Unsupervised Learning: Clustering

Clustering

Objective:

- Definition: Group or segment the data set (a collection of objects) into subsets so that those within each subset are more closely related to others than those objects assigned to other subsets.
- Each group (subset) is called a cluster.

Challenging:

- What is a meaningful cluster?
- How do we validate clustering results?

Clustering Challenges

What are meaningful clusters?

How many clusters?

Six Clusters

Two Clusters

Four Clusters

Clustering Concepts:

- Hard vs. Soft Clustering.
- Model-Based vs. Algorithmic.
- Flat vs. Nested.
- Clustering observations (most common) vs. Clustering features vs. Clustering both (Biclustering).

Proximity and Dissimilarity Matrices

Clustering results are crucially dependent on the measure of dissimilarity (or distance) between the "points" to be clustered.

- Proximity Matrix: $n \times n$ with the ij-th element d_{ij} measuring the proximity between the i-th and the jth objects (or observations). D is typically symmetric.
- One can use a dissimilarity matrix instead.
- Dissimilarity between x_i and $\mathrm{x}_{i'}$:

$$D(\mathrm{x}_i, \mathrm{x}_{i'}) = \sum_{j=1}^{p} d_j(x_{ij}, x_{i'j}).$$

- A weighted version:

$$D(\mathrm{x}_i, \mathrm{x}_{i'}) = \sum_{j=1}^{p} w_j d_j(x_{ij}, x_{i'j}); \sum_{j=1}^{p} w_j = 1.$$

Types of Distances:

- Squared distance: $d_j(x_{ij}, x_{i'j}) = (x_{ij} - x_{i'j})^2$.
- A more general distance: $d(\mathrm{x}_i, \mathrm{x}_{i'}) = l(|\,\mathrm{x}_i - \mathrm{x}_{i'}\,|)$.
- Correlation:

$$\rho(\mathrm{x}_i, \mathrm{x}_{i'}) = \frac{\sum_j (x_{ij} - \bar{x}_i)(x_{i'j} - \bar{x}_{i'})}{\sqrt{\sum_j (x_{ij} - \bar{x}_i)^2 \sum_j (x_{i'j} - \bar{x}_{i'})^2}}$$

- If inputs are standardized, $\sum_j (x_{ij} - x_{i'j})^2 \propto 2(1 - \rho(\mathrm{x}_i, \mathrm{x}_{i'}))$: clustering based on correlation (similarity) is equivalent to that based on squared distance (dissimilarity).
- Others? For Discrete Data? (`dist()` in `R`).

Clustering Algorithms:

- K-means.
  - Combinatorial algorithms.
  - NMF for soft-clustering.
  - Model-based soft-clustering.
- Hierarchical Clustering.
  - Biclustering - Cluster-Heatmap.
- Convex Clustering & Convex Biclustering.
- Spectral Clustering.

Combinatorial Algorithms

- Each observation is uniquely labeled by an integer $i \in \{1, 2, \ldots, n\}$.
- k clusters: $k \in \{1, \ldots, K\}$.
- Let $k = C(i)$ denote the ith observation get assigned to the k-th cluster.
- $d(x_i, x_{i'})$: dissimilarity between x_i and $x_{i'}$.
- Goal: search C^* such that $W(C)$ (within cluster dissimilarity) is minimized:

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} d(x_i, x_{i'}).$$

Combinatorial Algorithms

- Total Dissimilarity:

$$T = \frac{1}{2}\sum_{i=1}^{n}\sum_{i'=1}^{n} d_{ii'} = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\Big(\sum_{C(i')=k} d_{ii'} + \sum_{C(i')\neq k} d_{ii'}\Big).$$

-

$$B(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')\neq k} d_{ii'}.$$

- $W(C) = T - B(C)$.
- Minimizing $W(C)$ is equivalent to maximizing $B(C)$.

Combinatorial Algorithms

- One needs to minimize W over all possible assignments of n points to K clusters.
- The number of distinct assignments is

$$S(n, K) = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^n.$$

- It is not feasible for large n and K.
- It calls for more efficient algorithms: may not be optimal but a reasonably good suboptimal partition.

K-Means Clustering

K-means

- One of the most popular iterative descent clustering methods.
- Tries to find a fast, local solution to the combinatorial clustering problem.
- For the case that all variables are quantitative.
- Dissimilarity measure: Squared Euclidean distance

$$d(\mathrm{x}_i, \mathrm{x}_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = ||\,\mathrm{x}_i - \mathrm{x}_{i'}\,||^2.$$

K-means

Objective: Minimize $W(C)$.

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{i\neq i', C(i')=k} ||\, x_i - x_{i'}\,||_2^2$$

$$= \sum_{k=1}^{K} n_k \sum_{C(i)=k} ||\, x_i - \bar{x}_k||_2^2$$

where $\bar{x}_k = \frac{1}{n_k}\sum_{C(i)=k} x_i$.

K-means

Idea:

- Augment $W(C)$ with cluster means, m_k:

$$W(C, \mathrm{m}_k) = \sum_{k=1}^{K} n_k \sum_{C(i)=k} ||\mathrm{x}_i - \mathrm{m}_k||_2^2$$

- Minimize $W(C, \mathrm{m}_k)$ by iteratively optimizing:
  1. Cluster means: m_k (with $C(i)$ fixed).
  2. Cluster assignments: $C(i)$ (with m_k fixed).
- Subproblems are easy to solve!

K-means Algorithm

- Initialize each observation i to a cluster assignment k.
- Repeat until cluster assignments are unchanged:

1. Find cluster means. (cluster assignments fixed)

$$\hat{m}_k = \text{argmin}_m \sum_{C(i)=k} ||x_i - m_k||_2^2$$

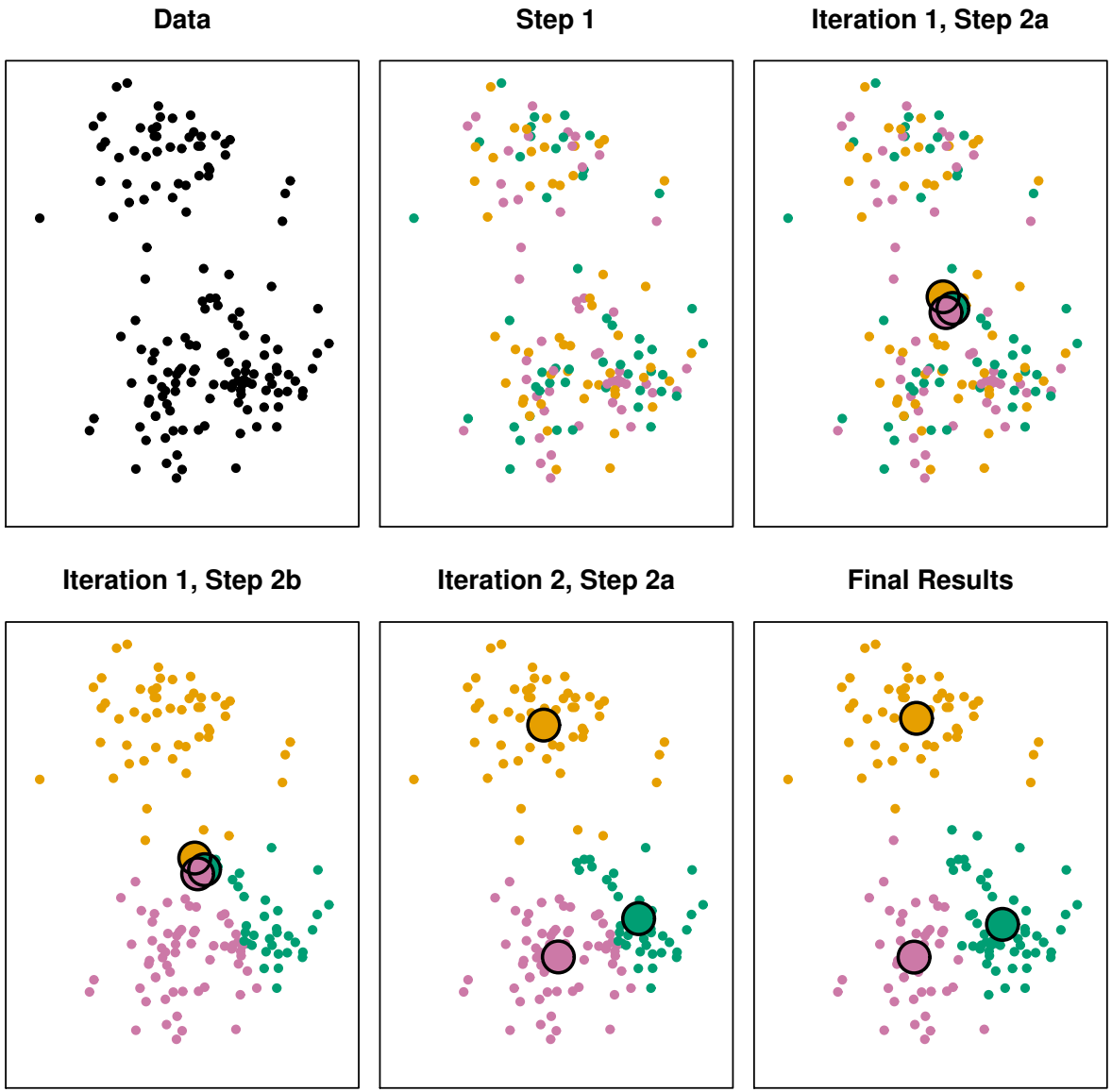
$\Longrightarrow$ Take mean of points in cluster.

2. Find cluster assignments. (cluster means fixed)

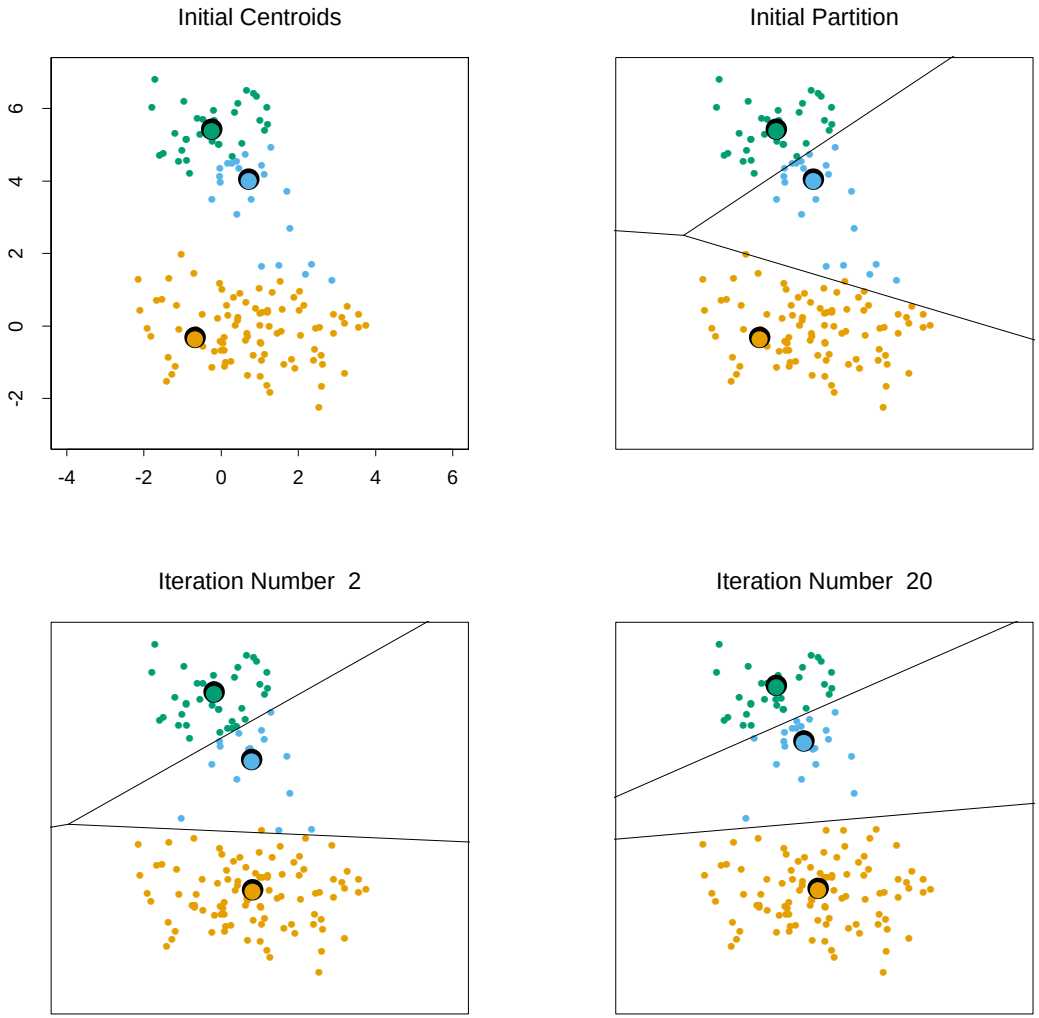
$$\hat{C}(i) = \text{argmin}_k \; ||x_i - m_k||_2^2$$

$\Longrightarrow$ Assign each observation to the closest cluster mean.

K-means Algorithm



K-means Algorithm



K-means Properties

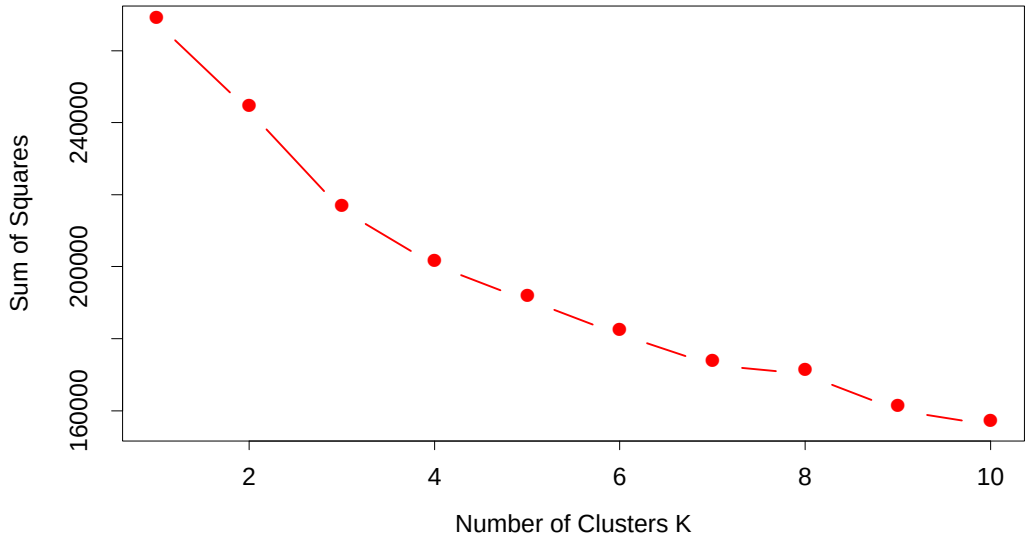
- Steps 1 and 2 decrease $W(C)$.
- Local solution - not necessarily global solution.
- Depends on starting values (initialization).
- K needs to be set before.
- Best for compact, spherical clusters.
- Does not work well when cluster sizes are different.

K-means in `R`: `kmeans`

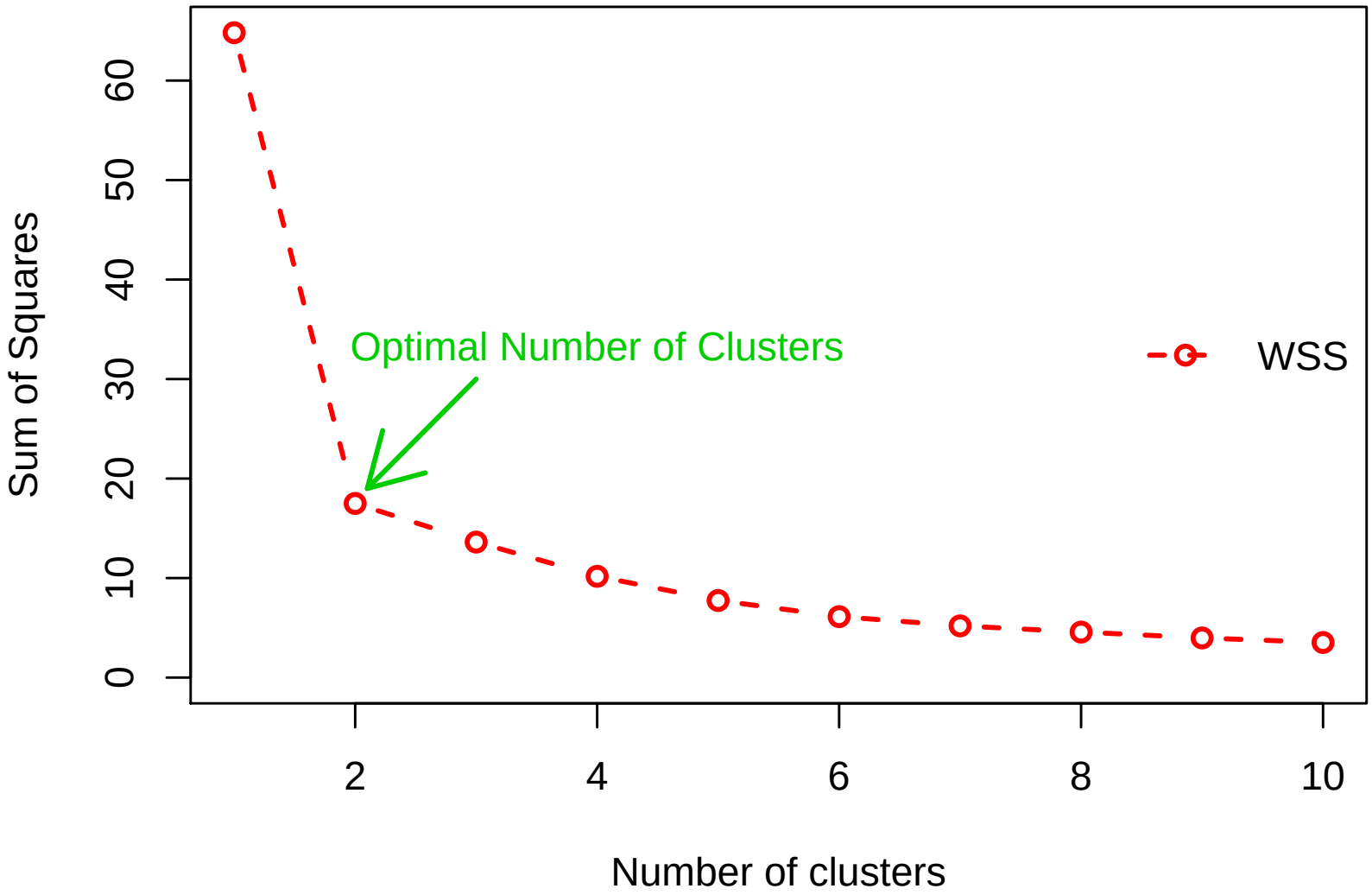
K-means - Initializations

How to Choose K?

- Can we choose K that minimizes $W(C)$?
- Can we choose K by a Validation set? Cross-Validation?



How to Choose K?



How to Choose K?

- Gap Statistic.
- Silhouette Statistic.
- Cluster Prediction Strength.
- Cluster Stability / Consensus Clustering.

How to Choose K?

Gap Statistic:

- Idea: Choose K that yields most grouped data compared to random data.
- Random uniform points over data domain.
- Cluster random points with K-means.
- Choose K that gives biggest "Gap" (difference) between random $W(C)^*$ and observed $W(C)$.
- Issues with this?
- `R`: `clusGap` in `cluster` package.

How to Choose K?

Silhouette Statistic:

- a_i - mean within-cluster dissimilarity with observation i.
- b_i - mean between-cluster dissimilarity with observation i.
- Silhouette Statistic:

$$S_i = \frac{b_i - a_i}{\max(b_i, a_i)}.$$

- S_i close to 1 = good clustering.
- S_i close to -1 = bad clustering.
- Choose K that maximizes average S_i.
- R: `silhouette` in `cluster` package.

How to Choose K?

Prediction Strength:

- Choose K where clusters have most overlap between many training and test set splits.

Metrics to measure overlap between cluster assignments:

- Rand Index.
  - R: `rand_indep` in `clusteval` package.
- Jaccard Index.
  - R: `jaccard_indep` in `clusteval` package.

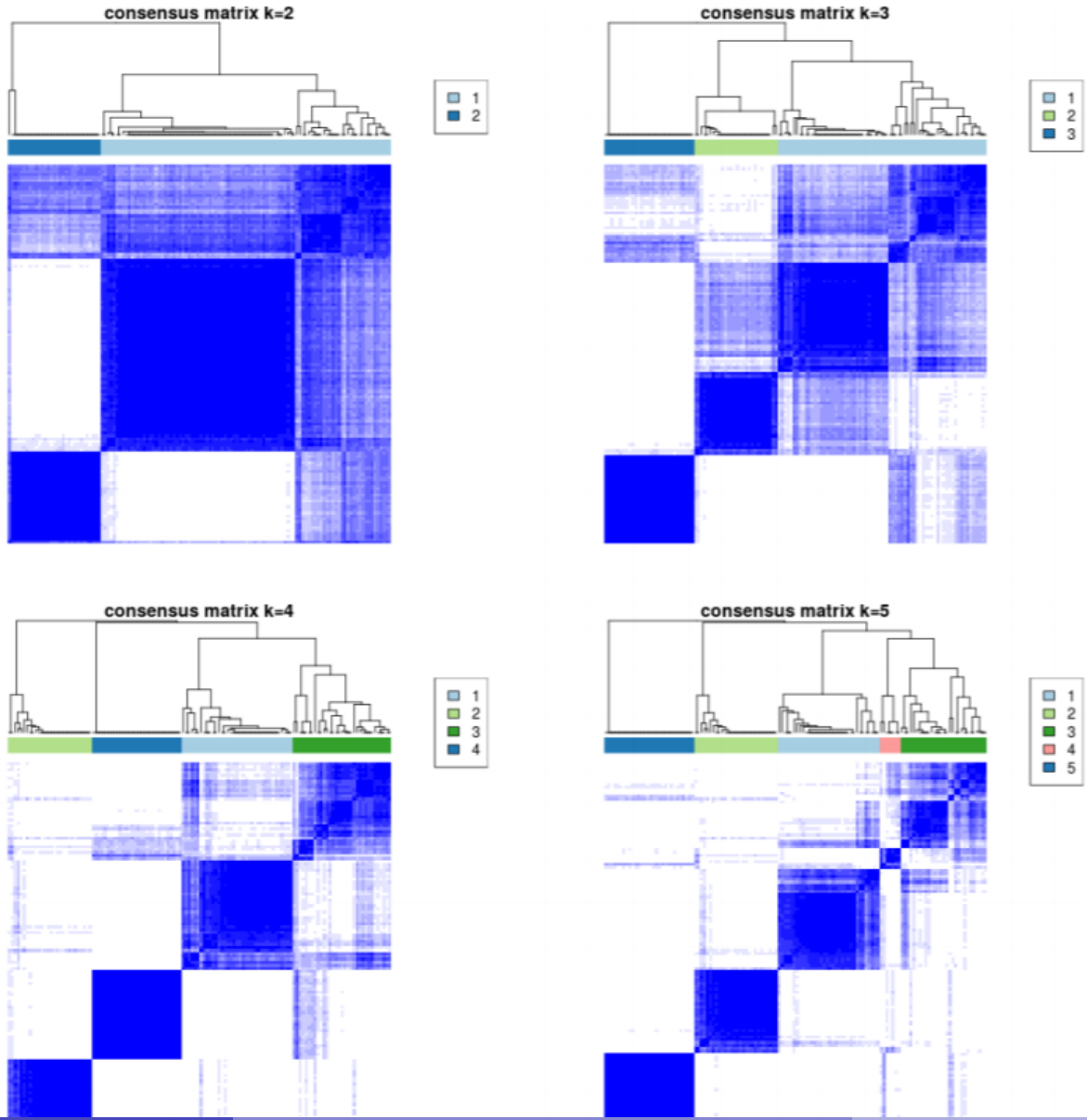
How to Choose K?

Cluster Stability / Consensus Clustering:

- Repeatedly perturb data (typically subsample observations).
- Cluster each perturbed data set.
- For each i and j, consensus = % of times where observation i and j are in same cluster.
- Visualize as a $n \times n$ Consensus Matrix.
- Choose K where cluster assignments are most stable over perturbations.
- R: `ConsensusClusterPlus` package.

How to Choose K?

Cluster Stability / Consensus Clustering:



Summary - K-means

Strengths:

- Fast.
- Simple.
- Others?

Weaknesses:

- Local solution - highly depends on initialization.
  - Kmeans++ a good initialization.
  - In `R`: `kmeanspp`.
- High-dimensional settings? ($p >> n$ - more features than observations)
  - Apply PCA to reduce dimension before using K-means.
- Others?

K-means - Related Algorithms

Sparse K-Means

- For high dimensional data, it can be useful to identify subset of variables for clustering.
- This can be very useful when "sparsity" exists: only a subset of variables relevant for the clusters.
- Sparse K-means algorithm can perform K-means and identify the sparse set simultaneously.
- Sparse K-means can be achieved by weighting the variables together with sparsity regularization.
- Computation is more challenging. Multiple packages availabe in R.

One example: `vimpclust` package in `R`.

K-Medoids

- K-means is a natural choice when the distances are Euclidean and means correspond to sensible points in the space (e.g. clustering of images).
- Sometime we want to use more general distances, or that means are not sensible in our space.
- The main difference is that K-Medoids requires the cluster centers to be in the data set: these special data can be viewed as "exemplars" for their clusters.
- K-Medoids is typically more computationally intensive to fit, but can be more interpretable because the clusters are represented by data examples.

`kmed` package in `R`.

Soft Clustering: Mixture Models

- Mixture of k distributions.
- Assign each observation a probability of arising from distribution k.
- Most Common: Gaussian Mixture model.
- Algorithm: EM (Expectation-Maximization).
  - E-step: Cluster probabilities for each observation.
  - M-Step: Given soft-cluster assignments, maximize likelihood for each distribution.

`mclust` package in `R`.

Soft-Clustering: NMF

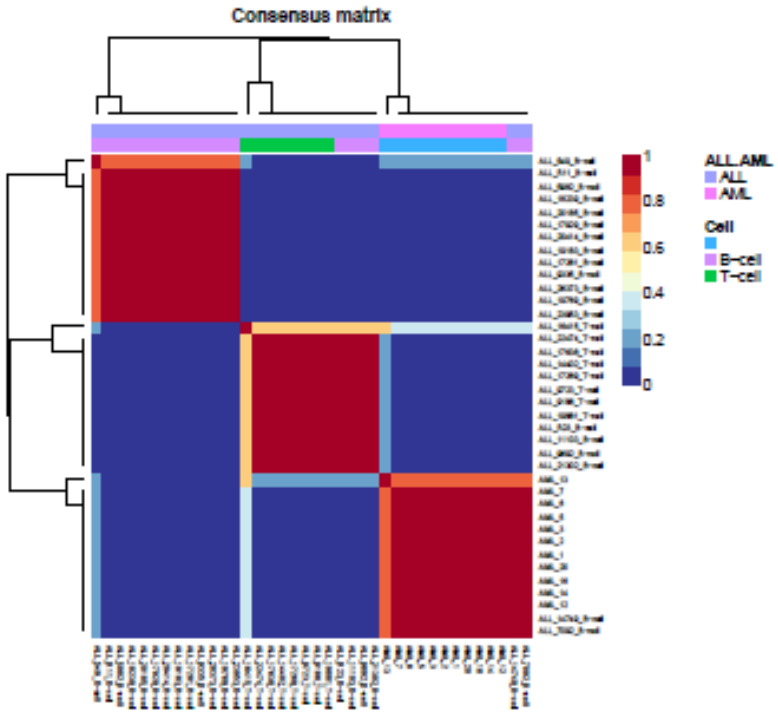
$$X_{n \times p} \approx W_{n \times K} H_{K \times p}$$

- Clusters: Each column of W.
- Soft-Cluster Assignments: $W_k^T = (.4, 1, 0, 0, 2.1, 0)$.
- Observations can be assigned non-zero weights to more than one cluster.
- Hard-Cluster Assignment: Cluster of i defined as argmax of W_i.
- Features contributing to cluster k: Rows of H_k.

`NMF` package in `R`.

Soft Clustering - NMF

Consensus NMF Clustering:



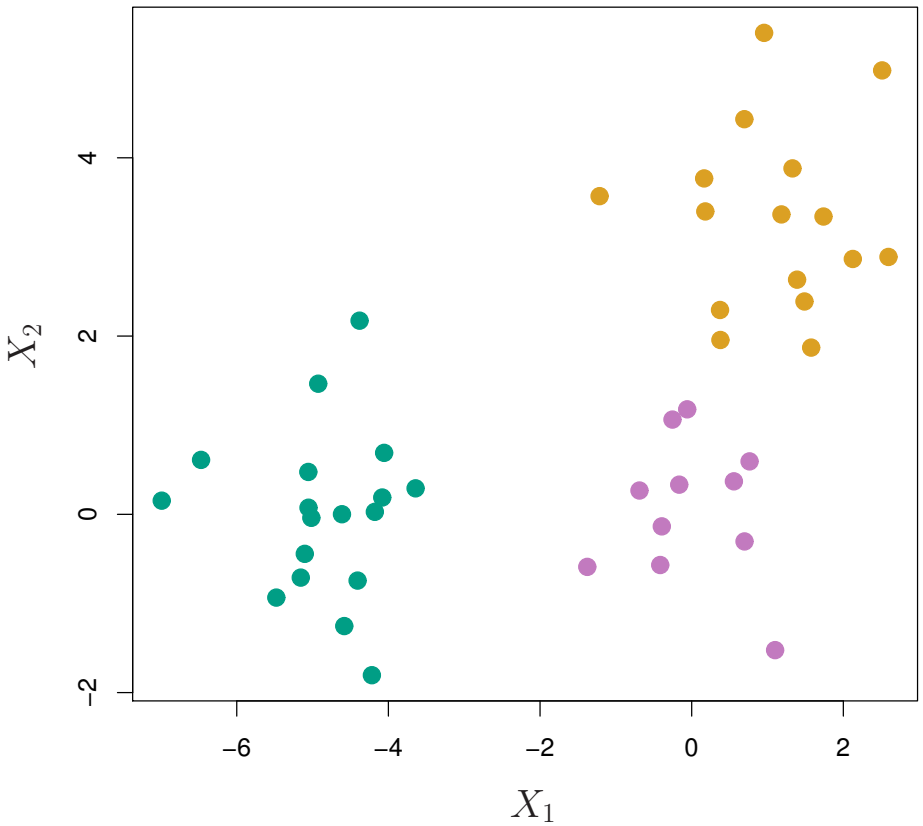
- Consensus = % time observation i and j in same cluster.

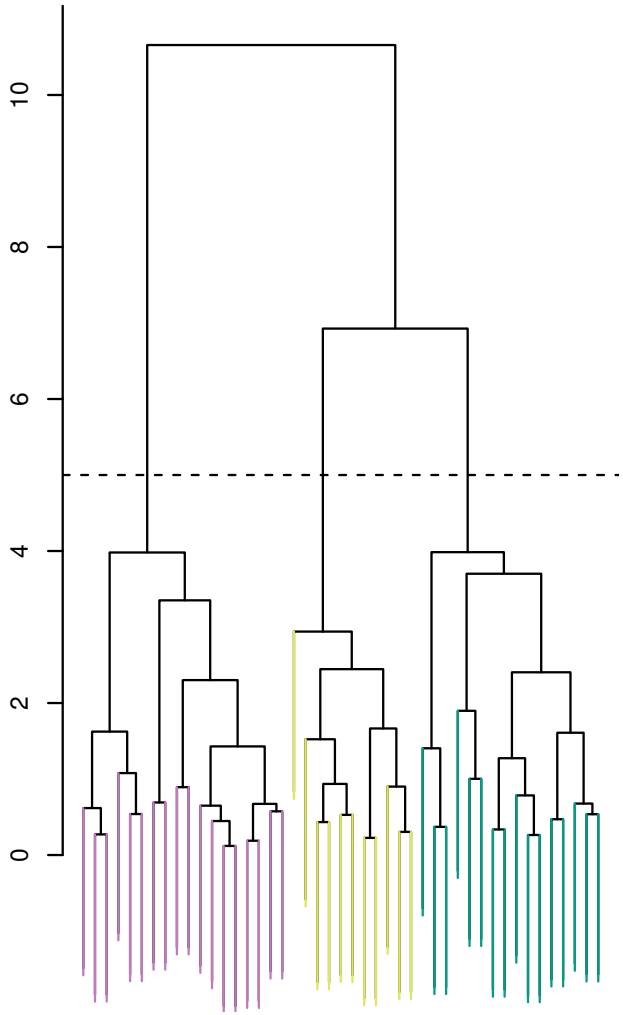
Hierarchical Clustering

Hierarchical Clustering

- Nested Clusters: Produce hierarchical representations in which the clusters at each level of the hierarchy are created by merging clusters at the next lower level.
- At the lowest level, each cluster contains a single observation.
- At the highest level there is only one cluster containing all observations.
- Two paradigms: agglomerative (bottom-up; most popular) and divisive (top-down; less popular).
- Use dendrogram to display the clustering result.

Interpreting a Dendrogram





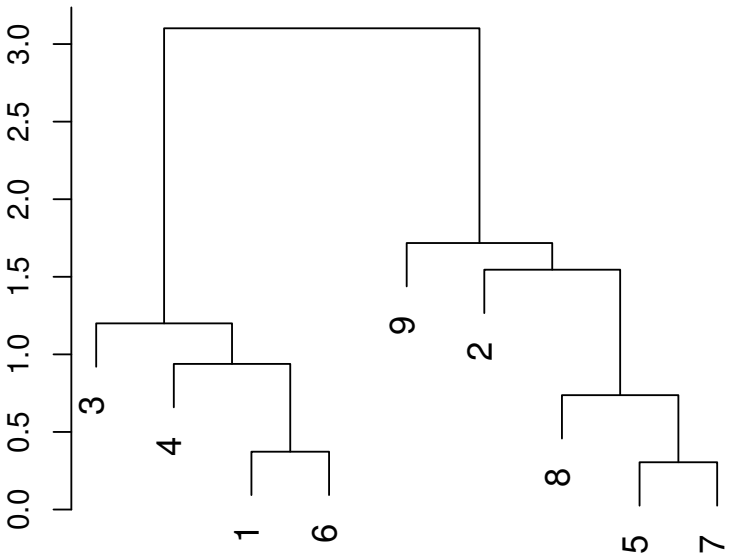
Interpreting a Dendrogram

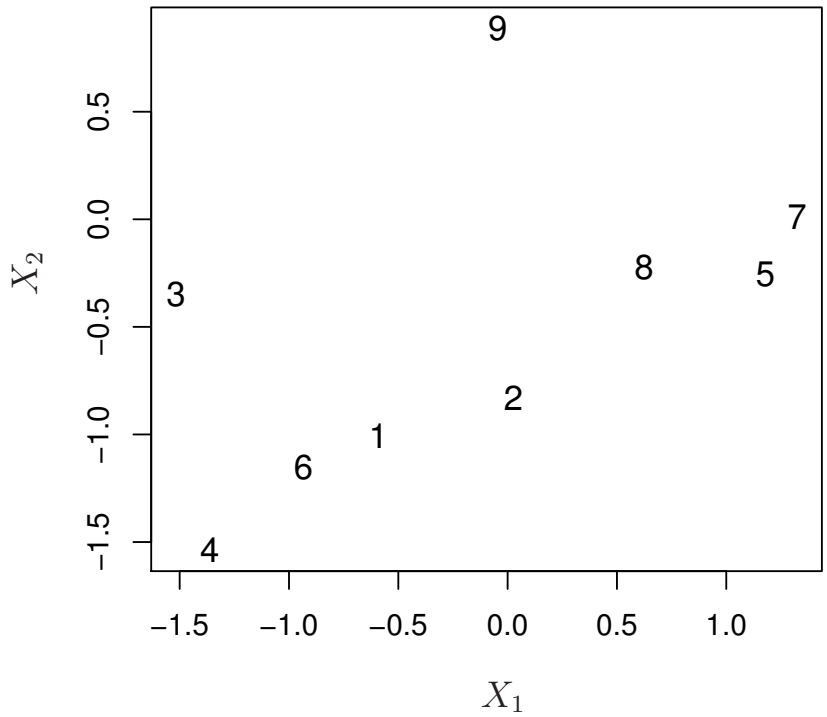
- Bottom of the tree - leaf for each observation.
- As we move up the tree, some leaves begin to fuse into branches: these are observations that are similar to each other.
- The lower in the tree fusions occur, the more similar the groups of observations are to each other.
- Observations that fuse near the top of the tree, can be quite different.
- Height of fusions indicate how similar objects are.
- Horizontal axis does not indicate how similar objects are - just the vertical.

Agglomerative Clustering

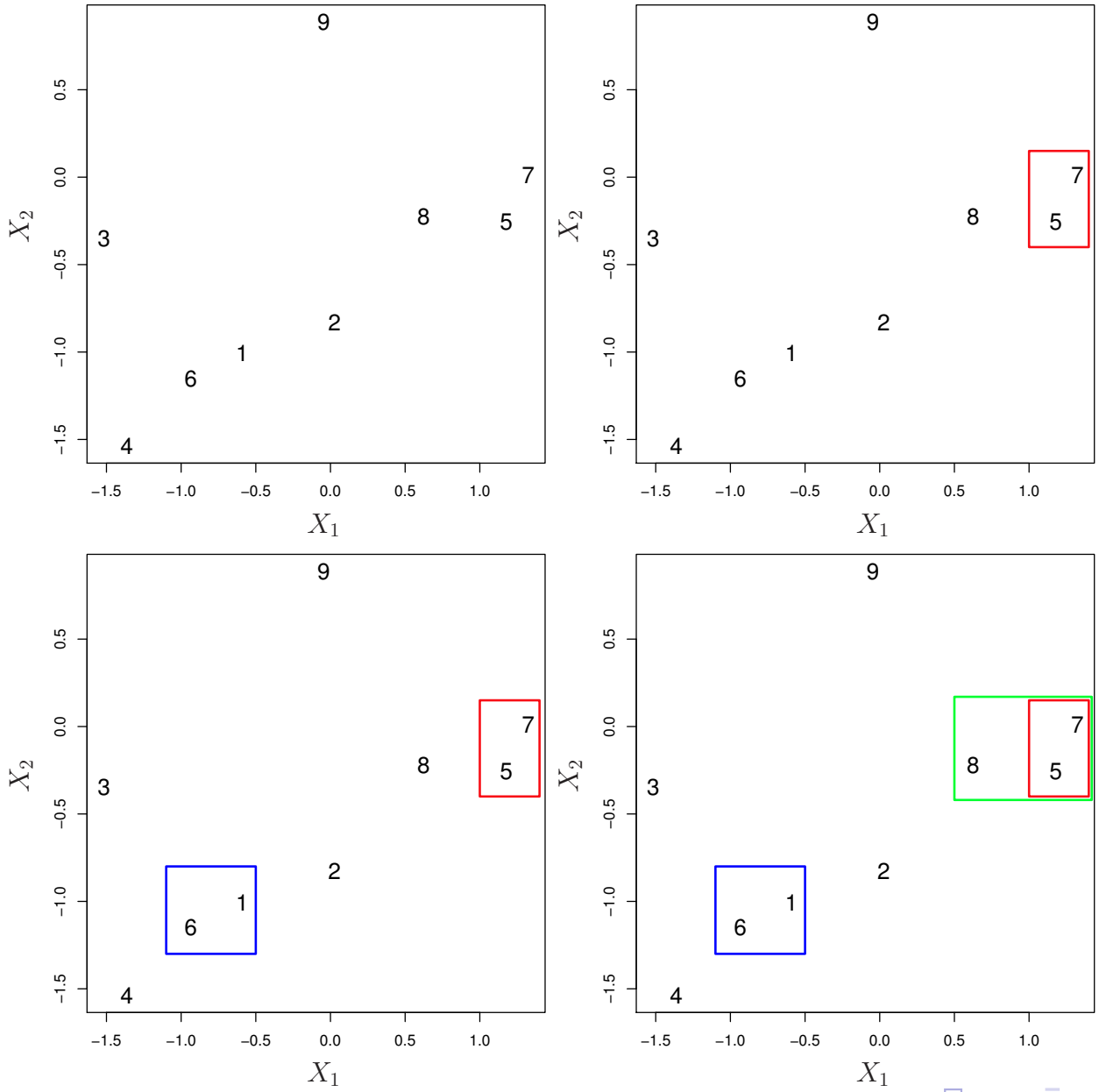
- Begin with every observation representing a singleton cluster.
- At each step, merge two "closest" clusters into one cluster and reduce the number of clusters by one.
- Need a measure of dissimilarity between two clusters - called linkages.
- Dissimilarity between G and H: $d(G, H)$, function of the set of pairwise dissimilarities $d_{ii'}$, point i is in G and point i' is in H.

Agglomerative Clustering





Agglomerative Clustering



Linkages

Linkages - Measure of dissimilarity between two sets of objects that determine how two set of objects are merged.

Major Types:

- Single linkage.
- Complete linkage.
- Average Linkage.
- Ward's Linkage.

Single Linkage

$$d_{SL}(G, H) = \min_{i \in G, i' \in H} d_{ii'}$$

.

- Minimum dissimilarity between points in two sets used to determine which two sets should be merged.
- Can handle diverse shapes.
- Very sensitive to outliers or noise.
- Often results in unbalanced clusters.
- Extended, trailing clusters in which observations fused one at a time - chaining.

Complete Linkage

$$d_{CL}(G, H) = \max_{i \in G, i' \in H} d_{ii'}$$

- Maximum dissimilarity between points in two sets used to determine which two sets should be merged.
- Often gives comparable cluster sizes.
- Less sensitive to outliers.
- Works better with spherical distributions.

Average Linkage

$$d_{AL}(G, H) = \frac{1}{n_G n_H} \sum_{i \in G} \sum_{i' \in H} d_{ii'}$$

- Average dissimilarity between points in two sets used to determine which two sets should be merged.
- A compromise between single and complete linkage.
- Less sensitive to outliers.
- Works better with spherical distributions.

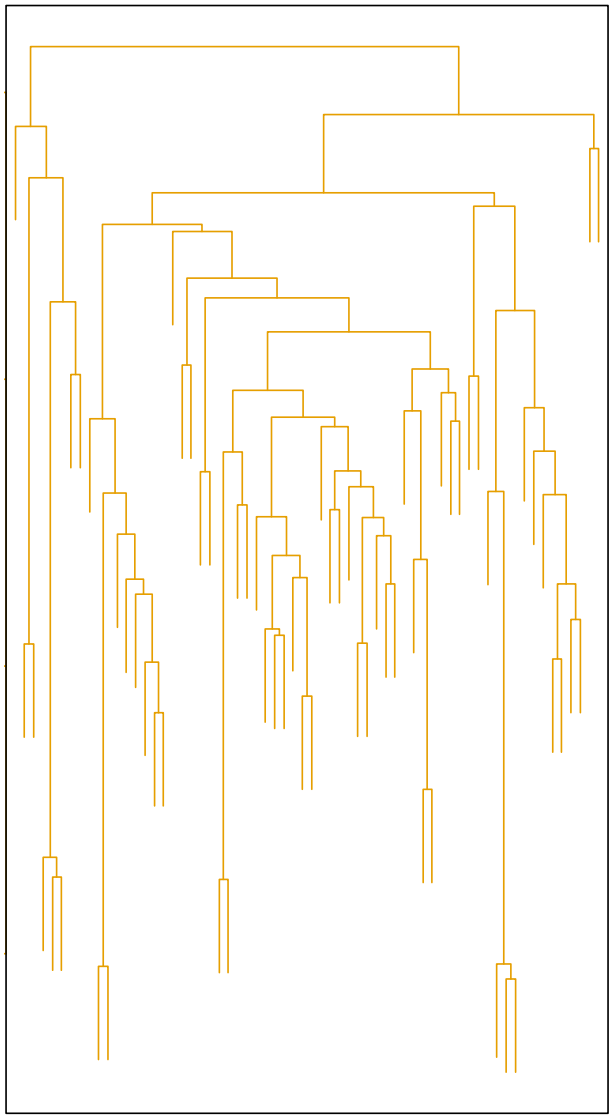
Ward's (Centroid) Linkage

$$d_{WL}(G, H) = \left(\frac{n_G n_H}{n_G + n_H} \right) ||\hat{\mu}_G - \hat{\mu}_H||_2^2$$

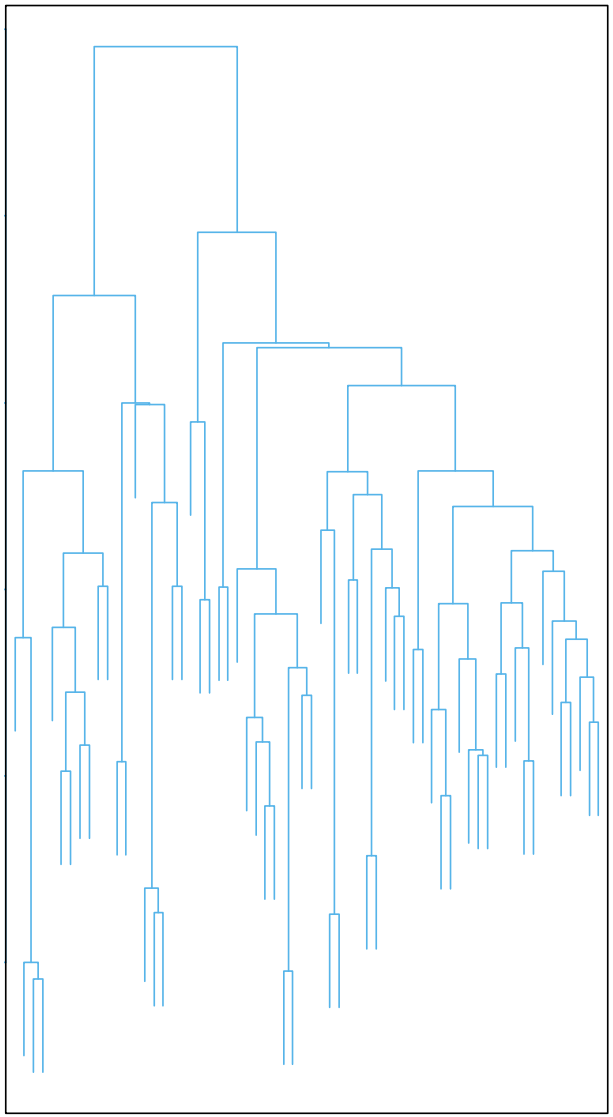
- $\hat{\mu}_G$ is the cluster mean.
- Joins clusters whose means are close (in Euclidean distance).
- Good for spherical and balanced clusters.
- Not always monotonic, leading to inversions.
  - Distance can decrease after a cluster merge.
  - Inversions "corrected" in dendrograms by just showing a merged horizontal line.

Linkage Examples

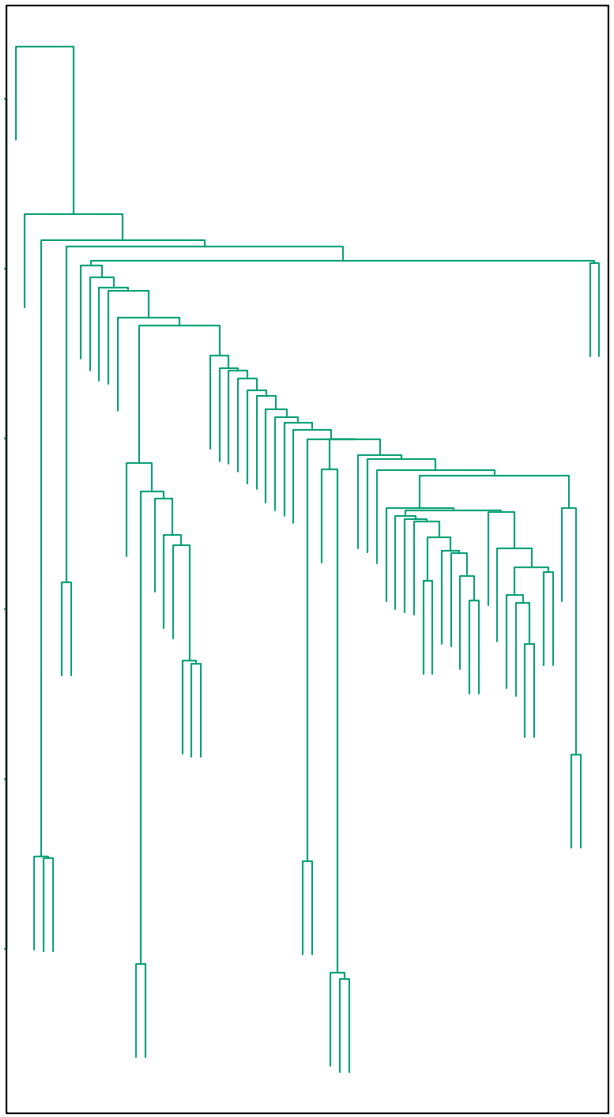
Average Linkage



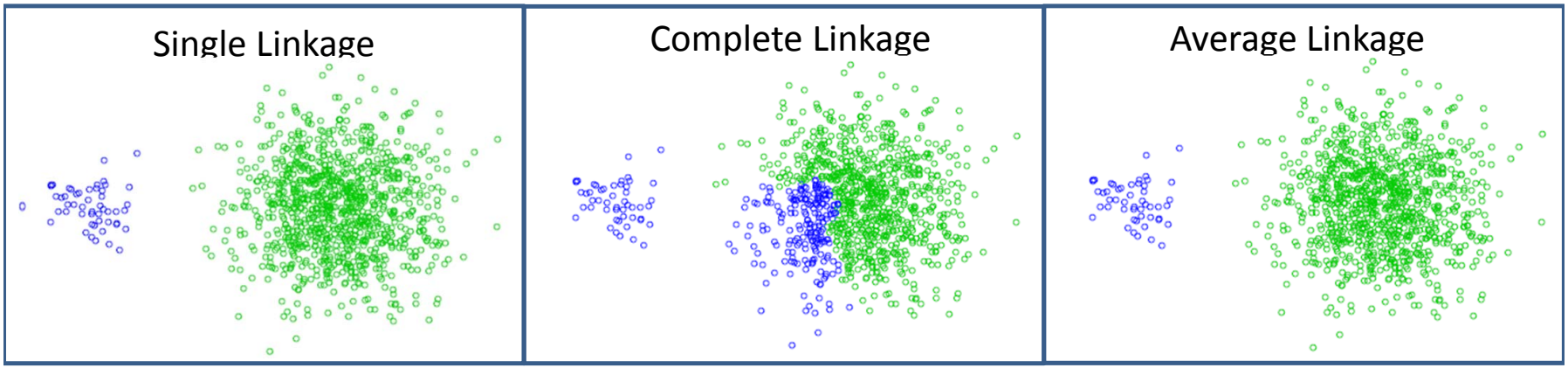
Complete Linkage



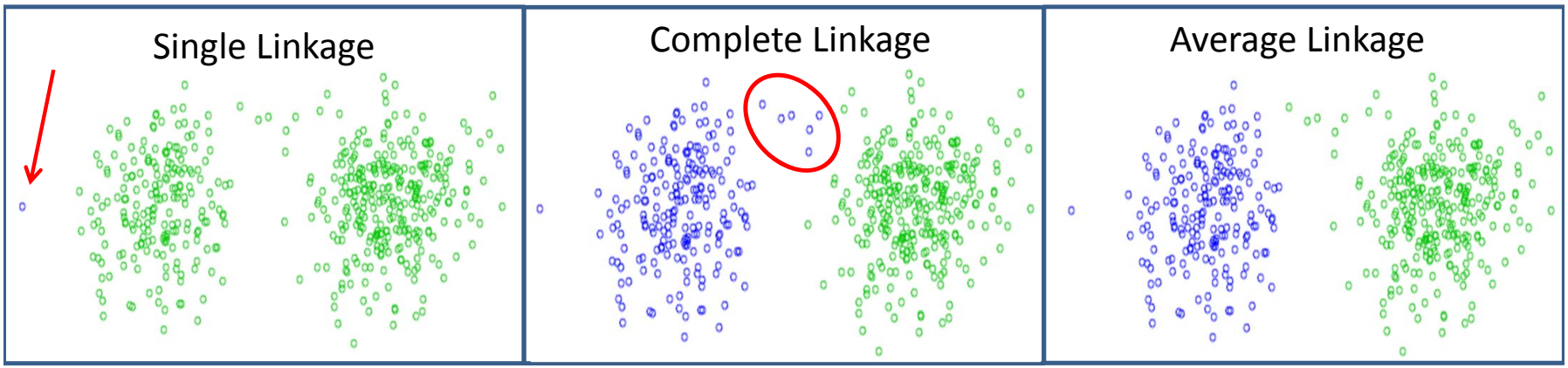
Single Linkage



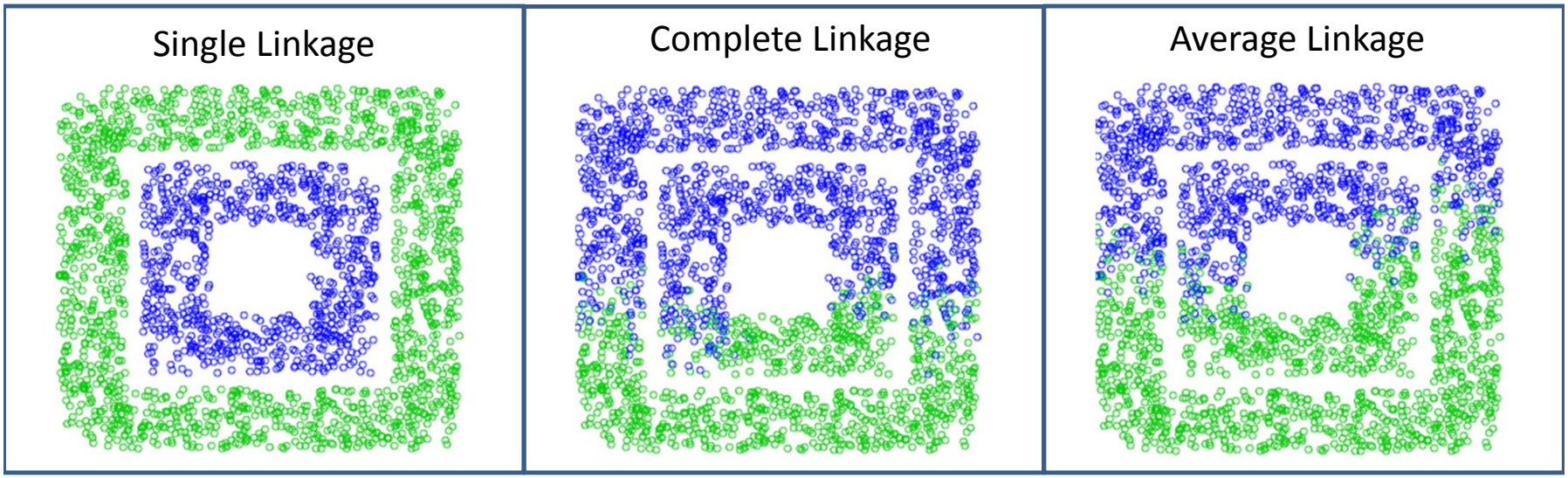
Linkage Examples



Linkage Examples



Linkage Examples



Linkages

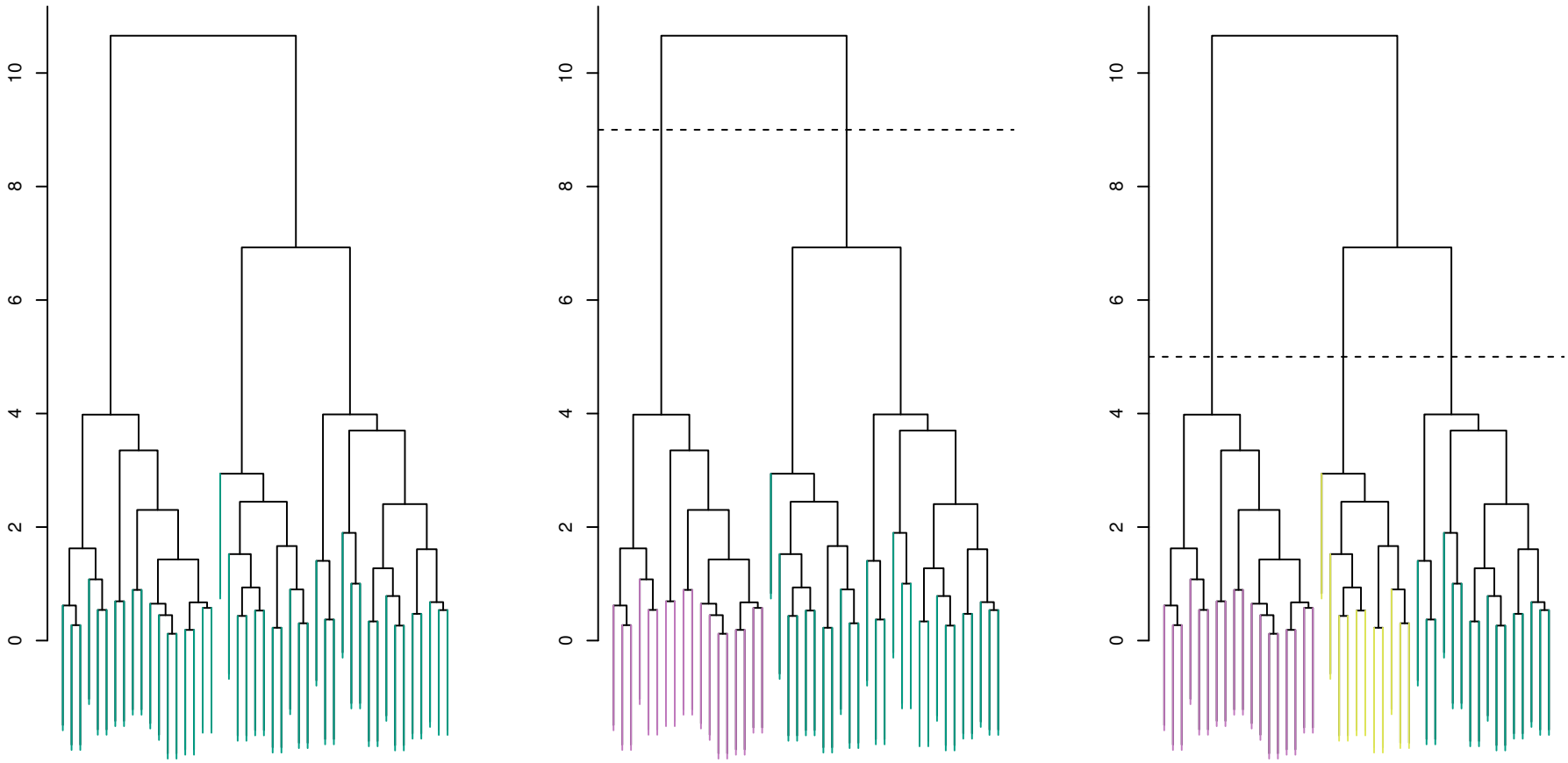
Discussion:

- When are different linkages appropriate?
- Average Linkage has a statistical consistency property violated by single and complete linkage.
- Average, single and complete avoid inversions.
- Most Robust?

R: `hclust` function with `method="single"`, `method="complete"`, `method="average"`, `method="ward.D"`

Number of Clusters

Tree-Cuts:



R: `cutree` function.

Hierarchical Clustering - Summary

Strengths:

- Simple / intuitive.
- Visualization.
- Family of possible clusterings (nested).

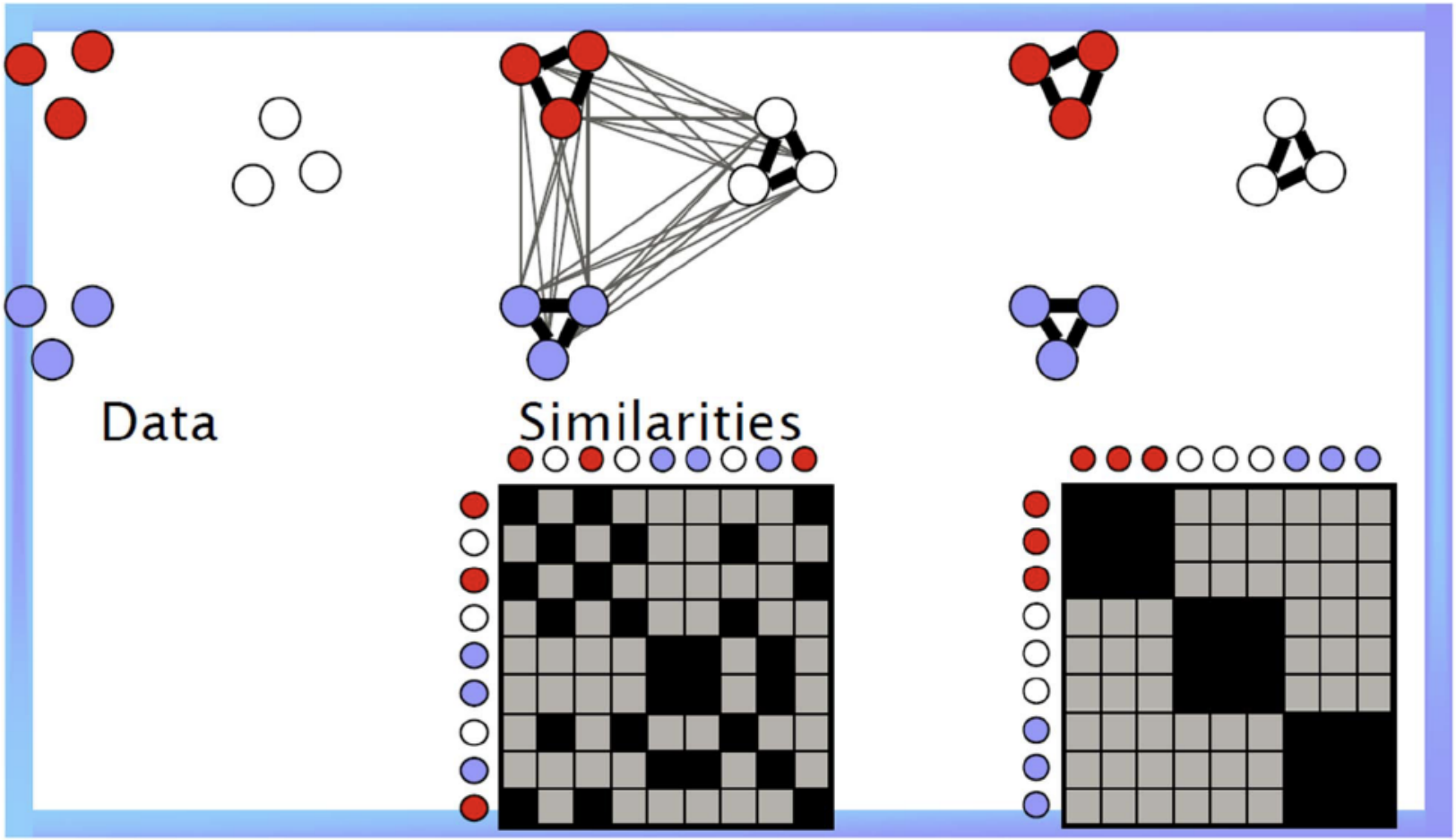
Extremely popular!!

Weaknesses:

- Local Solution.
- Unstable Solution.
- Depends heavily on type of linkage.
- No optimization criterion - purely algorithmic.

Spectral Clustering

Spectral Clustering



Sontag (2013)

Spectral Clustering

Idea:

- Represent data as a similarity graph and find connected components (clusters!).
- Highly flexible and customizable (can handle many shapes of clusters).

Approach:

1. Compute graph similarity matrix.
2. Calculate the Graph Laplacian.
3. Take eigenvalue decomposition (PCA!).
4. Perform K-means on the resulting eigenvectors.

Spectral Clustering

Step 1: Compute Graph.

- Similarity matrix.
  - $W_{n \times n}$ weighted graph where W_{ij} denotes the similarity between points i and j.
  - Similarities: Gaussian kernel, inverse distances, cosine similarity, etc.
- ϵ-neighborhood graph.
  - Connect all points whose pairwise distances are smaller than ϵ or similarities greater than τ.
- K-nearest neighbor graph.
  - Connect vertex v_i with vertex v_j if v_j is among the K-nearest neighbors of v_i.

Spectral Clustering

Step 2: Graph Laplacian.

- W a weighted adjacency matrix.
- D the degree matrix.

$$\mathrm{D} = \mathrm{diag}(d_1, \ldots d_n), \quad d_i = \sum_i W_{ij}$$

- Unnormalized Graph Laplacian

$$\mathrm{L} = \mathrm{D} - \mathrm{W}$$

- Normalized Graph Laplacian

$$\mathrm{L}_N = \mathrm{D}^{-1/2}\mathrm{L}\mathrm{D}^{-1/2} = \mathrm{I} - \mathrm{D}^{-1/2}\mathrm{W}\mathrm{D}^{-1/2}$$

Spectral Clustering

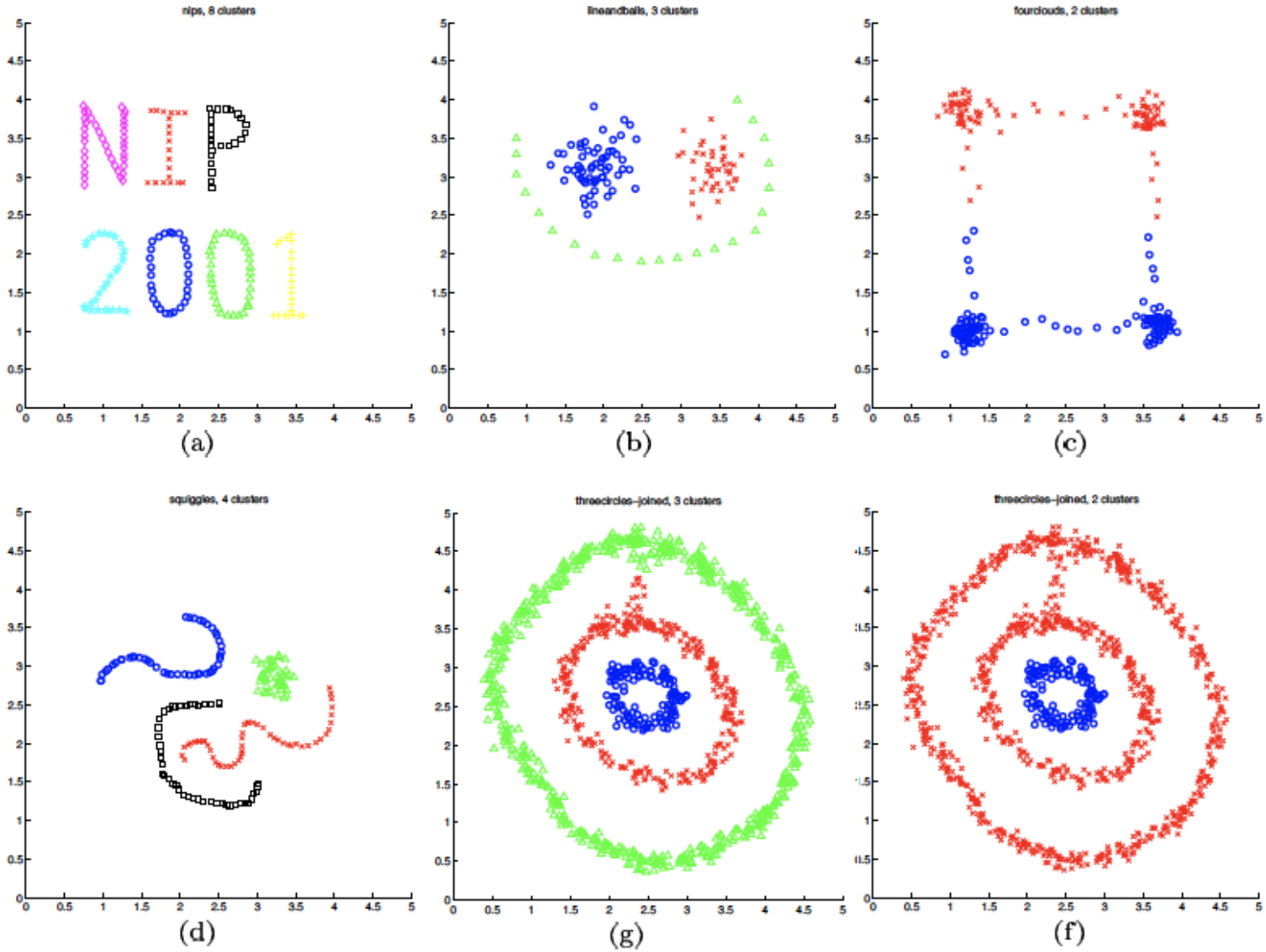
Step 3: Eigenvalue Decomposition.

- Take the eigenvalue decomposition (PCA!) of L.

Step 4: K-means.

- Perform K-means on smallest r eigenvectors with non-zero eigenvalues.

Spectral Clustering



Ng, Jordan, Weiss (2002)

Spectral Clustering - Summary

Strengths:

- Flexible clustering.
- Customizable - many graph representations.

Weaknesses:

- Local Solution (inherits properties of K-means).
- Many choices of graph similarity.
- Others?

Biclustering

Biclustering

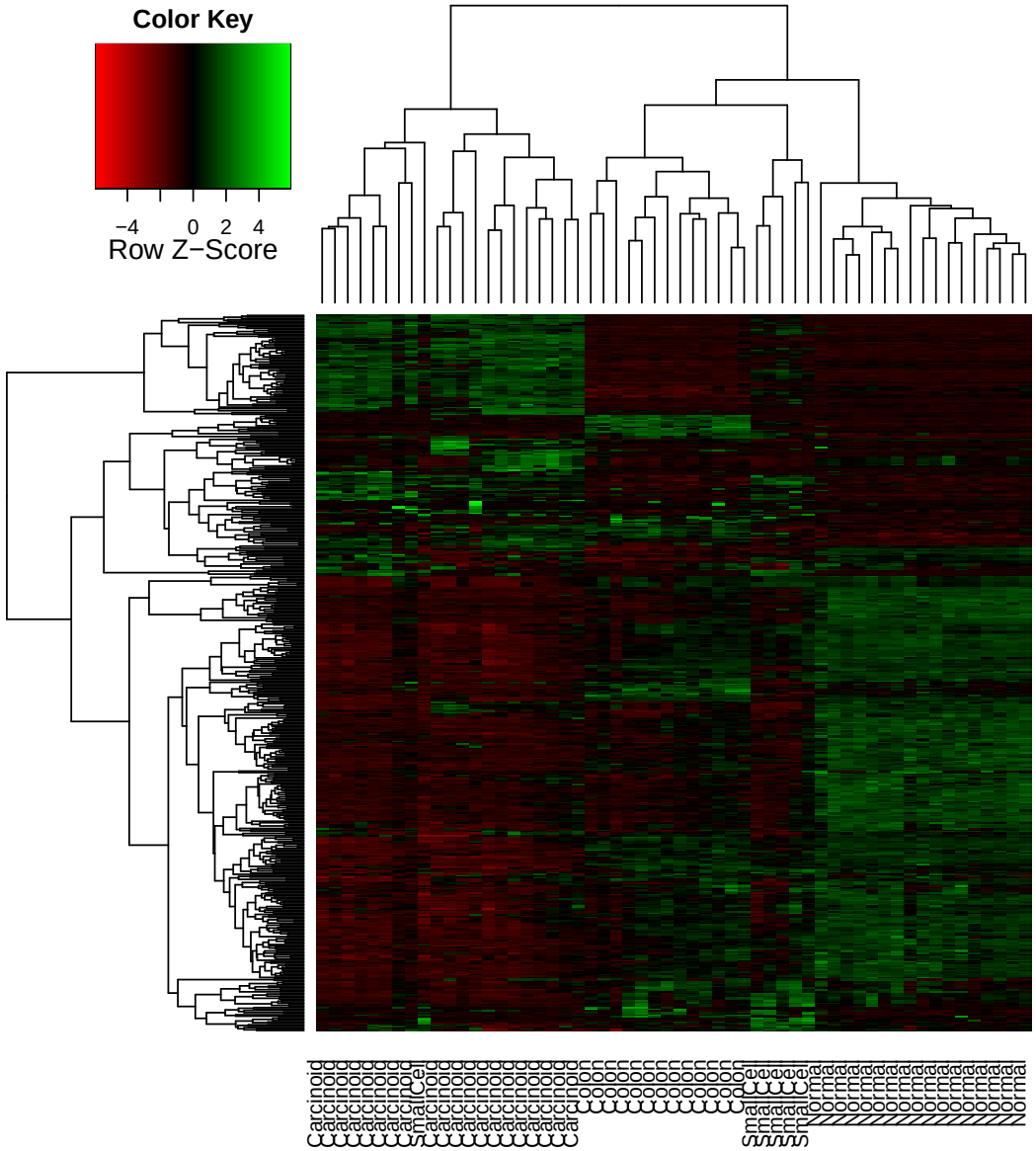
- Idea: Find groups of BOTH observations & features.
- Like clustering both rows and columns of data matrix.

Two main types:

1. Overlapping Biclusters.
   - Plaid models & Sparse SVD models.
2. Non-overlapping Biclusters (Checkerboard mean).
   - Cluster heatmap. (`heatmap` in `R`)

Cluster Heatmap

Hierarchical Clustering Separately on Rows & Columns:



Biclustering - Applications

- Biomedicine - "omics" data.
  - Cancer genomics: Finding subtypes. Find groups of patients (subtypes) and groups of genes (genomic signatures) that separate subtypes.
  - Famous Example: Breast Cancer.
- Text mining.
  - Word-Document associations.
- Collaborative Filtering.
  - Find users who highly rate particular products.
  - Famous Examples: Netflix, Amazon.

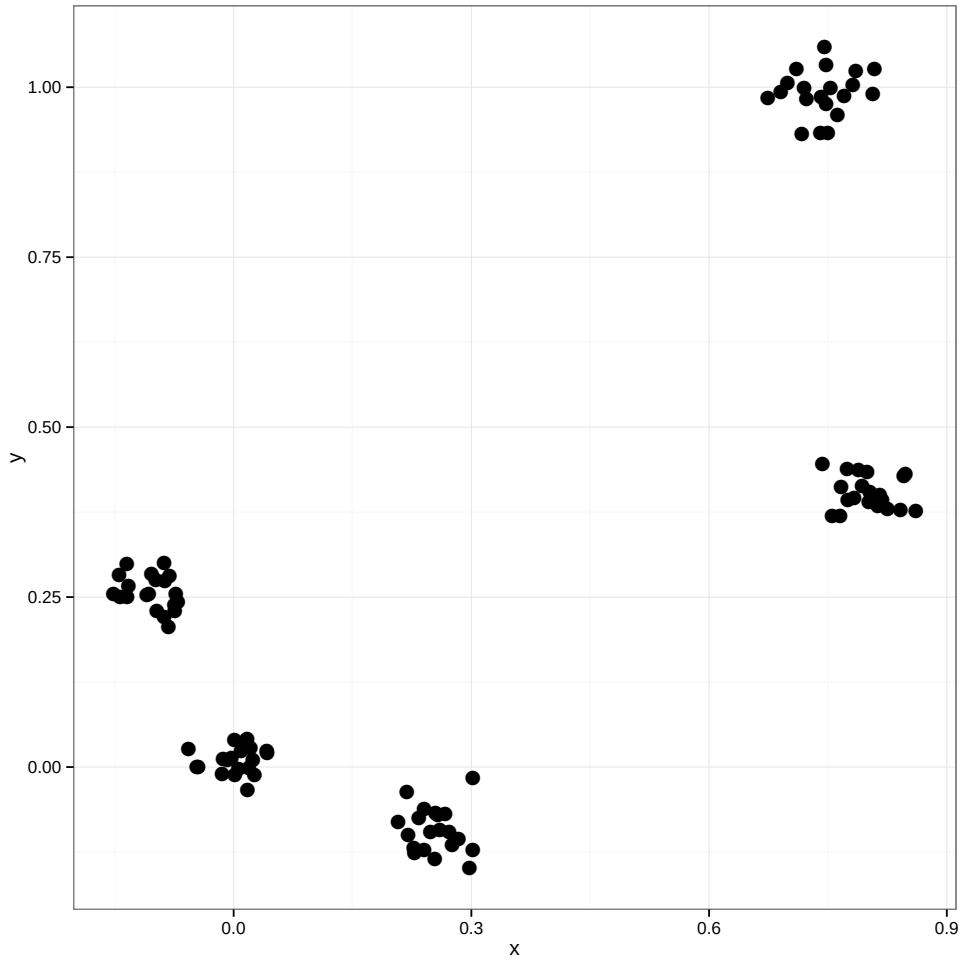
Convex Clustering

Convex Clustering

$$\underset{\mathsf{u}}{\text{minimize}} \quad \frac{1}{2}\sum_{i=1}^{n}\|\mathsf{x}_i - \mathsf{u}_i\|_2^2 + \lambda \sum_{i<j} w_{ij}\|\mathsf{u}_i - \mathsf{u}_j\|_2$$

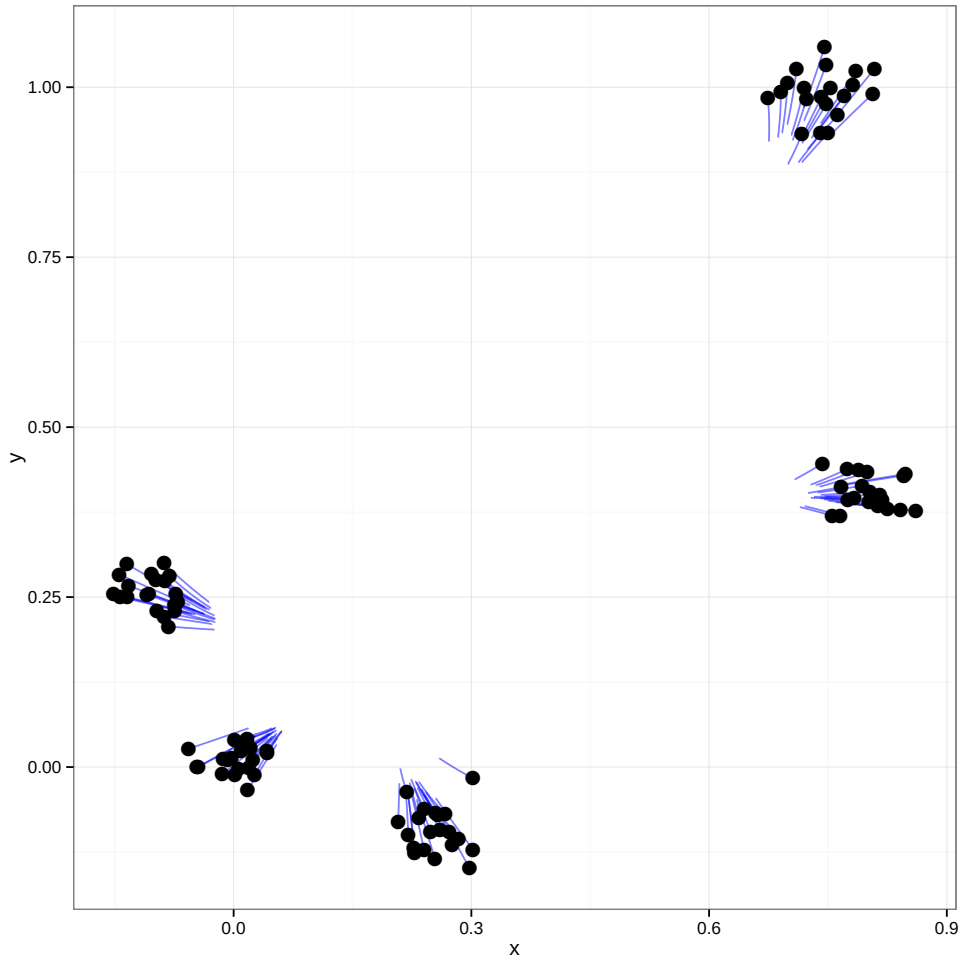
- x_i - each observation (p-vector).
- u_i - cluster centroid for each observation.
- $\frac{1}{2}\sum_{i=1}^{n}\|\mathsf{x}_i - \mathsf{u}_i\|_2^2$ - K-means loss function!
- $\sum_{i<j} w_{ij}\|\mathsf{u}_i - \mathsf{u}_j\|_2$ - Fusion as with Hierarchical!
- λ controls BOTH cluster assignments & number of clusters.

Convex Clustering Solution Path



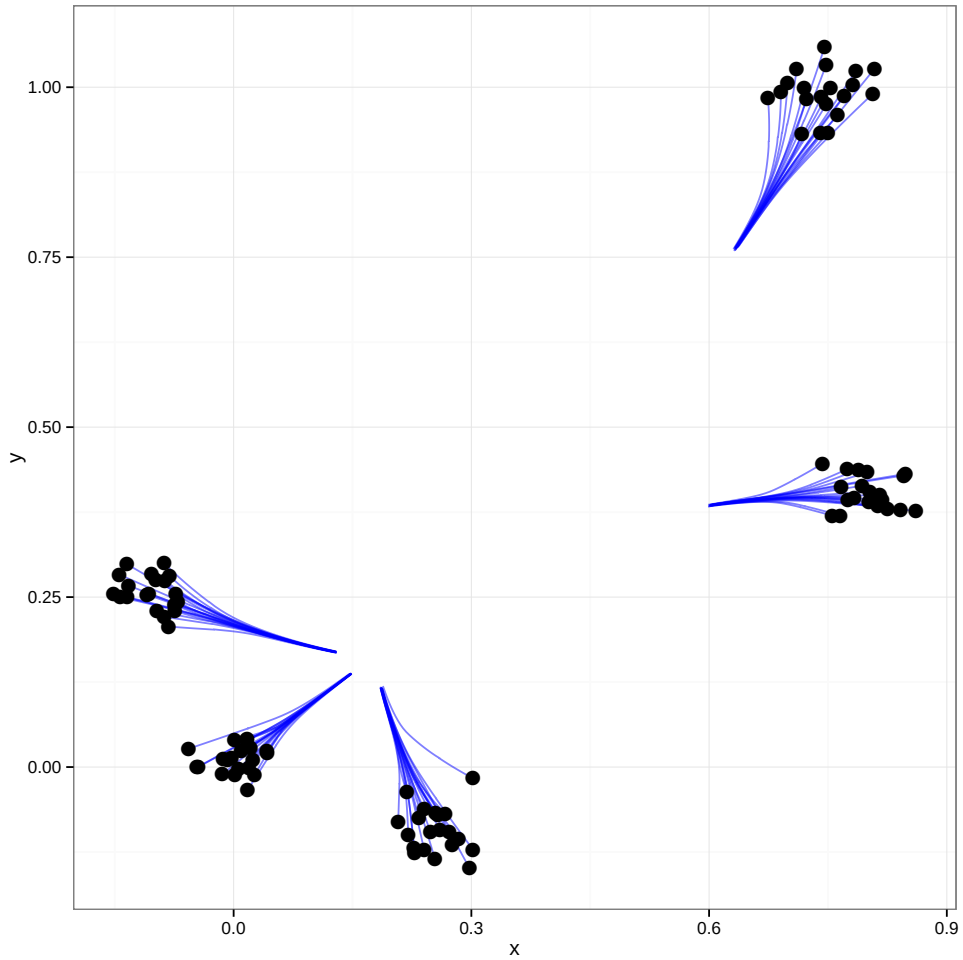
$\lambda = 0$

Convex Clustering Solution Path



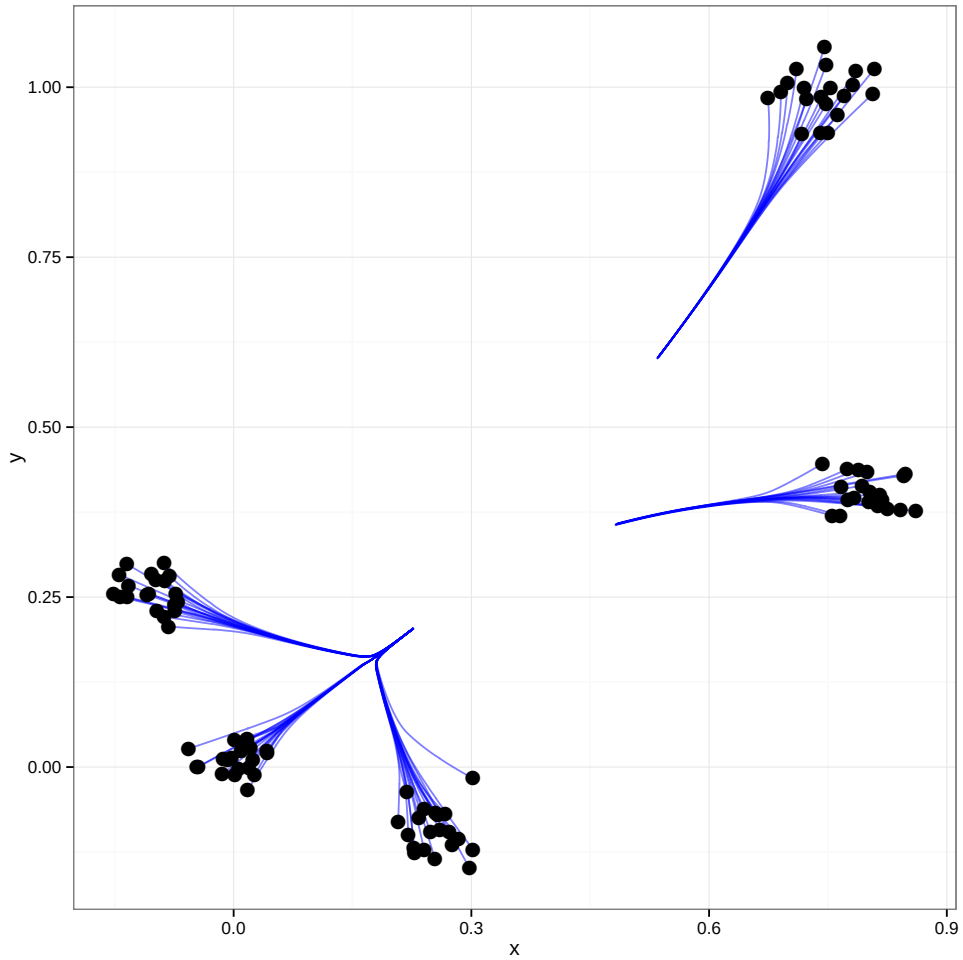
λ

Convex Clustering Solution Path



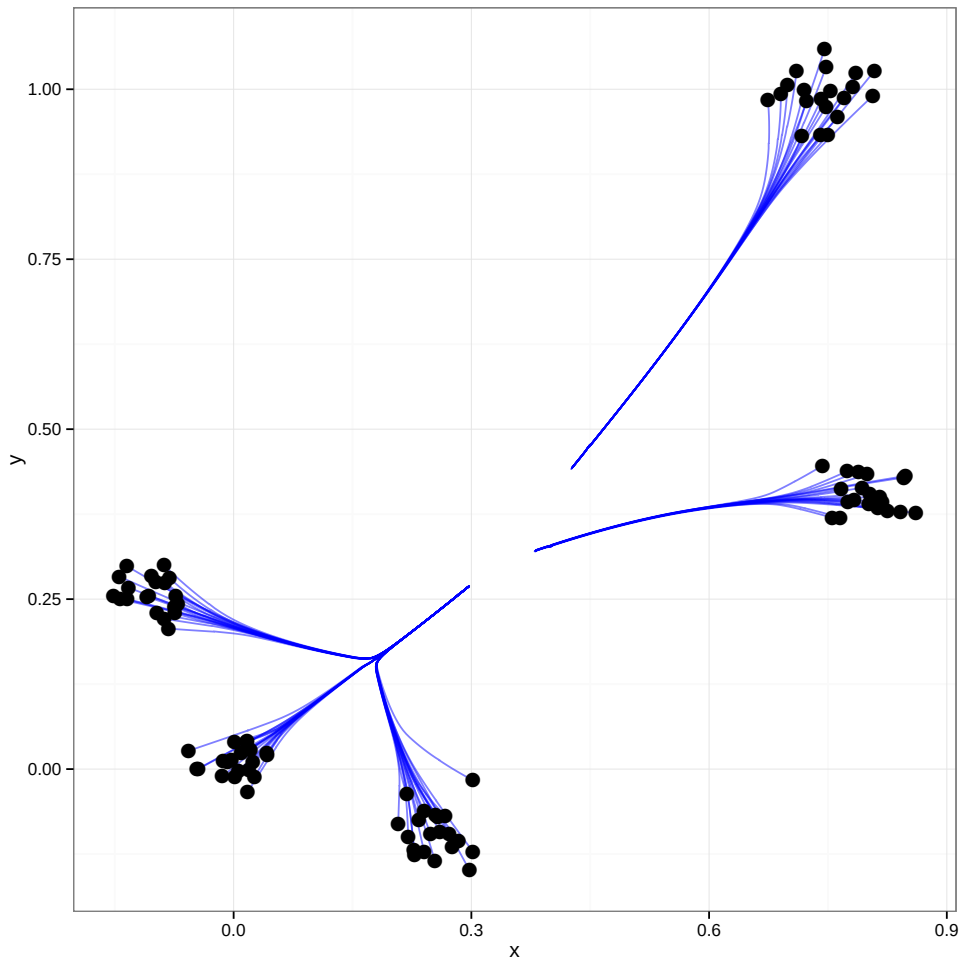
λ

Convex Clustering Solution Path

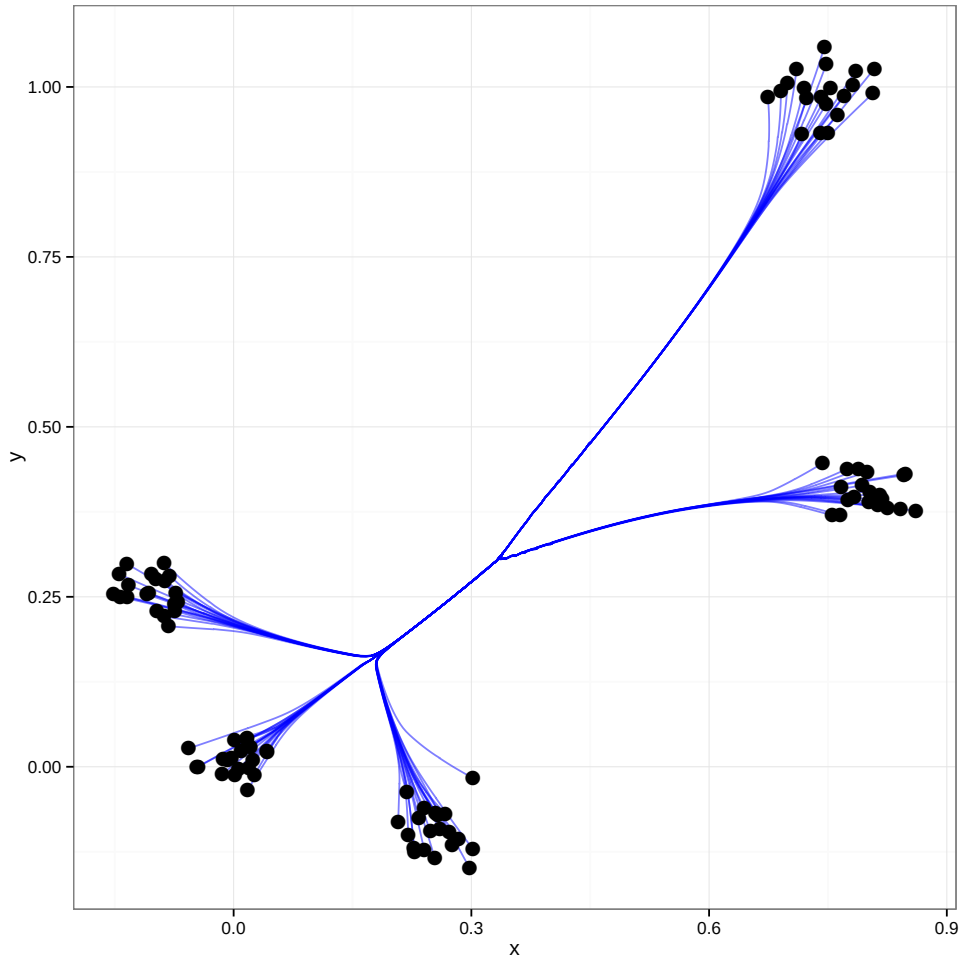


λ

Convex Clustering Solution Path



Convex Clustering Solution Path



λ

Research Highlight

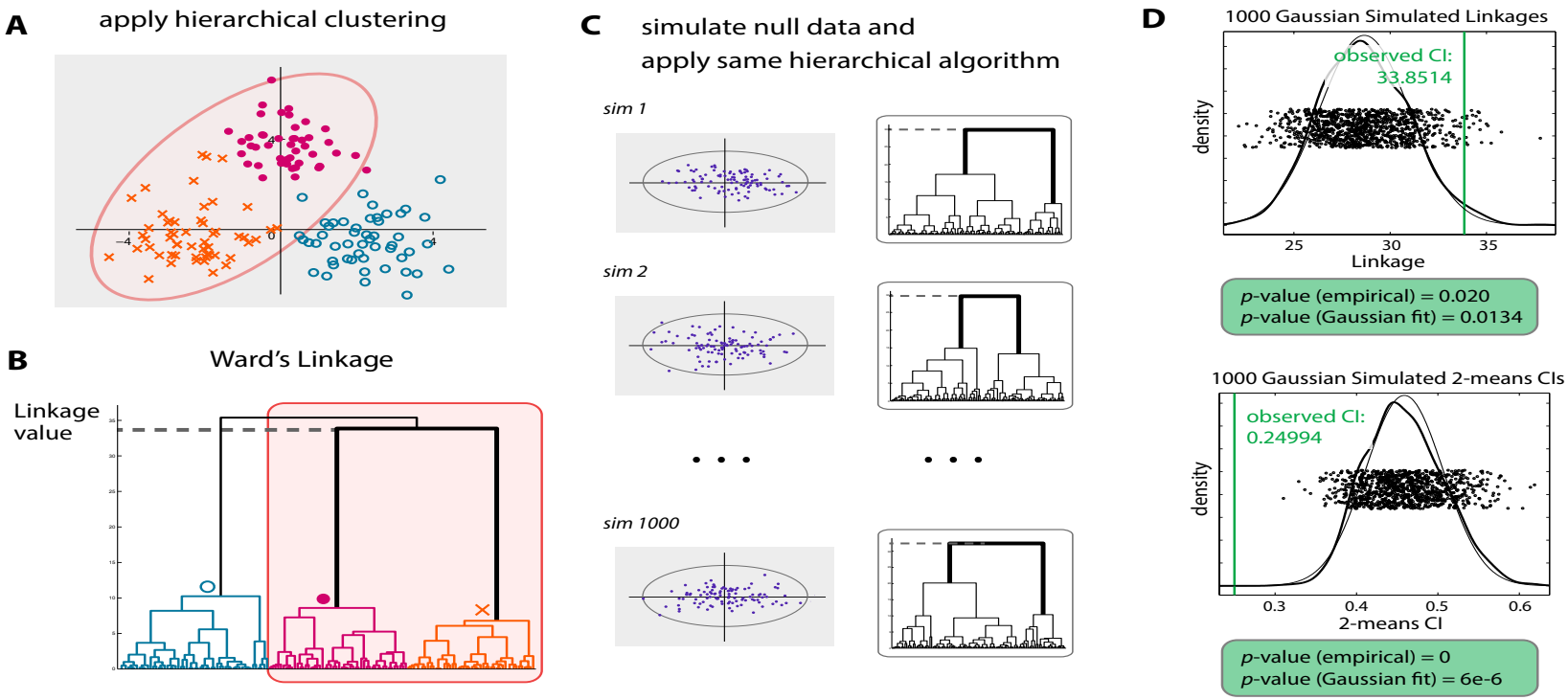
Fast, Dynamic & Interactive Visualization for Convex Clustering

Solution: Algorithmic Regularization Paths! (Weylandt, Nagorski, and Allen, 2020)

`R` Package: `clustRviz` (`github.com/DataSlingers/clustRviz`)

Research Highlight

How to assess significance of clusters found by clustering methods?



Solution: Statistical Significance of Clustering (SigClust) for both flat clustering and hierarchical clustering! (Liu et al., 08; Huang et al., 15; Kimes et al, 17)

`R` Packages: `sigclust` and `hsigclust`.

Hierarchical SigClust