Classification and Clustering

Classification and clustering are one of the central tasks in machine learning. Given a set of data points, the purpose is to classify the points into subgroups, which express closeness or similarity of the points and which are represented by a cluster head.

Discriminant Analysis

Suppose that g populations/groups/classes $C_1, \dots, C_g$ are given, each represented by a p.d.f. $f_i(\mathbf{x})$ on $\mathbb{R}^p$, $i = 1, \dots, g$.
A discriminant rule divides $\mathbb{R}^p$ into disjoint regions $R_1, \dots, R_g$, $\cup_{i=1}^{p} R_i = \mathbb{R}^p$. The discriminant rule is defined by

$$\text{allocate some observation } \mathbf{x} \text{ to } C_i \text{ if } \mathbf{x} \in R_i\,.$$

Often the densities $f_i(\mathbf{x})$ are completely unknown or its parameters, such as its mean and variance, must be estimated from a training set $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^p$ with known class allocation. This setup, where a training set with known class allocation is given, is called a "supervised" learning setup.

Fisher's Linear Discriminant Function

Fisher's linear discriminant function is a tool for supervised learning, where a training set $\mathbf{x}_1, \dots, \mathbf{x}_n$ with known classification is given. When a new observation $\mathbf{x}$ with unknown classification is obtained, a linear discriminant rule $\mathbf{a}^T\mathbf{x}$ is calculated such that $\mathbf{x}$ is allocated to some class in an optimal way.
Hence, an appropriate linear transformation $\mathbf{a} \in \mathbb{R}^p$ must be computed using the training set. Particularly, in this analysis, $\mathbf{a}$ is chosen such that the ratio of the *between-groups sum of squares* and the *within group sum of squares* is maximized. To formally define this ratio, let $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T$ be samples from g groups $C_1, \dots, C_g$. We as well define $\mathbf{X}_l = [\mathbf{x}_j]_{j \in C_l}$ and $n_l = |\{j : 1 \le j \le n; j \in C_l\}|$. Then, the average of the training set is given by

$$\overline{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \in \mathbb{R}^p\,,$$

and the average over the group C_l is given by

$$\overline{\mathbf{x}}_l = \frac{1}{n_l} \sum_{j \in C_l} \mathbf{x}_j \in \mathbb{R}^p\,.$$

Moreover, since $\mathbf{a} \in \mathbb{R}^p$ is defined to be the linear discriminant of data, the vector $\mathbf{y} \in \mathbb{R}^n$ that stores the discriminant values of the vectors within the training set is given by

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \mathbf{X}^T \mathbf{a}\,.$$

The discriminant values corresponding to the vectors of the training set, that belong to the group C_l, are stored in $\mathbf{y}_l = (y_j)_{j \in C_l}$. Similarly define the general average and between-groups average as

$$\overline{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \quad \text{and} \quad \overline{y}_l = \frac{1}{n_l} \sum_{j \in C_l} y_j\,.$$

Note that

$$\begin{aligned} \sum_{i=1}^{n} (y_i - \overline{y})^2 &= \sum_{l=1}^{g} \sum_{j \in C_l} (y_j - \overline{y}_l + \overline{y}_l - \overline{y})^2 \\ &\overset{(a)}{=} \sum_{l=1}^{g} \left[\sum_{j \in C_l} (y_j - \overline{y}_l)^2 + \sum_{j \in C_l} (\overline{y}_l - \overline{y})^2 \right] \\ &= \sum_{l=1}^{g} \sum_{j \in C_l} (y_j - \overline{y}_l)^2 + \sum_{l=1}^{g} n_l (\overline{y}_l - \overline{y})^2 \end{aligned}$$

where (a) follows from a similar argument behind Steiner's rule -Theorem 3.3. Finally, $\sum_{l=1}^{g} \sum_{j \in C_l} (y_j - \overline{y}_l)^2$ is the sum of squares within groups and $\sum_{l=1}^{g} n_l (\overline{y}_l - \overline{y})^2$ is the sum of squares between groups, so the problem of selecting the optimal $\mathbf{a}$ is formally defined.

To address this problem with compact notation, let $\mathbf{E}_n$ and $\mathbf{E}_{n_l} = \mathbf{E}_l$, $l = 1, \ldots, g$ be centering operators. Using matrix notation, we have

$$\begin{aligned} \sum_{l=1}^{g} \sum_{j \in C_l} (y_j - \overline{y}_l)^2 &= \sum_{l=1}^{g} \mathbf{y}_l^T \mathbf{E}_l \mathbf{y}_l \\ &= \sum_{l=1}^{g} \mathbf{a}^T \mathbf{X}_l^T \mathbf{E}_l \mathbf{X}_l \mathbf{a} \\ &= \mathbf{a}^T \Big(\sum_{l=1}^{g} \mathbf{X}_l^T \mathbf{E}_l \mathbf{X}_l \Big) \mathbf{a} = \mathbf{a}^T \mathbf{W} \mathbf{a}. \end{aligned}$$

where $\mathbf{W} = \sum_{l=1}^{g} \mathbf{X}_l^T \mathbf{E}_l \mathbf{X}_l$. Similarly,

$$\begin{aligned}\sum_{l=1}^{g} n_l(\bar{y}_l - \bar{y})^2 &= \sum_{l=1}^{g} n_l(\mathbf{a}^T(\bar{\mathbf{x}}_\mathbf{l} - \bar{\mathbf{x}}))^2 \\ &= \sum_{l=1}^{g} n_l \mathbf{a}^T(\bar{\mathbf{x}}_\mathbf{l} - \bar{\mathbf{x}})(\bar{\mathbf{x}}_\mathbf{l} - \bar{\mathbf{x}})^T\mathbf{a} \\ &= \mathbf{a}^T\left(\sum_{l=1}^{g} n_l(\bar{\mathbf{x}}_\mathbf{l} - \bar{\mathbf{x}})(\bar{\mathbf{x}}_\mathbf{l} - \bar{\mathbf{x}})^T\right)\mathbf{a} = \mathbf{a}^T\mathbf{B}\mathbf{a},\end{aligned}$$

where $\mathbf{B} = \sum_{l=1}^{g} n_l(\bar{\mathbf{x}}_\mathbf{l} - \bar{\mathbf{x}})(\bar{\mathbf{x}}_\mathbf{l} - \bar{\mathbf{x}})^T$. Then, the linear discriminant analysis requires obtaining the $\mathbf{a}$ that solves

$$\max_{\mathbf{a}\in\mathbb{R}^p} \frac{\mathbf{a}^T\mathbf{B}\mathbf{a}}{\mathbf{a}^T\mathbf{W}\mathbf{a}} \quad (\star)$$

Theorem 5.1. *The maximum value of* $(\star)$ *is attained at the eigenvector of* $\mathbf{W}^{-1}\mathbf{B}$ *corresponding to the largest eigenvalue.*

Proof. Assuming $\mathbf{a} = \mathbf{W}^{-1/2}\mathbf{b}$, we have

$$\max_{\mathbf{a}\in\mathbb{R}^p} \frac{\mathbf{a}^T\mathbf{B}\mathbf{a}}{\mathbf{a}^T\mathbf{W}\mathbf{a}} = \max_{\mathbf{b}\in\mathbb{R}^p} \frac{\mathbf{b}^T\mathbf{W}^{-1/2}\mathbf{B}\mathbf{W}^{-1/2}\mathbf{b}}{\mathbf{b}^T\mathbf{b}} = \lambda_{\max}(\mathbf{W}^{-1/2}\mathbf{B}\mathbf{W}^{-1/2}),$$

where the last part results from Theorem 2.4. Furthermore $\mathbf{W}^{-1/2}\mathbf{B}\mathbf{W}^{-1/2}$ and $\mathbf{W}^{-1}\mathbf{B}$ have the same eigenvalues, since:

$$\mathbf{W}^{-1}\mathbf{B}\mathbf{v} = \lambda\mathbf{v} \iff \mathbf{W}^{-1/2}\mathbf{B}\mathbf{v} = \lambda\mathbf{W}^{1/2}\mathbf{v} \iff \mathbf{W}^{-1/2}\mathbf{B}\mathbf{W}^{-1/2}\mathbf{W}^{1/2}\mathbf{v} = \lambda\mathbf{W}^{1/2}\mathbf{v}.$$

Therefore the two matrices have the same eigenvalues. Moreover suppose that $\mathbf{v}$ is the eigenvector of $\mathbf{W}^{-1}\mathbf{B}$ corresponding to $\lambda_{\max}$. Then we have

$$\frac{\mathbf{v}^T\mathbf{B}\mathbf{v}}{\mathbf{v}^T\mathbf{W}\mathbf{v}} = \frac{\mathbf{v}^T\mathbf{B}\mathbf{v}}{\mathbf{v}^T\mathbf{W}(\frac{1}{\lambda_{\max}}\mathbf{W}^{-1}\mathbf{B}\mathbf{v})} = \lambda_{\max}.$$

□

The linear function $\mathbf{a}^T\mathbf{x}$ is called Fisher's linear discriminant function or the first canonical variate. The ratio is invariant with the respect to scaling of $\mathbf{a}$.
The application of the linear discriminant analysis is as follows.

- Given the training set $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ with known groups, compute the optimum $\mathbf{a}$ from Theorem 5.1.
- For a new observation $\mathbf{x}$, compute $\mathbf{a}^T\mathbf{x}$.

- Allocate $\mathbf{x}$ to the group with closest value of $\mathbf{a}^T\overline{\mathbf{x}}_\mathbf{l} = \overline{y}_l$. Discriminant rule can be formulated as

  Discriminant Rule: Allocate $\mathbf{x}$ to the group l if $|\mathbf{a}^T\mathbf{x} - \mathbf{a}^T\overline{\mathbf{x}}_\mathbf{l}| < |\mathbf{a}^T\mathbf{x} - \mathbf{a}^T\overline{\mathbf{x}}_\mathbf{j}|$ for all $j = 1, \ldots, l-1, l+1, \ldots, g$.

Fisher's discriminant function is immediately relevant for the special case of $g = 2$, where there are two groups of size n_1 and n_2 with $n = n_1 + n_2$. In this case we have

$$\begin{aligned}
\mathbf{B} &= n_1(\overline{\mathbf{x}}_\mathbf{1} - \overline{\mathbf{x}})(\overline{\mathbf{x}}_\mathbf{1} - \overline{\mathbf{x}})^T + n_2(\overline{\mathbf{x}}_\mathbf{2} - \overline{\mathbf{x}})(\overline{\mathbf{x}}_\mathbf{2} - \overline{\mathbf{x}})^T \\
&= n_1(\overline{\mathbf{x}}_\mathbf{1} - \frac{n_1}{n}\overline{\mathbf{x}_\mathbf{1}} - \frac{n_2}{n}\overline{\mathbf{x}}_\mathbf{2})(\overline{\mathbf{x}}_\mathbf{1} - \frac{n_1}{n}\overline{\mathbf{x}_\mathbf{1}} - \frac{n_2}{n}\overline{\mathbf{x}}_\mathbf{2})^T + n_2(\overline{\mathbf{x}}_\mathbf{2} - \frac{n_2}{n}\overline{\mathbf{x}_\mathbf{2}} - \frac{n_1}{n}\overline{\mathbf{x}}_\mathbf{1})(\overline{\mathbf{x}}_\mathbf{2} - \frac{n_2}{n}\overline{\mathbf{x}_\mathbf{2}} - \frac{n_1}{n}\overline{\mathbf{x}}_\mathbf{1})^T \\
&= n_1(\frac{n_2}{n}\overline{\mathbf{x}_\mathbf{1}} - \frac{n_2}{n}\overline{\mathbf{x}}_\mathbf{2})(\frac{n_2}{n}\overline{\mathbf{x}_\mathbf{1}} - \frac{n_2}{n}\overline{\mathbf{x}}_\mathbf{2})^T + n_2(\frac{n_1}{n}\overline{\mathbf{x}_\mathbf{2}} - \frac{n_1}{n}\overline{\mathbf{x}}_\mathbf{1})(\frac{n_1}{n}\overline{\mathbf{x}_\mathbf{2}} - \frac{n_1}{n}\overline{\mathbf{x}}_\mathbf{1})^T \\
&= \frac{n_1 n_2^2}{n^2}(\overline{\mathbf{x}_\mathbf{1}} - \overline{\mathbf{x}}_\mathbf{2})(\overline{\mathbf{x}_\mathbf{1}} - \overline{\mathbf{x}}_\mathbf{2})^T + \frac{n_2 n_1^2}{n^2}(\overline{\mathbf{x}_\mathbf{1}} - \overline{\mathbf{x}}_\mathbf{2})(\overline{\mathbf{x}_\mathbf{1}} - \overline{\mathbf{x}}_\mathbf{2})^T \\
&= \frac{n_1 n_2}{n}(\overline{\mathbf{x}_\mathbf{1}} - \overline{\mathbf{x}}_\mathbf{2})(\overline{\mathbf{x}_\mathbf{1}} - \overline{\mathbf{x}}_\mathbf{2})^T = \frac{n_1 n_2}{n}\mathbf{d}\mathbf{d}^T,
\end{aligned}$$

where $\mathbf{d} = \overline{\mathbf{x}_\mathbf{1}} - \overline{\mathbf{x}}_\mathbf{2}$. Therefore $\mathbf{B}$ has rank one and only one eigenvalue that is not equal to 0. Therefore $\mathbf{W}^{-1}\mathbf{B}$ has only one non-zero eigenvalue, which is given by

$$\mathrm{tr}(\mathbf{W}^{-1}\mathbf{B}) = \frac{n_1 n_2}{n}\mathbf{d}^T\mathbf{W}^{-1}\mathbf{d}.$$

Since $\mathbf{W}$ is nonnegative definite, the above value is nonnegative and therefore is the maximum eigenvalue. Note that $\mathbf{d}$ is an eigenvector of $\mathbf{B}$. We have

$$\begin{aligned}
(\mathbf{W}^{-1}\mathbf{B})\mathbf{W}^{-1}\mathbf{d} &= \mathbf{W}^{-1}(\frac{n_1 n_2}{n}\mathbf{d}\mathbf{d}^T)\mathbf{W}^{-1}\mathbf{d} \\
&= \frac{n_1 n_2}{n}\mathbf{W}^{-1}\mathbf{d}\left(\mathbf{d}^T\mathbf{W}^{-1}\mathbf{d}\right) \\
&= \left(\frac{n_1 n_2}{n}\mathbf{d}^T\mathbf{W}^{-1}\mathbf{d}\right)\mathbf{W}^{-1}\mathbf{d}.
\end{aligned}$$

Therefore, $\mathbf{W}^{-1}\mathbf{d}$ is an eigenvector of $\mathbf{W}^{-1}\mathbf{B}$ corresponding to the eigenvalue $\frac{n_1 n_2}{n}\mathbf{d}^T\mathbf{W}^{-1}\mathbf{d}$. Then, the discriminant rule becomes

- Allocate $\mathbf{x}$ to C_1 if $\mathbf{d}^T\mathbf{W}^{-1}(\mathbf{x} - \frac{1}{2}(\overline{\mathbf{x}}_\mathbf{1} + \overline{\mathbf{x}}_\mathbf{2})) > 0$.

$\mathbf{a} = \mathbf{W}^{-1}\mathbf{d}$ is normal to the discriminating hyperplane between the classes.

One advantage of Fischer's approach is that it is distribution free. It is based on the general principle that the between-groups sum of squares is large relative to the within-groups sum of squares, which measured by the quotient of these two quantities.

Gaussian Maximum Likelihood (ML) Discriminant Rule

In general, the maximum likelihood rule allocates observation $\mathbf{x}$ to the class C_l which maximizes the likelihood $L_l(\mathbf{x}) = \max_j L_j(\mathbf{x})$. A Gaussian ML rule is the particular case when the likelihood functions $L_l(\mathbf{x})$ are Gaussian. Assume that the class distributions are Gaussian and known as $\mathcal{N}_p(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$ with $\boldsymbol{\mu}_l$ and $\boldsymbol{\Sigma}_l$ fixed and with densities

$$f_l(\mathbf{u}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_l|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{u}-\boldsymbol{\mu}_l)^T\boldsymbol{\Sigma}_l^{-1}(\mathbf{u}-\boldsymbol{\mu}_l)\right\}, \mathbf{u} \in \mathbb{R}^p.$$

Then, the Gaussian ML discriminant rule would assign a given $\mathbf{x}$ to the class C_l that maximizes $f_l(x)$ over l.

Theorem 5.2. *The ML discriminant allocates* $\mathbf{x}$ *to class* C_l *which maximizes* $f_l(\mathbf{x})$ *over* $l = 1, \ldots, g$.

(a) If $\boldsymbol{\Sigma}_l = \boldsymbol{\Sigma}$ *for all* l*, then the ML rule allocates* $\mathbf{x}$ *to* C_l *which minimizes the Mahalanobis distance:*

$$(\mathbf{x}-\boldsymbol{\mu}_l)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_l).$$

(b) If $g = 2$*, and* $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$*, then the ML rule allocates* $\mathbf{x}$ *to the class* C_1 *if*

$$\boldsymbol{\alpha}^T(\mathbf{x}-\boldsymbol{\mu}) > 0,$$

where $\boldsymbol{\alpha} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ *and* $\boldsymbol{\mu} = \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)$.

Proof. Part (a) follows directly from the definition of ML discriminant rule. The ML discriminant finds the class l such that:

$$l = \arg\max_{1\le j\le g} f_j(\mathbf{x}).$$

Since $\boldsymbol{\Sigma}$ is fixed for all classes, the maximization of $f_l(\mathbf{x})$ amounts to maximization of exponent which is minimization of the Mahalanobis distance. Part (b) is an exercise. □

Note that the rule (b) is analogue to Fisher's discriminant rule with parameters $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}$ substituting estimates $\overline{\mathbf{x}}_1$, $\overline{\mathbf{x}}_2$ and $\mathbf{W}$.

Application in practice: $\boldsymbol{\Sigma}_l$ and $\boldsymbol{\mu}_l$ are mostly not known. One can estimate these parameters from a training set with known allocations as $\hat{\boldsymbol{\Sigma}}_l$ and $\hat{\boldsymbol{\mu}}_l$ for $l = 1, \ldots, g$. Substitute $\boldsymbol{\Sigma}_l$ and $\boldsymbol{\mu}_l$ by their ML estimates $\hat{\boldsymbol{\Sigma}}_l$ and $\hat{\boldsymbol{\mu}}_l$ and compute the ML discriminant rule.

Cluster Analysis

The aim of cluster analysis is to group n objects into g homogeneous classes. g is normally unknown, but usually assumed to be much smaller than n. Homogeneous means that objects are close to each other. Members of different groups are significantly discriminable. A certain metric is needed to define success. Note that this problem is a case of unsupervised learning, since none of the n objects are known to belong to a certain group.

$k-$means Clustering

Given the training set $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, the purpose of $k-$means Clustering is to partition the data set into clusters $C_1, \ldots, C_g$ with centers in each cluster $\mathbf{u}_1, \ldots, \mathbf{u}_g$ as solution to:

$$\min_{C_1,\ldots,C_g,\mathbf{u}_1,\ldots,\mathbf{u}_g} \sum_{l=1}^{k} \sum_{i \in C_l} \|\mathbf{x}_i - \mathbf{u}_l\|^2$$

Since the optimization problem above is quite difficult to solve, the problem can be modified as the optimal centers are $\overline{\mathbf{x}}_l = \frac{1}{n_l} \sum_{j \in C_l} \mathbf{x}_j \in \mathbf{R}^p$, when given the partition.

Algorithm 1 $k-$means algorithm - Lloyd's algorithm

1: **procedure** $k-$MEANS ALGORITHM
2: **repeat**
3: Given centers $\mathbf{u}_1, \ldots, \mathbf{u}_g$, each point $\mathbf{x}_i$ is assigned to cluster $l = \arg\min_j \|\mathbf{x}_i - \mathbf{u}_j\|^2$
4: update the centers $\mathbf{u}_l = \frac{1}{n_l} \sum_{i \in C_l} \mathbf{x}_i$
5: **until** no more new data

The disadvantages of $k-$means clustering is that the number of clusters of g needs knowing priori, and Euclidean space is needed as well. The iteration may end in sub-optimal solutions which has always convex clusters. This is particularly difficult for the data pattern in Figure 5.1. We discuss this clustering problem in the next part.

Spectral Clustering

To overcome these difficulties, when given the data set $\mathbf{x}_1, \ldots, \mathbf{x}_n$, a weighted graph could be constructed $G = (V, E, \mathbf{W})$, each point $\mathbf{x}_i$ is a vertex $\mathbf{v}_i, i = 1, \ldots, n$, and the edges weights $\mathbf{w}_{i,j}$ are $\mathbf{w}_{i,j} = \mathbf{K}_\varepsilon(\|\mathbf{x}_i - \mathbf{x}_j\|)$ with kernel $\mathbf{K}_\varepsilon$, e.g., $\mathbf{K}_\varepsilon(u) = \exp(-\frac{1}{2\varepsilon}u^2)$. Here, it should be noted that $\|\mathbf{x}_i - \mathbf{x}_j\|$ can be substituted by any dissimilarity measure. Now, lets consider a random walk with transition matrix

$$\mathbf{M} = \mathbf{D}^{-1}\mathbf{W}\,.$$

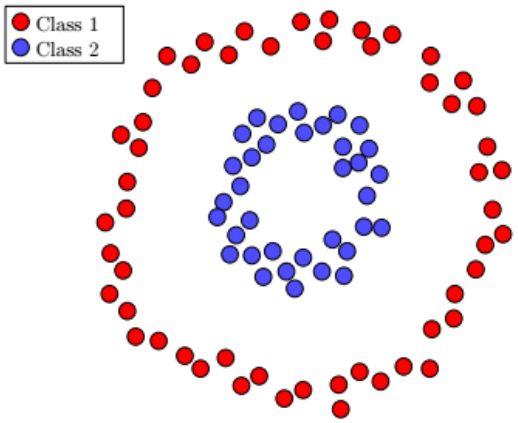

Figure 5.1: Example where k-means clustering is sub-optimal.

We assume that this matrix is constructed using a diffusion map as in Section 4.3, thus

$$\mathbf{P}(\mathbf{X}_{t+1} = j | \mathbf{X}_t = i) = \frac{w_{ij}}{\deg(i)} = M_{ij}, \tag{5.1a}$$

$$\mathbf{D} = \mathrm{diag}(\deg(i)), \deg(i) = \sum_{l=1}^{n} w_{il}\,. \tag{5.1b}$$

Therefore, $\mathbf{M}$ can be decomposed into

$$\mathbf{M} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Psi}^T = \sum_{k=1}^{n} \lambda_k \varphi_k \psi_k^T\,, \tag{5.2a}$$

$$\mathbf{\Phi} = (\varphi_1, \ldots, \varphi_n), \quad \mathbf{\Psi} = (\psi_1, \ldots, \psi_n)\,, \tag{5.2b}$$

since $\mathbf{M}$ is a biorthonormal system with $\mathbf{\Phi}(\mathbf{\Psi})$ as its right(left) eigenvectors. Then, $\mathbf{M}^t = \sum_{k=1}^{n} \lambda_k^t \varphi_k \psi_k^T$, with $\mathbf{m}_{i,j}^{(t)}$ denoting distribution of being in vertex j having started from i. So the whole distribution of being having started from i can be denoted as

$$\mathbf{v}_i \to \mathbf{e}_i^T \mathbf{M}^t = \sum_{k=1}^{n} \lambda_k^t e_i \varphi_k \psi_k^T = \sum_{k=1}^{n} \lambda_k^t \varphi_{k,i} \psi_k^T$$

where $\lambda_k^t \varphi_{k,i}$ are the coefficients, and ψ_k^T the orthonormal basis.
If $\mathbf{v}_i, \mathbf{v}_j$ are close or strongly connected, then $e_i^T \mathbf{M}^t$ and $e_j^T \mathbf{M}^t$ are similar. Moreover, this diffusion map can be truncated to d dimensions,

$$\mathbf{\Phi}_t^{(d)}(i) = \begin{pmatrix} \lambda_2^t \varphi_{2,i} \\ \vdots \\ \lambda_{d+1}^t \varphi_{d+1,i} \end{pmatrix}. \tag{5.3}$$

Algorithm 2 Spectral Clustering Algorithm

1: **procedure** SPECTRAL CLUSTERING
2: Given a graph $G = (V, E, \mathbf{W})$, k denotes number of clusters, and time t
3: compute the $(k-1)$dimension diffusion map

$$\mathbf{\Phi}_t^{(k-1)}(i) = \begin{pmatrix} \lambda_2^t \varphi_{2,i} \\ \vdots \\ \lambda_k^t \varphi_{k,i} \end{pmatrix}$$

4: cluster $\mathbf{\Phi}_t^{(k-1)}(1), \ldots, \mathbf{\Phi}_t^{(k-1)}(n) \in \mathbb{R}^{k-1}$ using, eg, k-means clustering

The aim of spectral clustering is to cluster vertices of the graph into k clusters, as summarized in algorithm 2.
Particularly for two the case of clusters C and C^c we have that

$$\mathbf{\Phi}_t^{(1)} \in \mathbb{R}^1, i = 1, \ldots, n$$

will be on a line ($\mathbf{\Phi}_t^{(1)}$ is 1-dimensional, i.e., a scalar). Then, a natural way of clustering on a line is to define a threshold q such that $\mathbf{v}_i \in C$ if $\mathbf{\Phi}_t^{(1)}(i) \leqslant q$. For example, for 5 points we would get 1D clustering problem as the one shown in Figure 5.2.

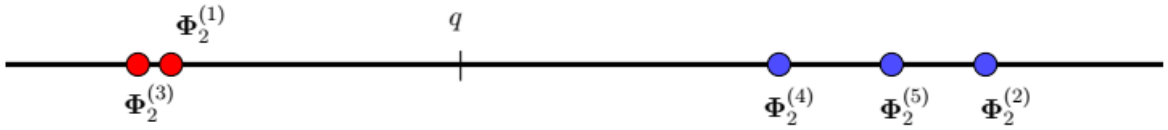

Figure 5.2: Example of spectral clustering with 2 classes and 5 points.

Hierarchical Clustering

Another type of unsupervised clustering algorithms are the hierarchical clustering methods. Hierarchical clustering algorithms are mainly divided into agglomerative (bottom up) and divisive (top down) clustering. Agglomerative methods assign a cluster to each observation and then reduce the number of clusters by iteratively merging smaller clusters into larger ones. On the other hand, divisive methods start with one large cluster containing all the observations and iteratively divide it into smaller clusters.
Since the principles behind agglomerative and divisive clustering are quite analogous, in this lecture we only explain agglomerative algorithms. To this end, given n objects $\mathbf{v}_1, \ldots, \mathbf{v}_n$ and pairwise dissimilarities $\delta_{i,j} = \delta_{j,i}, \mathbf{\Delta} = (\delta_{i,j})$ $i, j = 1, \ldots, n$, a linkage

function between two clusters C_1, C_2 is defined as

$$d(C_1, C_2) = \begin{cases} \min_{i \in C_1, j \in C_2} \delta_{i,j} & \textbf{single linkage} \\ \max_{i \in C_1, j \in C_2} \delta_{i,j} & \textbf{complete linkage} \\ \frac{1}{|C_1||C_2|} \sum_{i \in C_1, j \in C_2} \delta_{i,j} & \textbf{average linkage} \end{cases}. \tag{5.4}$$

Then, as summarized in algorithm 3, an agglomerative clustering algorithm builds larger clusters by merging similar clusters as we move up the hierarchy (see Figure 5.3). Note that algorithm 3 merges clusters until we are left with a single cluster containing all observations. In practice, we can stop iterating once the desired amount of clusters is reached.

Algorithm 3 Agglomerative Clustering

1: **procedure** AGGLOMERATIVE CLUSTERING
2: Initialize clusters as singletons: **for** $i = 1, \ldots, n$ **do** $C_i \leftarrow i$
3: Initialize the set of clusters available for merging as $S \leftarrow \{1, \ldots, n\}$
4: **repeat**
5: Pick the two most similar clusters to merge: $(j, k) \leftarrow \arg\min_{j,k \in S} d(C_j, C_k)$
6: Merge C_k into C_j as $C_j \leftarrow C_j \bigcup C_k$
7: Mark k as unavailable, $S \leftarrow S - \{k\}$
8: If $C_j = \{1, \ldots, n\}$, then mark j as unavailable, $S \leftarrow S - \{j\}$.
9: **for** each $i \in S$ **do** update dissimilarities $d(C_i, C_j)$
10: **until** no more clusters are available for merging

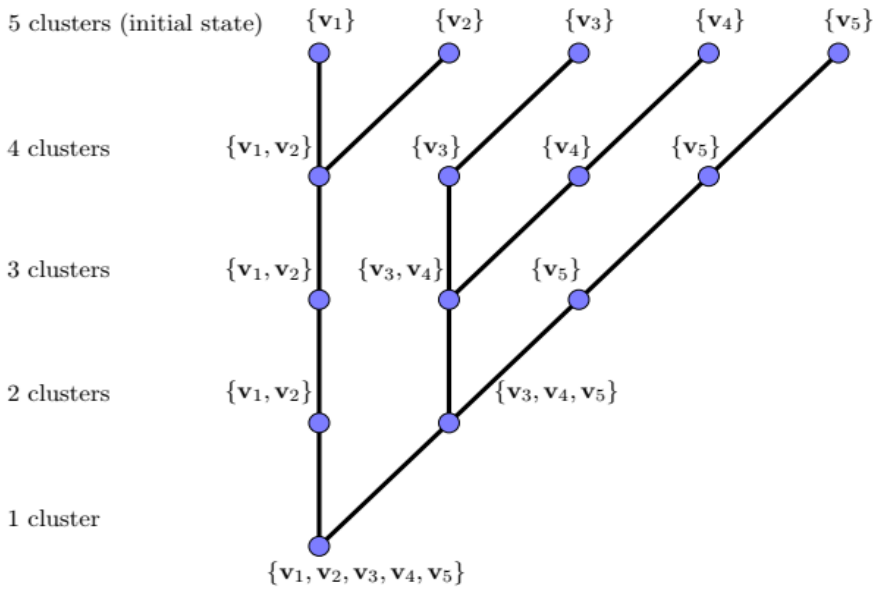

Figure 5.3: Graphical example of agglomerative clustering with 5 points ($n = 5$).