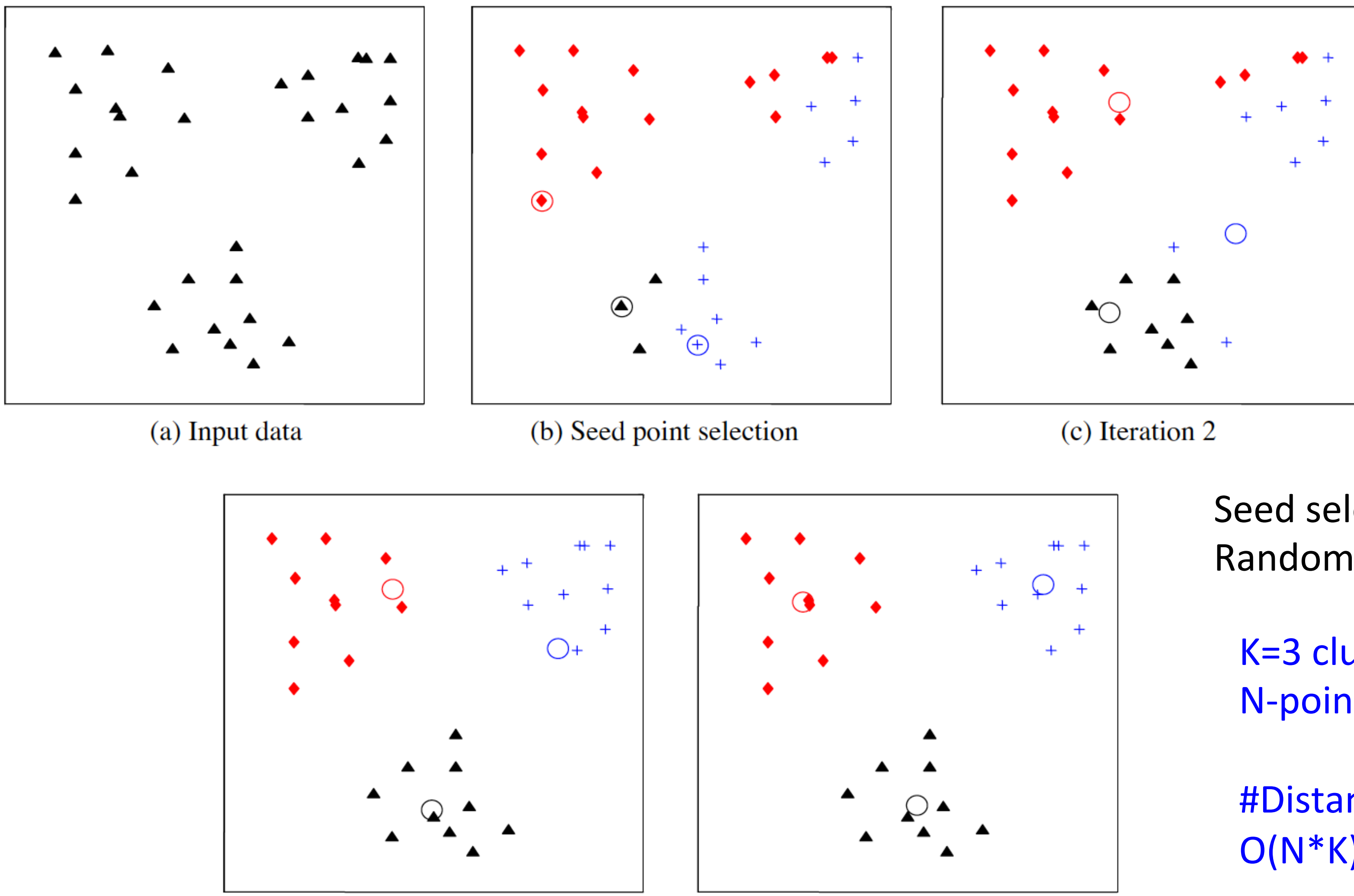
Interactive K-Means Clustering

CS8001 Visual Computing
K. Palaniappan & Josh Fraser
EECS Dept., University of Missouri

AK Jain, Data clustering: 50 years beyond K-means, PRL, 2010
David Arthur and Sergei Vassilvitskii, k-means++: The Advantages of Careful Seeding, 2006

Assignment#4
K-Means Clustering Algorithm



(a) Input data (b) Seed point selection (c) Iteration 2

(d) Iteration 3 (e) Final clustering

Seed selection
Randomly selectec

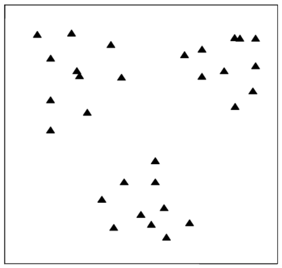
K=3 clusters
N-points

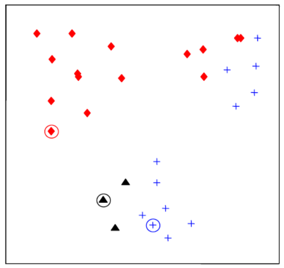
#Distances:
O(N*K)

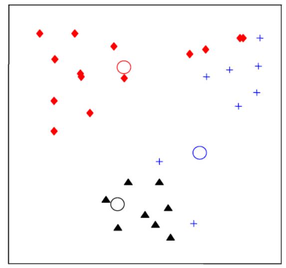
Mean vs Centroid

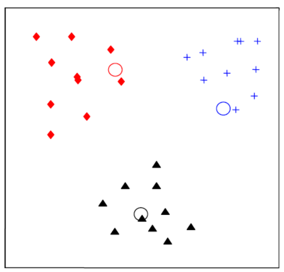
AK Jain, Data clustering: 50 years beyond K-means, PRL, 2010

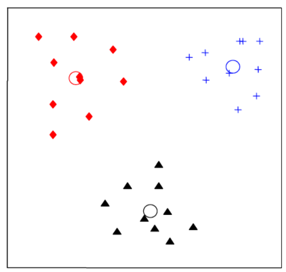
Assignment #4
Stage 1: K-Means Clustering Algorithm


(a) Input data


(b) Seed point selection


(c) Iteration 2


(d) Iteration 3


(e) Final clustering

Num points (2D, 3D, N-d), N>3
Num true clusters, T= 5
Gaussian distributed (simulated points)
K – user input, K=3

Overall complexity of distance computations per iteration
O(N*K*D)

Center selection:
Furthest away or based on distribution of distances
Streaming-based reservoir sampling

Equidistant points:
Breaking ties

K-Means algorithm (1955)

1. Select an initial partition with K clusters; repeat steps 2 and 3 until cluster membership stabilizes.
2. Generate a new partition by assigning each pattern to its closest cluster center.
3. Compute new cluster centers.

AK Jain, Data clustering: 50 years beyond K-means, PRL, 2010

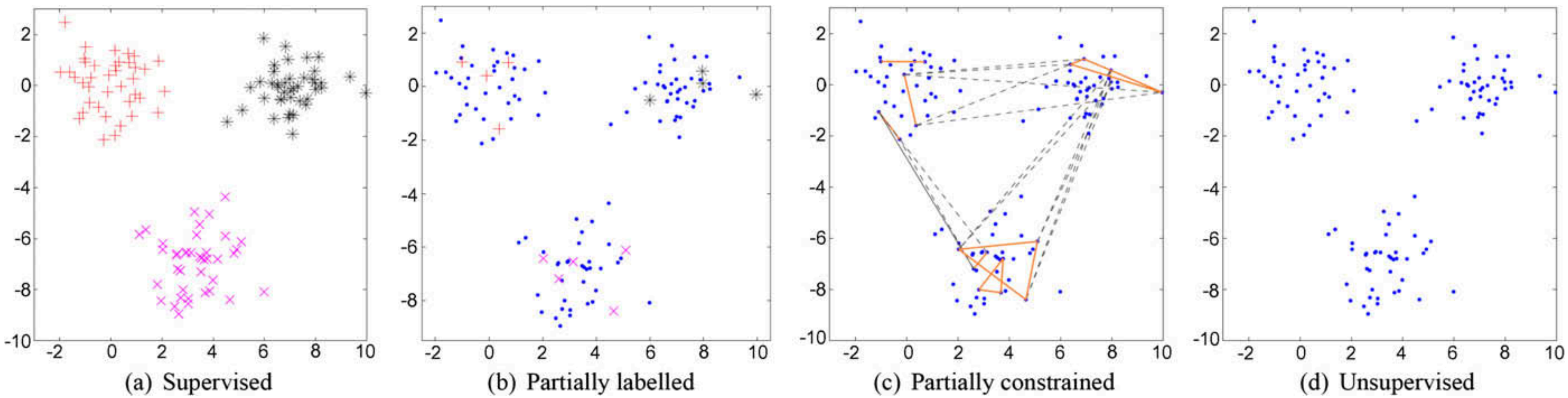
K-Means Clustering: Open Questions

1) What is a cluster?
2) What features should be used?
3) Should the data be normalized?
4) Does the data contain any outliers?
5) How do we define the pair-wise similarity? Or what is the distance function.
6) How many clusters are present in the data?
7) Which clustering method should be used?
8) Does the data have any clustering tendency?
9) Are the discovered clusters and partition valid?

K-Means Clustering: Extensions

- Semi-supervised or few/partially labeled samples
- Weights and constraints between points
- Lower dimensional projections before applying clustering (pre-dimensionality reduction)
- Incorporating density of points during clustering
- Hierarchical clustering
- Ensemble clustering

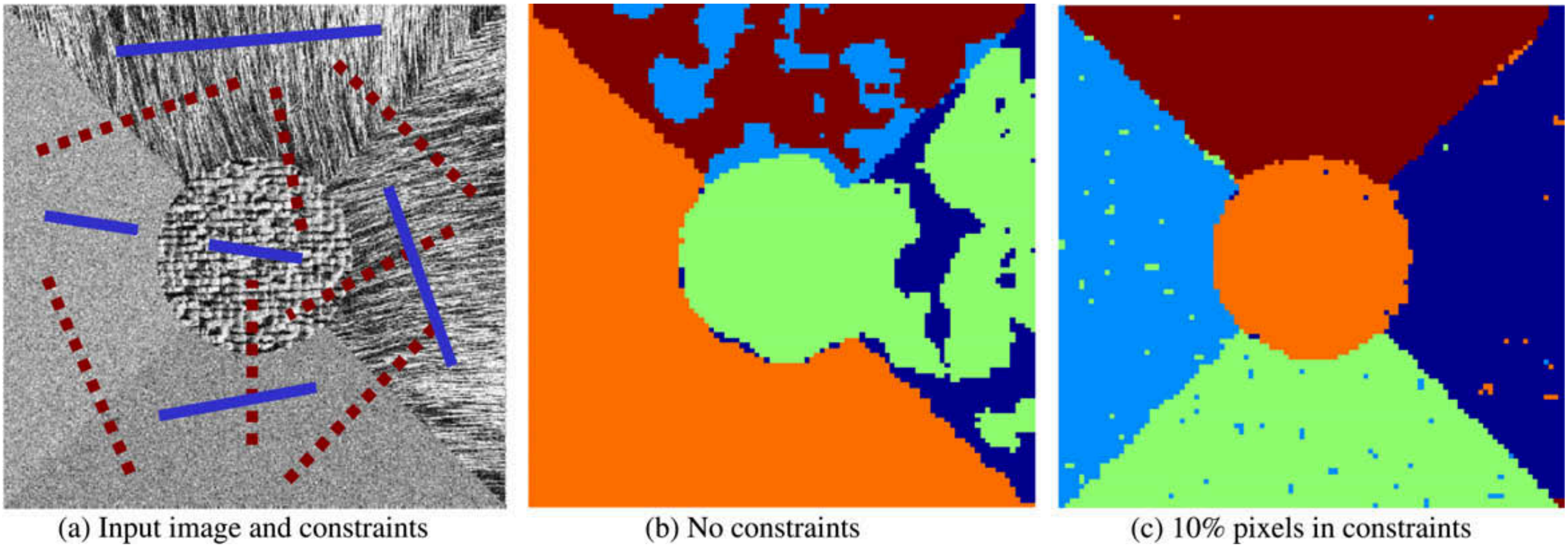
K-Means Clustering: Extensions



Learning problems: dots correspond to points without any labels. Points with labels are denoted by plus signs, asterisks, and crosses. In (c), the must-link and cannot link constraints are denoted by solid and dashed lines, respectively (figure taken from Lange et al. (2005).

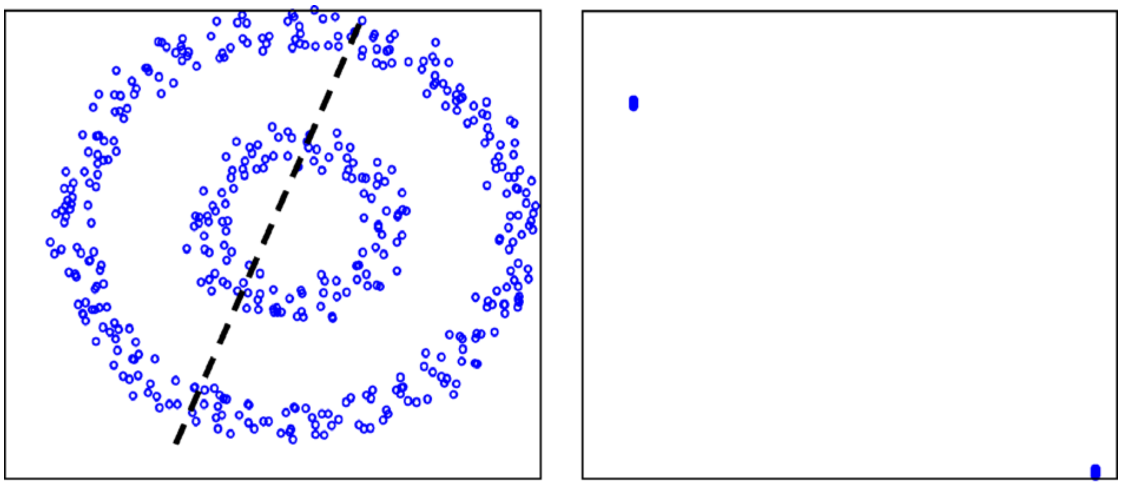
AK Jain, Data clustering: 50 years beyond K-means, PRL, 2010

Constrained K-Means: Semi-Supervised Clustering



Semi-supervised learning. (a) Input image consisting of five homogeneous textured regions; examples of must-link (solid blue lines) and must not link (broken red lines) constraints between pixels to be clustered are specified. (b) 5-Cluster solution (segmentation) without constraints. (c) Improved clustering (with five clusters) with 10% of the data points included in the pair-wise constraints (Lange et al., 2005).

K-Means Clustering: Extensions

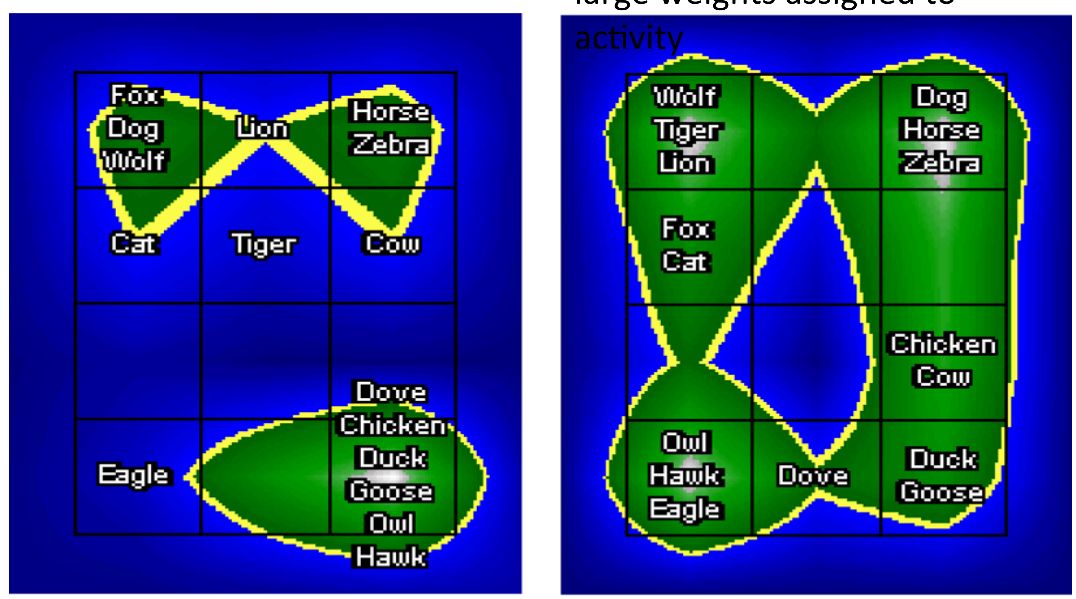


(a) (b)

Reprojecting the data using the top 2 eigenvectors of graph Laplacian of the data using RBF-kernel

Two “natural” clusters missed by K-Means with K=2

Weighted partitioning with large weights assigned to activity



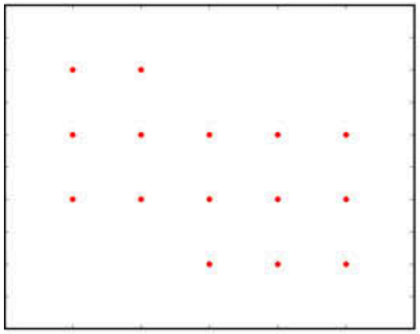
(a) (b)

Appearance vs Activity clustering of animals

Heat maps with color proportional to density

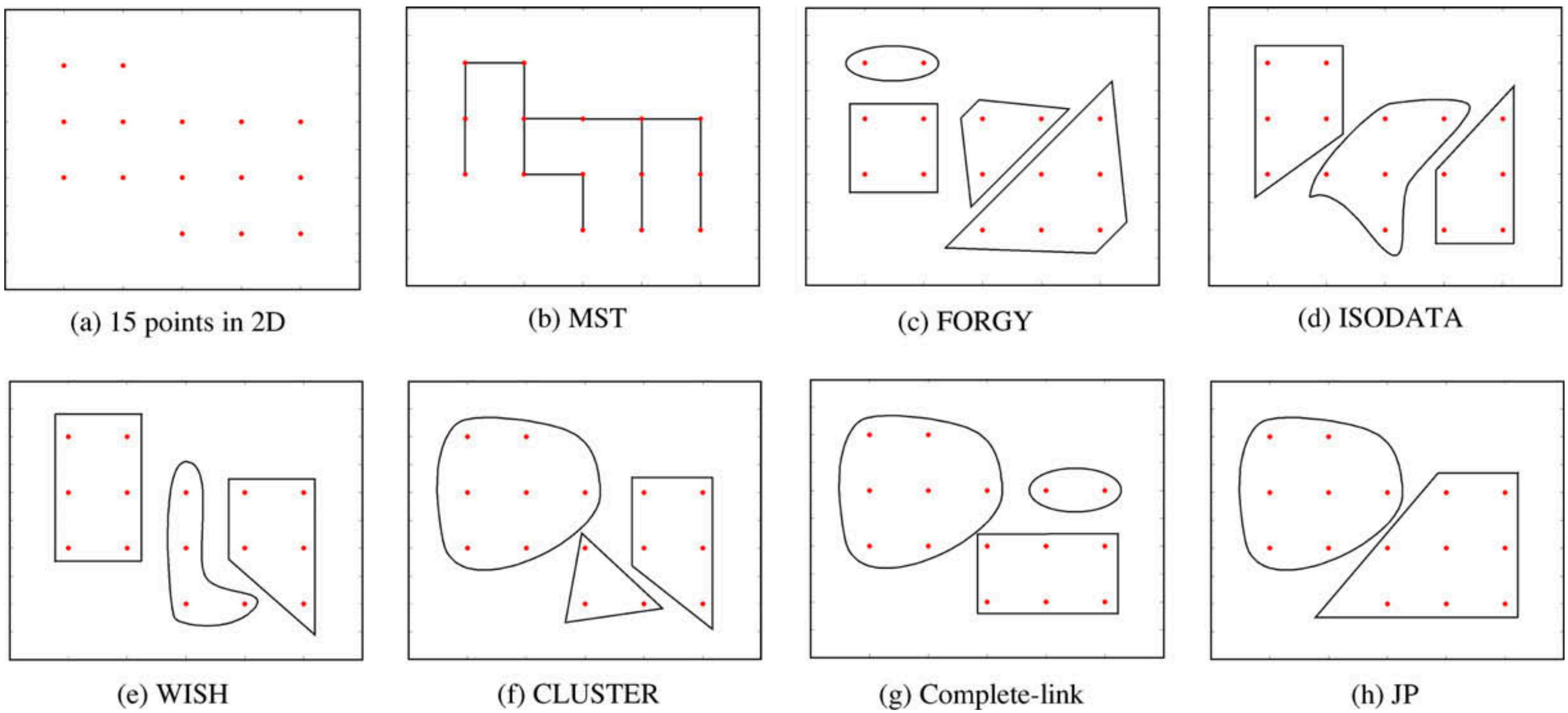
AK Jain, Data clustering: 50 years beyond K-means, PRL, 2010

How Many Clusters: 15 Points in 2D Using Different Clustering Algorithms



(a) 15 points in 2D

How Many Clusters: 15 Points in 2D Using Different Clustering Algorithms

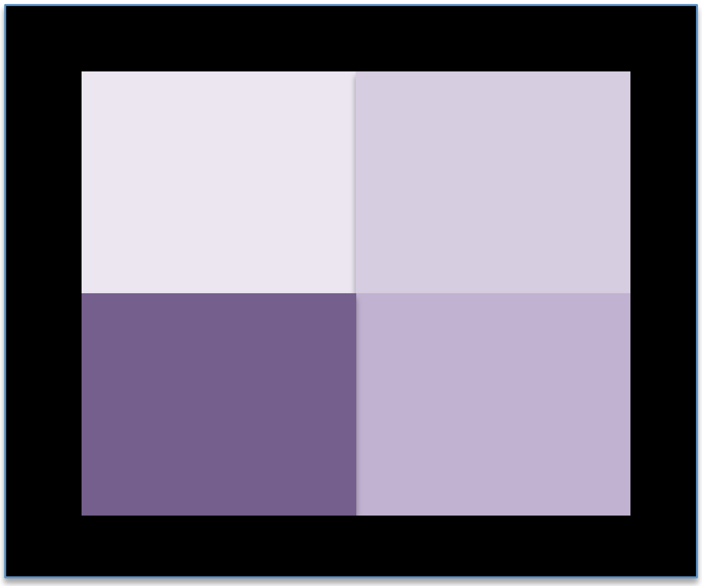


What will K-Means produce with K=2 or K=3?

(a) fifteen patterns; (b) minimum spanning tree of the fifteen patterns; (c) clusters from FORGY; (d) clusters from ISODATA; (e) clusters from WISH; (f) clusters from CLUSTER; (g) clusters from complete-link hierarchical clustering; and (h) clusters from Jarvis-Patrick clustering algorithm. (Dubes and Jain (1976))

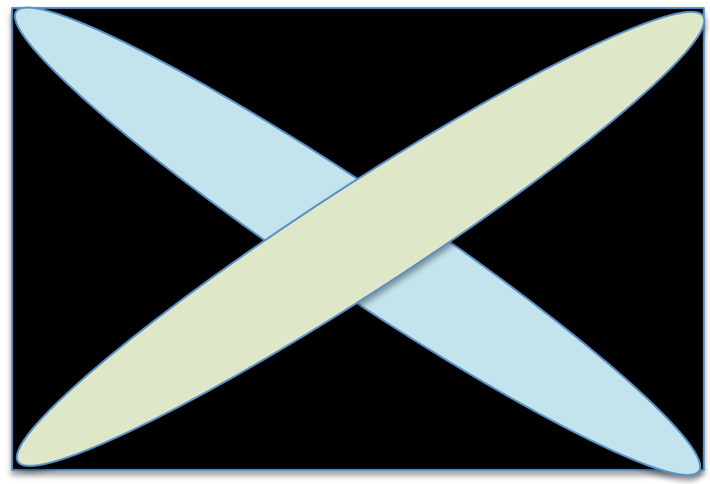
Test Cases & Edge Cases for K-Means

“Pancake”



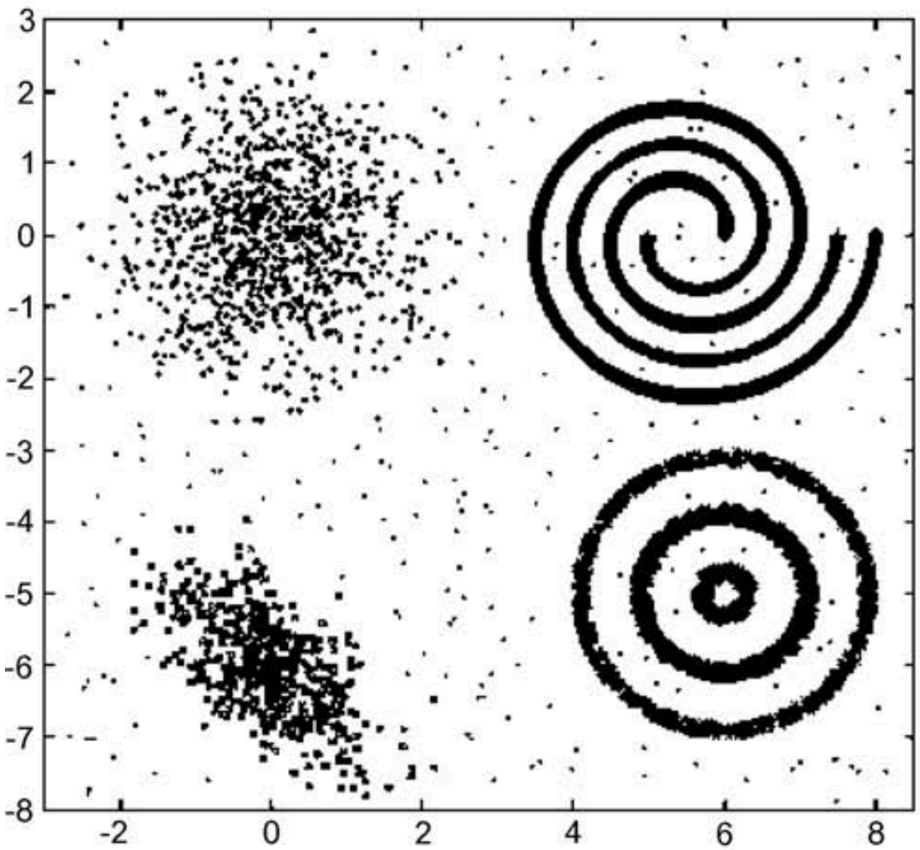
Uniform distribution of points
K=4
Where will be the cluster centers?

“Scissors” or X



Gaussian (elliptical) distribution of points
True K=2, with overlapping clusters
Where will the cluster centers be for K=2, 3, 4?
What will be the class labels (cluster membership) in the overlap region?

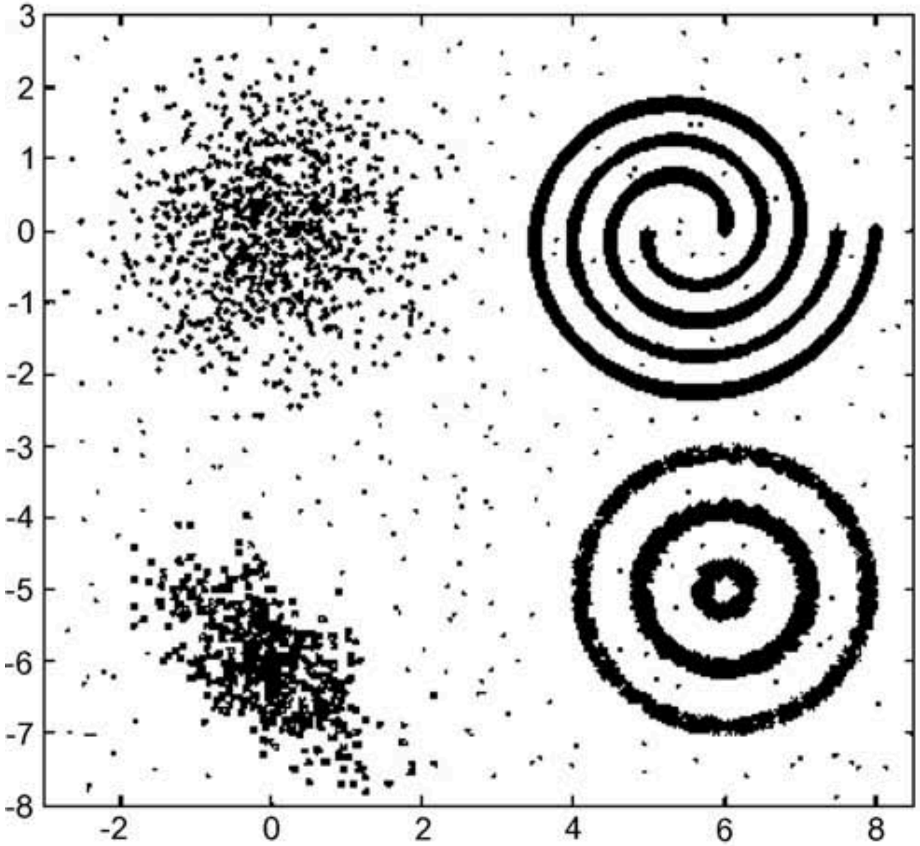
Test Cases & Edge Cases for K-Means



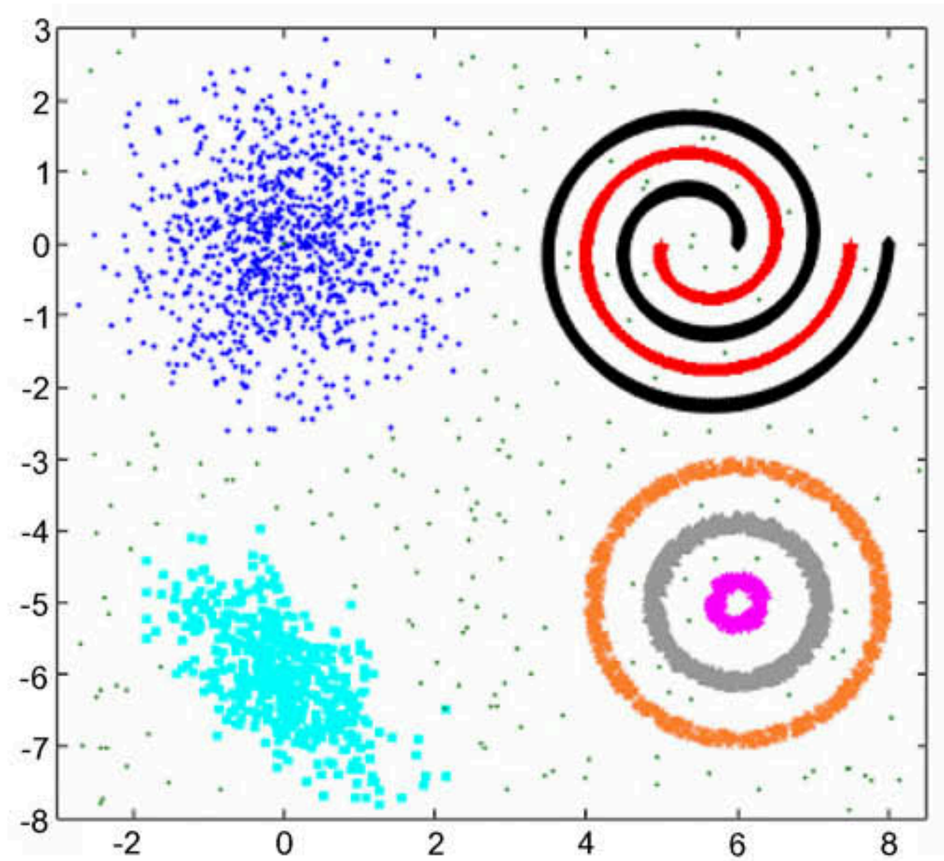
(a) Input data

AK Jain, Data clustering: 50 years beyond K-means, PRL, 2010

Test Cases & Edge Cases for K-Means



(a) Input data

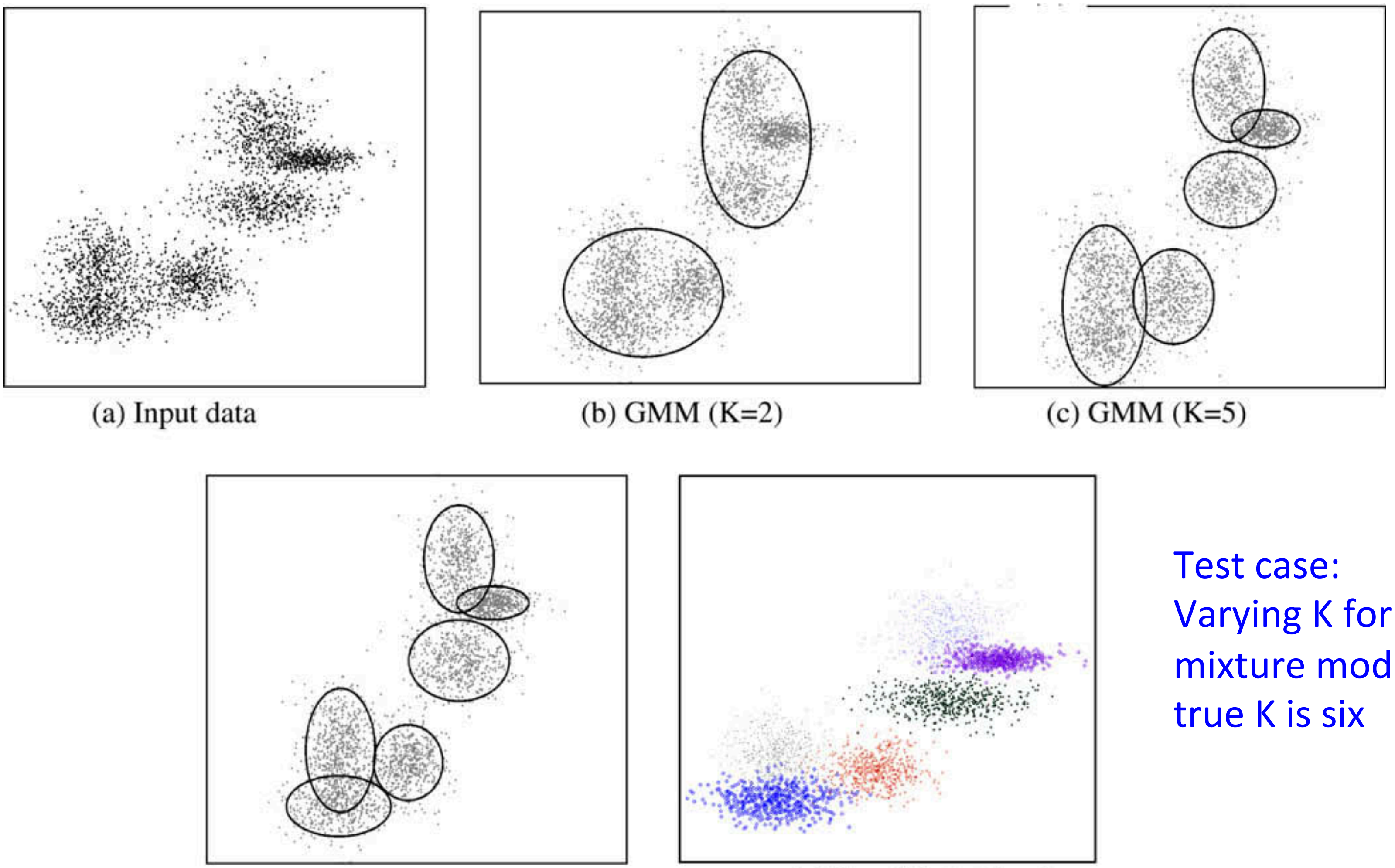


(b) Desired clustering

What will the K-means result be for K=7?

AK Jain, Data clustering: 50 years beyond K-means, PRL, 2010

Test Cases & Edge Cases for K-Means



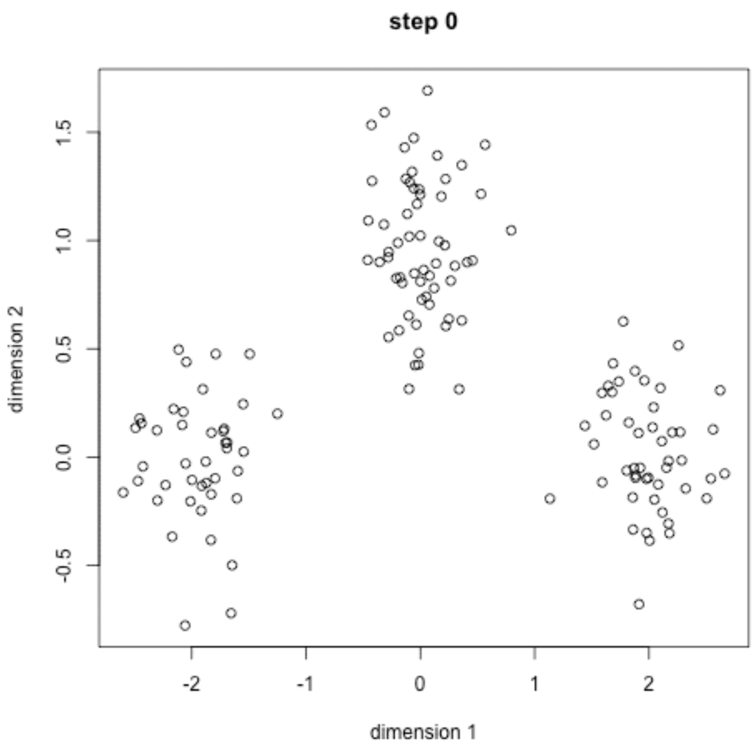
(a) Input data (b) GMM (K=2) (c) GMM (K=5)

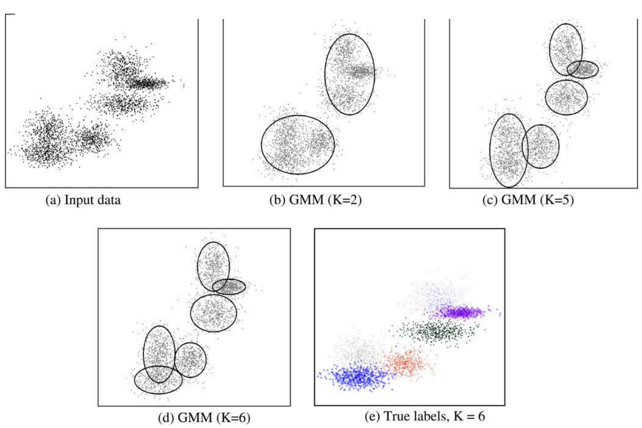
(d) GMM (K=6) (e) True labels, K = 6

Test case:
Varying K for a Gaussian mixture model when true K is six

AK Jain, Data clustering: 50 years beyond K-means, PRL, 2010

Edge Cases/Degenerate Conditions in K-Means





Degenerate case(s):

- K=0 (no centers)
- K=1 (one center)
- N=0 (no points)
- K>N (more centers than points)

-What if a cluster (center) has no data points

- N>>K, with

- N Identical points

-Termination condition

- How to break ties when two or more centers are equidistant

K-Means: Cluster Validity

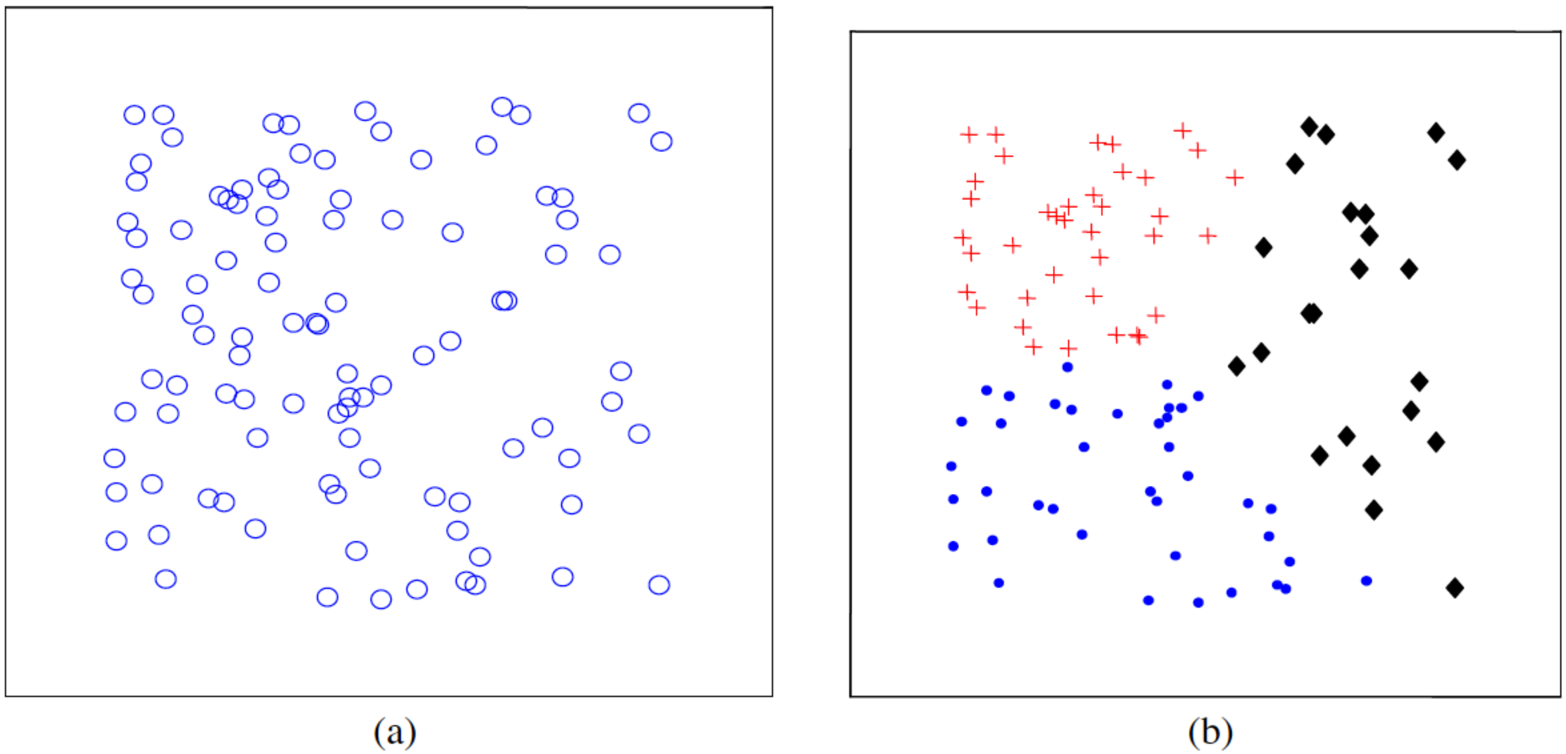


Fig. 8. Cluster validity. (a) A dataset with no "natural" clustering; (b) K-means partition with $K = 3$.

We do not know the true labels

AK Jain, Data clustering: 50 years beyond K-means, PRL, 2010

K-Means: Clustering Ensembles

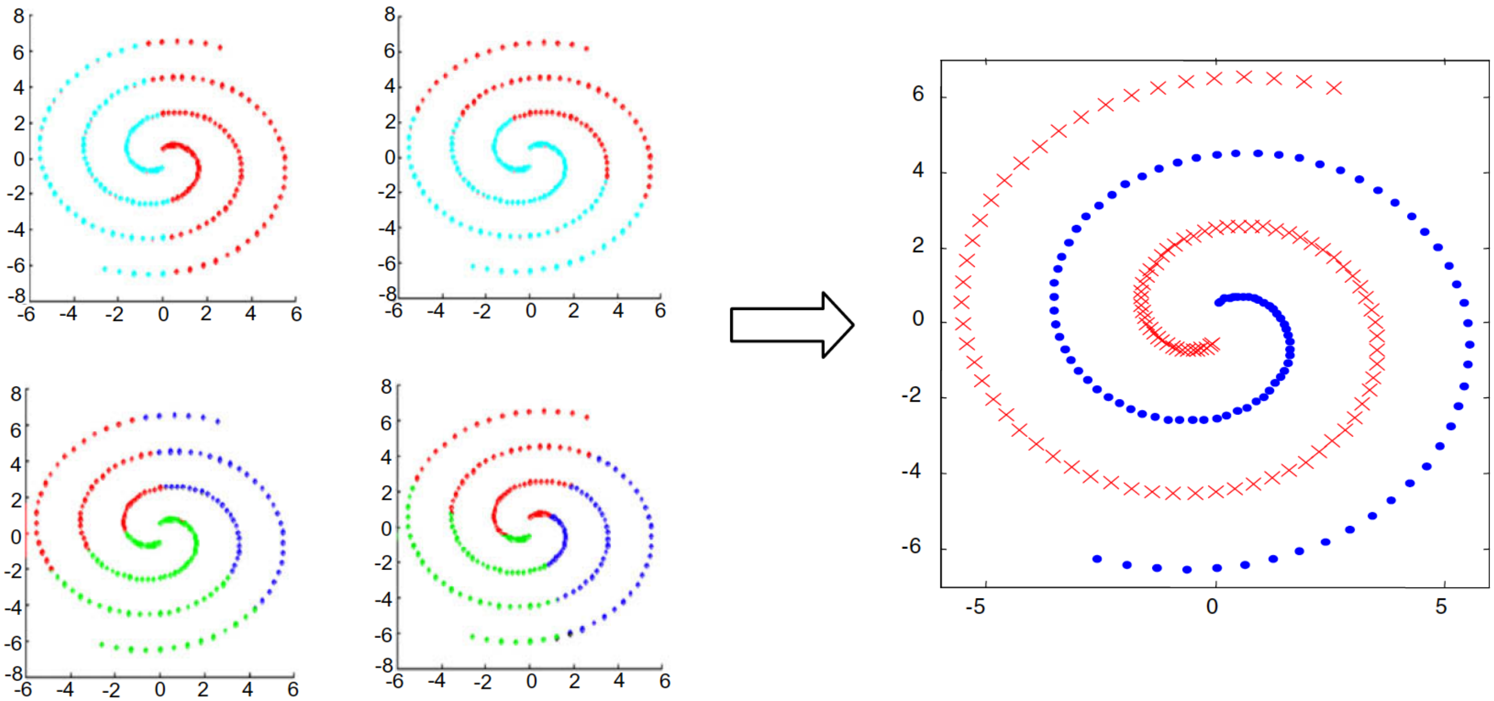
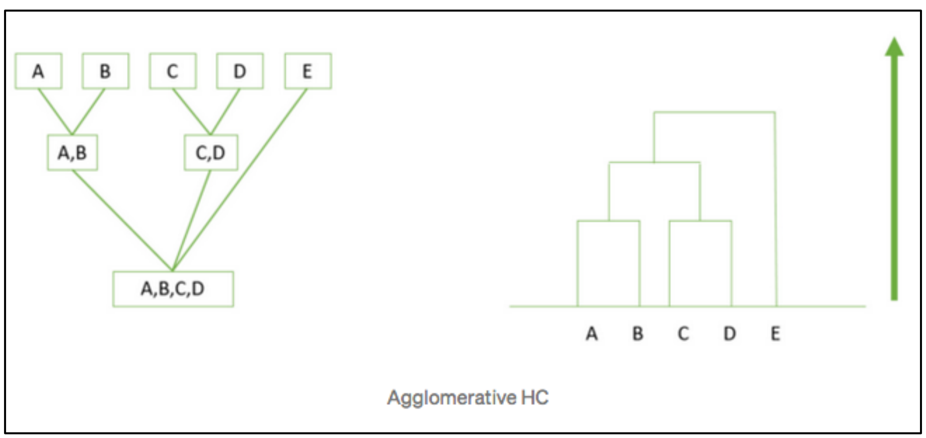


Fig. 11. Clustering ensembles. Multiple runs of K-means are used to learn the pair-wise similarity using the "co-occurrence" of points in clusters. This similarity can be used to detect arbitrary shaped clusters.

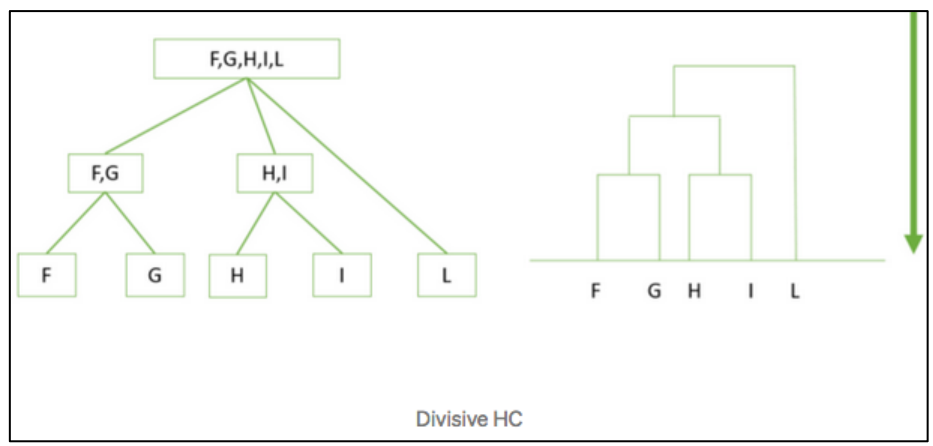
AK Jain, Data clustering: 50 years beyond K-means, PRL, 2010

Hierarchical Clustering

- Agglomerative (bottom-up from N to 1) or divisive (top-down from 1 to N)
- Nested clusters organized as a tree
- Interactively select K by cutting the dendrogram



Agglomerative HC

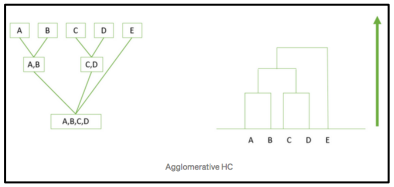


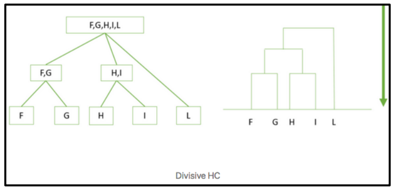
Divisive HC

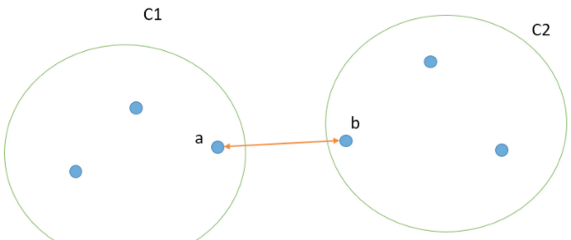
https://towardsdatascience.com/

Hierarchical Clustering

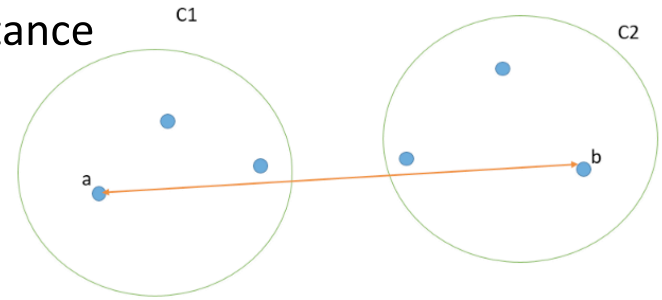
- Similarity between clusters: Minimum distance between any two members in each cluster, max or average



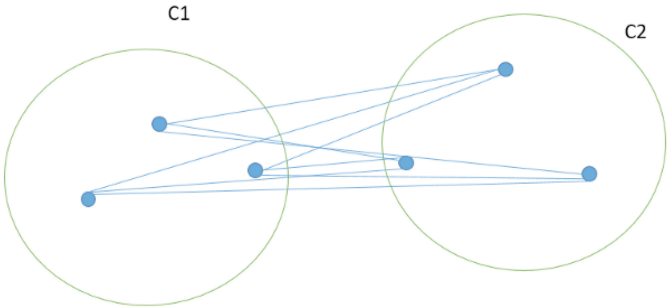




Min inter-cluster distance



Max inter-cluster distance



Average inter-cluster distance

https://towardsdatascience.com/

K-Means Clustering: Careful Seeding

k-means++: The Advantages of Careful Seeding
David Arthur and Sergei Vassilvitskii, 2006

For the `k-means` problem, we are given an integer k and a set of n data points $\mathcal{X} \subset \mathbb{R}^d$. We wish to choose k centers $\mathcal{C}$ so as to minimize the potential function,

$$\phi = \sum_{x \in \mathcal{X}} \min_{c \in \mathcal{C}} \|x - c\|^2.$$

Minimize sum of the squared errors over all K clusters (intra-class distances)

The `k-means` algorithm is a simple and fast algorithm for this problem, although it offers no approximation guarantees at all.

1. Arbitrarily choose an initial k centers $\mathcal{C} = \{c_1, c_2, \cdots, c_k\}$.
2. For each $i \in \{1, \ldots, k\}$, set the cluster C_i to be the set of points in $\mathcal{X}$ that are closer to c_i than they are to c_j for all $j \neq i$.
3. For each $i \in \{1, \ldots, k\}$, set c_i to be the center of mass of all points in C_i: $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$.
4. Repeat Steps 2 and 3 until $\mathcal{C}$ no longer changes.

It is standard practice to choose the initial centers uniformly at random from $\mathcal{X}$. For Step 2, ties may be broken arbitrarily, as long as the method is consistent.

The idea here is that Steps 2 and 3 are both guaranteed to decrease ϕ, so the algorithm makes local improvements to an arbitrary clustering until it is no longer possible to do so. To see Step 3 decreases ϕ, it is helpful to recall a standard result from linear algebra (see for example [2]).

K-Means++ Clustering: Careful Seeding

We propose a specific way of choosing centers for the `k-means` algorithm. In particular, let $D(x)$ denote the shortest distance from a data point to the closest center we have already chosen. Then, we define the following algorithm, which we call `k-means++`.

1a. Take one center c_1, chosen uniformly at random from $\mathcal{X}$.
1b. Take a new center c_i, choosing $x \in \mathcal{X}$ with probability $\frac{D(x)^2}{\sum_{x \in \mathcal{X}} D(x)^2}$.
1c. Repeat Step 1b. until we have taken k centers altogether.
2-4. Proceed as with the standard `k-means` algorithm.

We call the weighting used in Step 1b simply "D^2 weighting".

How do you sample from the pdf of squared distances?

K-Means++ Clustering: Empirical Results (2006)

k	Average ϕ		Minimum ϕ		Average T	
	k-means	k-means++	k-means	k-means++	k-means	k-means++
10	10898	5.122	2526.9	5.122	0.48	0.05
25	787.992	4.46809	4.40205	4.41158	1.34	1.59
50	3.47662	3.35897	3.40053	3.26072	2.67	2.84

Table 1: Experimental results on the *Norm-10* dataset (n = 10000, d = 5)

Synthetic dataset Norm-10: 10 “real” centers uniformly from hypercube of side length 500. Add point from Gaussian with variance 1, centered at each center for well-separated “ideal” clusters.

20 trials for each case
No special optimizations for timing

K-Means++ Clustering: Empirical Results (2006)

Synthetic dataset Norm-25: 25 “real” centers uniformly from hypercube of side length 500. Add point from Gaussian with variance 1, centered at each center for well-separated “ideal” clusters.

	Average ϕ		Minimum ϕ		Average T	
k	k-means	k-means++	k-means	k-means++	k-means	k-means++
10	135512	126433	119201	111611	0.14	0.13
25	48050.5	15.8313	25734.6	15.8313	1.69	0.26
50	5466.02	14.76	14.79	14.73	3.79	4.21

Table 2: Experimental results on the *Norm-25* dataset (n = 10000, d = 15)

1024 points in 10-D, cloud cover database from UCI Machine Learning Repository

	Average ϕ		Minimum ϕ		Average T	
k	k-means	k-means++	k-means	k-means++	k-means	k-means++
10	7553.5	6151.2	6139.45	5631.99	0.12	0.05
25	3626.1	2064.9	2568.2	1988.76	0.19	0.09
50	2004.2	1133.7	1344	1088	0.27	0.17

Table 3: Experimental results on the *Cloud* dataset (n = 1024, d = 10)

Potential values by k-means++ is 10 to 1000 times better (ie lower energy)

Speed up to 2.7 times faster

	Average ϕ		Minimum ϕ		Average T	
k	k-means	k-means++	k-means	k-means++	k-means	k-means++
10	$3.45 \cdot 10^8$	$2.31 \cdot 10^7$	$3.25 \cdot 10^8$	$1.79 \cdot 10^7$	107.5	64.04
25	$3.15 \cdot 10^8$	$2.53 \cdot 10^6$	$3.1 \cdot 10^8$	$2.06 \cdot 10^6$	421.5	313.65
50	$3.08 \cdot 10^8$	$4.67 \cdot 10^5$	$3.08 \cdot 10^8$	$3.98 \cdot 10^5$	766.2	282.9

Table 4: Experimental results on the *Intrusion* dataset (n = 494019, d = 35)

Stage 2: Challenges With/Speeding Up K-Means Clustering & Writing Good Code

- Code optimizations for feature distance computations
  - Features could be binary (Hamming) or Euclidean or mixed
  - General distance function (template) implementation
- How do you select K?
- Initial set of centers selected – effects convergence rate
  - Seed points are observations in the dataset
  - Seed points are any random feature values (ie may not be an observation)
  - Distance-based selection
  - Density-based selection
- Most expensive computation – distances/similarities vs metrics and triangle inequality
  - Effect of Euclidean (L2-norm) vs other Lp-norm (Taxicab L1)
- Parallelization of the distance computations
  - Multiple threads, GPU
  - Parallelize on the number of points/observations
- Reproducibility
- Profiling code
- Generalization of algorithm
  - Distance vs ordering/inequality operator
- Subsample the set of observations when N is very large (like one billion) to get a fast initial clustering and energy value

Assignment#4
K-Means Clustering Algorithm

- Part 1: 2D interactive K-Means clustering
  - Euclidean (L2) feature distance computation
  - Unique membership label (ie each point belongs to one and only one class)
  - Let the user specify K
  - For equidistant points break ties randomly
  - Visualize the evolution of the class membership of each point using colors for each iteration of K-Means loop
  - Draw the updated centroid (mean) of each cluster
  - Animate/iterate until the class labels do not change or the maximum iterations has been reached
  - Use simple test cases: 15 dots in 2D (K=2 or K=3), 2D GMM (K=6)
  - Let the user click on a cluster mean and recluster all of these observations into K_2 clusters (where K_2 can be different from K)
  - Compare the speed of K-Means with random initialization vs K-Means++ initialization (D^2 weighting)
  - When D>2 (or D>3) randomly select the 2 features to project and display the clustering visualization OR allow the user to select the two (or three) axes for display
  - Check for degenerate conditions

- Part 2: 3D interactive K-Means clustering

- Optional Part 3: nD interactive K-Means clustering

K-Means Animation