Chapter 9. Clustering Analysis

Wei Pan

Division of Biostatistics, School of Public Health, University of Minnesota,
Minneapolis, MN 55455
Email: weip@biostat.umn.edu

PubH 7475/8475

Outline

- Introduction
- Hierachical clustering
- Combinatorial algorithms
- K-means clustering
- K-medoids clustering
- Mixture model-based clustering
- Spectral clustering
- Other methods: kernel K-means, PCA, ...
- Practical issues
  # of clusters, stability of clusters,...
- Big Data

Introduction

- Given: $X_i = (X_{i1}, ..., X_{ip})'$, $i = 1, ..., n$.
- Goal: Cluster or group together those X_i's that are "similar" to each other;
  Or, predict X_i's class Y_i with no training info on Y's.
- Unsupervised learning, class discovery,...
- Ref: 1. textbook, Chapter 14;
  2. A.D. Gordon (1999), *Classification*, Chapman&Hall/CRC;
  3. A. Kaufman & P. Rousseeuw (1990). *Finding groups in data: An introduction to cluster analysis*, Wiley;
  4. G. McLachlan, D. Peel (2000). *Finite Mixture Models*, Wiley;
  5. Many many papers...

- Define a metric of distance (or similarity):

$$d(X_i, X_j) = \sum_{k=1}^{p} w_k d_k(X_{ik}, X_{jk})$$

  - X_{ik} quantitative: d_k can be Euclidean distance, absolute distance, Pearson correlation, etc.
  - X_{ik} ordinal: possibly coded as $(i - 1/2)/M$ (or simply as i?) for $i = 1, ..., M$; then treated as quantitative.
  - X_{ik} categorical: specify $L_{l,m} = d_k(l, m)$ based on subject-matter knowledge; 0-1 loss is commonly used.
  - $w_k = 1$ for all k commonly used, but it may not treat each variable (or attribute) equally!
    standardize each variable to have var=1, but see Fig 14.5.
  - Distance $\leftrightarrow$ similarity, e.g. $sim = 1 - d$.

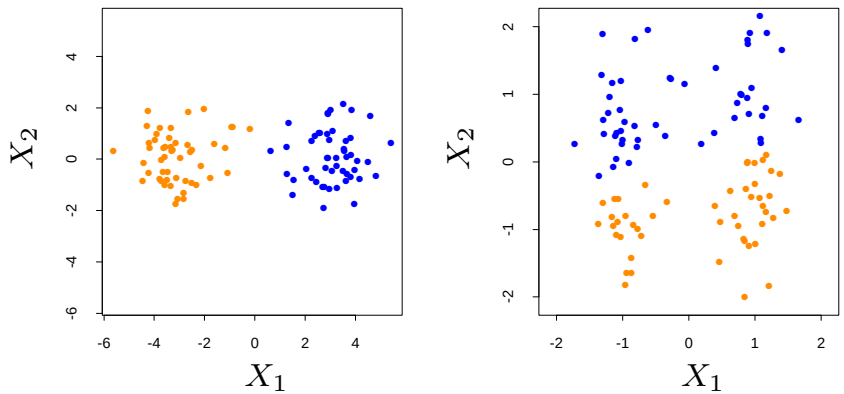


FIGURE 14.5. *Simulated data: on the left, K-means clustering (with $K=2$) has been applied to the raw data. The two colors indicate the cluster memberships. On the right, the features were first standardized before clustering. This is equivalent to using feature weights $1/[2 \cdot \mathrm{var}(X_j)]$. The standardization has obscured the two well-separated groups. Note that each plot uses the same units in the horizontal and vertical axes.*

Hierachical Clustering

- A dendrogram (an upside-down tree):
  Leaves represent observations X_i's; each subtree represents a group/cluster, and the height of the subtree represents the degree of dissimilarity within the group.
- Fig 14.12

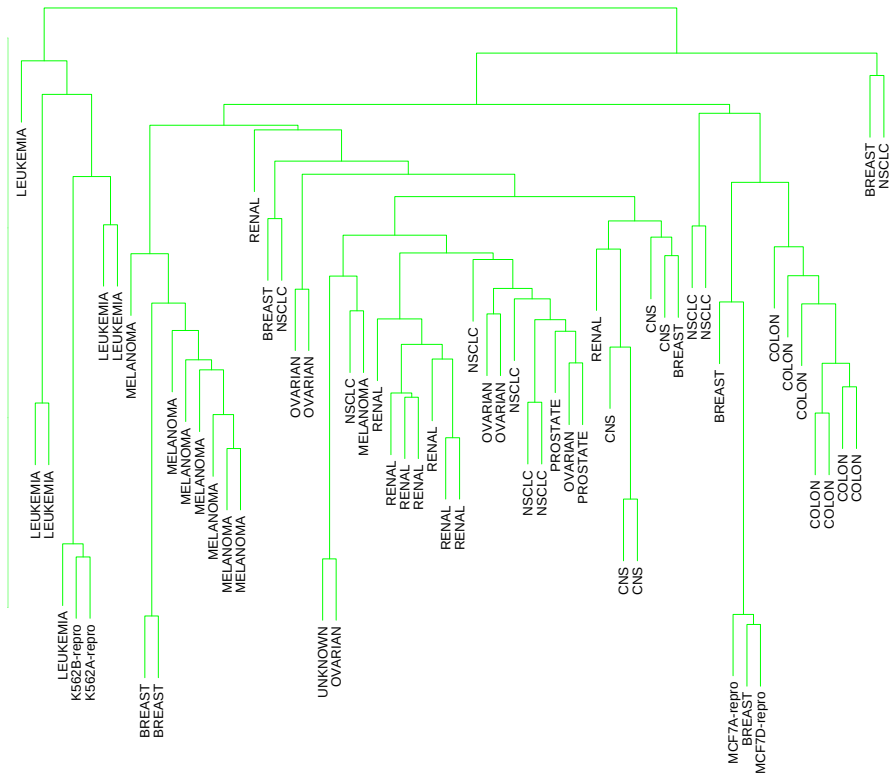


FIGURE 14.12. *Dendrogram from agglomerative hierarchical clustering with average linkage to the human tumor microarray data.*

Bottom-up (agglomerative) algorithm

given: a set of observations $\{X_1, ..., X_n\}$.
for $i := 1$ to n do
    $c_i := \{X_i\}$ /* each obs is initially a cluster */
$C := \{c_1, ..., c_n\}$
$j := n + 1$
while $|C| > 1$
    $(c_a, c_b) := argmax_{(c_u, c_v)} sim(c_u, c_v)$
        /* find most similar pair */
    $c_j := c_a \cup c_b$ /* combine to generate a new cluster*/
    $C := [C - \{c_a, c_b\}] \cup c_j$
    $j := j + 1$

- Similarity of two clusters
  Similarity of two clusters can be defined in three ways:
  - *single link*: similarity of two most similar members
    $sim(C_1, C_2) = max_{i \in C_1, j \in C_2} sim(Y_i, Y_j)$
  - *complete link*: similarity of two least similar members
    $sim(C_1, C_2) = min_{i \in C_1, j \in C_2} sim(Y_i, Y_j)$
  - *average link*: average similarity b/w two members
    $sim(C_1, C_2) = ave_{i \in C_1, j \in C_2} sim(Y_i, Y_j)$
- R: `hclust()`

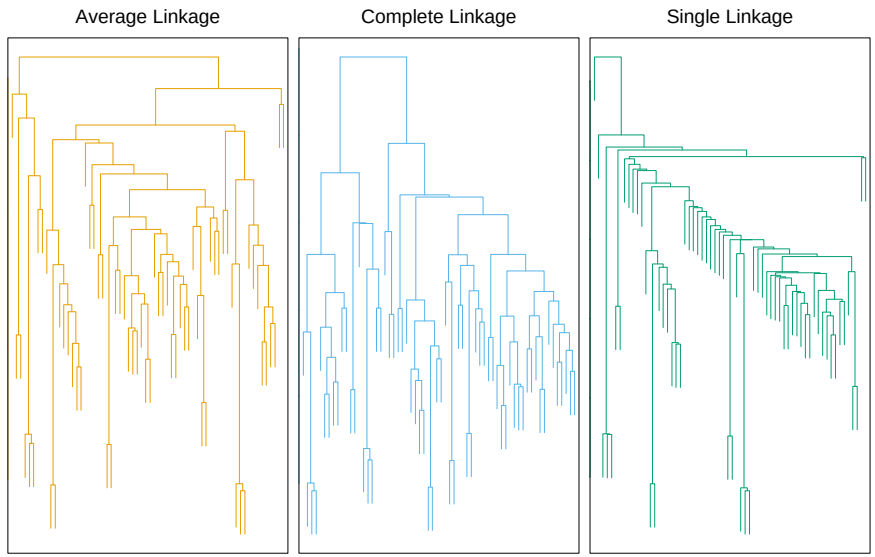


FIGURE 14.13. *Dendrograms from agglomerative hierarchical clustering of human tumor microarray data.*

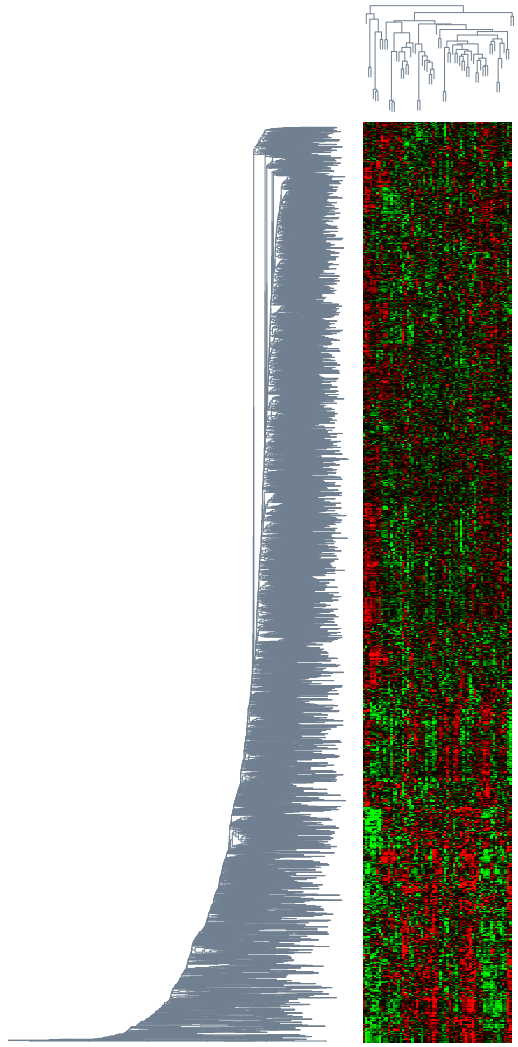


FIGURE 14.14. *DNA microarray data: average linkage hierarchical clustering has been applied independently to the rows (genes) and columns (samples), determining the ordering of the rows and columns (see*

Combinatorial Algorithms

- No probability model; group observations to min/max a criterion
- Clustering: find a mapping C: $\{1, 2, ..., n\} \to \{1, ..., K\}$, $K < n$
- A criterion

$$W(C) = \frac{1}{2} \sum_{c=1}^{K} \sum_{C(i)=c} \sum_{C(j)=c} d(X_i, X_j)$$

- $T = \frac{1}{2} \sum_{i=1}^{K} \sum_{j=1}^{K} d(X_i, X_j) = W(C) + B(C)$,

$$B(C) = \frac{1}{2} \sum_{c=1}^{K} \sum_{C(i)=c} \sum_{C(j)\neq c} d(X_i, X_j)$$

- Min $B(C) \leftrightarrow$ Max $W(C)$
- Algorithms: search all possible C to find $C_0 = argmin_C W(C)$

- Only feasible for small n and K: # of possible C's

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

  E.g. $S(10,4) = 34105$, $S(19,4) \approx 10^{10}$.
- Alternatives: iterative greedy search!

K-means Clustering

- Each observation is a point in a p-dim space
- Suppose we know/want to have K clusters
- First, (randomly) decide K cluster centers, M_k
- Then, iterate the two steps:
  - assignment of each obs i to a cluster $C(i) = argmin_k ||X_i - M_k||^2$;
  - a new cluster center is the mean of obs's in each cluster $M_k = Ave_{C(i)=k} X_i$.
- Euclidean distance $d()$ is used
- May stop at a local minimum for $W(C)$; multiple tries
- R: kmeans()
- +: simple and intuitive
- -: Euclidean distance $\Longrightarrow$ 1) sensitive to outliers; 2) if X_{ij} is categorical then ?
- Assumptions: really assumption-free?

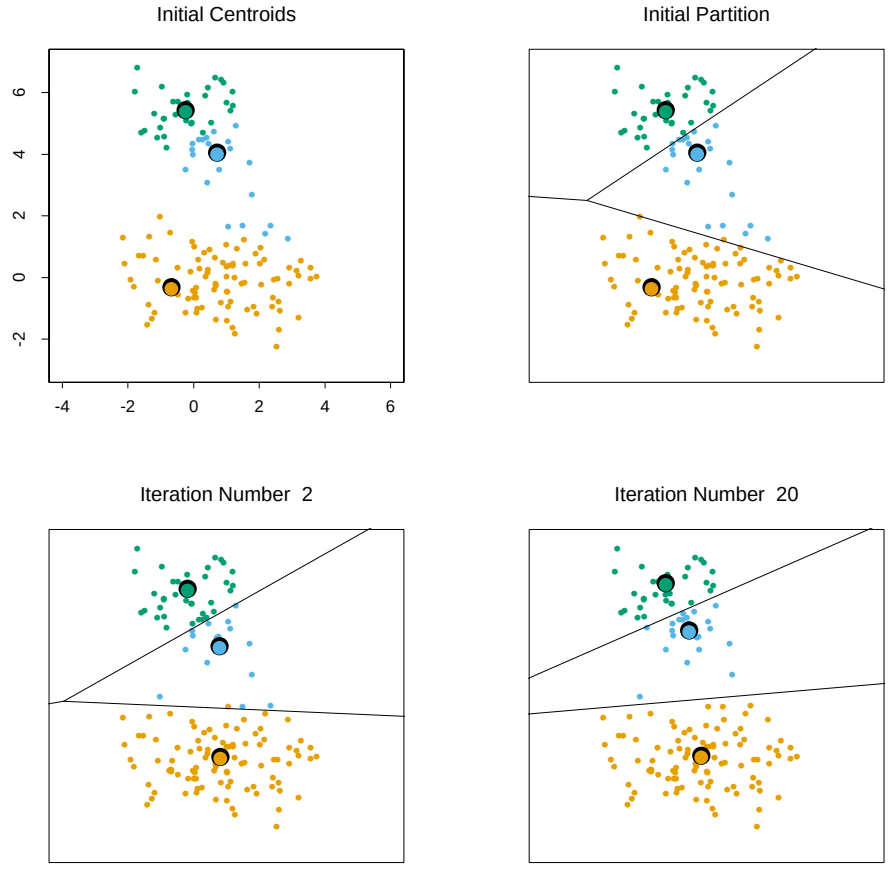


FIGURE 14.6. *Successive iterations of the K-means clustering algorithm for the simulated data of Figure 14.4.*

K-medoids Clustering

- Similar to K-means; rather than using the mean of a cluster to represent the cluster, use an observation within it! why?
- First, (randomly) start with a C
- Find $M_k = X_{i_k^*}$ with

$$i_k^* = argmin_{\{i:C(i)=k\}} \sum_{C(j)=k} d(x_i, x_j);$$

- Update C: $C(i) = argmin_k d(X_i, M_k)$.
- Repeat the above 2 steps until convergence
- R: package `cluster`, containing pam() for partitioning around medoids, clara() for large datasets with pam, silhouette() for calculating silhouette widths, diana() for divisive hierarchical clustering, etc.
- Both K-means and K-medoids: not a probabilistic method; "hard", not "soft", grouping $\Longrightarrow$ An alternative:

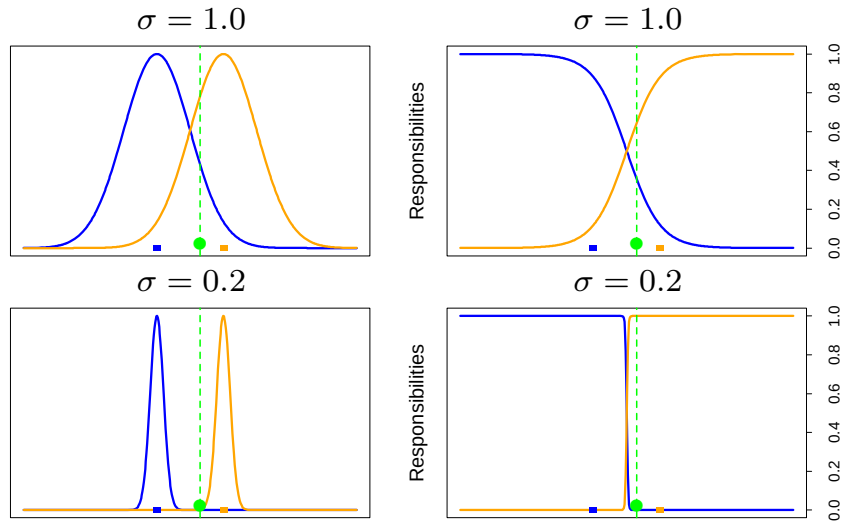


FIGURE 14.7. *(Left panels:) two Gaussian densities $g_0(x)$ and $g_1(x)$ (blue and orange) on the real line, and a single data point (green dot) at $x = 0.5$. The colored squares are plotted at $x = -1.0$ and $x = 1.0$, the means of each density. (Right panels:) the relative densities $g_0(x)/(g_0(x) + g_1(x))$ and $g_1(x)/(g_0(x) + g_1(x))$, called the "responsibilities" of each cluster, for this data point. In the top panels, the Gaussian standard deviation $\sigma = 1.0$; in the bottom panels $\sigma = 0.2$. The EM algorithm uses these responsibilities to make a "soft" assignment of each data point to each of the two clusters. When σ is fairly large, the responsibilities can be near 0.5 (they are 0.36 and 0.64 in the top right panel). As $\sigma \to 0$, the responsibilities $\to 1$, for the cluster center closest to the target point, and 0 for all other clusters. This "hard" assignment is seen in the bottom right panel.*

Mixture Model-based Clustering

- Can use mixture of Poissons or binomials if needed (McLachlan & Peel 2000).
- Assume each X_i is from a mixture of Normal distributions with pdf

$$f(x; \Phi_K) = \sum_{k=1}^{K} \pi_k \phi(x; \mu_k, V_k)$$

  where $\phi(x; \mu_k, V_k)$ is the pdf of $N(\mu_k, V_k)$.
- Each component k is a cluster with a prior prob π_k, $\sum_{k=1}^{K} \pi_k = 1$.
- For a fixed K, use the EM to estimate Φ_K (to obtain MLE).

- Try various values of $K = 1, 2, ...,$ then use AIC/BIC to select the one with the first local (or global?) minimum.

$$\log L(\Phi_K) = \sum_{i=1}^{n} \log f(X_i; \Phi_K)$$

$$AIC = -2 \log L(\hat{\Phi}_K) + 2\nu_K$$

$$BIC = -2 \log L(\hat{\Phi}_K) + \nu_K \log(n)$$

  where ν_K is #para. in Φ_K.
- Or, test H_0: $K = k_0$ vs H_A: $K = k_0 + 1$; use bootstrap (McLachlan)

EM algorithm

Given: a set of observations $\{X_1, ..., X_n\}$.
Init $r = 1$; $\pi_k^{(0)}$, $\mu_k^{(0)}$'s and $V_k^{(0)}$'s.
While (not converged) do

For all $i = 1, ..., n$ and $r = 1, 2, ...$ do

$$\tau_{ki}^{(r)} = \frac{\pi_k^{(r)} \phi(X_i; \mu_k^{(r)}, V_k^{(r)})}{f(X_i; \Phi^{(r)})}$$

/* τ_{ki} is posterior prob X_i in component k */

$$\pi_k^{(r+1)} = \sum_{i=1}^{n} \tau_{ki}^{(r)} / n$$

$$\mu_k^{(r+1)} = \sum_{i=1}^{n} \tau_{ki}^{(r)} X_i / \sum_{i=1}^{n} \tau_{ki}^{(r)}$$

$$V_k^{(r+1)} = \frac{\sum_{i=1}^{n} \tau_{ki}^{(r)} (X_i - \mu_k^{(r+1)})(X_i - \mu_k^{(r+1)})^T}{\sum_{i=1}^{n} \tau_{ki}^{(k)}}$$

$r := r + 1$

At end, each X_i is assigned to the component $C(i) = \arg\max_k \tau_{ki}$.

EM algorithm: derivation

- $Z_{ki} = I(X_i \text{ from component } k)$.
- Complete data log-likelihood

$$\log L_C = \sum_{i=1}^{n} \sum_{k=1}^{K} Z_{ki} \log[\pi_k \phi(X_i; \mu_k, V_k)].$$

- E-step:

$$Q(\Phi|\Phi^{(r)}) = E_{Z|X,\Phi^{(r)}} \log L_C = \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ki}^{(r)} \log[\pi_k \phi(X_i; \mu_k, V_k)].$$

- M-step:

$$\Phi^{(r+1)} = \arg\max_{\Phi} Q(\Phi|\Phi^{(r)})).$$

- Repeat the E- and M-steps with $r := r + 1$ until convergence.

- Non-convex: many local solutions; use good starting values and/or multi-tries.
- $+$: a cluster is a set of obs's from a Normal distribution–clear def; can model V_k and thus shape/size/orientation of clusters; probablistic
- $-$: why Normal?
  (try nonparametric clustering; find modes; see Li et al 2007.)
  Slow
  Requires cluster size $>=$ dim of X_i if no restriction on $V_k \Longrightarrow$ have to do variable selection or dim reduction if p is large
- K-means: a special case of Normal mixture model-based clustering by assuming all $V_k = \sigma^2 I$ (and all $\pi_k = 1/K$).
- R: package mclust

Implementation in `mclust`

Table: Table 1 in Fraley et al (2012) `http://www.stat.washington.edu/research/reports/2012/tr597.pdf`: Parameterizations of the covariance matrix V_k currently available in mclust for hierarchical clustering (HC) and/or EM for multidimensional data. (Y indicates availability.)
$A = diag(1, a_{22}, ..., a_{pp})$ is **diagonal** with $1 \geq a_{22} \geq ... \geq a_{pp} > 0$.

identifier	Model	HC	EM	Distribution	Volume	Shape	Orientation
E		Y	Y	(univariate)	equal		
V		Y	Y	(univariate)	variable		
EII	λI	Y	Y	Spherical	equal	equal	NA
VII	$\lambda_k I$	Y	Y	Spherical	variable	equal	NA
EEI	λA		Y	Diagonal	equal	equal	coordinate axes
VEI	$\lambda_k A$		Y	Diagonal	variable	equal	coordinate axes
EVI	λA_k		Y	Diagonal	equal	variable	coordinate axes
VVI	$\lambda_k A_k$		Y	Diagonal	variable	variable	coordinate axes
EEE	λDAD^T	Y	Y	Ellipsoidal	equal	equal	equal
EEV	$\lambda D_k A D_k^T$		Y	Ellipsoidal	equal	equal	variable
VEV	$\lambda_k D_k A D_k^T$		Y	Ellipsoidal	variable	equal	variable
VVV	$\lambda_k D_k A_k D_k^T$	Y	Y	Ellipsoidal	variable	variable	variable

Spectral clustering

- Given: a graph $G = (V, E)$ with nodes V and edges E.
  1) each obs is a node;
  2) binary edges $w_{ij} \in \{0, 1\}$, or weighted ones ($w_{ij} \geq 0$);
  3) with the usual data, need to construct a graph (e.g. v nearest neighbors, or a complete graph) based on their similarities, e.g., $W = (w_{ij})$ with $w_{ij} = k(X_i, X_j) = \exp(-||X_i - X_j||^2/2\sigma^2)$ and $w_{ii} = 0$.
  —-a kernel method!
- Goal: to partition the nodes into K groups.
  can be used in network community detection.
- Unnormalized graph Laplacian: $L_u = D - W$,
  $D = diag(d_1, ..., d_n)$ with node degrees $d_i = \sum_{j=1}^{n} w_{ij}$;
  $W = (w_{ij})$ is the weight/adjacency matrix; $w_{ii} = 0\ \forall i$.
- Normalized graph Laplacian: $L_n = I - D^{-1}W$,
  or, $L_s = I - D^{-1/2}WD^{-1/2}$.
- Several variants: based on each Laplacian.

Spectral clustering algorithm (Ng et al)

- Find the m eigenvectors $U_{n\times m}$ corresponding to the m **smallest** eigenvalues of L;
- (Optional?) Form matrix $N = (N_{ij})$ with $N_{ij} = U_{ij}/(\sum_{j=1}^{m} U_{ij}^2)^{1/2}$;
- Treating each row of N as an observation (corresponding to the original obs) and apply the K-means.
- Why? (8000) von Luxburg.
  Fig 14.29.
- Remark: the choice of the kernel (e.g. σ^2 in the radial basis kernel) and v-NN to form a graph very important!
- Remark: related to the (normalized) min cut algorithm (Zhang & Jordan 2008).
- R: function specc() in package kernlab.
  Other functions for kernel methods, e.g. kkmeans() for kernel k-means.

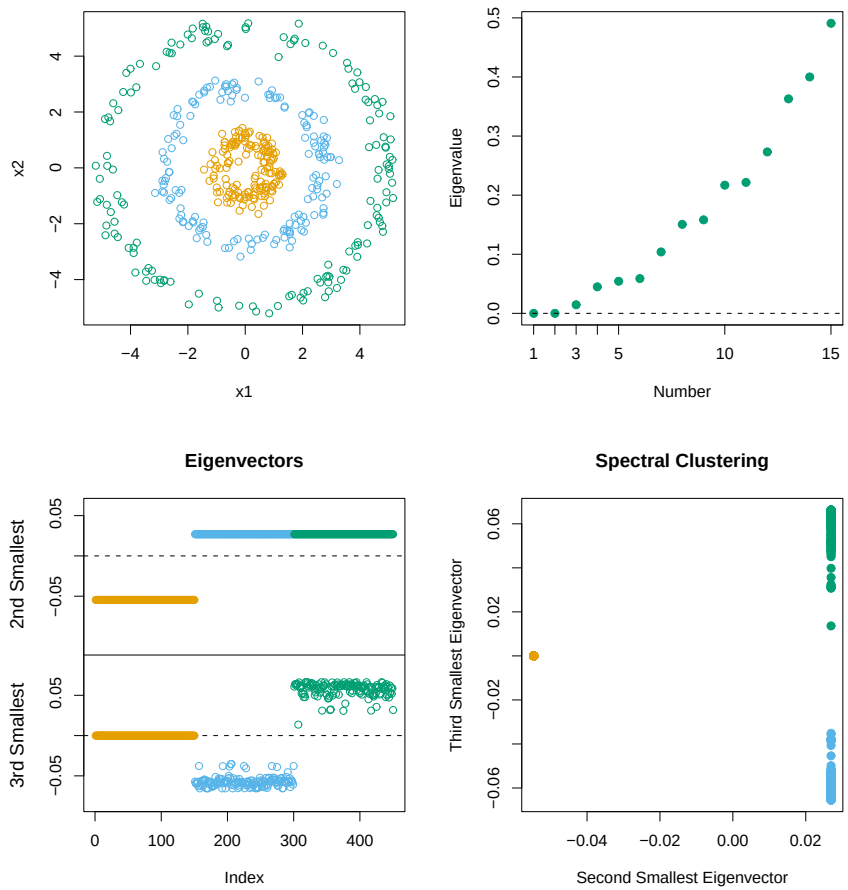


FIGURE 14.29. *Toy example illustrating spectral clustering. Data in top left are* 450 *points falling in three concentric clusters of* 150 *points each. The points are uniformly distributed in angle, with radius* $1, 2.8$ *and* 5 *in the three groups, and Gaussian noise with standard deviation* 0.25 *added to each point. Using a* $k = 10$ *nearest neighbor similarity graph, the eigen-*

(8000) Some properties of the Laplacian matrices (von Luxburg)

- Proposition. For any vector $f = (f_1, ..., f_n)'$, we have
  $f' L_u f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$,
  $f' L_s f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$.
- Remark: smoothing over a network; related to graph kernels (e.g. diffusion kernel).
- Proposition. The multiplicity k of the eigenvalue 0 of all L_u, L_n and L_s equals to the number of connected components A_1,...,A_k in the graph. For both L_u and L_n, the eigenspace of eigenvalue 0 is spanned by the indicator vectors $1_{A_1}, ..., 1_{A_k}$ of those components. For L_s, the eigenspace of eigenvalue 0 is spanned by the indicator vectors $D^{1/2} 1_{A_1}, ..., D^{1/2} 1_{A_k}$ of those components.
- Remark: theoretical foundation of spectral clustering.

(8000) Other Methods

- Hierarchical clustering: divisive (top-down) algorithm (p. 526);
- Self-Organizing Maps: a constrained version of K-means (section 14.4).
- PRclust (Pan et al 2013): formulated as penalized regression.
  R package `prclust` (Wu 2016, JMLR, 17(188), 1-25).
  Each X_i with its own centroid/mean μ_i;
  Cluster: shrink some μ_i's to be exactly the same;
  Objective function:

$$\sum_{i=1}^{n}(X_i - \mu_i)^2 + \lambda \sum_{i<j} TLP(||\mu_i - \mu_j||_2; \tau).$$

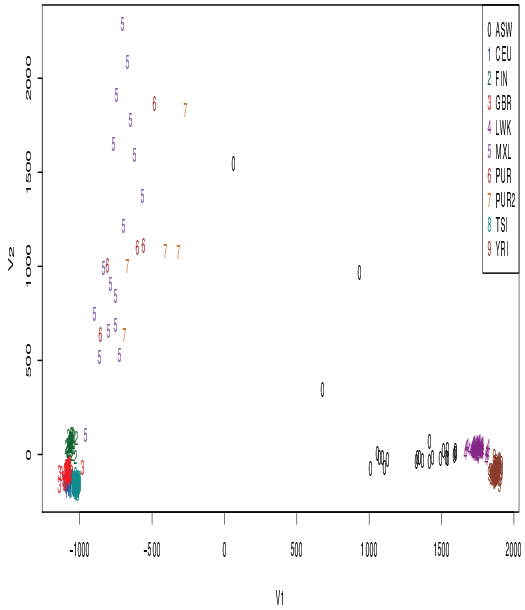
(8000) Other Methods: Kernel K-means

- Motivation: since K-means finds linear boundaries between clusters, in the presence of non-linear boundaries it may be better to work on non-linearly mapped $h(X_i)$'s (in a possibly higher dim space).
- The (naive) algorithm is the same as the K-means (except replacing X_i by $h(X_i)$).
- Kernel trick: as before, no need to specify $h(.)$ but a kernel $k(x, z) =< h(x), h(z) >$.
- Key: a center $M_C = \sum_{j \in C} h(X_j)/|C|$,
  $||h(X_i) - M_C||^2 =$
  $k(X_i, X_i) - 2\sum_{j \in c} k(X_i, X_j)/|C| + \sum_{j \neq l} k(X_j, X_l)/|C|^2$.
- Remark: related to spectral clustering; $K = L^+$. (Zhang & Jordan)
- R: kkmeans() in package kernlab.

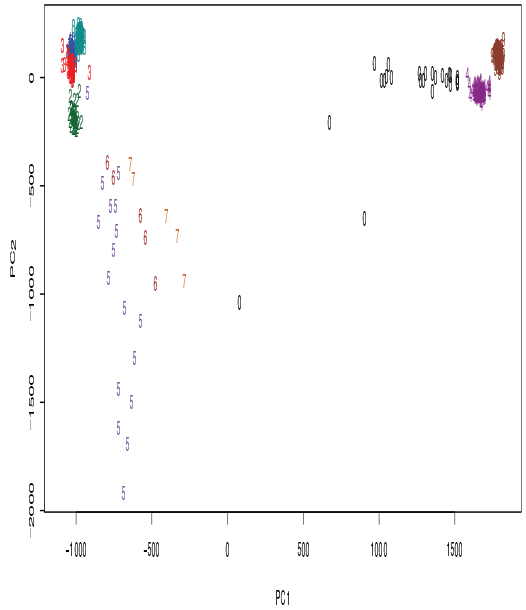
Other Methods: PCA

- PCA: dim reduction; why here?
- Population structure in human genetics: each person has a vector of 100,000s of SNPs (=0, 1 or 2) as X_i; X_i can reflect population/racial/ethnic group differences—-a possible confounder. Apply PCA (Zhang, Guan & Pan, 2013, Genet Epi): next two figures.
- Clustering?!
- See also Novembre et al (2008, Nature) "Genes mirror geography within Europe".
  `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2735096/`
- Other uses: PCA can be used to obtain good starting values for K-means (Xu et al 2015, Pattern Recognition Letter, 54, 50-55); K-means can be used to approx SVD for large datasets (...?).
- R: prcomp(), svd(), ...

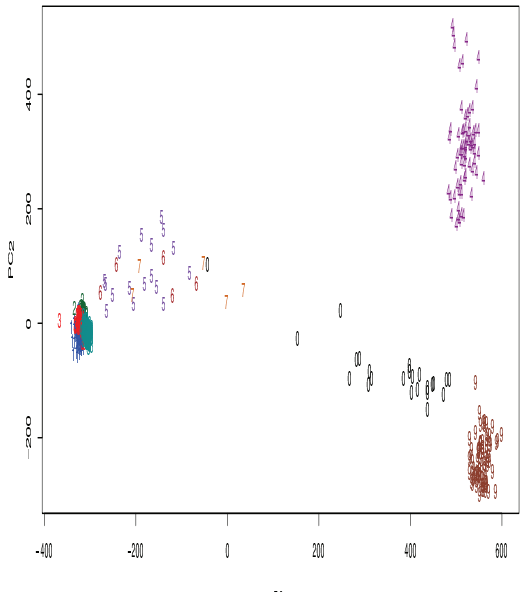
all variants



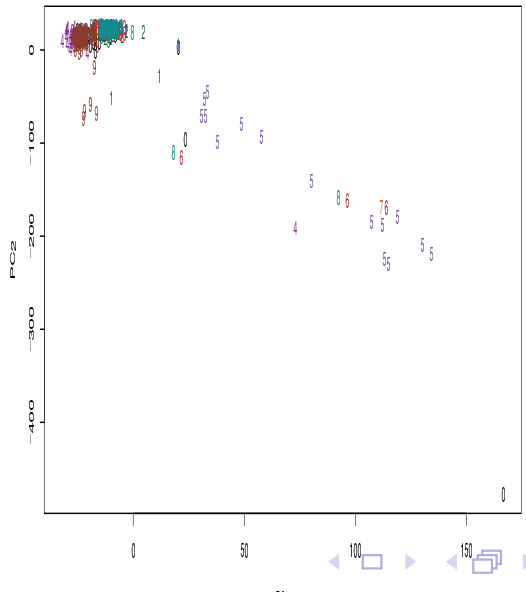
all CVs



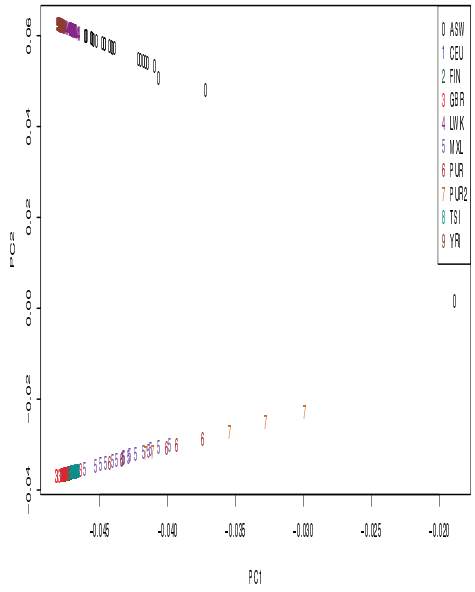
all LFVs



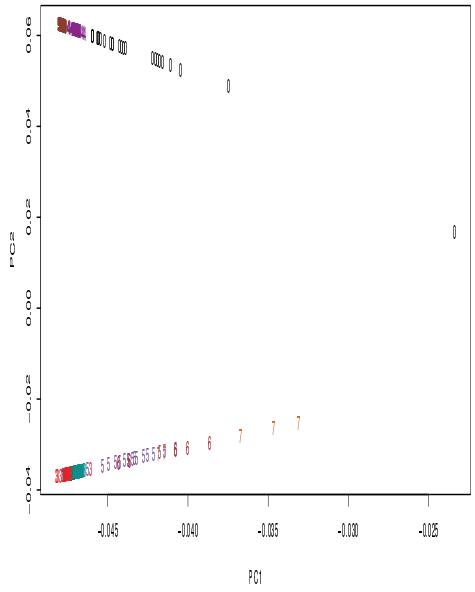
all RVs (zoom in)



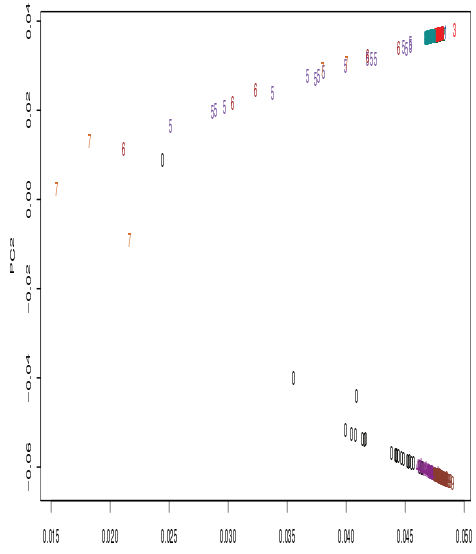
all variants



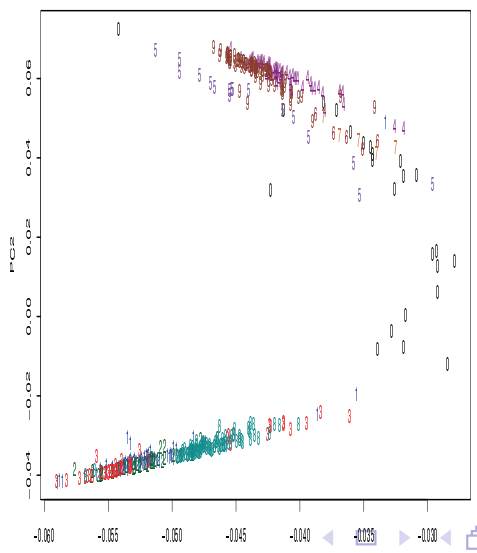
all CVs



all LFVs



all RVs



Other Methods: (8000) PCA $\approx$ K-means

- Conclusion: "principal components are the continuous solutions to the discrete cluster membership indicators to K-means clustering." (Ding & He 2004)
- Data: $X = (X_1, X_2, ..., X_n)$; WLOG assume $X1 = 0$.
- Review of PCA and SVD:
  Covariance $V = XX' = \sum_{k=1}^{r} d_k^2 u_k u_k'$,
  Gram (kernel) matrix $X'X = \sum_{k=1}^{r} d_k^2 v_k v_k'$,
  SVD $X = \sum_{k=1}^{r} d_k u_k v_k' = UDV'$, $|d_1| \geq |d_2| \geq ... \geq |d_r| > 0$,
  $U'U = I$, $V'V = I$.
  Principal directions: u_k's; Principal components: v_k's
- Eckart and Young (1936) Theorem: for any $0 < r_1 \leq r$,
  $\sum_{k=1}^{r_1} d_k u_k v_k' = \arg\min_{rank(Y)=r_1} ||X - Y||_F^2$.
- Denote $C = (C_1, ..., C_K)$ with each column $C_j \in R^p$ as a centroid; $H = (H_1, ..., H_n)$ with each column $H_j \in \{0, 1\}^K$, $H_{kj} = I(X_j \in \mathcal{C}_k)$ and $1'H_j = 1 \ \forall j$ (or, $H'H = I$ after normalized).
- K-means: $\min_{C,H} W = ||X - CH||_F^2$ s.t. H ...

Other Two Matrix Factorization Methods:

- Non-negative Matrix Factorization (NMF): given $X \geq 0$ (elementwise).
  $\min_{C,H} ||X - CH||_F^2$ s.t. $C \geq 0$, $H \geq 0$.
  1) Clustering property (as PCA for K-means);
  2) A "sum of parts" interpretation.
  R package: `NMF`.
  Ref: `https://en.wikipedia.org/wiki/Non-negative_matrix_factorization`
- Recommendation systems:
  R package: `recosystem`, `recommenderlab`, ...
  Ref:
  `https://developers.google.com/machine-learning/recommendation/collaborative/matrix`

Figure 3: Background modeling from video. Three frames from a 250 frame sequence taken in a lobby, with varying illumination [32]. (a) Original video M. (b)-(c) Low-rank $\hat{L}$ and sparse $\hat{S}$ obtained by PCP. (d)-(e) Low-rank and sparse components obtained by a competing approach based on alternating minimization of an m-estimator [47]. Again, convex programming yields a more appealing result despite using less prior information.

Figure: Fig 3, Candes, Li, Ma and Wright 2009.

(8000) Robust PCA

- Ref: Candes et al 2009.
- SVD: given an $n \times p$ data matrix X,
  minimize $||X - L||_F^2$
  subject to $\text{rank}(L) \leq k$.
- PCA: given an $n \times n$ data cov matrix X,
  minimize $||X - L||_F^2$
  subject to $\text{rank}(L) \leq k$.
- rPCA:
  minimize $||L||_* + \lambda||S||_1$
  subject to $L + S = X$.
  where $||L||_* = \sum_i \sigma_i(L)$ is the nuclear norm with $\sigma_i(L)$ as the ith eigen-value of L.
- rPCA2: Shen et al 2011. `http://www.caam.rice.edu/~zhang/reports/tr1102.pdf`
  minimize $||X - L||_1$ s.t. $L = UV$
  with U and V as $n \times k$ and $k \times n$ (and thus $\text{rank}(UV) = k$.

(8000) Other Methods:

- Variable selection (VS) for high-dim data:
  model-based clustering: add an L_1 (or other) penalty on μ_i's (Pan & Shen 2007); ...
  k-means: Sun, Wang & Fang (2012, EJS, 6, 148-167); ...
  sparse PCA (or SDA): add a penalty term in SVD (Shen & Huang 2008, *JMA*), or ...
- Consensus clustering (Monti et al 2003, ML, 91-118):
  unstability of clustering; analog of Bagging.
  R: ConsensusClusterPlus (Wilkerson & Hayes 2010).

Practical Issues

- How to select the number of clusters?
  Why is it difficult? see Fig 14.8.
  Stability or significance of clusters.
- Any clusters?
  - A global test: parametric bootstrap (McShane et al, 2002, Bioinformatics, 18(11):1462-9).

Practical Issues

- Any clusters?
  - H_0: a Normal distr (or a uniform or ...?).
  - (optional) Principal component analysis (PCA): use first 3 PC's for each obs; PC's are orthogonal
  - Under H_0, simulate data Y_i^b from a MVN; component-wise mean/var same as that of the data's PC's
  - For each obs Y_i, i) d_i is the distance from Y_i to its closest neighbor; ii) similarly for $d_i^{(b)}$ using $Y_i^{(b)}$, $b = 1, ..., B$.
  - G_0 is the empirical distr func (EDF) of d_i's; G_b is the EDF of $d_i^{(b)}$'s
  - Test stat: $u_k = \int [G_k(y) - \bar{G}(y)]^2 dy$ for $k = 0, 1, ..., B$, and $\bar{G} = \sum_b G_b/B$.
  - $P = \#\{b : u_b > u_0\}/B$
  - Available in BRB ArrayTools: `http://linus.nci.nih.gov./BRB-ArrayTools.html`
- Significance of clusters: Liu et al (JASA, 2012); R package `sigclust`. See also R package `pvclust`.

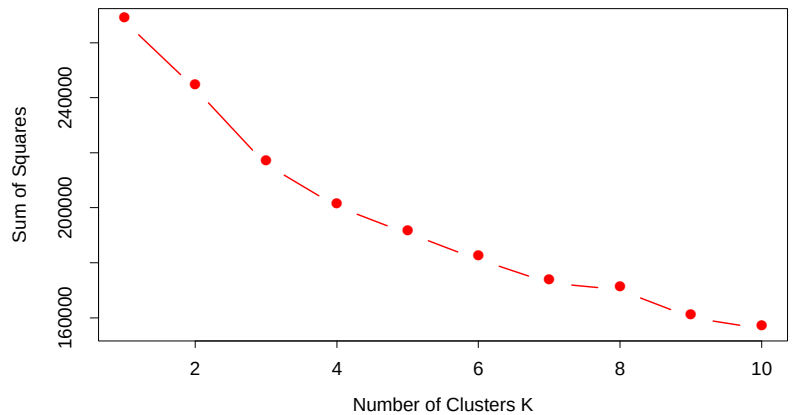


FIGURE 14.8. *Total within-cluster sum of squares for K-means clustering applied to the human tumor microarray data.*

Reproducibility

- Use of the bootstrap
  Ref: Zhang & Zhao (FIG, 2000); Kerr & Churchill (PNAS, 2001); ...
- Reproducibility indices
  - Ref: McShane et al (2002, Bioinformatics, 18:1462-9)
  - Robustness (R) index and Discrepancy (D) index
  - Again, use the parametric bootstrap:
- R package clusterv

- Y_i's: original obs's
- $Y_{ij}^{(b)} = Y_{ij} + \epsilon_{ij}^{(b)}$, where $\epsilon_{ij}^{(b)}$ iid $N(0, v_0)$, and
  $v_0 = median(v_i's)$,
  $v_i = var(Y_{i1}, ..., Y_{iK})$
- Cluster $\{Y_j^{(b)} : j = 1, ..., K\}$ for each $b = 1, ..., B$
- Find the best-matched clusters from $\{Y_j^{(b)}\}$ and $\{Y_j\}$,
- For each paired clusters, $r_k^{(b)}$ =proprotion of pairs of obs's in both clusters (i.e kth clusters)
- R is an average of $r_k^{(b)}$'s
- D is an avarege of proportions of pairs of obs's not in the same cluster
- Note: Finding best-matched clusters may not be easy.

Determine # of clusters: PS

- In general, a tough problem; many many methods
- Ref: Tibshirani & Walther (2005), "Clustering validation by prediction strength". *JCGS*, 14, 511-528.
  many ref's therein;
  R: prediction.strength() in package fpc
- Clustering and classification
- Main idea: suppose we have a training dataset and a test dataset; comparing the agreement b/w the two clustering results; $k = k_0$ will give the best agreement
  1) Cluster the test data into k clusters;
  2) Cluster the training data into k clusters;
  3) Measure how well the training set cluster centers predict c-membership in the test set.
  - Fig 1

- Define "prediction strength":

$$ps(k) = \min_{1 \le j \le k} \frac{1}{n_{kj}(n_{kj}-1)} \sum_{i \neq i' \in A_{kj}} I(D[C(X_{tr}, k), X_{te}]_{ii'} = 1)$$

  where A_{kj}: test obseravtions in test cluster j, and $n_{kj} = |A_{kj}|$; $D[C(.,.), X]$ is a matrix with ii'th element $D[C(.,.), X]_{ii'} = 1$ if obs's i and i' fall into the same cluster in C, and $= 0$ o/w.

- Choice of k: largest k such that $ps(k) > ps_0$.
  ps_0: 0.8-0.9
  $ps(1) = 1$
- Fig 2 therein
- In practice, use repeated 2-fold (or 5-fold) cross-validation.
- See also Wang (2010, Biometrika, 97, 893-904) by CV;
  Fang & Wang (2012, CSDA, 56, 468-477): nselectboot() in R package fpc.

Other criteria

- R: package fpc
- Let $B(k)$ and $W(k)$ be the between- and within-cluster sum of squares
- Calinski & Harabasz (1974):

$$\hat{k} = argmax_k \frac{B(k)/(k-1)}{W(k)/(n-k)}$$

  note: $CH(1)$ not defined.
- Hartigan (1975):

$$H(k) = \frac{W(k)/W(k+1) - 1}{n-k-1}$$

  $\hat{k}$: smallest $k \geq 1$ such that $H(k) \leq 10$.

- Krzanowski & Lai (1985):

$$\hat{k} = argmax_k \left| \frac{DIFF(k)}{DIFF(k+1)} \right|$$

  where $DIFF(k) = (k-1)^{2/p} W_{k-1} - k^{2/p} W_k$, p is the dim of an obs.
- Gap stat (Tibshirani et al, JRSS-B, 2001)
  R: clusGap() in package cluster.
- Use of bagging: Dudoit & Fridlyand (Genome Biology, 2002)
  more ref's

Gap stat

- Motivation: as k increases, W_k ...?
- $Gap(k) = E^*[\log(W_k)] - \log(W_k)$, where E^* is expectation under a reference distribution (e.g. uniform).
- Algorithm:

Step 1. Cluster the observed data and obtain W_k, $k = 1, ..., k_{max}$.

Step 2. Generate B reference data sets (e.g. using the uniform distr), and obtain $W_k^{(b)}$, $b = 1, ..., B$ and $k = 1, ..., k_{max}$.
Compute the gap stat: $Gap(k) = \log(\bar{W})_k - \log(W_k)$. where $\log(\bar{W})_k = \sum_b \log(W_k^{(b)})/B$.

Step 3. Compute SD: $sd_k = \sum_b [\log(W_k^{(b)}) - \log(\bar{W})_k]^2/B$. and define $s_k = sd_k\sqrt{1 + 1/B}$.

Step 4. Choose a smallest k such that

$$Gap(k) \geq Gap(k+1) - s_{k+1}$$

- Fig 14.11

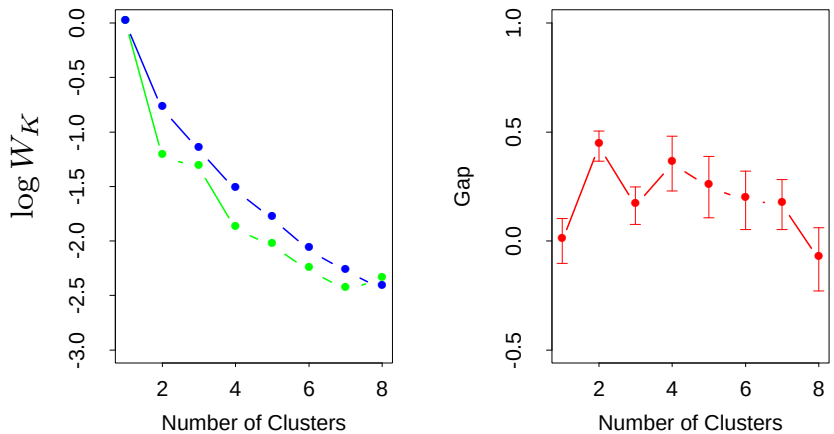


FIGURE 14.11. *(Left panel): observed (green) and expected (blue) values of* $\log W_K$ *for the simulated data of Figure 14.4. Both curves have been translated to equal zero at one cluster. (Right panel): Gap curve, equal to the difference between the observed and expected values of* $\log W_K$*. The Gap estimate* K^* *is the smallest* K *producing a gap within one standard deviation of the gap at* $K+1$*; here* $K^* = 2$*.*

Assessing clustering results

- Define a_i = average dissimilarity between obs i and all other obs's of the cluster to which obs i belong;
- For all other clusters A, $d(i, A)$ = average dissimilarity of obs i to all obs's of cluster A;
- $b_i = min_A d(i, A)$
- Silhouette width: $s_i = \frac{b_i - a_i}{max(a_i, b_i)}$
- a large $s_i \Longrightarrow$ obs i is well clustered; a small s_i (close to 0) $\Longrightarrow$ obs i lies between two clusters; a negative $s_i \Longrightarrow$ obs i is probably in a wrong cluster.

Measuring clustering agreement

- Q: how to measure the agreement between two clustering results, C_1 vs C_2?
  note: #s of clusters in the two may be different!
- Rand (1971, JASA) index: for n obs's,
  $a = \#$ of obs pairs in the same cluster in both C_1 and C_2;
  $b = \#$ of obs pairs in different clusters in both C_1 and in C_2;
  $R = (a + b)/C(n, 2)$.
- Adjusted RI: removing the agreement due to random chance. HA (Hubert and Arabie, 1985, J Classification), MA (Morey and Agresti's)
- Other ones: Fowlkes and Mallows (1983, JASA) index; Jaccard index, ...
- For more, see Wagner & Wagner (2007). "Comparing clusterings–An Overview".
  `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.164.6189&rep=rep1&type=pdf`
- R package clues.

Big Data

- Kurasova et al (2014) "Strategies for Big Data Clustering". http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6984551
- Littau D, Borey D (2009). Clustering Very Large Datasets using a Low Memory Matrix Factored Representation. Computational Intelligence, 25: 114-135. http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8640.2009.00331.x/full
- Main idea: data $X_{p\times n}$, cluster centers $C_{p\times k}$; $p, n >> k$.
  $X \approx CZ$;
  $Z_{k\times n}$ estimated by LS.
  Save space: $n \times p >> p \times k + k \times n$.