CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

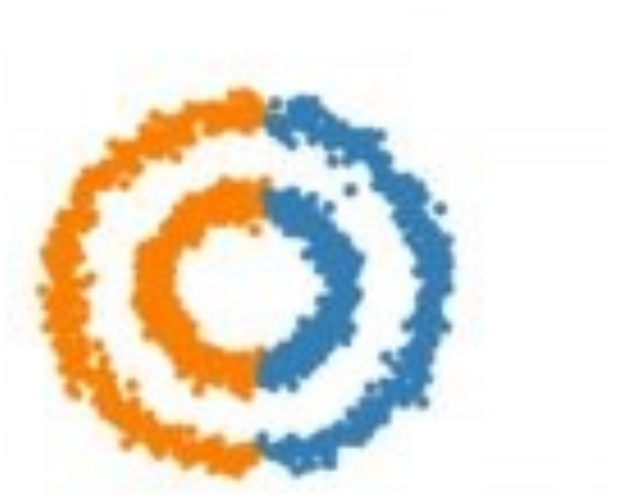
MODULE 3: CLUSTERING 2

DAT405 / DIT407, 2022-2023, READING PERIOD 4

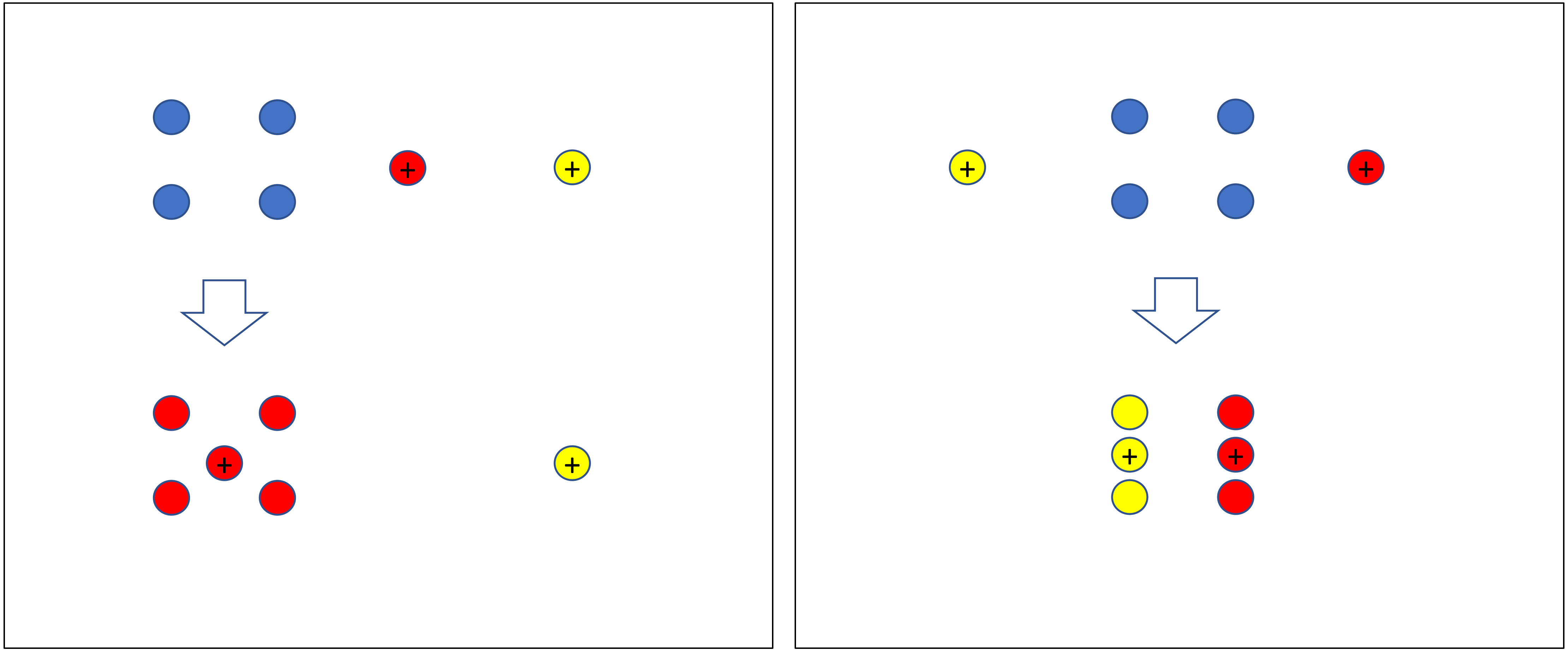
Topics

- DBSCAN clustering
- Hierarchical clustering
- Validating clusterings

Limitations of K-means clustering



K-means: result depends on initialization



DBSCAN clustering

Steven Bierwagen

DBSCAN

- Density-Based Spatial Clustering of Applications with Noise

- From 1996

Ingredients for DBSCAN

- A *distance* measure (or metric or similarity measure)
  - often Euclidean distance

scanning radius

- A number defining the meaning of *neighbor*
  - epsilon: the max distance between two points considered neighbors.

- A number defining the meaning of *cluster* (vs outlier or noise)
  - minpts: the minimum number of points in a cluster.

min points inside radius

Two hyperparameters

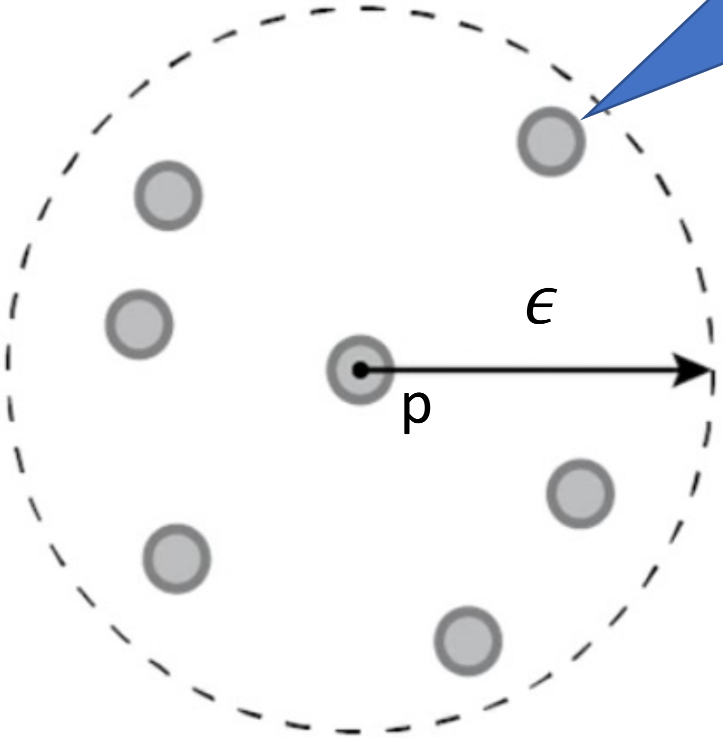
Labeling step

All points in dataset labeled as one of these:

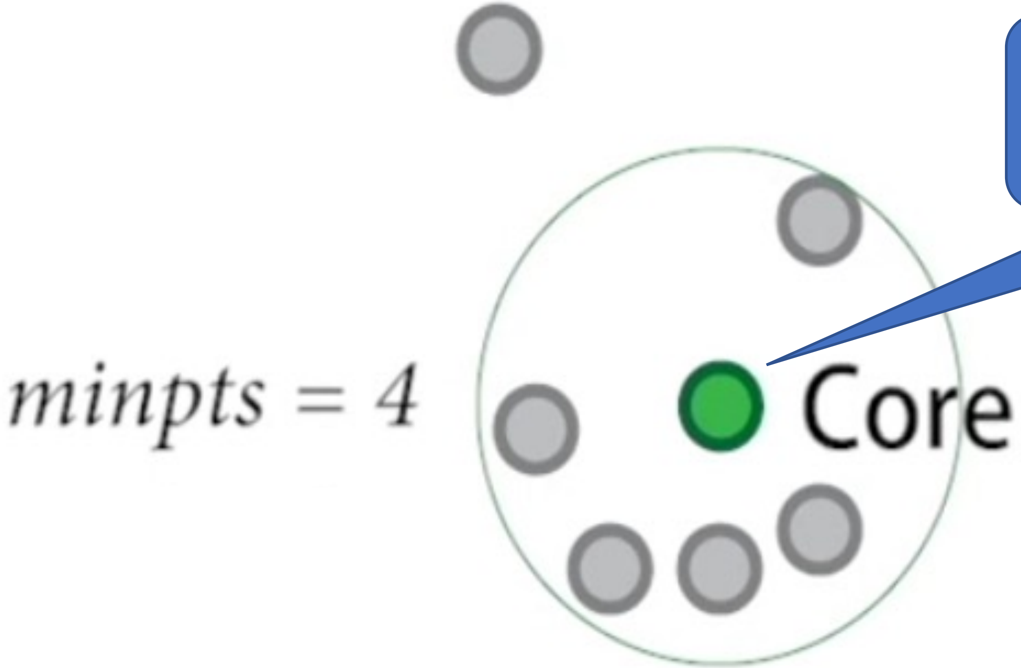
- Core point
- Boarder point
- Noise point

Neighbors

A *neighbor* of a point p is a point that is within distance ϵ from p.

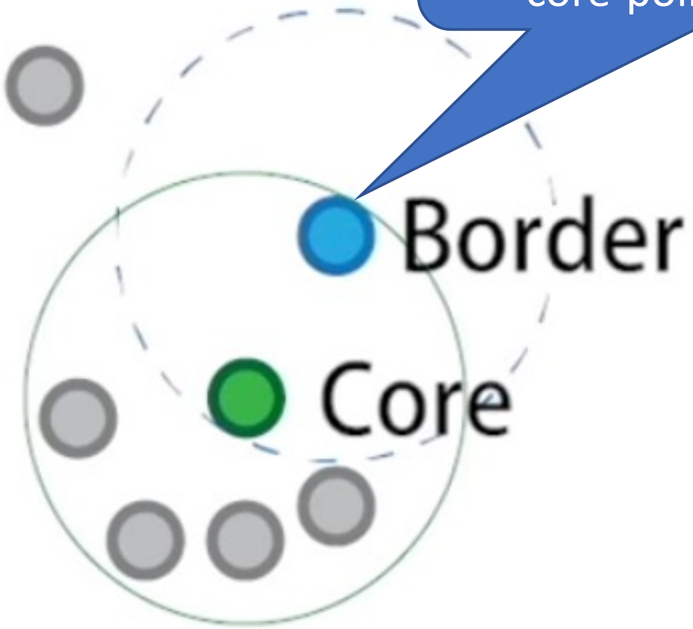


Core points



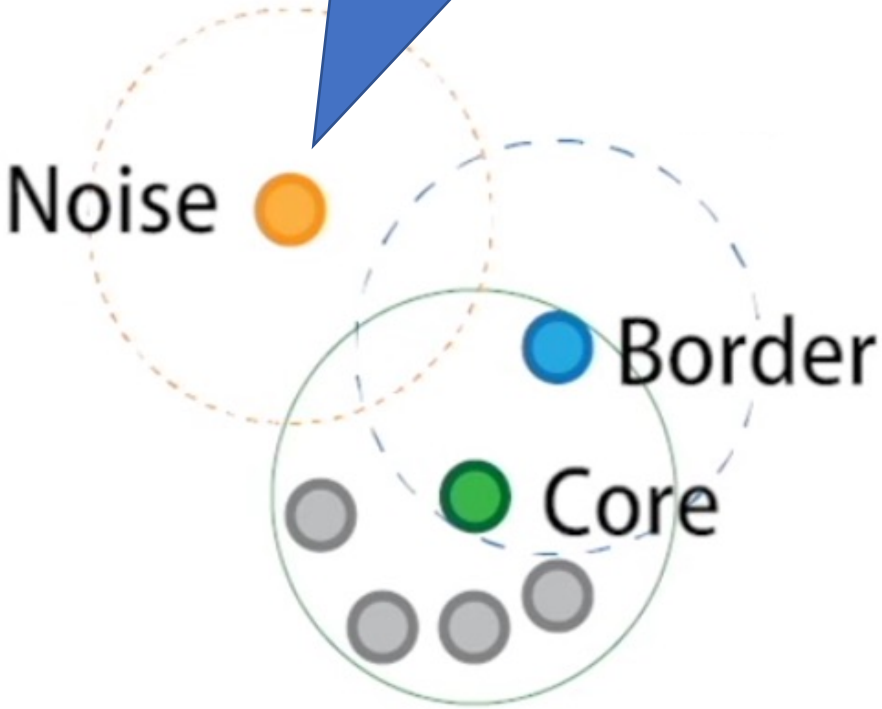
A *core* point is a point that has at least minpts neighbors

Border points



A *border* point is a non-core point that has at least one core-point as neighbor.

Noise points

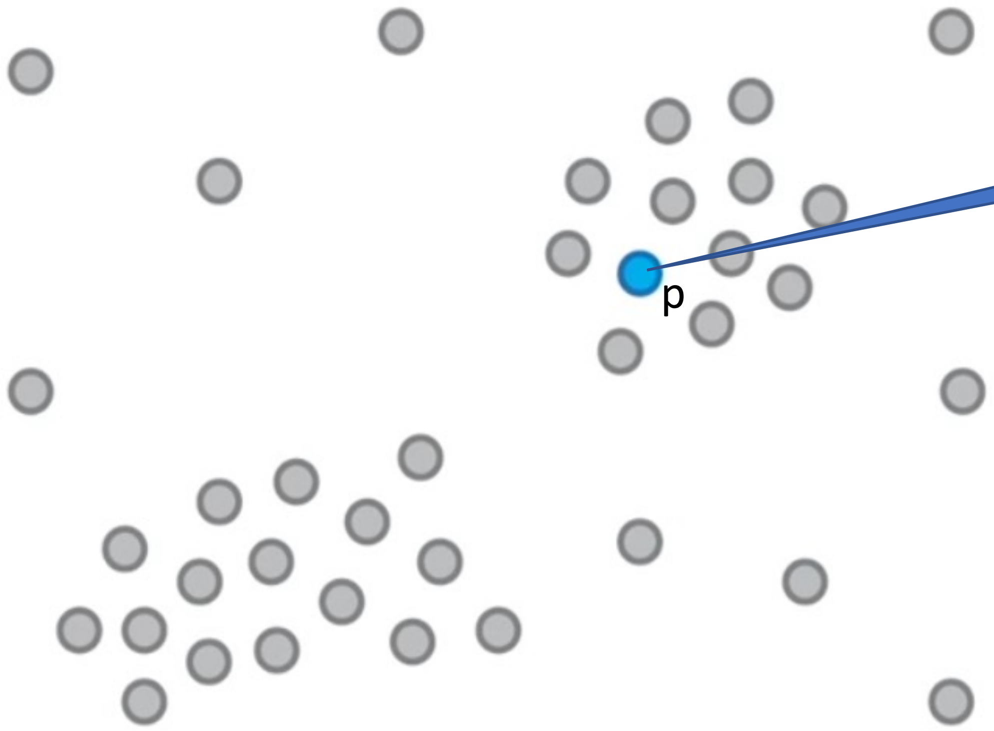


A *noise* point or *outlier* is a non-core and non-border point

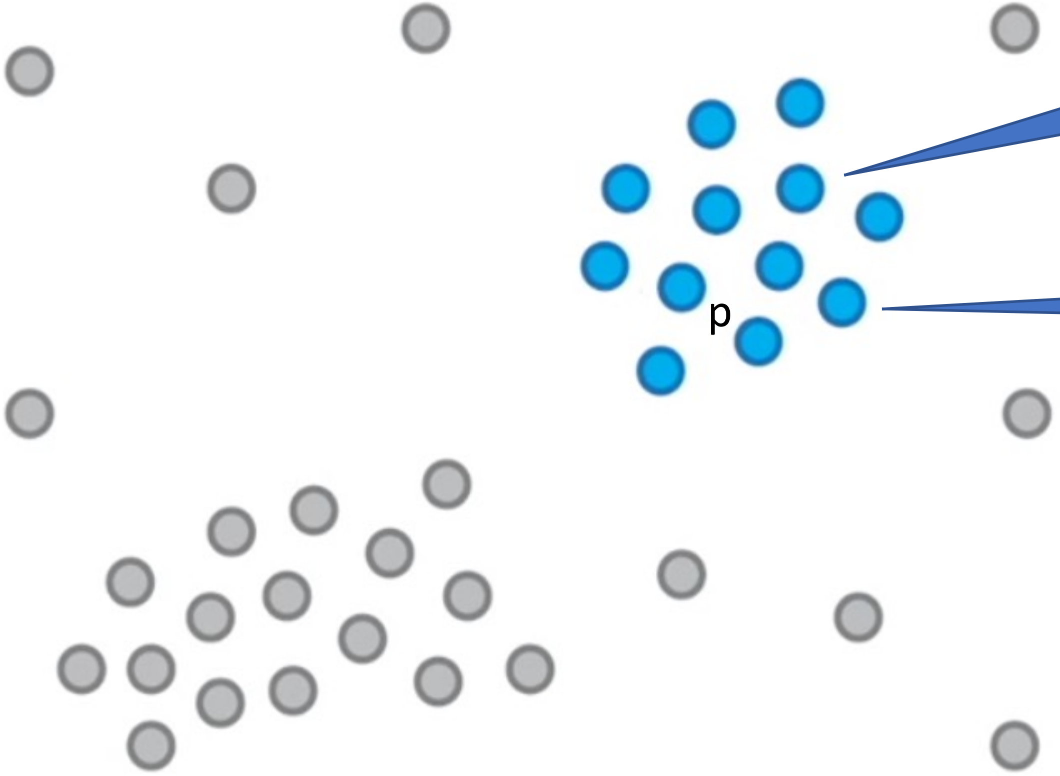
Clustering step

Clusters all core points and border points. Outliers will not be clustered!

Start by picking a new color c and an uncoloured core point p.



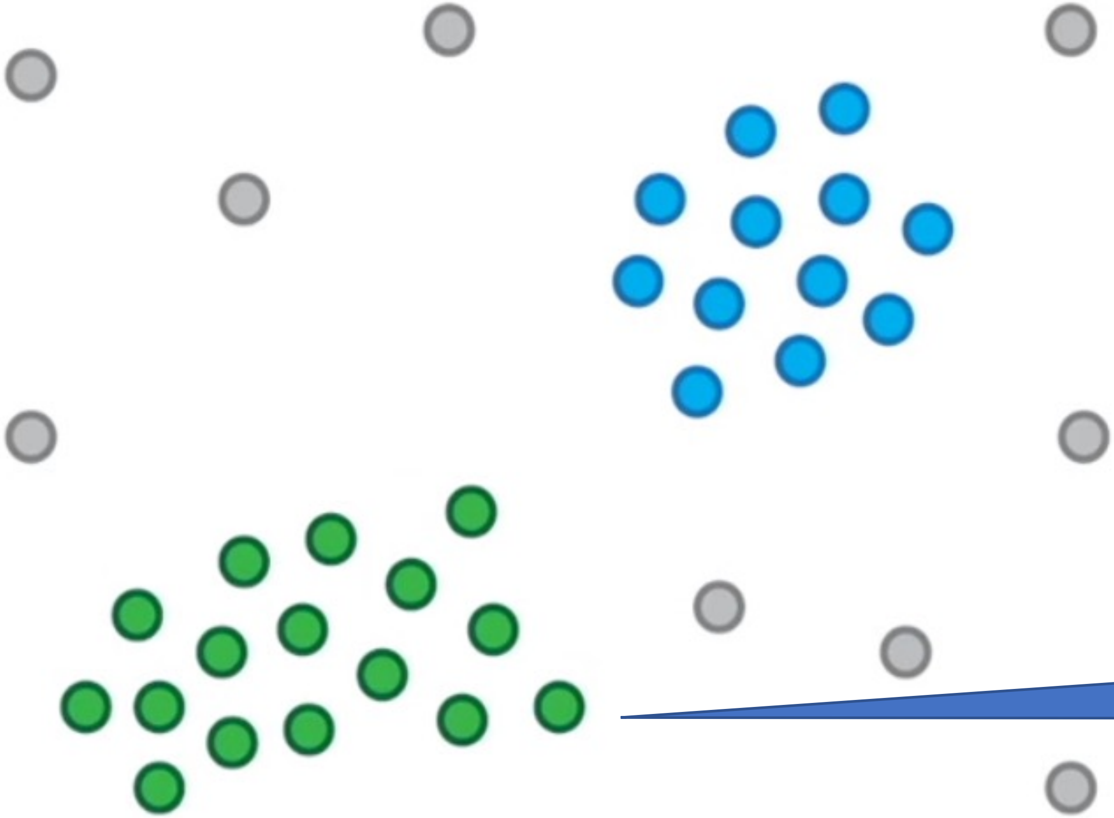
Clustering step



Put an edge between core points that are neighbors. Color those connected components with c

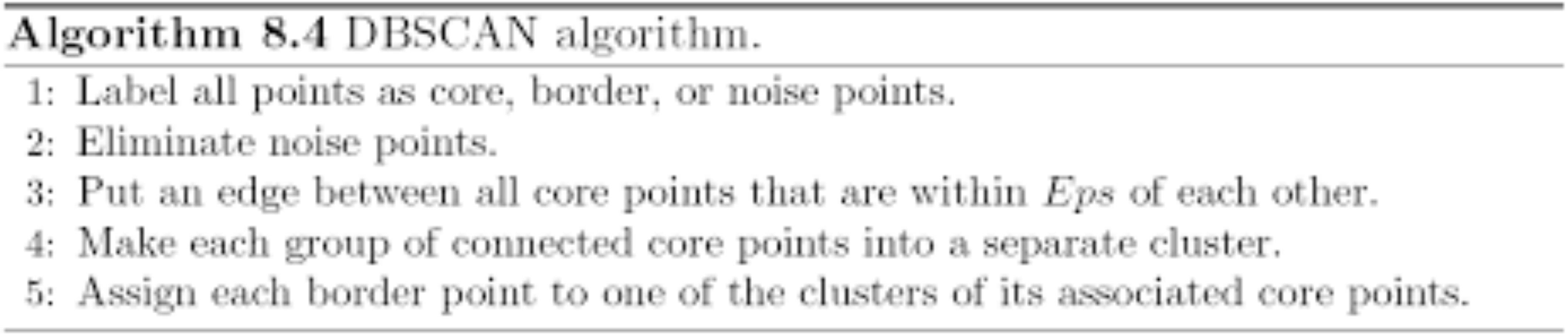
Also color the border points of those nodes with c

Clustering step



Repeat until all core points and border points have been colored!

Algorithm



Algorithm 8.4 DBSCAN algorithm.

1: Label all points as core, border, or noise points.

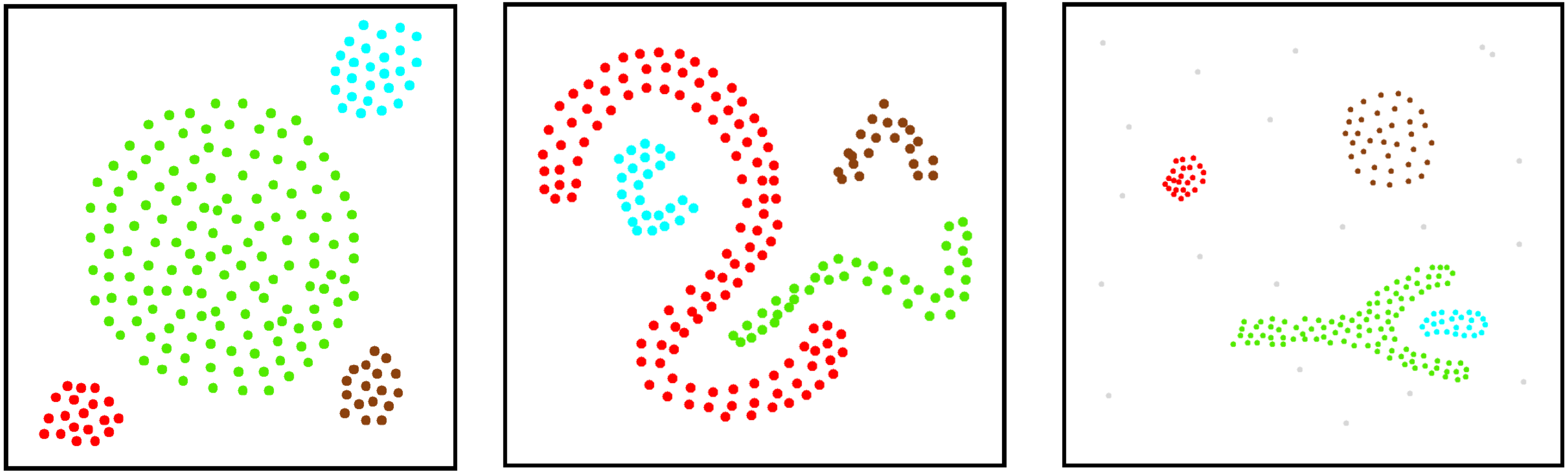
2: Eliminate noise points.

3: Put an edge between all core points that are within *Eps* of each other.

4: Make each group of connected core points into a separate cluster.

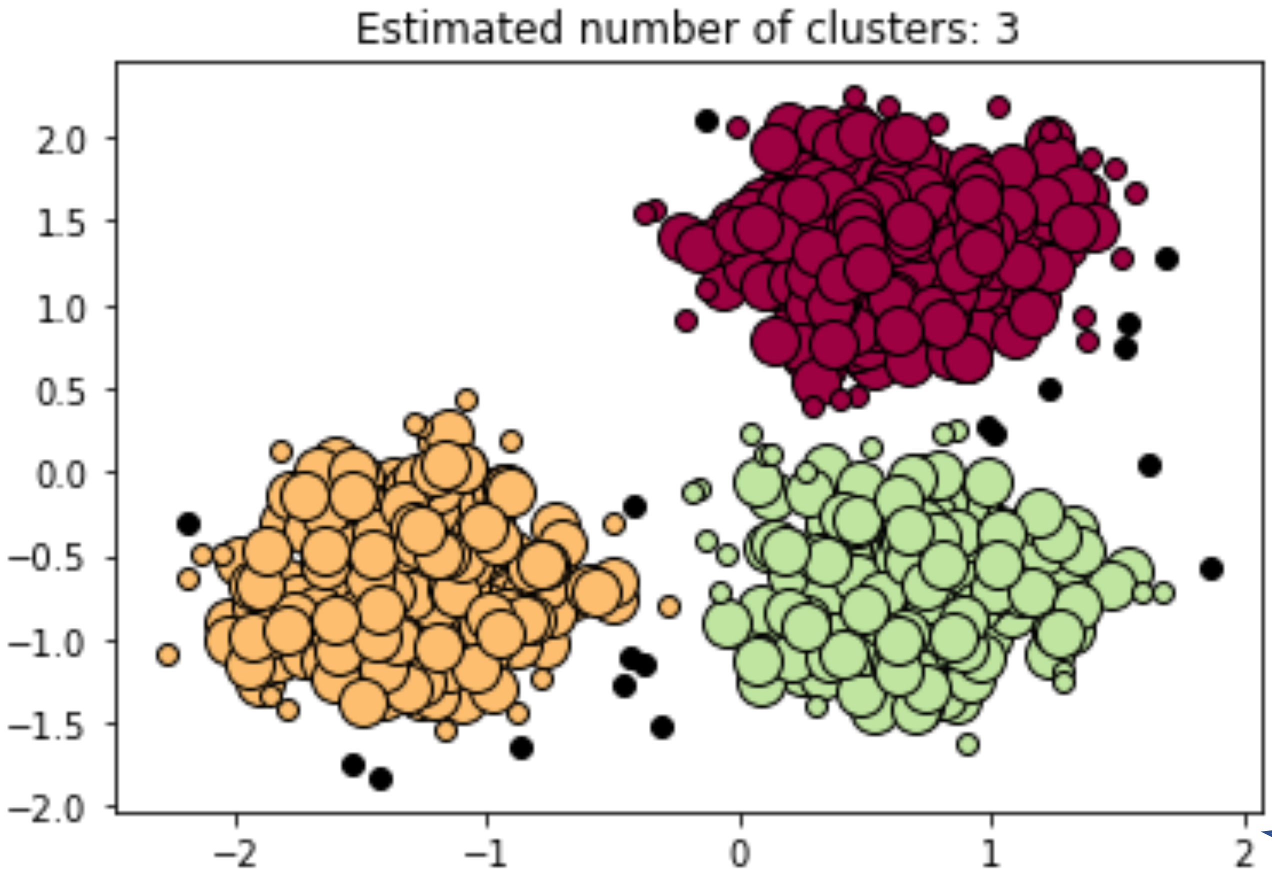
5: Assign each border point to one of the clusters of its associated core points.

Clusterings created by DBSCAN



Ester, Kriegel, Sander, Xu (1996), In Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, (KDD), AAAI Press, pp. 226–231

Using DBSCAN



Large = core point
Small = border point
Black = outlier

Note: data is standardised (scaled to range -2 – 2). This facilitates parameter search for epsilon.

See Jupyter notebooks for Module 3 for code examples

K-means vs. DBSCAN

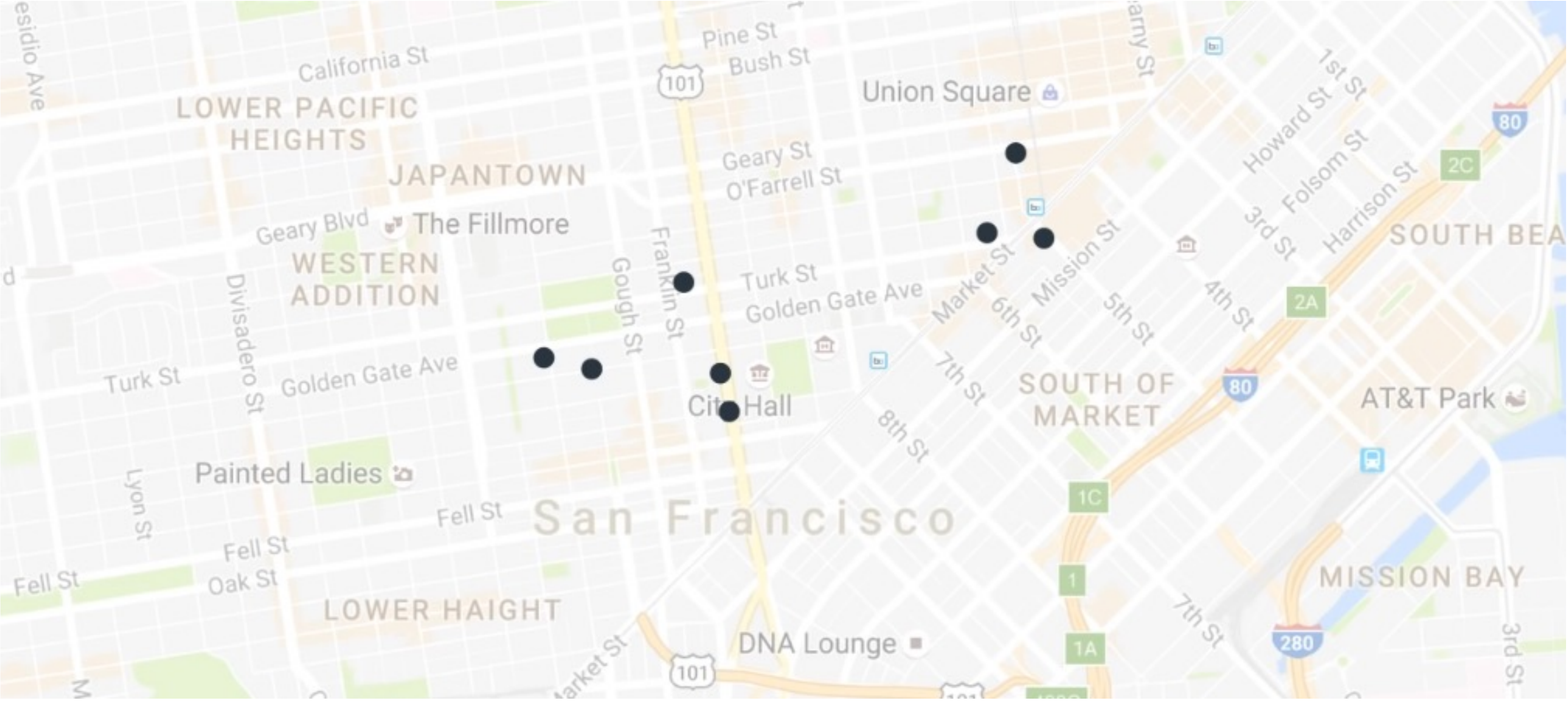
- K-means assigns all points to a cluster, whereas DBSCAN doesn’t necessarily do this. DBSCAN treats outliers as outliers.
- K-means works best when clusters are basically spherical. DBSCAN can find arbitrarily-shaped clusters.
- DBSCAN doesn’t require the number of clusters to be specified by the user.

Hierarchical clustering

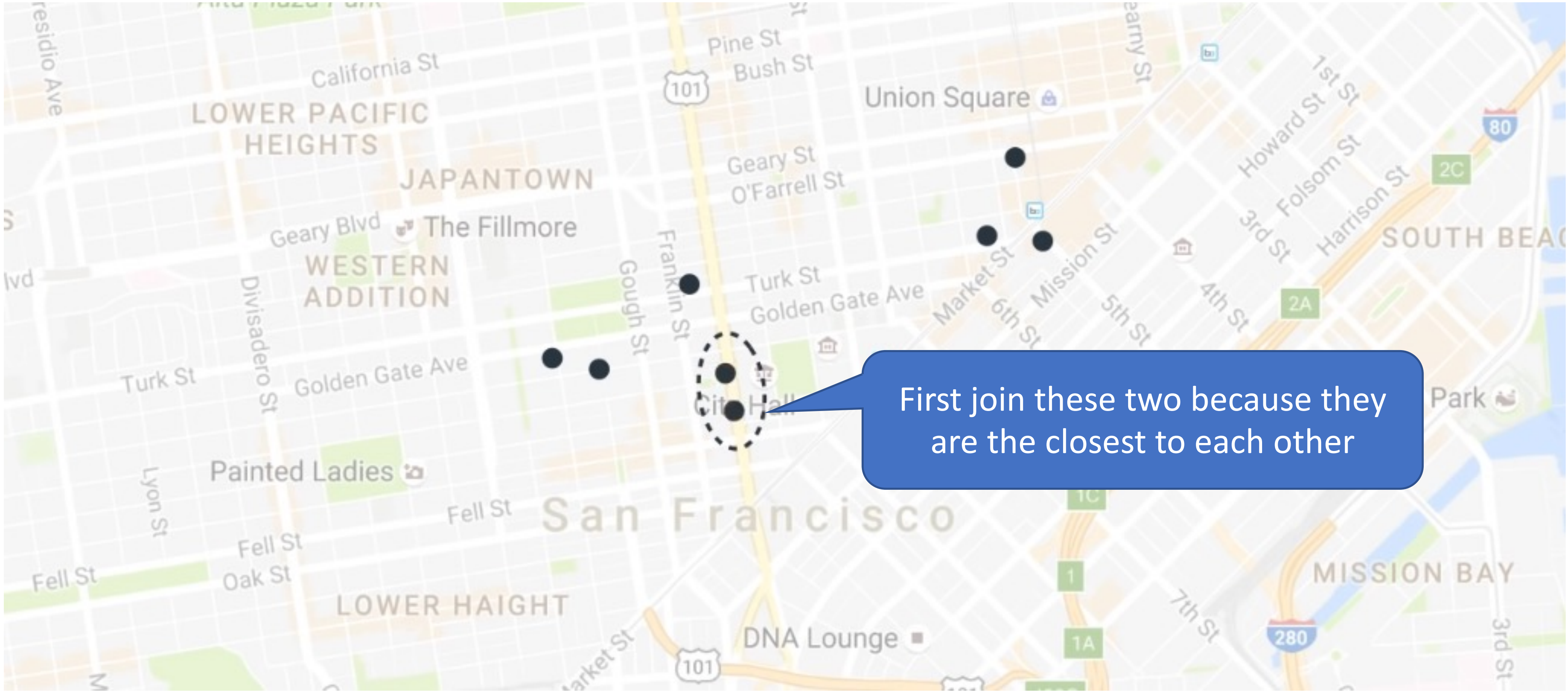
Luis Serrano

A new set of points

Suppose we want to cluster these addresses by proximity. No pizza parlors involved this time!

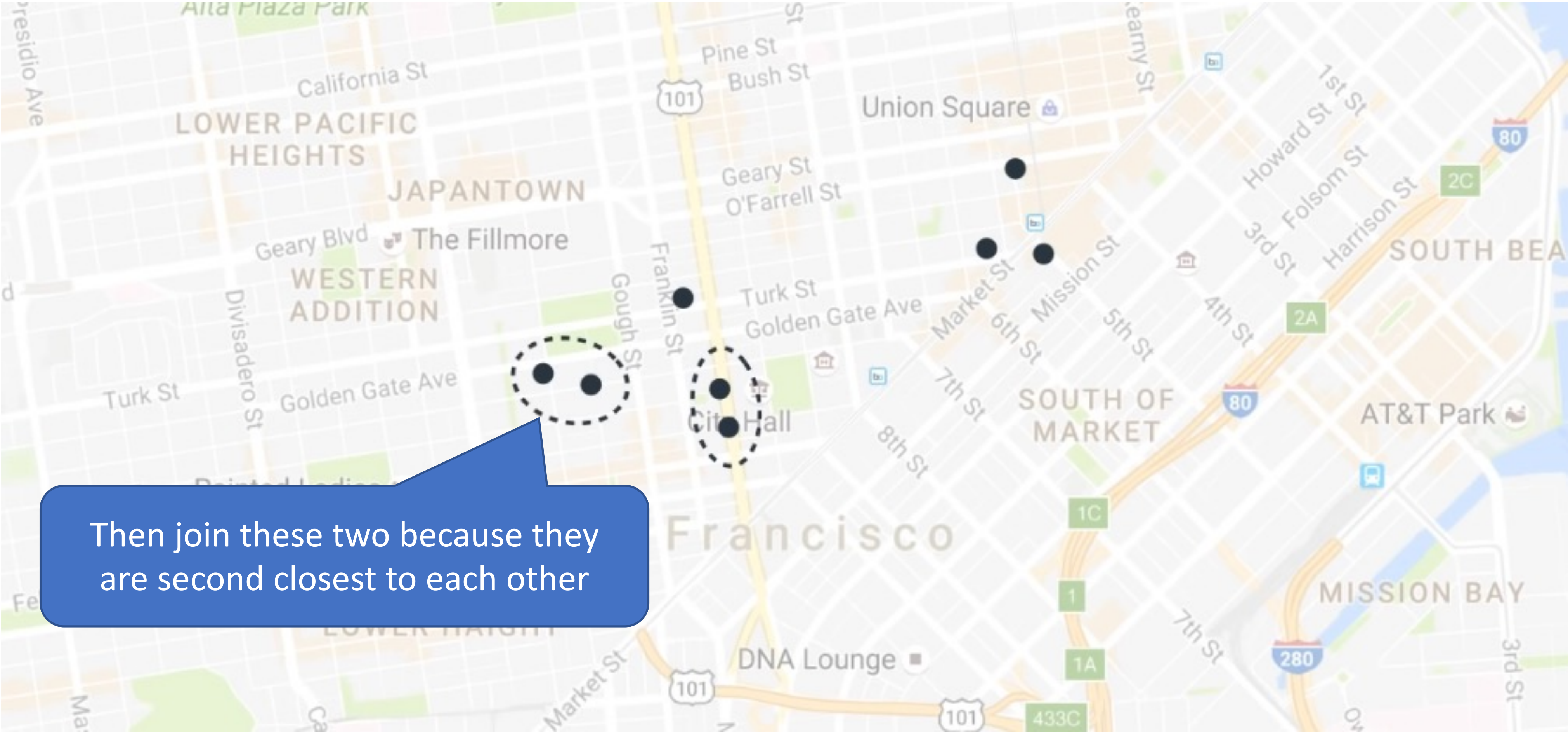


Join the close ones



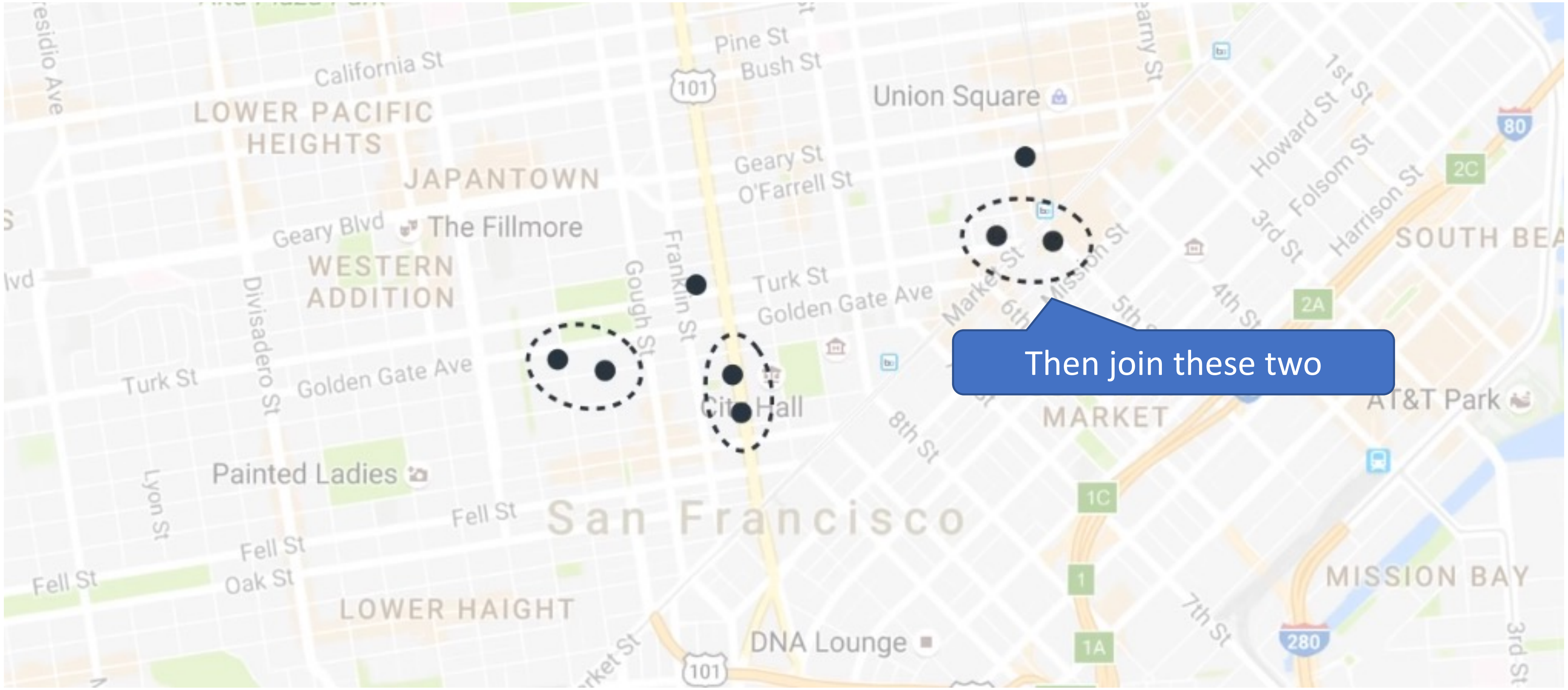
First join these two because they are the closest to each other

Join the close ones



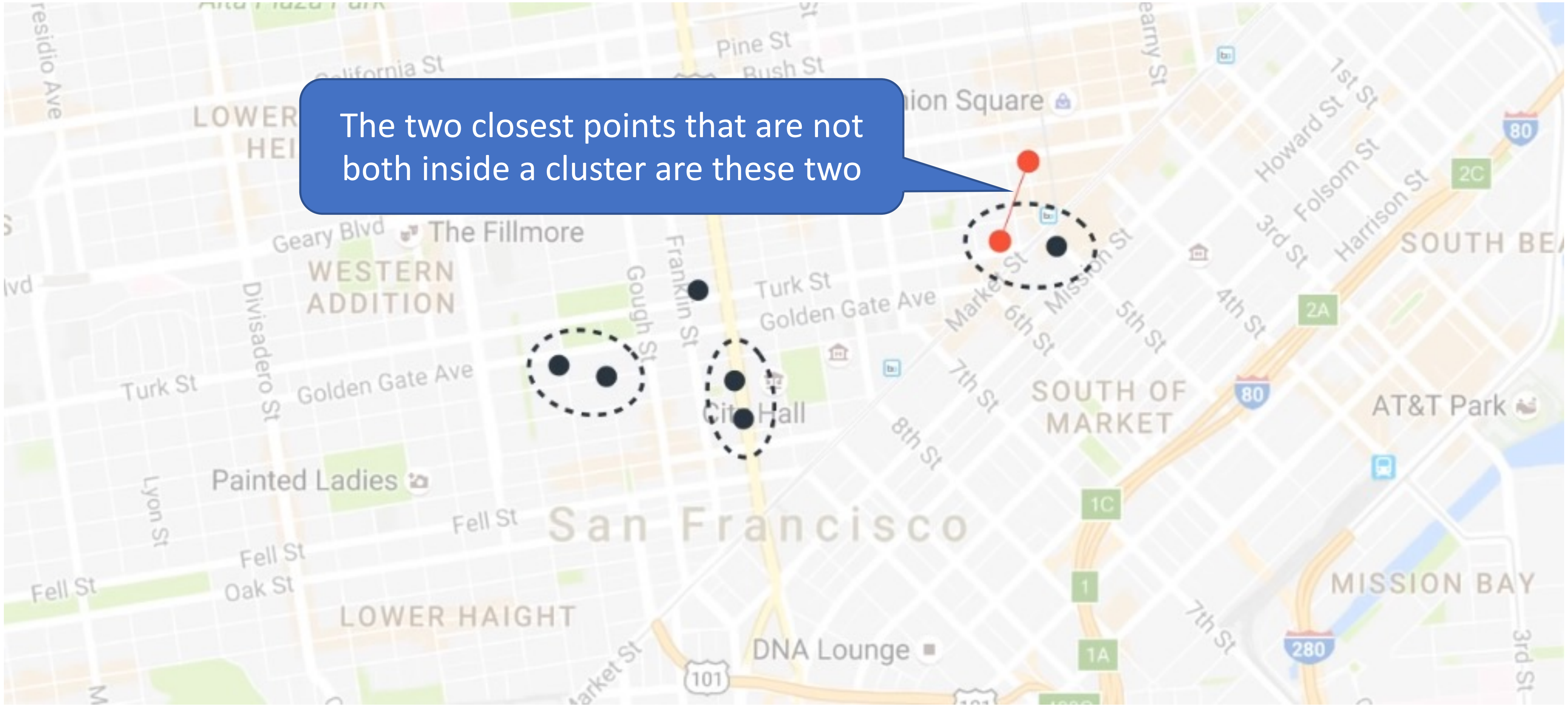
Then join these two because they are second closest to each other

Join the close ones



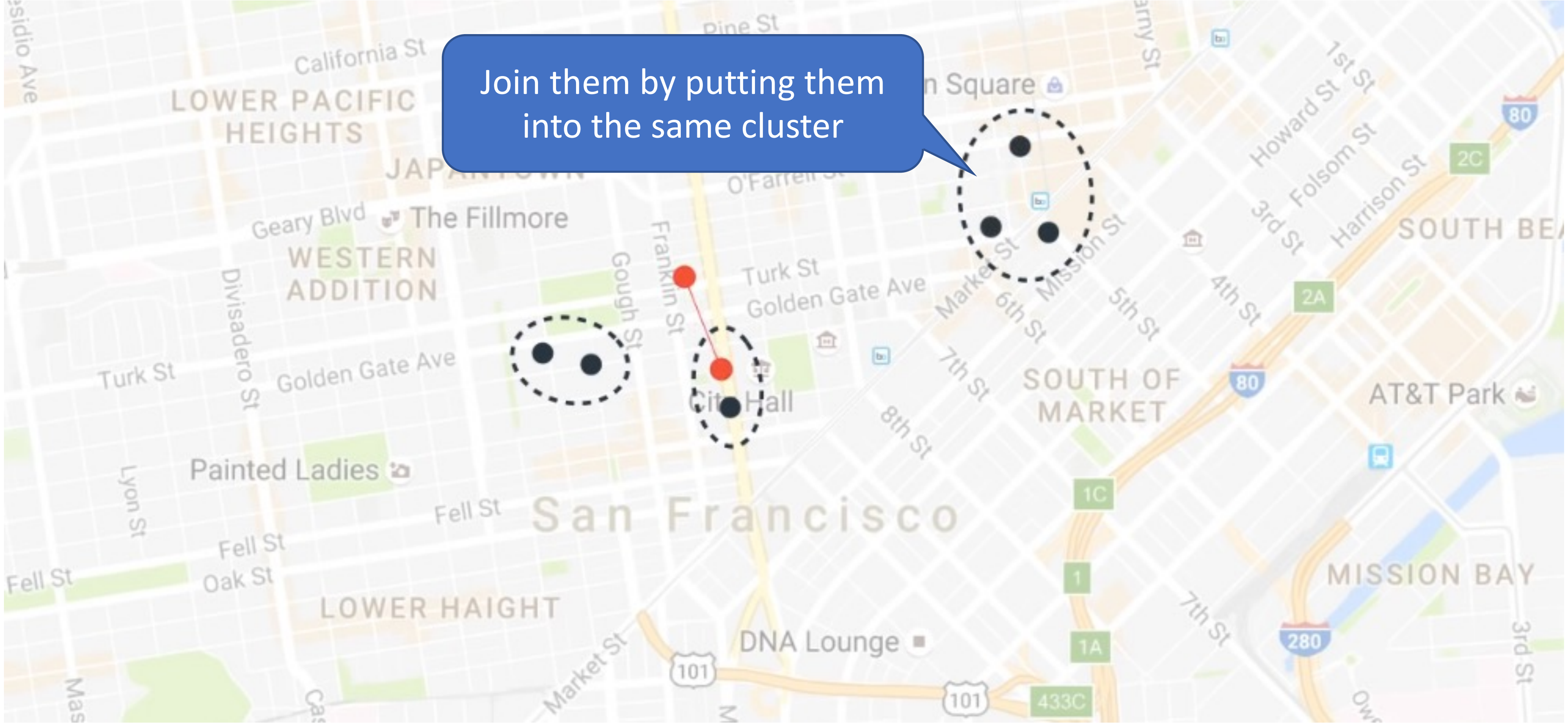
Then join these two

Join the close ones



The two closest points that are not both inside a cluster are these two

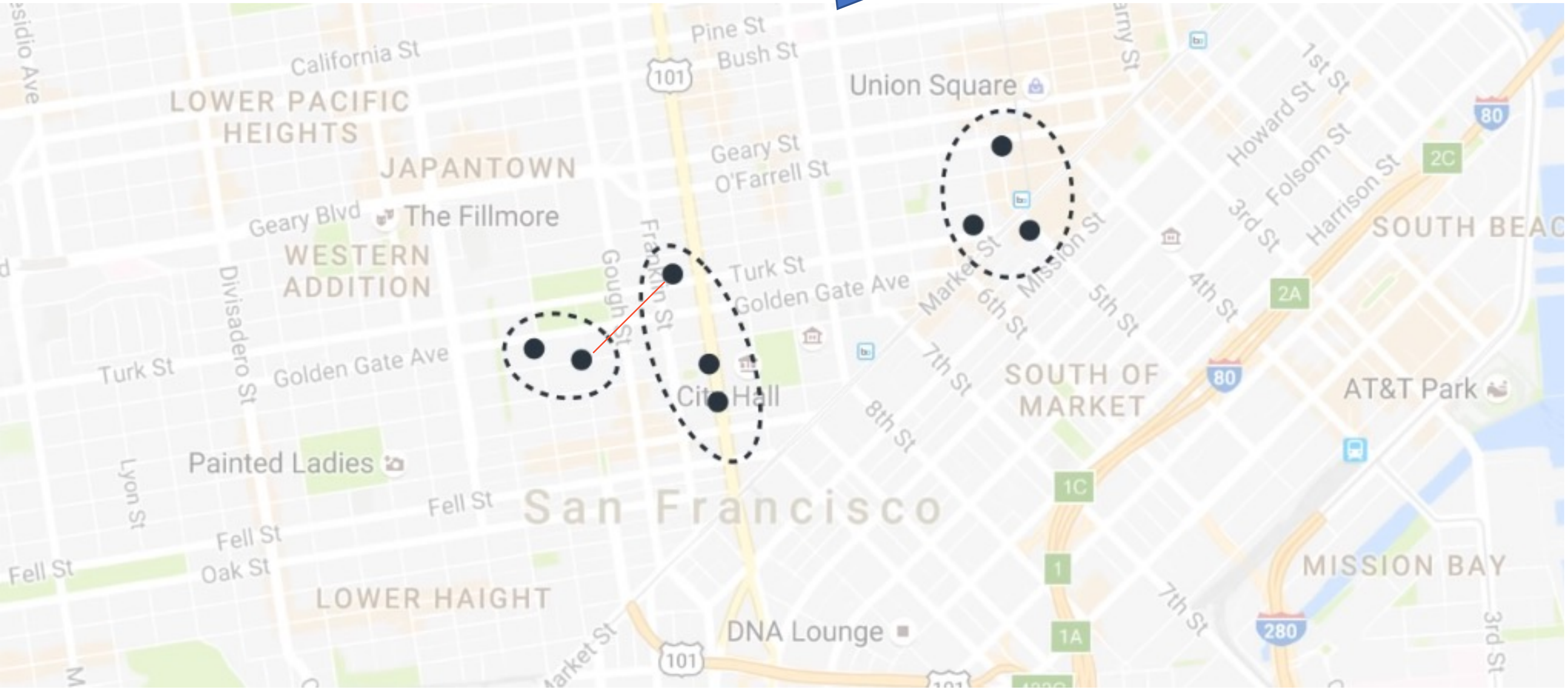
Join the close ones



Join them by putting them into the same cluster

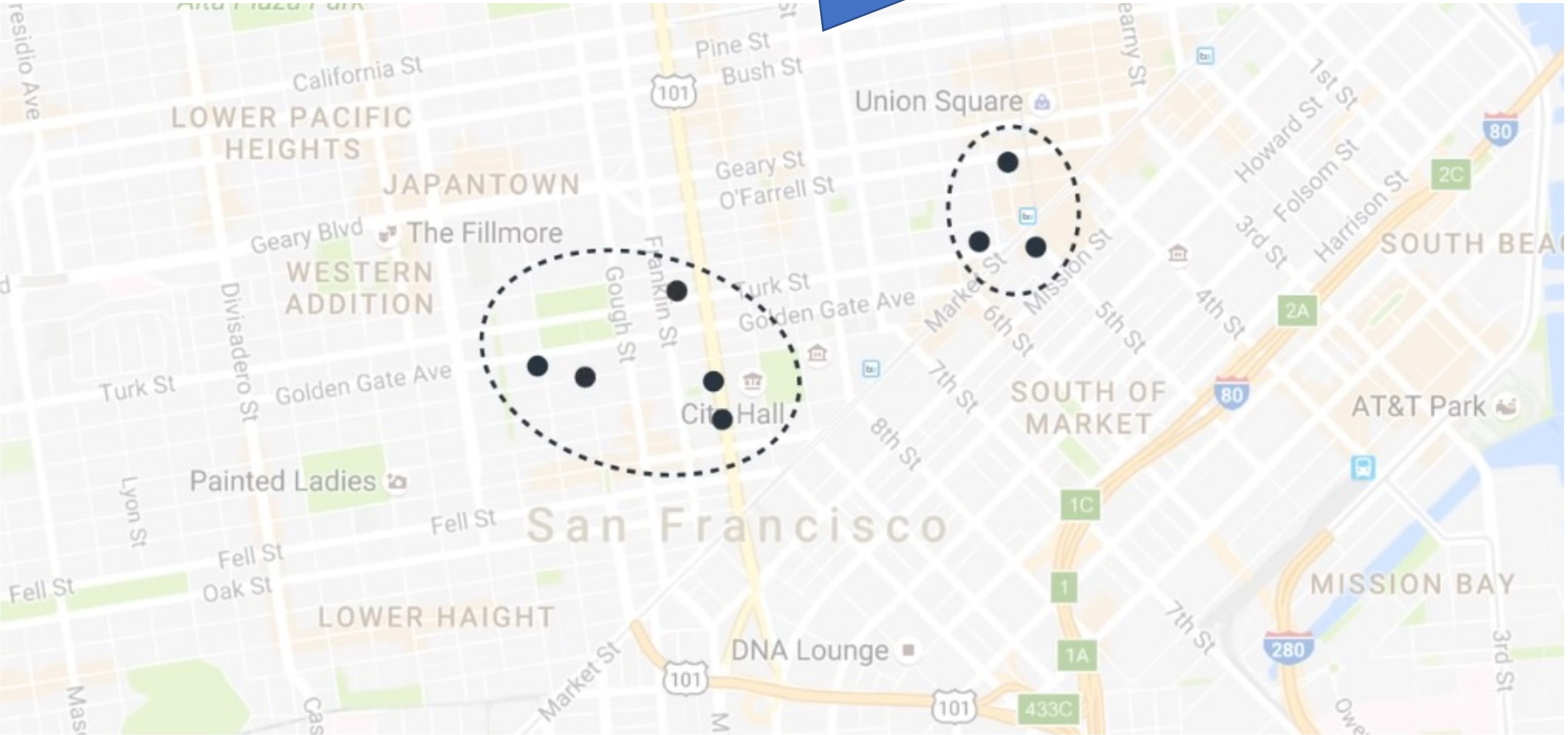
Join the close ones

Continue in the same way…



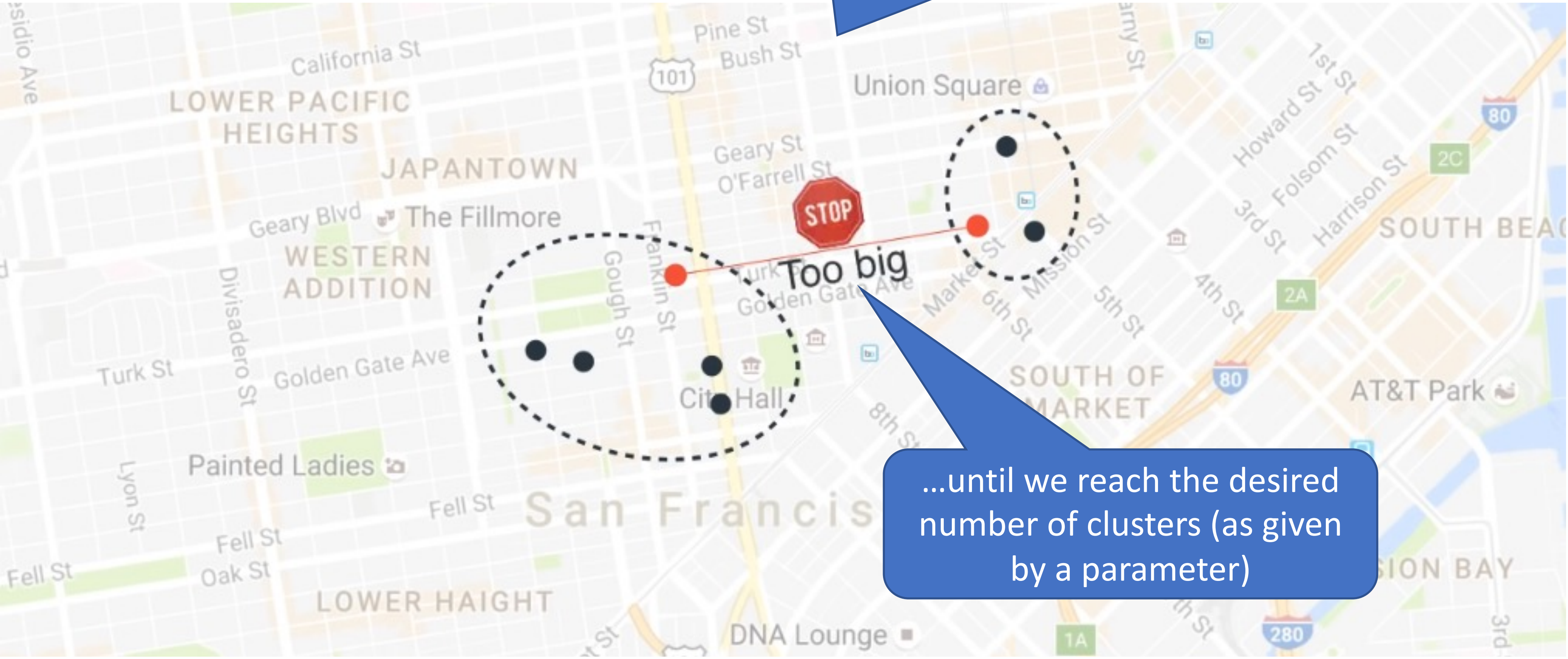
Join the close ones

Continue in the same way…



Join the close ones

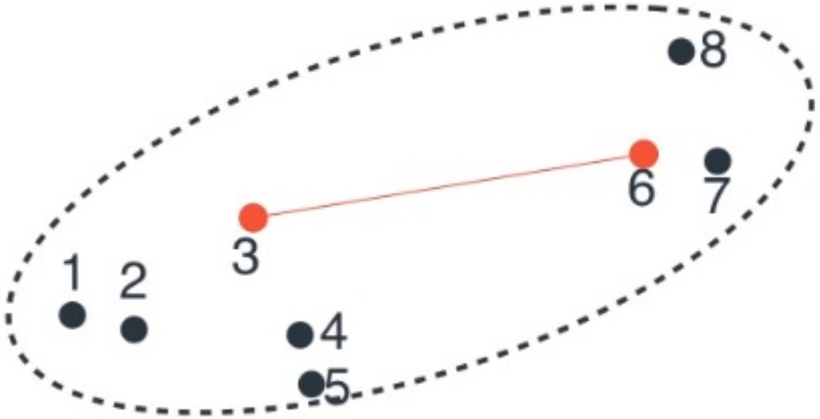
Continue in the same way…

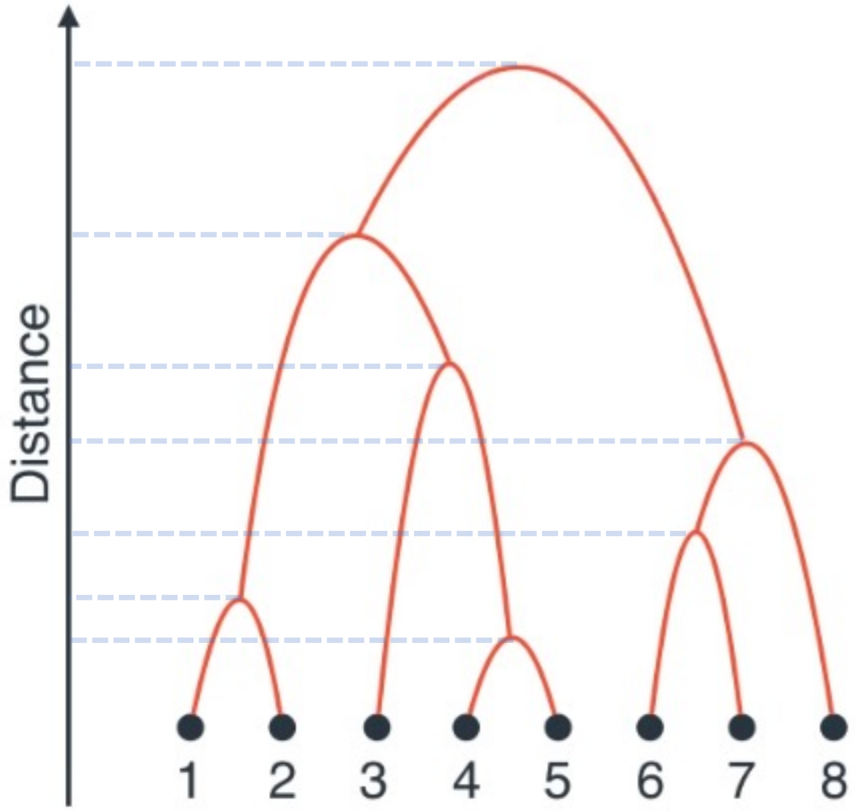


…until we reach the desired number of clusters (as given by a parameter)

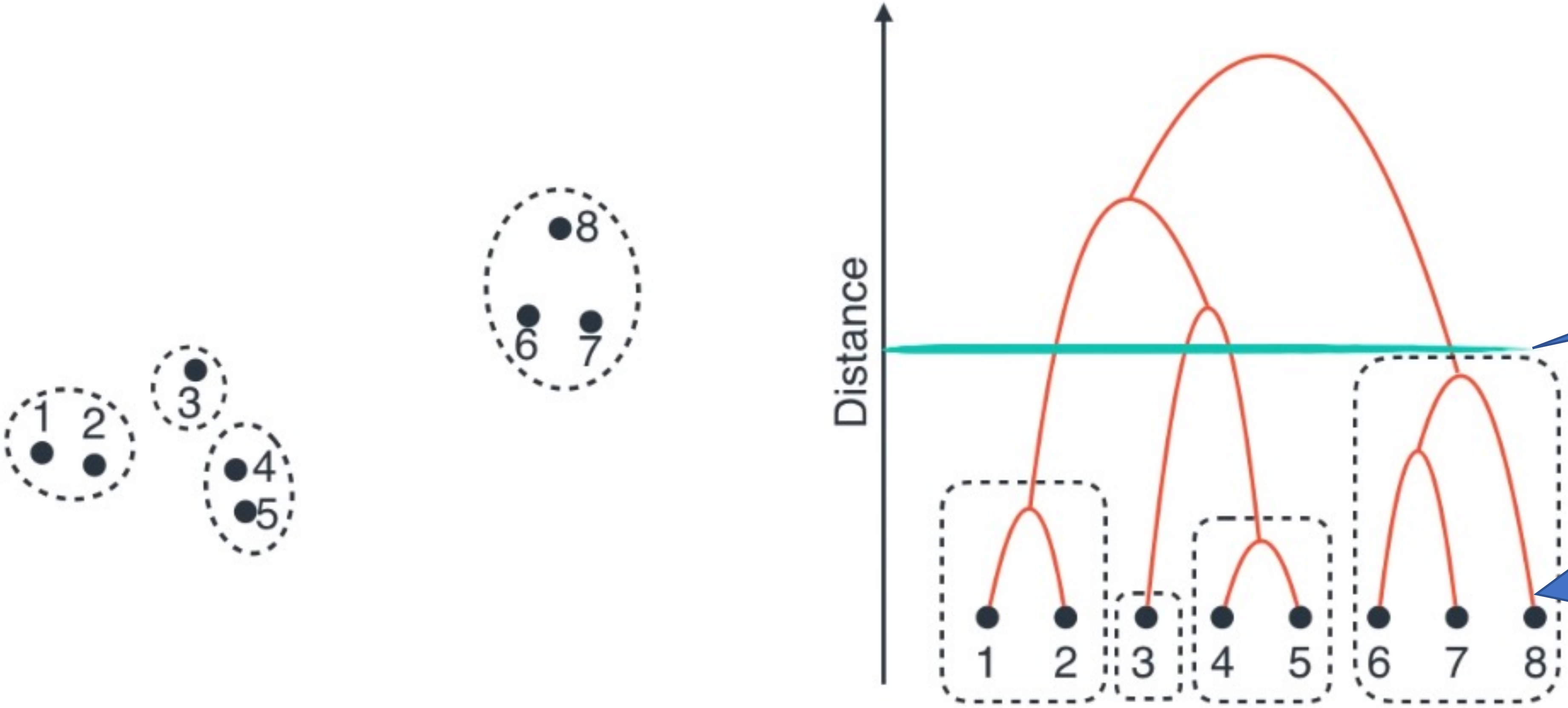
Dendrogram

A *dendrogram* shows the entire hierarchichal clustering process (without STOP)





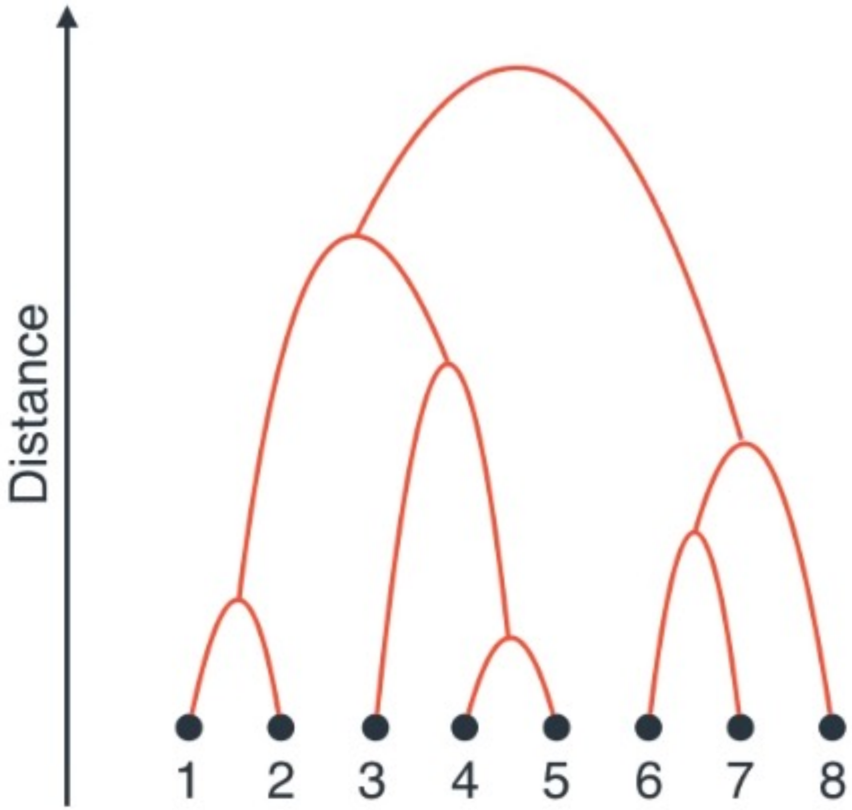
Dendrogram



If we want 4 clusters, we can cut here.

If we want 8 clusters, we can cut here.

Dendrogram

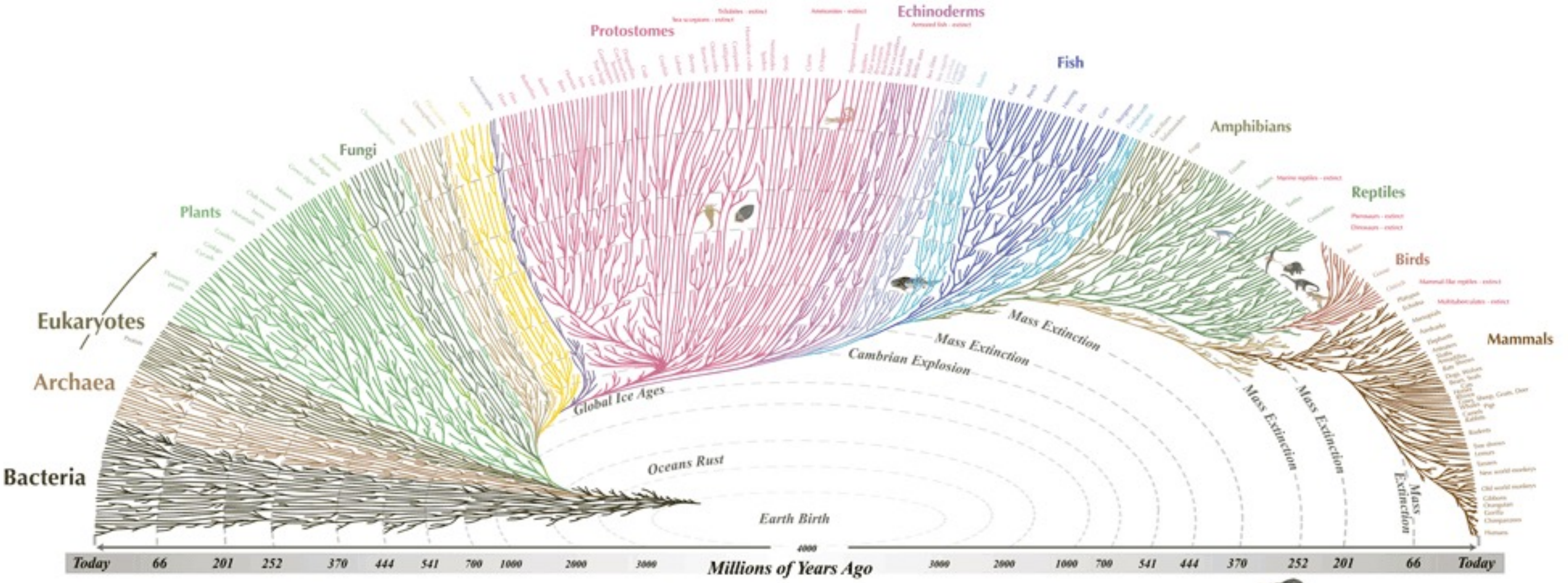


If we have a space with billions of points in thousands of dimensions, the dendrogram is still a 2D graph!

For example the tree of life!

Hierarchichal clustering gives more than a clustering: a hierarchy (or taxonomy)

The tree of life



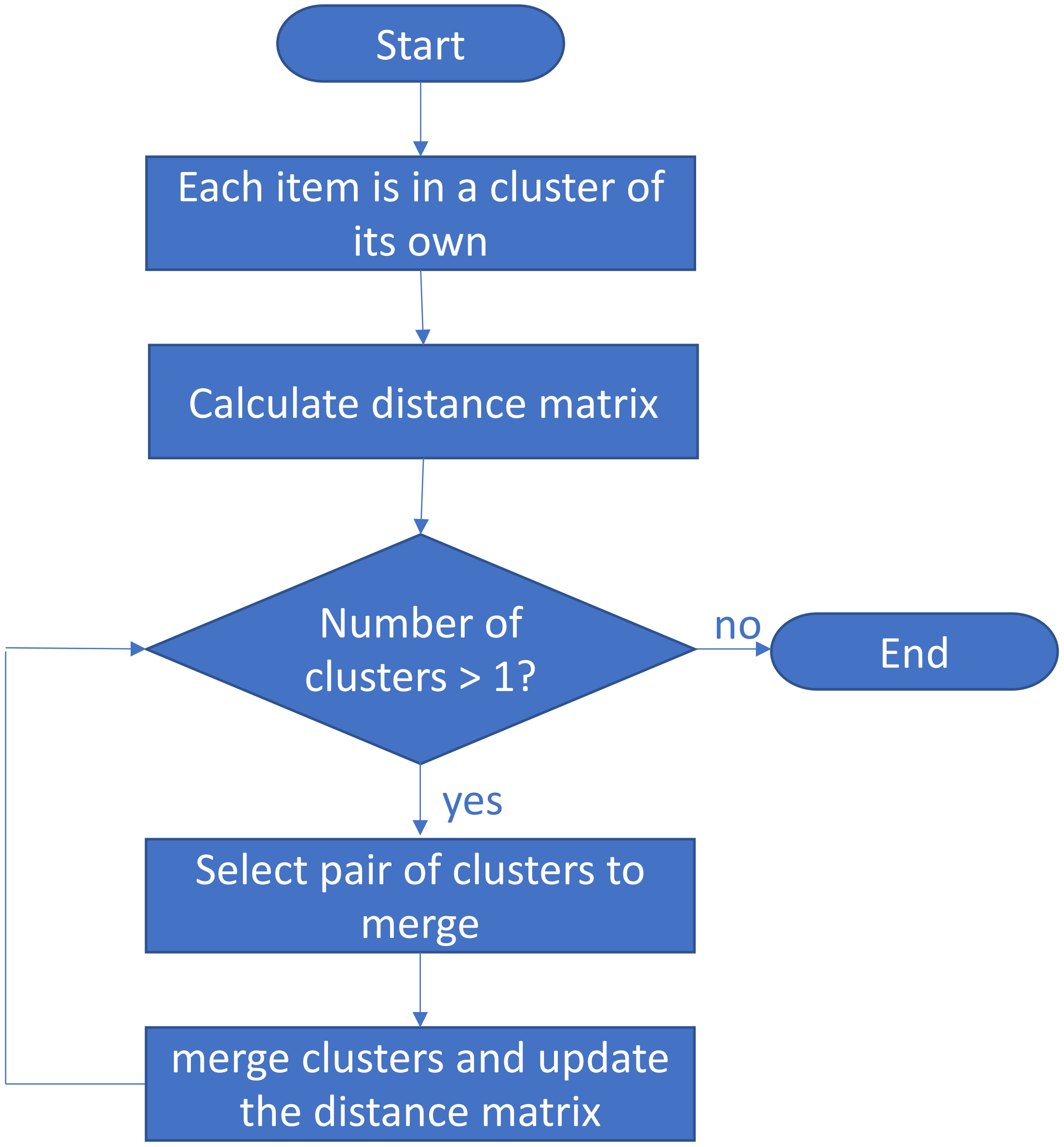
All the major and many of the minor living branches of life are shown on this diagram, but only a few of those that have gone extinct are shown. Example: Dinosaurs - extinct

https://www.evogeneao.com/

Hierarchical clustering

- Sometimes called agglomerative clustering, when done bottom-up

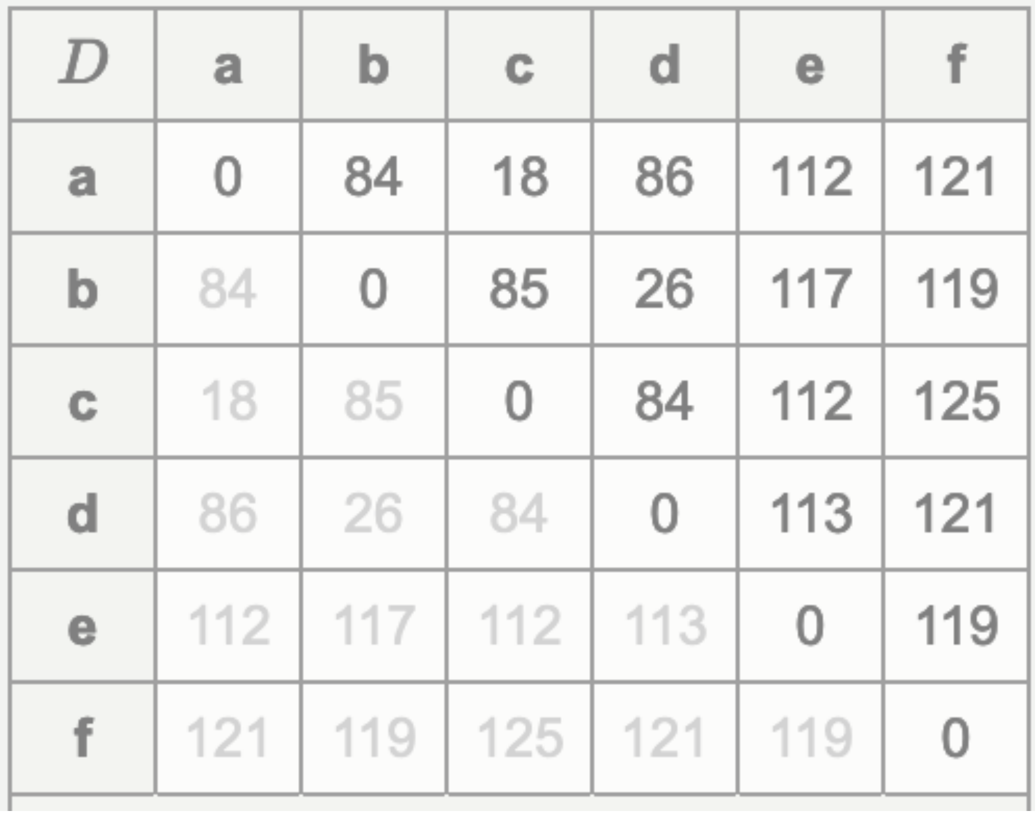
- From one extreme case (many clusters, each containing one item) to another (one cluster that contains all items)


Start
Each item is in a cluster of its own
Calculate distance matrix
Number of clusters > 1?
no
End
yes
Select pair of clusters to merge
merge clusters and update the distance matrix

Distance matrix

Edit distances between protein sequences (strings)

a. Human haemoglobin alpha chain
b. Human haemoglobin beta chain
c. Horse haemoglobin alpha chain
d. Horse haemoglobin beta chain
e. Marine bloodworm haemoglobin
f. Yellow lupine leghaemoglobin



D	a	b	c	d	e	f
a	0	84	18	86	112	121
b	84	0	85	26	117	119
c	18	85	0	84	112	125
d	86	26	84	0	113	121
e	112	117	112	113	0	119
f	121	119	125	121	119	0

Six proteins with a common evolutionary ancestor

Amino acid sequences of six proteins

```
> human_alpha

VLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR

> human_beta

VHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH

> horse_alpha

VLSAADKTNVKAAWSKVGGHAGEYGAEALERMFLGFPTTKTYFPHFDLSHGSAQVKAHGKKVADGLTLAVGHLDDLPGALSDLSNLHAHKLRVDPVNFKLLSHCLLSTLAVHLPNDFTPAVHASLDKFLSSVSTVLTSKYR

> horse_beta

VQLSGEEKAAVLALWDKVNEEEVGGEALGRLLVVYPWTQRFFDSFGDLSNPGAVMGNPKVKAHGKKVLHSFGEGVHHLDNLKGTFAALSELHCDKLHVDPENFRLLGNVLALVVARHFGKDFTPELQASYQKVVAGVANALAHKYH

> marine_bloodworm

GLSAAQRQVIAATWKDIAGADNGAGVGKKCLIKFLSAHPQMAAVFGFSGASDPGVAALGAKVLAQIGVAVSHLGDEGKMVAQMKAVGVRHKGYGNKHIKAQYFEPLGASLLSAMEHRIGGKMNAAAKDAWAAAYADISGALISGLQS

> yellow_lupine

GALTESQAALVKSSWEEFNANIPKHTHRFFILVLEIAPAAKDLFSFLKGTSEVPQNNPELQAHAGKVFKLVYEAAIQLEVTGVVVTDATLKNLGSVHVSKGVADAHFPVVKEAILKTIKEVVGAKWSEELNSAWTIAYDELAIVIKKEMDDAA
```

Edit distance is the number of single character operations that are required to change one string into another.

Merging clusters

- When clusters **u** and **v** are merged, how do we calculate the distance between the merged cluster and each of the other clusters?

- Various algorithms to choose from, e.g.
  - complete linkage (furthest inter-cluster distance) *max(dist(u[i]), v[j]))*
  - single linkage (closest inter-cluster distance) *min(dist(u[i]), v[j]))*
  - average linkage
    - Unweighted Pair Group Method with Arithmetic Mean (UPGMA)
    - Weighted Pair Group Method with Arithmetic Mean (WPGMA)
  - … and many more

See e.g.
https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html#scipy.cluster.hierarchy.linkage

Example: merging clusters

D	a	b	c	d	e	f
a	0	84	18	86	112	121
b		0	85	26	117	119
c			0	84	112	125
d				0	113	121
e					0	119
f						0

1) Shortest distance a – c
2) Merge {a,c}
3) Recompute distance matrix, use max distance (complete linkage)

D	a,c	b	d	e	f
a,c	0	**85**	**86**	**112**	**125**
b		0	26	117	119
d			0	113	121
e				0	119
f					0

a. Human haemoglobin alpha chain
b. Human haemoglobin beta chain
c. Horse haemoglobin alpha chain
d. Horse haemoglobin beta chain
e. Marine bloodworm haemoglobin
f. Yellow lupine leghaemoglobin

Example: merging clusters

D	a	b	c	d	e	f
a	0	84	18	86	112	121
b		0	85	26	117	119
c			0	84	112	125
d				0	113	121
e					0	119
f						0

1) Shortest distance a – c
2) Merge {a,c}
3) Recompute distance matrix, use max distance between points

D	a,c	b	d	e	f
a,c	0	85	86	112	121
b		0	26	117	119
d			0	113	121
e				0	119
f					0

1) Shortest distance b – d
2) Merge {b,d}
3) Recompute distance matric

a. Human haemoglobin alpha chain
b. Human haemoglobin beta chain
c. Horse haemoglobin alpha chain
d. Horse haemoglobin beta chain
e. Marine bloodworm haemoglobin
f. Yellow lupine leghaemoglobin

Example: merging clusters

D	a	b	c	d	e	f
a	0	84	18	86	112	121
b		0	85	26	117	119
c			0	84	112	125
d				0	113	121
e					0	119
f						0

1) Shortest distance a – c
2) Merge {a,c}
3) Recompute distance matrix, use max distance between points

D	a,c	b	d	e	f
a,c	0	85	86	112	121
b		0	26	117	119
d			0	113	121
e				0	119
f					0

1) Shortest distance b – d
2) Merge {b,d}
3) Recompute distance matric

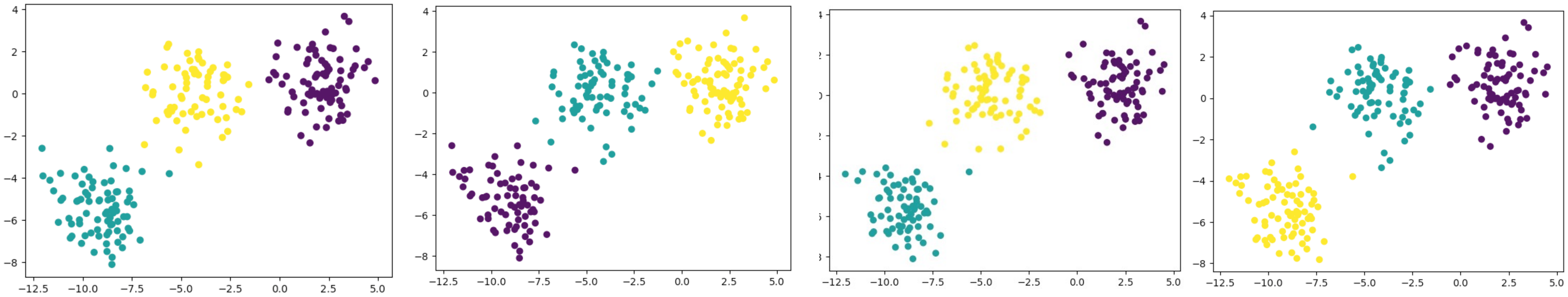
D	a,c	b,d	e	f
a,c	0	**86**	112	121
b,d		0	**117**	**121**
e			0	119
f				0

a. Human haemoglobin alpha chain
b. Human haemoglobin beta chain
c. Horse haemoglobin alpha chain
d. Horse haemoglobin beta chain
e. Marine bloodworm haemoglobin
f. Yellow lupine leghaemoglobin

Validating clustering

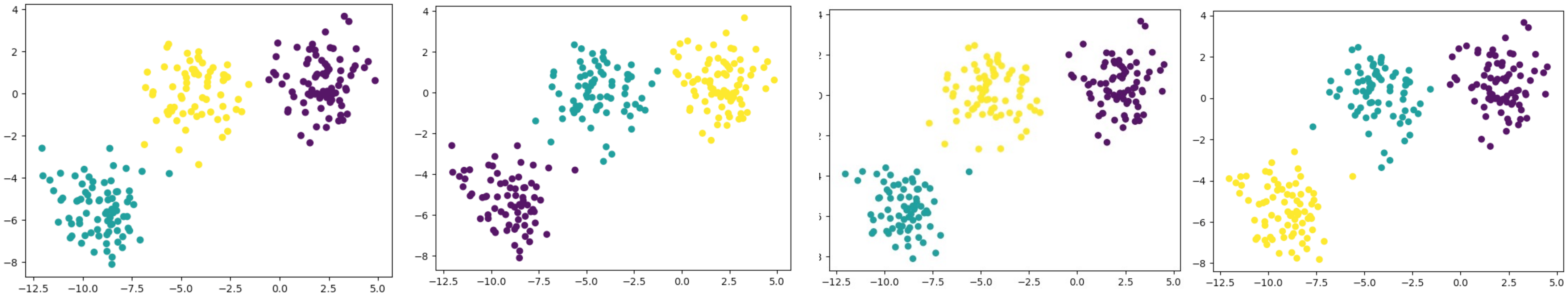
Stability on subsets

Clustering stable if removing a proportion of random points does not change the clustering fundamentally



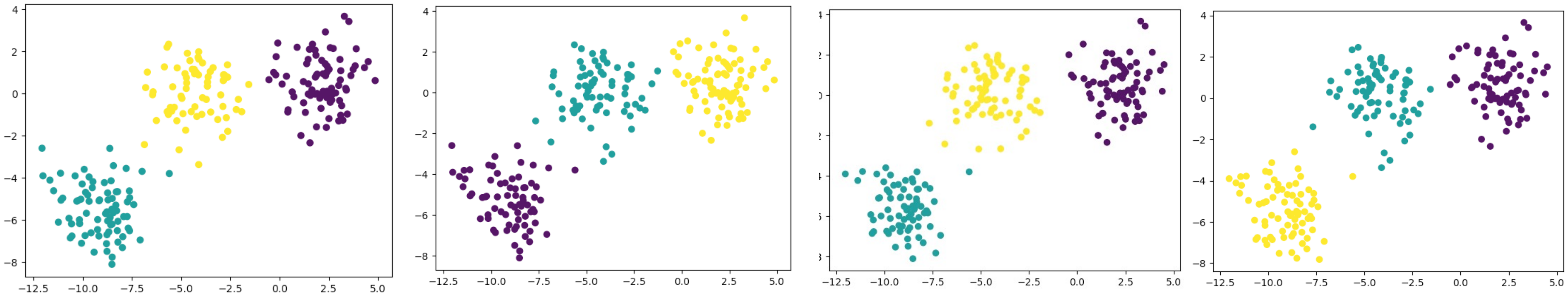
Stability on subsets

Note colors change as labeling clusters into first, second, third … changes!

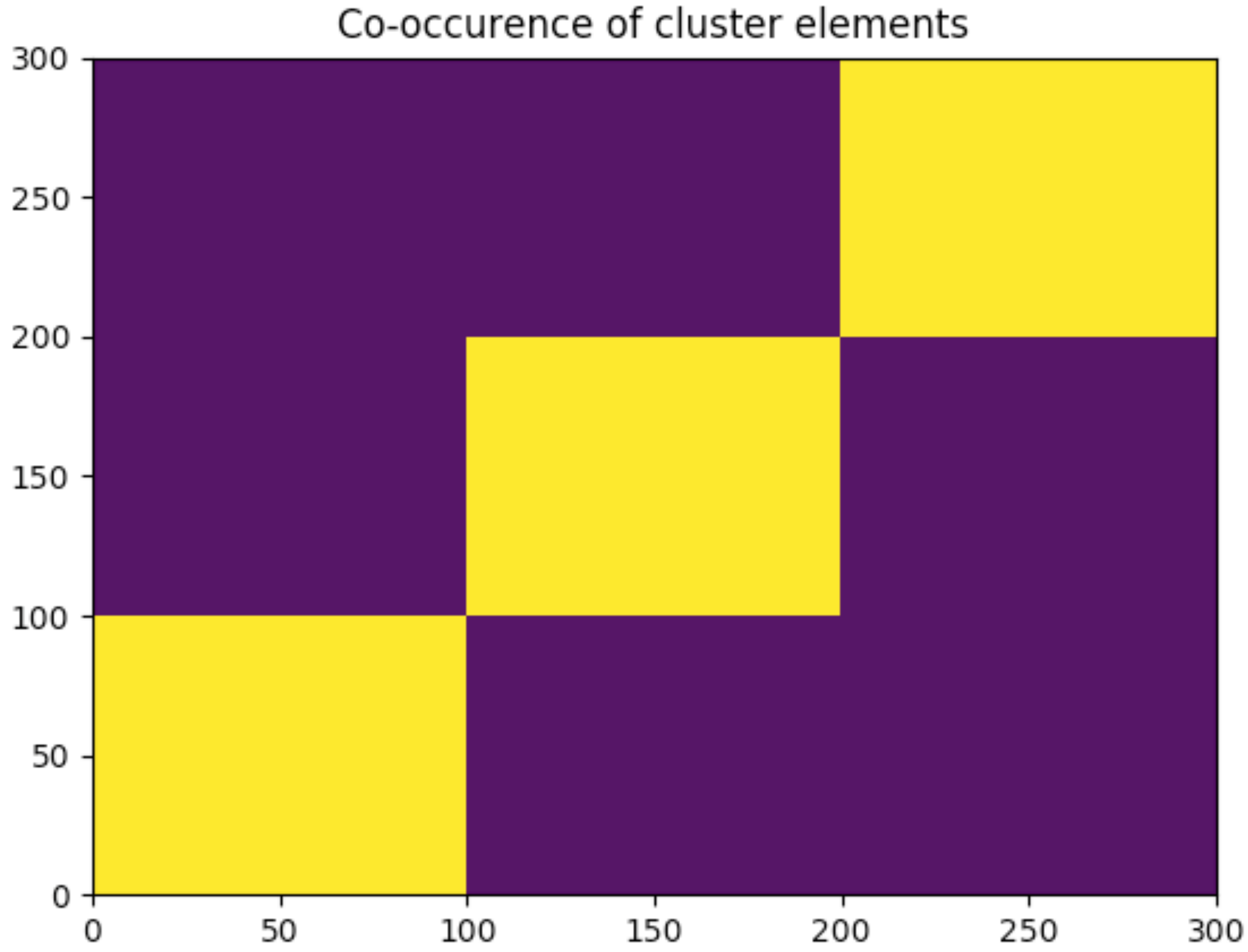


Co-occurrence

For all pairs (i,j) count how frequently i and j are in the same cluster.



Co-occurrence



Silhouette coefficient

a: The mean distance between a sample and all other points in the same class.

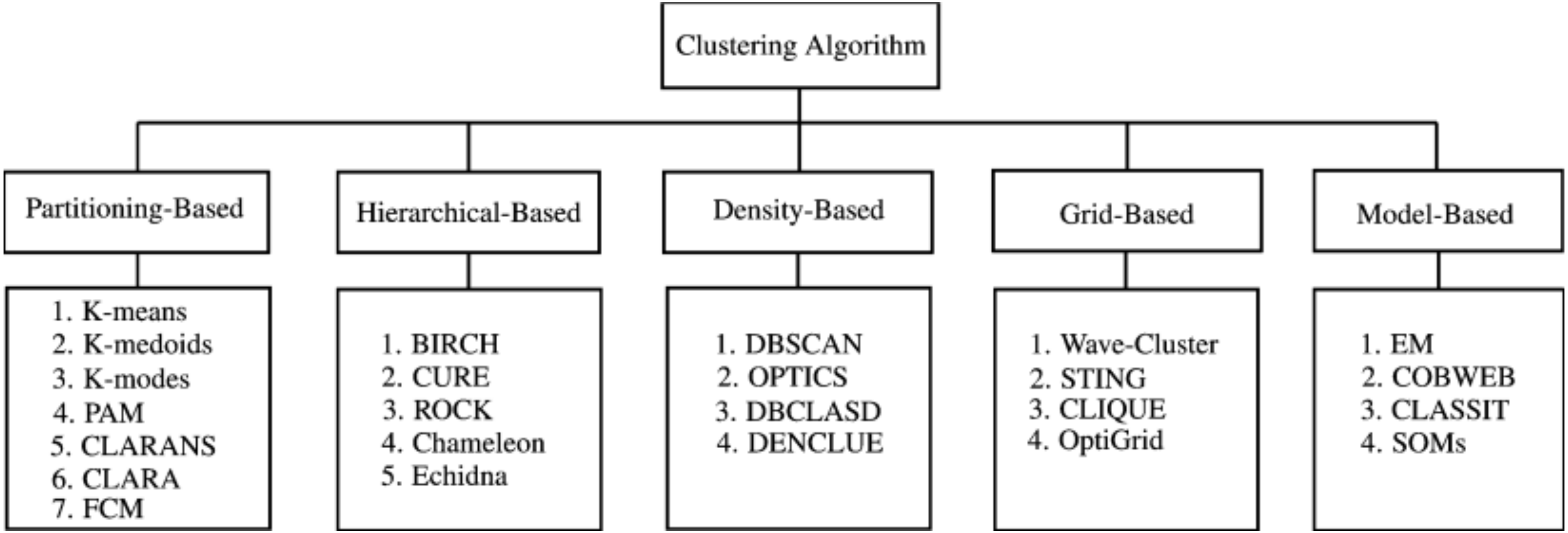
b: The mean distance between a sample and all other points in the *next nearest cluster*.

$$s = \frac{b - a}{\max(a, b)}$$

Ranges between -1 and 1. High value indicate good separation between clusters.

https://scikit-learn.org/stable/modules/clustering.html#silhouette-coefficient

Clustering clustering algorithms



Clustering Algorithm

Partitioning-Based

1. K-means
2. K-medoids
3. K-modes
4. PAM
5. CLARANS
6. CLARA
7. FCM

Hierarchical-Based

1. BIRCH
2. CURE
3. ROCK
4. Chameleon
5. Echidna

Density-Based

1. DBSCAN
2. OPTICS
3. DBCLASD
4. DENCLUE

Grid-Based

1. Wave-Cluster
2. STING
3. CLIQUE
4. OptiGrid

Model-Based

1. EM
2. COBWEB
3. CLASSIT
4. SOMs

Fahad et al. (2014) IIEEE Trans. Emerging Topics in Computing, volume 2, 267-279

Combining clustering and classification

Useful idea when labeling is expensive

- Take a dataset with handwritten digits
- Provide only one label per digit (10 labels for the whole dataset)
- Use 10-means with the ten labeled images as starting points for clustering the whole dataset.
- Then use 1nn for classifying new handwritten digits.

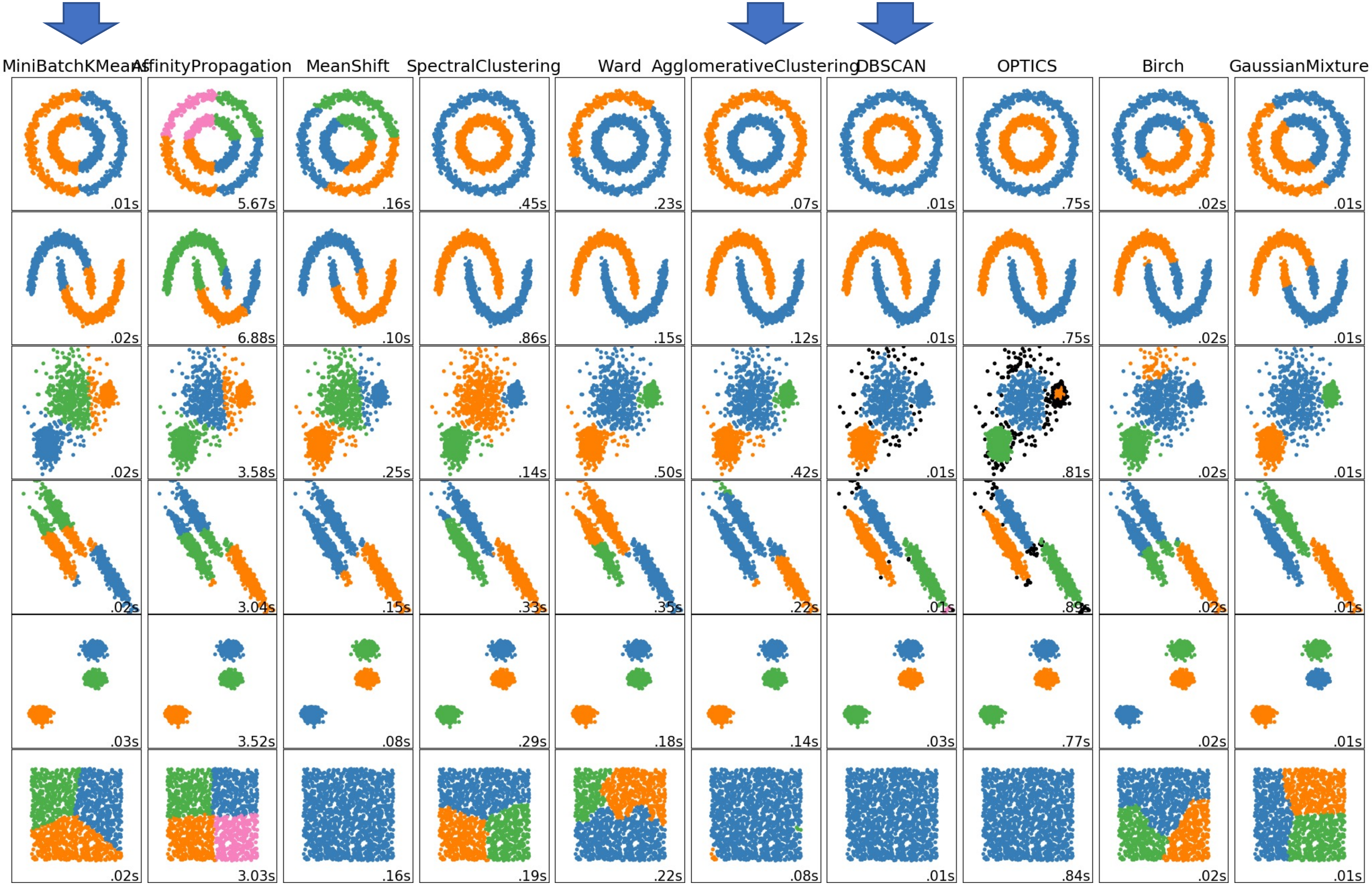
Reflections on clustering

Clustering is successful, but difficult

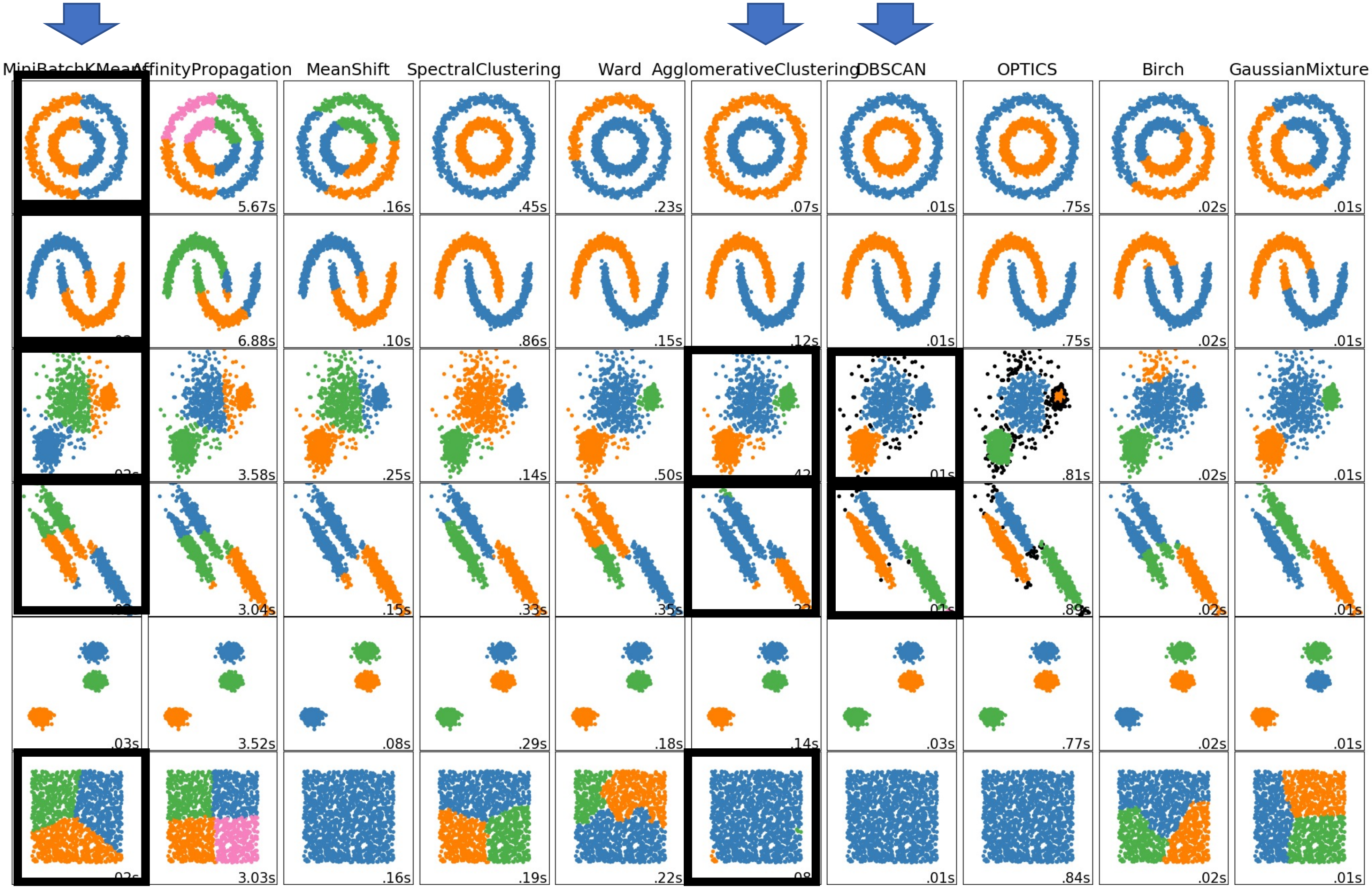
- Inherent vagueness in the definition of a cluster

- Can be difficult to define an appropriate similarity measure

Jain, A.K. (2010) Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, **31**, 651-666



https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

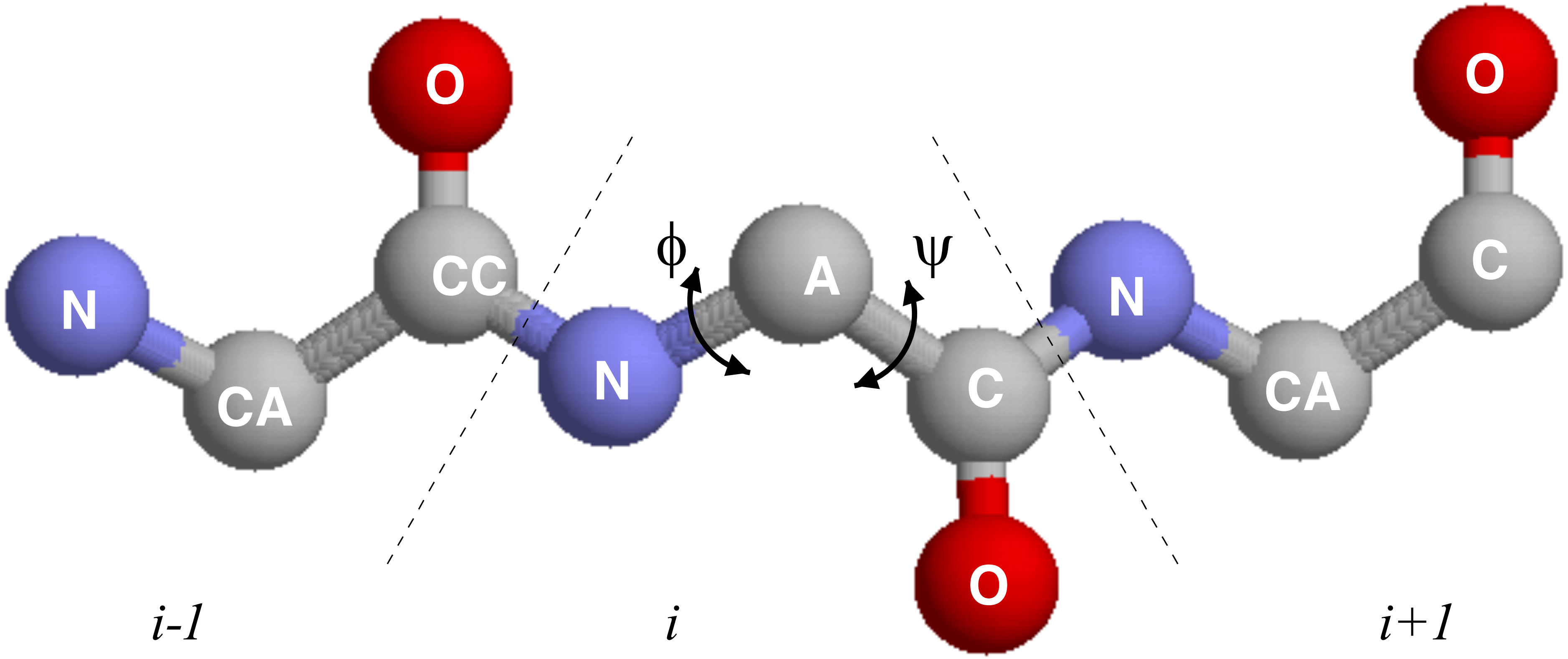


Are the framed cases as desired?

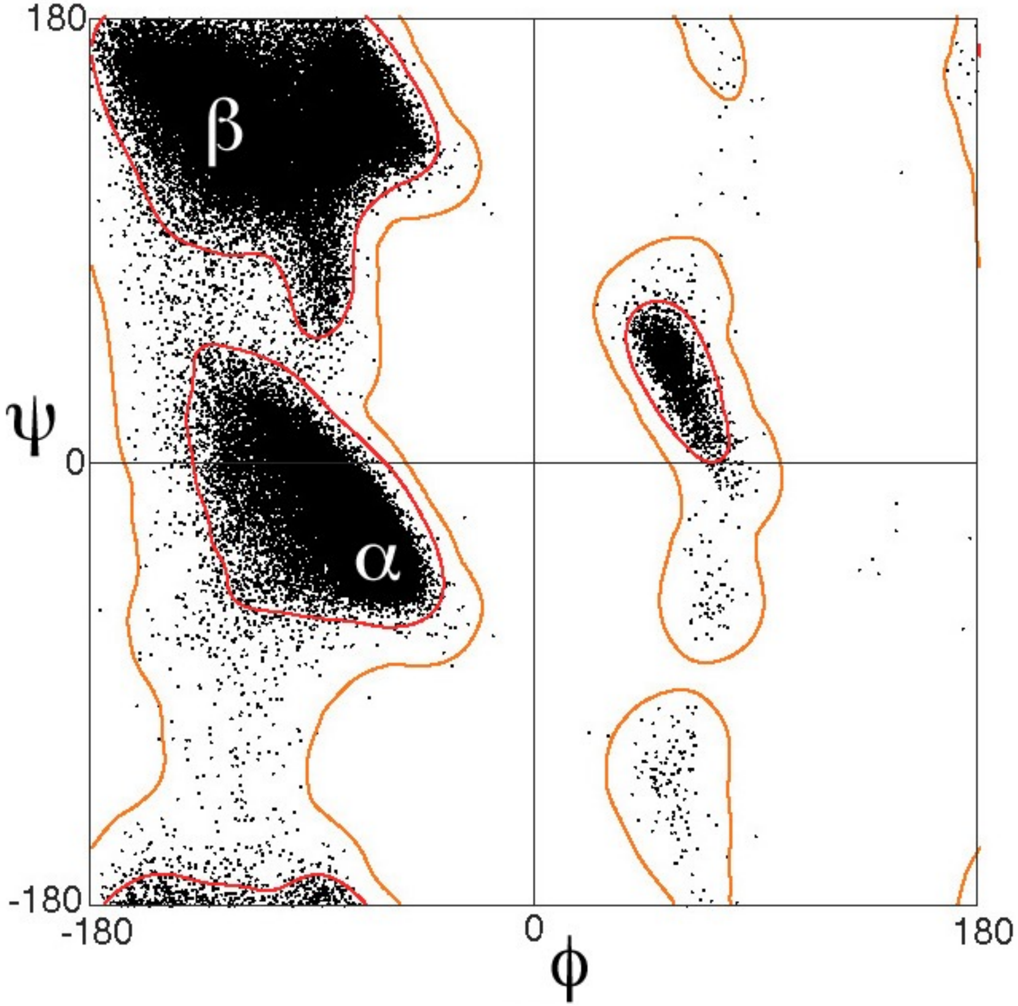
Assignment 3

- Using K-means and density-based clustering to cluster the main chain conformations of amino acid residues in proteins.
- If curious for more information on the problem domain, look at:
  - http://bioinformatics.org/molvis/phipsi/
  - http://tinyurl.com/RamachandranPrincipleYouTube

Protein main chain



Ramachandran plot



Around 100000 data points shown here

http://bioinformatics.org/molvis/phipsi/