Clustering

Aaditya Ramdas

Carnegie Mellon University

Thanks, class notes
by Larry Wasserman

Outline

1. *Density-based clustering (eg: mean shift)*

2. *Hierarchical clustering (eg: single linkage)*

3. *K-means and K-medoids*

4. *Spectral clustering*

Modal clustering

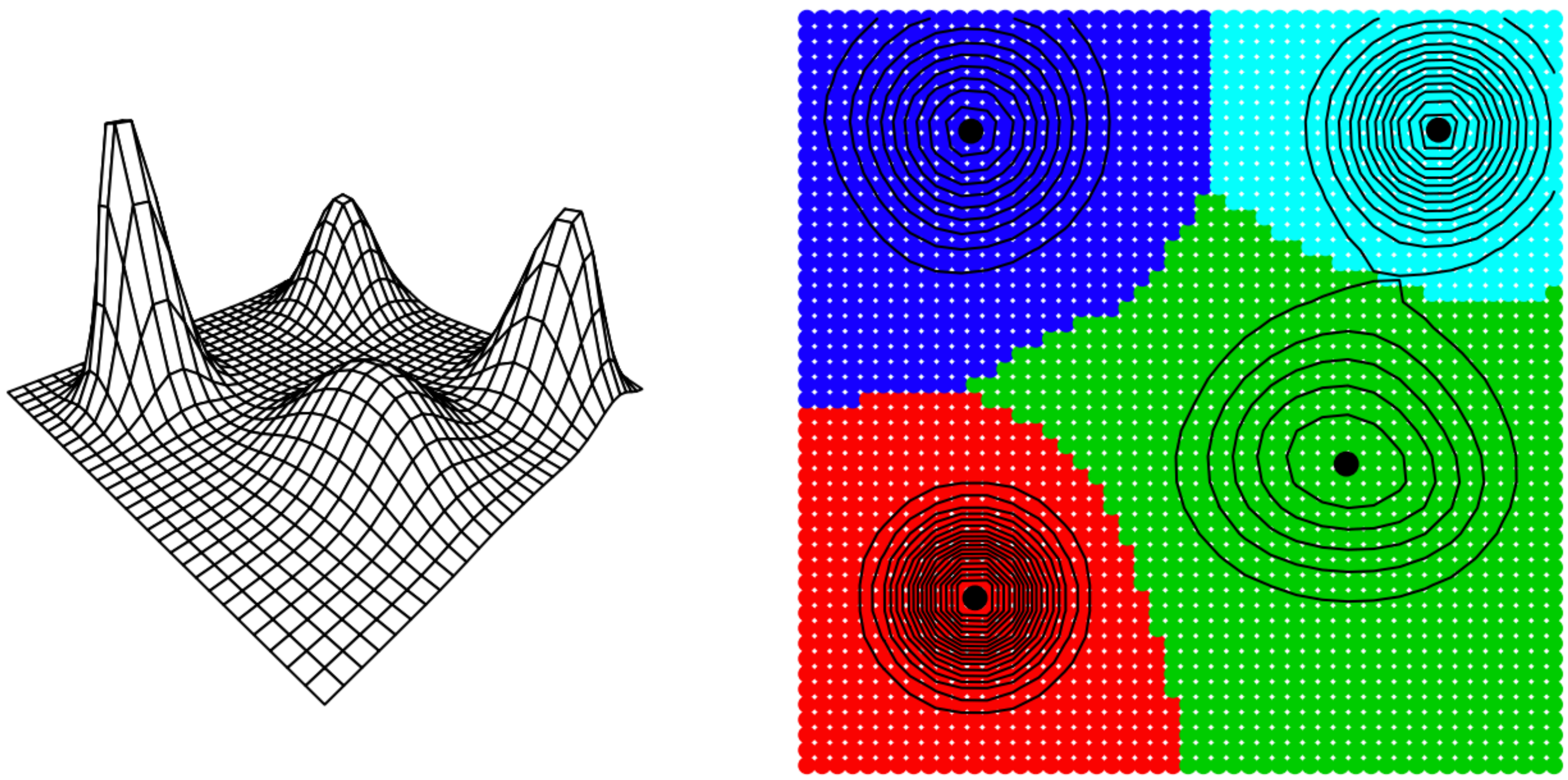

Figure 1: *The left plot shows a function with four modes. The right plot shows the ascending manifolds (basins of attraction) corresponding to the four modes.*

Intuitively, gradient (wrt density) ascent from each point

Mean Shift Algorithm

1. Input: $\widehat{p}(x)$ and a mesh of points $A = \{a_1, \ldots, a_N\}$ (often taken to be the data points).
2. For each mesh point a_j, set $a_j^{(0)} = a_j$ and iterate the following equation until convergence:

$$a_j^{(s+1)} \longleftarrow \frac{\sum_{i=1}^{n} X_i K\left(\frac{||a_j^{(s)} - X_i||}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{||a_j^{(s)} - X_i||}{h}\right)}.$$

3. Let $\widehat{\mathcal{M}}$ be the unique values of the set $\{a_1^{(\infty)}, \ldots, a_N^{(\infty)}\}$.
4. Output: $\widehat{\mathcal{M}}$.

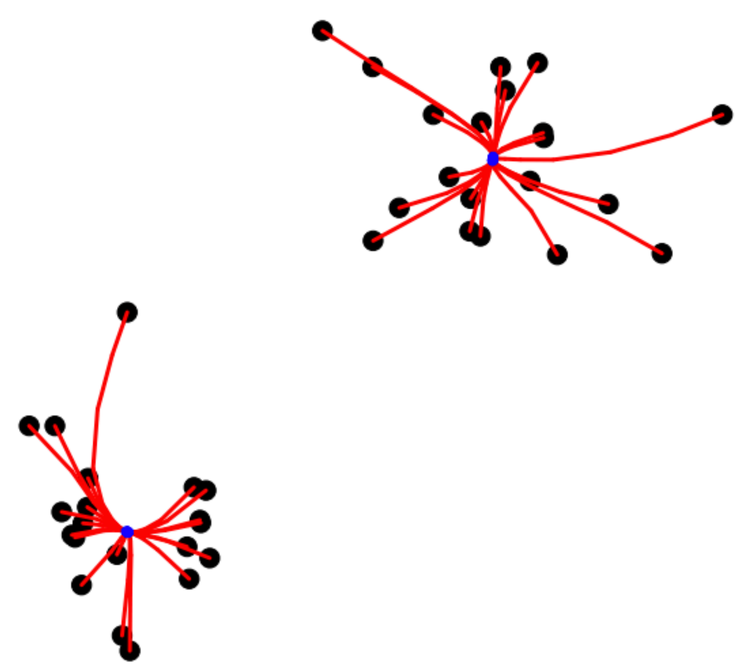

Figure 3: A simple example of the mean shift algorithm.

Blurred mean-shift: Replace data by mean-shifted data, stop early

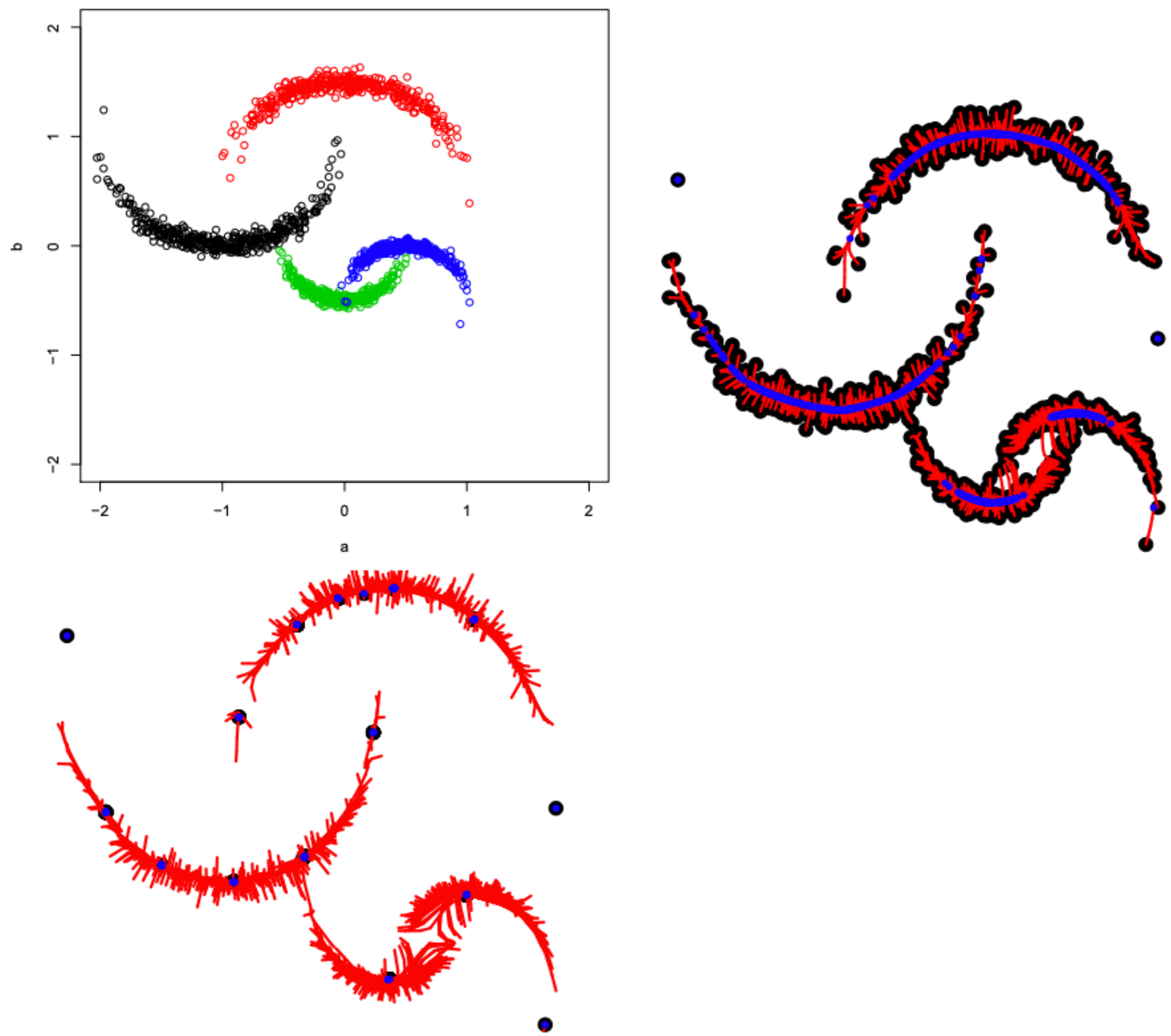

Figure 4: The crescent data example. Top left: data. Top right: a few steps of mean-shift. Bottom left: a few steps of blurred mean-shift.

Blurred mean-shift: Replace data by mean-shifted data, stop early

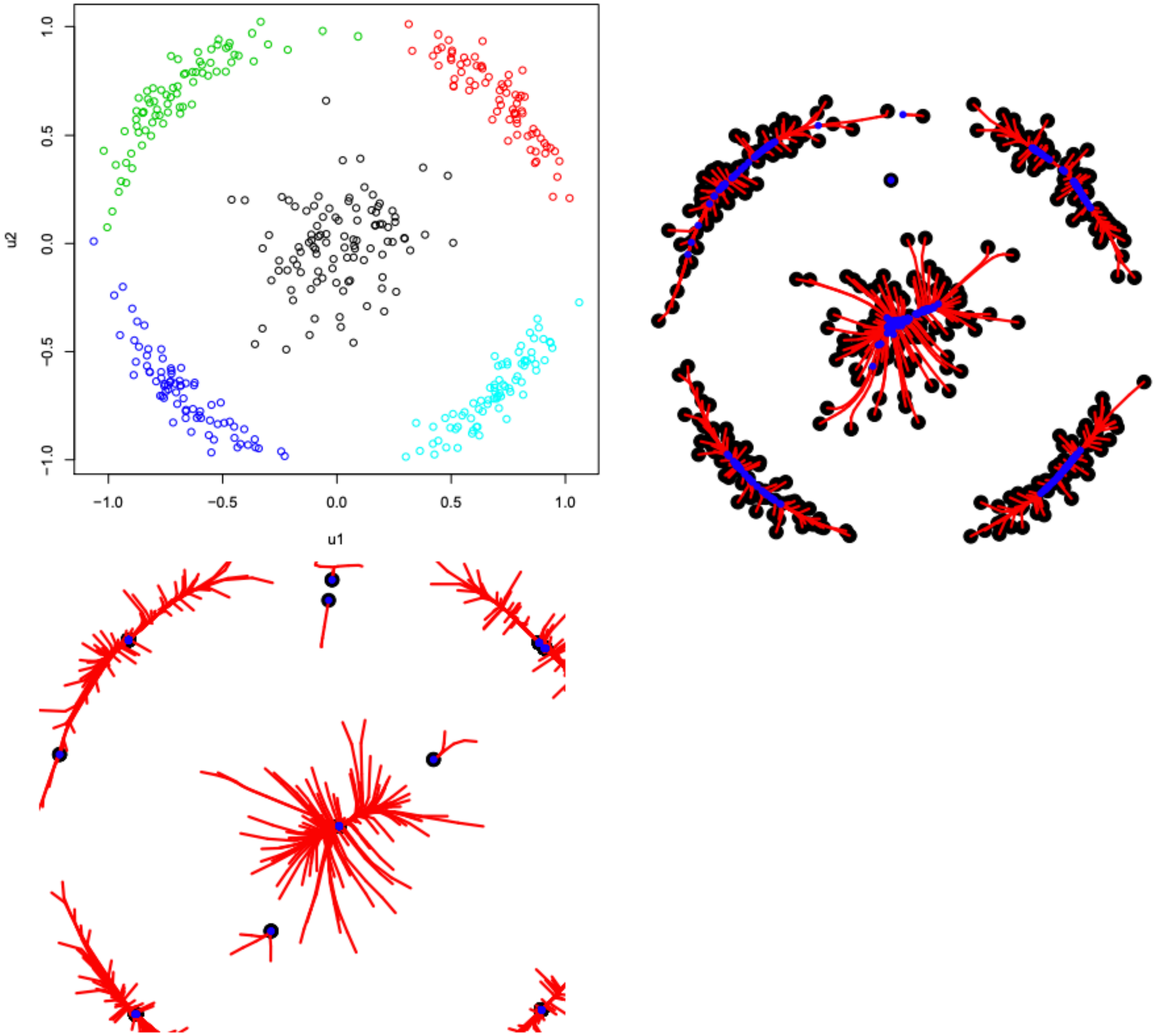

Figure 5: The Broken Ring example. Top left: data. Top right: a few steps of mean-shift. Bottom left: a few steps of blurred mean-shift.

Level-set clustering

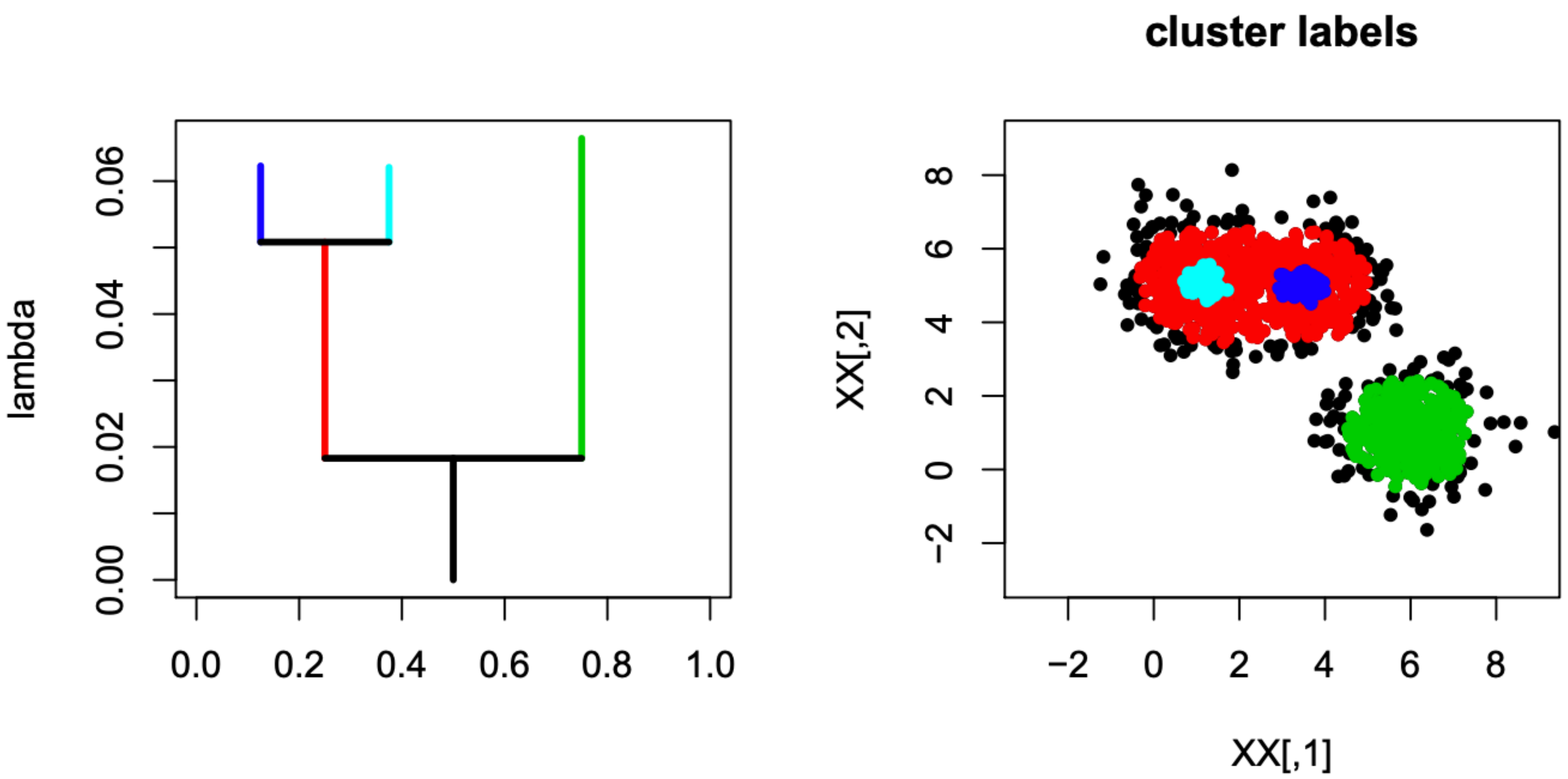

Figure 6: DeBaClR in two dimensions.

Intuitively, scan the density in a top-down fashion (from modes) and note when the clusters merge

K-means

Let $X_1,\dots,X_n \sim P$ where $X_i \in \mathbb{R}^d$. Let $C = \{c_1,\dots,c_k\}$ where each $c_j \in \mathbb{R}^d$. We call C a codebook. Let $\Pi_C[X]$ be the projection of X onto C:

$$\Pi_C[X] = \text{argmin}_{c\in C}||c - X||^2. \tag{1}$$

Define the empirical clustering risk of a codebook C by

$$R_n(C) = \frac{1}{n}\sum_{i=1}^{n}\big|\big|X_i - \Pi_C[X_i]\big|\big|^2 = \frac{1}{n}\sum_{i=1}^{n}\min_{1\le j\le k}||X_i - c_j||^2. \tag{2}$$

Let $\mathscr{C}_k$ denote all codebooks of length k. The optimal codebook $\widehat{C} = \{\widehat{c}_1,\dots,\widehat{c}_k\} \in \mathscr{C}_k$ minimizes $R_n(C)$:

$$\widehat{C} = \text{argmin}_{C\in\mathscr{C}_k} R_n(C). \tag{3}$$

1. Choose k centers $c_1,\dots,c_k$ as starting values.
2. Form the clusters $C_1,\dots,C_k$ as follows. Let $g = (g_1,\dots,g_n)$ where $g_i = \text{argmin}_j||X_i - c_j||$. Then $C_j = \{X_i :\ g_i = j\}$.
3. For $j = 1,\dots,k$, let n_j denote the number of points in C_j and set
$$c_j \longleftarrow \frac{1}{n_j}\sum_{i:\ X_i\in C_j} X_i.$$
4. Repeat steps 2 and 3 until convergence.
5. Output: centers $\widehat{C} = \{c_1,\dots,c_k\}$ and clusters $C_1,\dots,C_k$.

Figure 6: The k-means (Lloyd's) clustering algorithm.

One step of k-means: Voronoi tessellation

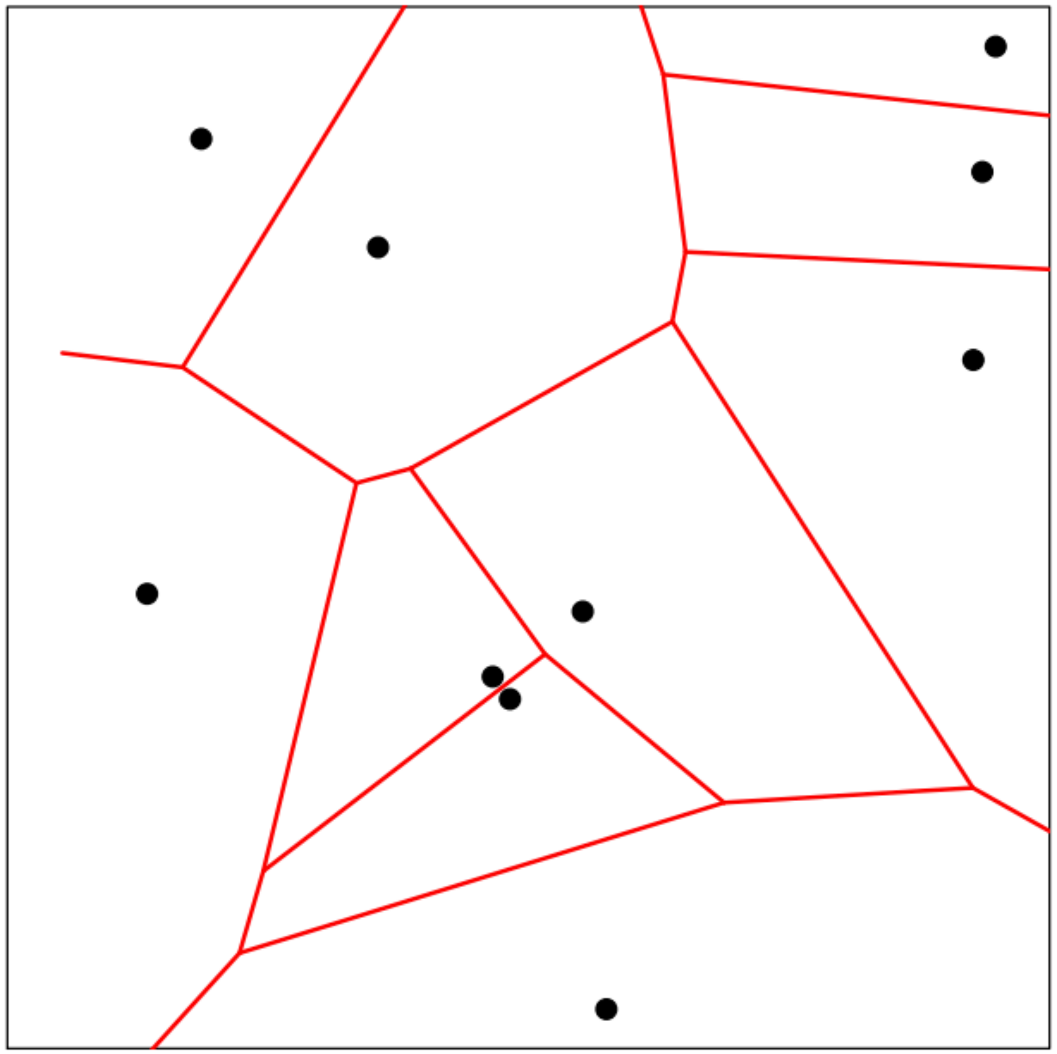

Figure 5: The Voronoi tesselation formed by 10 cluster centers $c_1, \ldots, c_{10}$. The cluster centers are indicated by dots. The corresponding Voronoi cells $T_1, \ldots, T_{10}$ are defined as follows: a point x is in T_j if x is closer to c_j than c_i for $i \neq j$.

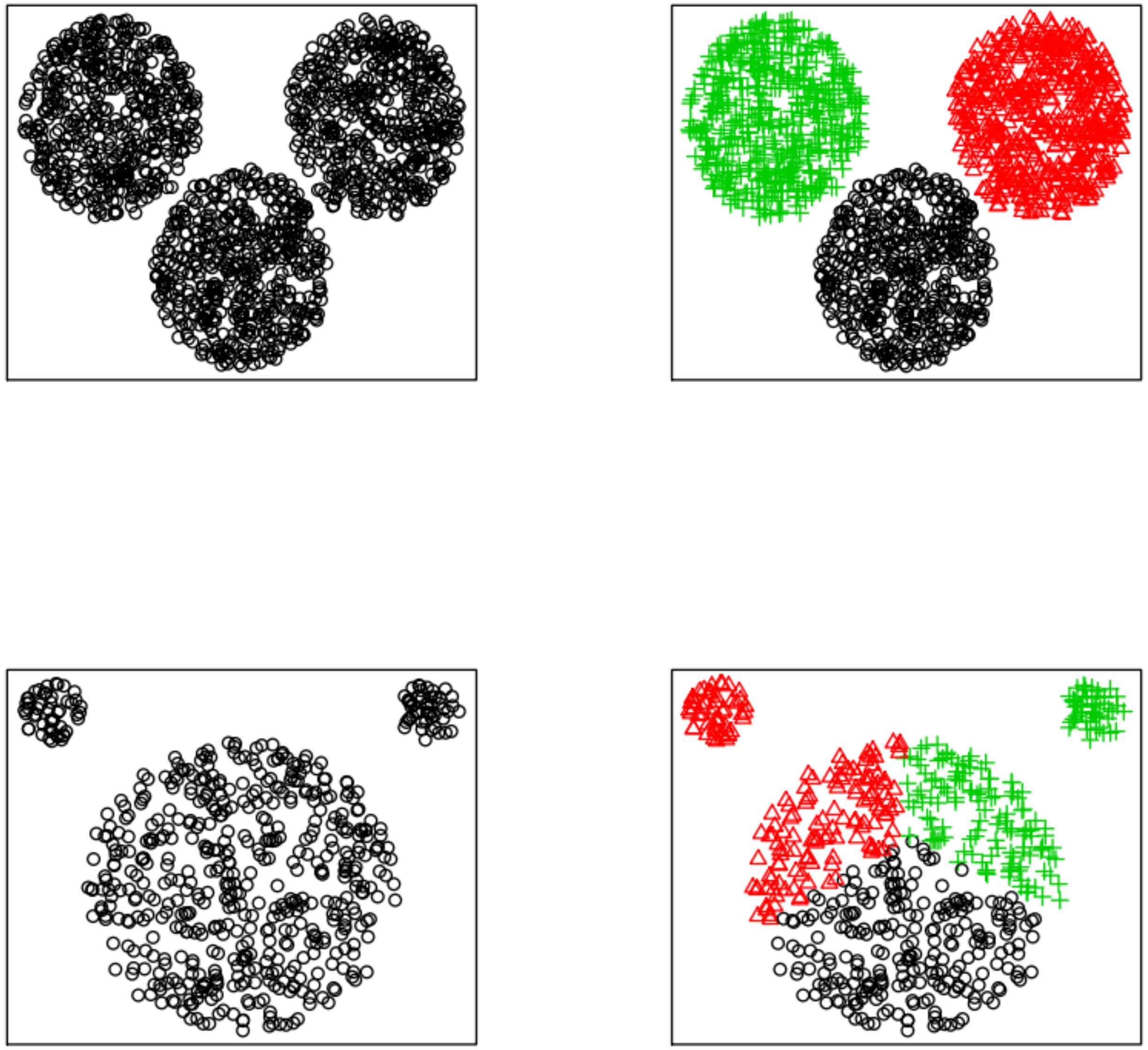

Figure 7: Synthetic data inspired by the “Mickey Mouse” example from wikipedia. Top left: three balanced clusters. Top right: result from running k means with $k = 3$. Bottom left: three unbalanced clusters. Bottom right: result from running k means with $k = 3$ on the unbalanced clusters. k-means does not work well here because the clusters are very unbalanced.

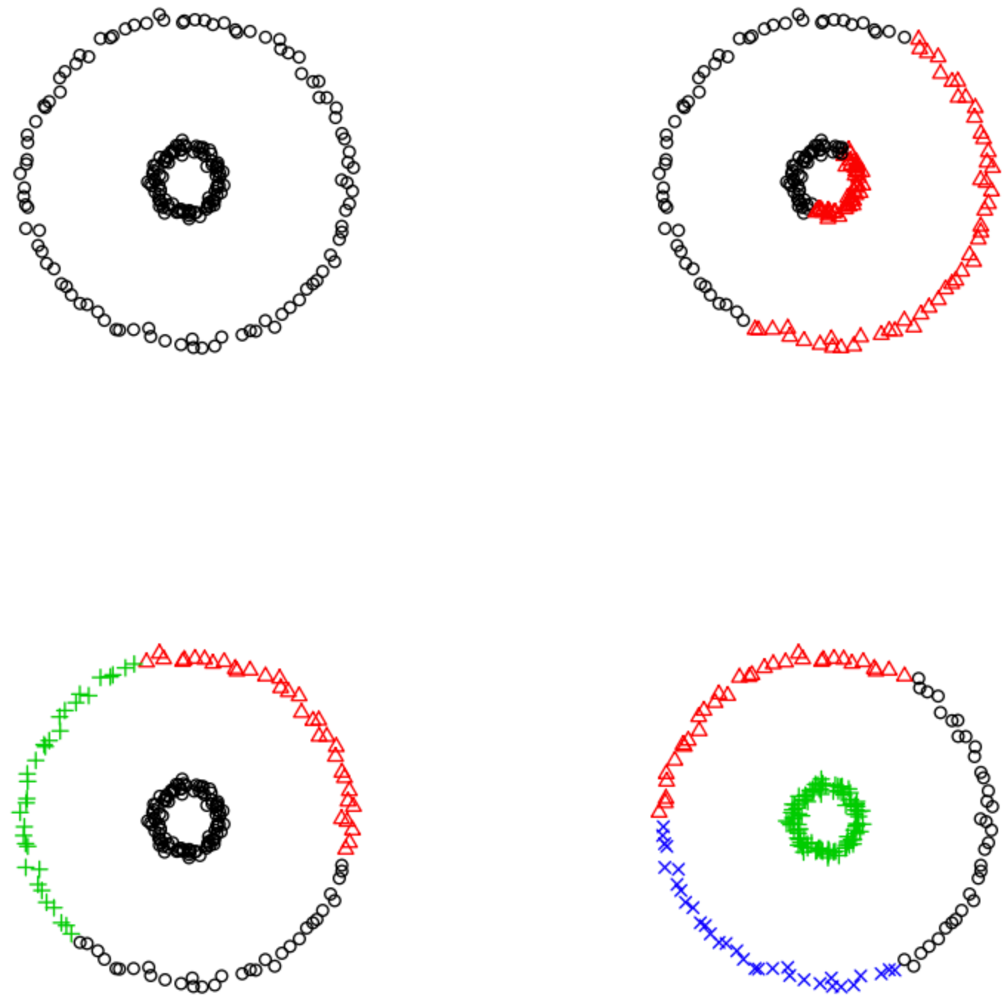

Figure 10: Top left: a dataset with two ring-shaped clusters. Top right: k-means with $k = 2$. Bottom left: k-means with $k = 3$. Bottom right: k-means with $k = 4$.

K-means++

1. Input: Data $X = \{X_1, \ldots, X_n\}$ and an integer k.
2. Choose c_1 randomly from $X = \{X_1, \ldots, X_n\}$. Let $C = \{c_1\}$.
3. For $j = 2, \ldots, k$:
   (a) Compute $D(X_i) = \min_{c \in C} ||X_i - c||$ for each X_i.
   (b) Choose a point X_i from X with probability
   $$p_i = \frac{D^2(X_i)}{\sum_{j=1}^n D^2(X_j)}.$$
   (c) Call this randomly chosen point c_j. Update $C \longleftarrow C \cup \{c_j\}$.
4. Run Lloyd's algorithm using the **seed points** $C = \{c_1, \ldots, c_k\}$ as starting points and output the result.

Figure 12: The k-means^{++} algorithm.

Theorem 10 *(Arthur and Vassilvitskii, 2007). Let $C = \{c_1, \ldots, c_k\}$ be the seed points from the k-means^{++} algorithm. Then,*

$$\mathbb{E}\big(R_n(C)\big) \leq 8(\log k + 2) \left(\min_C R_n(C)\right) \tag{6}$$

where the expectation is over the randomness of the algorithm.

A theoretical property of the k-means method is given in the following result. Recall that $C^* = \{c_1^*, \ldots, c_k^*\}$ minimizes $R(C) = \mathbb{E}||X - \Pi_C[X]||^2$.

Theorem 12 *Suppose that* $\mathbb{P}(||X_i||^2 \leq B) = 1$ *for some* $B < \infty$. *Then*

$$\mathbb{E}(R(\widehat{C})) - R(C^*) \leq c\sqrt{\frac{k(d+1)\log n}{n}} \tag{11}$$

for some $c > 0$.

Warning! The fact that $R(\widehat{C})$ is close to $R(C_*)$ does not imply that $\widehat{C}$ is close to C_*.

Geometric graph clustering

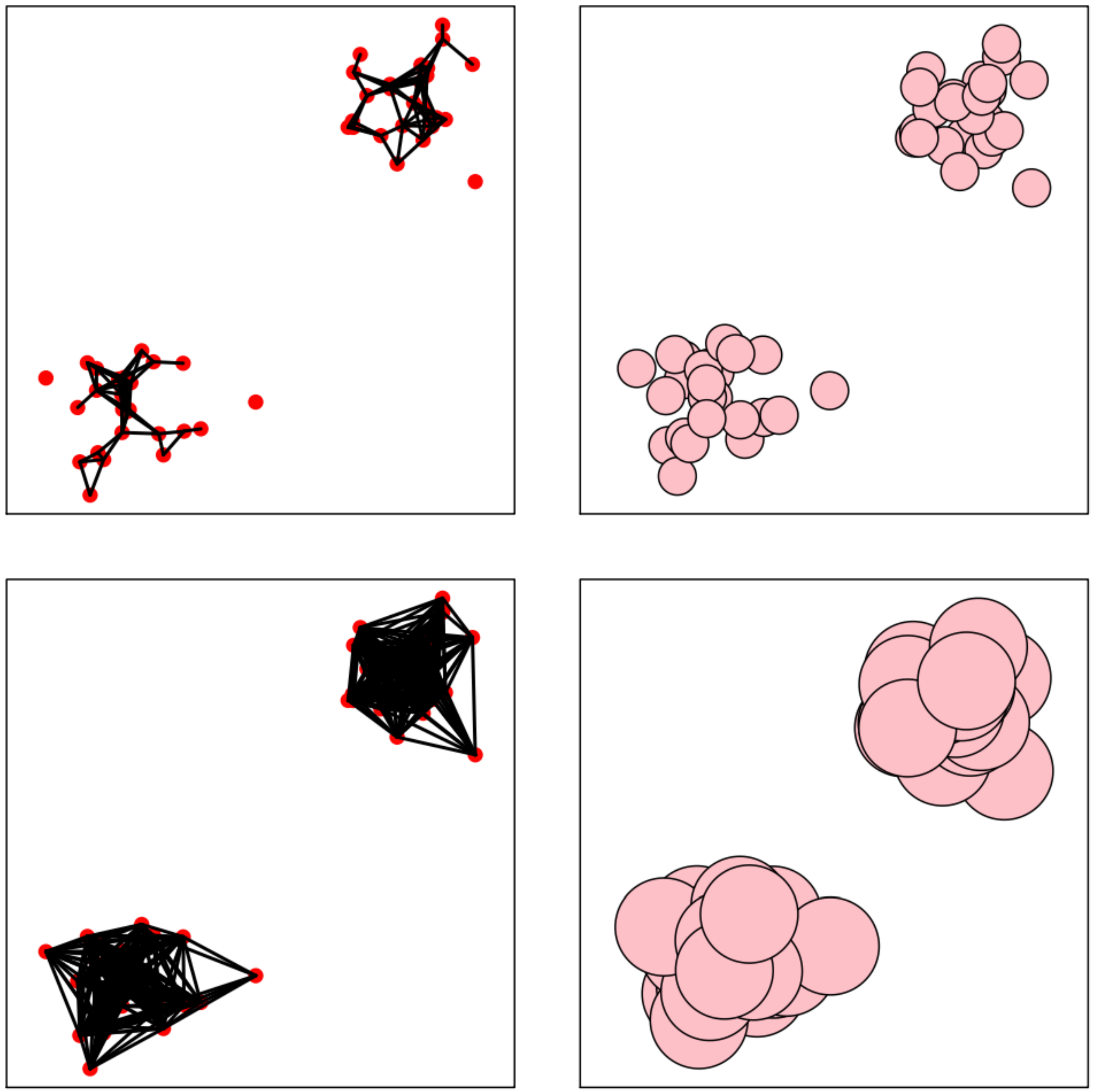

Figure 15: Geometric graph clustering. Top row: $\epsilon = 1$. Bottom row: $\epsilon = 3$. Left column: the graphs. Right column: a ball of radius $\epsilon/2$ around each X_i.

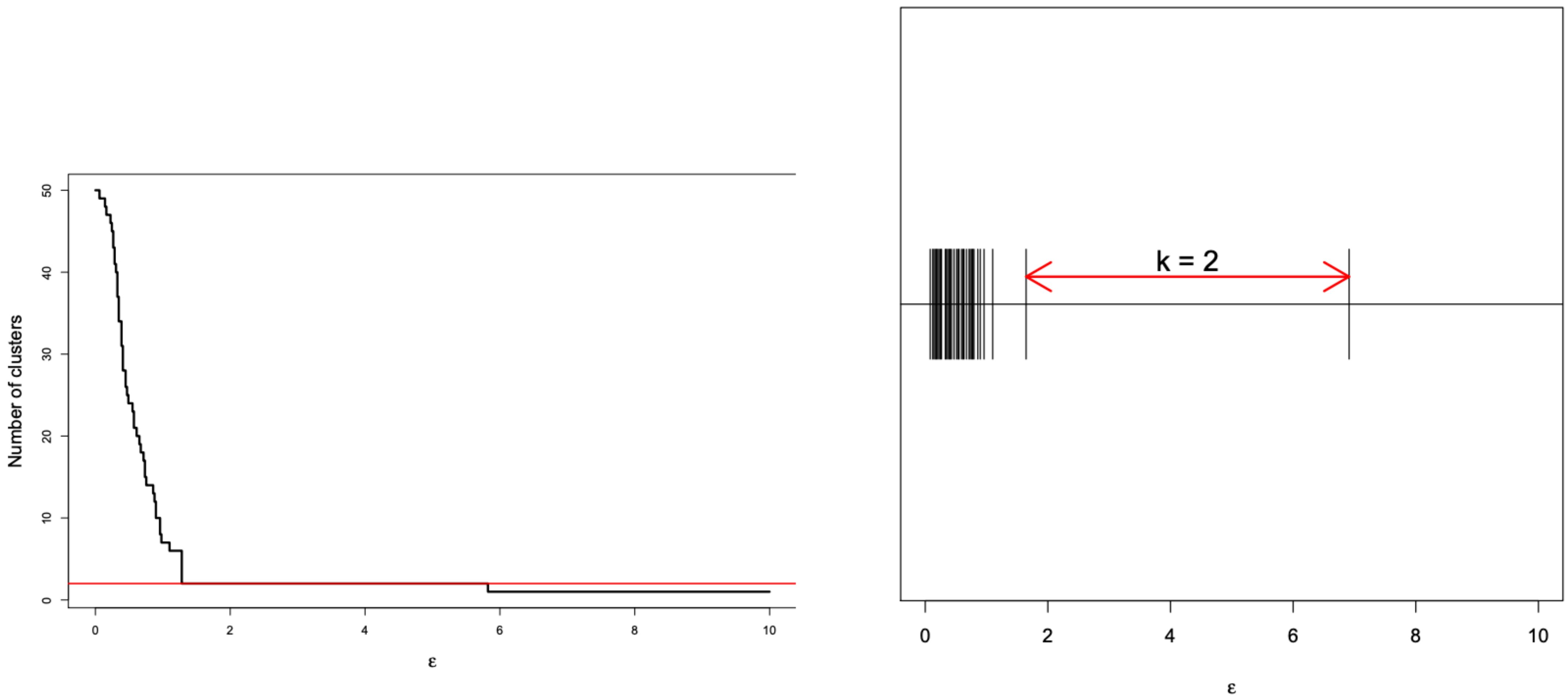

Figure 16: Geometric graph clustering. The top plot shows the number of connected components $k(\epsilon)$ versus ϵ. There is a fairly large range of values of ϵ that yields two connected components. The bottom plot shows the values of ϵ at which $k(\epsilon)$ changes.

Hierarchical clustering

1. Input: data $X = \{X_1, \ldots, X_n\}$ and metric d giving distance between clusters.
2. Let $T_n = \{C_1, C_2, \ldots, C_n\}$ where $C_i = \{X_i\}$.
3. For $j = n-1$ to 1:
   (a) Find j, k to minimize $d(C_j, C_k)$ over all $C_j, C_k \in T_{j+1}$.
   (b) Let T_j be the same as T_{j+1} except that C_j and C_k are replaced with $C_j \cup C_k$.
4. Return the sets of clusters $T_1, \ldots, T_n$.

Figure 20: Agglomerative Hierarchical Clustering

Single Linkage	$d(A,B) = \min_{x \in A, y \in B} d(x,y)$
Average Linkage	$d(A,B) = \frac{1}{N_A N_B} \sum_{x \in A, y \in B} d(x,y)$
Complete Linkage	$d(A,B) = \max_{x \in A, y \in B} d(x,y)$

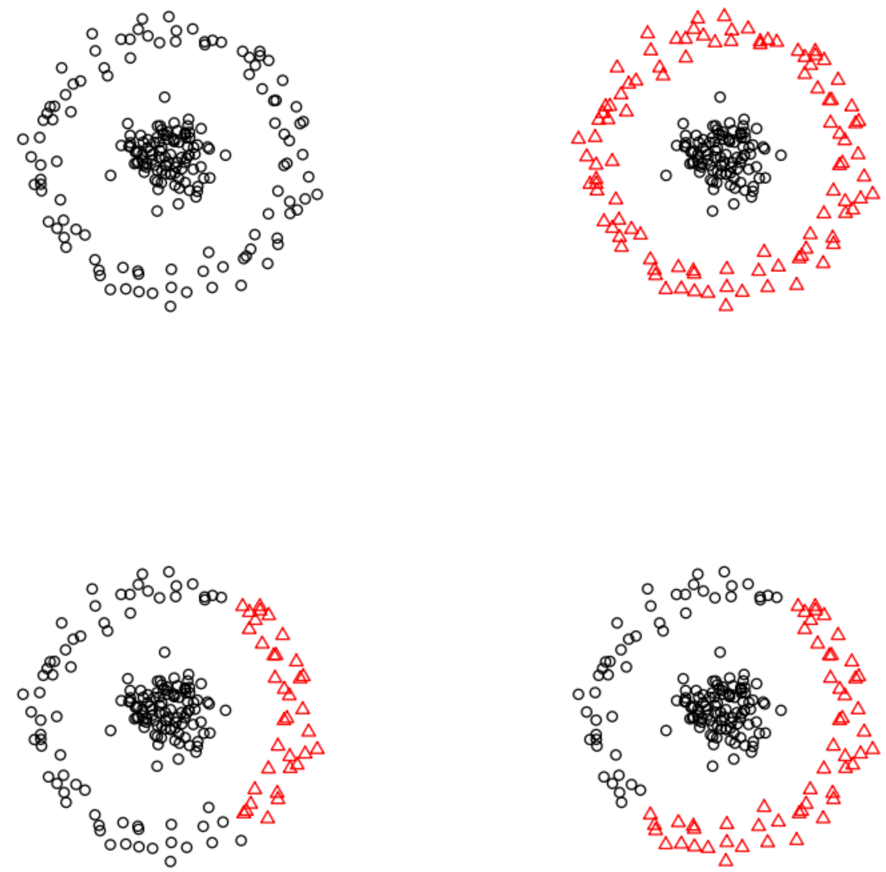

Figure 21: Hierarchical clustering applied to two noisy rings. Top left: the data. Top right: two clusters from hierarchical clustering using single linkage. Bottom left: average linkage. Bottom right: complete linkage.

Single Linkage	$d(A,B) = \min_{x \in A, y \in B} d(x,y)$
Average Linkage	$d(A,B) = \frac{1}{N_A N_B} \sum_{x \in A, y \in B} d(x,y)$
Complete Linkage	$d(A,B) = \max_{x \in A, y \in B} d(x,y)$

Spectral clustering

For a graph G on n nodes, let W be its adjacency matrix, and D be its degree matrix.

The graph Laplacian L = D - W has the following properties:

1. *For any vector* $f = (f_1, \ldots, f_n)^T$,
$$f^T L f = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij}(f_i - f_j)^2.$$
2. *L is symmetric and positive semi-definite.*
3. *The smallest eigenvalue of L is 0. The corresponding eigenvector is* $(1,1,\ldots,1)^T$.
4. *L has n non-negative, real-valued eigenvalues* $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k$.
5. *The number of eigenvalues that are equal to 0 is equal to the number of connected components of G. That is,* $0 = \lambda_1 = \ldots = \lambda_k$ *where k is the number of connected components of G. The corresponding eigenvectors $v_1, \ldots, v_k$ are orthogonal and each is constant over one of the connected components of the graph.*

Example 20 *Consider the graph*

$$\begin{tikzcd} X_1 \arrow[r, dash] & X_2 & X_3 \arrow[r, dash] & X_4 \arrow[r, dash] & X_5 \end{tikzcd}$$

and suppose that $W_{ij} = 1$ if and only if there is an edge between X_i and X_j. Then

$$W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \qquad D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

and the Laplacian is

$$L = D - W = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 0 \end{pmatrix}.$$

The eigenvalues of W, from smallest to largest are $0, 0, 1, 2, 3$. The eigenvectors are

$$v_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad v_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad v_3 = \begin{pmatrix} 0 \\ 0 \\ -.71 \\ 0 \\ .71 \end{pmatrix} \quad v_4 = \begin{pmatrix} -.71 \\ .71 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad v_5 = \begin{pmatrix} 0 \\ 0 \\ -.41 \\ .82 \\ -.41 \end{pmatrix}$$

Note that the first two eigenvectors correspond to the connected components of the graph.

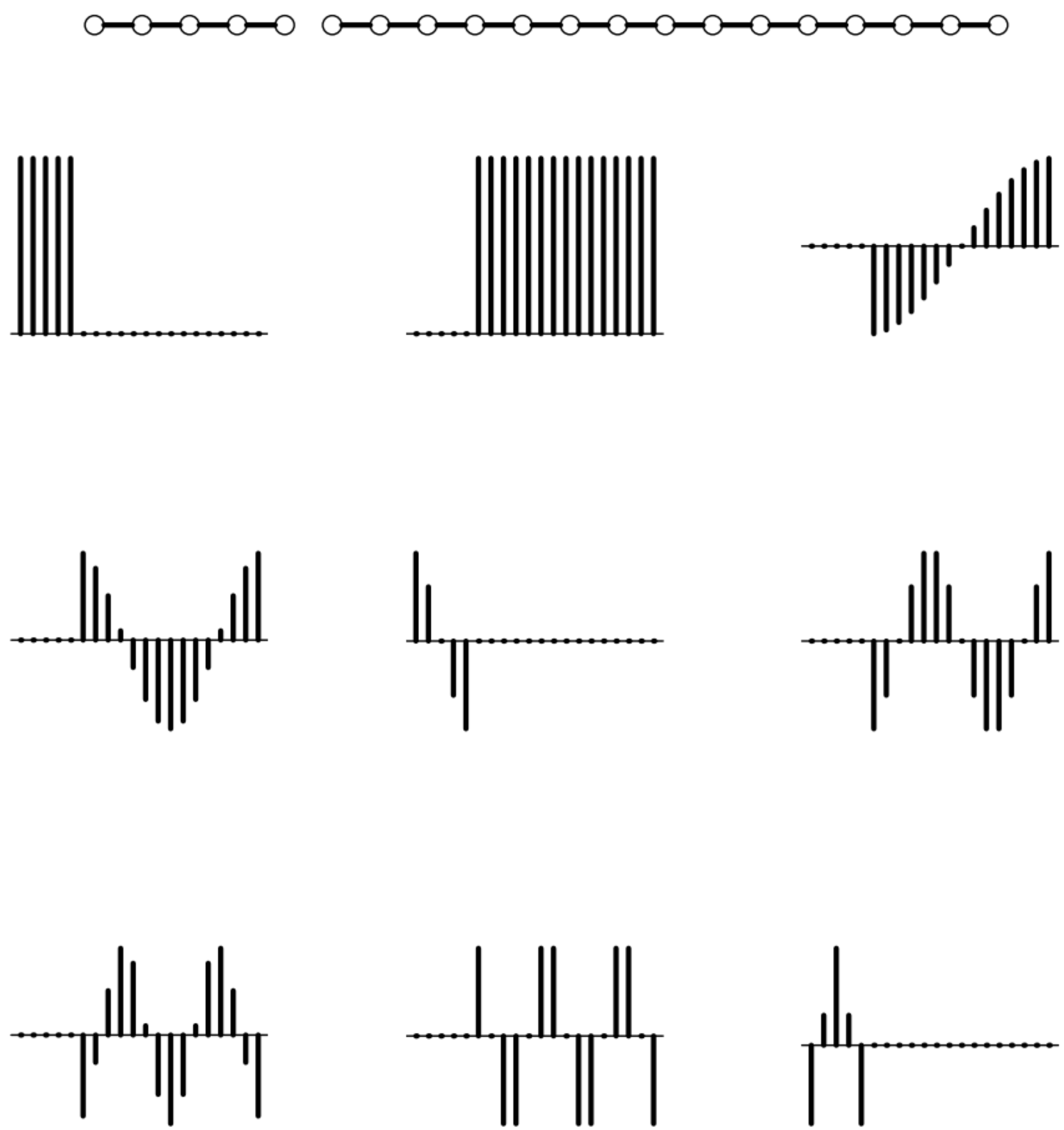

Figure 22: The top shows a simple graph. The remaining plots are the eigenvectors of the graph Laplacian. Note that the first two eigenvectors correspond to the two connected components of the graph.

Input: $n \times n$ similarity matrix W.

1. Let D be the $n \times n$ diagonal matrix with $D_{ii} = \sum_j W_{ij}$.
2. Compute the Laplacian $\mathscr{L} = D^{-1}W$.
3. Find first k eigenvectors $v_1, \ldots, v_k$ of $\mathscr{L}$.
4. Project each X_i onto the eigenvectors to get new points $\widehat{X}_i$.
5. Cluster the points $\widehat{X}_1, \ldots, \widehat{X}_n$ using any standard clustering algorithm.

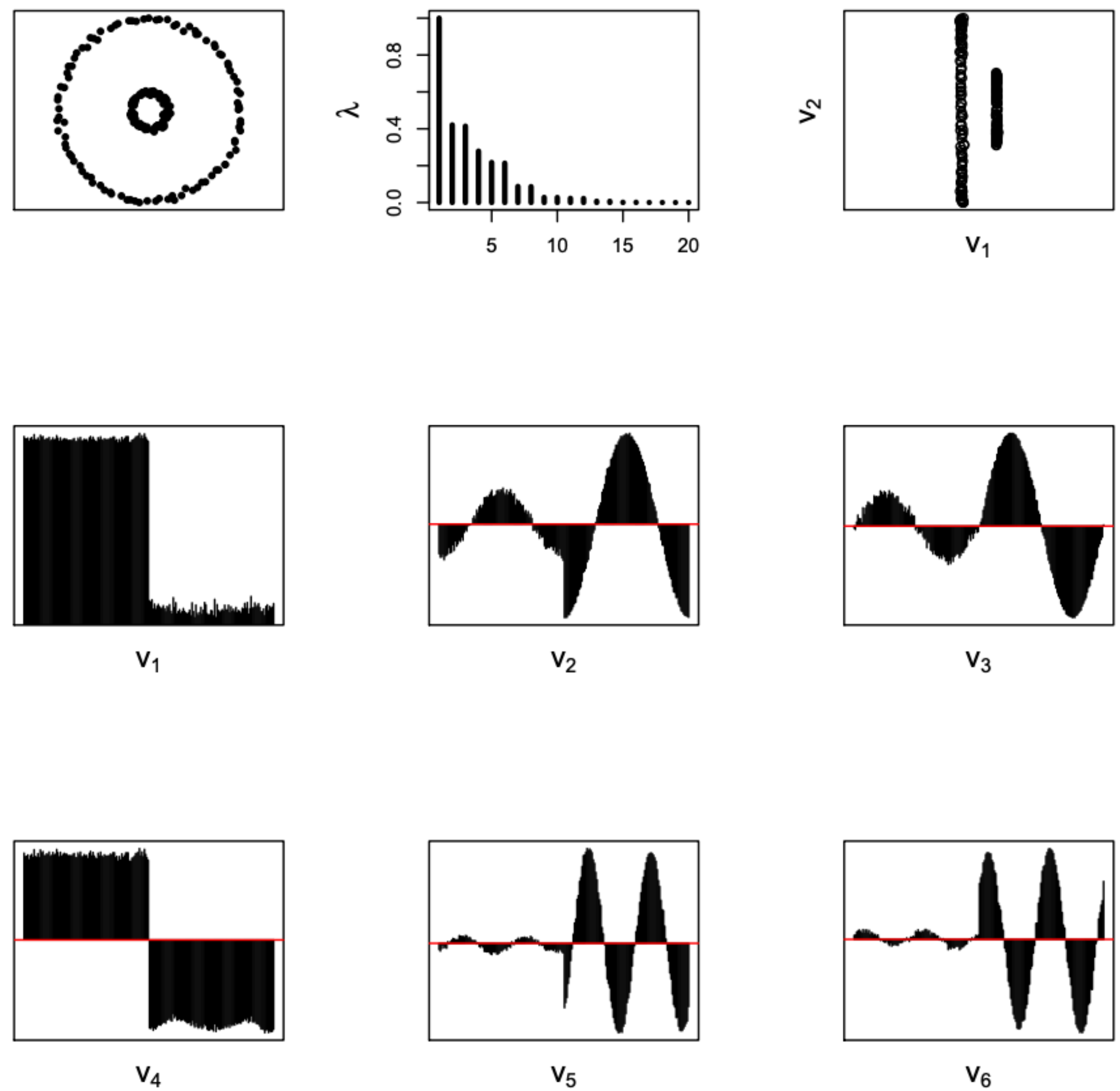

Figure 23: Top left: data. Top middle: eigenvalues. Top right: second versus third eigenvectors. Remaining plots: first six eigenvectors.