Performing Clustering Using Multiple Techniques

Janani Ravi

CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Hierarchical clustering techniques

Agglomerative and BIRCH clustering

DBSCAN clustering

Mean-shift clustering

Affinity clustering

Spectral clustering

Mini-batch K-means clustering

Categories of Clustering Algorithms

Clustering Algorithms

Centroid-based	Hierarchical
Distribution-based	Density-based

Clustering Algorithms

Centroid-based	Hierarchical
Distribution-based	Density-based

Cluster represented by a central reference vector which may not be a part of the original data e.g. k-means clustering

Clustering Algorithms

Centroid-based

Hierarchical

Distribution-based

Density-based

Connectivity-based clustering based on the core idea that points are connected to points close by rather than further away

Clustering Algorithms

Centroid-based	**Hierarchical**
Distribution-based	Density-based

A cluster can be defined largely by the maximum distance needed to connect different parts of the cluster

Clustering Algorithms

Centroid-based	Hierarchical
Distribution-based	Density-based

Algorithms do not partition the dataset but instead construct a tree of points which are typically merged together

Clustering Algorithms

Centroid-based	**Hierarchical**
Distribution-based	Density-based

Agglomerative and BIRCH clustering

Clustering Algorithms

Centroid-based	Hierarchical
Distribution-based	Density-based

Built on statistical distribution models - objects of a cluster are the ones which belong most likely to the same distribution

Clustering Algorithms

Centroid-based	Hierarchical
Distribution-based	Density-based

Tend to be complex clustering models which might be prone to overfitting on data points

Clustering Algorithms

Centroid-based	Hierarchical
Distribution-based	Density-based

Gaussian mixture models

Clustering Algorithms

Centroid-based	Hierarchical
Distribution-based	**Density-based**

Create clusters from areas which have a higher density of data points

Clustering Algorithms

Centroid-based	Hierarchical
Distribution-based	**Density-based**

Objects in sparse areas, which separate clusters, are considered noise and border points

Clustering Algorithms

Centroid-based	Hierarchical
Distribution-based	**Density-based**

DBSCAN and mean-shift clustering

Demo

Setting up helper functions

Implementing k-means clustering using helper functions

Choosing Clustering Algorithms

Choosing Clustering Algorithms

Size of Dataset

	Small	Medium	Large
Many			
Moderate			
Few			

Number of Clusters

Choosing Clustering Algorithms

Size of Dataset

Number of Clusters	Small	Medium	Large
Many			
Moderate			
Few			

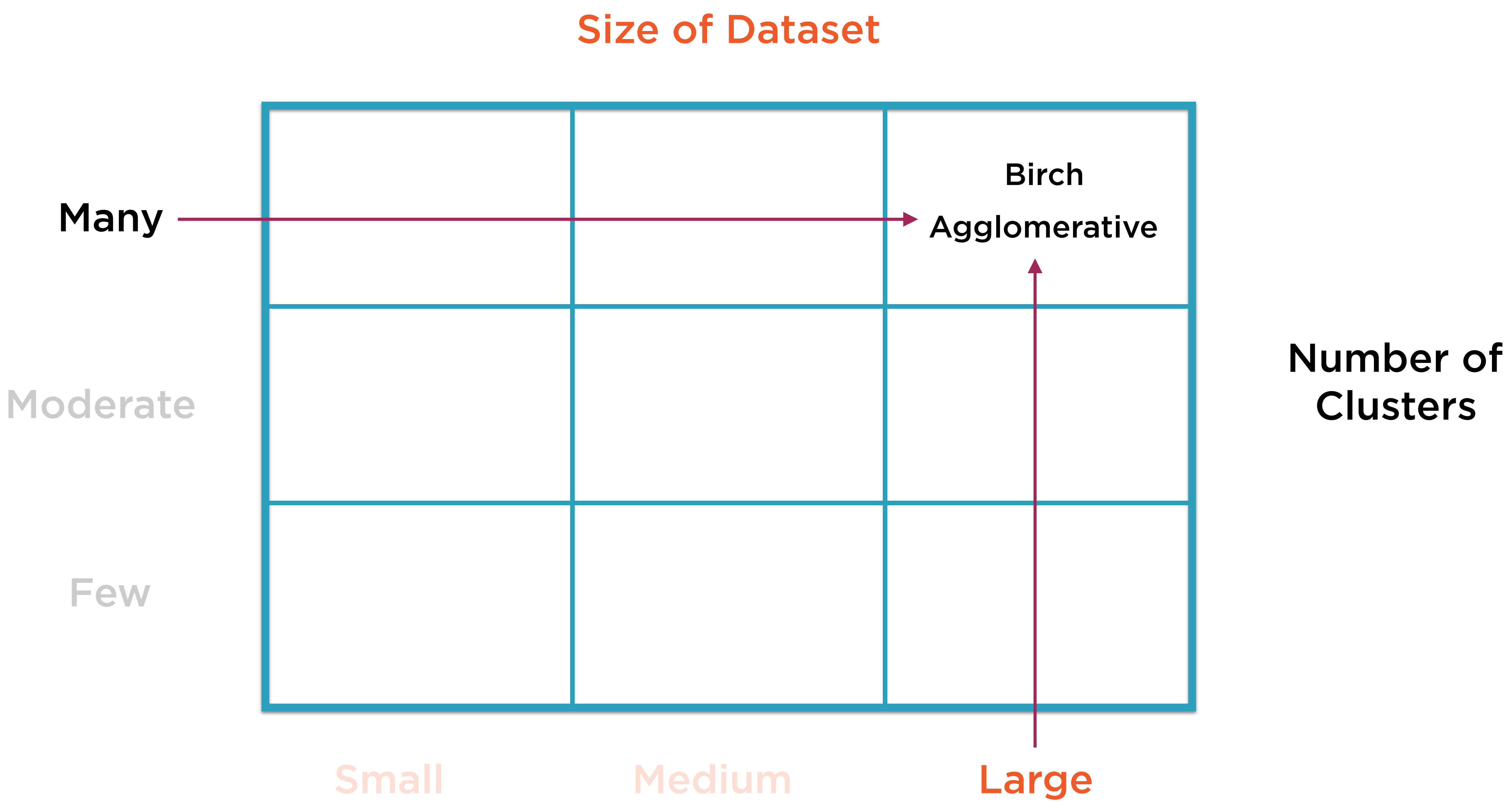
Choosing Clustering Algorithms

Size of Dataset

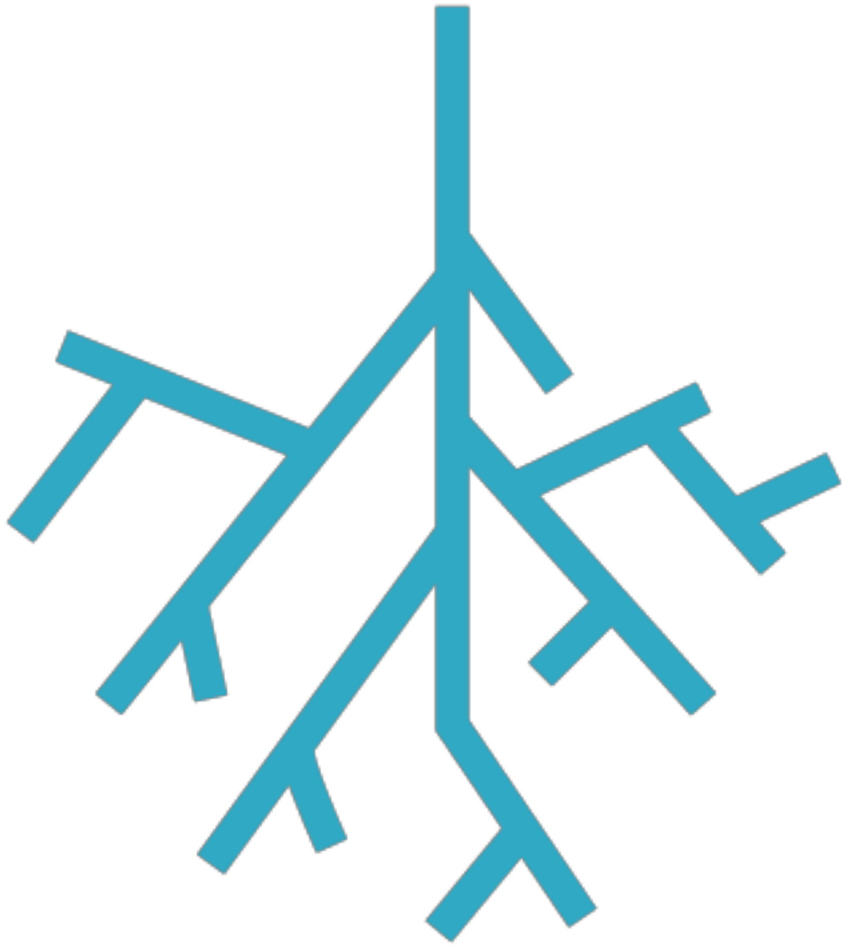
Number of Clusters

	Small	Medium	Large
Many			
Moderate			
Few			

Choosing Clustering Algorithms



BIRCH, Agglomerative Clustering

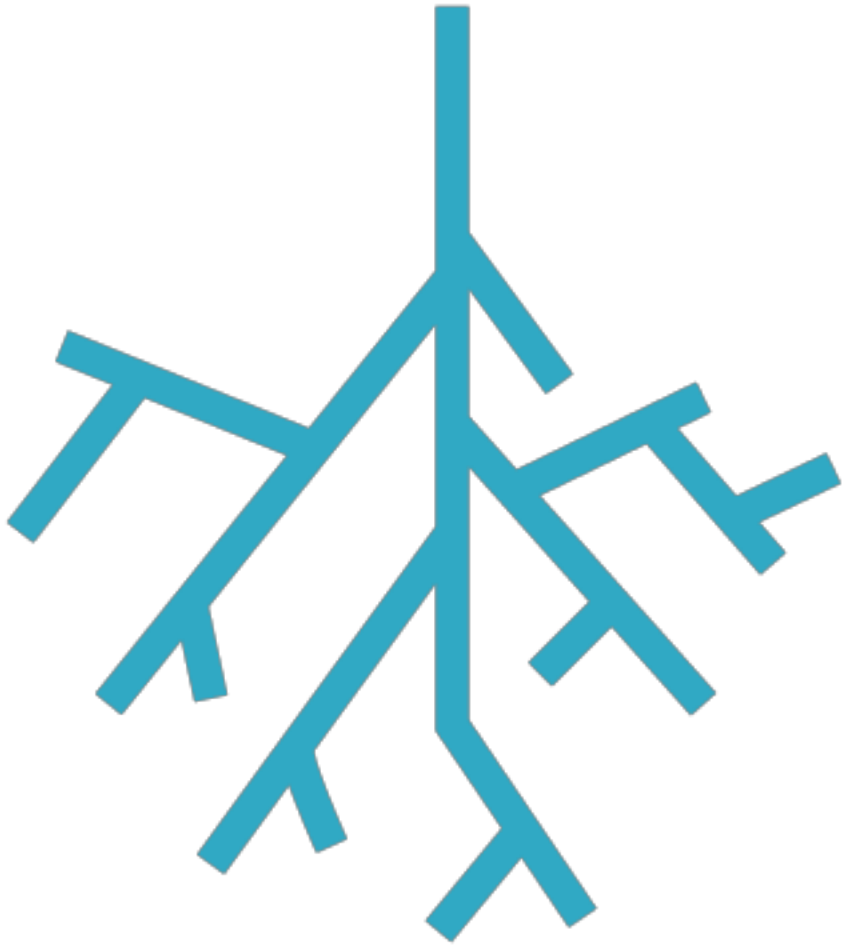


Hierarchical clustering algorithms

Build a tree representation of the data

Which may then be merged together into different numbers of clusters

BIRCH, Agglomerative Clustering



Large datasets, large number of clusters

Birch detects and removes outliers

Also incrementally processes incoming data and updates clusters

Agglomerative clustering works even in absence of Euclidean distance

Choosing Clustering Algorithms

Size of Dataset

Number of Clusters	Small	Medium	Large
Many	Mean-shift Affinity Propagation		Birch Agglomerative
Moderate			
Few			

Mean-shift, Affinity Propagation

Small datasets, large number of clusters

Both work well with uneven cluster sizes and manifold shapes

Mean-shift uses pairwise distances between points

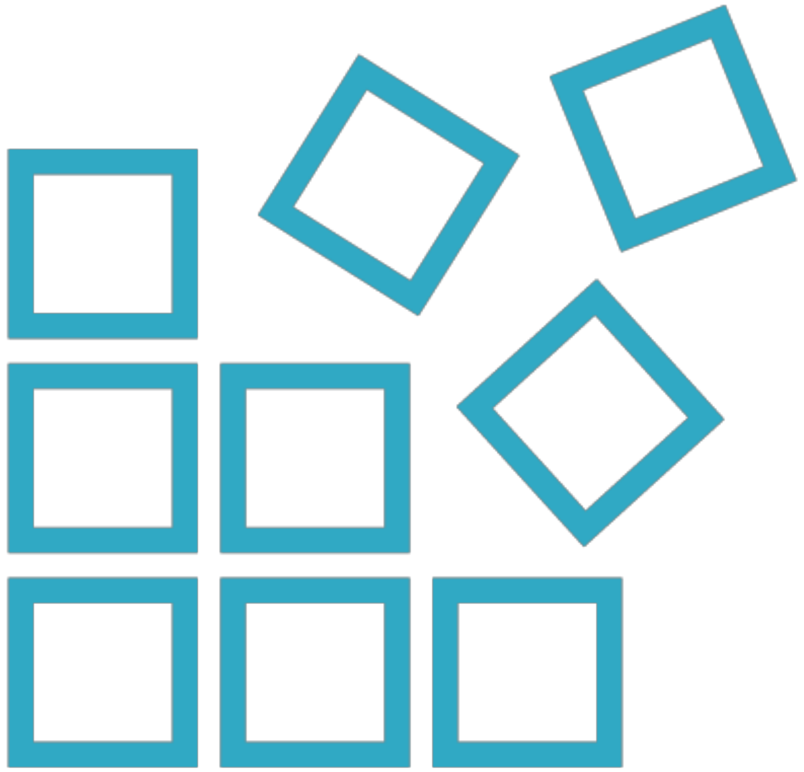
Affinity Propagation does not need number of clusters to be specified

Choosing Clustering Algorithms

Size of Dataset

Number of Clusters	Small	Medium	Large
Many	Mean-shift Affinity Propagation		Birch Agglomerative
Moderate			K-means DBSCAN
Few			

K-means, DBSCAN



Large datasets, moderate number of clusters

K-means for even cluster sizes and flat surfaces

Mini-batch K-means tweaks algorithm to be much faster, almost as good

DBSCAN for uneven cluster sizes and manifolds

Choosing Clustering Algorithms

Size of Dataset

Number of Clusters	Small	Medium	Large
Many	Mean-shift Affinity Propagation		Birch Agglomerative
Moderate			K-means DBSCAN
Few		Spectral	

Spectral Clustering

Small datasets, small number of clusters

Simple to implement

Intuitive results for data exploration

Even cluster sizes

Fine for manifolds

Relies on distances between points

Choosing Clustering Algorithms

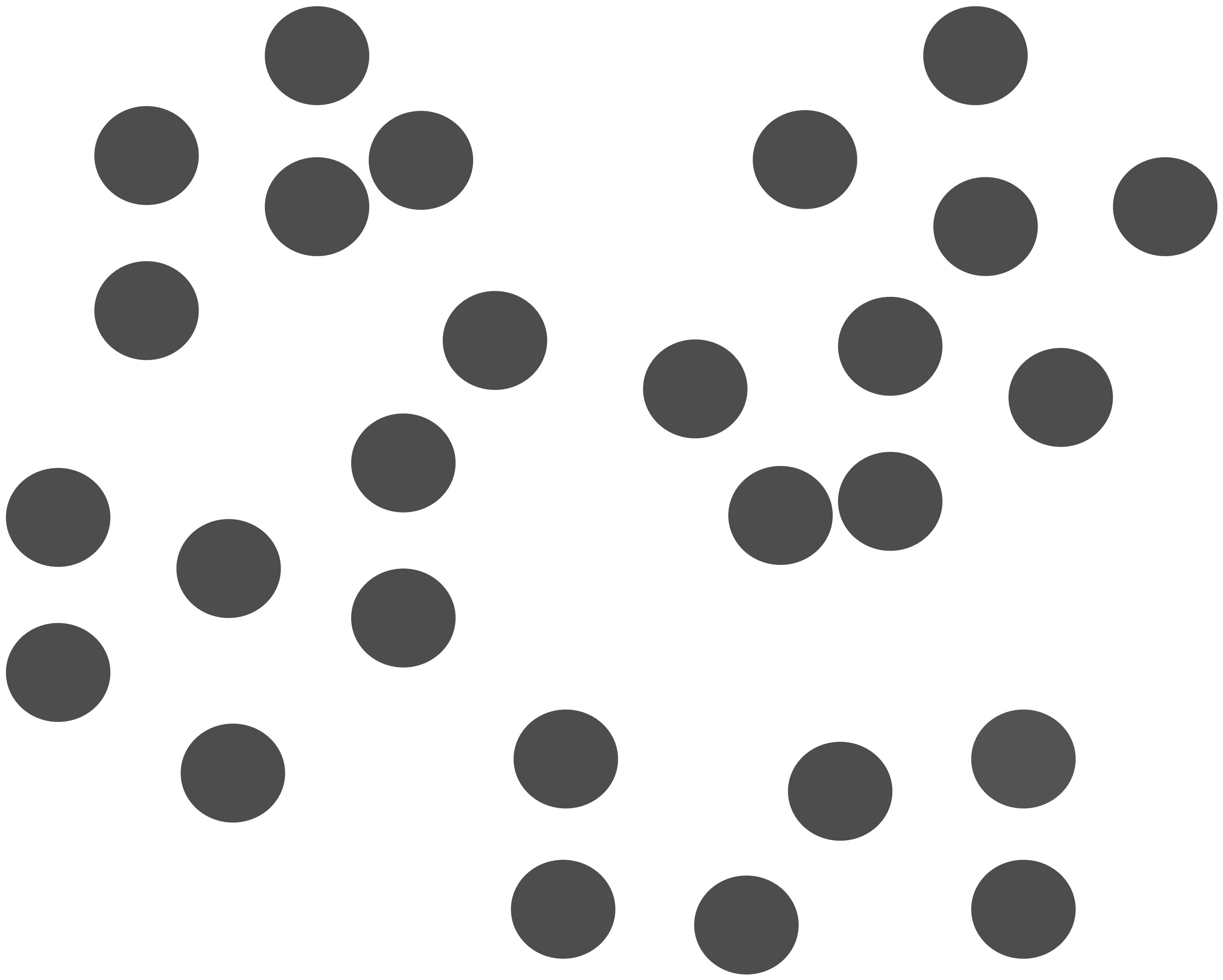
Size of Dataset

Number of Clusters	Small	Medium	Large
Many	Mean-shift Affinity Propagation		Birch Agglomerative
Moderate			K-means DBSCAN
Few		Spectral	

Hierarchical Clustering

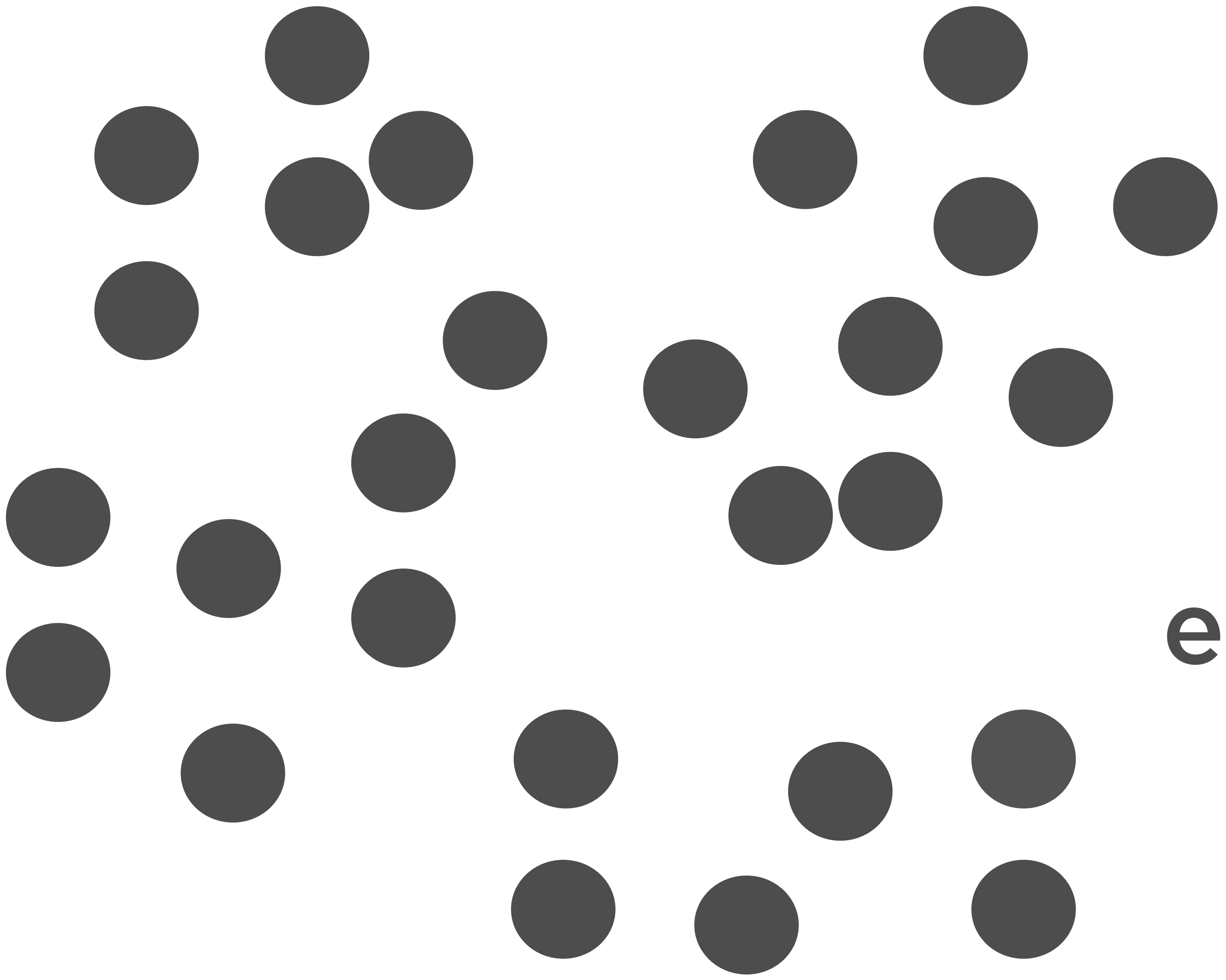
Hierarchical Clustering

Given t data points



Hierarchical Clustering

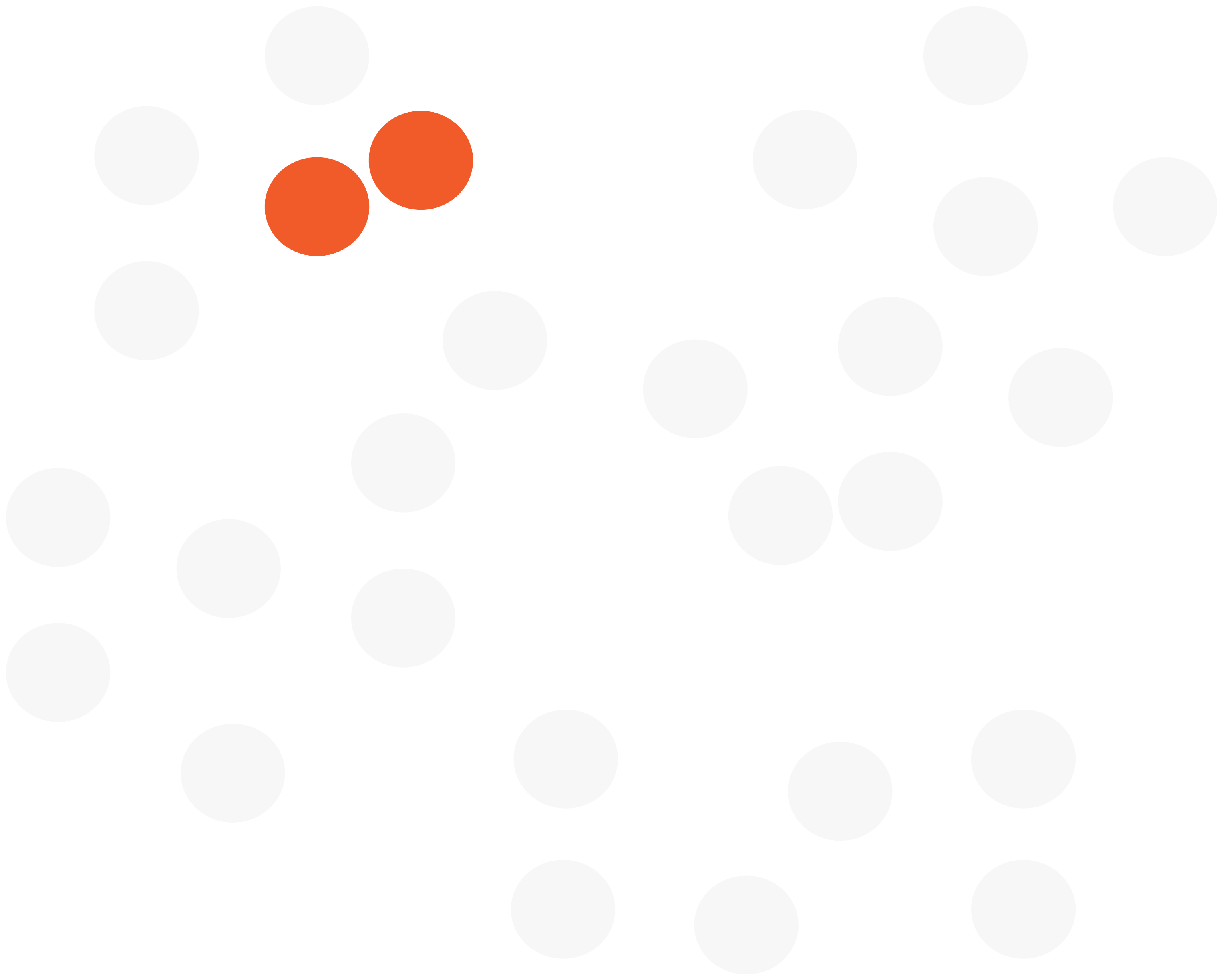
Start with t clusters, each with 1 point



t clusters, each of 1 point

Hierarchical Clustering

Merge the two clusters that are closest to each other



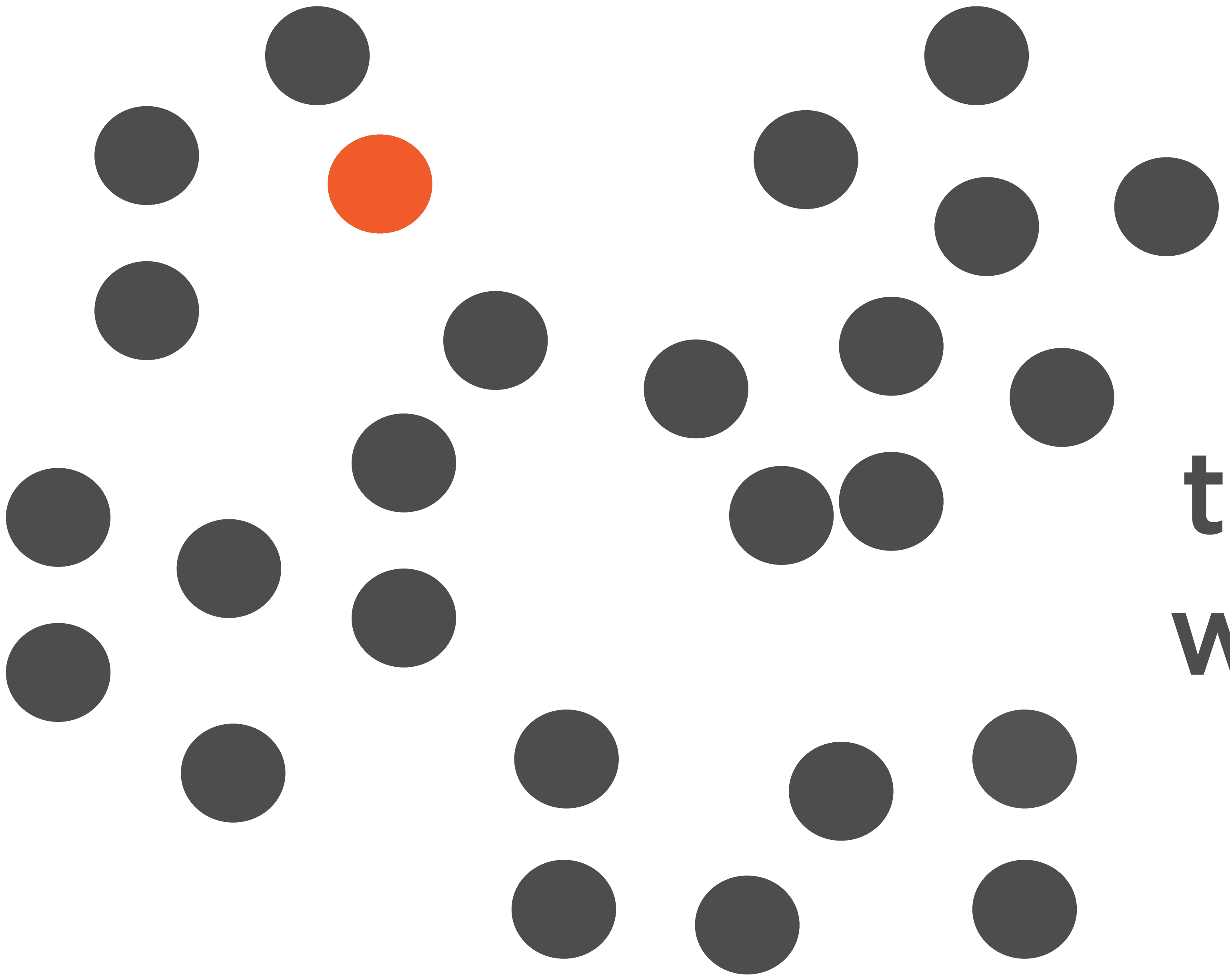
Hierarchical Clustering

Merge the two clusters that are closest to each other

t-1 clusters, 1 with 2 points

Hierarchical Clustering

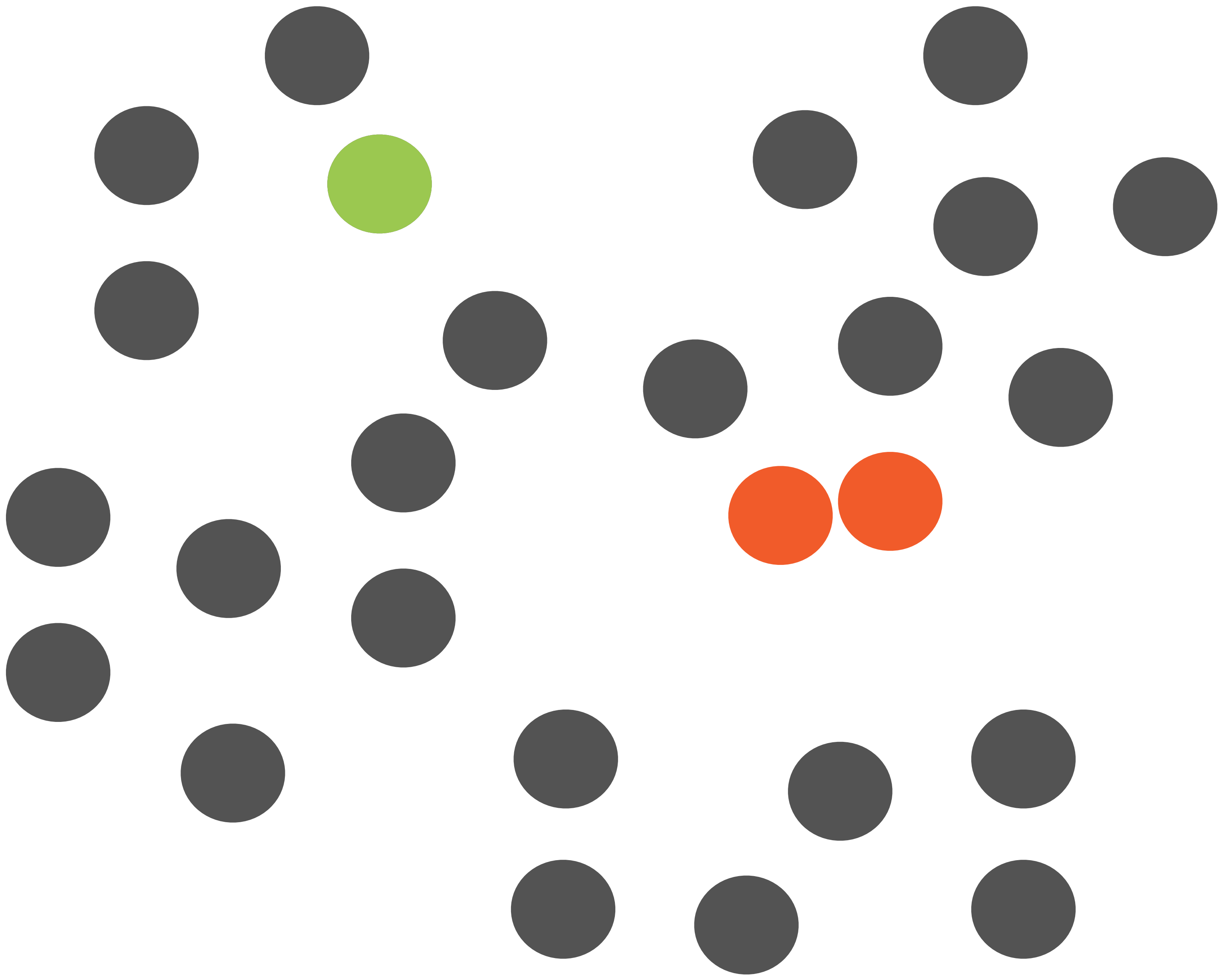
Rinse-and-repeat



t-1 clusters, 1 with 2 points

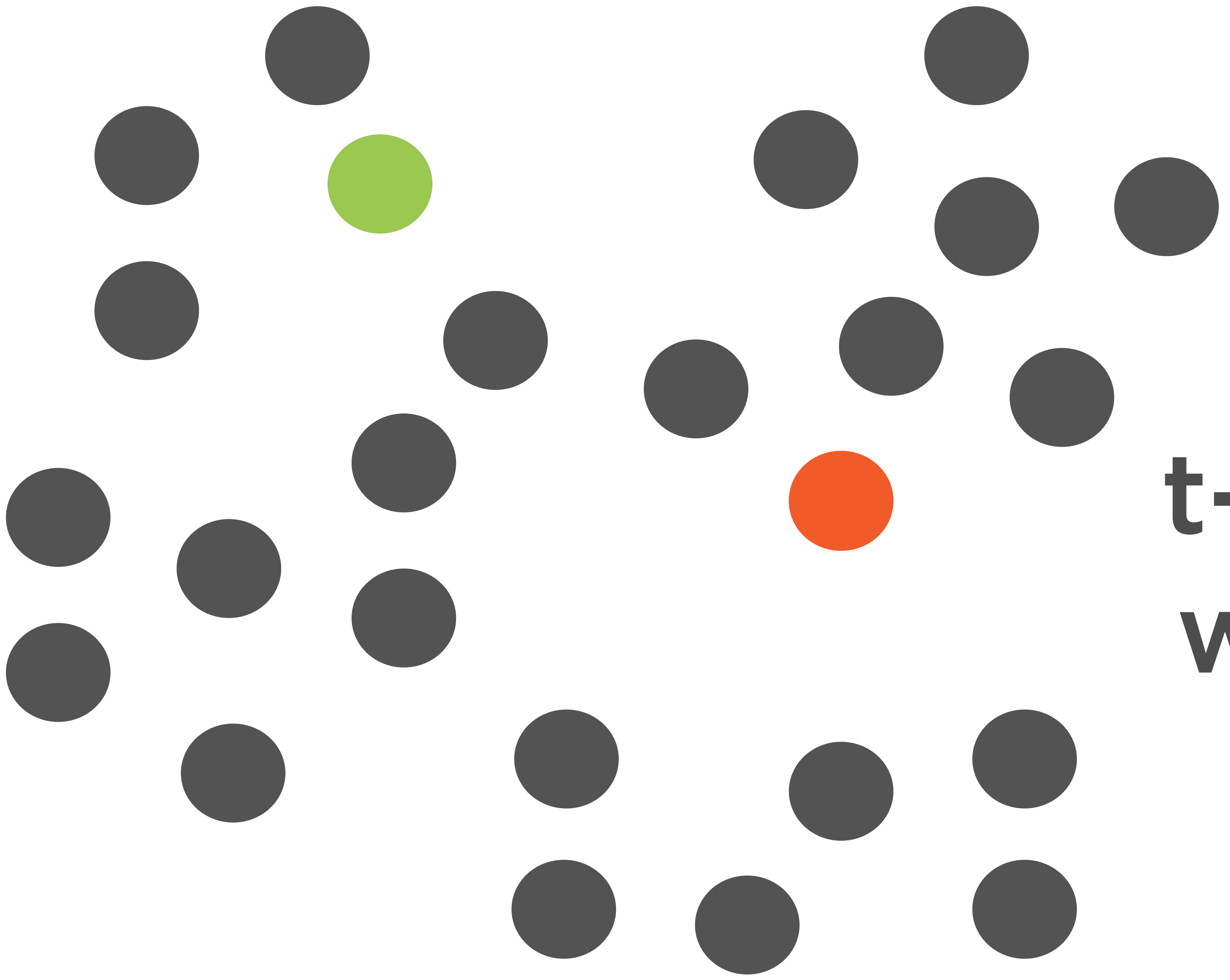
Hierarchical Clustering

Rinse-and-repeat



Hierarchical Clustering

Rinse-and-repeat



t-2 clusters, 2 with 2 points

Hierarchical Clustering

Rinse-and-repeat

6 clusters, each with multiple points

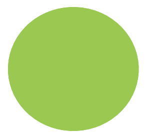
Hierarchical Clustering

The number of clusters keeps reducing

2 clusters, each with multiple points

Hierarchical Clustering

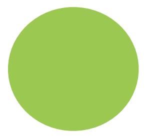
The number of clusters keeps reducing



1 cluster, with all t points

Hierarchical Clustering

Until just 1 cluster remains

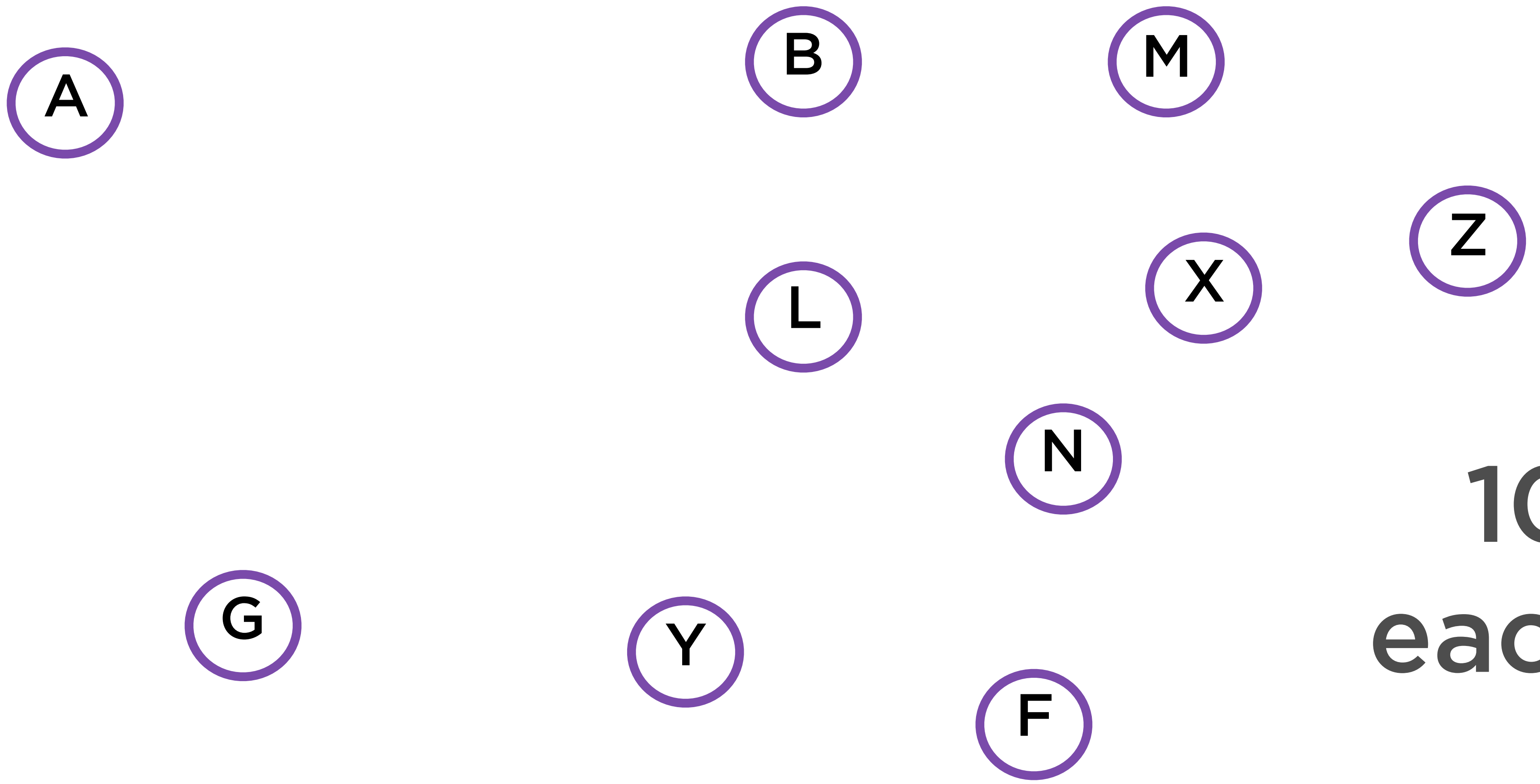


1 cluster, with all t points

Dendrogram

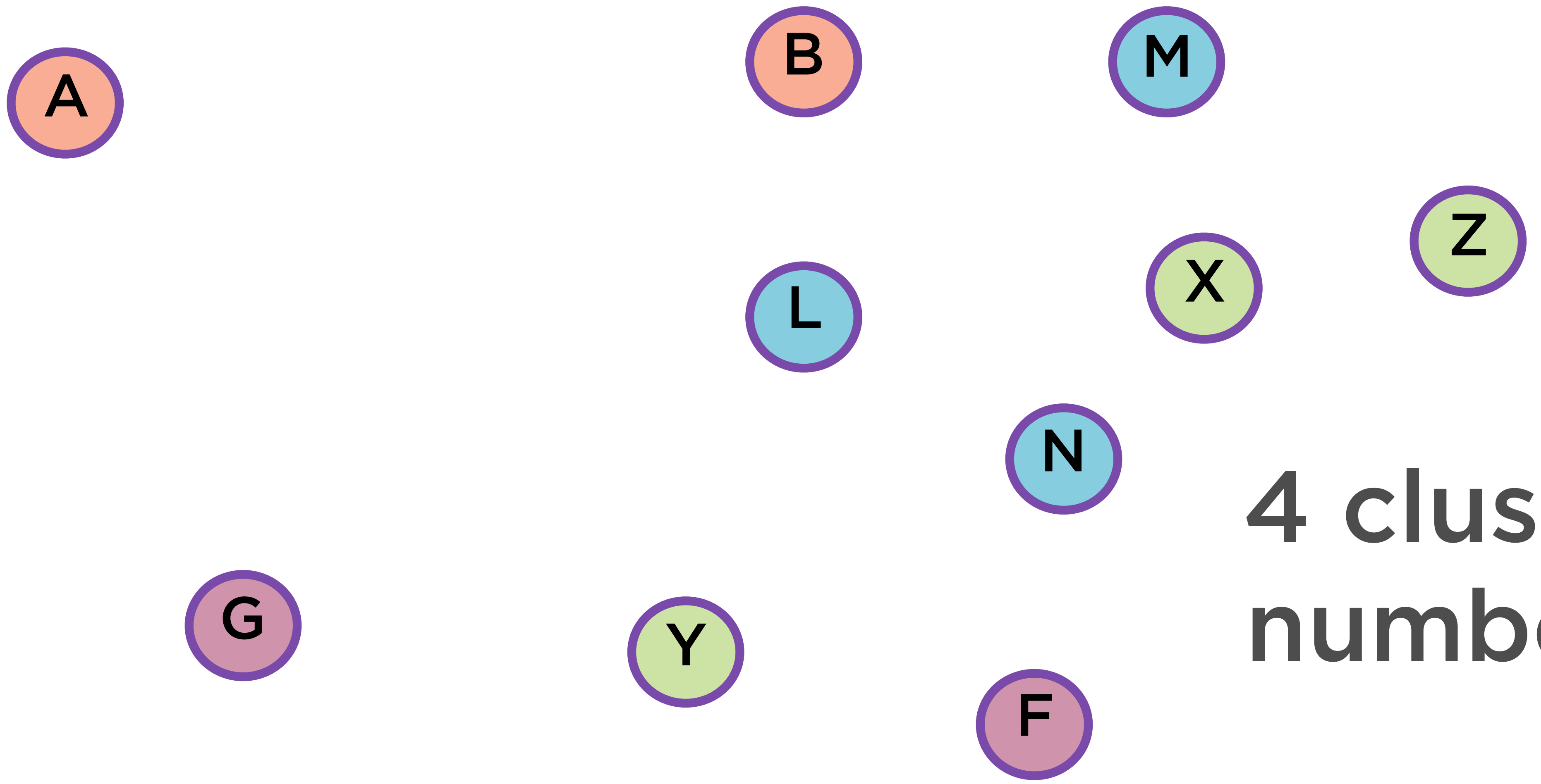
A tree diagram used to illustrate the arrangement of the clusters produced by hierarchical clustering

Dendrogram



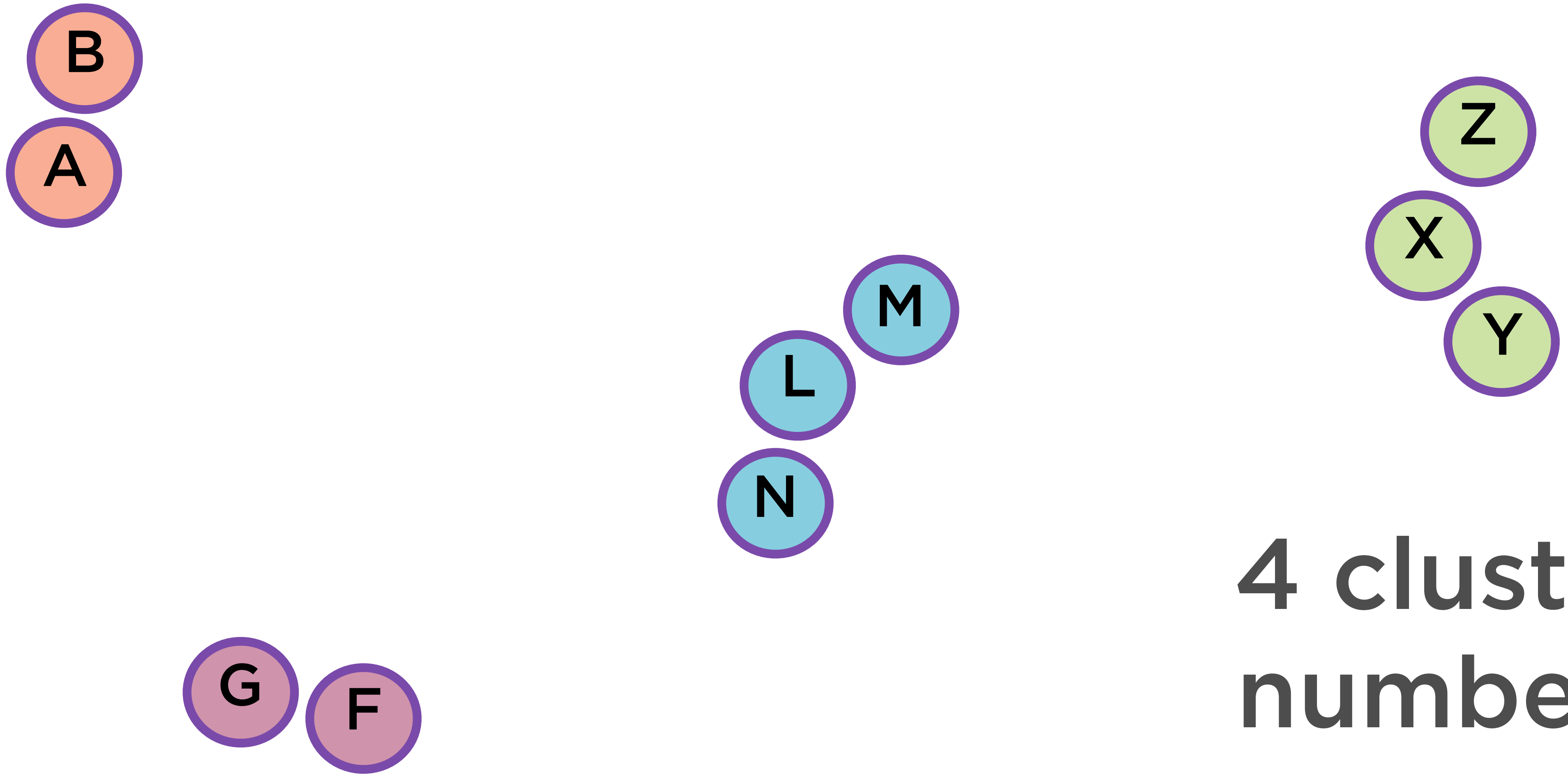
10 clusters, each of 1 point

Dendrogram



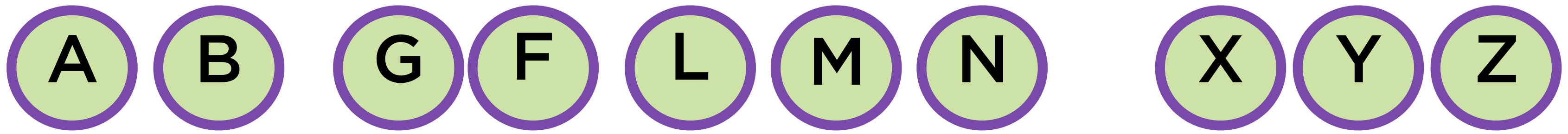
4 clusters, varying numbers of points

Dendrogram



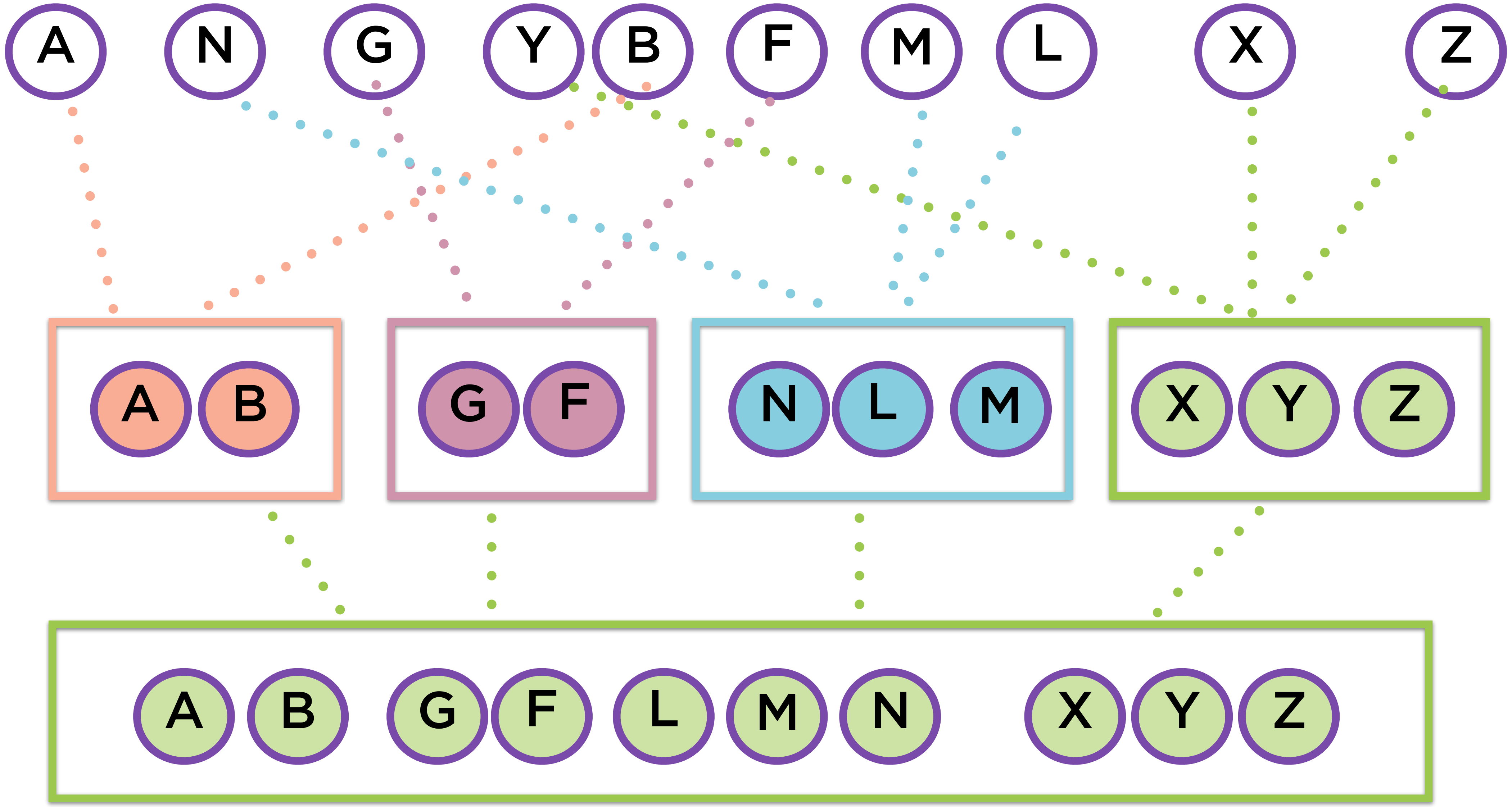
4 clusters, varying numbers of points

Dendrogram



1 clusters, all 10 points

Dendrogram

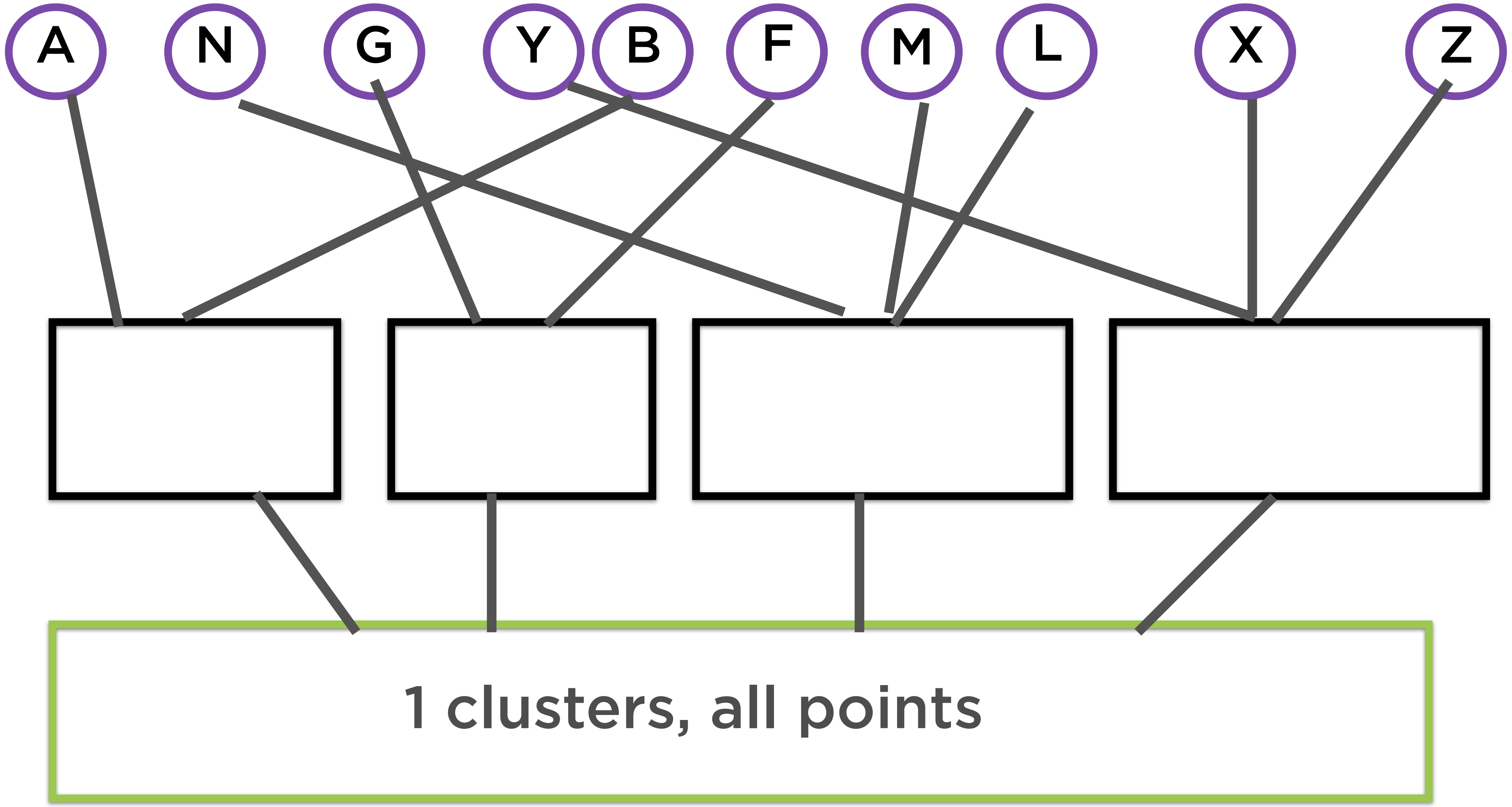


10 clusters, each of 1 point

4 clusters, varying numbers of points

1 clusters, all 10 points

Dendrogram



10 clusters, each of 1 point

Now, easy to vary number of clusters

Hierarchical Clustering

Agglomerative - start with many 1-point clusters, end with 1 big cluster

Divisive - start with 1 big cluster, end with many 1-point clusters

Contrasting Clustering Algorithms

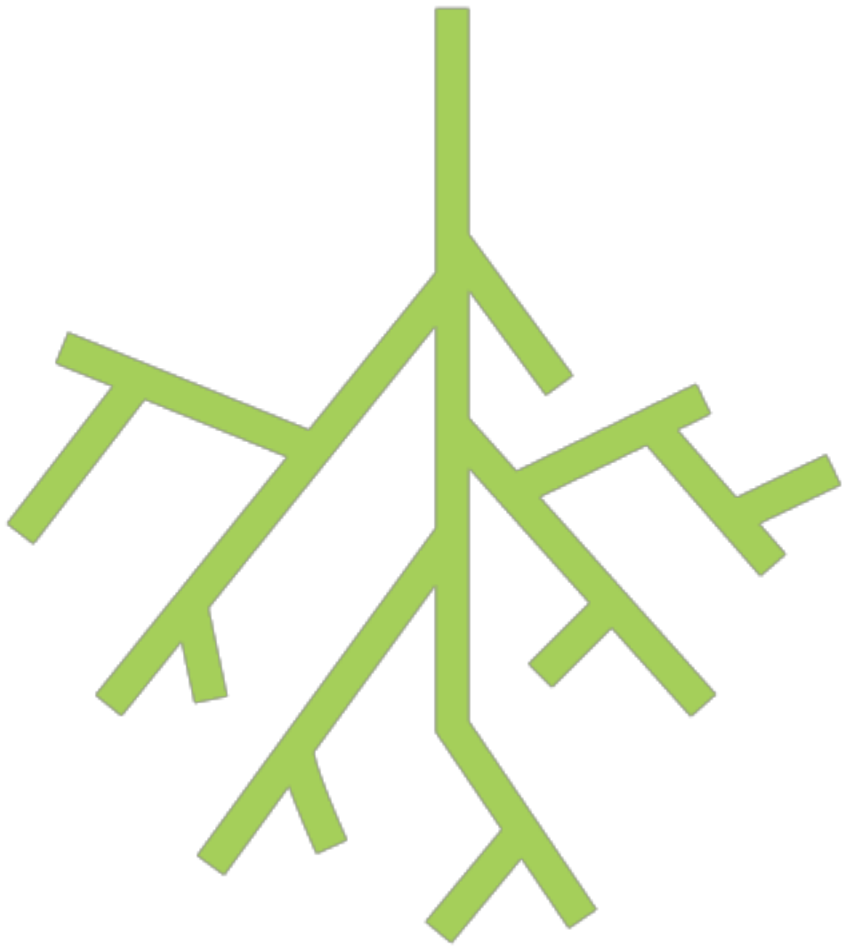
K-Means	Hierarchical
Need distance measure as well as way to aggregate points in a cluster	Only need distance measure; do not need way to combine points in cluster
Must represent data as vectors in N-dimensional hyperspace	No need to express data as vectors in N-dimensional hyperspace
Data representation can be difficult for complex data types	Relatively simple to represent even complex data e.g. graphs, documents
Variants (e.g. BFR) can efficiently deal with very large datasets on disk	Even with careful construction too computationally expensive for large datasets on disk

Demo

Implementing agglomerative clustering

Agglomerative Clustering:
Bottom-up hierarchical clustering

Choosing Clusters to Merge



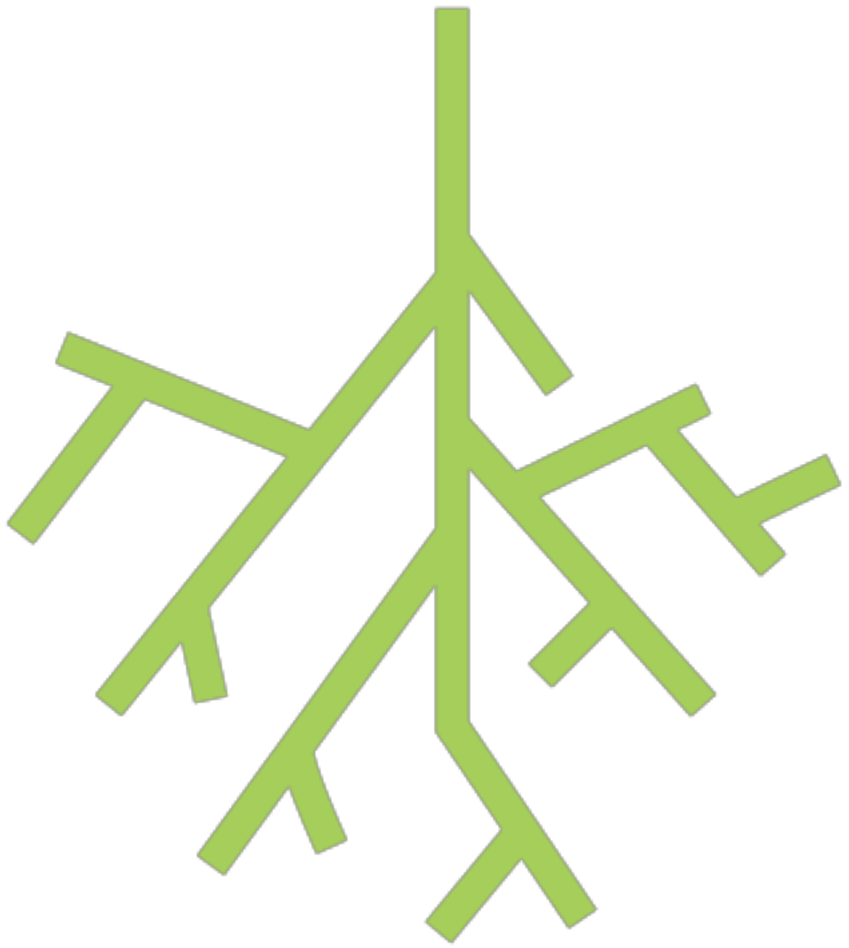
Each step of agglomerative clustering merges the two clusters nearest to each other

What is the metric for nearness?

How is nearness measured?

Several different approaches possible

Choosing Clusters to Merge



Each step of agglomerative clustering merges the two clusters nearest to each other

What is the metric for nearness?

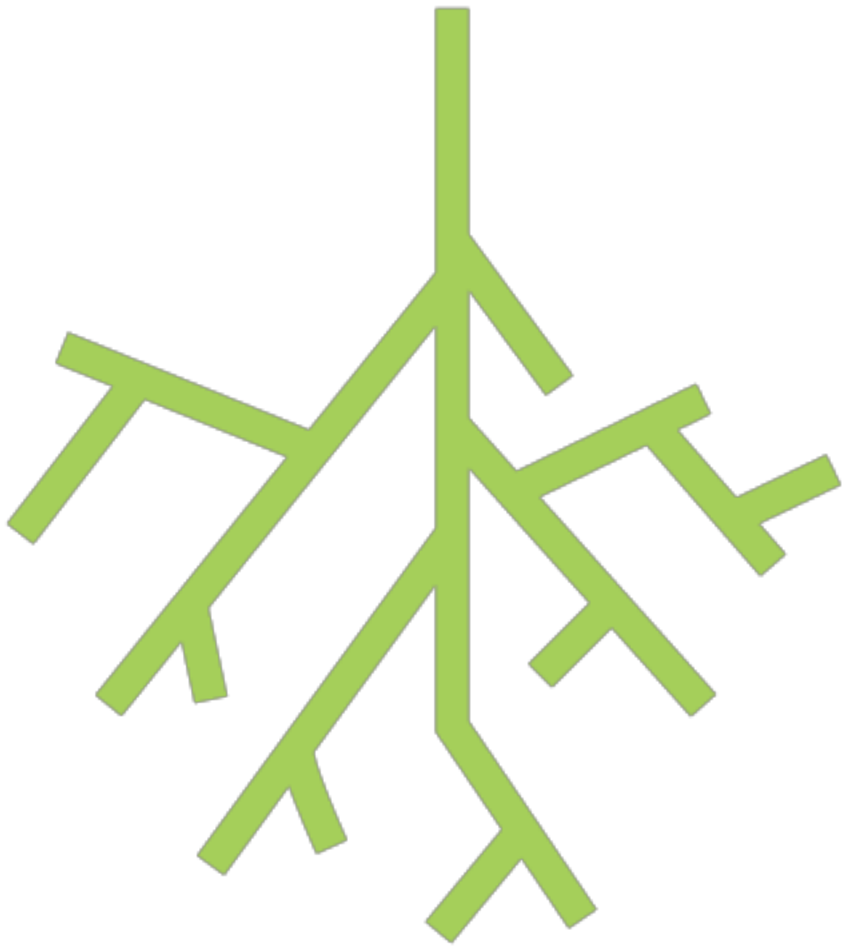
How is nearness measured?

Several different approaches possible

Nearness Metric or Distance Measure

Euclidean	L1
Cosine	Precomputed

Choosing Clusters to Merge



Each step of agglomerative clustering merges the two clusters nearest to each other

What is the metric for nearness?

How is nearness measured?

Several different approaches possible

Linkage criterion determines the **distance to be minimized** when merging clusters

Linkage Criterion

Single

Complete

Average

Ward

Linkage Criterion

Single	Complete
Average	Ward

Minimum of the distances between all points in the two clusters

Linkage Criterion

Single	**Complete**
Average	Ward

Maximum of the distances between all points in the two clusters

Linkage Criterion

Single	Complete
Average	Ward

Average distance between points in clusters

Linkage Criterion

Single	Complete
Average	Ward

Minimizes the variances of the data points in the two clusters

Demo

Implementing DBSCAN clustering

Choosing Clustering Algorithms

Size of Dataset

Number of Clusters	Small	Medium	Large
Many	Mean-shift Affinity Propagation		Birch Agglomerative
Moderate			K-means DBSCAN
Few			

Large Datasets, Moderate Cluster Count

Consider K-means and DBSCAN

K-means for even cluster sizes and flat surfaces

DBSCAN for uneven cluster sizes and manifolds

DBSCAN

Density-based Spatial Clustering of Applications with Noise

Density-based clustering groups together closely packed points

Points with few near neighbors are marked as outliers

Not as good as BIRCH at dealing with noise and outliers

Two Parameters for DBSCAN

eps

Minimum distance, points closer than this are neighbors

min_samples

Minimum number of points to form a dense region

eps

Minimum distance, points closer than this are neighbors

If too small most of the data will not be clustered

Unclustered points will be considered to be outliers

If too large clustering will be too coarse

Most of the points will be in the same cluster min_samples

Minimum number of points to form a dense region

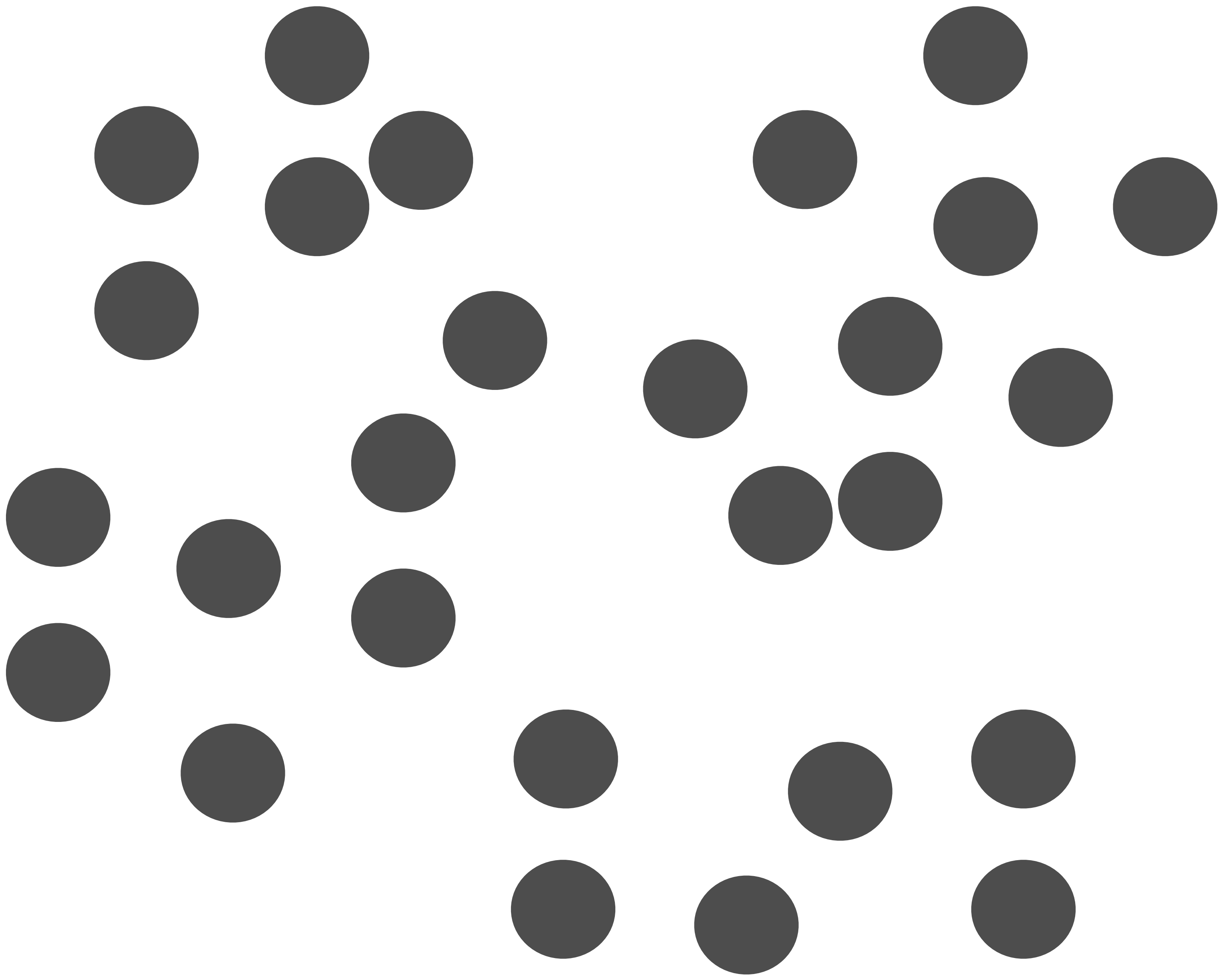
Generally this should be greater than number of dimensions in the data

Large values better for noisy data points, will form significant clusters

Mean-shift Clustering

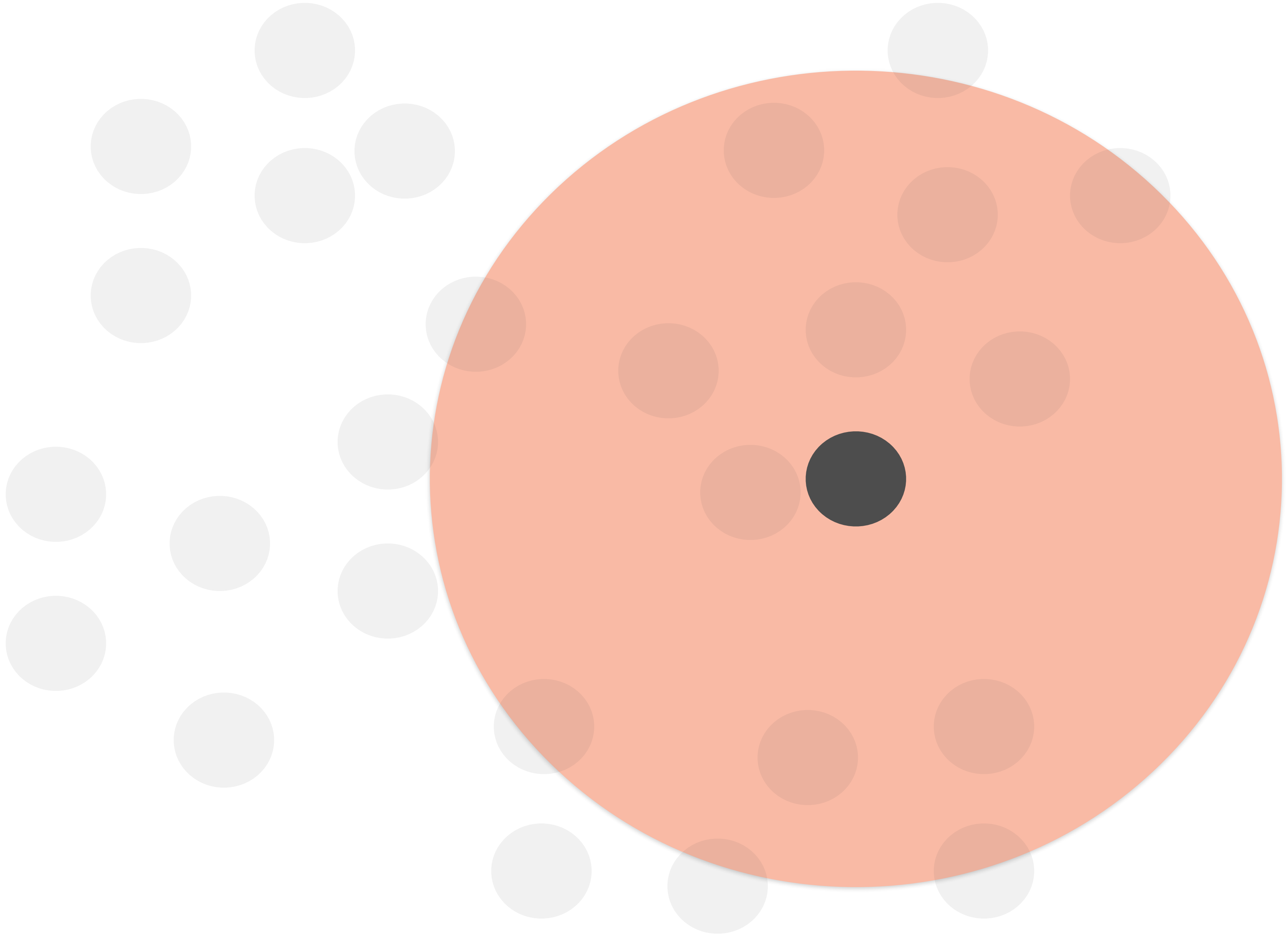
Mean Shift Clustering

Start with a set of points in space



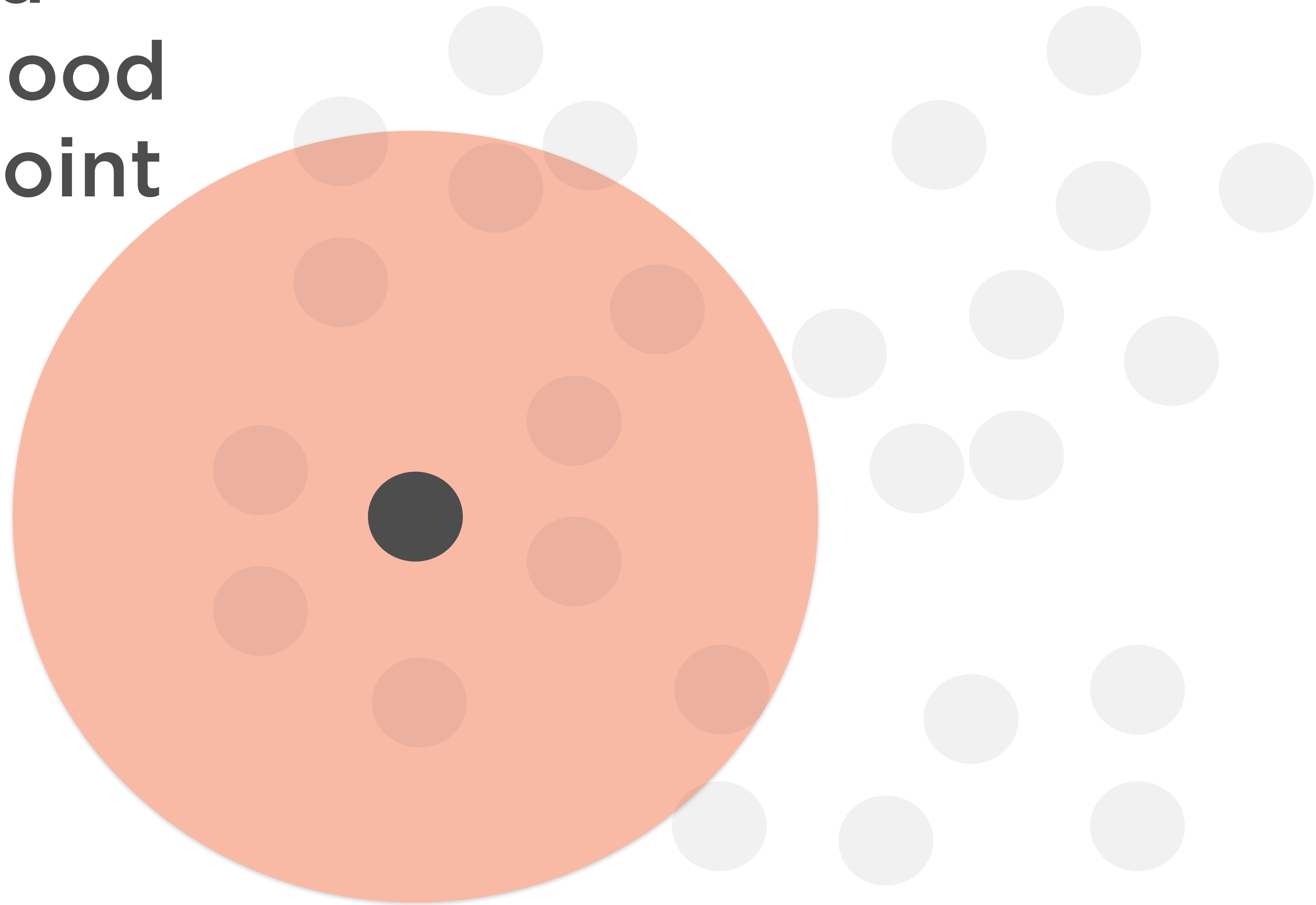
Mean Shift Clustering

Define a neighborhood for each point



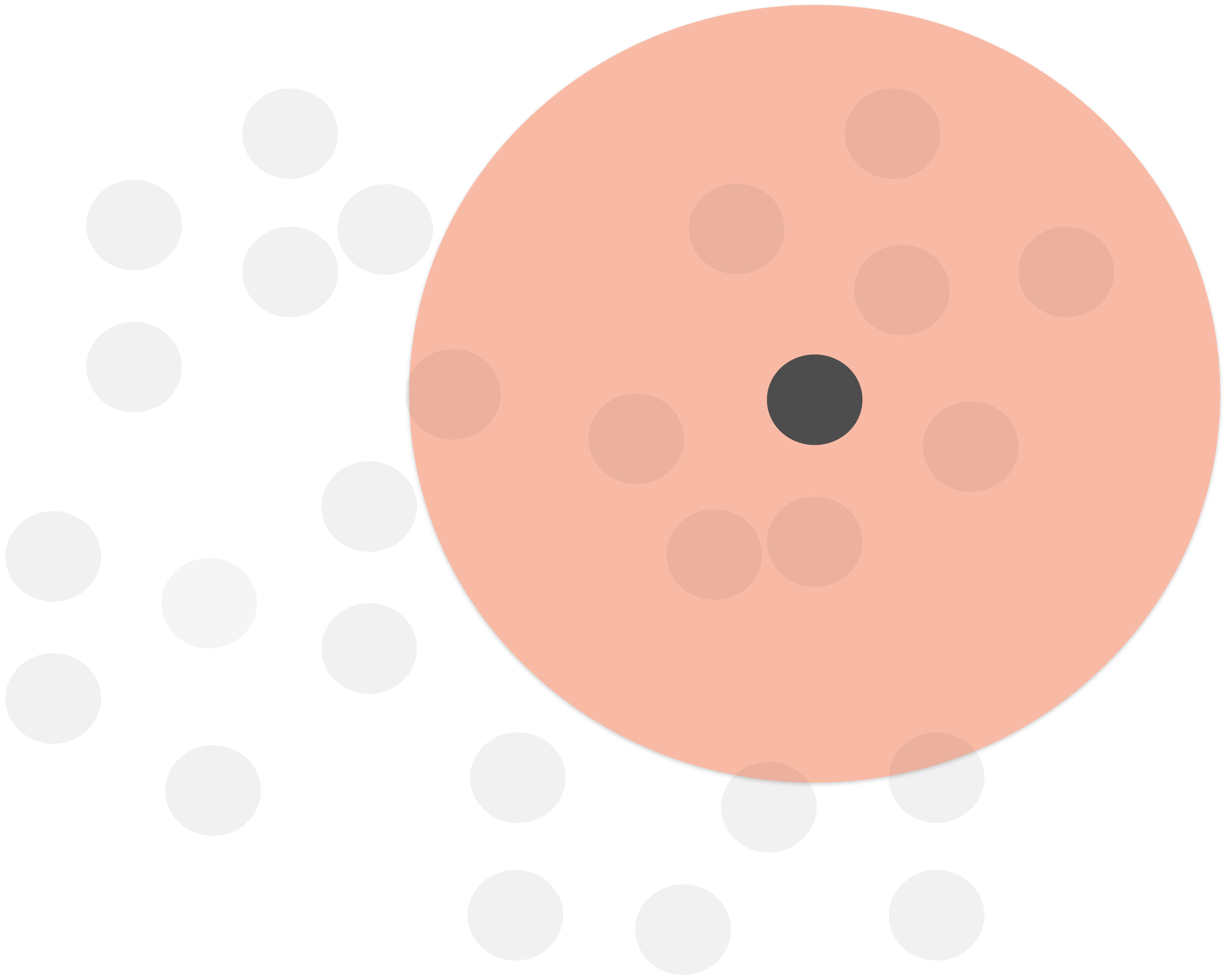
Mean Shift Clustering

Define a neighborhood for each point



Mean Shift Clustering

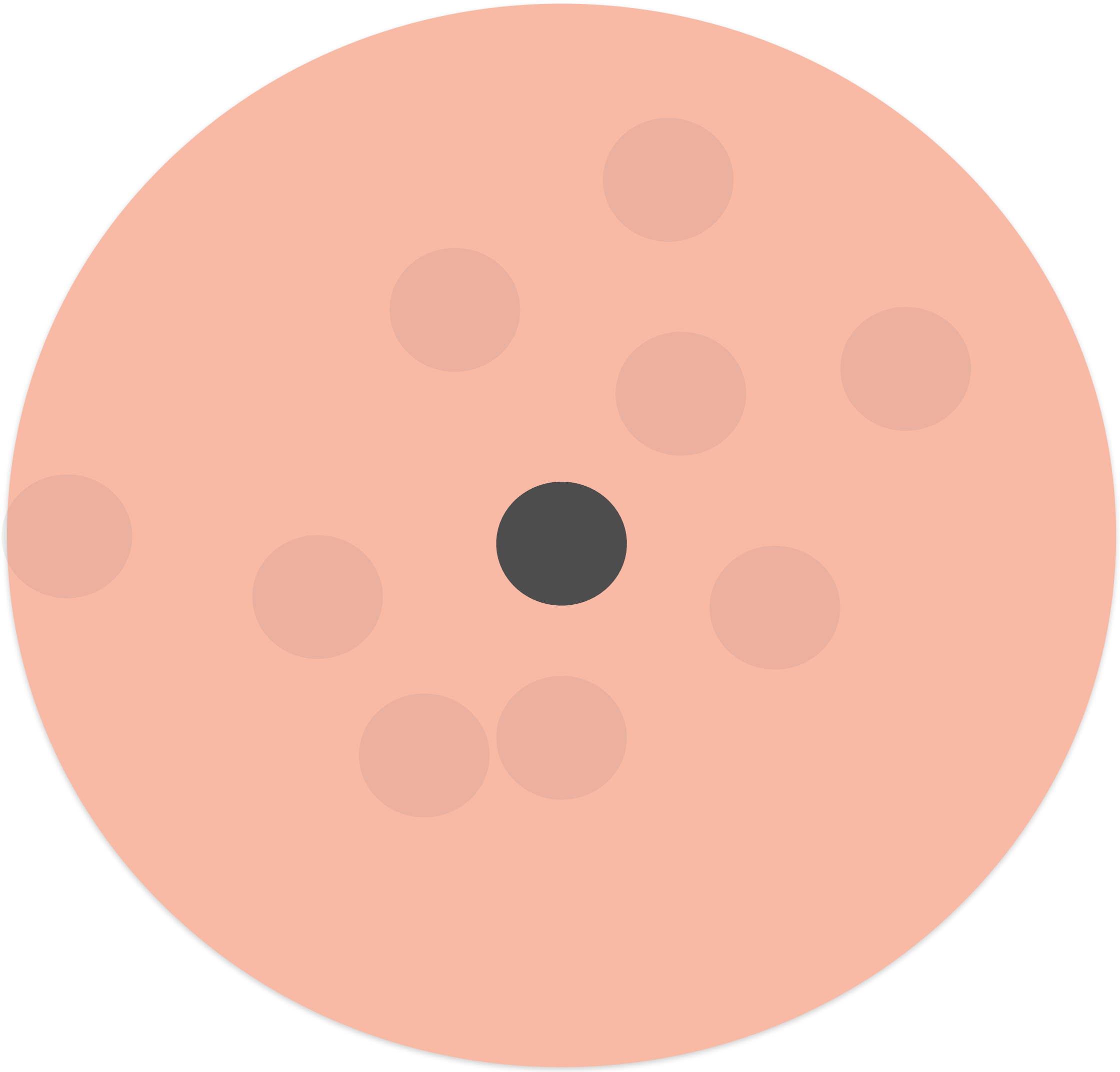
Define a neighborhood for each point



Mean Shift Clustering

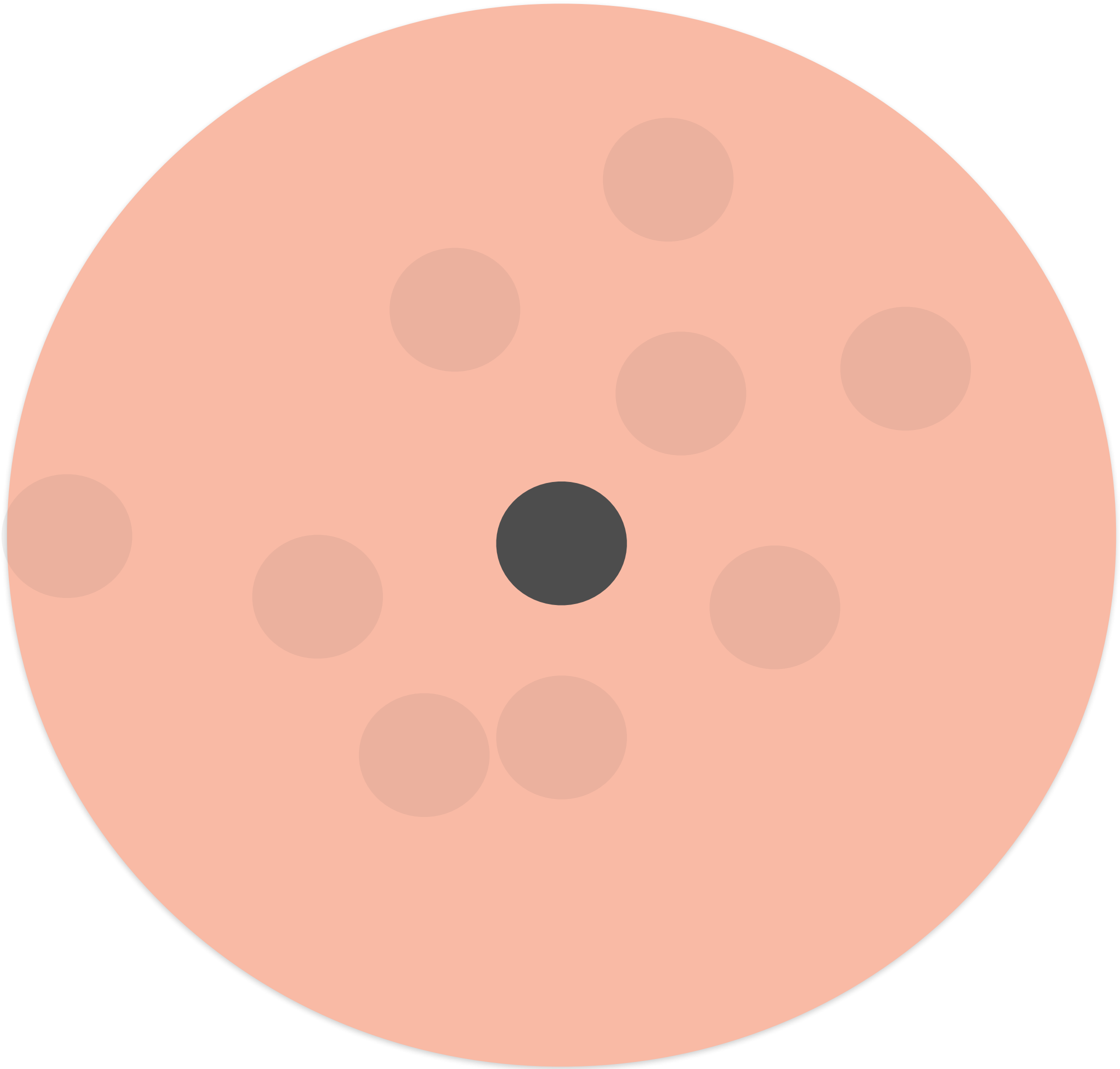
For each point, calculate a function based on all points in the neighborhood

That function is called the kernel



Flat Kernel

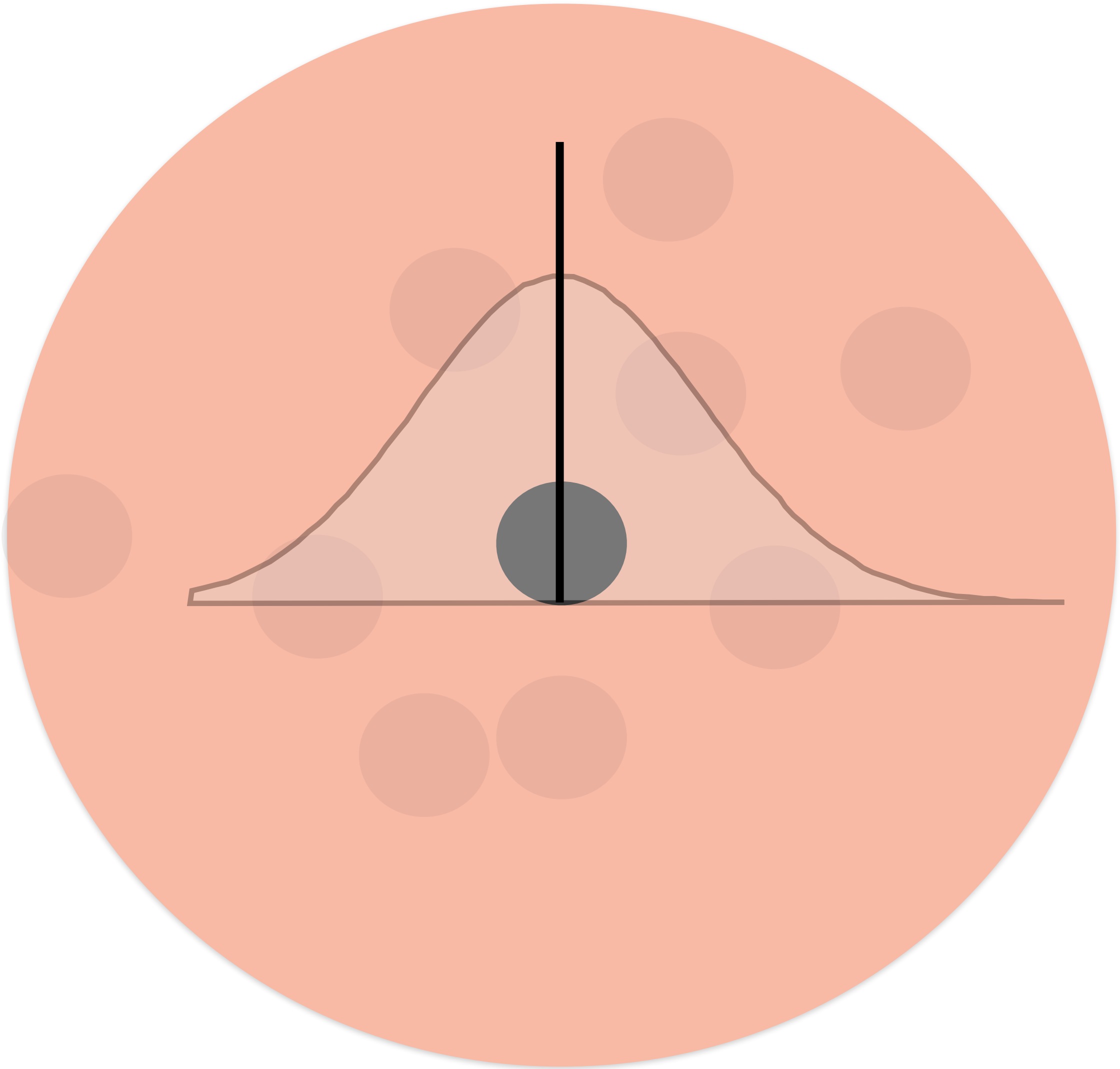
Flat kernel: sum of all points in neighborhood



Each point gets the same weight

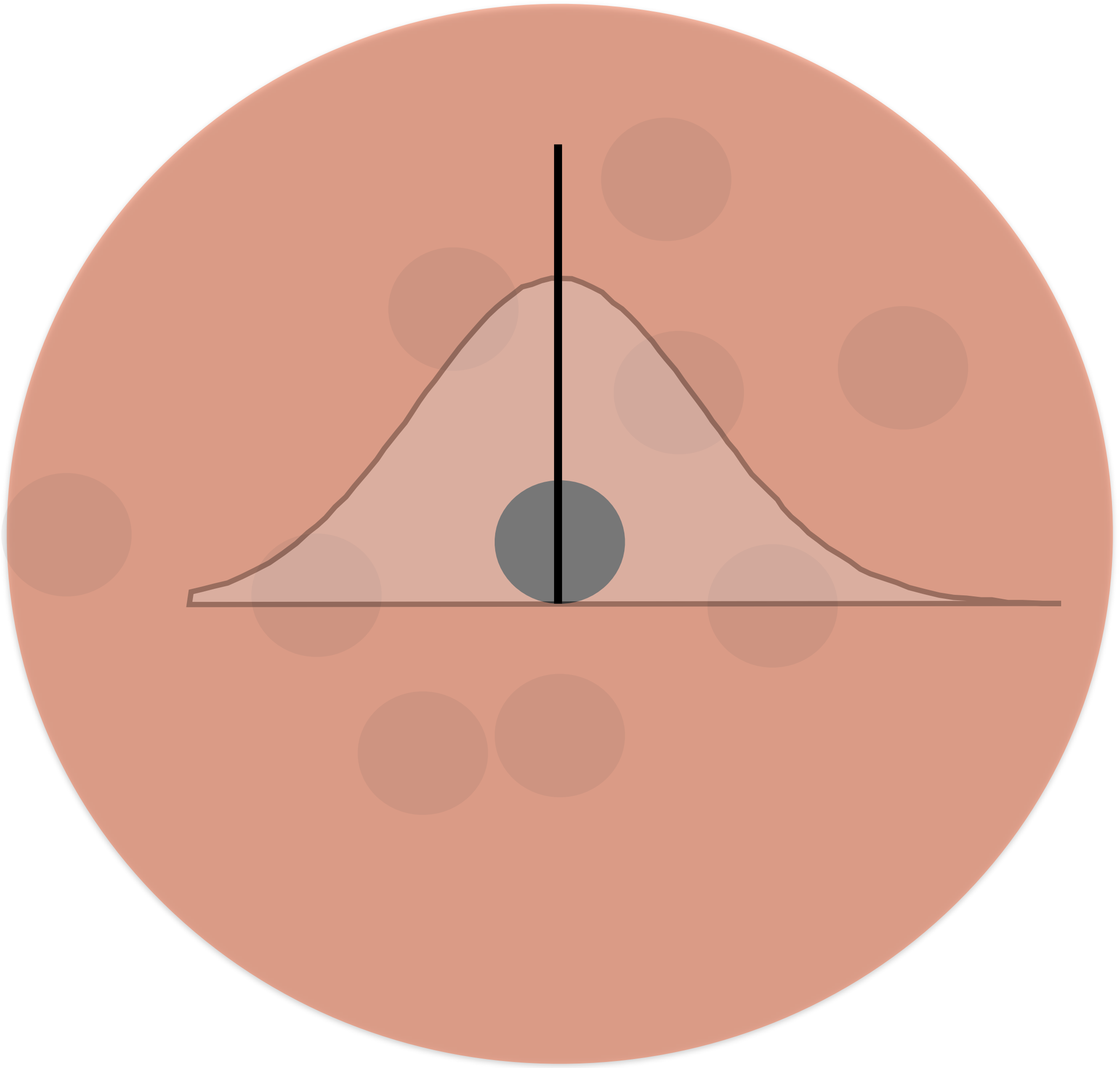
Gaussian (RBF) Kernel

Probability-weighted sum of points



What probability distribution?

Gaussian (RBF) Kernel

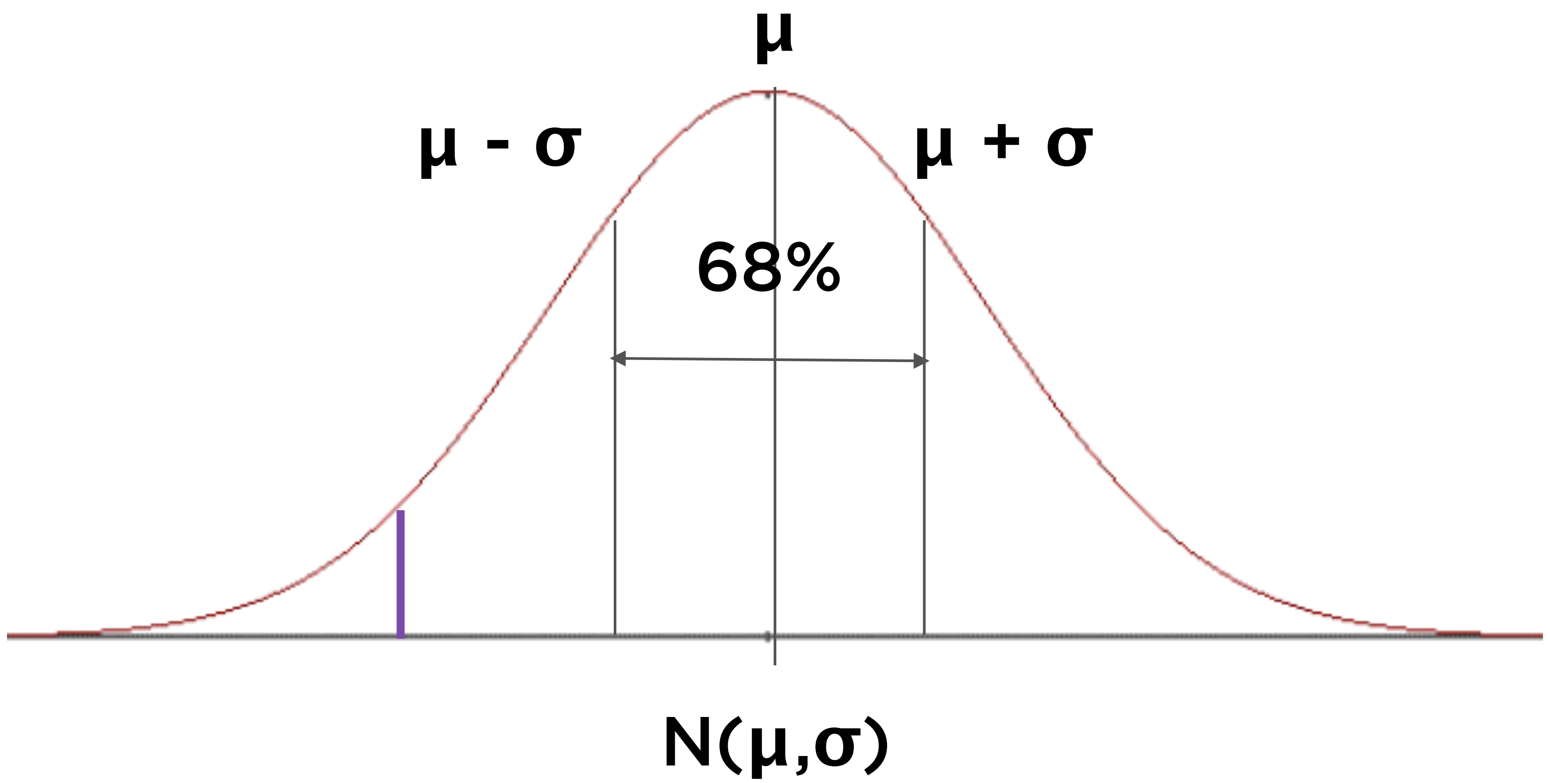


Gaussian probability distribution

Defined by

- mean μ
- standard deviation σ

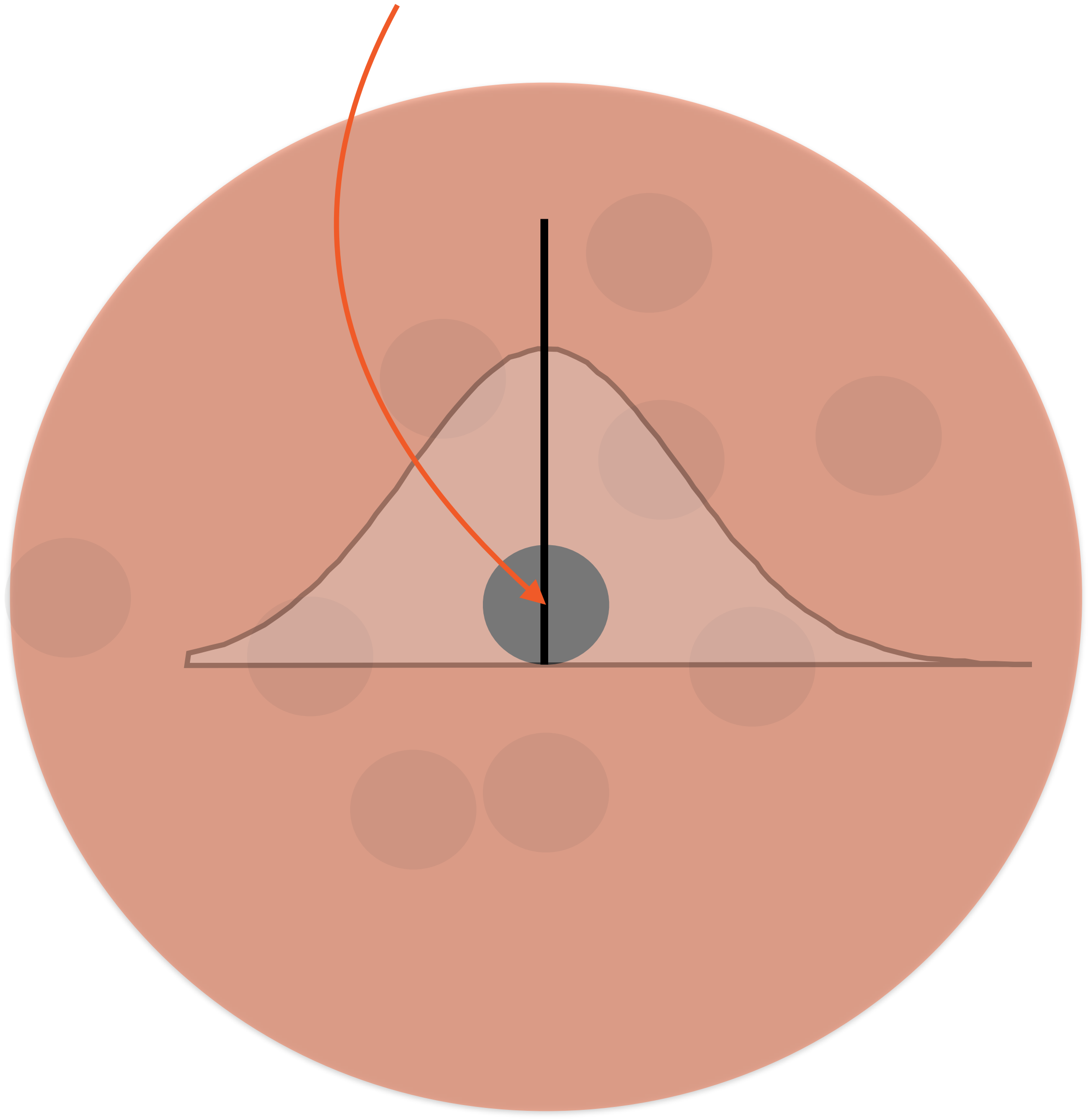
Gaussian Distribution



$$N(\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Gaussian (RBF) Kernel

Mean = Center point



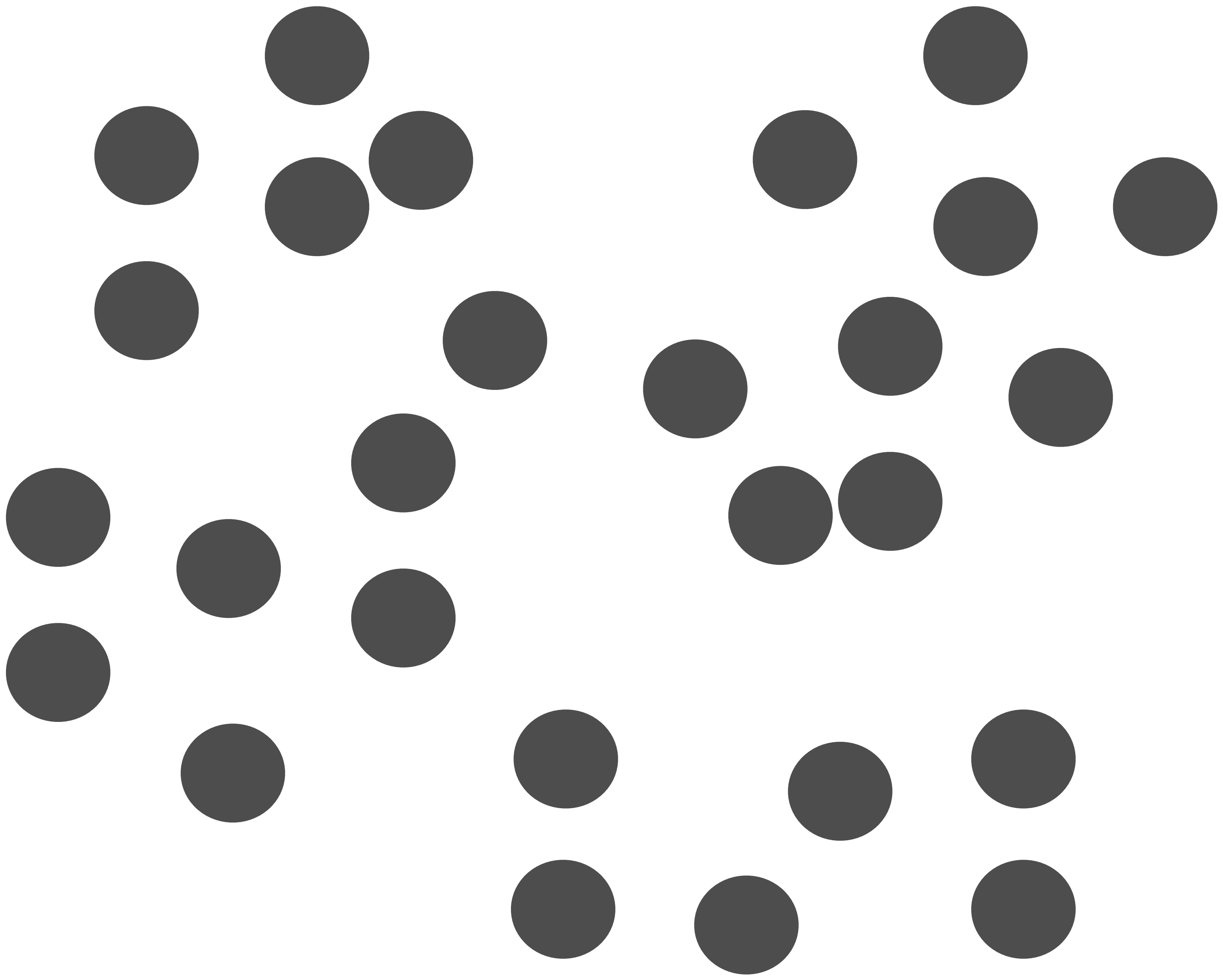
Mean μ = center point

Standard deviation σ ~ bandwidth

(Bandwidth is a hyperparameter)

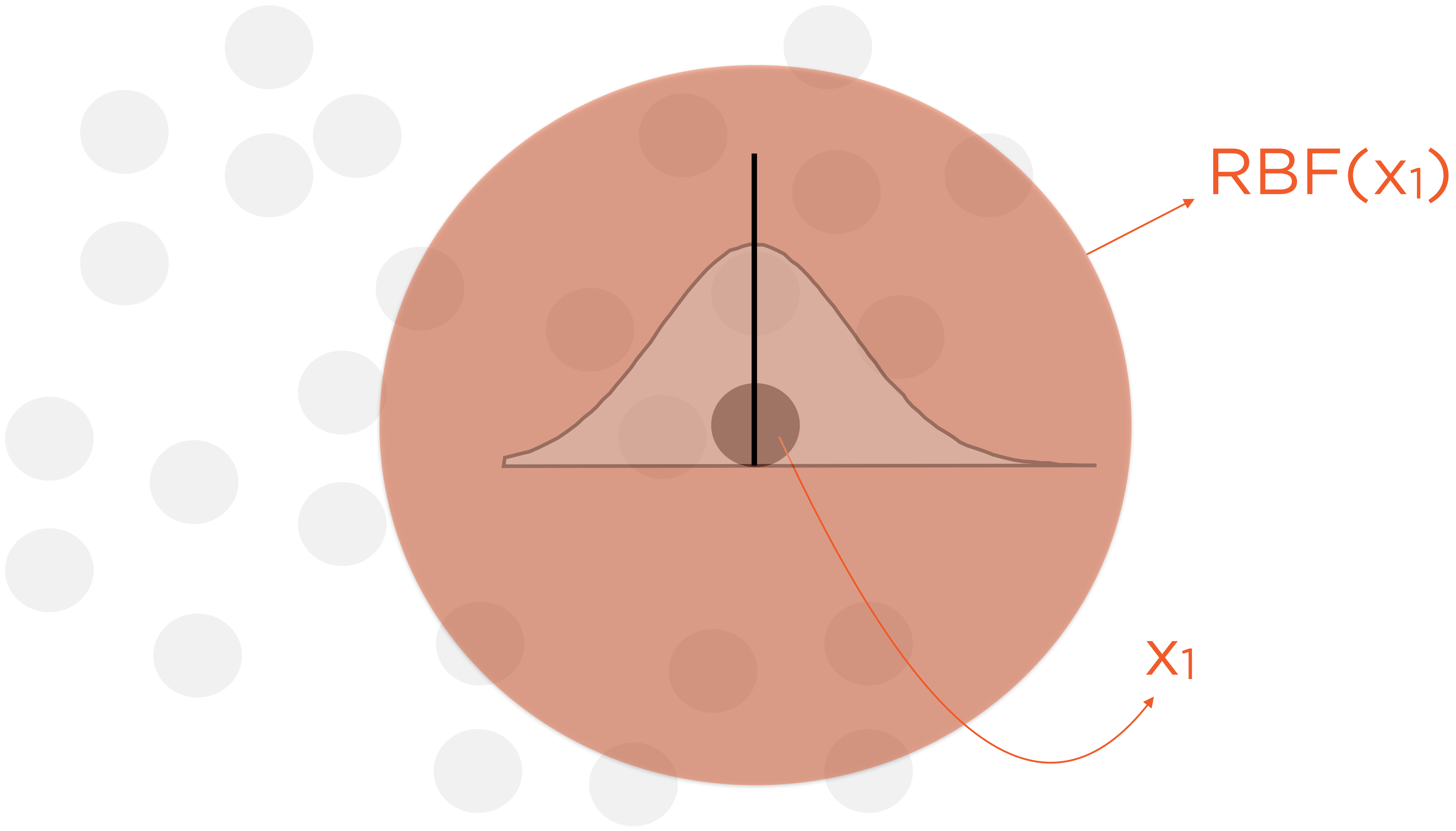
Mean Shift Clustering

Kernel is applied to each point



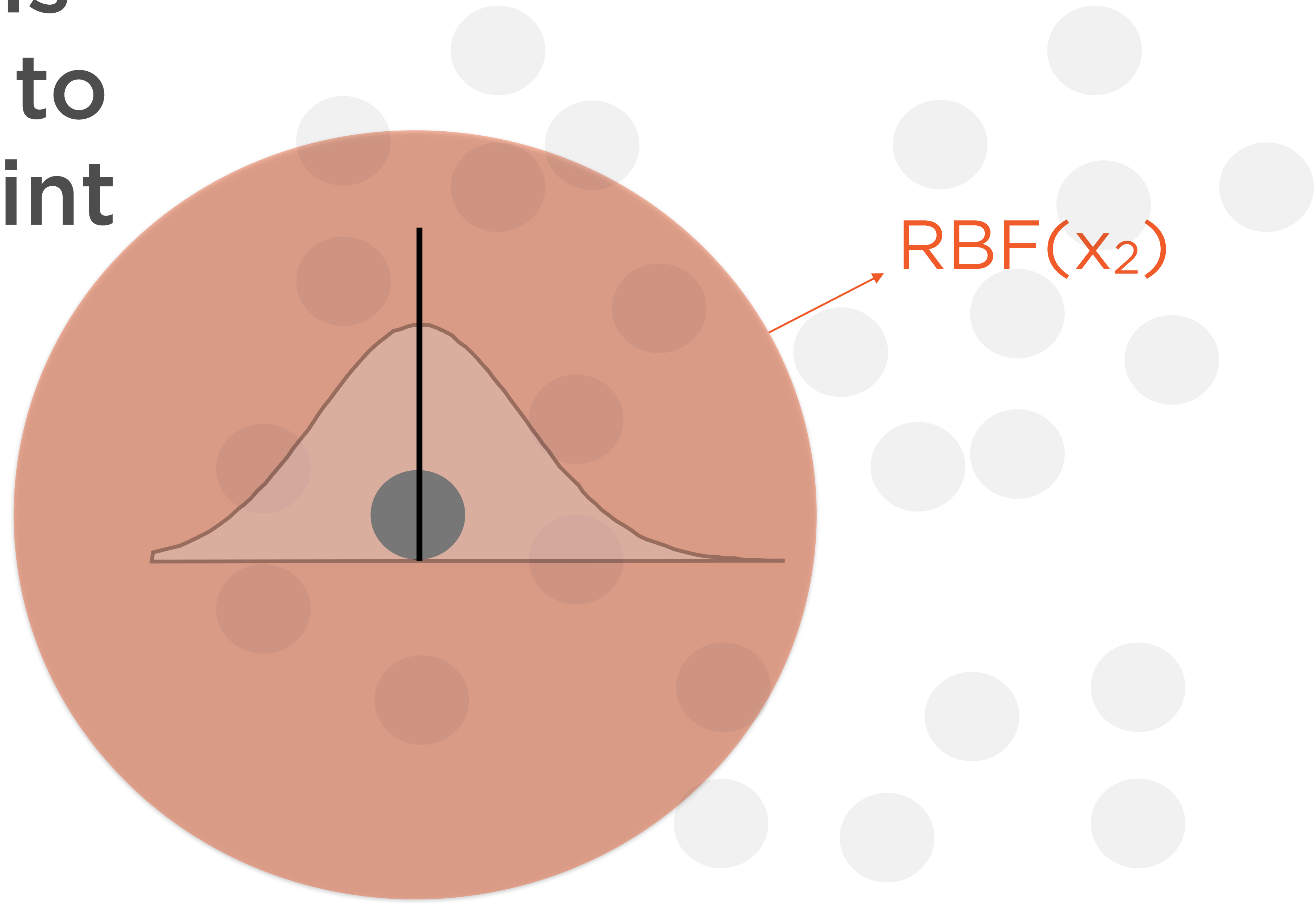
Mean Shift Clustering

Kernel is applied to each point



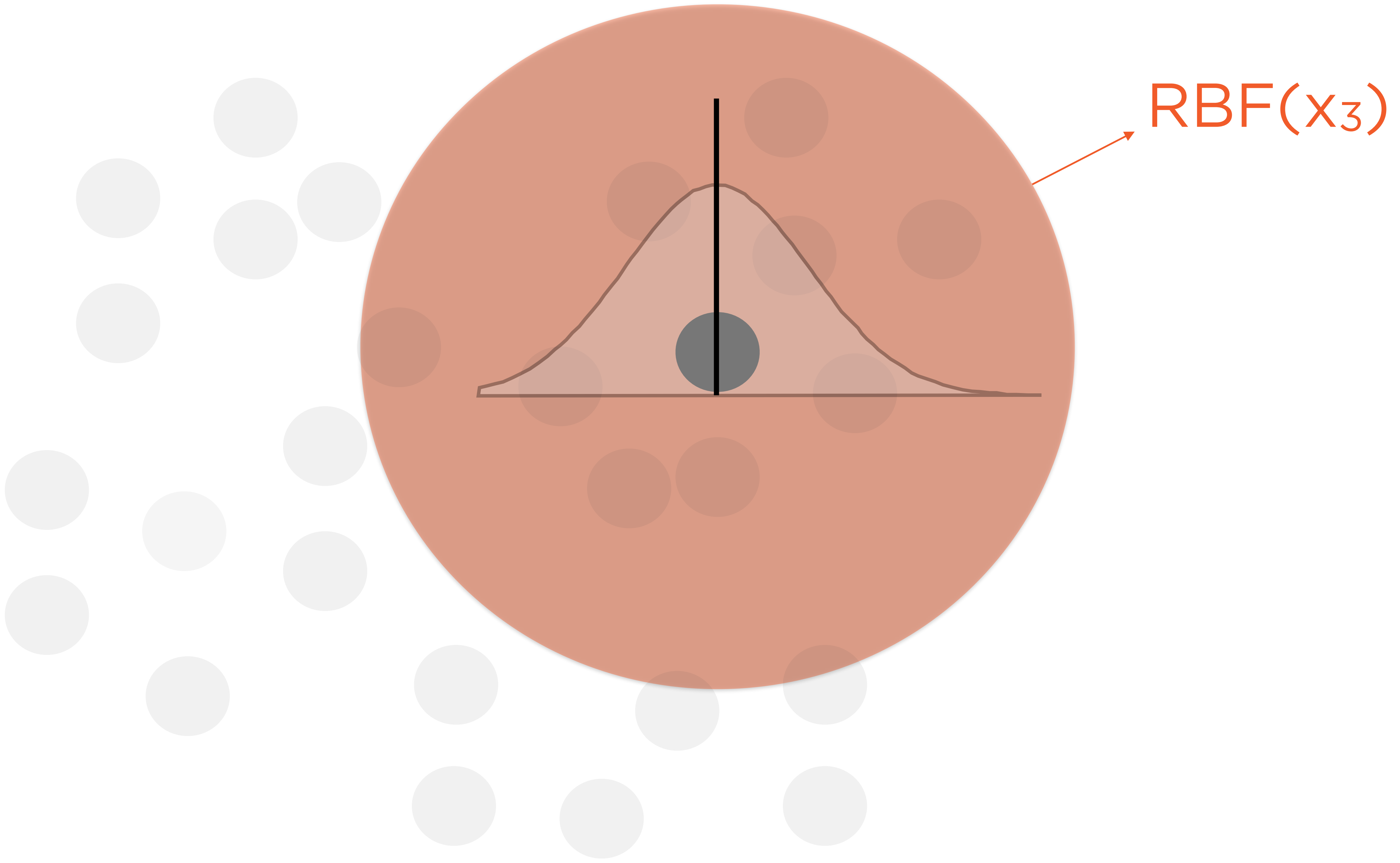
Mean Shift Clustering

Kernel is applied to each point



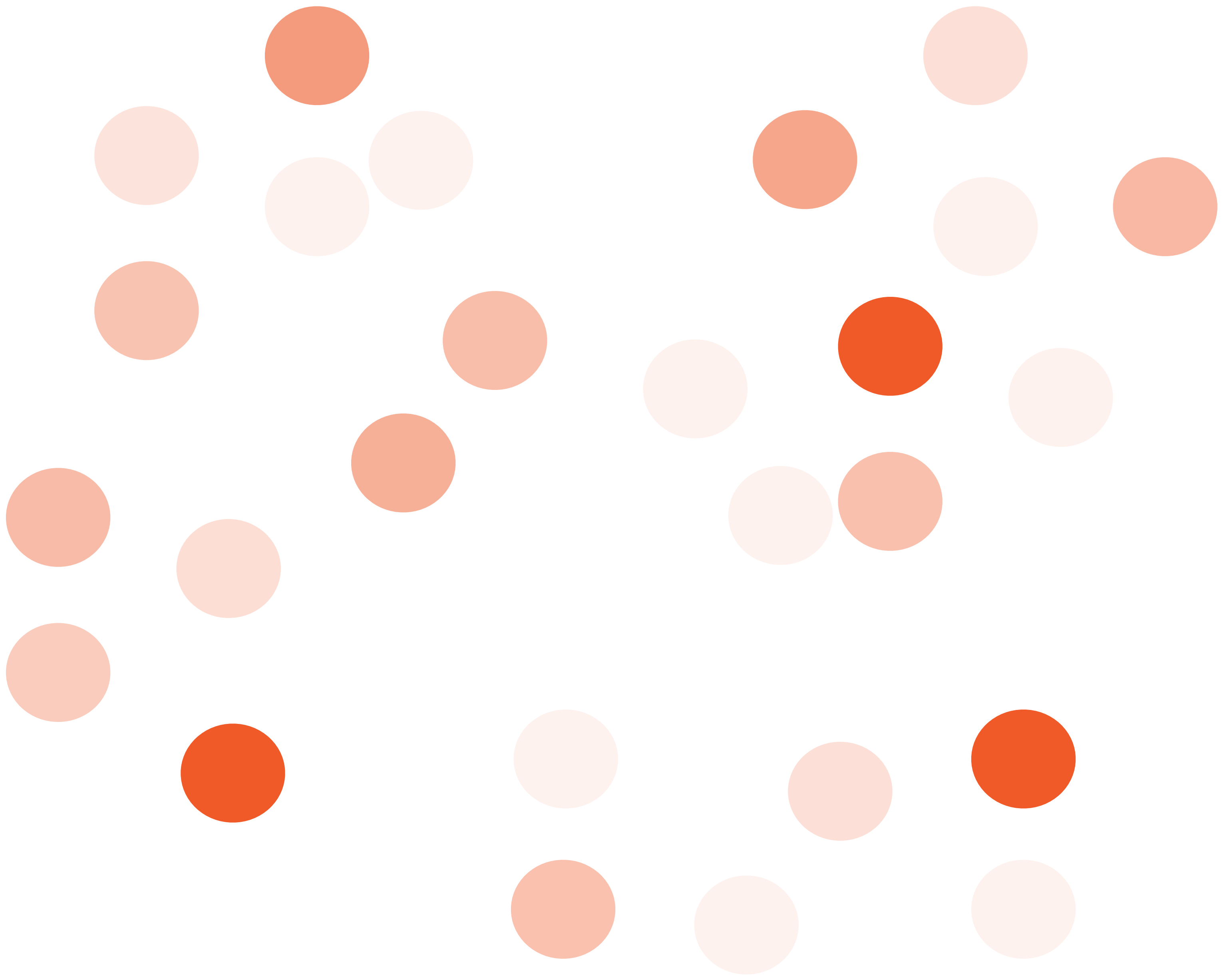
Mean Shift Clustering

Kernel is applied to each point



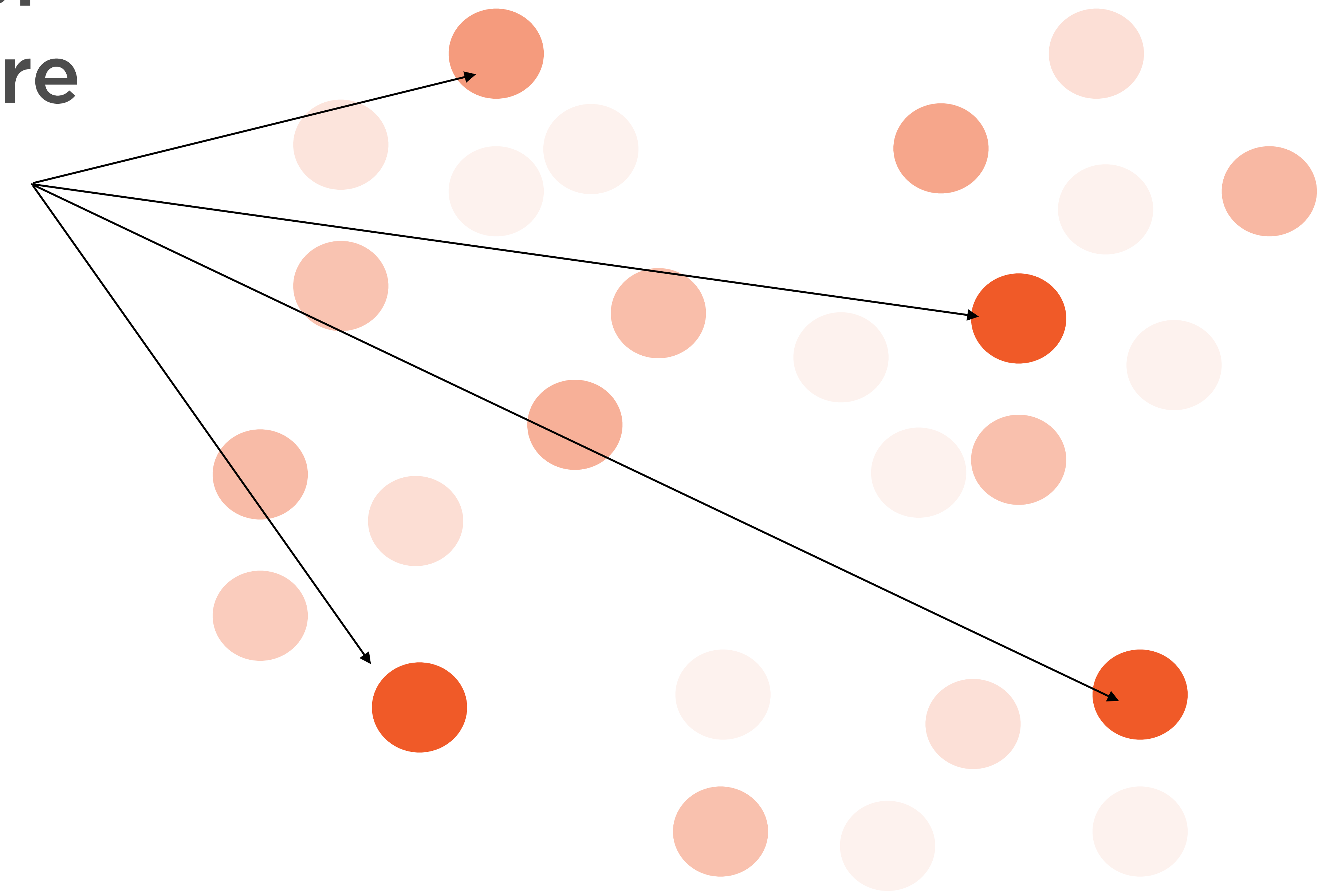
Mean Shift Clustering

Assume points are color-coded by magnitude of RBF



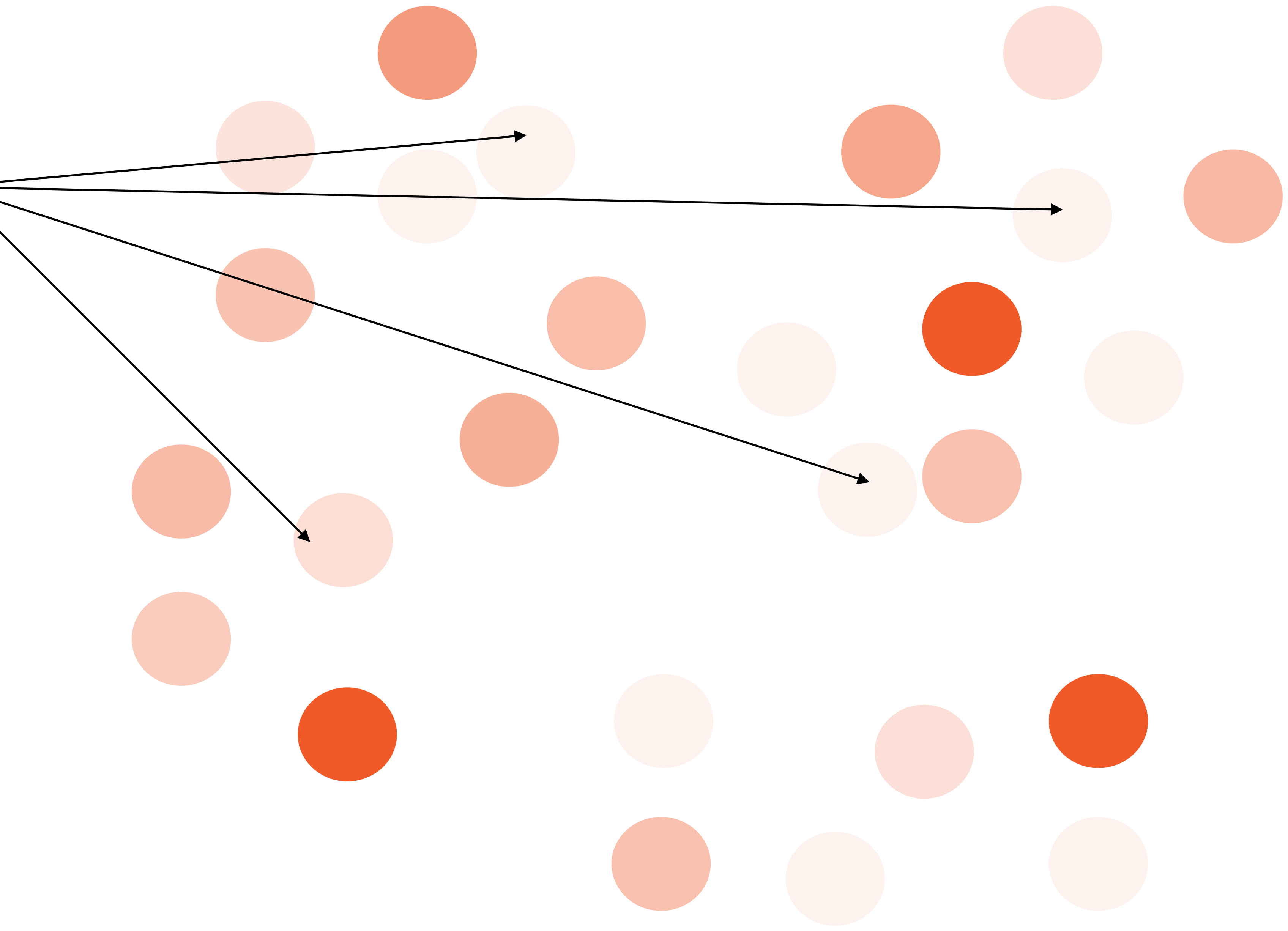
Mean Shift Clustering

High RBF values are peaks



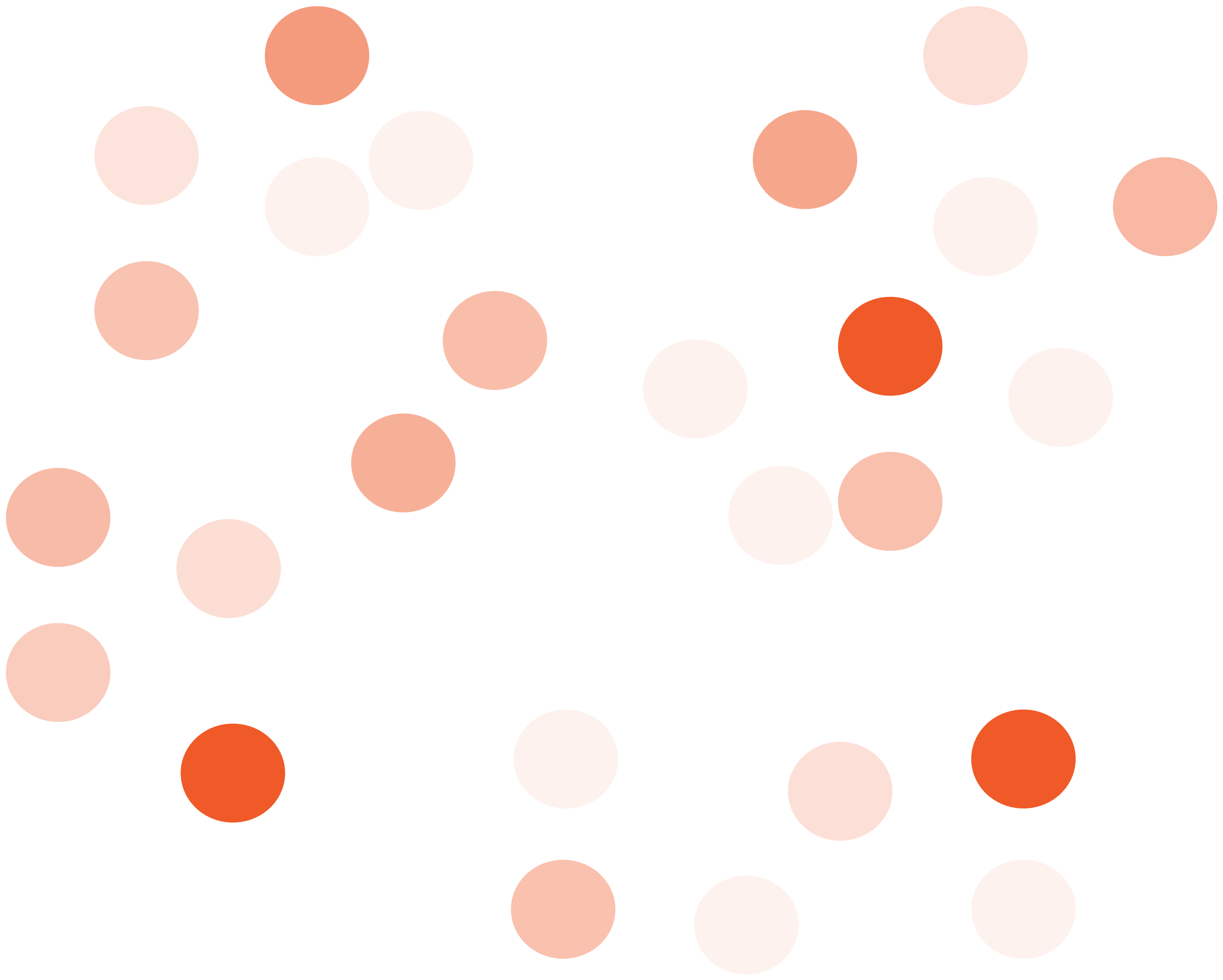
Mean Shift Clustering

Low RBF values are troughs



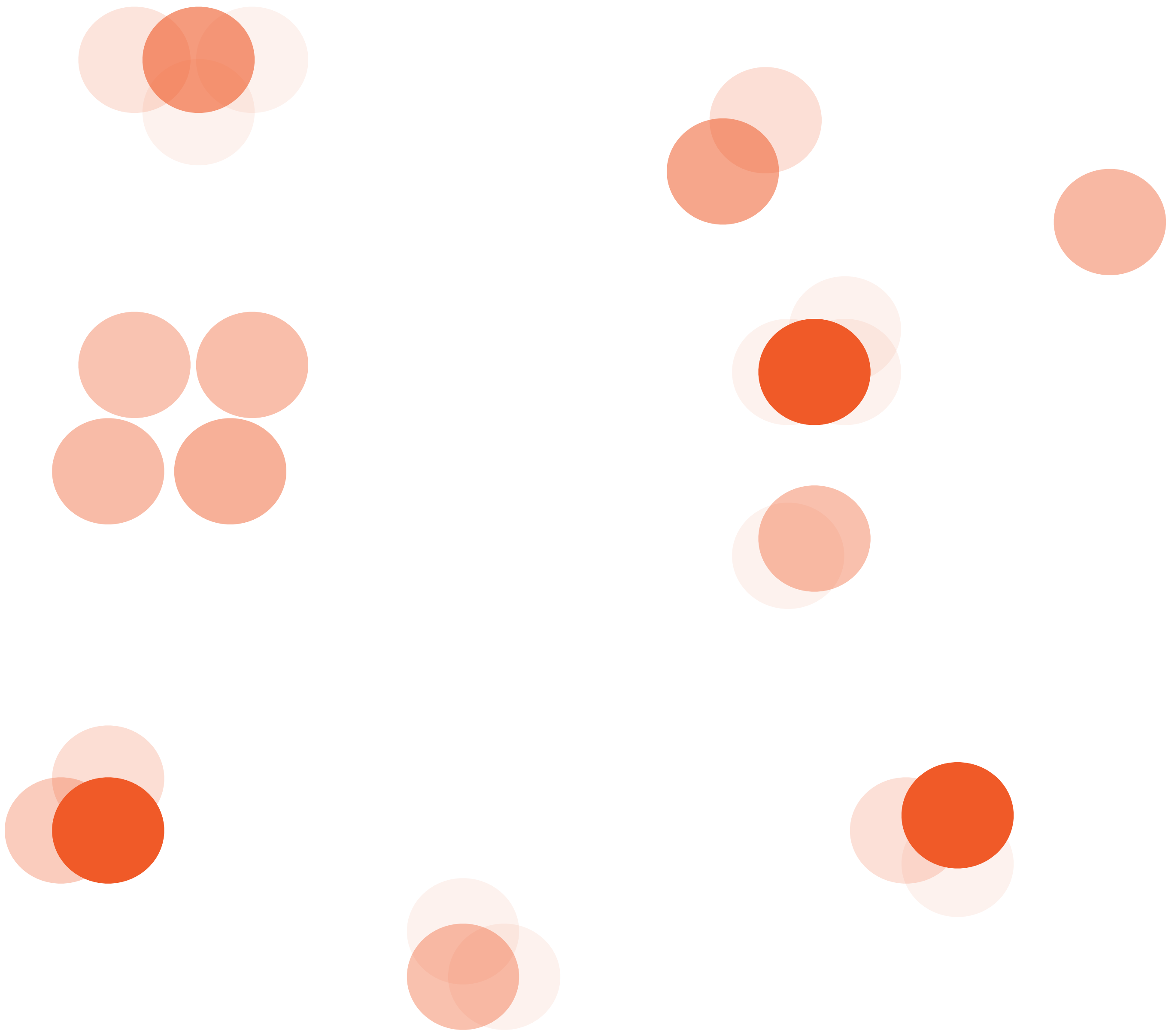
Mean Shift Clustering

Now, all points start to “shift” towards the nearest peak



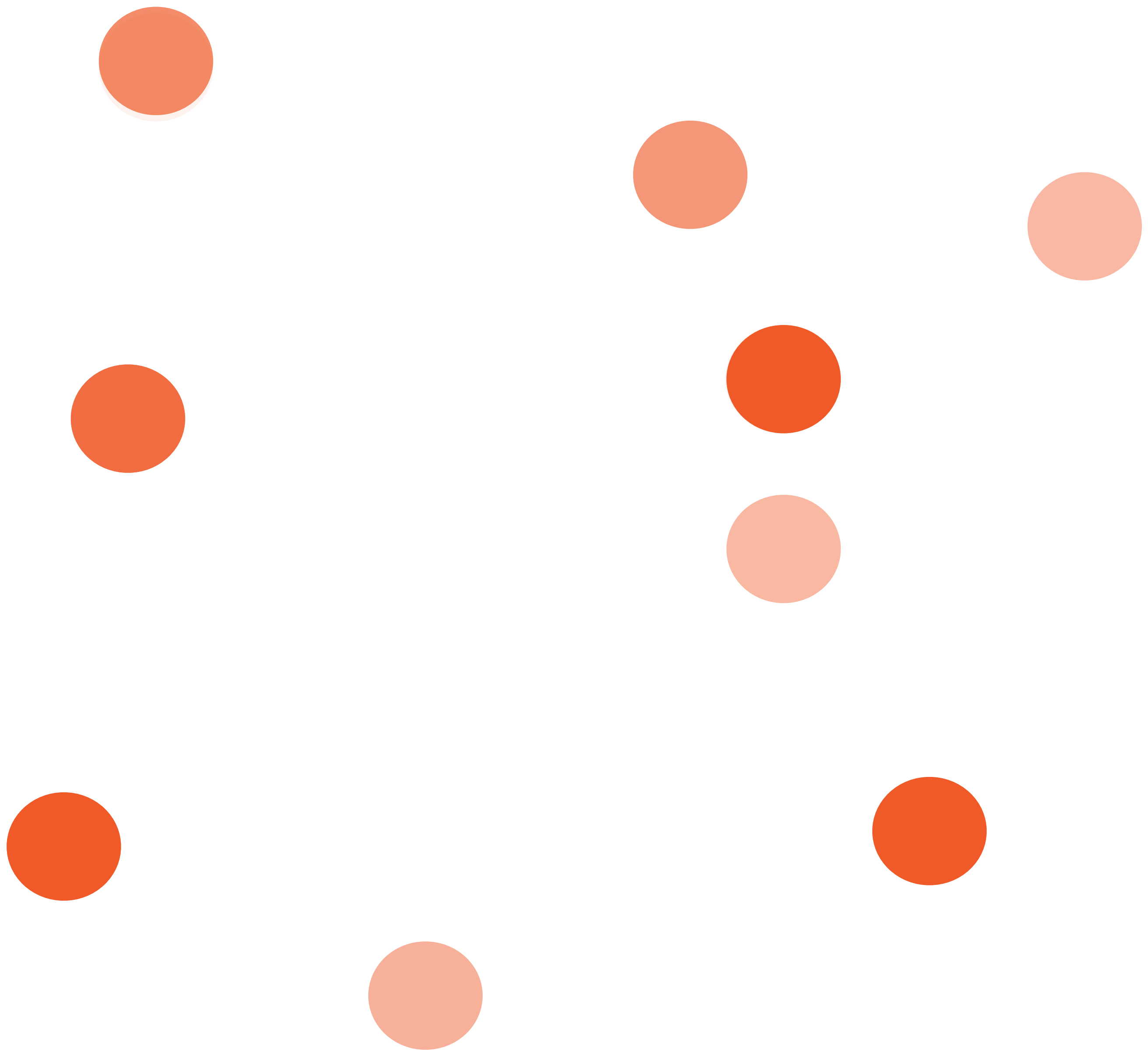
Mean Shift Clustering

Now, all points start to "shift" towards the nearest peak



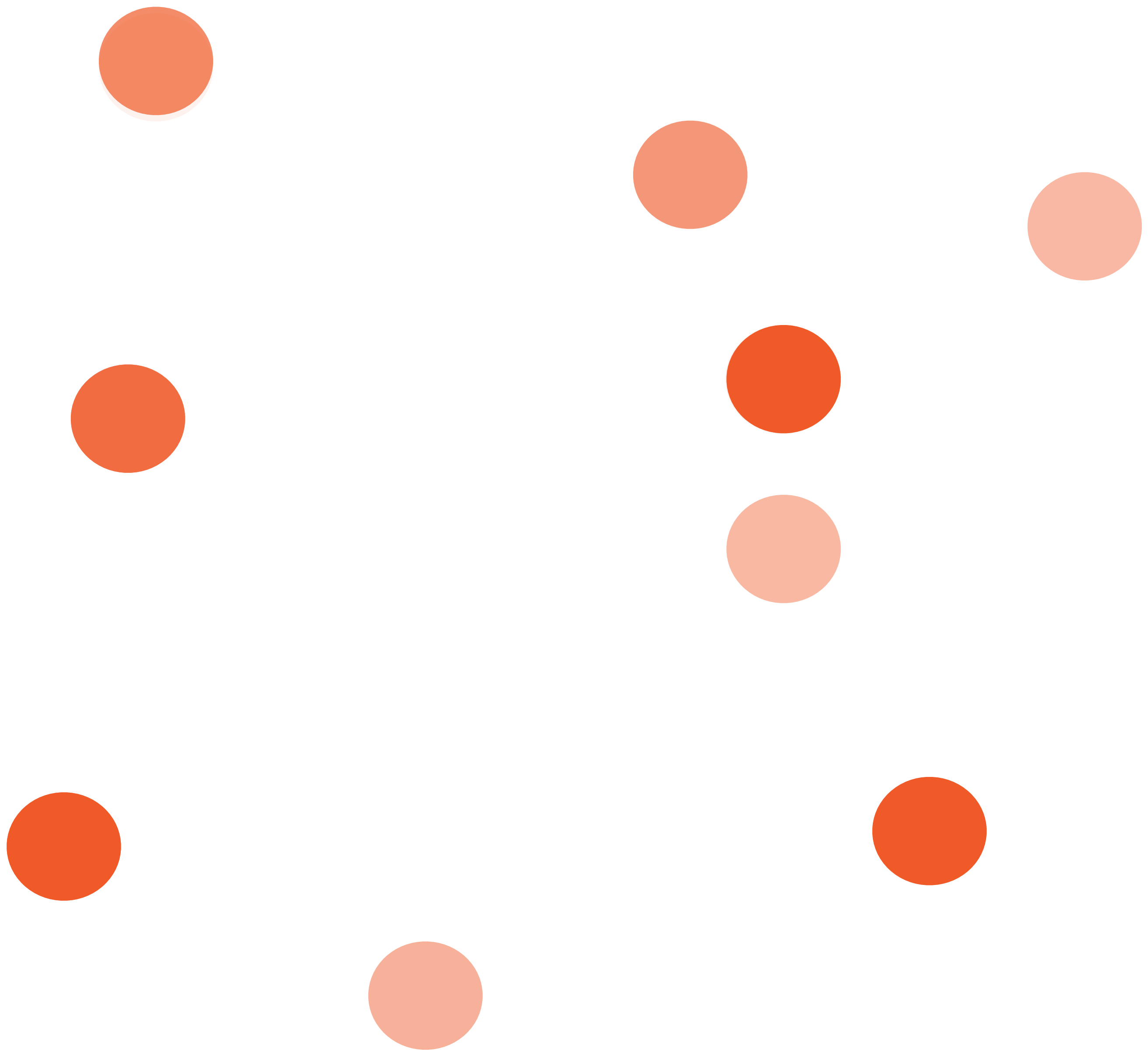
Mean Shift Clustering

Now, all points start to "shift" towards the nearest peak



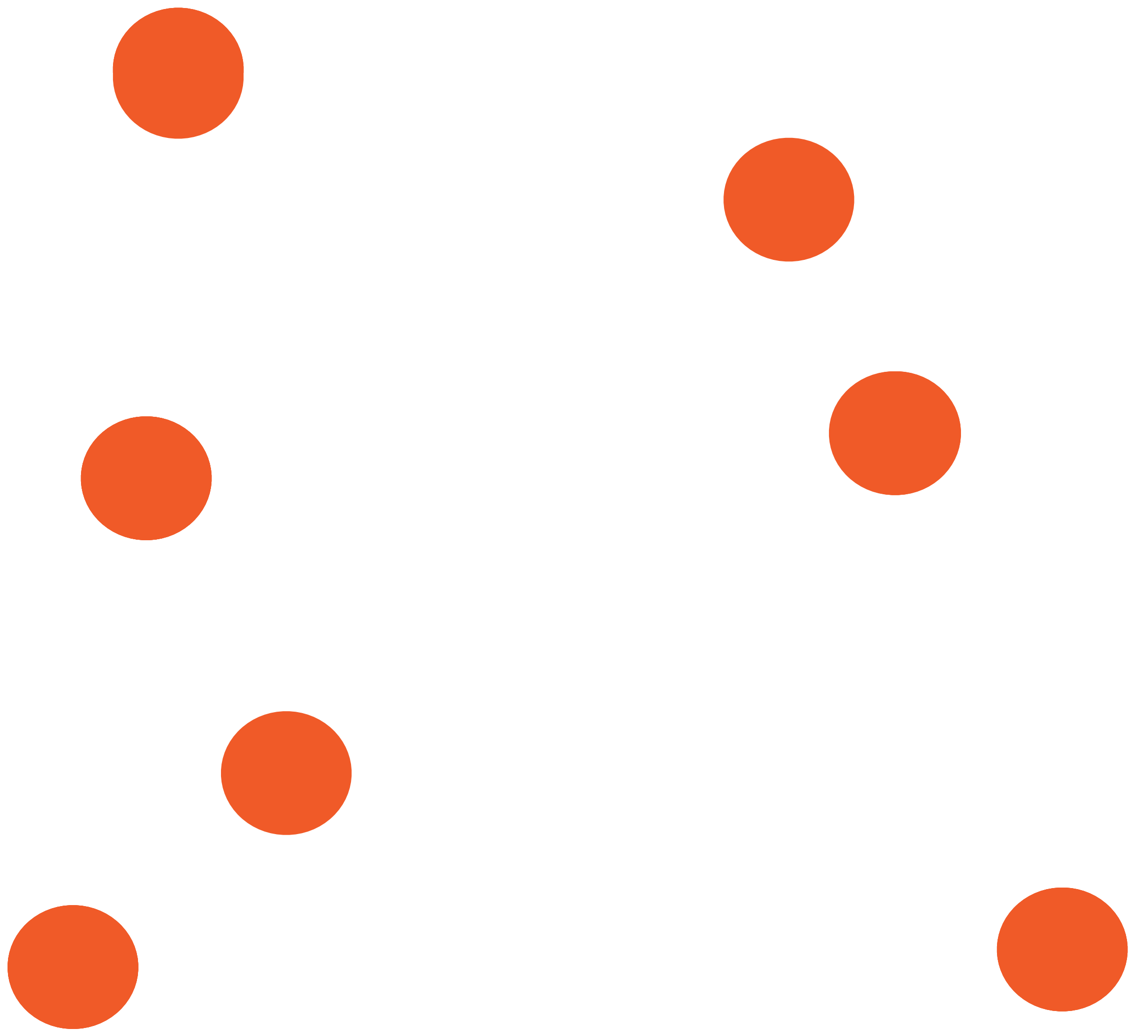
Mean Shift Clustering

This is the “mean shift”



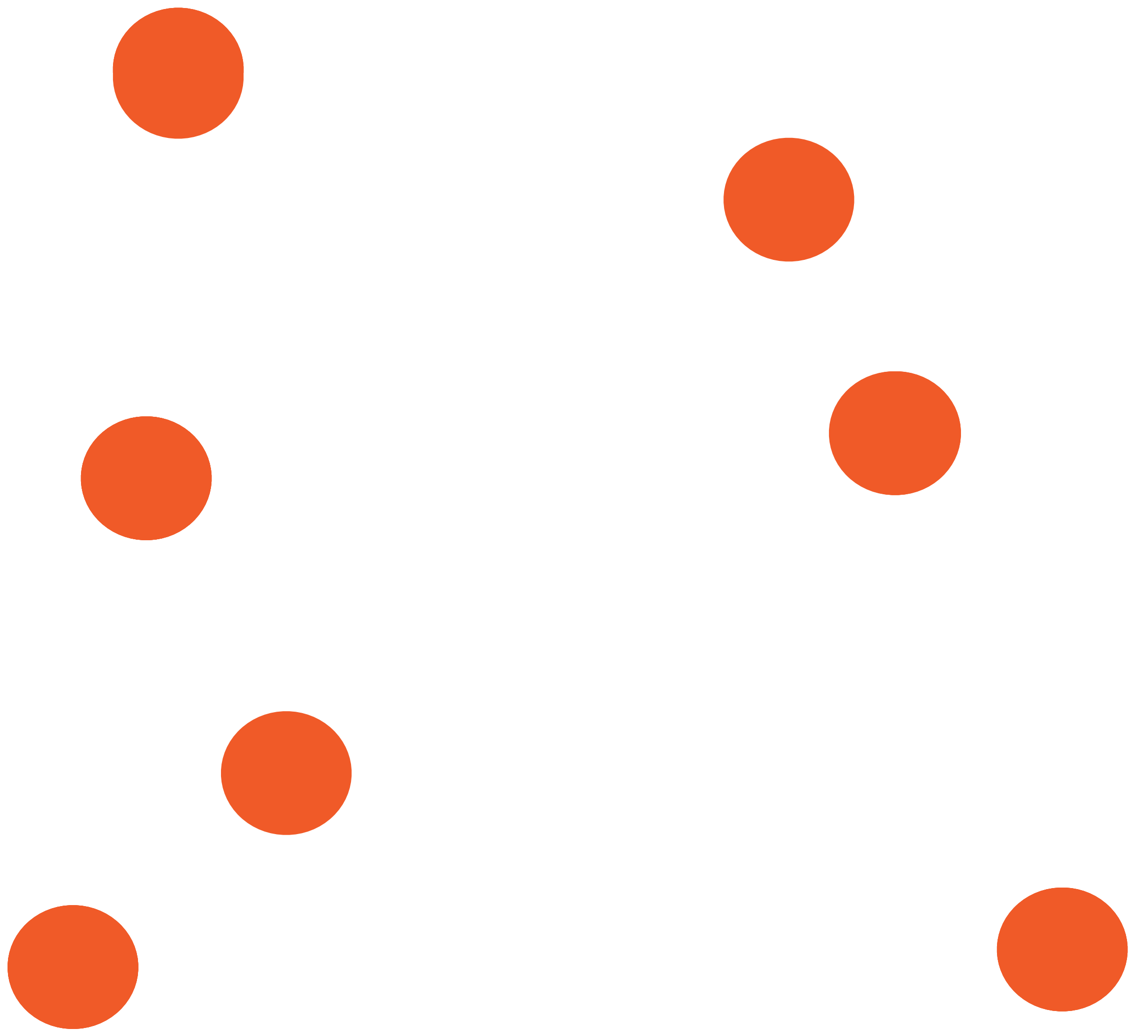
Mean Shift Clustering

This is the “mean shift”

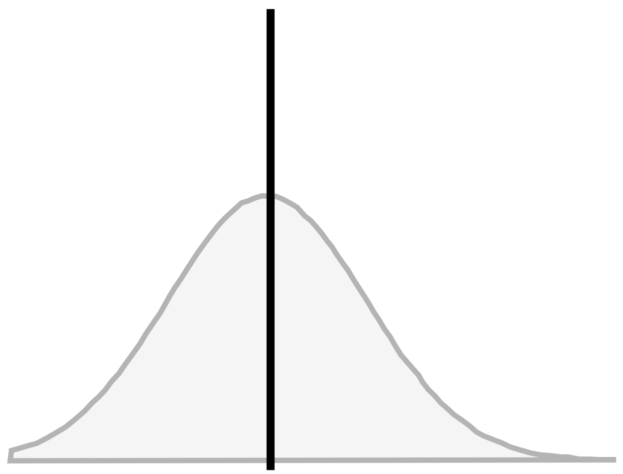


Mean Shift Clustering

Algorithm converges when points stop moving



Role of Bandwidth



$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

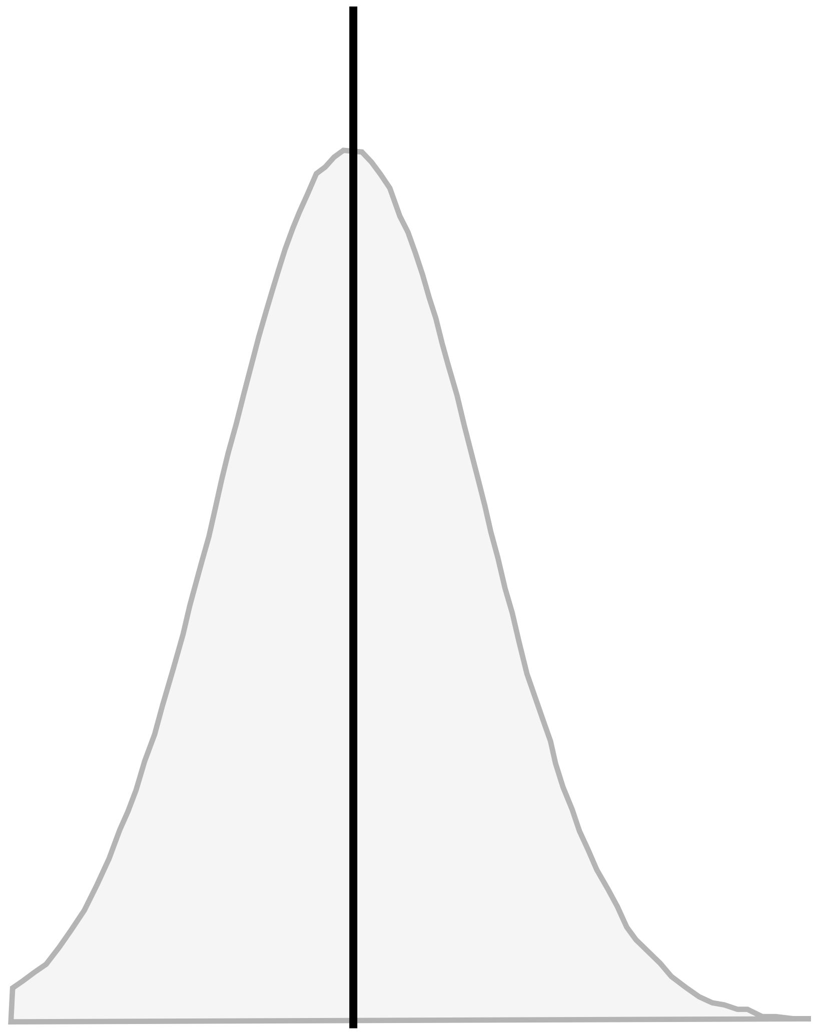
Standard deviation σ ~ bandwidth

Bandwidth is the only hyperparameter

Small bandwidth ~ tall skinny kernel

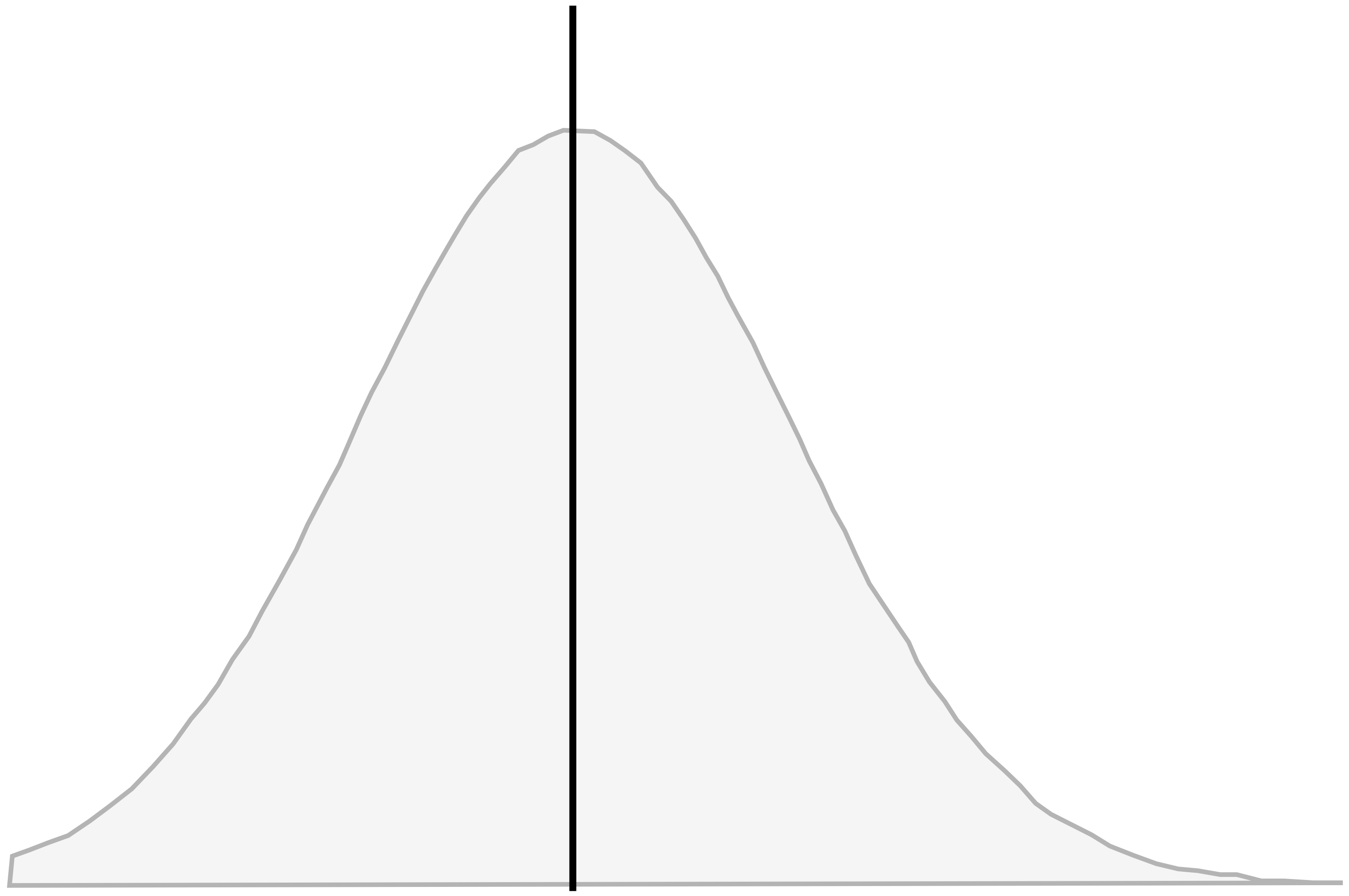
Large bandwidth ~ flat kernel

Role of Bandwidth



Tall skinny kernel

Ignore points far from the mean



Flatter kernel

Considers points far from the mean

Similar, yet Different

K-Means Clustering	Mean Shift Clustering
Need to specify number of clusters as hyperparameter	No need to specify number of clusters upfront as hyperparameter
Can’t handle some complex non-linear data	Uses density function to handle even complex non-linear data (e.g. pixels)
Less hyperparameter tuning needed	Hyperparameter tuning very important

Similar, yet Different

K-Means Clustering	Mean Shift Clustering
Computationally less intensive	Computationally very intensive
O(N) in number of data points	$O(N^2)$ in number of data points
Struggles with outliers	Copes better with outliers

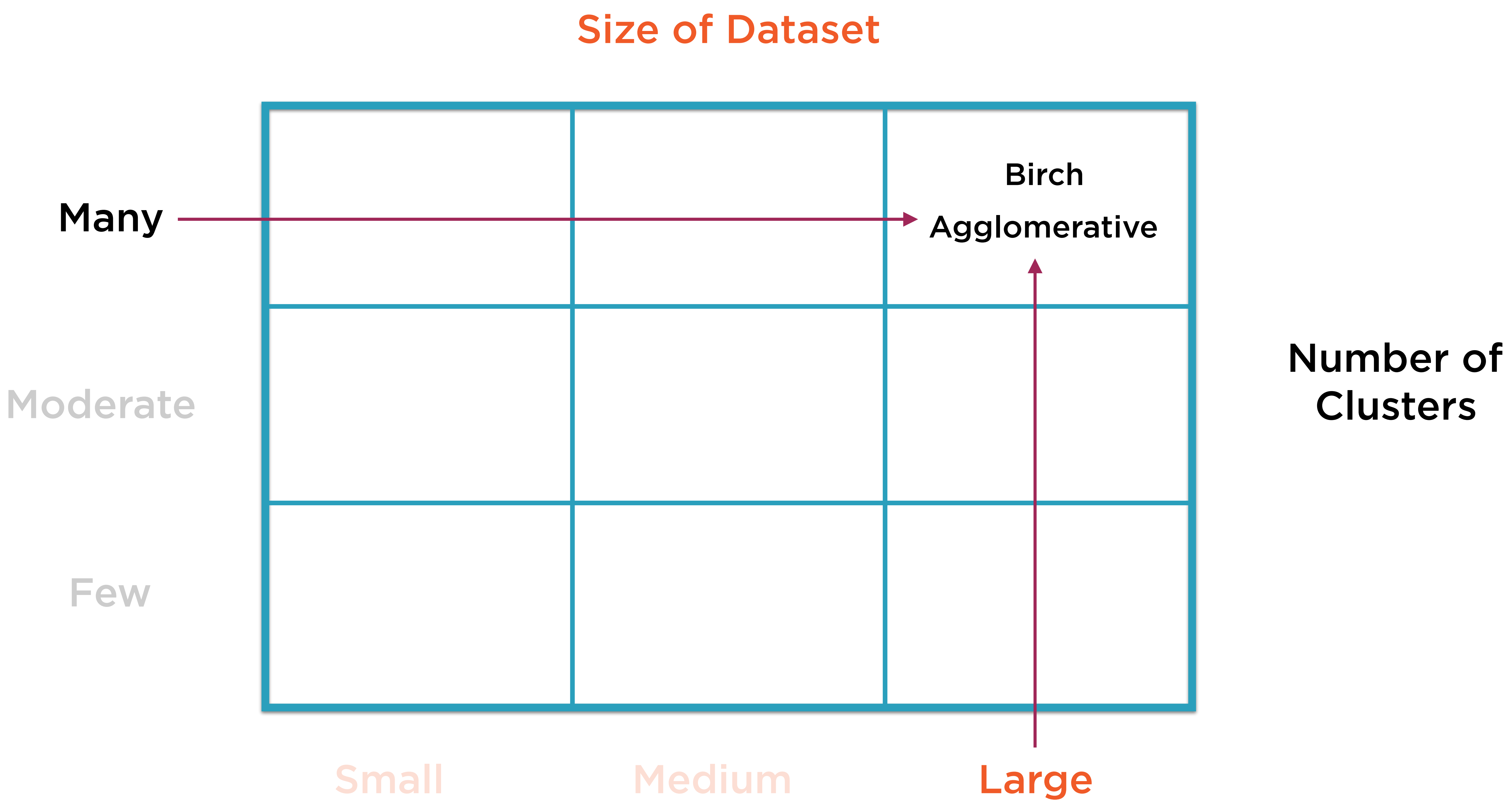
Demo

Implementing mean-shift clustering

Demo

Implementing BIRCH clustering

Choosing Clustering Algorithms



Large Datasets, Many Clusters

Consider BIRCH or Agglomerative clustering

BIRCH detects and removes outliers

Also incrementally processes incoming data and updates clusters

BIRCH Algorithm

Balanced **I**terative **R**educing and **C**lustering using **H**ierarchies

Hierarchical clustering algorithm

Very effective at handling noise and outliers

Very memory and time efficient

Entire dataset need not be loaded into memory

BIRCH Algorithm

Incrementally clusters incoming data points

Updates clusters as new data arrives

Online-learning algorithm

Demo

Implementing affinity propagation clustering

Choosing Clustering Algorithms

Size of Dataset

Number of Clusters	Small	Medium	Large
Many	Mean-shift Affinity Propagation		Birch Agglomerative
Moderate			
Few			

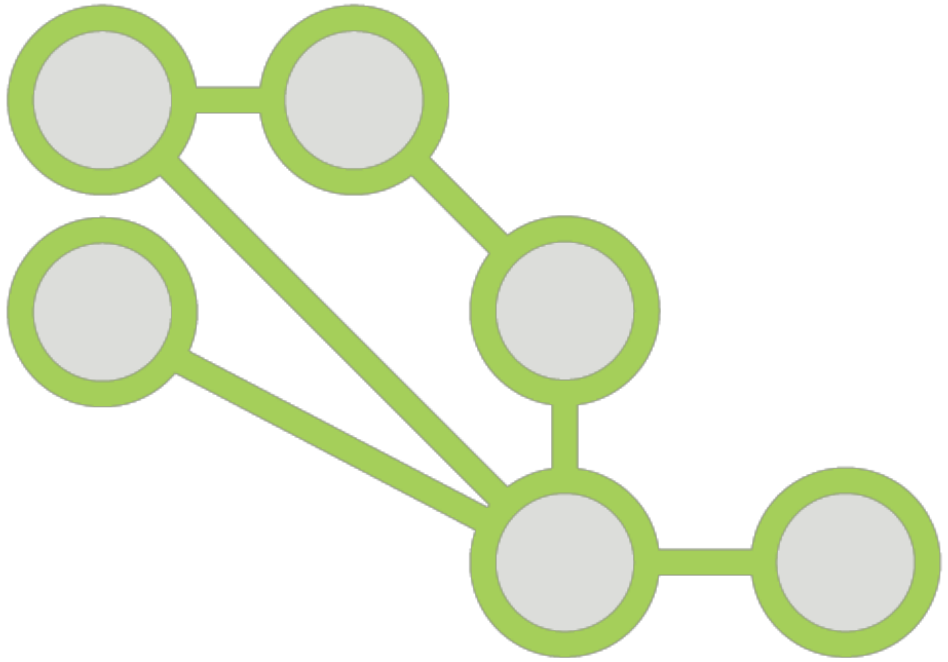
Small Datasets, Many Clusters

Consider Mean-shift or Affinity Propagation clustering

Both work well with uneven cluster sizes and manifold shapes

Affinity Propagation does not need number of clusters to be specified

Affinity Propagation



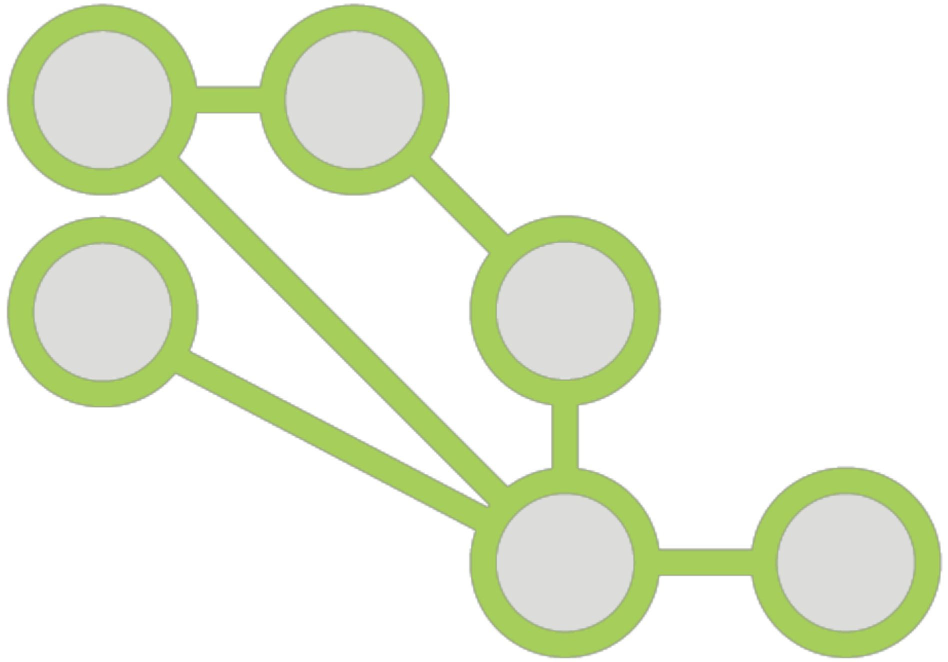
Makes no assumptions about internal data of points

Accepts graph distances (nearest neighbor graphs)

Attempts to find exemplars

Exemplars are points in training data that are representative of clusters

Affinity Propagation



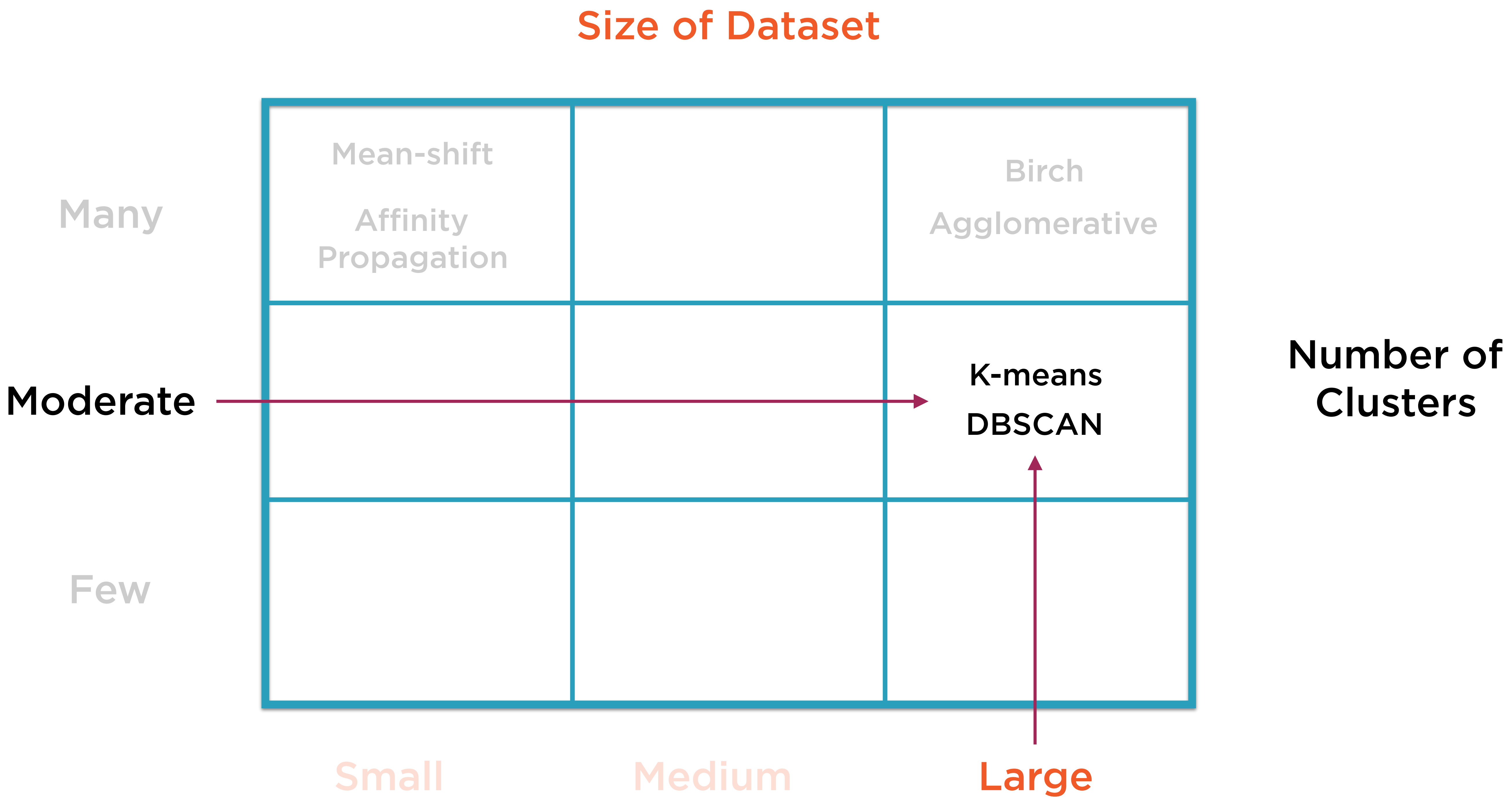
Data points are network nodes which send messages to one another

Messages express the willingness of points to be exemplars

Demo

Implementing mini-batch K-means clustering

Choosing Clustering Algorithms



Large Datasets, Moderate Cluster Count

Consider K-means and DBSCAN

K-means for even cluster sizes and flat surfaces

Mini-batch K-means

Perform K-means on a randomly sampled subsets

Iteratively performed on batches called mini-batches

Far faster than full K-means

Performance usually only slightly worse

Demo

Implementing spectral clustering with a precomputed similarity matrix

Choosing Clustering Algorithms

Size of Dataset

Number of Clusters	Small	Medium	Large
Many	Mean-shift Affinity Propagation		Birch Agglomerative
Moderate			K-means DBSCAN
Few		**Spectral**	

Small Datasets, Few Clusters

Consider Spectral Clustering

Simple to implement, intuitive results

Even cluster size

Fine for manifolds

Relies on distances between points

Spectral Clustering

Creates an affinity matrix of input data points

Input can be a precomputed similarity matrix

Eigenvalue (spectrum) decomposition applied

Dimensionality reduction is followed by pairwise similarity measurement

Spectral Clustering

DBSCAN is a special case of spectral clustering

K-means kernel clustering is a spectral clustering too

First applies kernel trick, then implements K-means

Summary

Hierarchical clustering techniques

Agglomerative and BIRCH clustering

DBSCAN clustering

Mean-shift clustering

Affinity clustering

Spectral clustering

Mini-batch K-means clustering