

# 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:

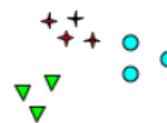
- What is a meaningful cluster?
- How do we validate clustering results?

# Clustering Challenges

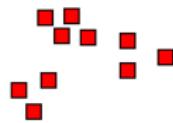
What are meaningful clusters?



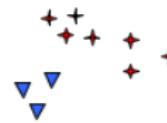
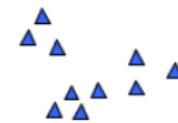
How many clusters?



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  $j$ th objects (or observations).  $D$  is typically symmetric.
- One can use a dissimilarity matrix instead.
- Dissimilarity between  $x_i$  and  $x_{i'}$ :

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

- A weighted version:

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

## Types of Distances:

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

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

- If inputs are standardized,  $\sum_j (x_{ij} - x_{i'j})^2 \propto 2(1 - \rho(x_i, x_{i'}))$ : 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  $i \in \{1, 2, \dots, n\}$ .
- $k$  clusters:  $k \in \{1, \dots, K\}$ .
- Let  $k = C(i)$  denote the  $i$ th 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:

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

- 

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

- $W(C) = T - B(C)$ .
- 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

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

## *K*-means

Objective: Minimize  $W(C)$ .

$$\begin{aligned} W(C) &= \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{i \neq i', C(i')=k} \|x_i - x_{i'}\|_2^2 \\ &= \sum_{k=1}^K n_k \sum_{C(i)=k} \|x_i - \bar{x}_k\|_2^2 \end{aligned}$$

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

# *K*-means

Idea:

- Augment  $W(C)$  with cluster means,  $\mathbf{m}_k$ :

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

- Minimize  $W(C, \mathbf{m}_k)$  by iteratively optimizing:
  - ① Cluster means:  $\mathbf{m}_k$  (with  $C(i)$  fixed).
  - ② Cluster assignments:  $C(i)$  (with  $\mathbf{m}_k$  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)

$$\hat{\mathbf{m}}_k = \operatorname{argmin}_{\mathbf{m}} \sum_{C(i)=k} \|\mathbf{x}_i - \mathbf{m}_k\|_2^2$$

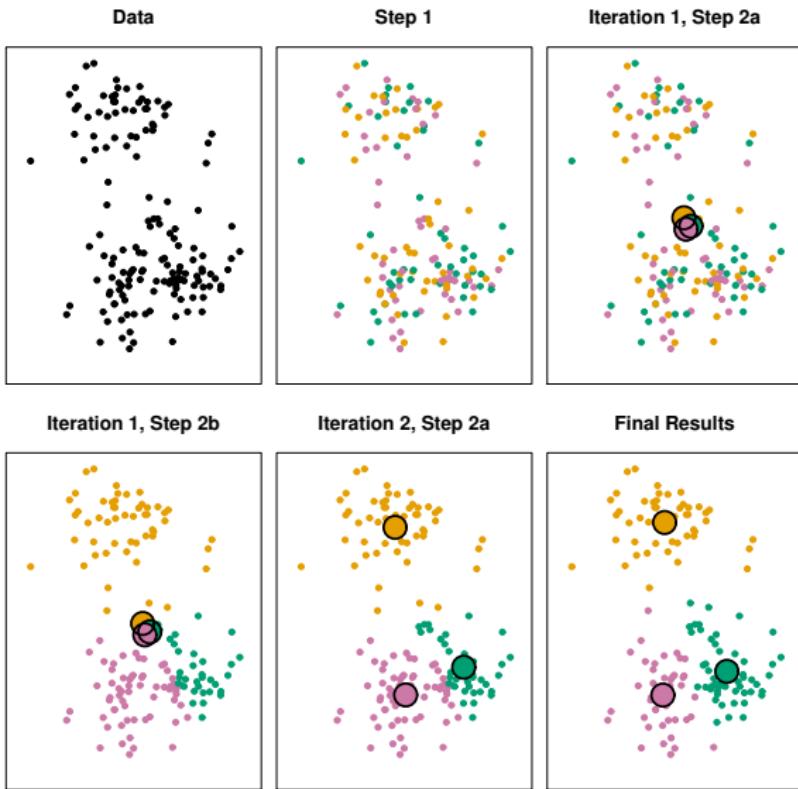
⇒ Take mean of points in cluster.

- ② Find cluster assignments. (cluster means fixed)

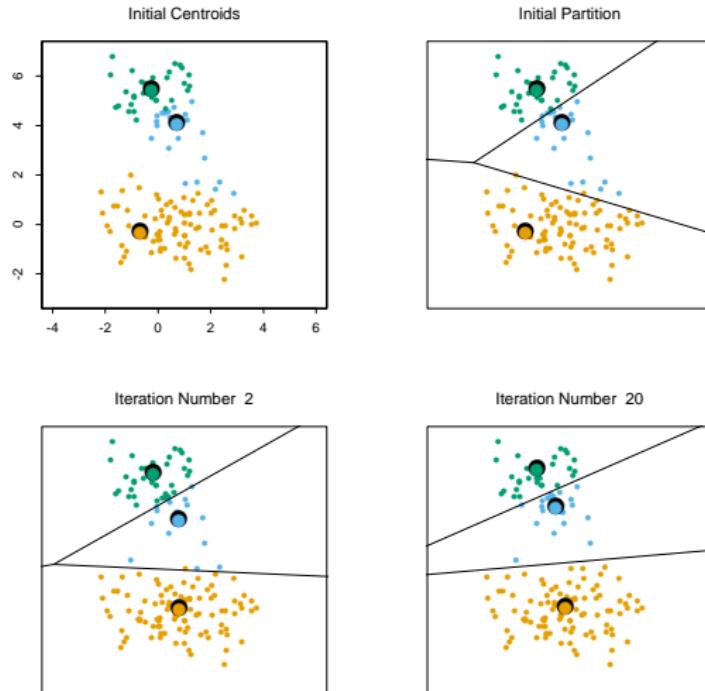
$$\hat{C}(i) = \operatorname{argmin}_k \|\mathbf{x}_i - \mathbf{m}_k\|_2^2$$

⇒ Assign each observation to closest cluster mean.

# K-means Algorithm



# K-means Algorithm

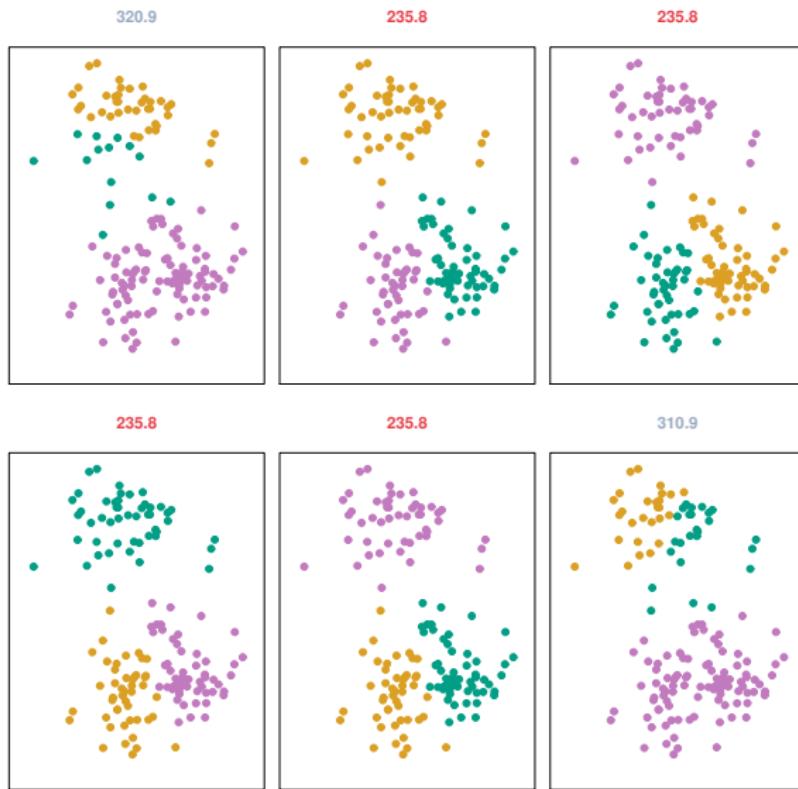


# *K*-means Properties

- 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.

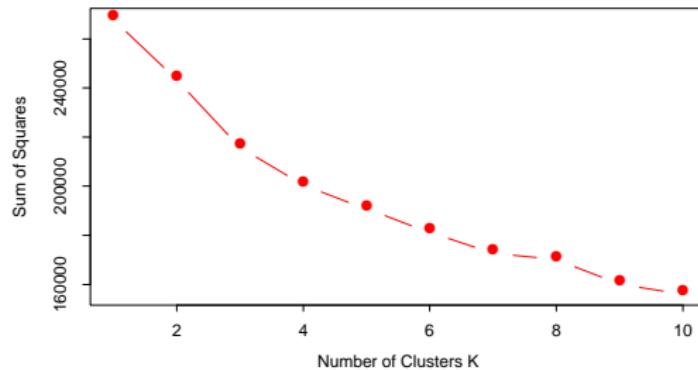
*K*-means in R: `kmeans`

# K-means - Initializations

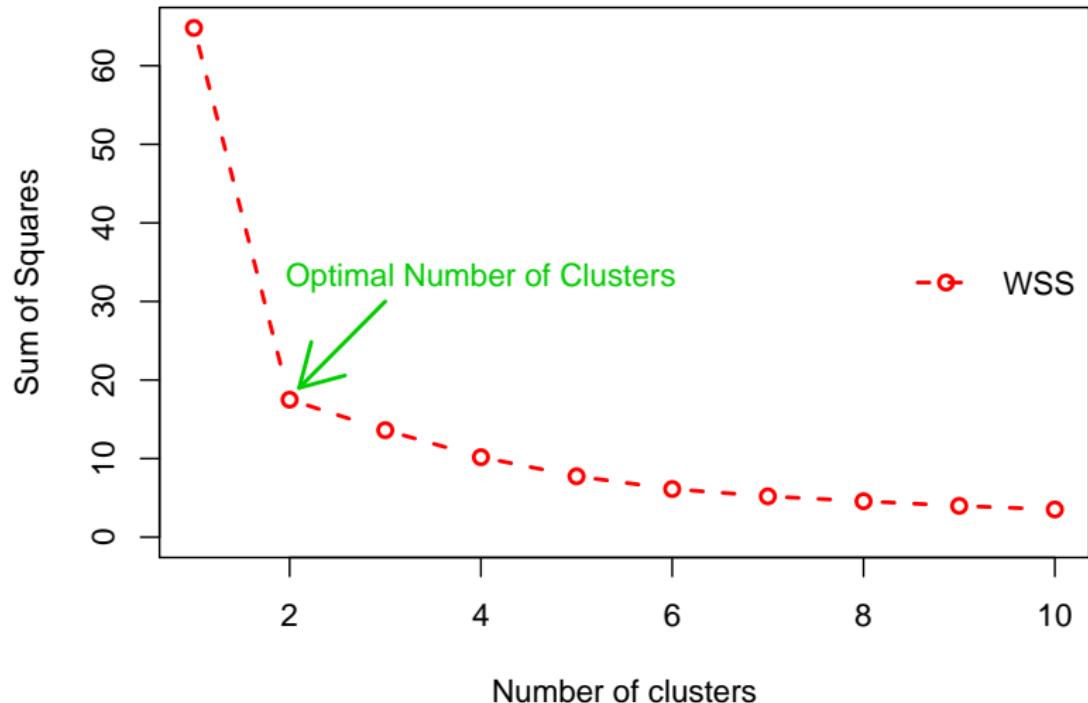


# How to Choose $K$ ?

- 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.

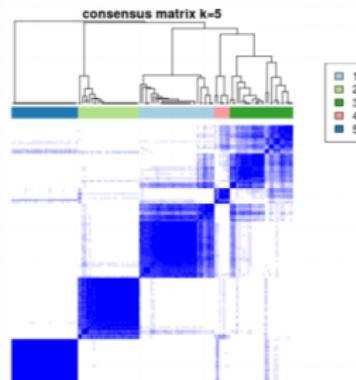
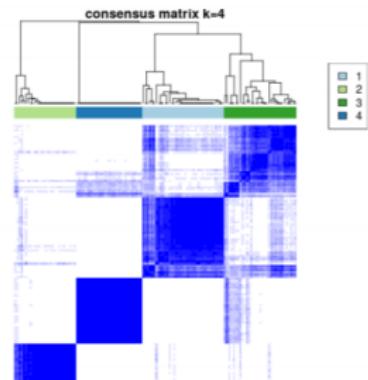
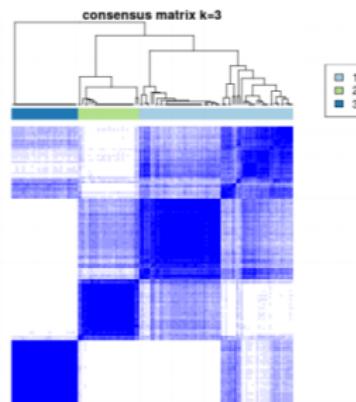
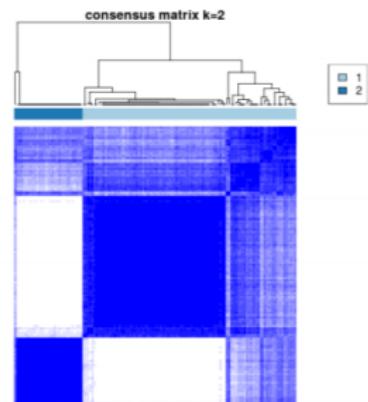
# How to Choose $K$ ?

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.
- High-dimensional settings? ( $p \gg n$  - more features than observations)
  - ▶ (In high-dimensions, apply PCA to reduce dimension before using  $K$ -means.)
- Others?

# $K$ -means - Related Algorithms

## 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

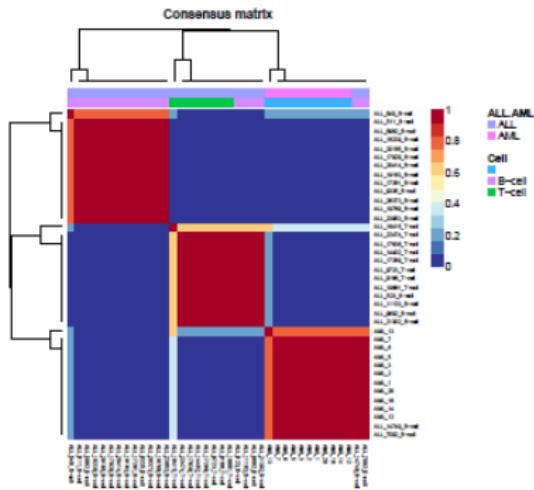
$$\mathbf{X}_{n \times p} \approx \mathbf{W}_{n \times K} \mathbf{H}_{K \times p}$$

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

NMF package in R.

# Soft Clustering - NMF

Consensus NMF Clustering:



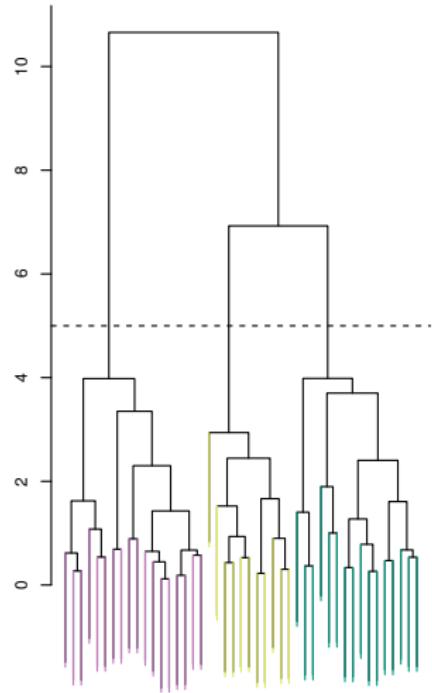
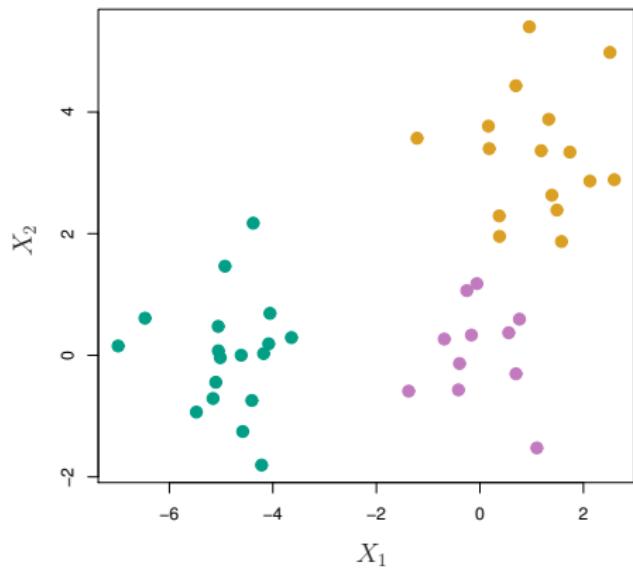
- Consensus = % time observation  $i$  and  $j$  in same cluster.

# 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



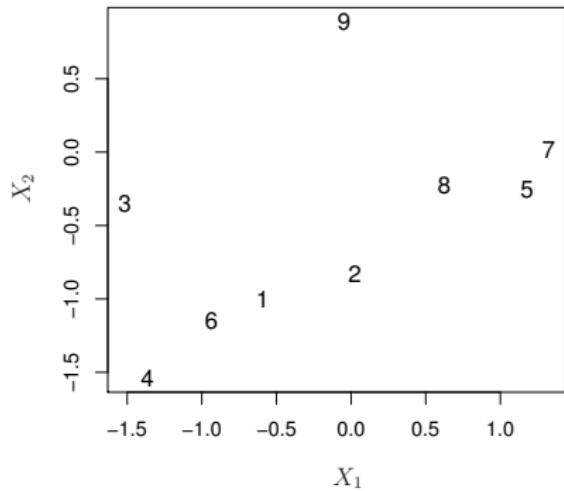
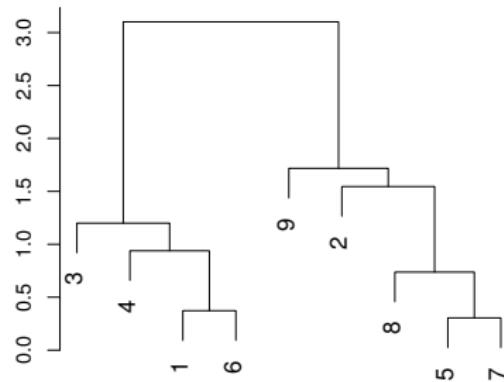
## Interpreting a Dendrogram

- 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.

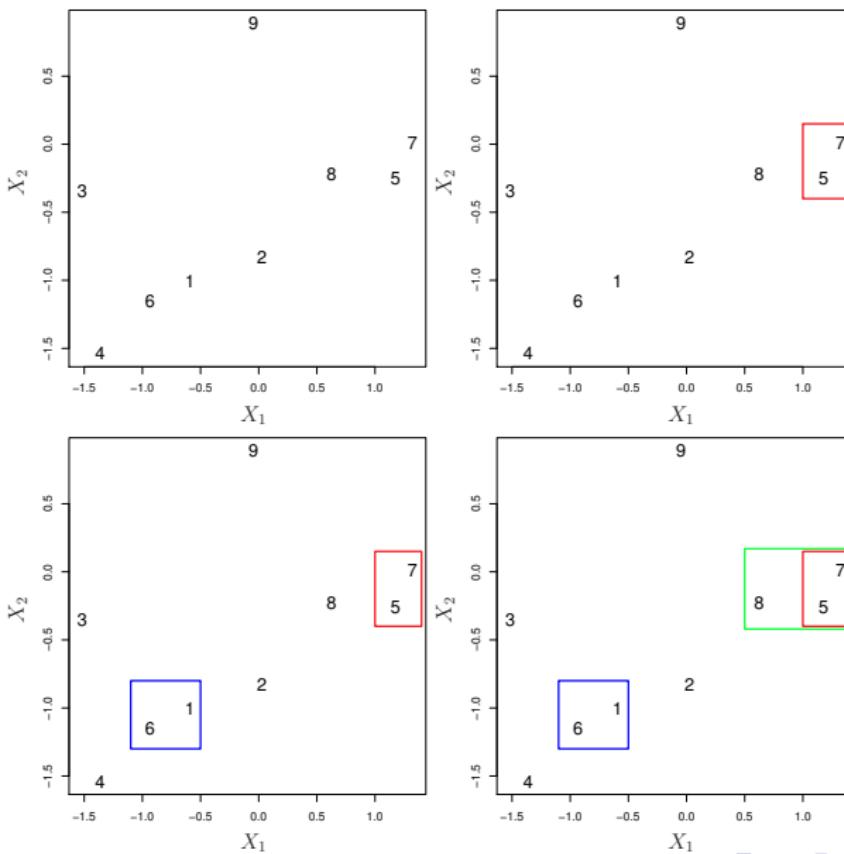
# Agglomerative Clustering

- 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.

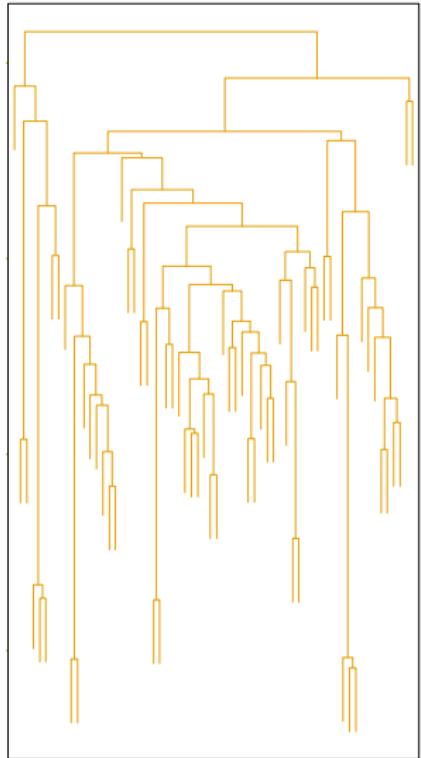
## Ward's Linkage

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

- $\hat{\mu}_G$  is the cluster mean.
- Joins clusters whose means are close (in Euclidean distance).
- Good for spherical and balanced clusters.

# Linkage Examples

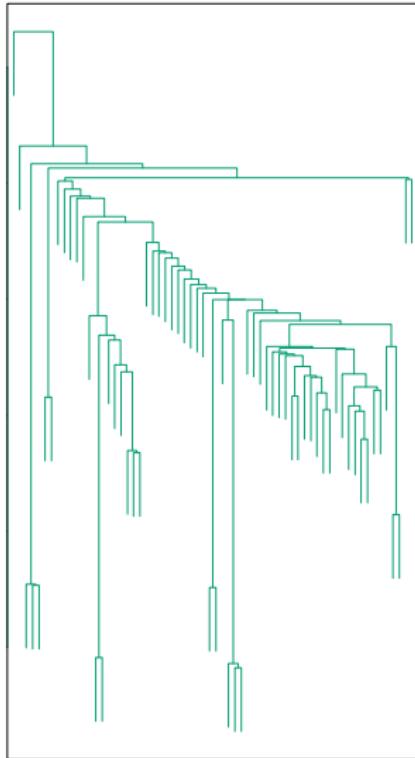
Average Linkage



Complete Linkage

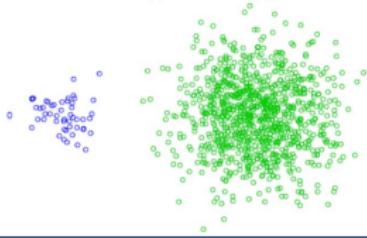


Single Linkage

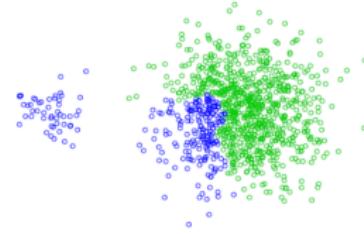


# Linkage Examples

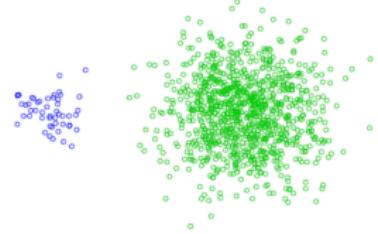
Single Linkage



Complete Linkage

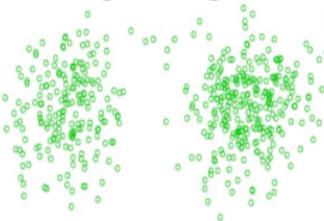


Average Linkage

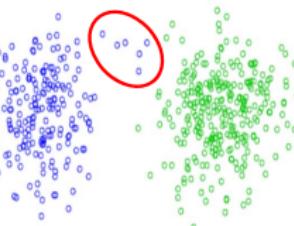


## Linkage Examples

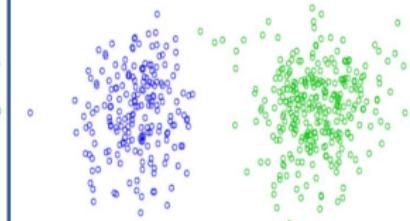
## Single Linkage



## Complete Linkage

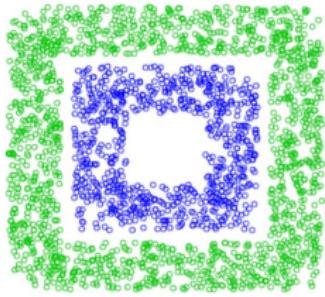


## Average Linkage

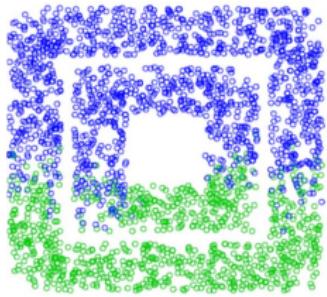


# Linkage Examples

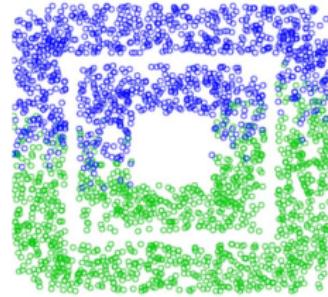
Single Linkage



Complete Linkage



Average Linkage



# Linkages

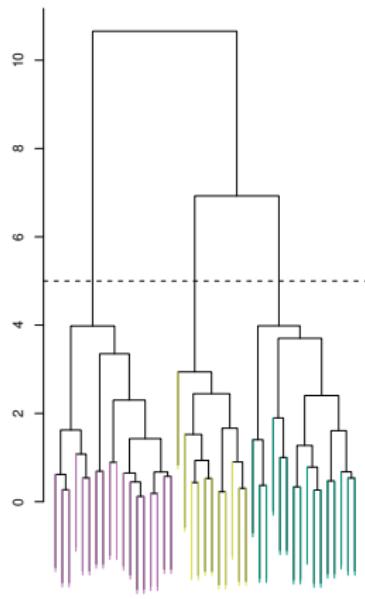
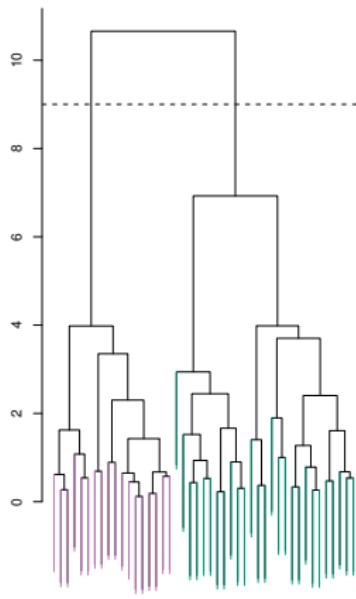
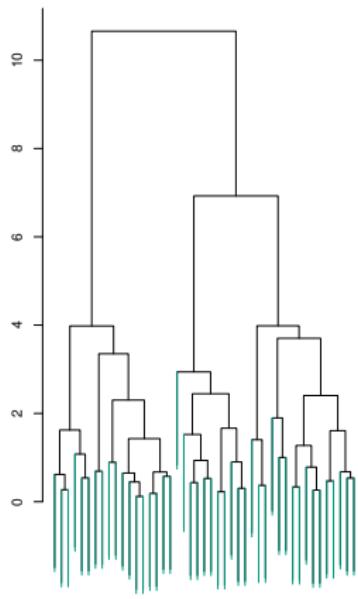
## Discussion:

- When are different linkages appropriate?
- Average Linkage has a statistical consistency property violated by single and complete linkage.
- 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).

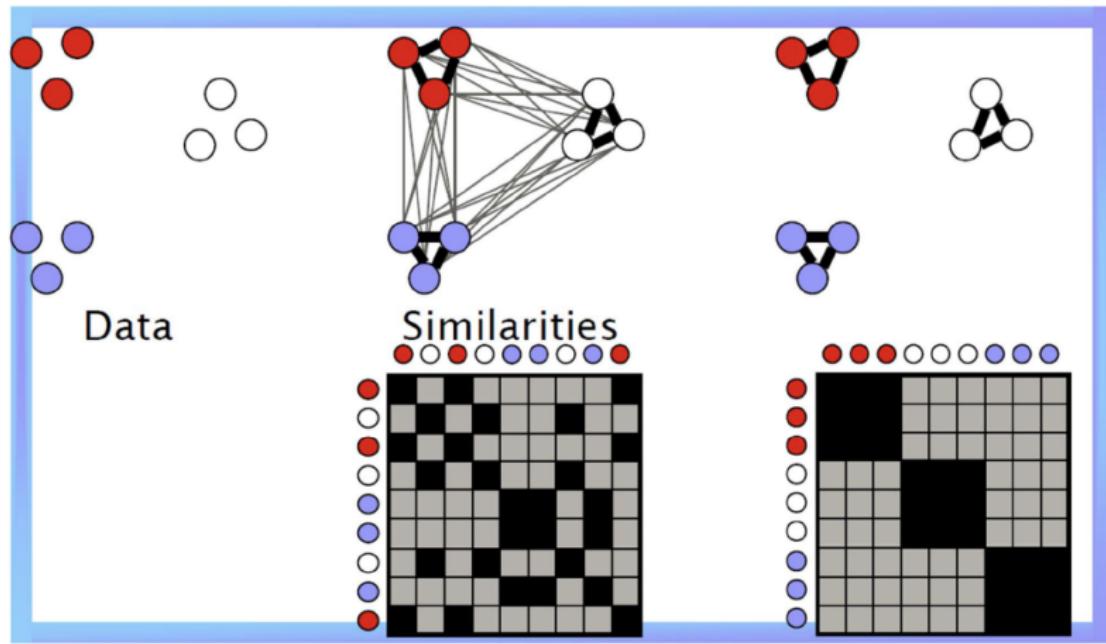
Extremely popular!!

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.
- $\epsilon$ -neighborhood graph.
  - ▶ Connect all points whose pairwise distances are smaller than  $\epsilon$ .
- $K$ -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.

$$D = \text{diag}(d_1, \dots, d_n), \quad d_i = \sum_j W_{ij}$$

- Unnormalized Graph Laplacian

$$L = D - W$$

- Normalized Graph Laplacian

$$L_N = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

# Spectral Clustering

Step 3: Eigenvalue Decomposition.

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

Step 4:  $K$ -means.

- Perform  $K$ -means on smallest  $r$  non-zero eigenvectors.

## Graph Cut Point of View

- Alternative way to derive the spectral clustering algorithm
- Consider a partition of the graph into two parts A and B
- One intuitive goal is to find the partition minimizing the cut
- $\text{Cut}(A, B)$ : sum of the weights of the set of edges that connect the two groups A and B
- Drawback of cut: the result can be sensitive to outliers (e.g. singleton clusters)

## Graph Cut Point of View

- Improving Ncut: RatioCut and Ncut

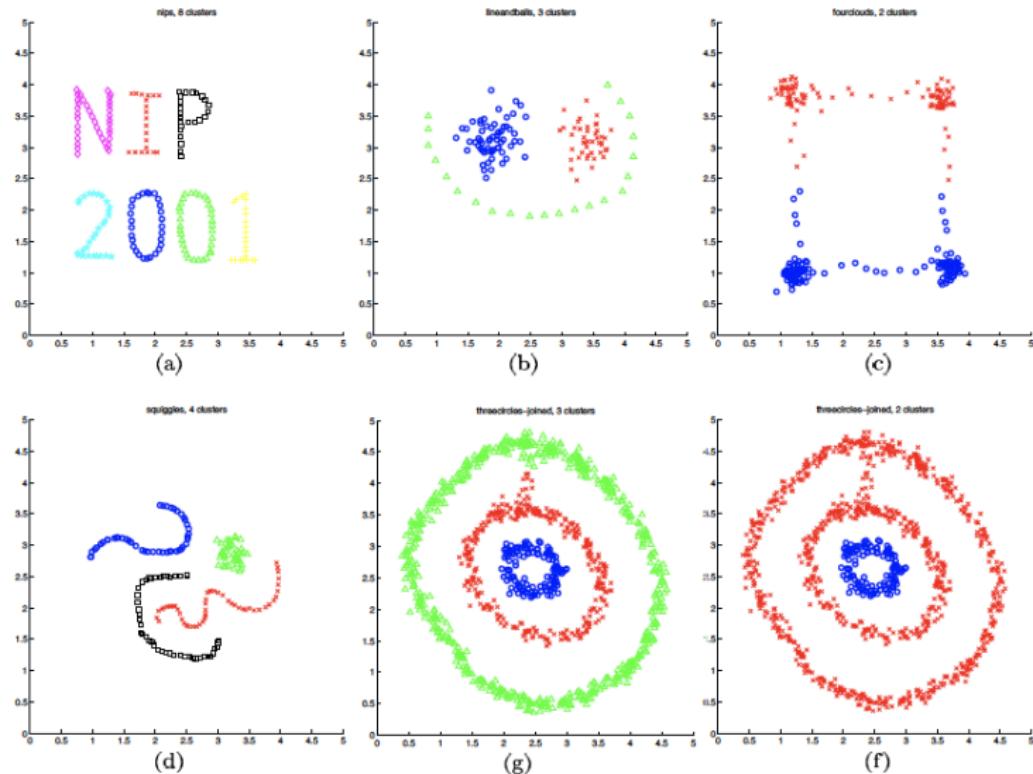
$$\text{RatioCut}(A, B) = \frac{\text{cut}(A, B)}{|A|} + \frac{\text{cut}(A, B)}{|B|}$$

$$Ncut(A, B) = \frac{\text{cut}(A, B)}{\text{Vol}(A)} + \frac{\text{cut}(A, B)}{\text{Vol}(B)}$$

$|A| = \# \text{ of vertices in } A; \text{Vol}(A) = \sum_{i \in A} d_i$

- These criterion can be directly extended for general  $k$
- Direct optimization of RatioCut and Ncut is NP hard. Approximations are used instead.
- Approximation of Ncut: Normalized spectral Clustering
- Approximation of RatioCut: Unnormalized spectral Clustering

# 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

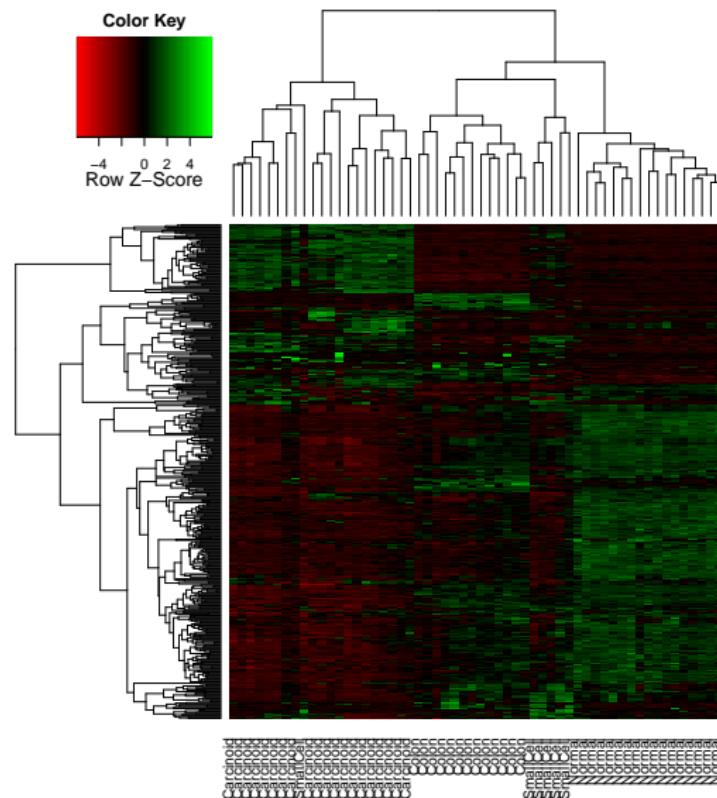
- Idea: Find groups of BOTH observations & features.
- Like clustering both rows and columns of data matrix.

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 & Bi-clustering

# Convex Clustering

Motivation:

- Clustering algorithms yield local solutions, dependent on initializations, unstable results.
- Can we formulate a *convex* method for clustering that will yield a **unique & global** solution?

Idea: Fuse mean column vectors together to form clusters.

# Convex Clustering

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

- $\mathbf{x}_i$  - each observation ( $p$ -vector).
- $\mathbf{u}_i$  - cluster centroid for each observation.

Convex fusion penalty shrinks centroids together!

*Pelckmans et al. 2005; Lindsten et al. 2011; Hocking et al. 2011.*

# Convex Clustering

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

- $\frac{1}{2} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{u}_i\|_2^2$  - K-means loss function!
- $\sum_{i < j} \mathbf{w}_{ij} \|\mathbf{u}_i - \mathbf{u}_j\|_2$  - Fusion as with Hierarchical!

# Convex Clustering

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

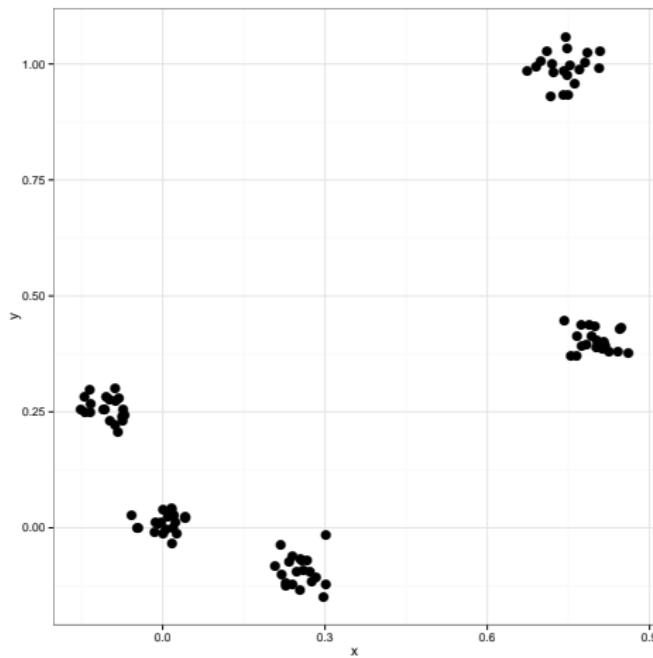
- $\lambda$  controls BOTH cluster assignments & number of clusters.
  - ▶  $\lambda = 0$  - each observation is its own cluster.
  - ▶  $\lambda$  larger - column means begin to coalesce together into clusters.
  - ▶  $\lambda$  very large - all observations fused into one cluster.

# Convex Clustering

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

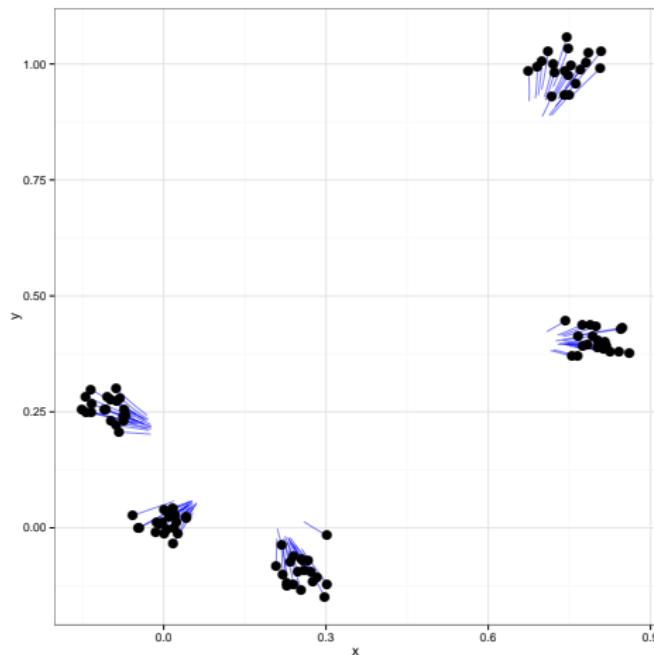
- **Weights** - sparse, proportional to distance between observations, and pre-assigned.
- Algorithm: Alternating Minimization Algorithm.
- In R: `cvxclustr`.

# Convex Clustering Solution Path



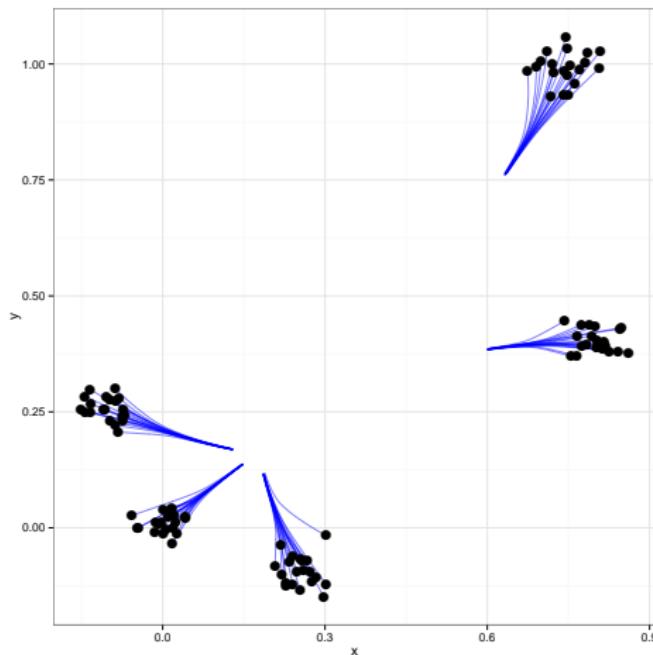
$\lambda = 0$

# Convex Clustering Solution Path

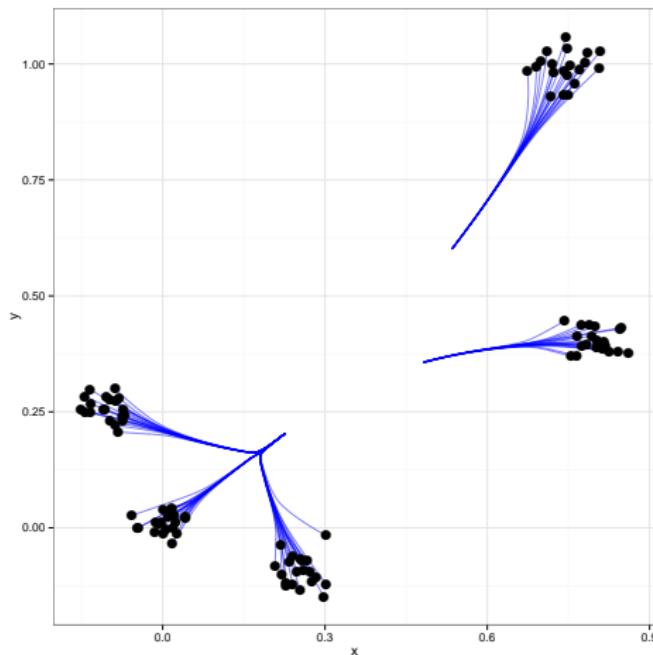


$\lambda$

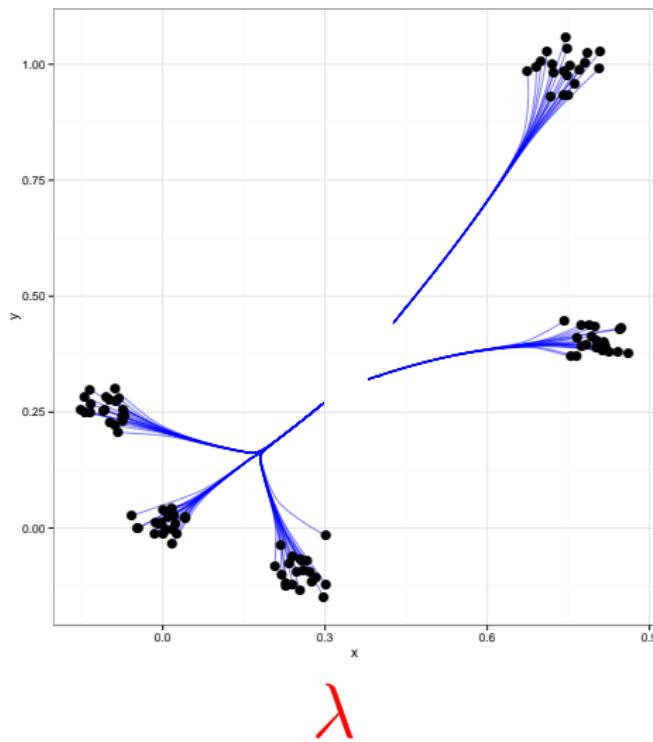
# Convex Clustering Solution Path

 $\lambda$

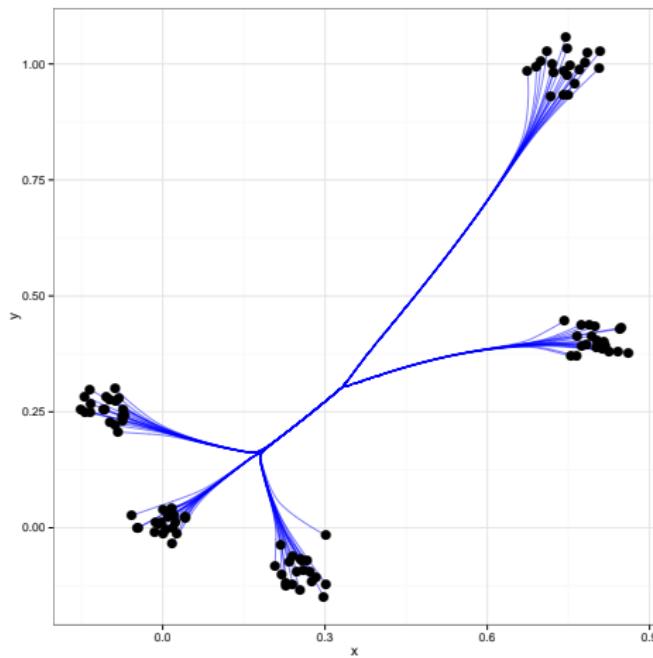
# Convex Clustering Solution Path

 $\lambda$

# Convex Clustering Solution Path

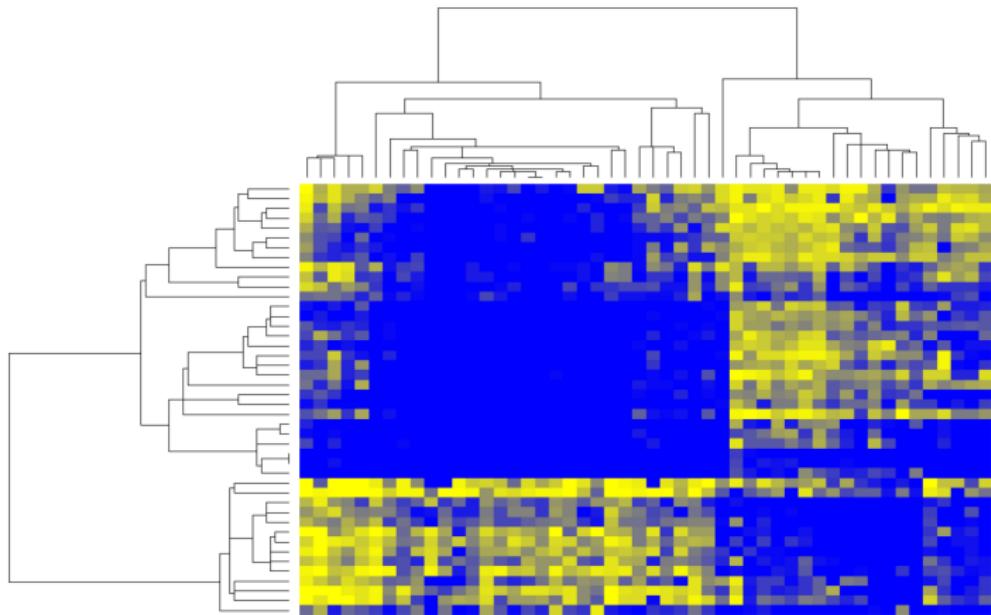


# Convex Clustering Solution Path



$\lambda$

# Convex Biclustering



# Convex Biclustering

$$\begin{aligned} \underset{\mathbf{U}}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{X} - \mathbf{U}\|_F^2 + \lambda \left( \sum_{i < j} w_{ij} \|\mathbf{U}_{i\cdot} - \mathbf{U}_{j\cdot}\|_2 \right. \\ & \left. + \sum_{l < k} \tilde{w}_{lk} \|\mathbf{U}_{\cdot l} - \mathbf{U}_{\cdot k}\|_2 \right) \end{aligned}$$

- Checkerboard-like pattern: every data point  $X_{ij}$  has its own bicluster centroid  $U_{ij}$ .
- Simultaneously fuses **row centroids** AND **column centroids** to yield biclusters!

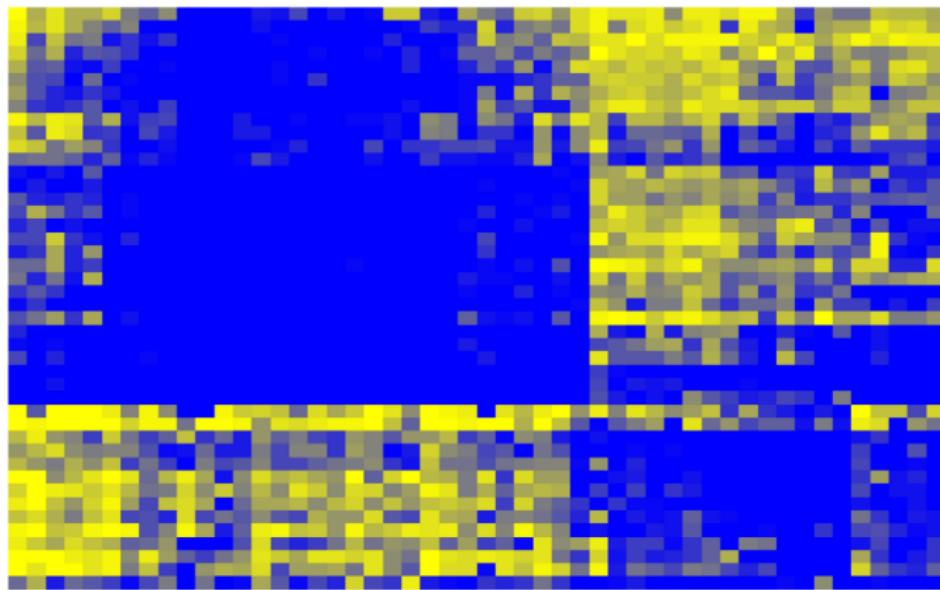
Chi, Allen, and Baraniuk, 2017

# Convex Biclustering

$$\begin{aligned} \underset{\mathbf{U}}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{X} - \mathbf{U}\|_F^2 + \lambda \left( \sum_{i < j} w_{ij} \|\mathbf{U}_{i\cdot} - \mathbf{U}_{j\cdot}\|_2 \right. \\ & \left. + \sum_{l < k} \tilde{w}_{lk} \|\mathbf{U}_{\cdot l} - \mathbf{U}_{\cdot k}\|_2 \right) \end{aligned}$$

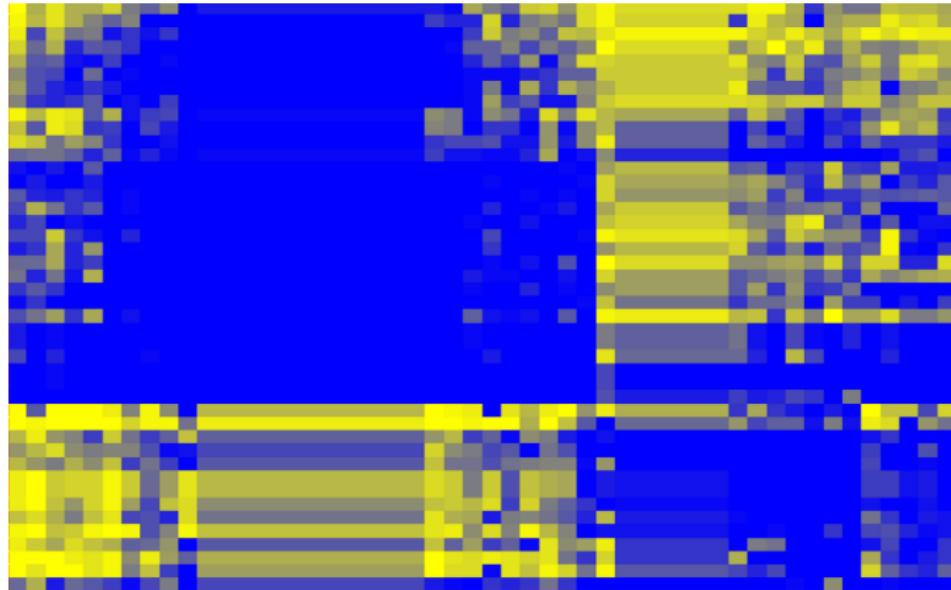
- $\lambda$  controls BOTH bicluster assignments and # of biclusters.
- Weights similar to convex clustering.
  - Must sum to  $1/\sqrt{p}$  and  $1/\sqrt{n}$  to ensure the same fusion rate.
- Algorithm: Dystra-like Proximal Algorithm + AMA.
- In R: `cvxbiclustr`.

# Convex Biclustering Solution Path



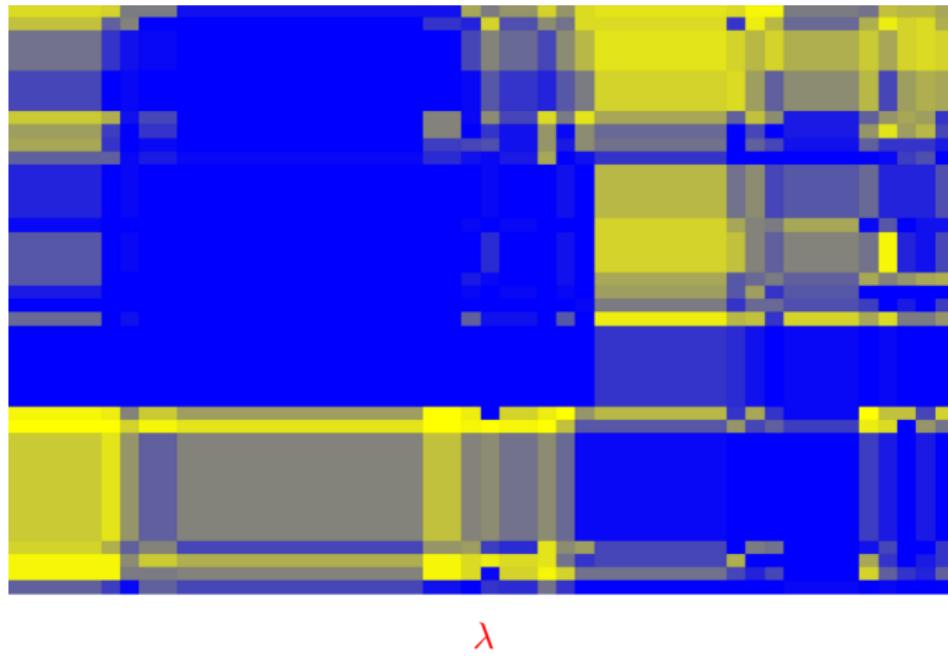
$\lambda = 0$

# Convex Biclustering Solution Path

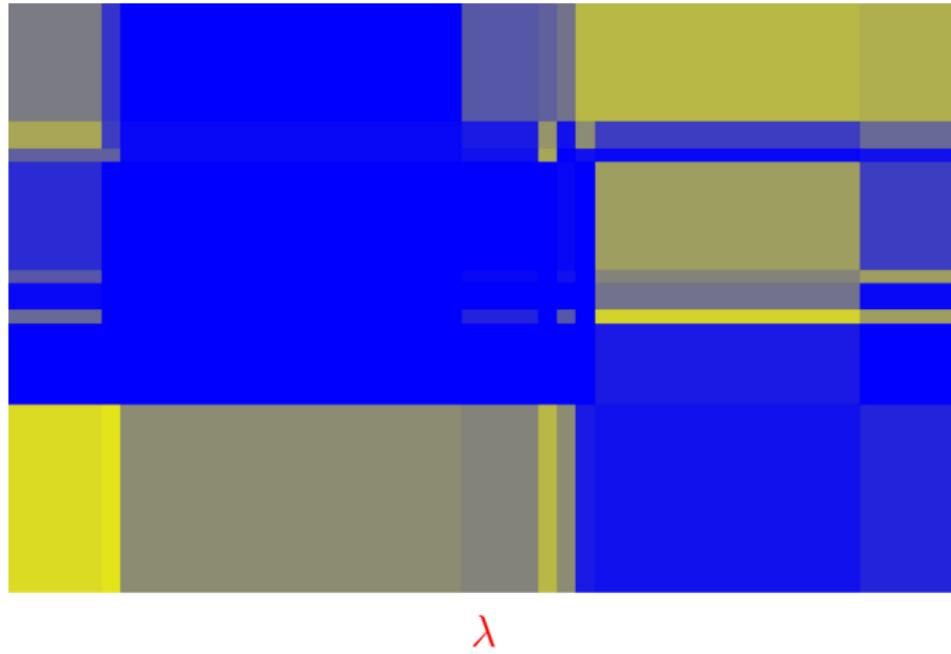


$\lambda$

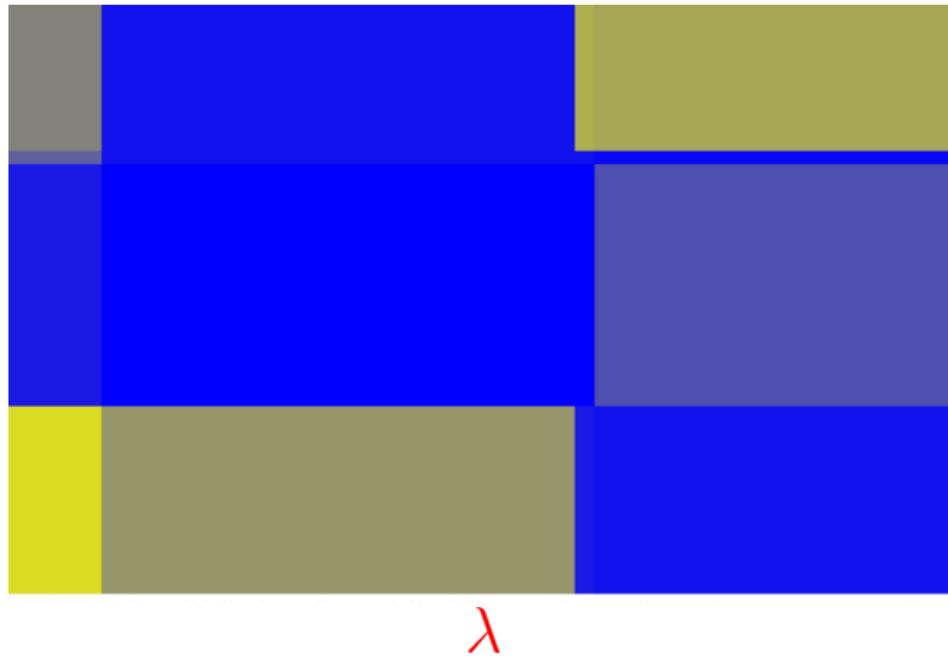
# Convex Biclustering Solution Path



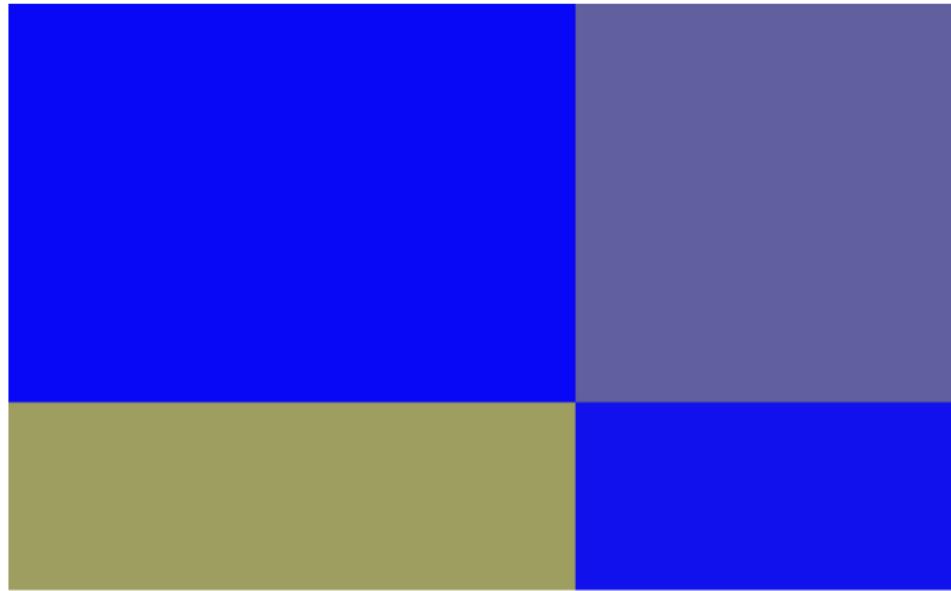
# Convex Biclustering Solution Path



# Convex Biclustering Solution Path



# Convex Biclustering Solution Path



# Convex Biclustering Solution Path



$\lambda$

# Convex Clustering & Biclustering

## Strengths:

- Unique, global solution.
- Stable solution.
- (Mostly) Nested family of clustering solutions.
- One tuning parameter - can be selected in a data-driven manner.

## Weaknesses:

- Performance can depend on choice of weights.
- Slower iterative algorithms.

clustRviz in R.

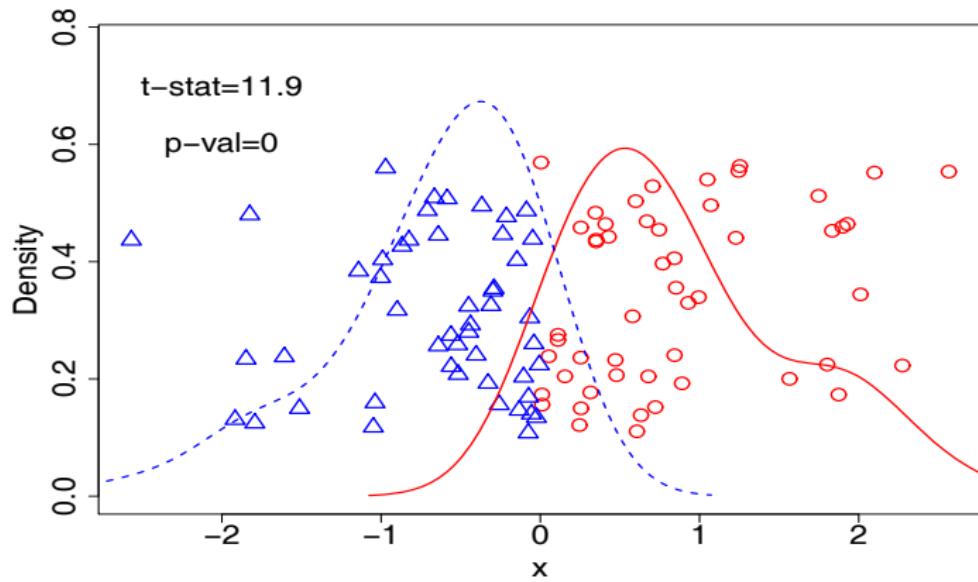
# Significance of Clustering

## Significance of Clustering (SigClust)

# Question of Cluster Significance

- When clusters seem to appear, e.g. found by clustering method, how do we know they are really there?
- How to assess significance?
- High dimensional data in modern data analysis make it more challenging to assess significance
- Need a more formal definition of clusters

# Simple Gaussian Example



## Lessons of Simple Gaussian Example

- Clearly only 1 Cluster in this Example.
- Random relabelling T-statistic is not significant, but extreme T-statistic is strongly significant.
- This comes from clustering operation
- Conclude sub-populations are different
- Message from the toy example: sub-populations are not the same as clusters really there.

## Definition of Cluster

- No universal definition
- SigClust definition of a cluster: data coming from a single Gaussian distribution (may not be standard spherical normal).
- Gaussian mixtures have similar definitions (Titterington et al. (1985), McLachlan and Basford (1988), etc.)
- Main difference: SigClust does not attempt to estimate full parameters of Gaussian components of mixture models.

# Statistical Significance of Clustering (SigClust)

- Significance of clustering (hypothesis of testing based on Gaussian definition of clusters: Liu et al. (2008, JASA), Huang et al. (2015, JCGS), Kimes et al. (2017, Biometrics))
- Formulate a test procedure
  - $H_0$  : data from a single Gaussian distribution
  - $H_1$  : data came from  $d$ -dimensional non Gaussian distribution.
- Test statistic: 2-means cluster index
- Estimate null distribution from Monte Carlo simulation
- Significance judged by p-value
- Key points: **Rotation Invariance of the test statistic** (the cluster index) & **Factor Model** (simplify parameter estimation for HDLSS data)

## 2-means Cluster Index

- $CI_2 = \frac{\sum_{k=1}^2 \sum_{j \in C_k} ||x_j - \bar{x}^{(k)}||^2}{\sum_{j=1}^n ||x_j - \bar{x}||^2}$
- Within Class Sum of Square Distances to Class Means divide (normalize) by Overall Sum of Squared Distance to Mean
- Puts on scale of proportions
- Small  $CI_2$  gives tight clustering (within SS contains little variation)
- Large  $CI_2$  gives poor clustering (within SS contains most of variation)

# SigClust Gaussian null distribution

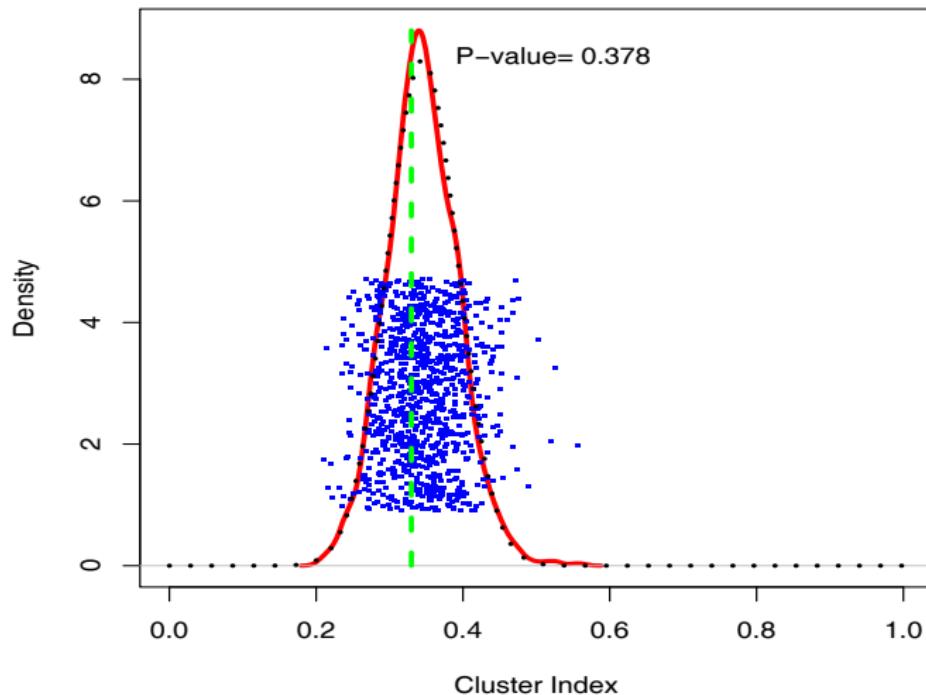
- Key step in SigClust is to estimate null distribution  $N(\mu, \Sigma)$
- $C\ell_2$  is location-invariant  $\rightarrow \mu = 0$
- Estimation of  $\Sigma = M\Lambda M^T$  is crucial in HDLSS data  
 $C\ell_2$  is orthogonal rotation-invariant:  $\rightarrow$  only need to estimate diagonal matrix  $\Lambda$   
(If  $z_i = Mx_i$  and  $M^{-1} = M^T$ , then  $\|z_i - z_I\|^2 = \|x_i - x_I\|^2$ .)
- Model covariance as:  $\Lambda = \Lambda_B + \sigma_N^2 \times I_d$  (Biology + Noise)
  - ▶  $\Lambda_B$  is “fairly low dimensional”
  - ▶  $\sigma_N^2$  is estimated from background noise  $\hat{\sigma}_N = \frac{\text{MAD}_{d \times n \text{ data set}}}{\text{MAD}_{N(0,1)}}$
- Estimated Null distribution using univariate Gaussian distributions.

# Procedures for the SigClust

- Step 1: Calculate 2-means cluster index for original dataset.
- Step 2: Get  $\hat{\sigma}_N^2$  using all entries from original data matrix.
- Step 3: Simulate data from null distribution:  $(x_1, \dots, x_d)$  are independent with  $x_j \sim N(0, \tilde{\lambda}_j)$  (Liu et al., 08, Huang et al., 14).
- Step 4: Perform 2-means clustering on simulated data from Step 3 and calculate corresponding 2-means cluster index.
- Step 5: Repeat Steps 3 and 4  $N_{Sim}$  times to obtain an empirical distribution of cluster index based on  $H_0$ .
- Step 6: Using cluster indices of the simulated data, calculate a p-value for the cluster index of the original dataset.

# SigClust Results on the Toy Example

Simulated Null Cluster Index Distribution



# Two real cancer data sets from The Cancer Genome Atlas Network (TCGA)

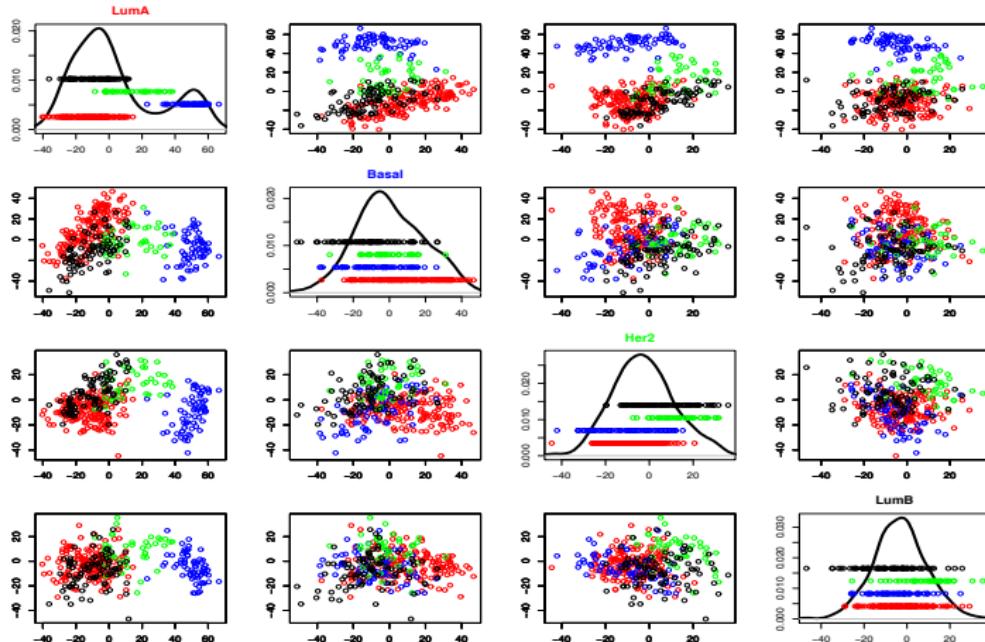
Glioblastoma Multiforme (GBM):

- $n = 383$ ,  $d = 2727$
- 117 Mesenchymal, 69 Neural, 96 Proneural, and 101 Classical

Breast Cancer data (BRCA):

- $n = 348$ ,  $d = 4000$
- 154 LumA, 81 LumB, 42 Her2, and 66 Basal

## PCA projection scatter plot view of the BRCA data

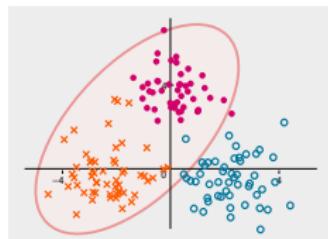


GBM	MES.CL	MES.PN	MES.NL	CL.PN	CL.NL	PN.NL
Sample	0.9	0	0	0	0	0
Hard	$< 10^{-4}$	0	0	0	0	0
combined	$10^{-3}$	0	0	0	0	0
BRCA	LA.B	LA.H	LA.LB	B.H	B.LB	H.LB
Sample	$< 10^{-4}$	0.8	1	0.01	$10^{-3}$	0.9
Hard	0	$< 10^{-4}$	0.9	0	0	0.06
combined	0	0.03	1	$< 10^{-4}$	0	0.9

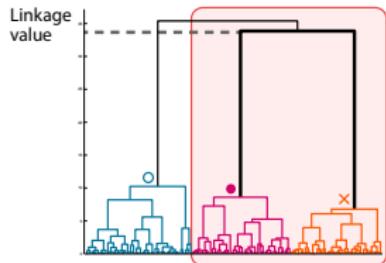
sigclust and hsigclust in R.

# Hierarchical SigClust (Kimes et al., 2017 Biometrics)

A apply hierarchical clustering

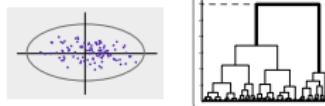


B Ward's Linkage



C simulate null data and apply same hierarchical algorithm

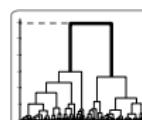
sim 1



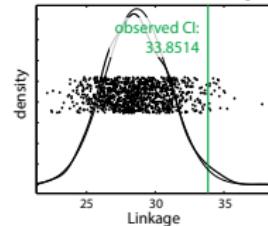
sim 2



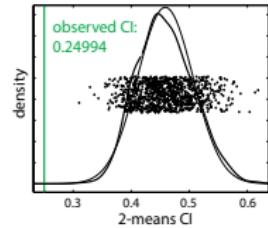
...  
sim 1000



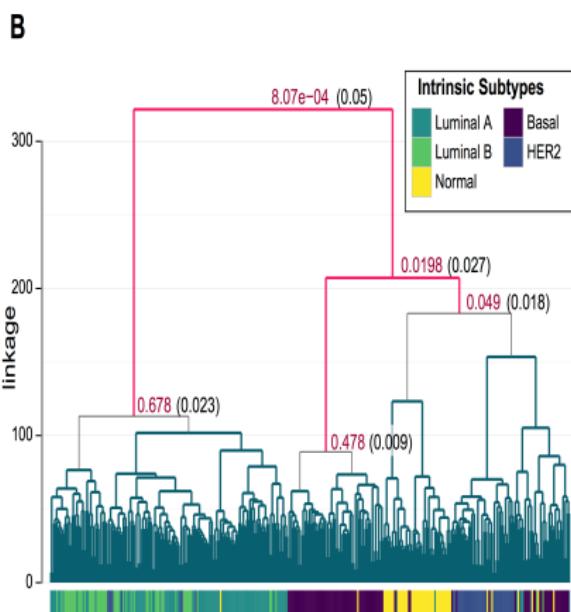
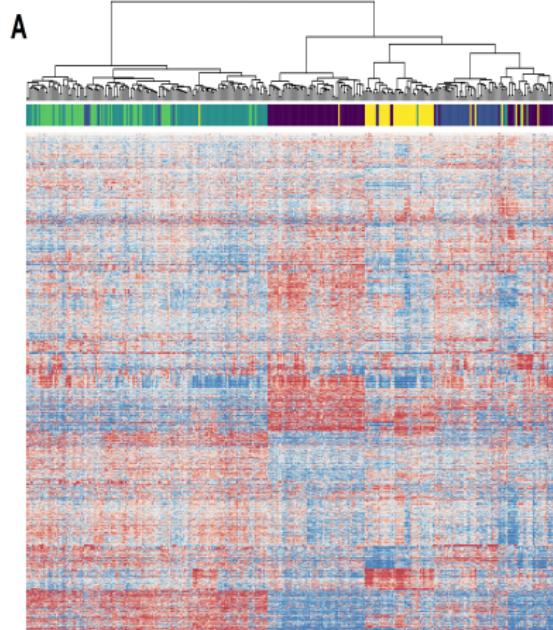
D 1000 Gaussian Simulated Linkages



1000 Gaussian Simulated 2-means Cls



# Hierarchical SigClust



# References

Textbooks:

- Elements of Statistical Learning by Hastie, Tibshirani & Friedman.  
<http://statweb.stanford.edu/~tibs/ElemStatLearn/>
- Introduction to Statistical Learning by James, Witten, Tibshirani & Hastie.  
<http://www-bcf.usc.edu/~gareth/ISL/>

Some of the figures in this presentation are taken from these two textbooks with permission from the authors.