

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*

Level-set clustering

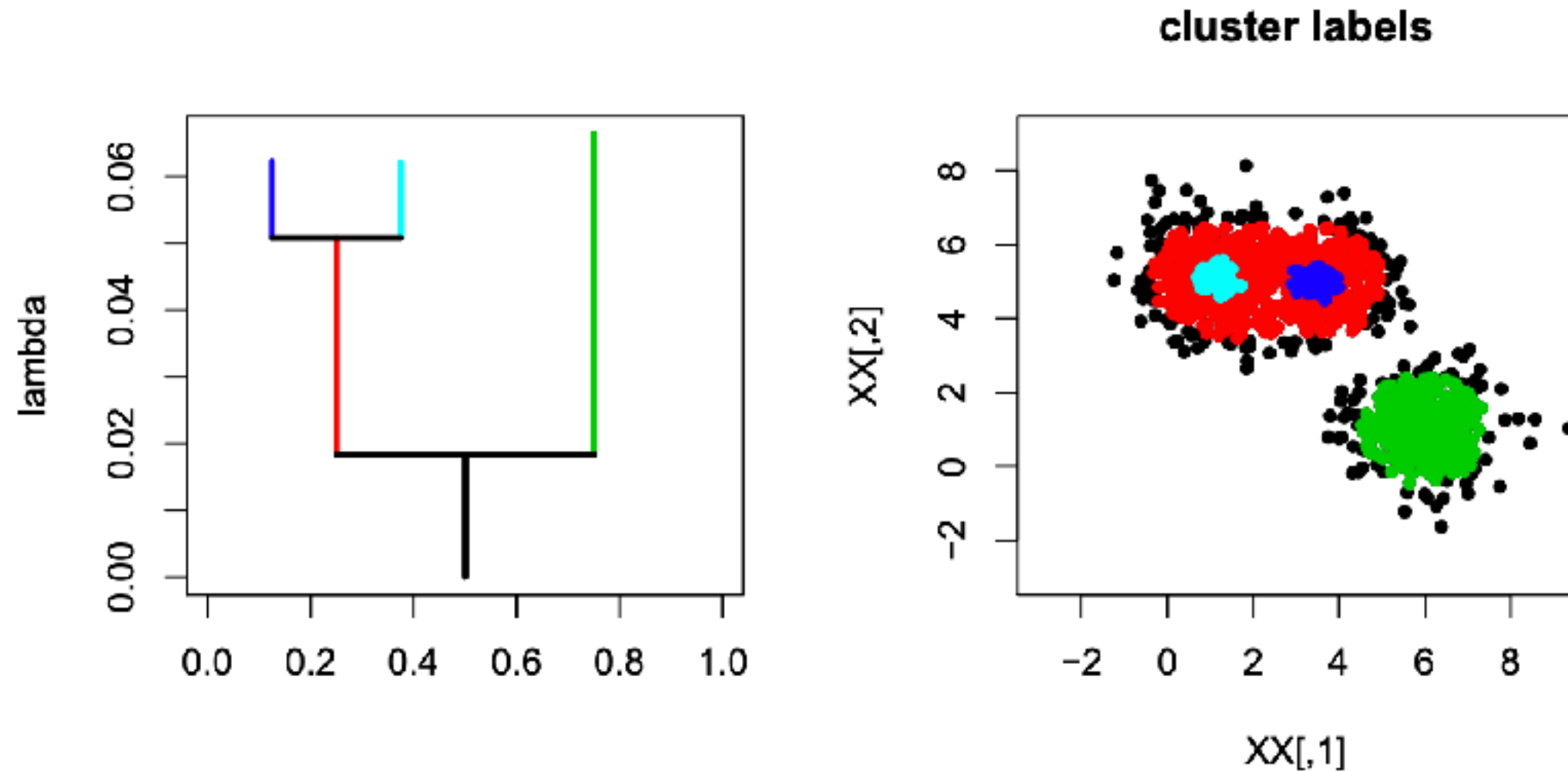


Figure 6: DeBaCLR in two dimensions.

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

Modal clustering

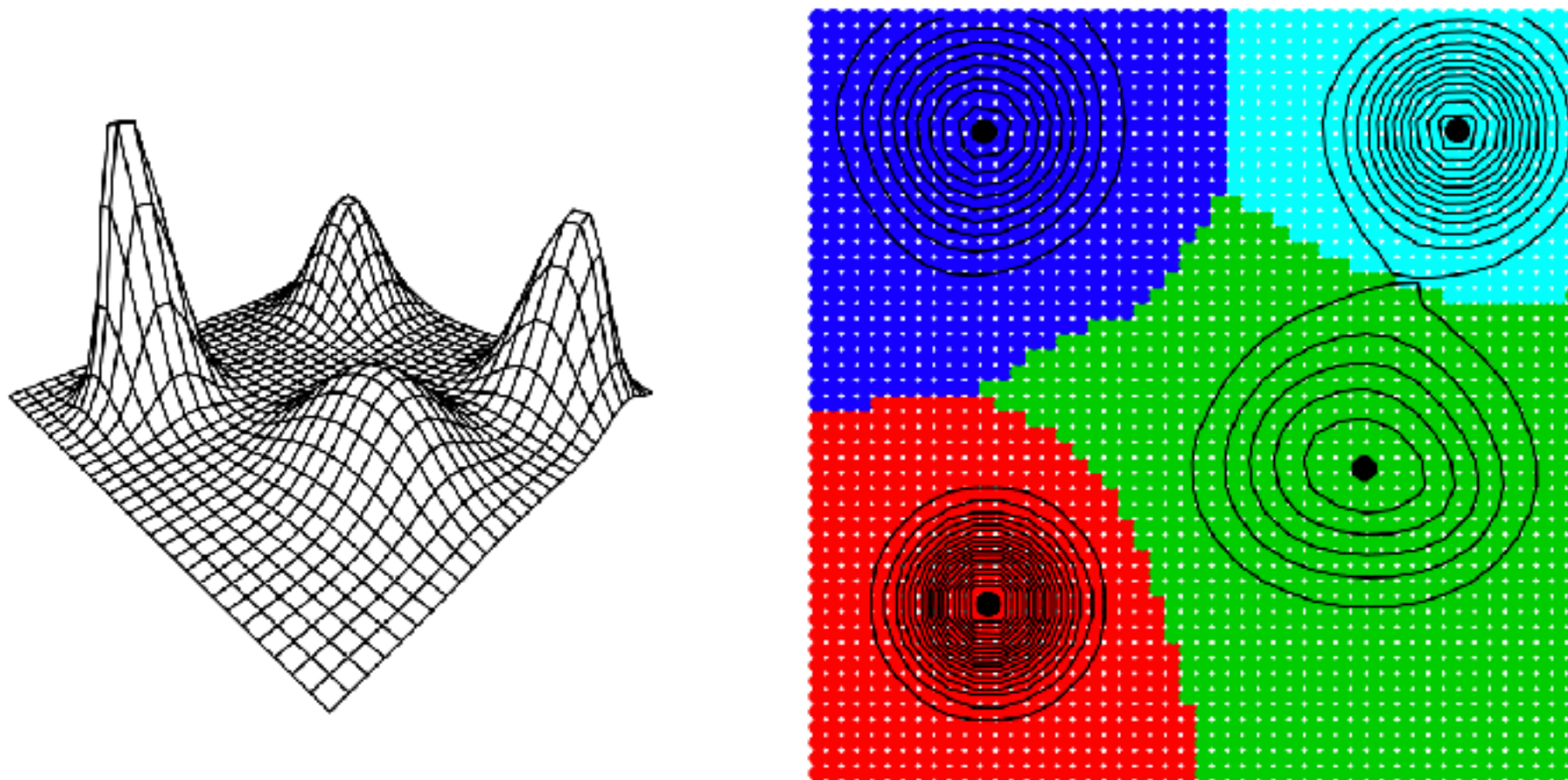


Figure 1: *The left plot shows a function with four modes. The right plot shows the ascending manifolds (basins of attraction) corresponding to the four modes.*

Intuitively, gradient (wrt density) ascent from each point

Mean Shift Algorithm

1. Input: $\hat{p}(x)$ and a mesh of points $A = \{a_1, \dots, a_N\}$ (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:

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

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

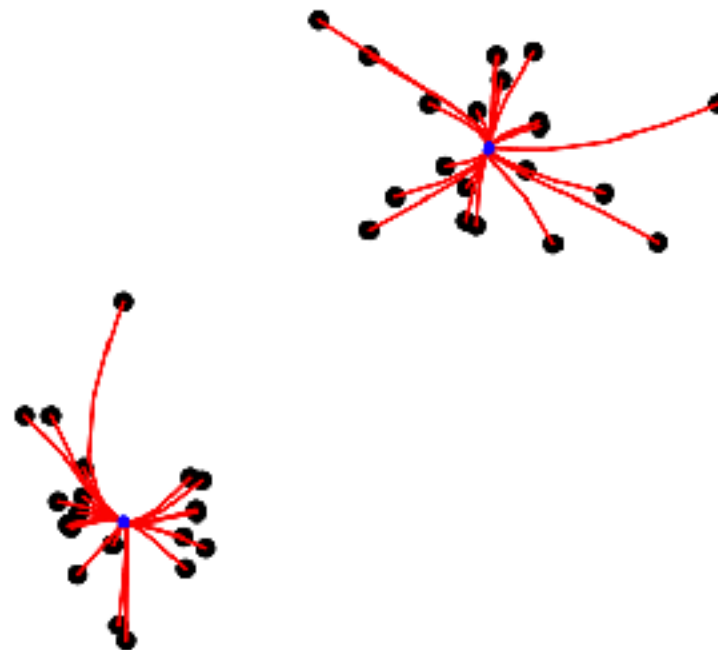


Figure 3: A simple example of the mean shift algorithm.

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

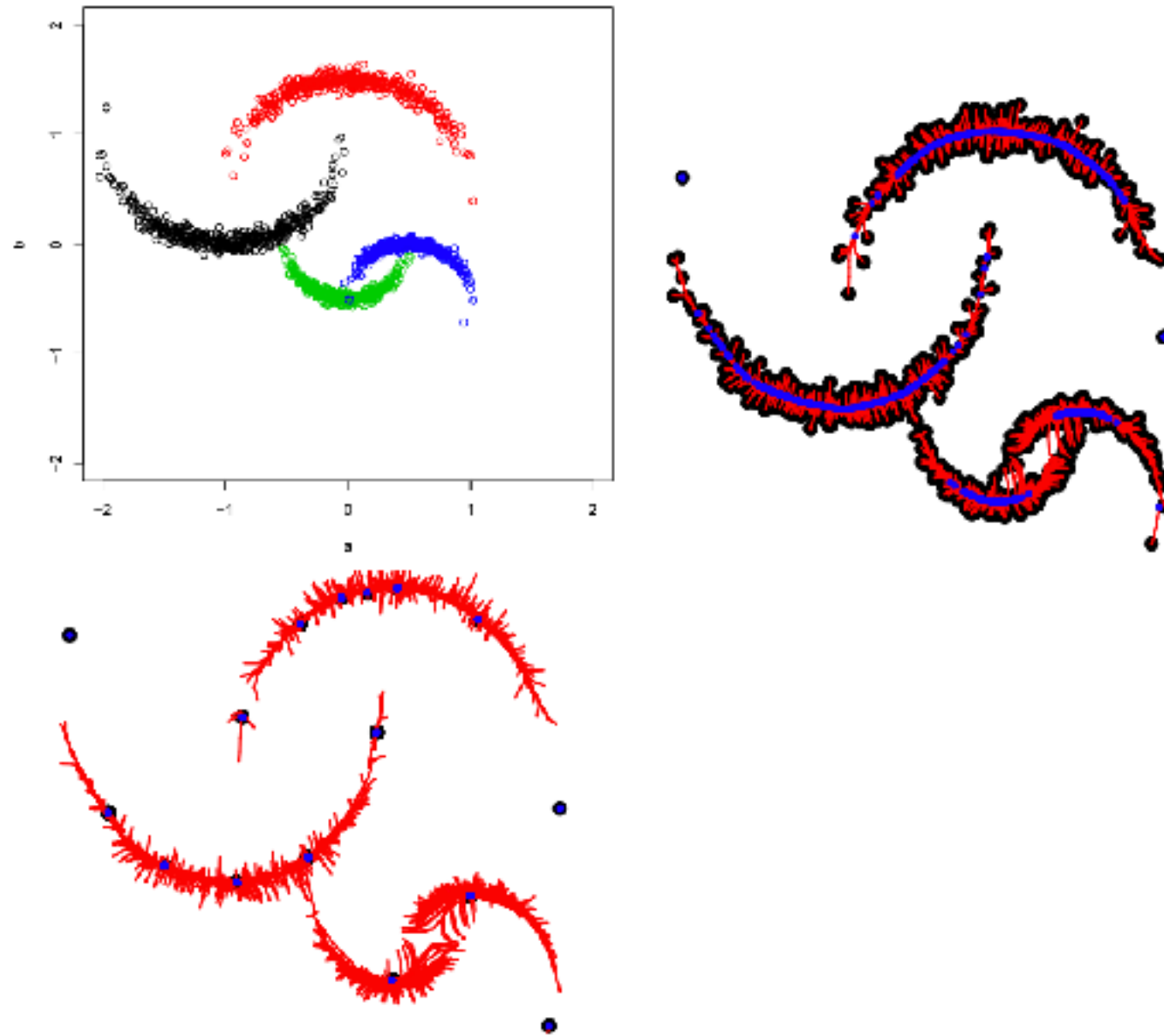


Figure 4: The crescent data example. Top left: data. Top right: a few steps of mean-shift. Bottom left: a few steps of blurred mean-shift.

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

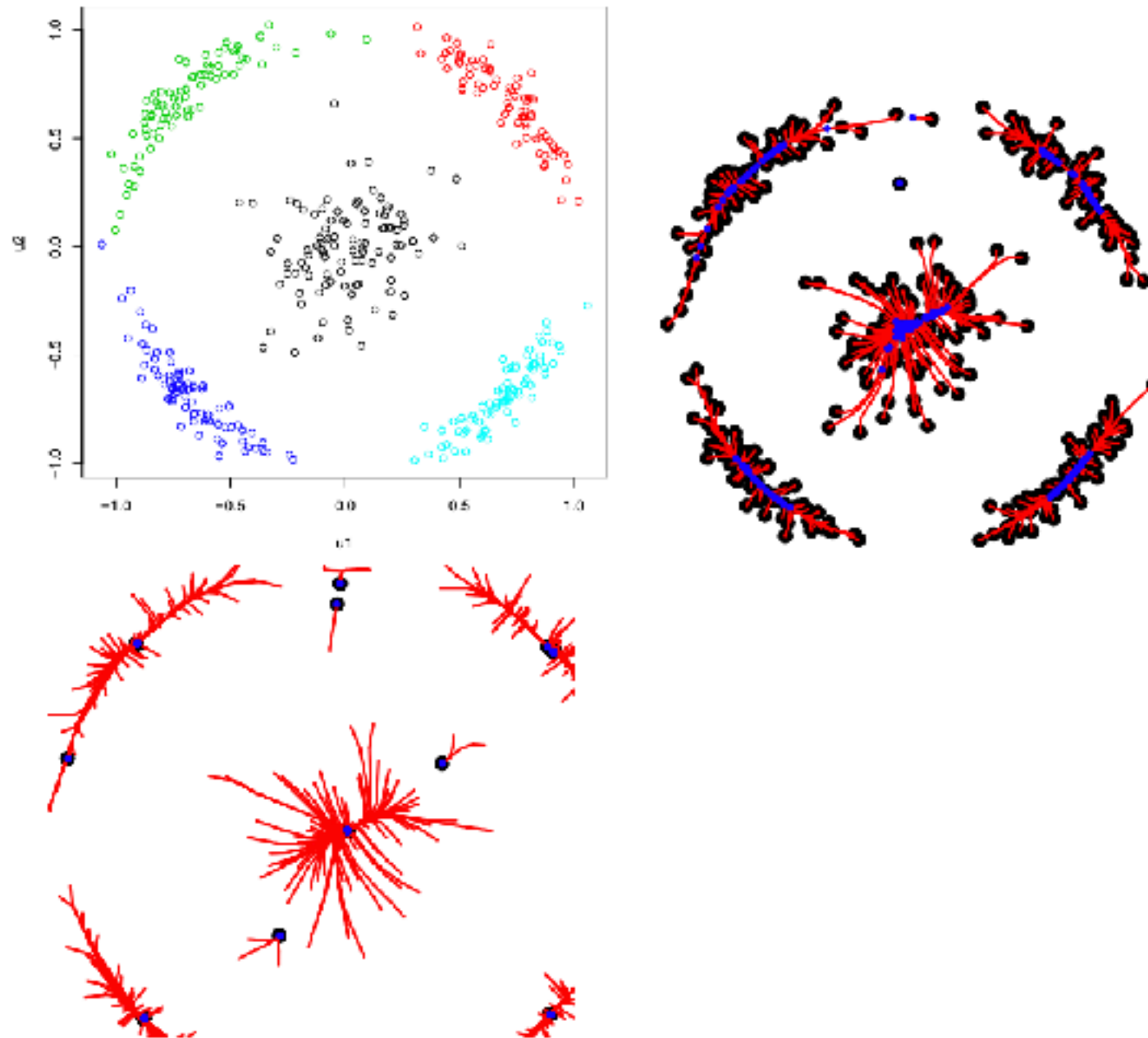


Figure 5: The Broken Ring example. Top left: data. Top right: a few steps of mean-shift. Bottom left: a few steps of blurred mean-shift.

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 :

$$\Pi_C[X] = \operatorname{argmin}_{c \in C} \|c - X\|^2. \quad (1)$$

Define the empirical clustering risk of a codebook C by

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

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

$$\hat{C} = \operatorname{argmin}_{C \in \mathcal{C}_k} R_n(C). \quad (3)$$

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 $g_i = \operatorname{argmin}_j \|X_i - c_j\|$. 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

$$c_j \leftarrow \frac{1}{n_j} \sum_{i: X_i \in C_j} X_i.$$

4. Repeat steps 2 and 3 until convergence.
5. Output: centers $\hat{C} = \{c_1, \dots, c_k\}$ and clusters C_1, \dots, C_k .

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

One step of k-means: Voronoi tessellation

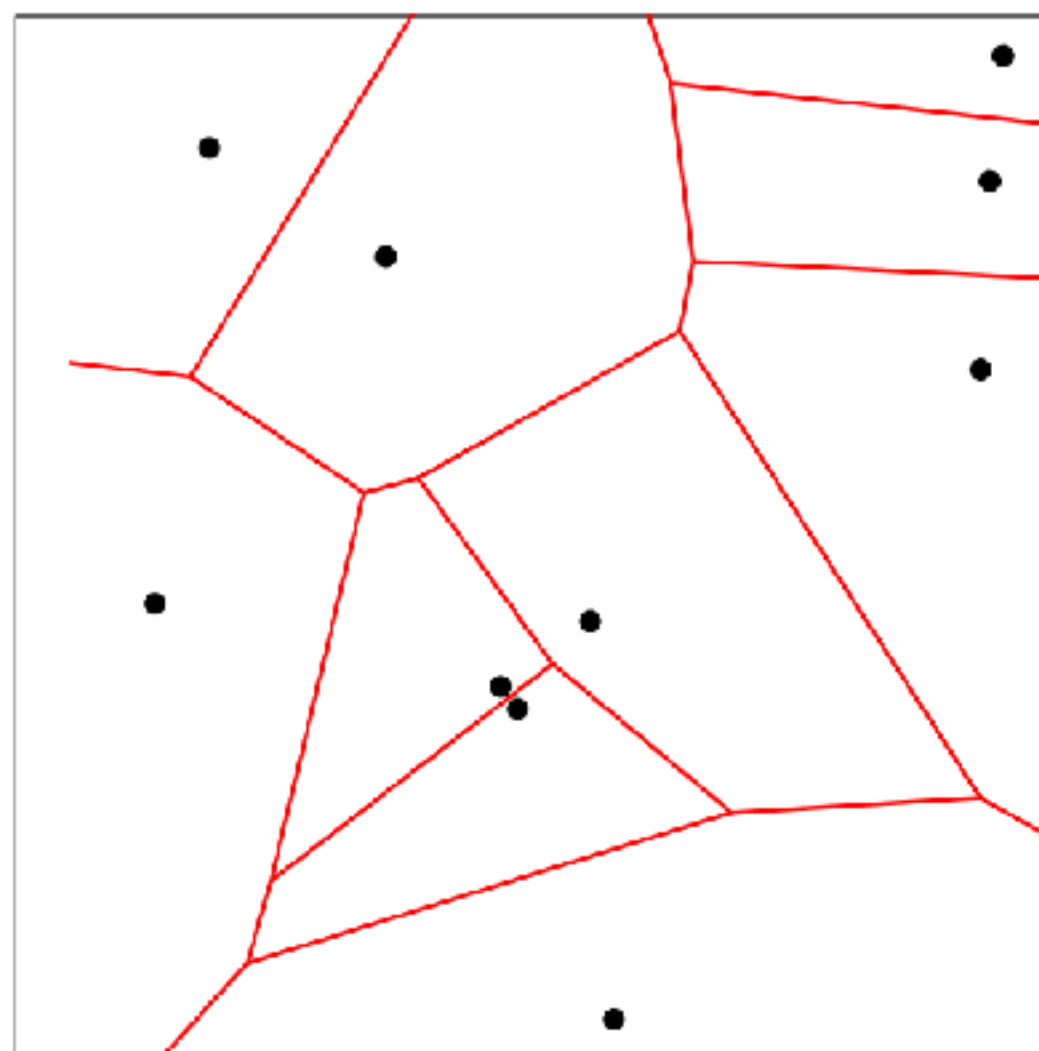


Figure 5: The Voronoi tessellation formed by 10 cluster centers c_1, \dots, c_{10} . The cluster centers are indicated by dots. The corresponding Voronoi cells T_1, \dots, T_{10} are defined as follows: a point x is in T_j if x is closer to c_j than c_i for $i \neq j$.

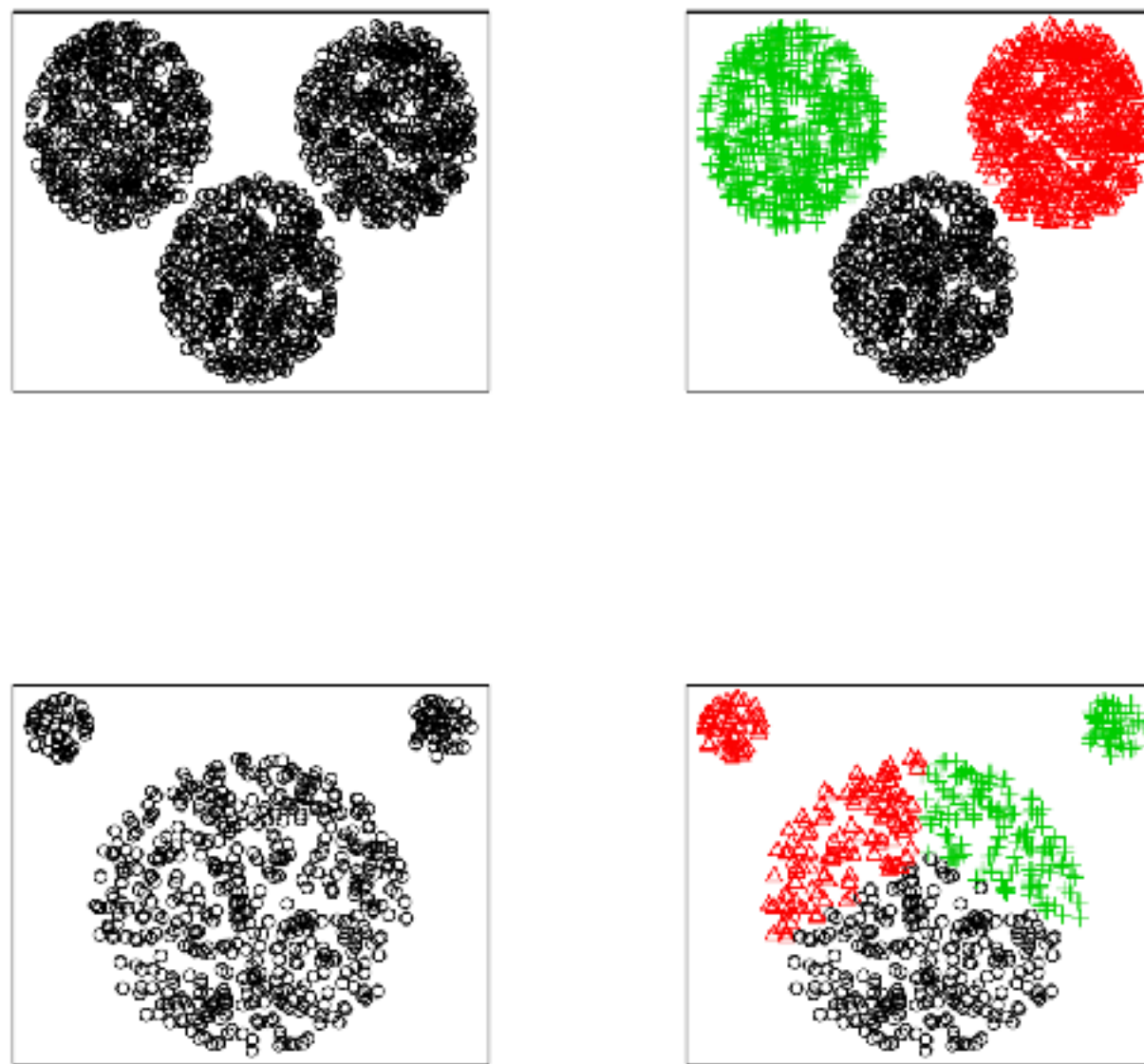


Figure 7: Synthetic data inspired by the “Mickey Mouse” example from wikipedia. Top left: three balanced clusters. Top right: result from running k means with $k = 3$. Bottom left: three unbalanced clusters. Bottom right: result from running k means with $k = 3$ on the unbalanced clusters. k -means does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

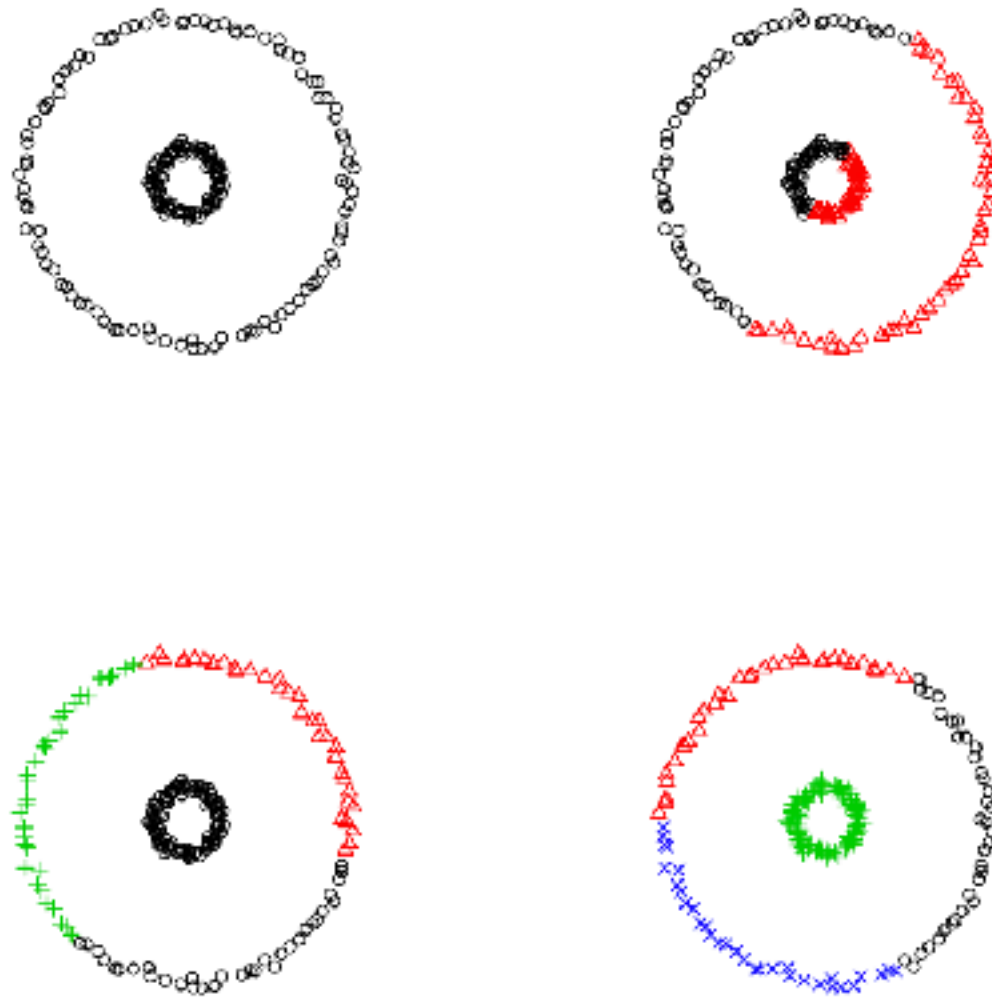


Figure 10: Top left: a dataset with two ring-shaped clusters. Top right: k -means with $k = 2$. Bottom left: k -means with $k = 3$. Bottom right: k -means with $k = 4$.

K-means++

1. **Input:** Data $X = \{X_1, \dots, X_n\}$ and an integer k .
2. Choose c_1 randomly from $X = \{X_1, \dots, X_n\}$. Let $C = \{c_1\}$.
3. For $j = 2, \dots, k$:
 - (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 $C \leftarrow C \cup \{c_j\}$.
4. Run Lloyd's algorithm using the **seed points** $C = \{c_1, \dots, c_k\}$ as starting points and output the result.

Figure 12: The k -means⁺⁺ algorithm.

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

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

where the expectation is over the randomness of the algorithm.

Hierarchical clustering

1. Input: data $X = \{X_1, \dots, X_n\}$ and metric d giving distance between clusters.
2. Let $T_n = \{C_1, C_2, \dots, C_n\}$ 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, \dots, 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	$d(A, B) = \max_{x \in A, y \in B} d(x, y)$

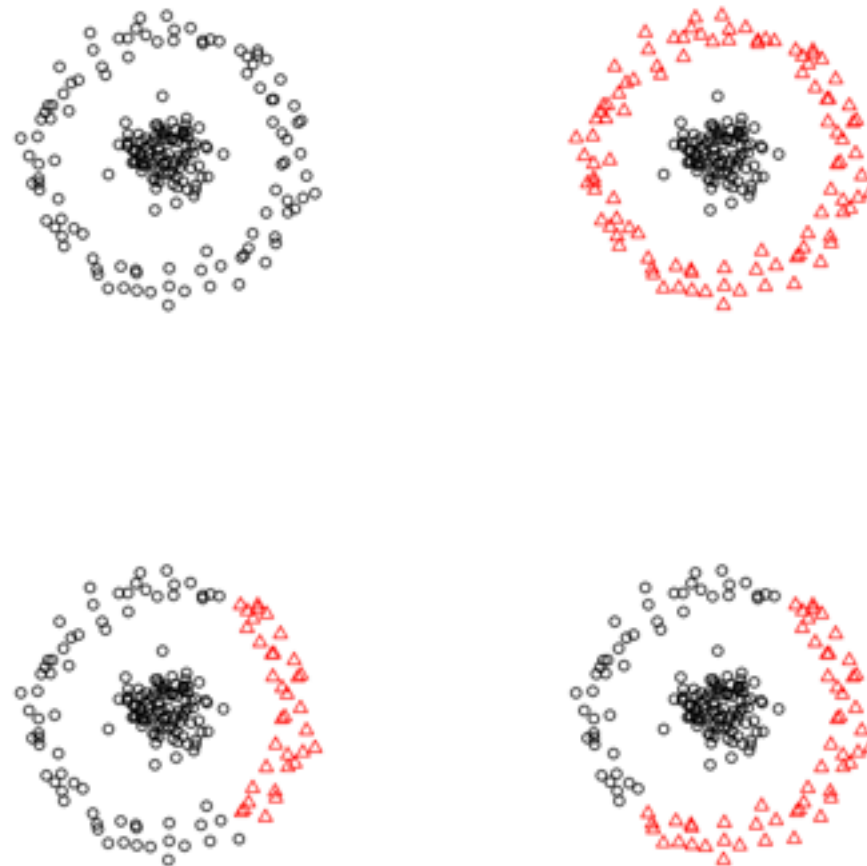


Figure 21: Hierarchical clustering applied to two noisy rings. Top left: the data. Top right: two clusters from hierarchical clustering using single linkage. Bottom left: average linkage. Bottom right: complete linkage.

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	$d(A,B) = \max_{x \in A, y \in B} d(x,y)$

Geometric graph clustering

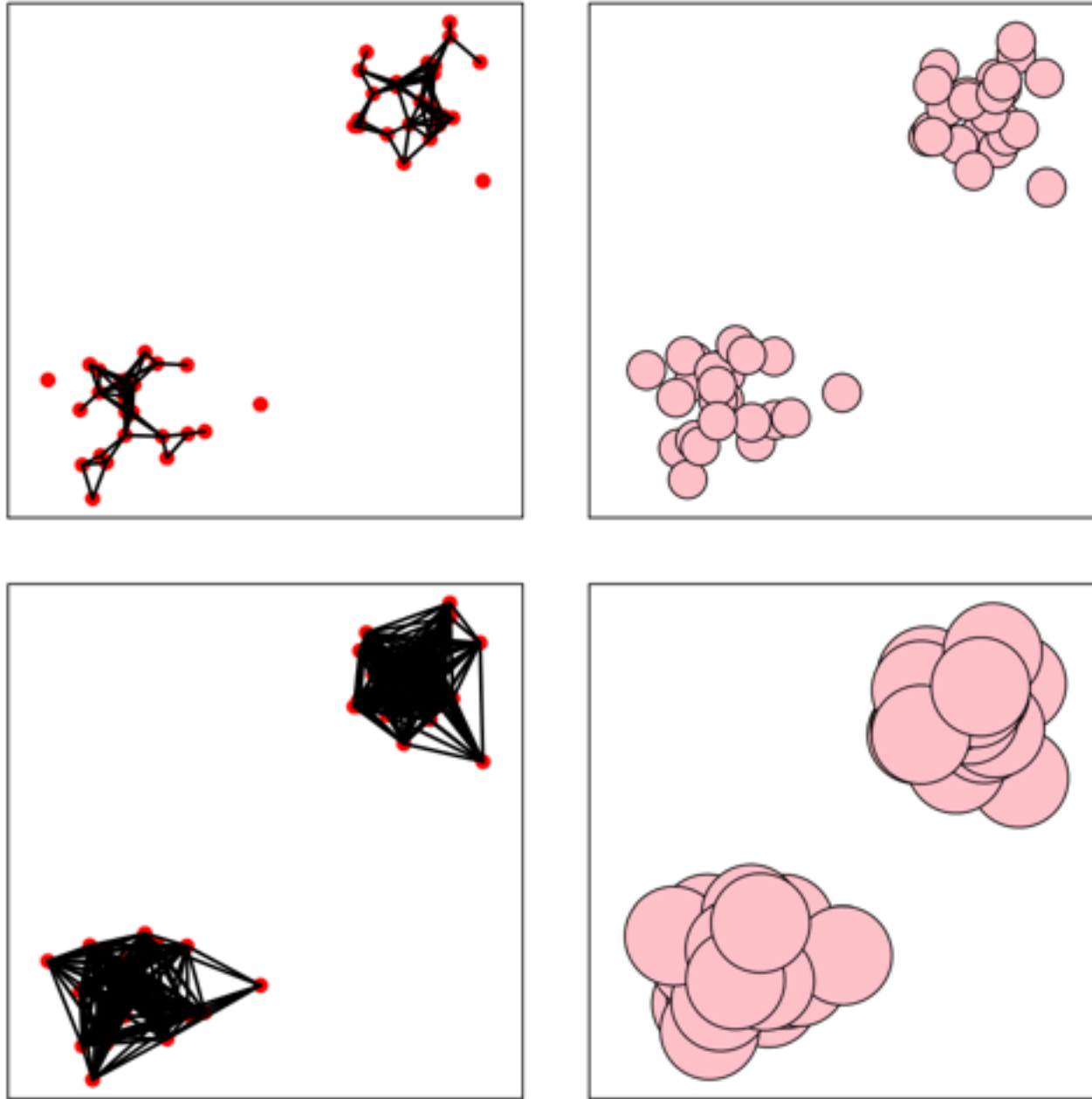


Figure 15: Geometric graph clustering. Top row: $\epsilon = 1$. Bottom row: $\epsilon = 3$. Left column: the graphs. Right column: a ball of radius $\epsilon/2$ around each X_i .

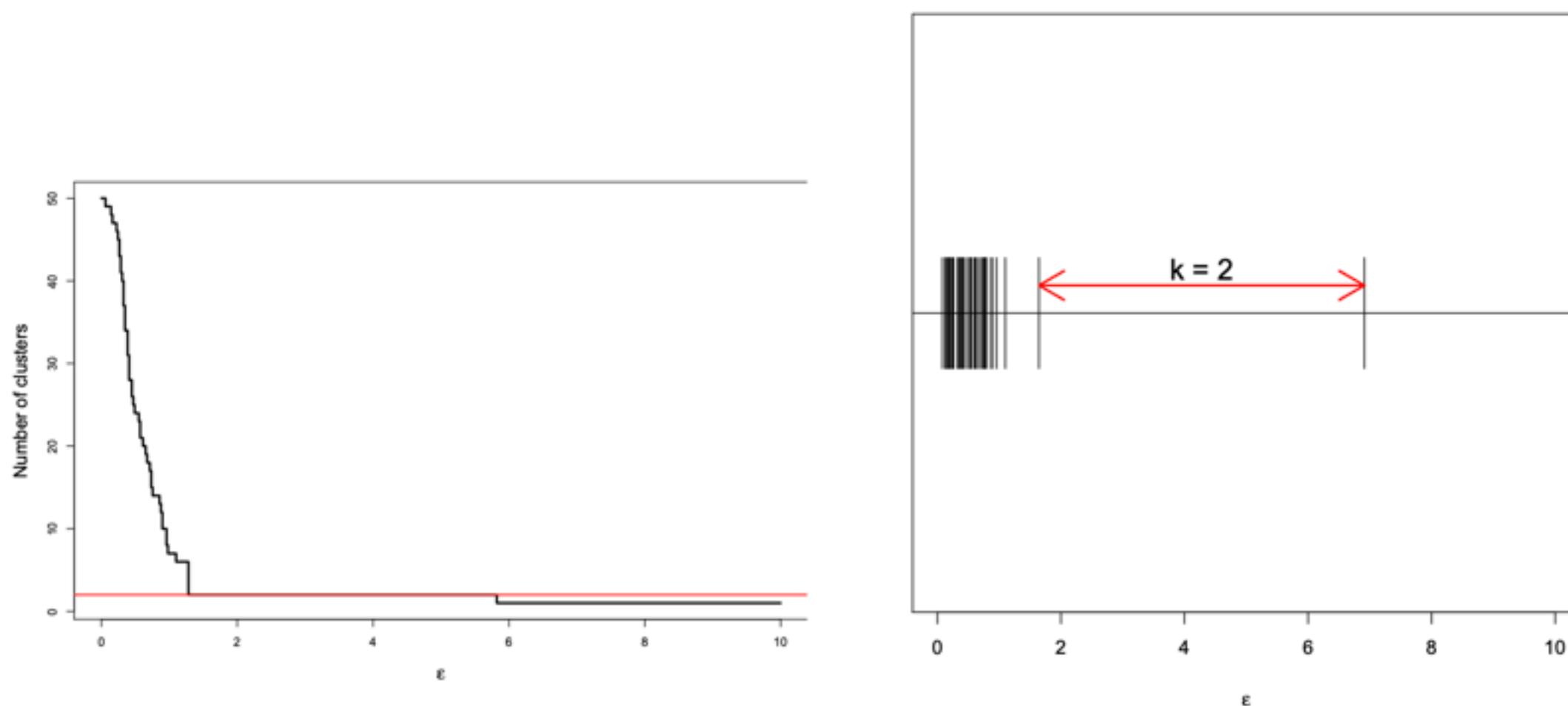


Figure 16: Geometric graph clustering. The top plot shows the number of connected components $k(\epsilon)$ versus ϵ . There is a fairly large range of values of ϵ that yields two connected components. The bottom plot shows the values of ϵ at which $k(\epsilon)$ changes.

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 $f = (f_1, \dots, f_n)^T$,

$$f^T L f = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n W_{ij} (f_i - f_j)^2.$$

2. L is symmetric and positive semi-definite.

3. The smallest eigenvalue of L is 0. The corresponding eigenvector is $(1, 1, \dots, 1)^T$.

4. L has n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_k$.

5. The number of eigenvalues that are equal to 0 is equal to the number of connected components of G . That is, $0 = \lambda_1 = \dots = \lambda_k$ where k is the number of connected components of G . The corresponding eigenvectors v_1, \dots, v_k are orthogonal and each is constant over one of the connected components of the graph.

Example 20 Consider the graph

$$X_1 \text{ --- } X_2 \qquad X_3 \text{ --- } X_4 \text{ --- } X_5$$

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

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

and the Laplacian is

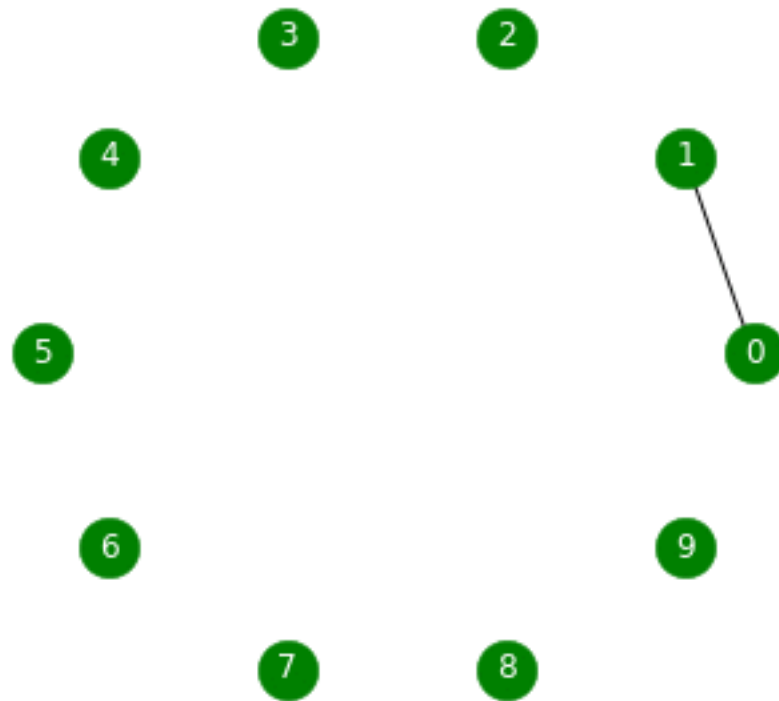
$$L = D - W = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 0 \end{pmatrix}.$$

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

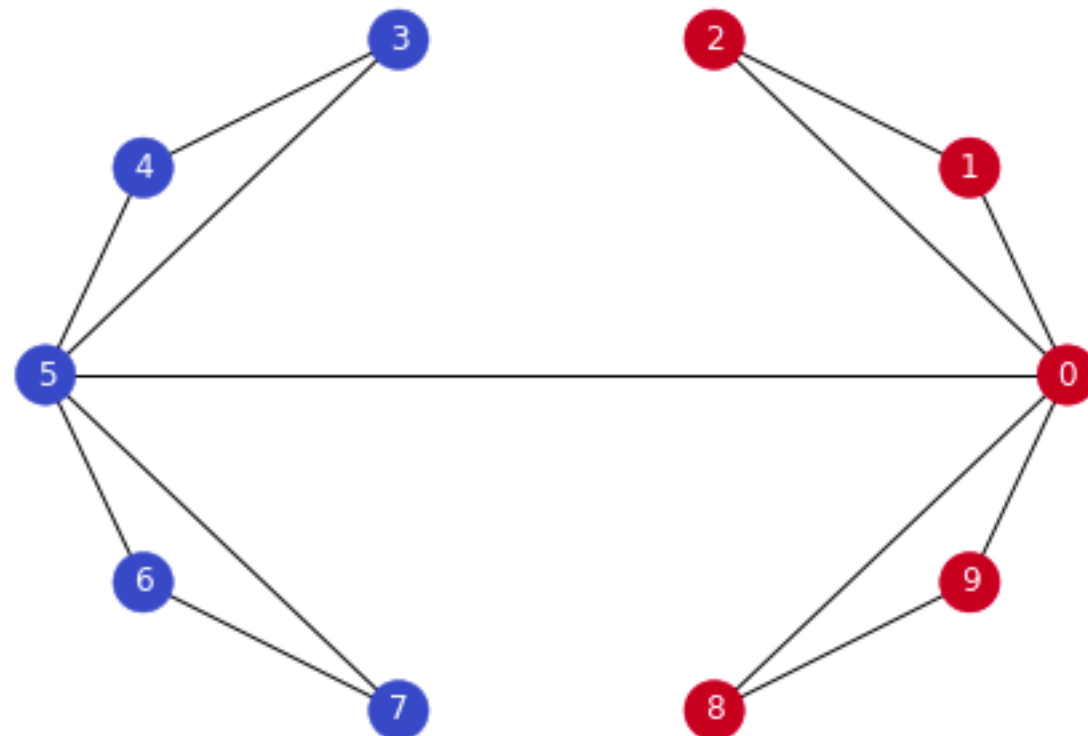
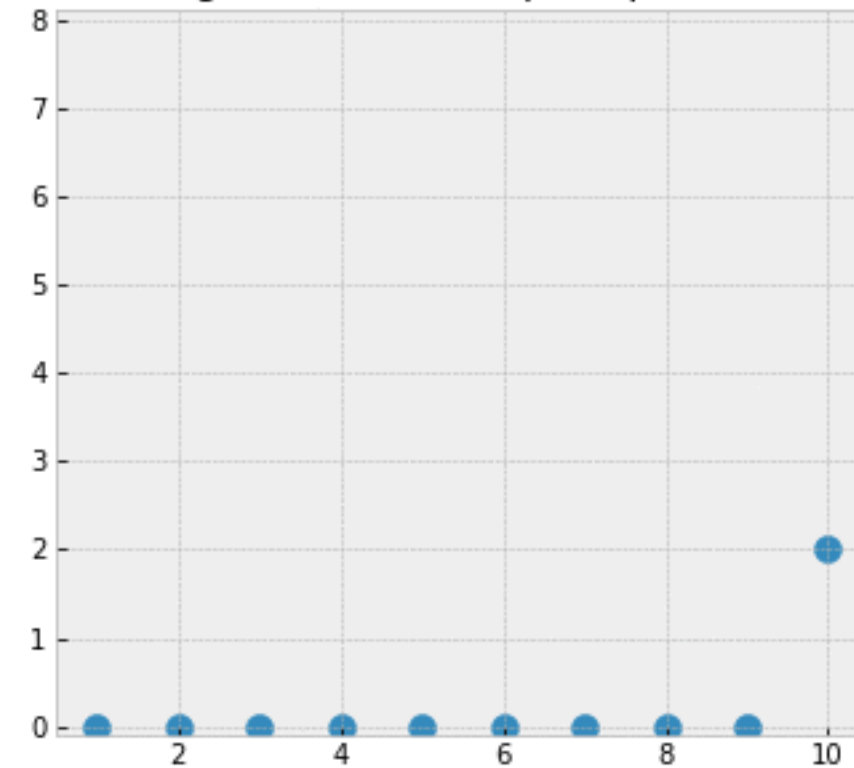
$$v_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad v_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad v_3 = \begin{pmatrix} 0 \\ 0 \\ -.71 \\ 0 \\ .71 \end{pmatrix} \quad v_4 = \begin{pmatrix} -.71 \\ .71 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad v_5 = \begin{pmatrix} 0 \\ 0 \\ -.41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

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

Graph



Eigenvalues of Graph Laplacian



Nodes colored based on the sign of the second eigenvector (“Fiedler vector”). Recall first e-vec is constant.

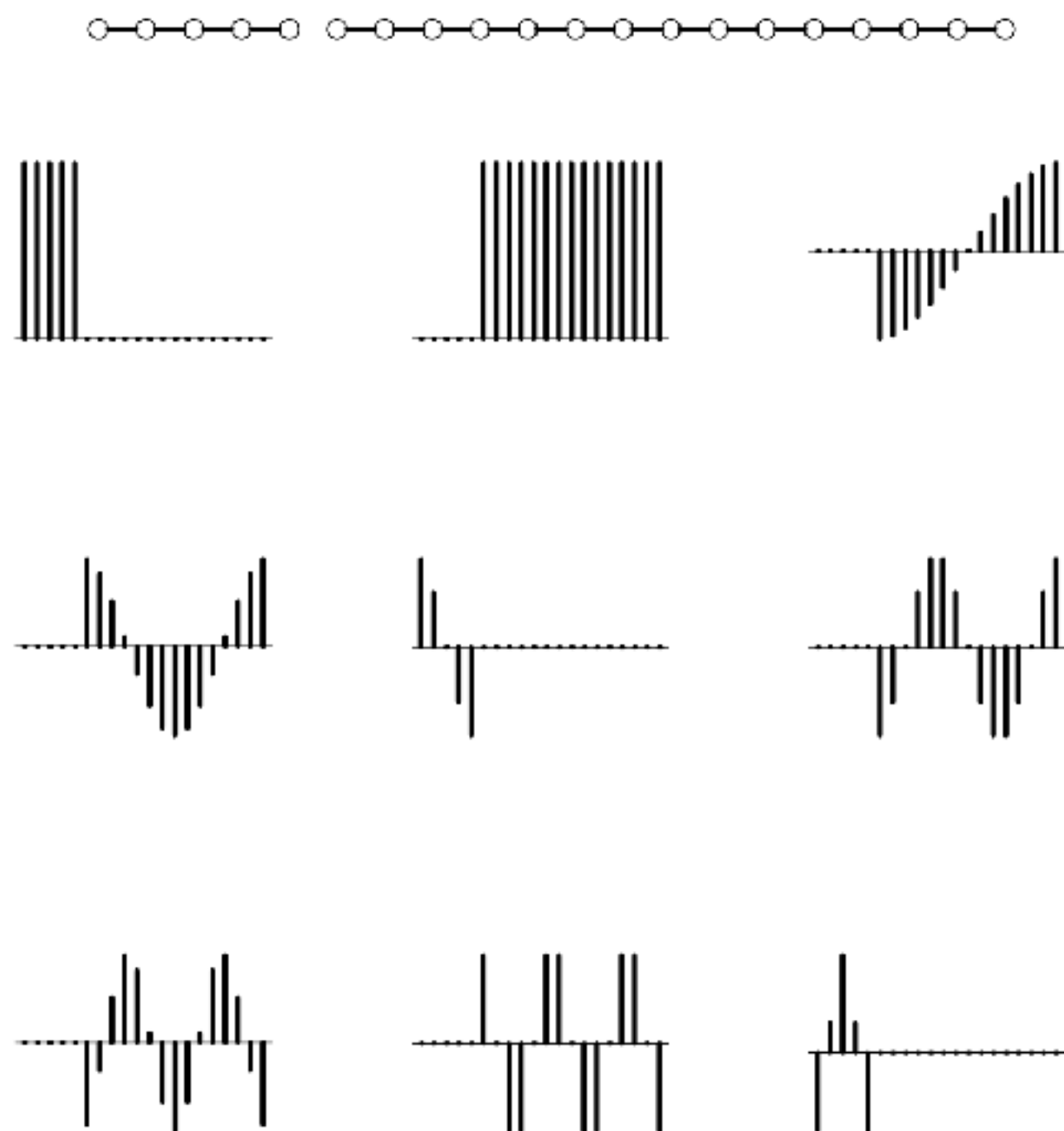


Figure 22: The top shows a simple graph. The remaining plots are the eigenvectors of the graph Laplacian. Note that the first two eigenvectors correspond to the two connected components of the graph.

Three Laplacians: $D - W$, $D^{-1}W$, $I - D^{-1/2}WD^{-1/2}$
 (“Laplacian”, “random walk Laplacian”, “normalized Laplacian”)

Spectral clustering

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 $\mathcal{L} = D^{-1}W$.
3. Find first k eigenvectors v_1, \dots, v_k of \mathcal{L} .
4. Project each X_i onto the eigenvectors to get new points \hat{X}_i .
5. Cluster the points $\hat{X}_1, \dots, \hat{X}_n$ using any standard clustering algorithm.

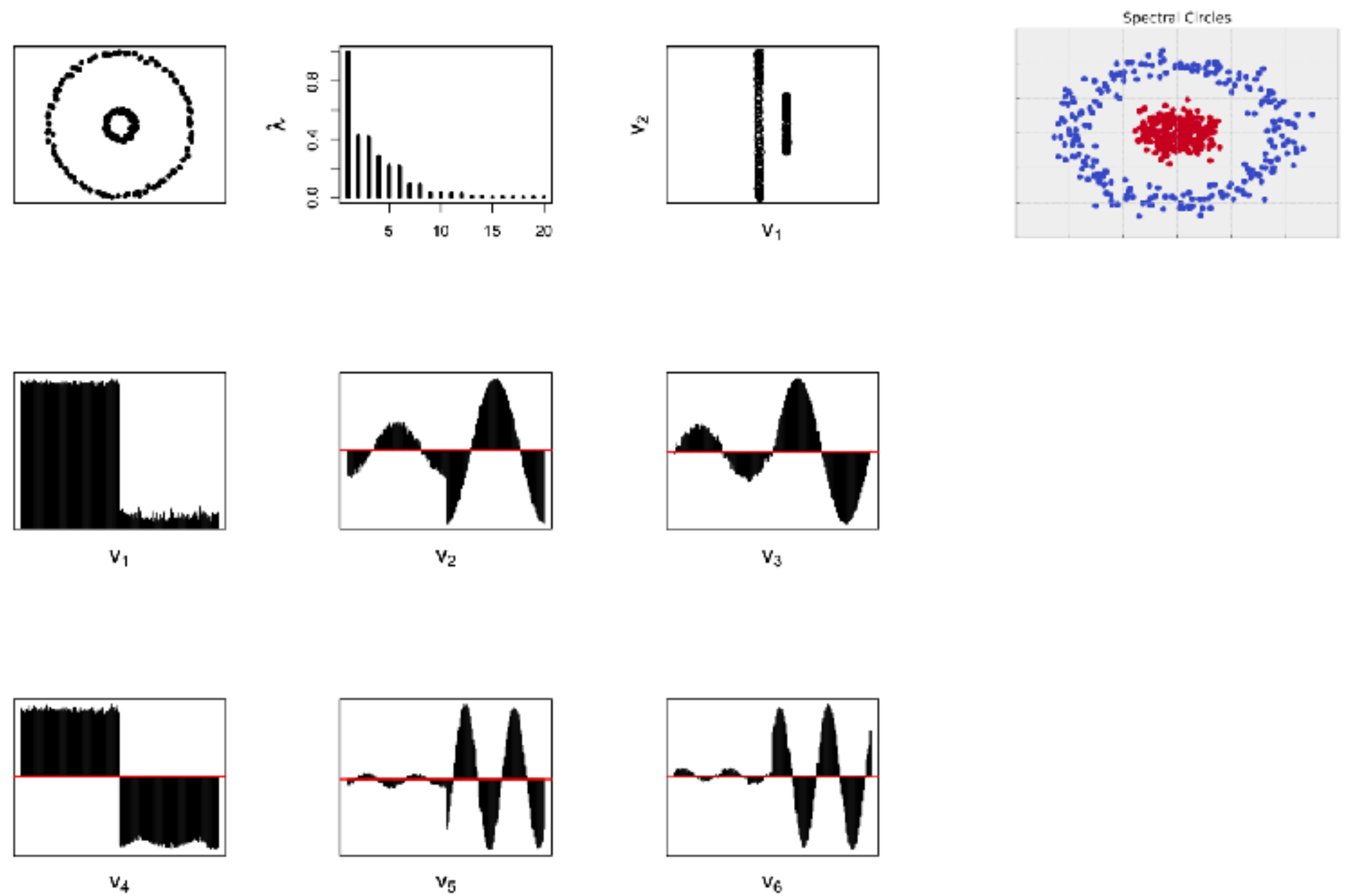


Figure 23: Top left: data. Top middle: eigenvalues. Top right: second versus third eigenvectors. Remaining plots: first six eigenvectors.

Kernelized methods

1. *Kernel spectral clustering (define similarity matrix, ie graph)*
2. *Kernel K-means (define distance metric between points)*
3. *Kernel hierarchical clustering (define distance between clusters)*
4. *Kernel PCA (“eigenfunctions”: functions that explain variance)*
5. ...

“Kernel trick”: any algorithm that can be represented solely in terms of dot products (or distances) can be “kernelized” by replacing $x_i^T x_j$ by $K(x_i, x_j)$ for a chosen kernel K , or replacing $d(x_i, x_j)$ by

$$\|\phi(x_i) - \phi(x_j)\|_K = \sqrt{K(x_i, x_i) + K(x_j, x_j) - 2K(x_i, x_j)}$$

Kernel PCA

Define $\bar{\phi} := \sum_{i=1}^n \phi(x_i)/n$, so $\langle f, \bar{\phi} \rangle_K = \sum_{i=1}^n f(X_i)/n$

$$f_1 := \arg \max_{\|f\|_K \leq 1} \text{Var}_n(f(X)) \equiv \text{Var}_n(\langle f, \phi(X) \rangle_K) \equiv \langle f, C_n f \rangle_K$$

Covariance operator:

$$C_n := \frac{1}{n} \sum_{i=1}^n \tilde{\phi}(x_i) \otimes \tilde{\phi}(x_i)$$

“Matrix-vector” multiplication

$$(a \otimes b)c = \langle b, c \rangle a$$

$$= \arg \max_{\|f\|_K \leq 1} \mathbb{E}_n[f(X) - \mathbb{E}_n f(X)]^2$$

$$= \arg \max_{\|f\|_K \leq 1} \frac{1}{n} \sum_{i=1}^n \langle f, \phi(x_i) - \bar{\phi} \rangle^2$$

Representer theorem style argument:

$$f_1 = \sum_i \alpha_i \tilde{\phi}(x_i)$$

Define $\tilde{\phi}(x_i) := \phi(x_i) - \bar{\phi}$

$$= \arg \max_{\alpha^T \tilde{K} \alpha \leq 1} \alpha^T \tilde{K}^2 \alpha$$

Solution: $\tilde{K} \alpha = \tilde{\lambda}_1 \alpha$ + normalize

After solving for f_1, \dots, f_d , we can project any point x onto this eigenspace as $f_1(x), \dots, f_d(x)$ and we can also represent it in the feature space as

$$P_d(x) := f_1(x)f_1 + f_2(x)f_2 + \dots$$

USPS hand-written digits data:
7191 images of hand-written digits of 16×16 pixels.



Sample of original images (not used for experiments)



Sample of noisy images



Sample of denoised images (linear PCA)



Sample of denoised images (kernel PCA, Gaussian kernel)

Generated by Matlab Stprtool (by V. Franc).

Figure 6.2: Hand-written digit denoising example (from Kenji Fukumizu's slides).

$$y^* = \arg \min_{y \in \mathcal{X}} \|\phi(y) - P_d \phi(x^*)\|_{\mathcal{H}}^2.$$