Measure of difference.
- Distance is special case of dissimilarity:

$$d(x, y) \geq 0 \ , \ d(x, y) = d(y, x) \ , \ d(x, z) \leq d(x, y) + d(y, z)$$

Hierarchical clustering

- Some measures of distance between examples:

- Euclidean: $\|x - y\|_2 = \sqrt{\sum_d (x_d - y_d)^2}$
- Squared Euclidean: $\|x - y\|_2^2 = \sum_d (x_d - y_d)^2$
- Manhattan: $\|x - y\|_1 = \sum_d |x_d - y_d|$
- Mahalanobis: Normalized by variance

$$\sqrt{(x - y)^T Cov^{-1} (x - y)}$$

- Hamming: differences between strings

$$d(x, y) = \sum_i x_i \neq y_i$$

- Levenshtein: min number of edits: insertion, substitution, deletion
- (many problem dependent measures)

Linkage

- In hierarchical clustering we need to compare clusters

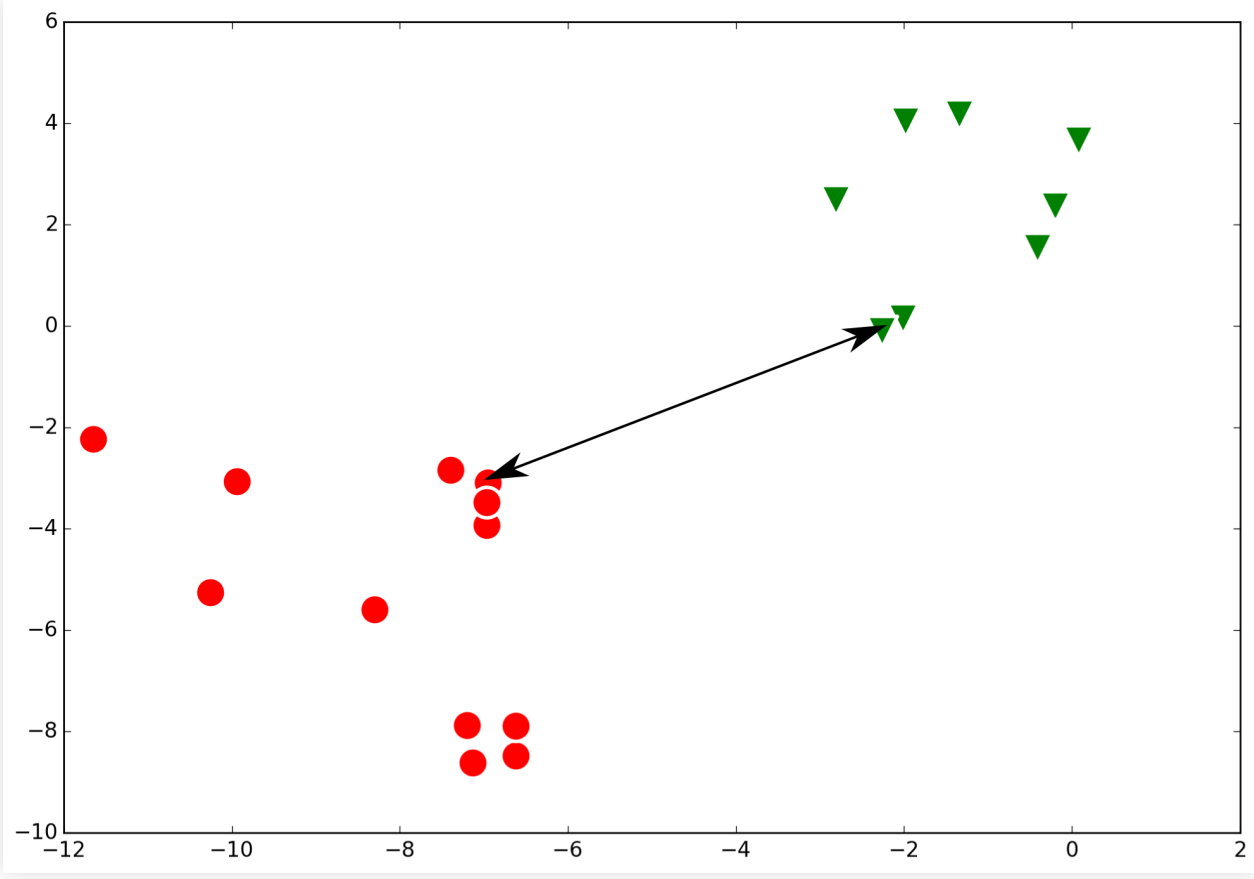
- We need to divide clusters or agglomerate them

- Different linkage criteria:

- Single linkage
- Complete linkage
- Centroid linkage
- Average
- Median
- Ward
- ...

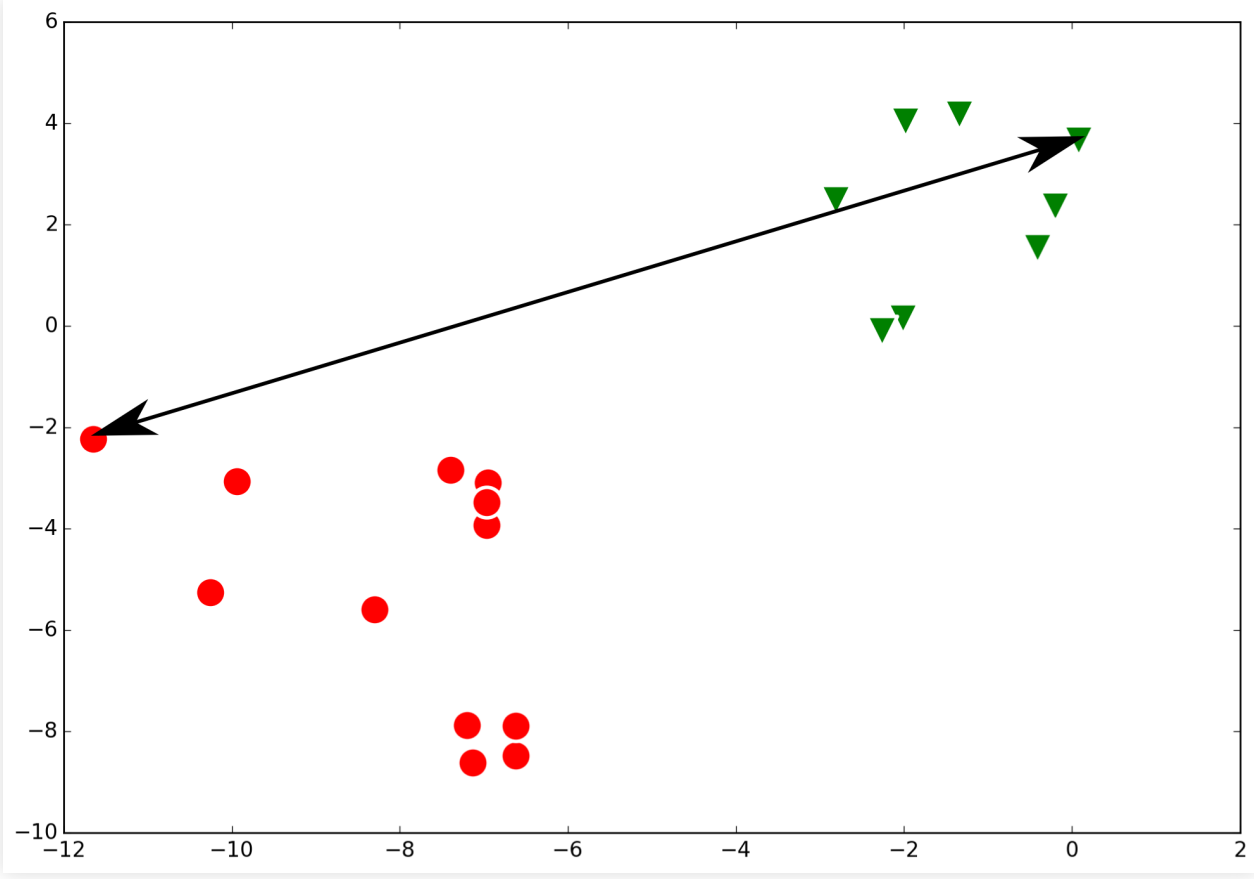
Linkage

- Single linkage : $dist(C_j, C_k) = min\left(dist(x \in C_j, y \in C_k)\right)$



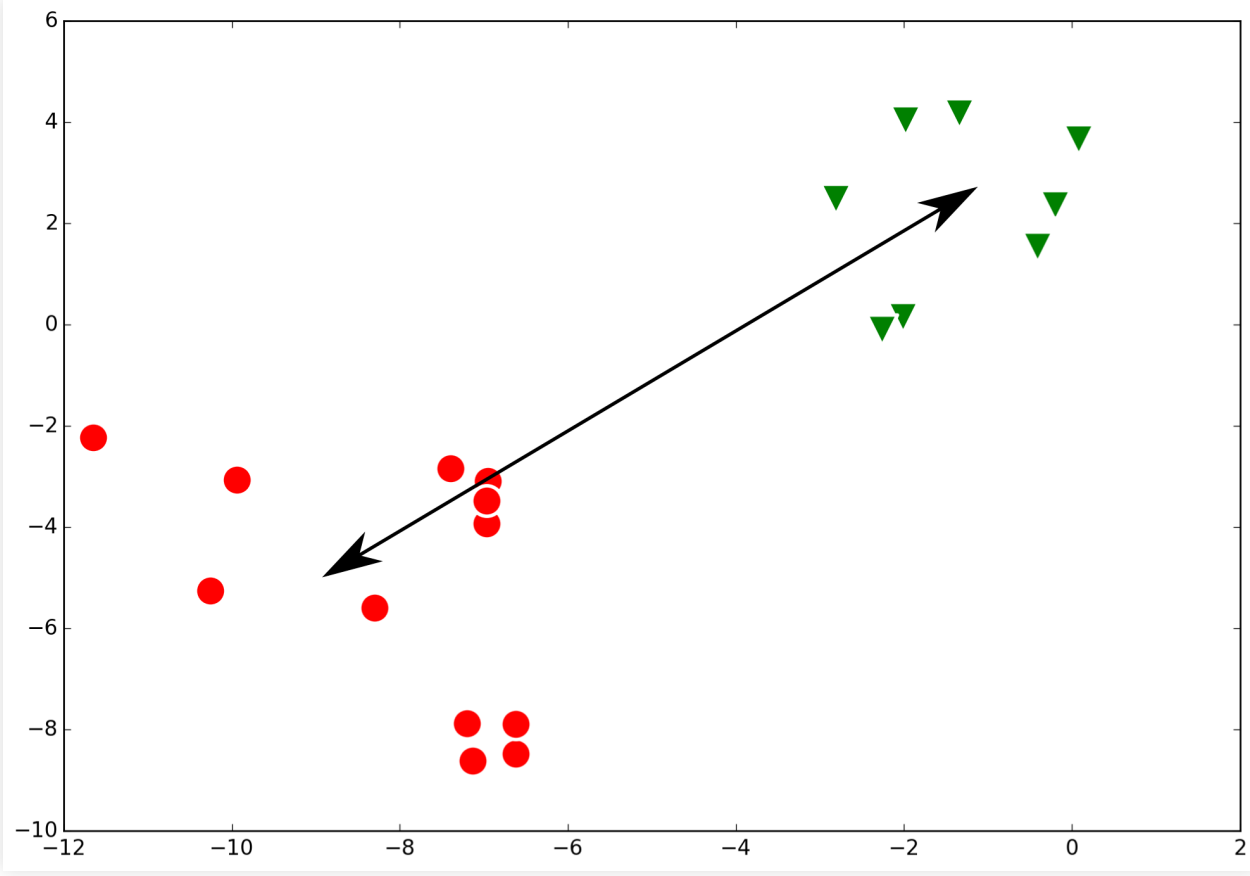
Linkage

- Complete linkage: $dist(C_j, C_k) = max\left(dist(x \in C_j, y \in C_k)\right)$



Linkage

- Centroid linkage : $dist(C_j, C_k) = dist\left(\frac{\sum x \in C_j}{|C_j|}, \frac{\sum y \in C_k}{|C_k|}\right)$



More examples of linkage

- Average linkage

$$dist(C_j, C_k) = mean\left(dist(x \in C_j, y \in C_k)\right)$$

- Median linkage

$$dist(C_j, C_k) = median\left(dist(x \in C_j, y \in C_k)\right)$$

- Ward linkage : minimize SSE

$$\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|x_n - \mu_k\|^2$$

Hierarchical clustering

- Some way to compare examples: distance, similarity, etc...
- Some way to compare clusters (linkage): single, complete, etc...

Advantages:

- No need to assume number of clusters
- Hierarchical organization may correspond to some aspect of the data (e.g. phylogeny)

Disadvantages:

- Single pass, local decisions may be wrong
- Hierarchical organization may be confusing or reflect idiosyncrasies of the clustering algorithm

Hierarchical clustering

Clustering algorithms

- Agglomerative clustering (bottom-up)

- Start with singleton clusters, join best two (linkage), repeat until all joined
- Generally $O(n^3)$, but can be better with linkage constraints

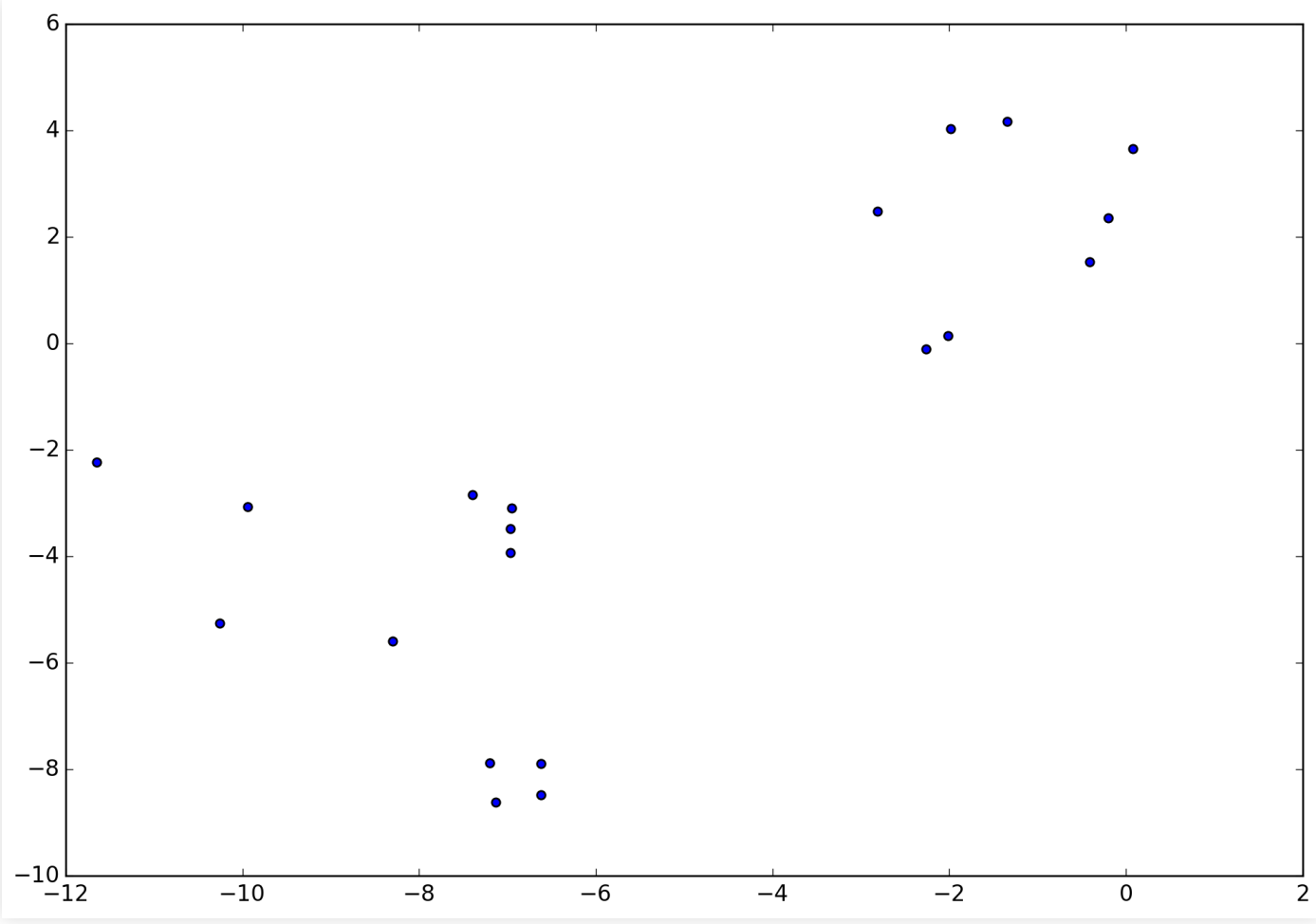
- Divisive clustering (top-down)

- Start with single cluster, pick cluster to split, repeat until all examples separated or level k reached
- Generally $O(2^n)$ for exhaustive search, and needs additional clustering algorithm for splitting.
- But can be better if we only want a few levels of clustering from the top.

Agglomerative Clustering

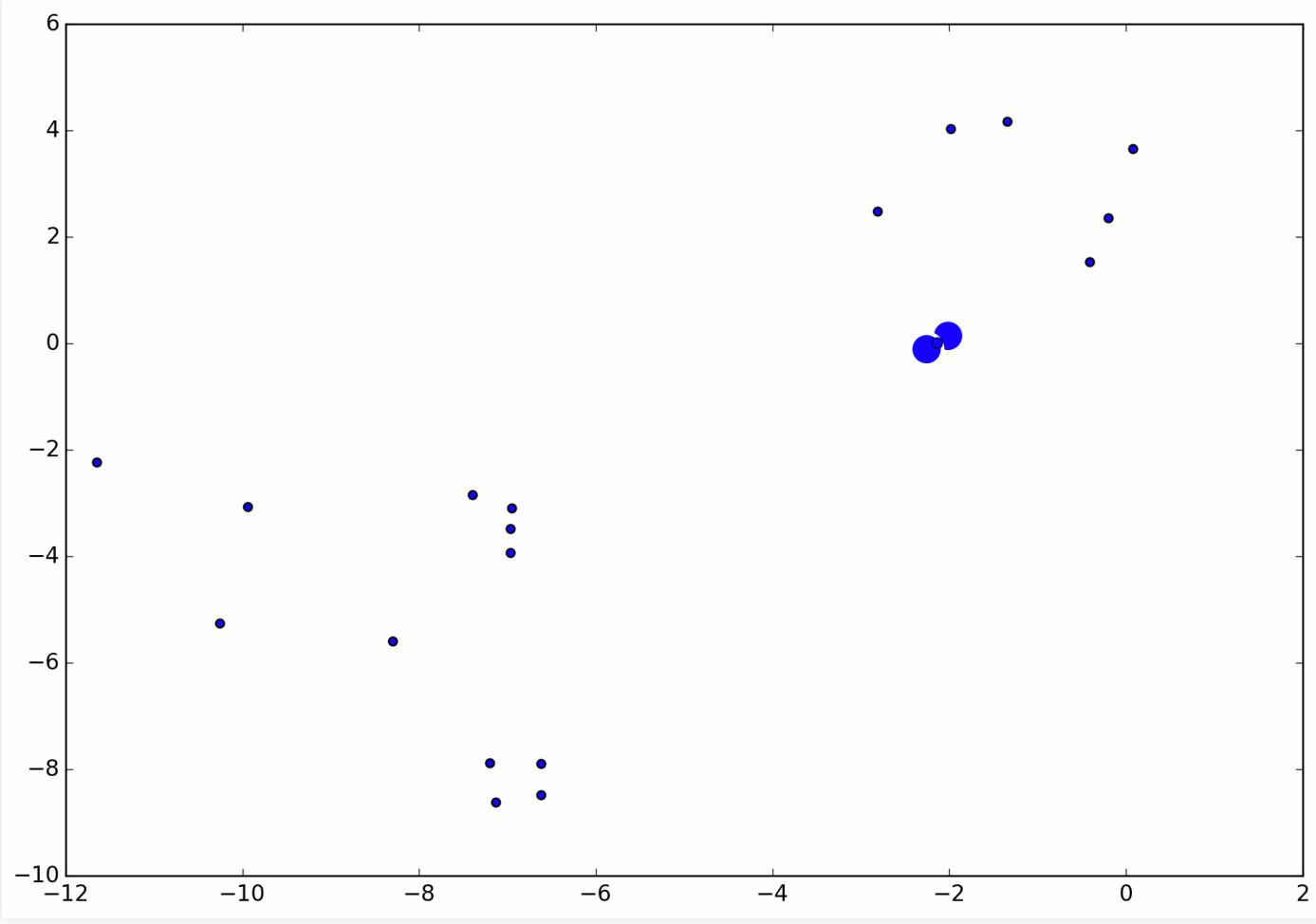
Agglomerative Clustering

- Start with singleton clusters



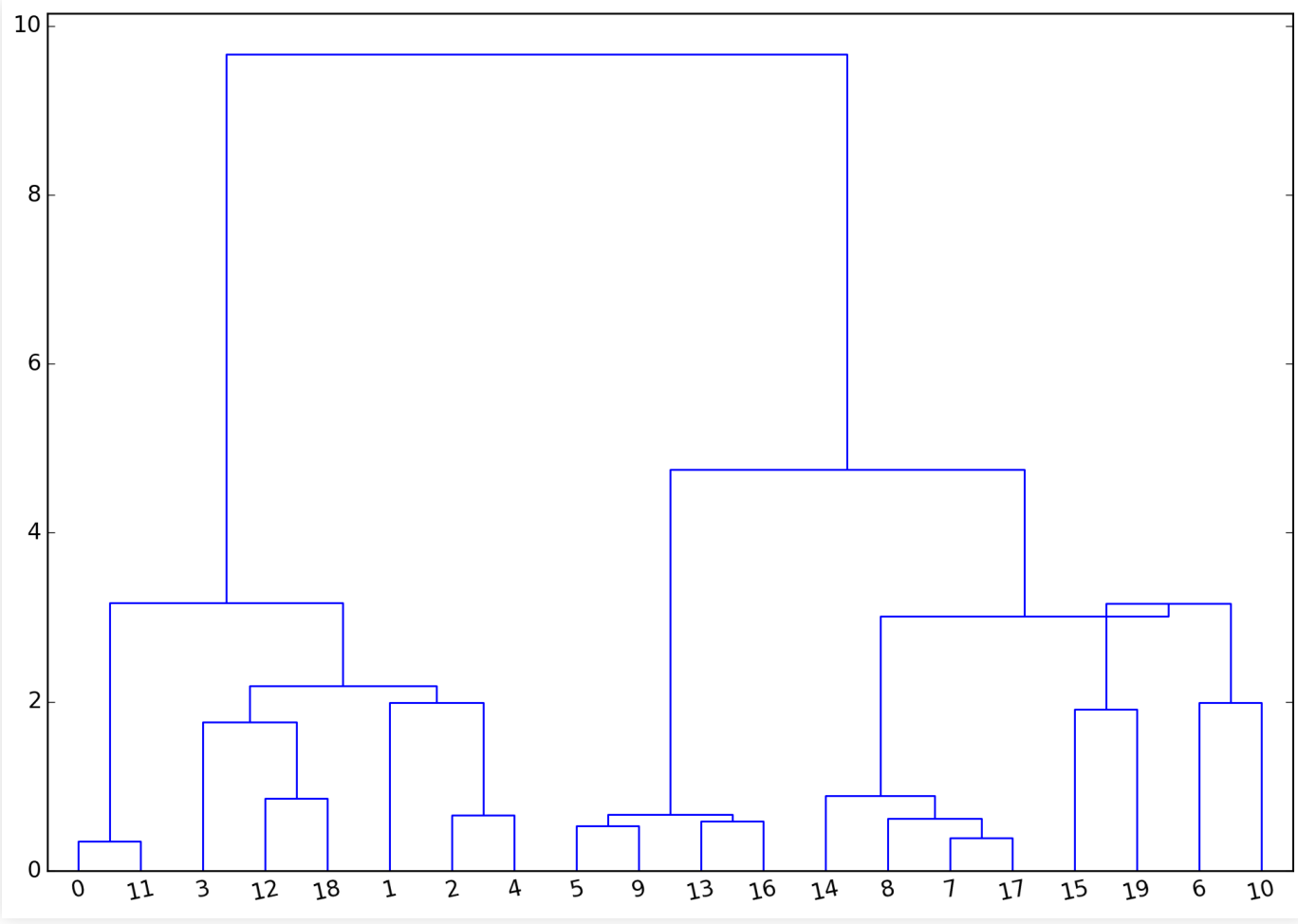
Agglomerative Clustering

- Join closest (linkage function), repeat



Agglomerative Clustering

- Result represented in a dendrogram

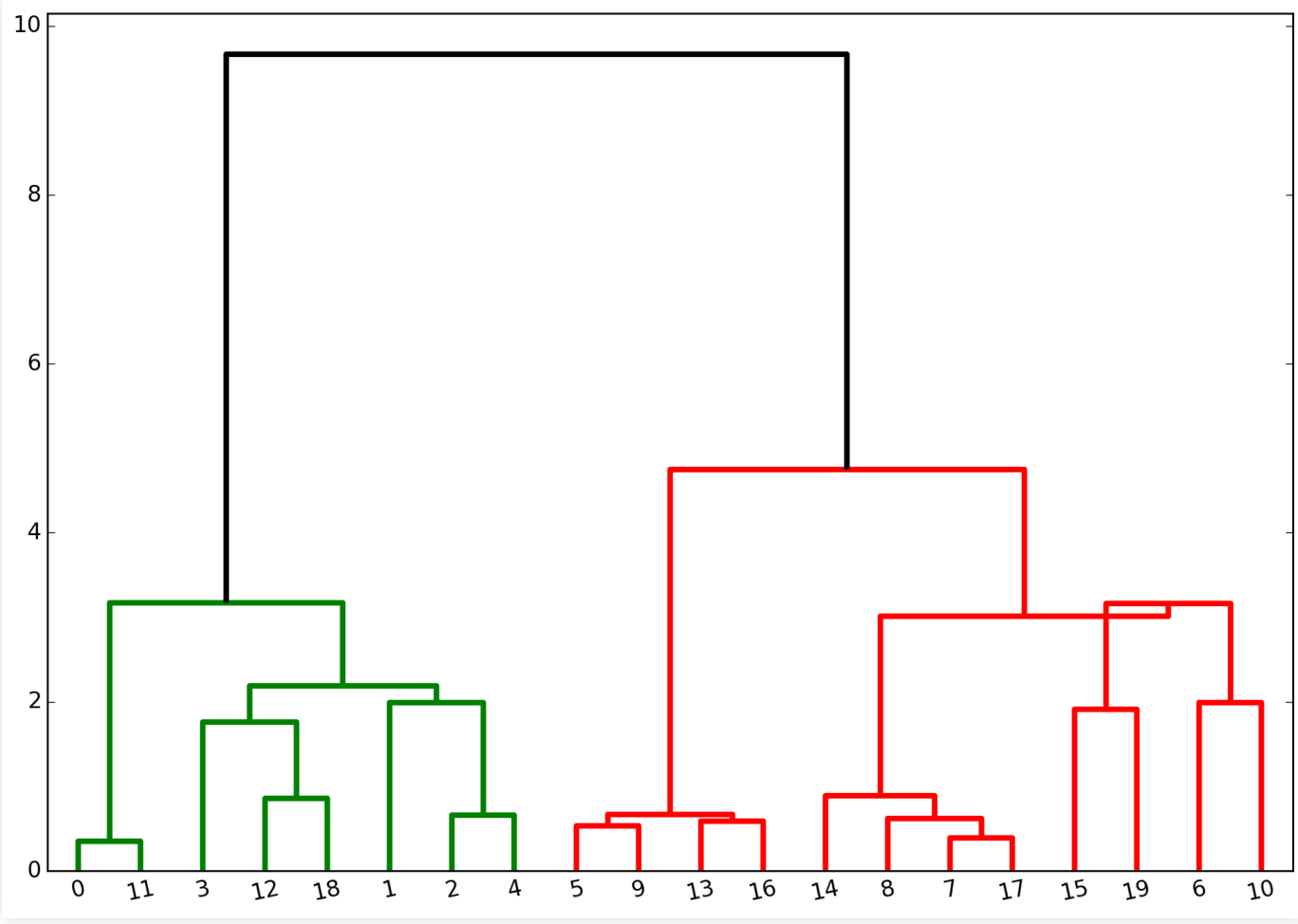


Agglomerative Clustering

- The result is a hierarchy of clusters
- But we may want a partitional clustering
- The solution is to select a level on the dendrogram

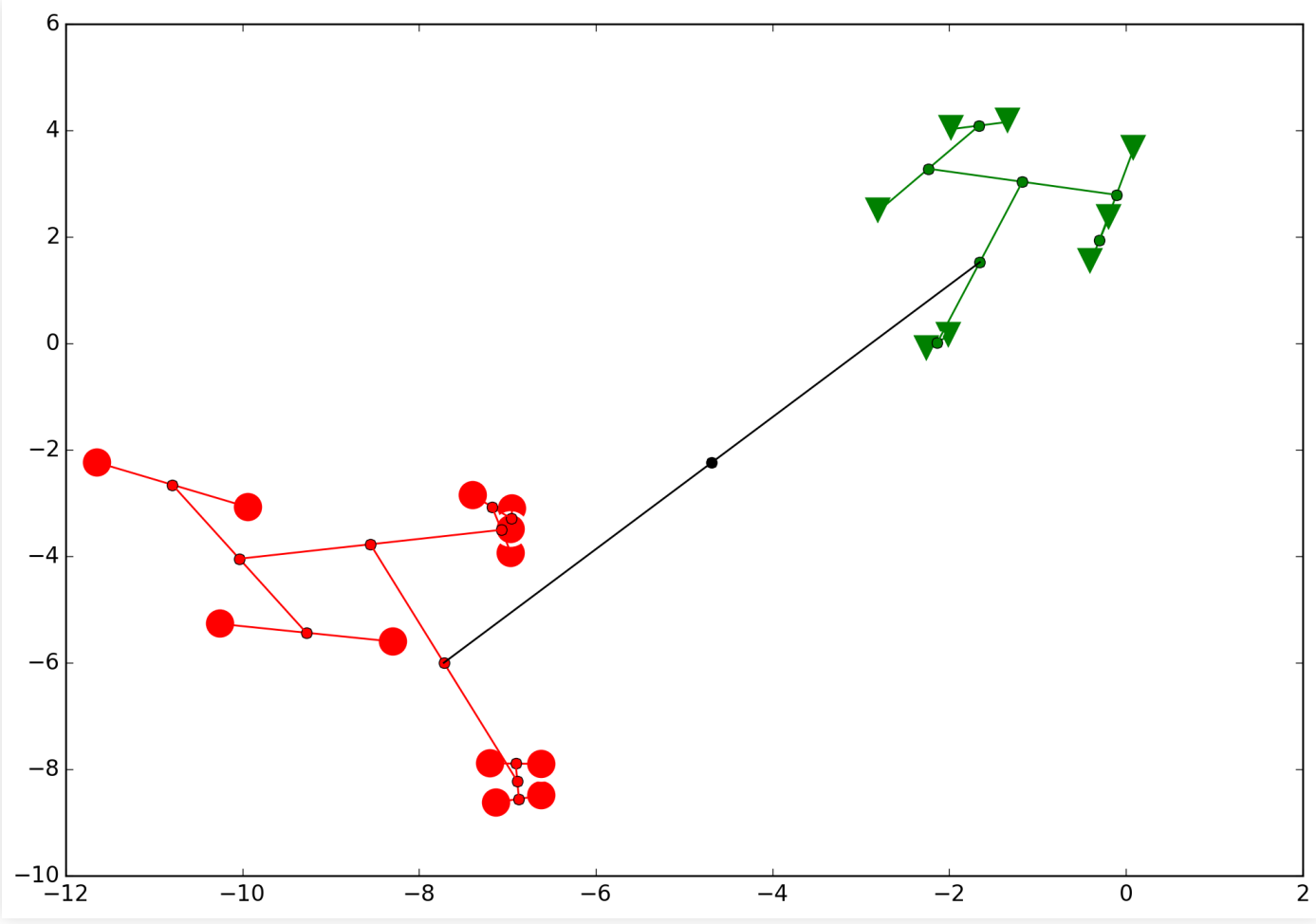
Agglomerative Clustering

- Two clusters



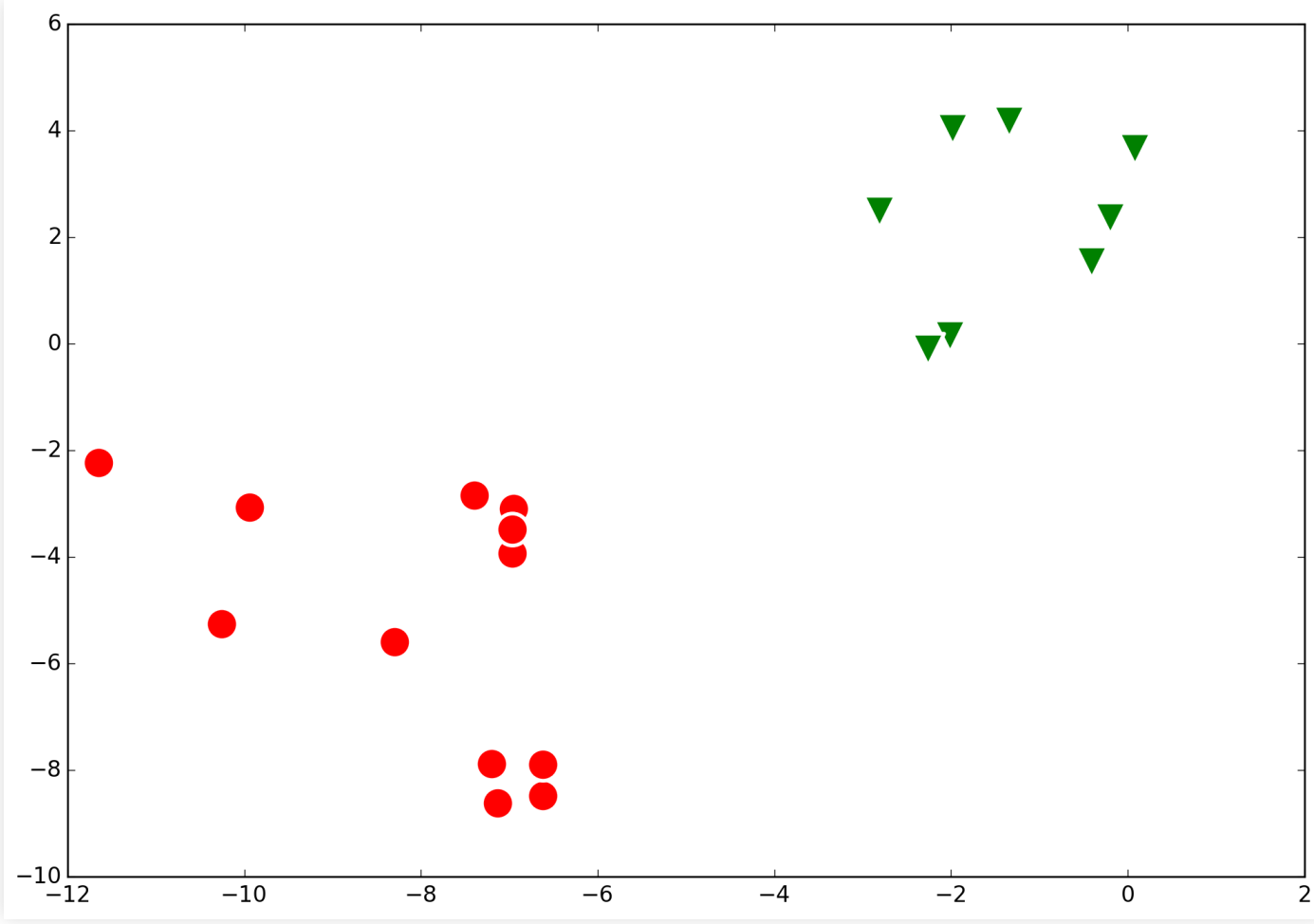
Agglomerative Clustering

- Two clusters



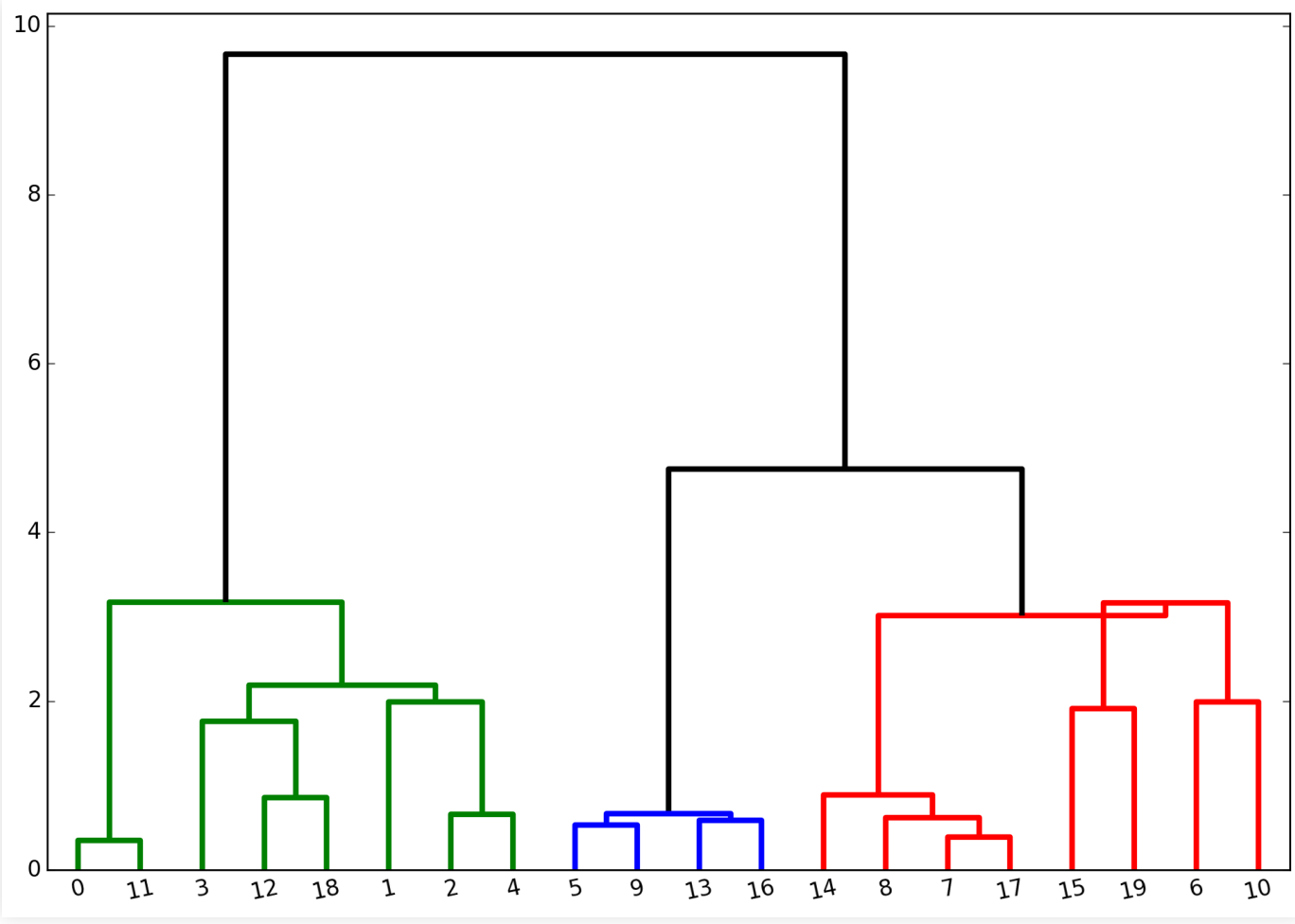
Agglomerative Clustering

- Two clusters



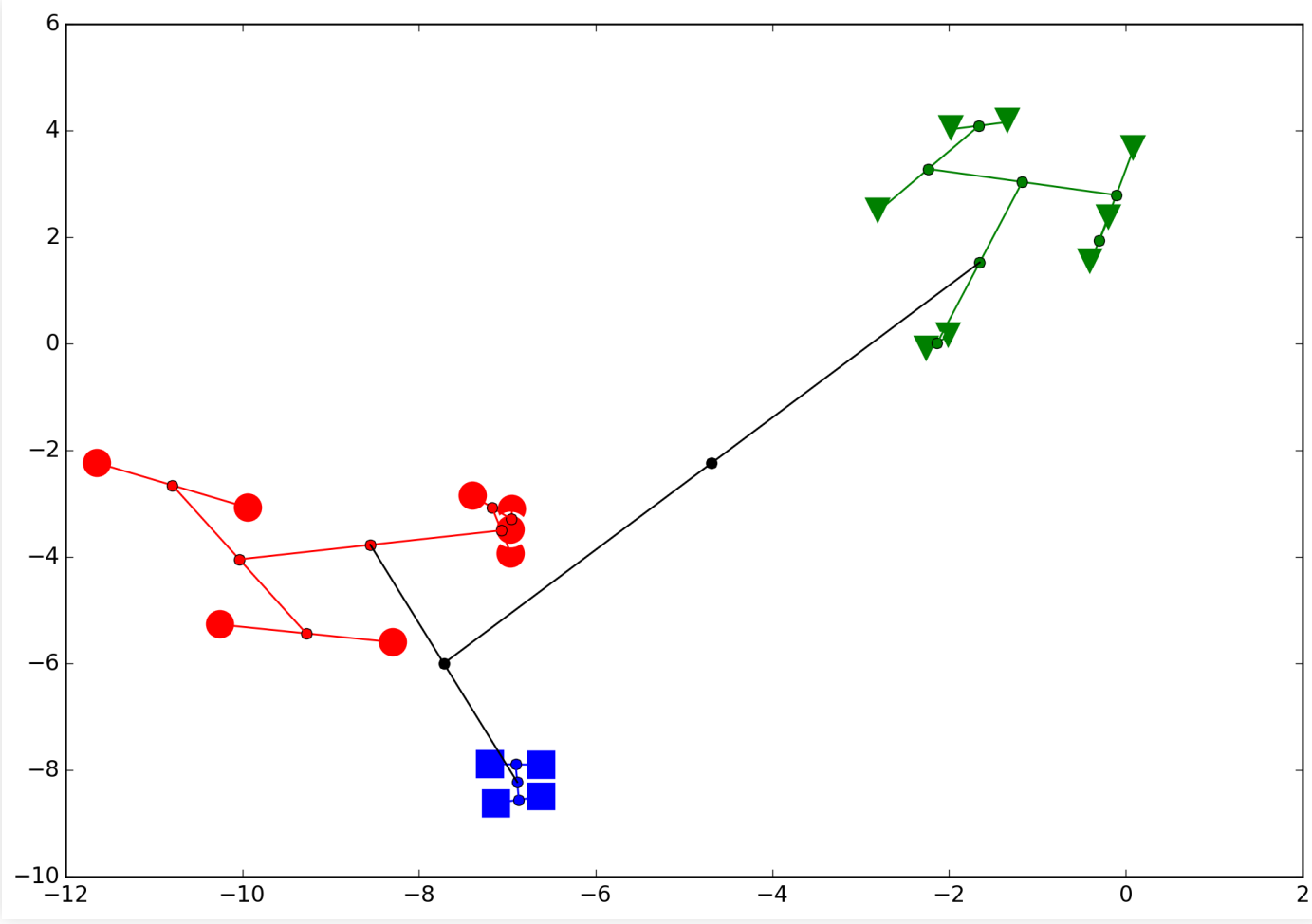
Agglomerative Clustering

- Three clusters



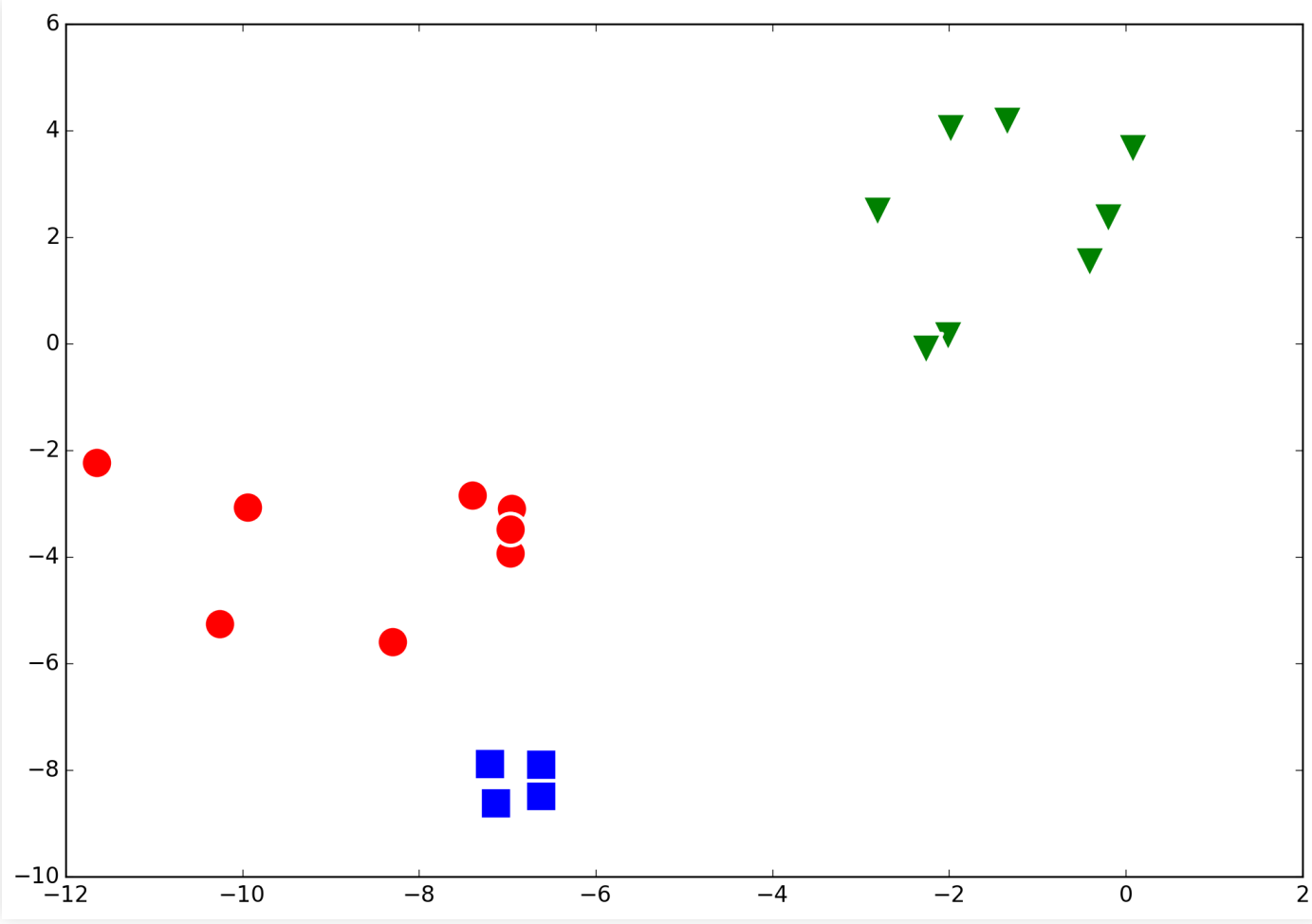
Agglomerative Clustering

- Three clusters



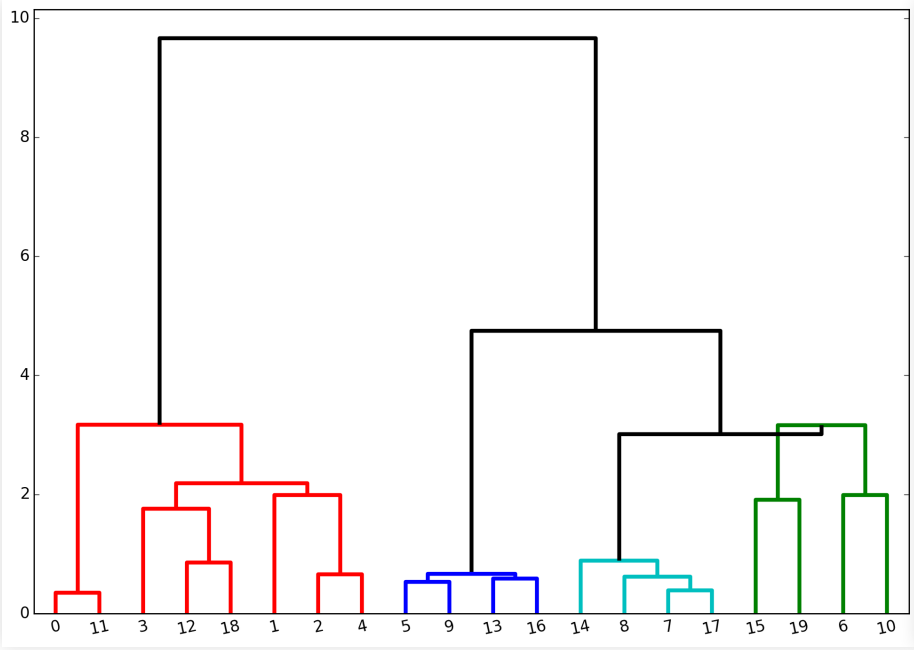
Agglomerative Clustering

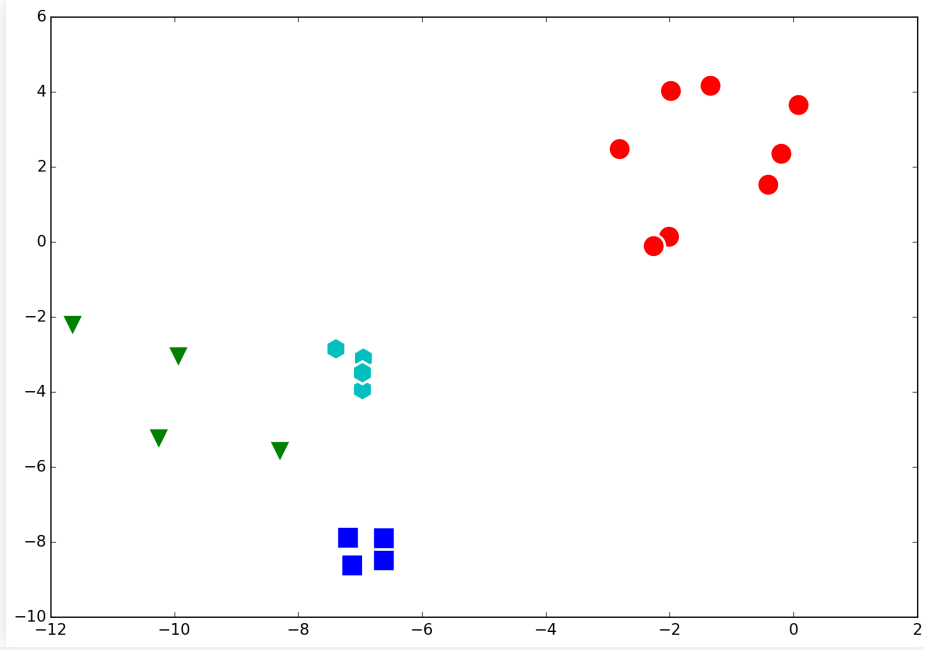
- Three clusters



Agglomerative Clustering

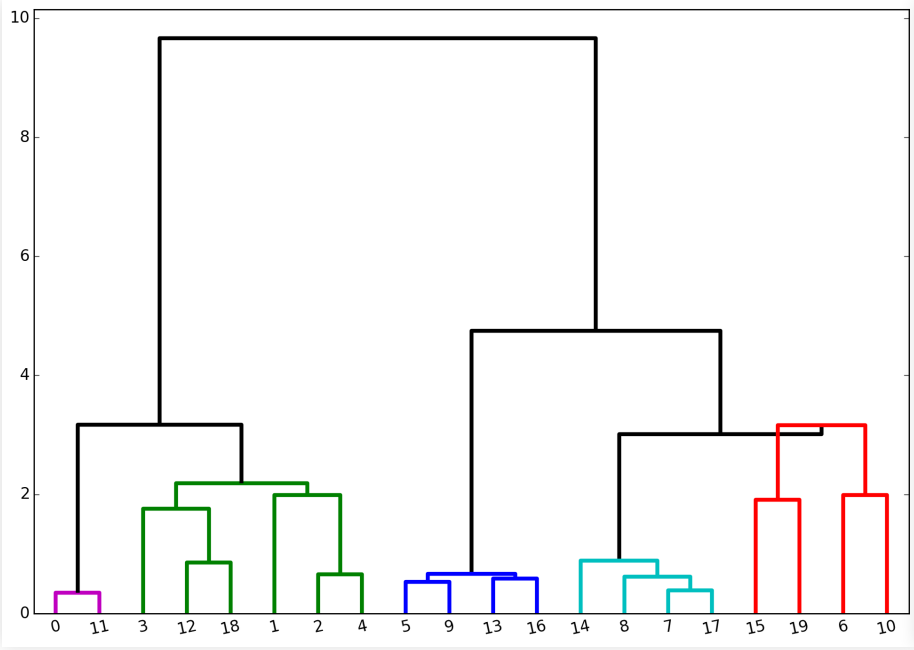
- Four clusters

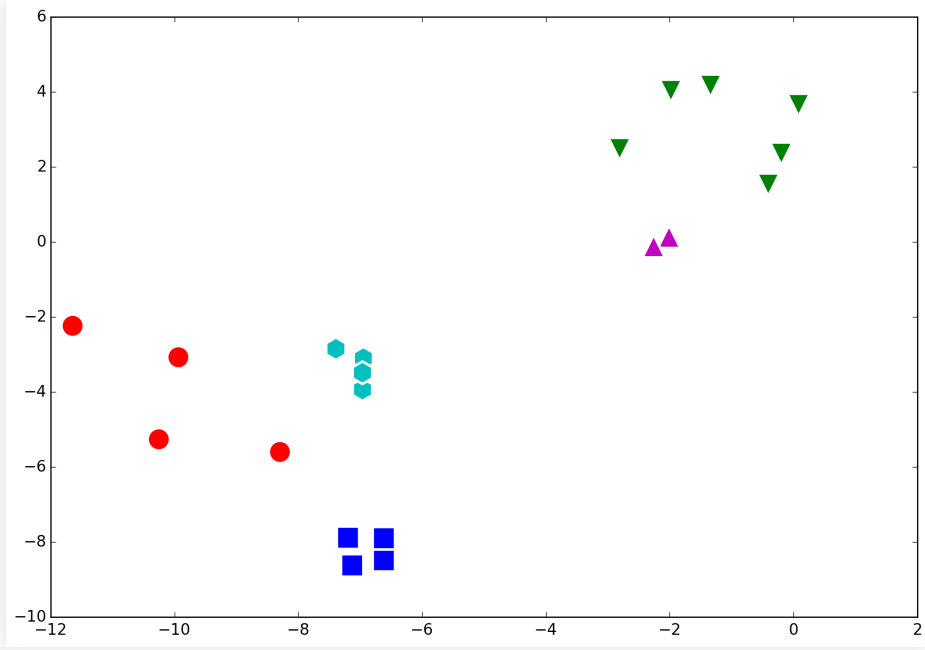




Agglomerative Clustering

- Five clusters

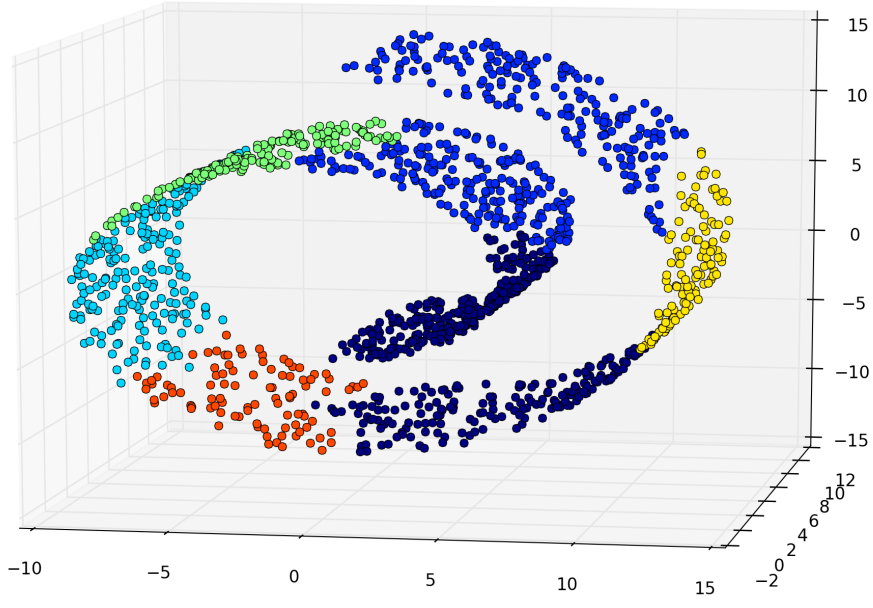




Agglomerative Clustering

- Connectivity constraints

- Agglomerative clustering is generally $O(n^3)$, not good for large datasets
- Also, we may not want clustering to aggregate solely by distance



Agglomerative Clustering

- We can prevent this by providing some structure via connectivity constraints

- Connectivity constraints define the graph of connections between examples
- Only clusters with connected examples can be joined
- Forces clustering to respect structure and can greatly speedup computation

- With Scikit-Learn, we can use the nearest neighbours graph:

- (returns a sparse matrix of $N \times N$ with 1 on connected)

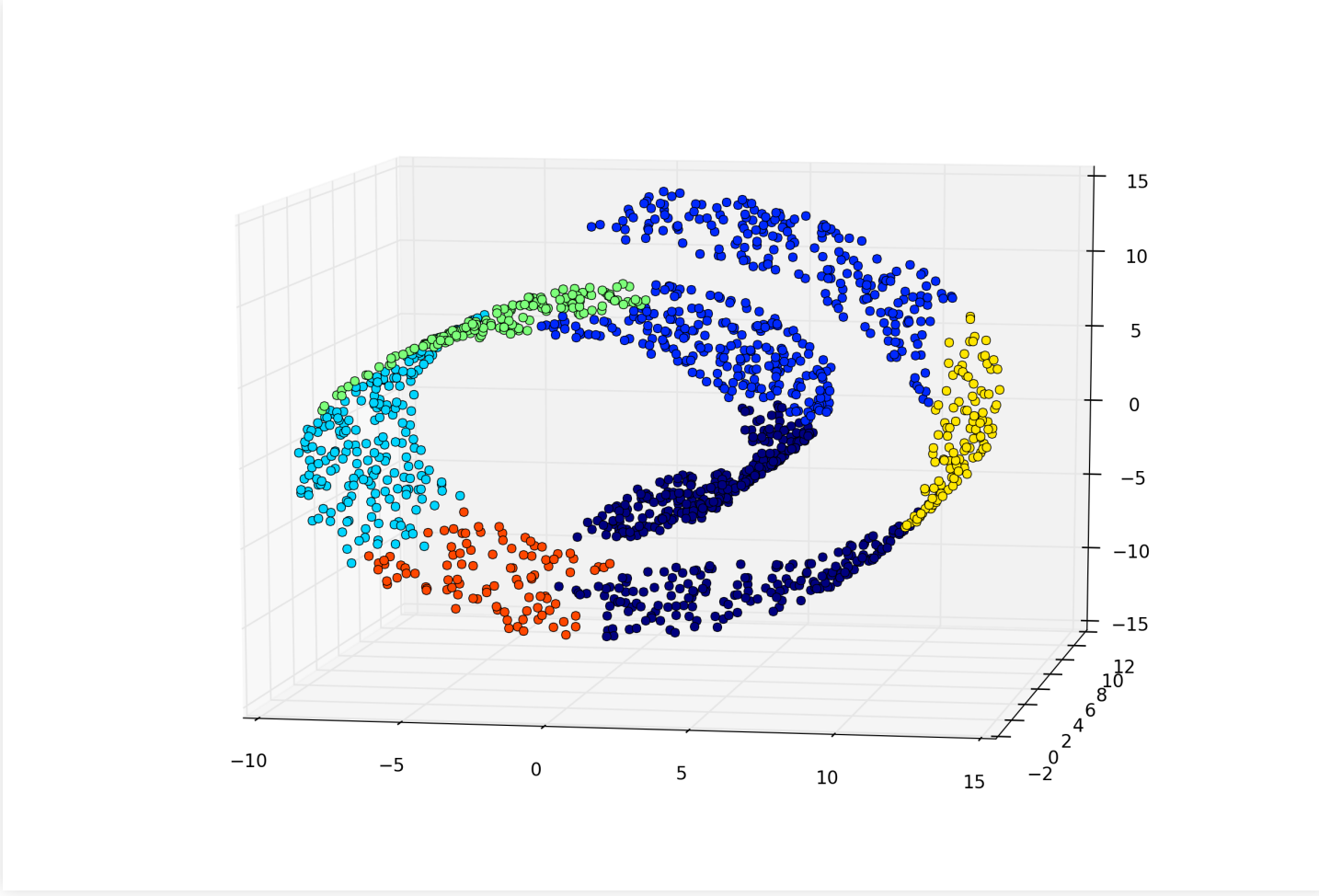
```
from sklearn.cluster import AgglomerativeClustering
from sklearn.neighbors import kneighbors_graph

connectivity = kneighbors_graph(X, n_neighbors=10, include_self=False)
ward = AgglomerativeClustering(n_clusters=6, connectivity=connectivity,
                               linkage='ward').fit(X)
```

- Based on this Scikit-Learn tutorial:

http://scikit-learn.org/stable/auto_examples/cluster/plot_ward_structured_vs_unstructured.html

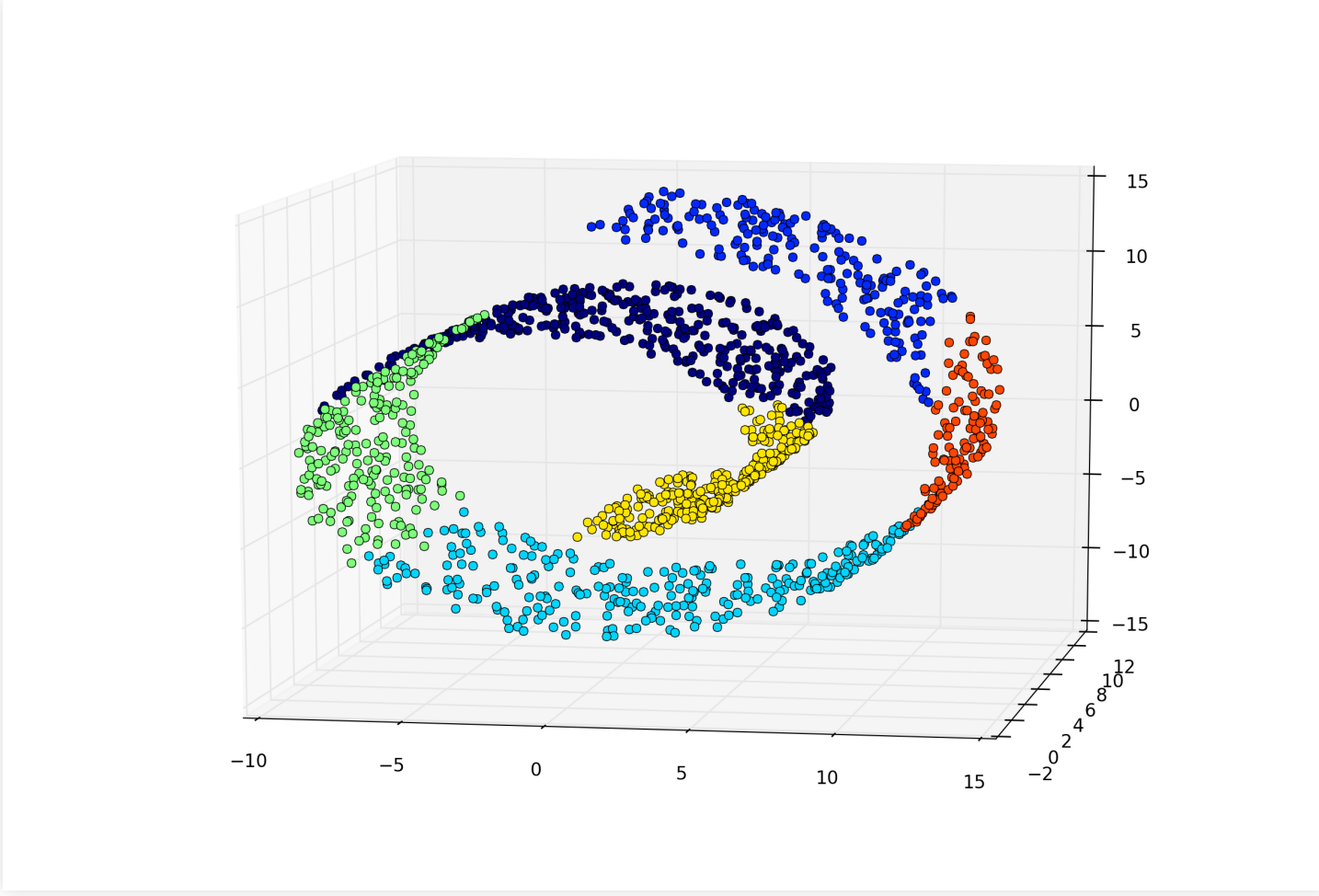
Agglomerative Clustering

- Without constraints clusters reach out over space:



Agglomerative Clustering

- Constraints speed up and guide clustering



Agglomerative Clustering

- Using AC with Scikit-Learn:

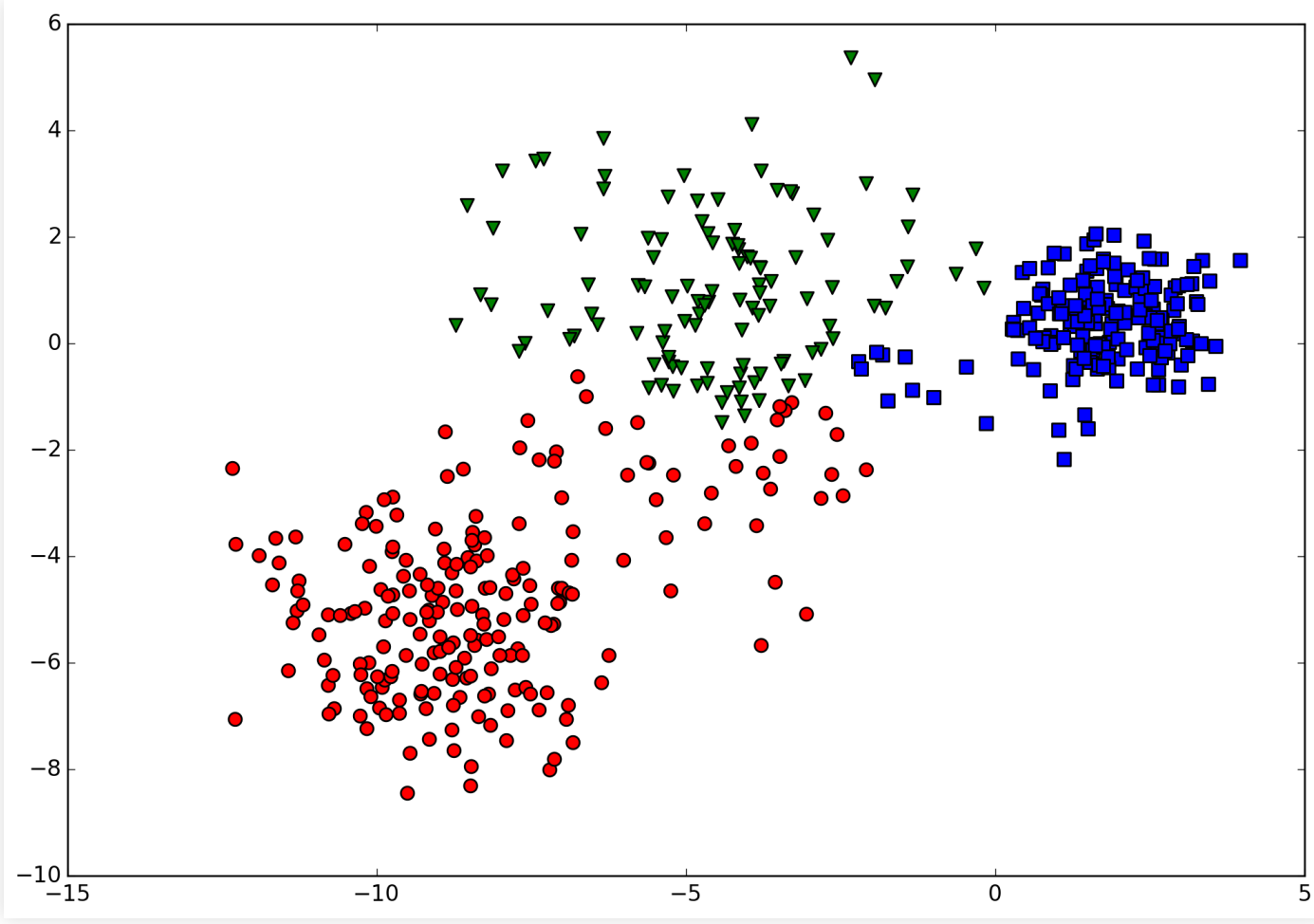
```
class sklearn.cluster.AgglomerativeClustering:
#arguments
n_clusters=2,          #number of clusters
affinity='euclidean', #distance between examples
connectivity=None,     #connectivity constraints
linkage='ward'         #'ward', 'complete', 'average'

#attributes
labels_                # array [n_samples]
children_              # array, shape (n_nodes-1, 2)
```

- Three linkage options available in Scikit-Learn

- Complete linkage $dist(C_j, C_k) = max\left(dist(x \in C_j, y \in C_k)\right)$
- Average linkage $dist(C_j, C_k) = mean\left(dist(x \in C_j, y \in C_k)\right)$
- Ward linkage : minimize SSE, $\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|x_n - \mu_k\|^2$

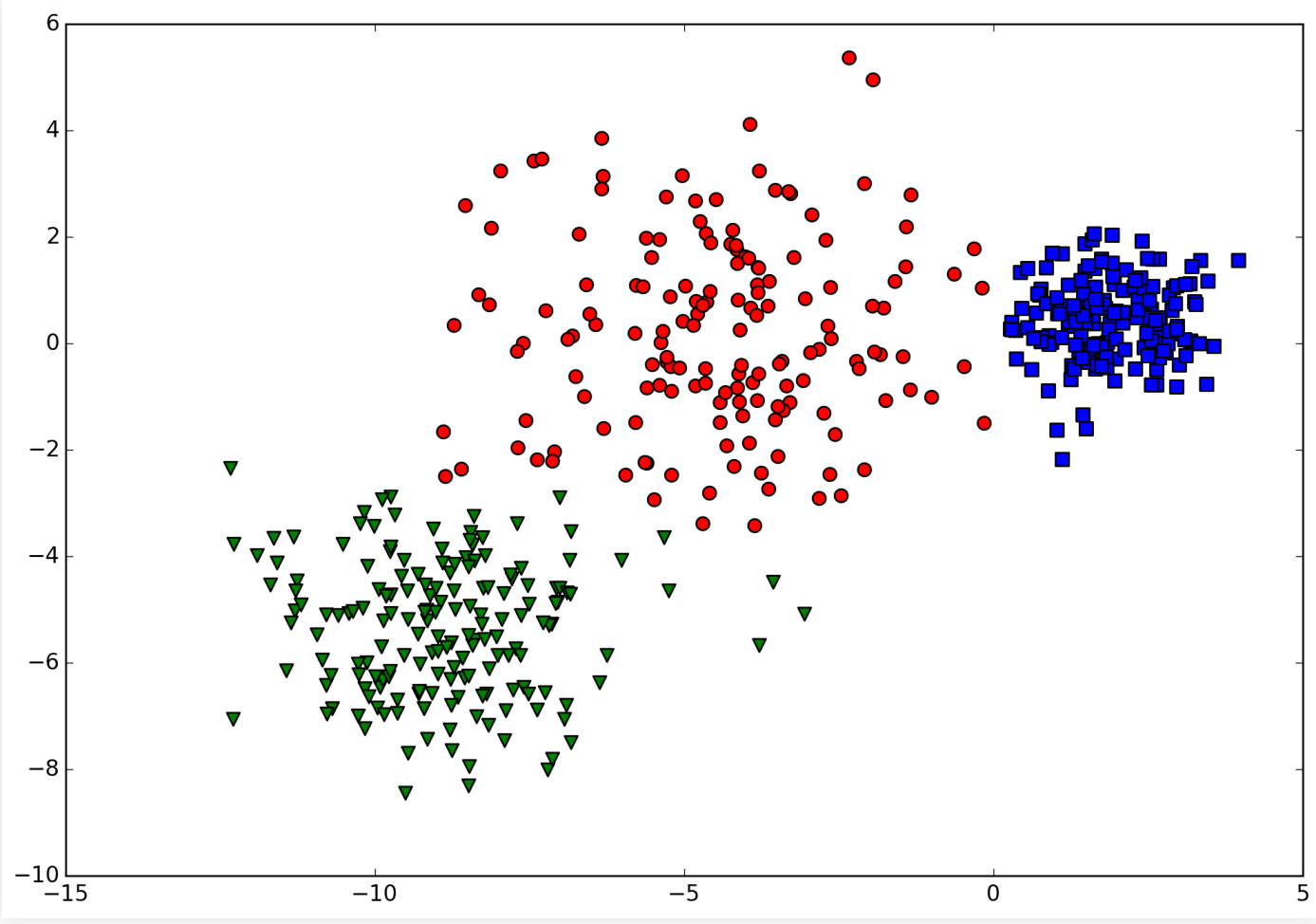
A.C. and Linkage

- Complete linkage tends to favour larger clusters



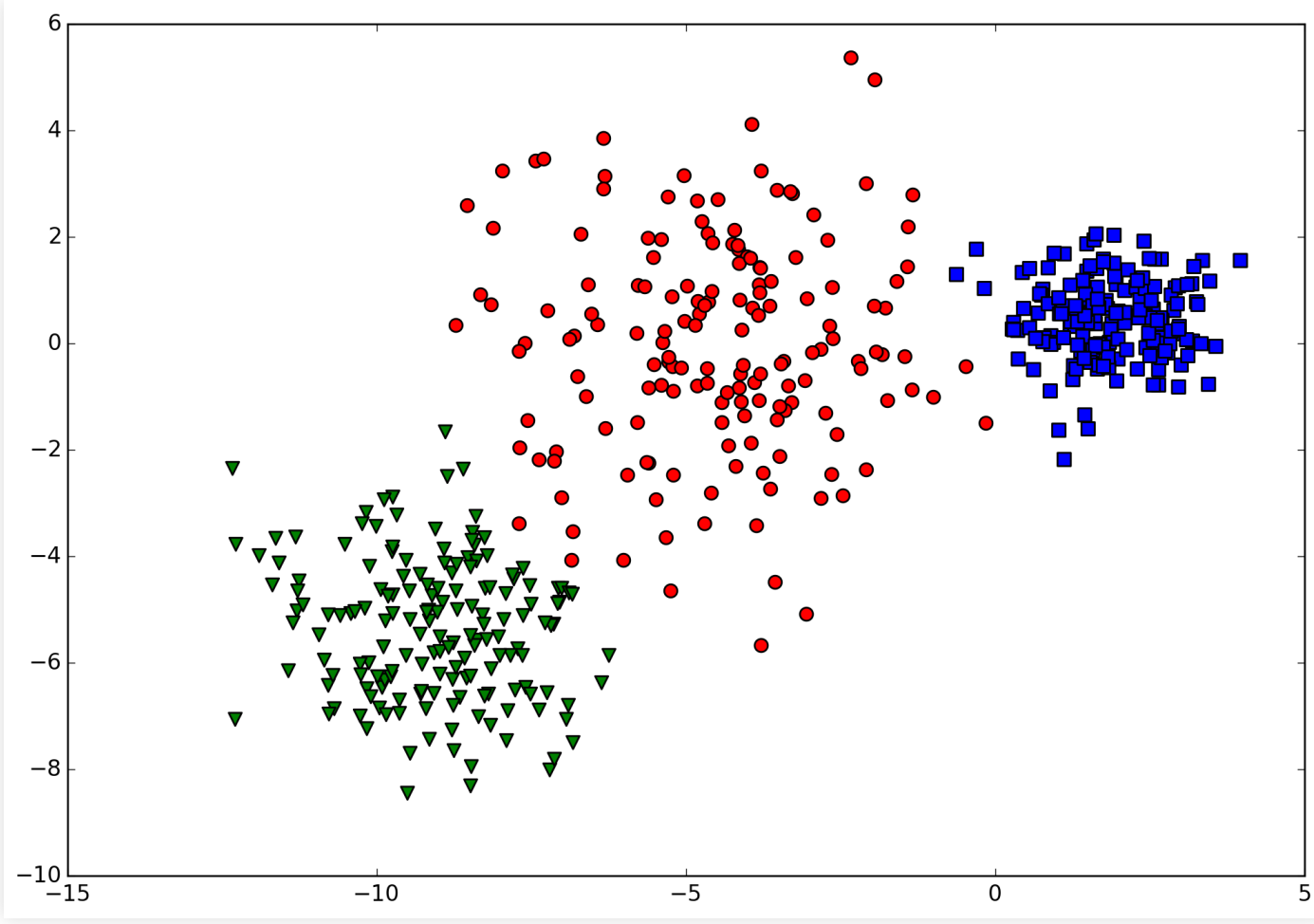
A.C. and Linkage

- Average linkage solves that partially



A.C. and Linkage

- Ward linkage is generally best but Euclidean only



Divisive Clustering

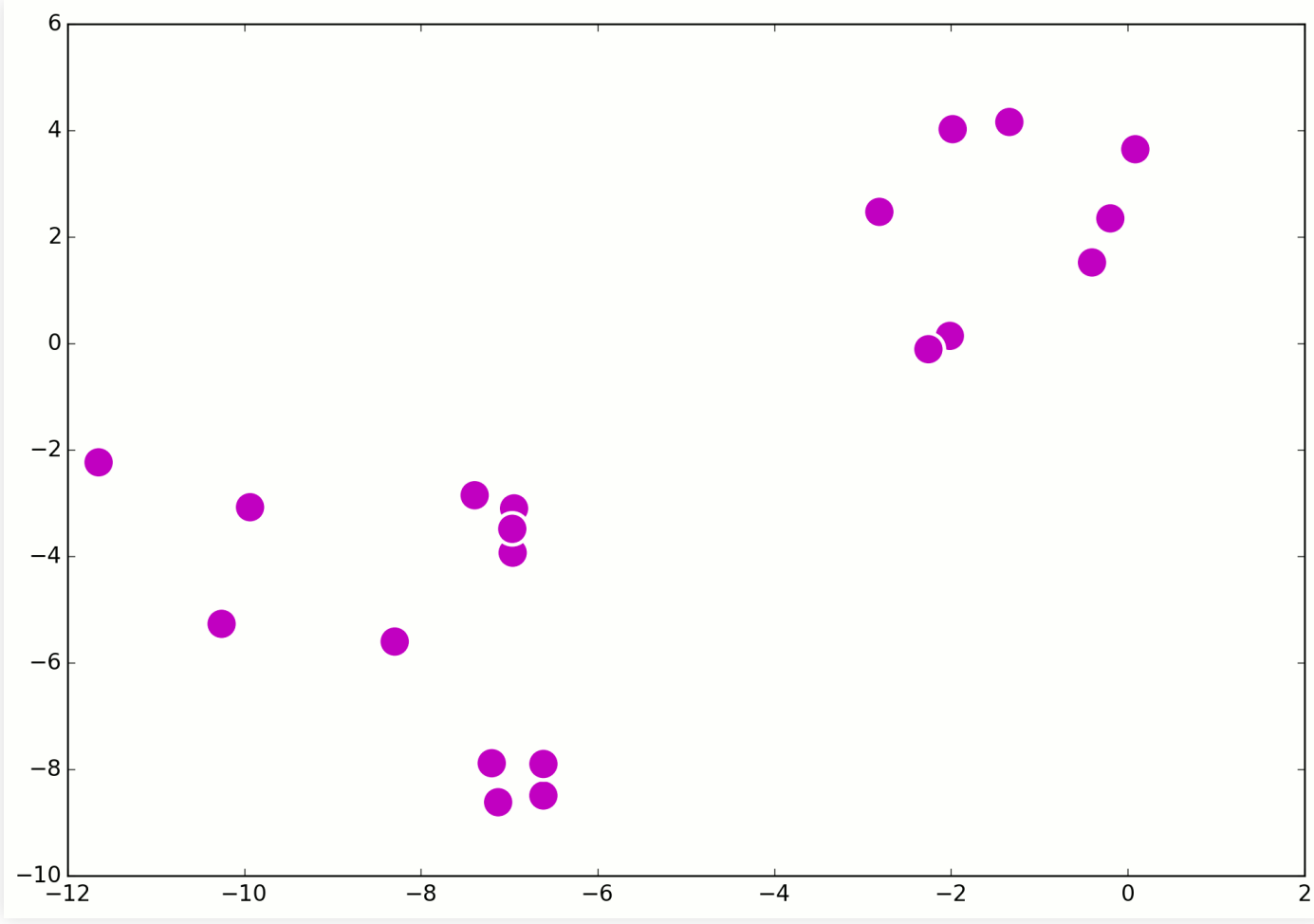
Divisive Clustering

Bisecting k-Means algorithm:

- Start with a single cluster for all examples
- Select one cluster (largest, lowest score, ...)
- Split cluster with k-means ($k = 2$)
- Repeat until desired number of clusters

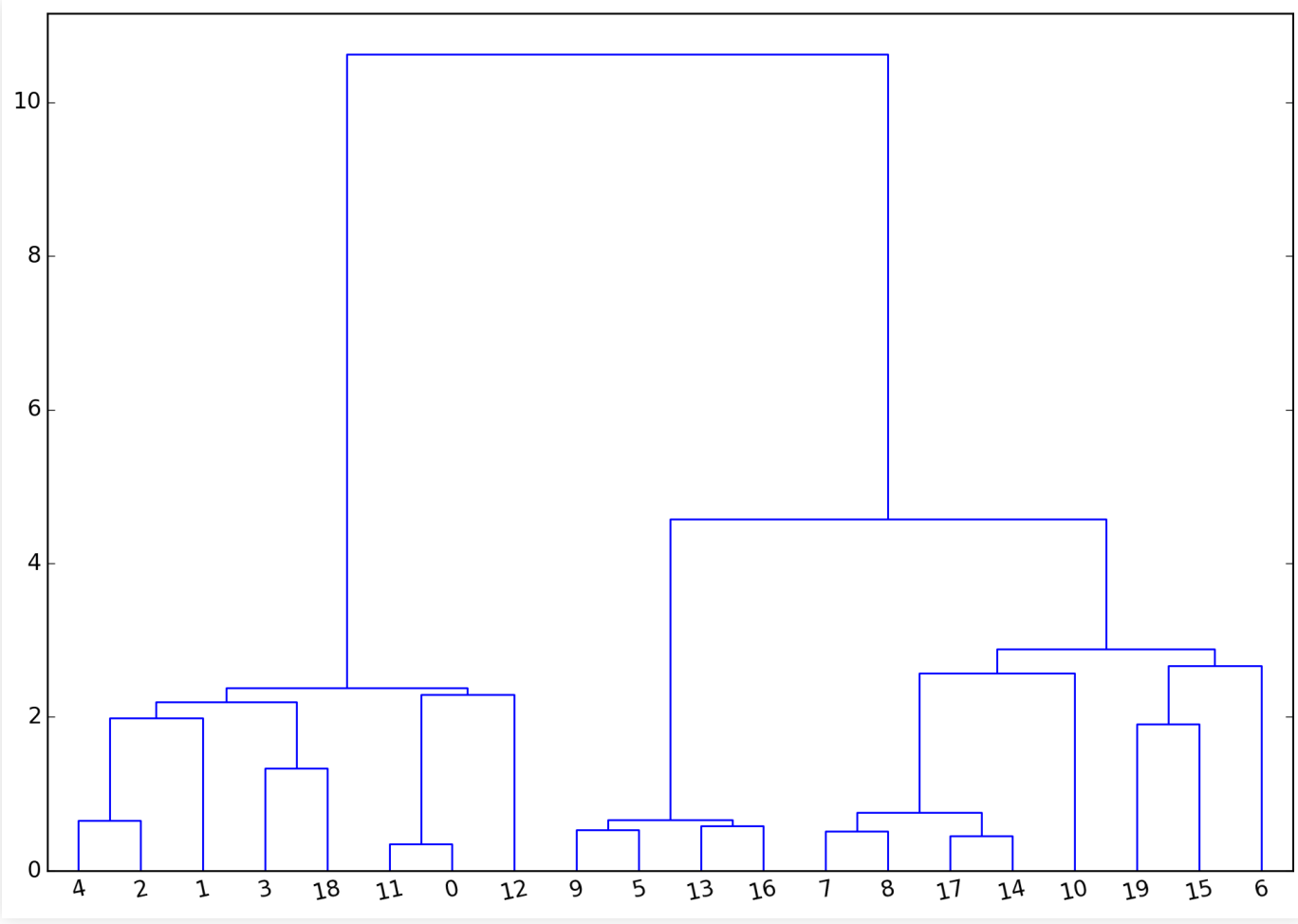
Divisive Clustering

- Splitting on largest cluster



Divisive Clustering

- Resulting hierarchy:



Divisive Clustering

- Exhaustive search is $O(2^n)$
- Top-down clustering requires clustering at each step to split (e.g. k-means)
- However, it may be a good option if we want few large clusters and the auxiliary clustering algorithm is fast

Clustering Features

Clustering Features

- Clustering can also be used for dimensionality reduction
- Clustering features allows us to agglomerate different features, average them and extract new features
- E.g. a matrix of examples and features:

	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	. . .
Example 1	0,05	0,09	0,80	0,18	0,76	0,23	. . .
Example 2	0,97	0,79	0,90	0,26	0,94	0,66	. . .
Example 3	0,93	0,43	0,23	0,27	0,80	0,64	. . .
Example 4	0,89	0,45	0,58	0,95	0,22	0,92	. . .
Example 5	0,68	0,42	0,60	0,46	0,29	0,55	. . .
Example 6	0,69	0,75	0,60	0,42	0,82	0,08	. . .
Example 7	0,32	0,74	0,56	0,86	0,86	0,65	. . .
Example 8	0,31	0,28	0,53	0,05	0,60	0,00	. . .
Example 9	0,04	0,39	0,52	0,21	0,60	0,57	. . .
Example 10	0,51	0,93	0,30	0,80	0,61	0,77	. . .
Example 11	0,35	0,35	0,26	0,74	0,40	0,70	. . .
. . .	. . .	. . .	. . .	. . .	. . .	. . .	

Clustering Features

- Transposing, features as examples

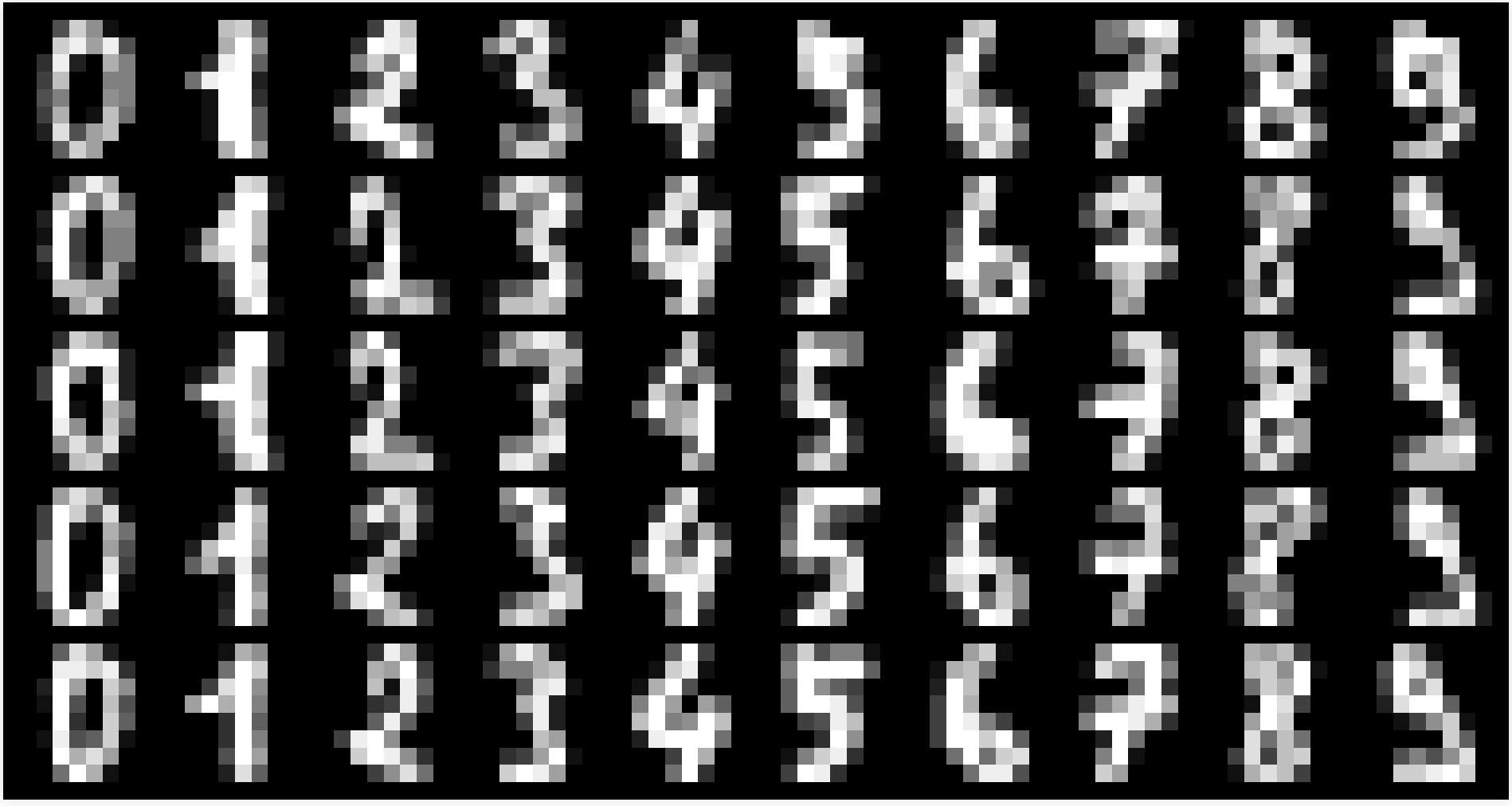
	Example 1	Example 2	Example 3	Example 4	Example 5	Example 6	. . .
Feature 1	0,05	0,97	0,93	0,89	0,68	0,69	. . .
Feature 2	0,09	0,79	0,43	0,45	0,42	0,75	. . .
Feature 3	0,80	0,90	0,23	0,58	0,60	0,60	. . .
Feature 4	0,18	0,26	0,27	0,95	0,46	0,42	. . .
Feature 5	0,76	0,94	0,80	0,22	0,29	0,82	. . .
Feature 6	0,23	0,66	0,64	0,92	0,55	0,08	. . .
Feature 7	0,51	0,38	0,78	0,64	0,88	0,40	. . .
Feature 8	0,18	0,93	0,16	0,47	0,62	0,61	. . .
Feature 9	0,07	0,56	0,88	0,49	0,91	0,06	. . .
Feature 10	0,78	0,09	0,81	0,18	0,61	0,82	. . .
Feature 11	0,86	0,91	0,58	0,02	0,79	0,78	. . .
. . .	. . .	. . .	. . .	. . .	. . .	. . .	

Clustering Features

- Clustering will group similar features together
- Then agglomerate into smaller set of features

Clustering Features

- Example: handwritten digits data set:



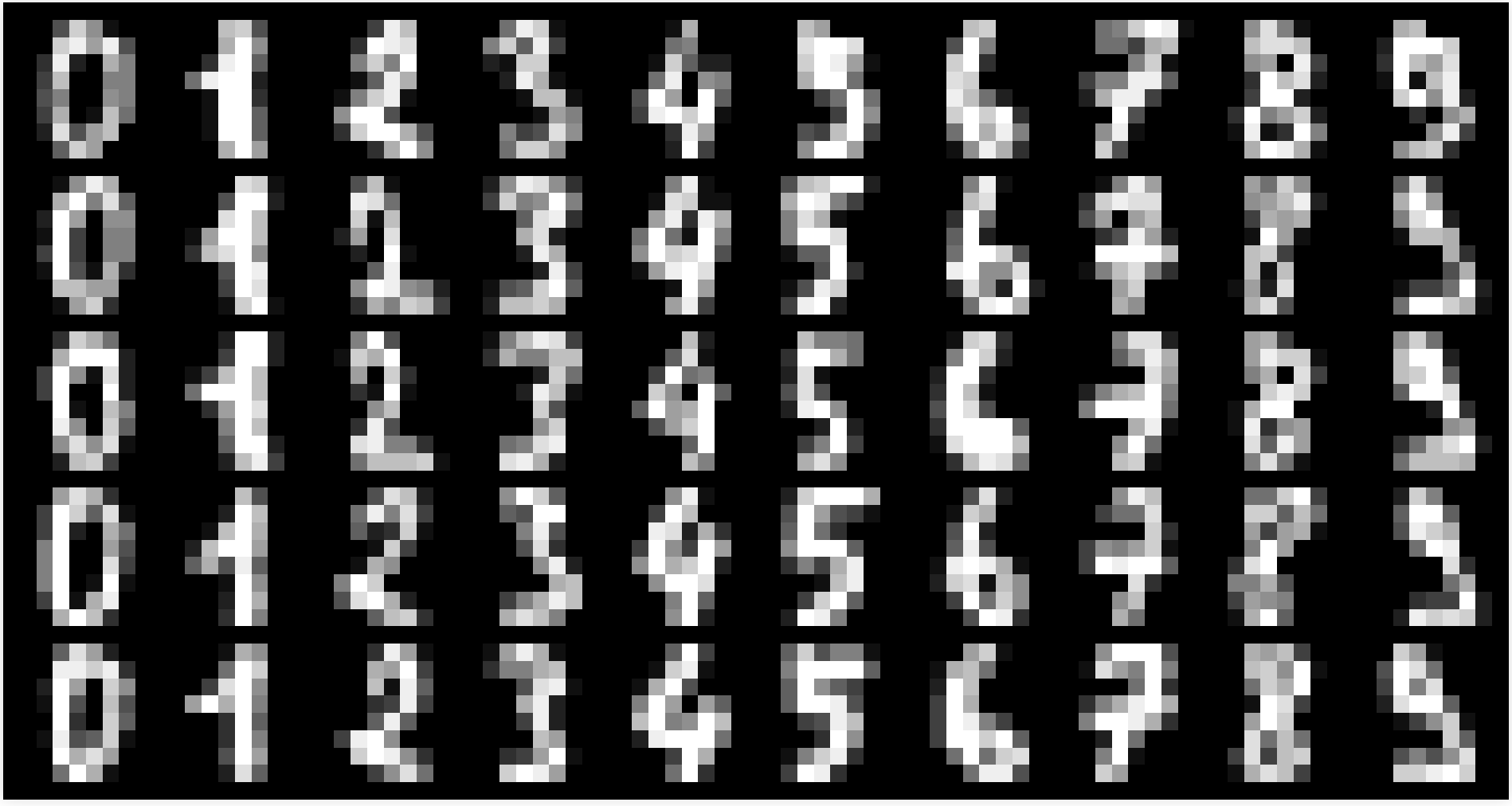
Clustering Features

- Example: handwritten digits data set
- Each digit is represented with $8 \times 8 = 64$ features
- To reduce, we convert the 1797 examples of 64 features into 64 examples of 1797 features
- Then we cluster the 64 into 16 clusters of similar features

- But restrict linkage to adjacent pixels; so similar in same region of image

Clustering Features

- Original data:



Clustering Features

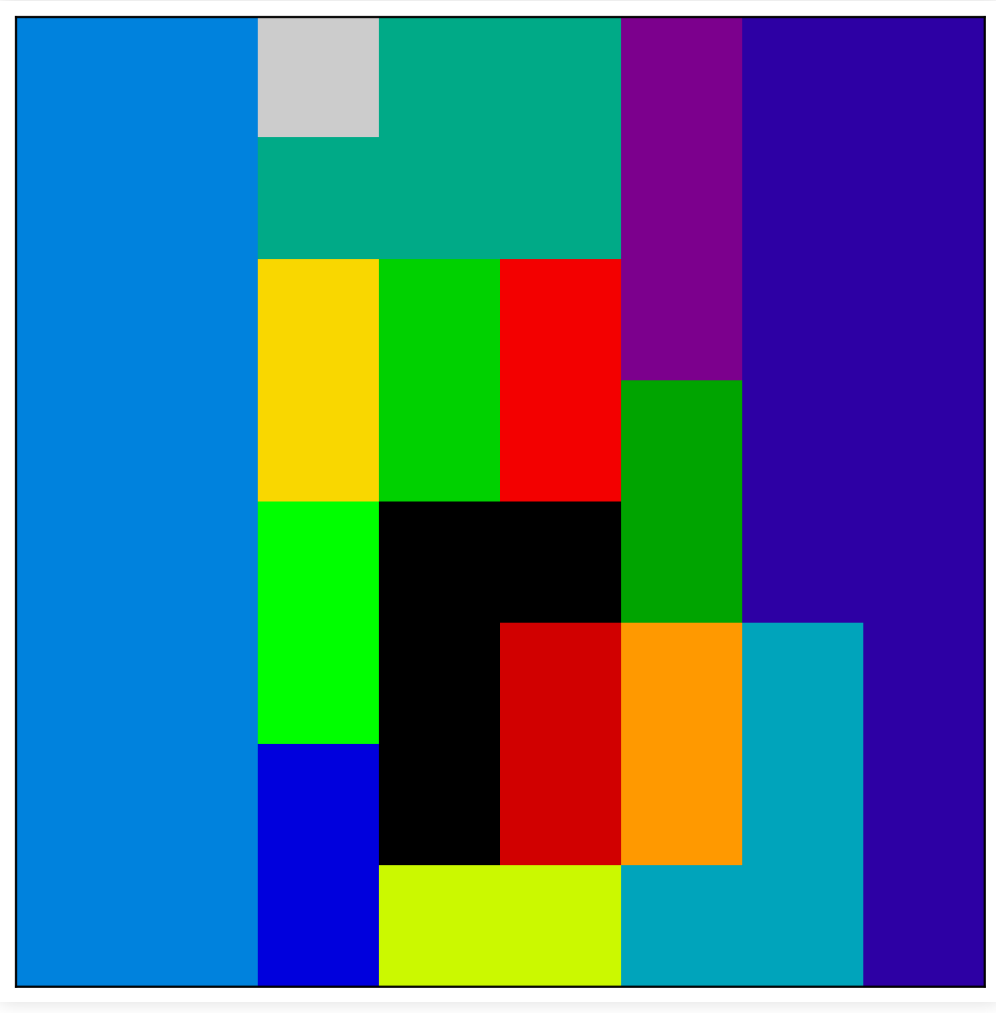
- With Scikit-Learn

- (Based on Feature Agglomeration example at Scikit Learn)

```
import numpy as np
from sklearn import datasets, cluster
from sklearn.feature_extraction.image import grid_to_graph

digits = datasets.load_digits()
images = digits.images
X = np.reshape(images, (len(images), -1))
connectivity = grid_to_graph(images[0].shape[0],images[0].shape[1])
agglo = cluster.FeatureAgglomeration(connectivity=connectivity,
                                     n_clusters=16)
agglo.fit(X)
X_reduced = agglo.transform(X)
X_restored = agglo.inverse_transform(X_reduced)
```

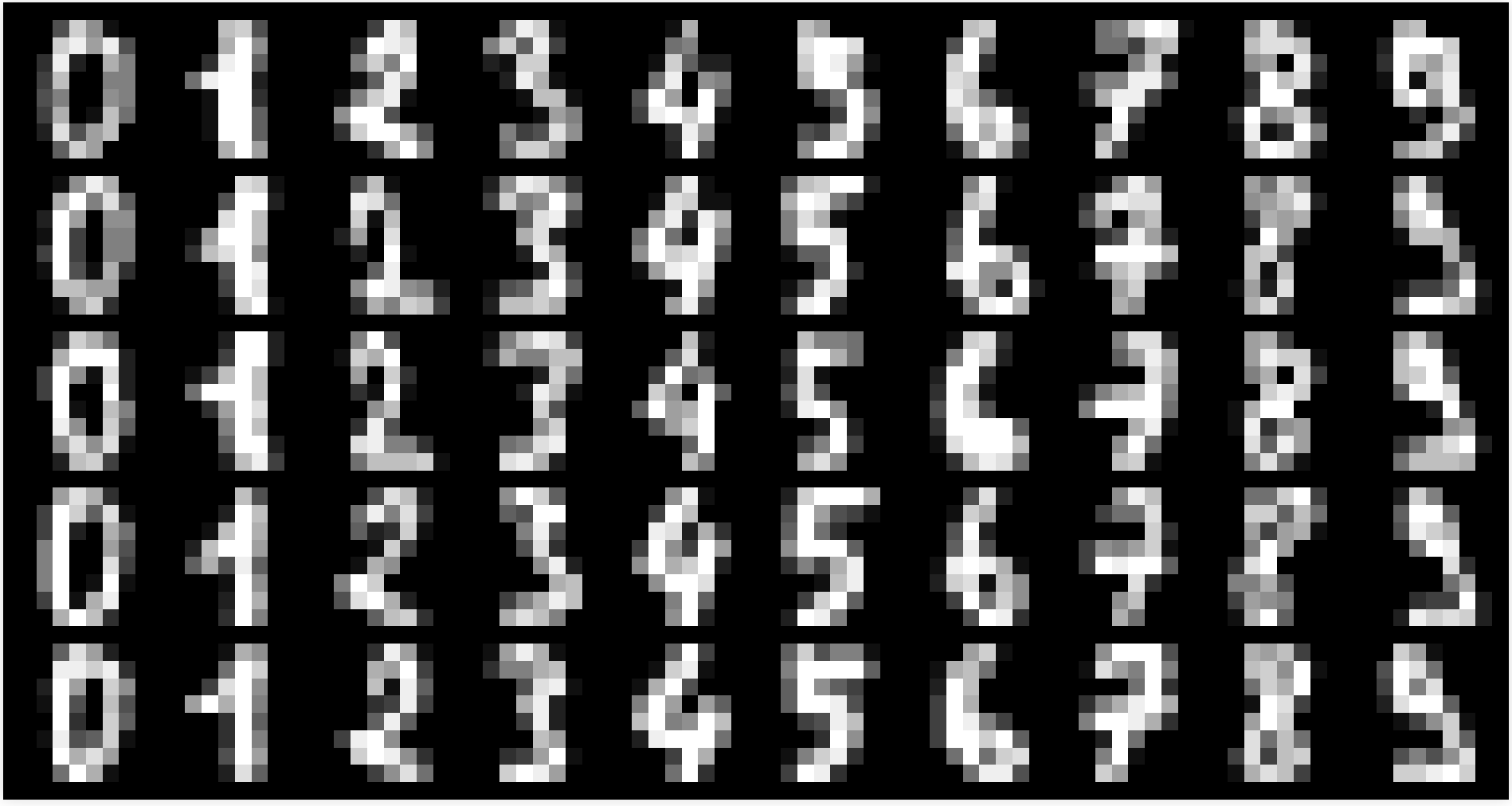
Clustering Features

- Feature clusters, linkage to adjacent pixels



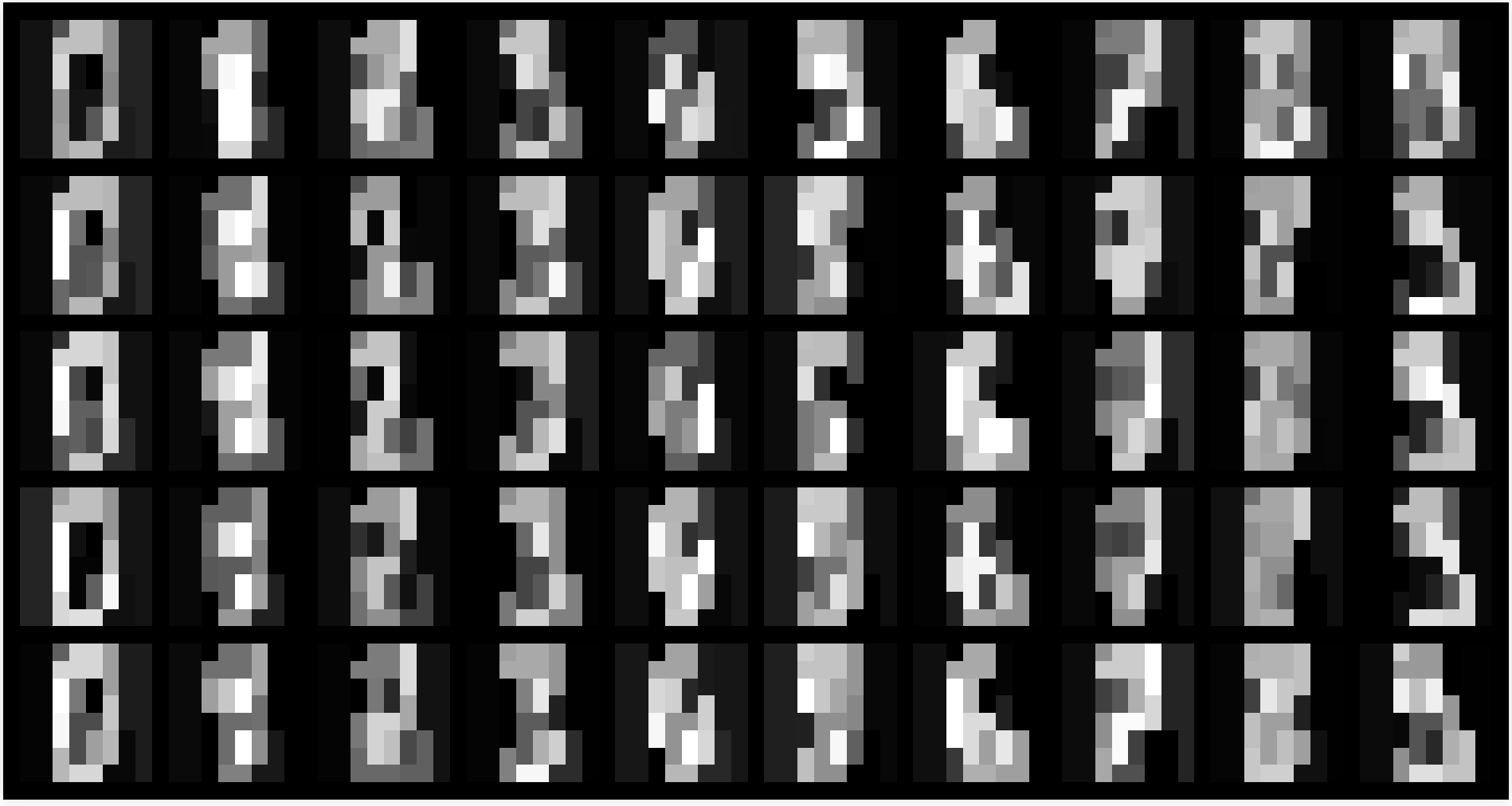
Clustering Features

- Original data:



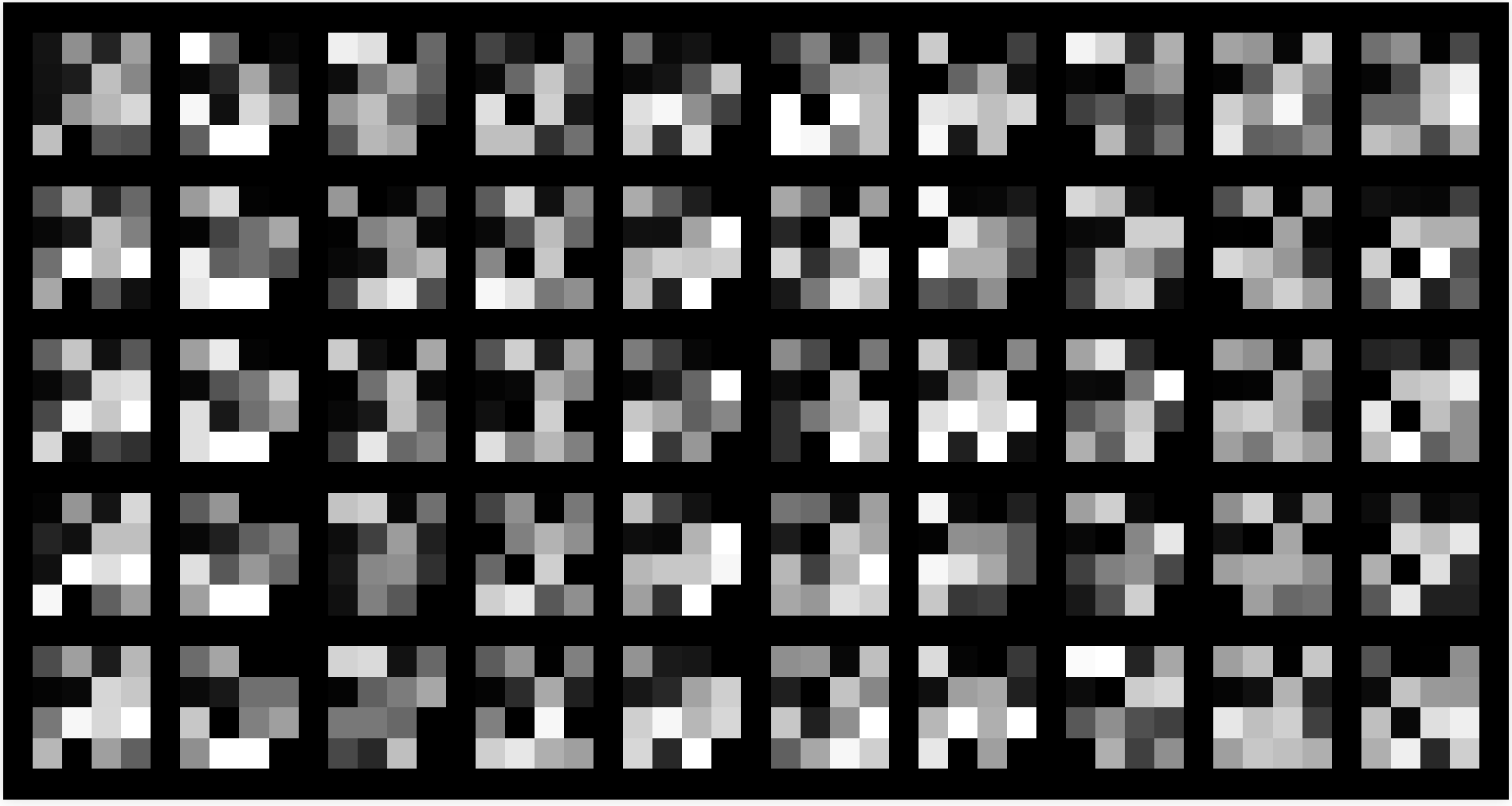
Clustering Features

- Restored data (same size, repeated averages):



Clustering Features

- Reduced data ($4 \times 4 = 16$ features):



Summary

Hierarchical Clustering

Summary

- Nested clusters
- Measures: examples and clusters (linkage)
- Bottom-up: Agglomerative Clustering
- Top-down: divisive (bisecting k-means)
- Effects of different linkage options
- Feature agglomeration with hierarchical clustering

Further reading

- Alpaydin, 7.7
- Optional: Scikit-learn documentation on clustering:

• http://scikit-learn.org/stable/modules/clustering.html