CHAPTER FOUR

UNSUPERVISED CLASSIFICATION

4.1 OVERVIEW

In this chapter a brief overview is given of the notion of grouping objects into different categories without any supervision. The previous chapter described a supervised approach to grouping objects and how the relationship between the desired class membership and input vectors was derived using labels. The possibility is now explored of grouping objects based on their perceived intrinsic similarities. A formal definition is provided on an unsupervised method known as clustering, followed by the advantages of exploring an unsupervised approach. The design considerations behind producing good clustering results are then explored, followed by the challenges inherent when using clustering methods to solve real world problems.

Clustering algorithms are broadly divided into hierarchical and partitional clustering approaches [40, 170]. Four popular hierarchical clustering methods and two partitional clustering methods are discussed with their corresponding properties. The chapter concludes with a discussion on how clusters can be converted to classes to obtain a supervised classifier.

4.2 CLUSTERING

Clustering is a form of conceptual clustering, which is an unsupervised method used for grouping unlabelled input vectors into a set of categories. Clustering groups a set of input vectors through perceived intrinsically similar or dissimilar characteristics.

Let $\{y^k\}$, $y^k \in \mathbb{N}$, $1 \leq y^k \leq K$, denotes the set of cluster labels. Let $\mathcal{F}_\mathcal{C} : \mathbb{R}^n \to \{y^k\}$ denote the function that maps the input vector $\vec{\tilde{x}}^p$, $\vec{\tilde{x}}^p \in \mathbb{R}^n$, to a cluster label. The variable p denotes the index of the vector within the input vector set. The function $\mathcal{F}_\mathcal{C}$ is said to cluster the input vector set $\{\vec{\tilde{x}}^p\}$ into K clusters.

Several motivations exist to justify the use of clustering algorithms for many non-synthetic data sets:

1. Significant costs are involved when gathering information about the data set to create reliable class labels for supervised classification.

2. The underlying data structure of a large unlabelled data set can be captured to provide reliable clustering on a smaller labelled data set.

3. Accommodate a dynamic input space. This is when the input space changes over time or in response to a triggered event.

4. Assisting in creating a well-conditioned input vector from the input space to gain insight into what improves the cluster label allocation.

4.2.1 Mapping of vectors to clusters

A cluster label is derived by evaluating several different input data sources from the input space. These data sources are grouped together to form an input vector $\vec{\tilde{x}}$. These input vectors are the same as with the supervised classifier and have descriptive forms that can be interpreted. The preprocessing and postprocessing of the input and output vectors is an optional procedure used to improve the clustering algorithm's performance [136]. Using feature vectors $\vec{x}$ and postprocessed output value y is assumed to improve the performance significantly and is used throughout this chapter.

The clustering algorithm constructs a function $\mathcal{F}_\mathcal{C}$ to determine the cluster label and is based on the set of feature vectors $\{\vec{x}^p\}$. The mapping function is expressed as

$$y^k = \mathcal{F}_\mathcal{C}(\vec{x}^p). \tag{4.1}$$

The clusters typically encapsulate properties of the non-synthetic data set; each cluster should have a homogeneous set of feature vectors.

4.2.2 Creating meaningful clusters

No theoretical guideline exists on how to extract the optimal feature vector set from the input vector set for a specific clustering application. Owing to the limited intrinsic information within the feature vector set, it is difficult to design a clustering algorithm that will find clusters to match the desired cluster labels.

This constraint is created by a clustering algorithm, as it tends to find clusters in the feature space irrespective of whether any real clusters exist. This constraint motivates the notion that any two

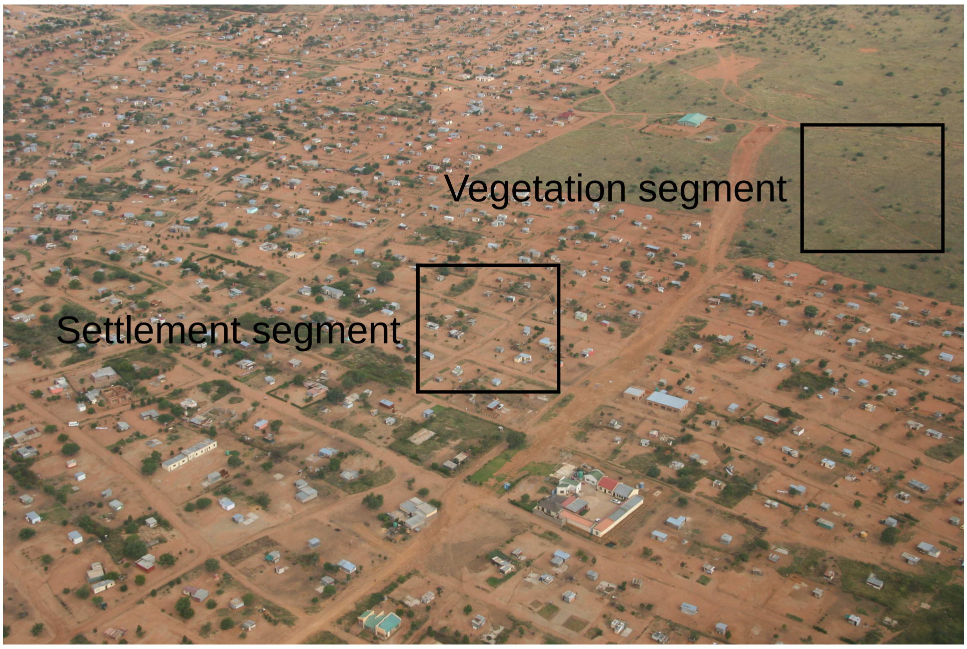

FIGURE 4.1: An aerial photo taken in the Limpopo province, South Africa of two different land cover which are indicated by a natural vegetation segment and settlement segment. A segment is defined as a collection of pixels within a predefined size bounding box.

arbitrary patterns can be made to appear equally similar when evaluating a large number of dimensions of information in the feature space. This will result in defining a meaningless clustering function $\mathcal{F}_{\mathcal{C}}$. This makes clustering a subjective task in nature, which can be modified to fit any particular application.

The advantage in this versatility is that the clustering algorithm can be used as either an exploratory or a confirmatory analysis tool [170]. Clustering used as an exploratory analysis tool is there to explore the underlying structures of the data. No predefined models or hypotheses are needed when exploring the data set. Clustering used as a confirmatory analysis tool is to confirm any set of hypotheses or assumptions. In certain applications, clustering is used as both; first to explore the underlying structures to form new hypotheses. Second, to test these hypotheses by clustering the feature vector set. This makes clustering a data-driven learning algorithm and any domain knowledge that is available can improve the forming of clusters [170].

Domain knowledge is used to reduce complexity by aiding in processes such as feature selection and feature extraction. Proper domain knowledge leads to good feature vector representation that will yield exceptional performance with the most common clustering algorithms, while incomplete domain knowledge leads to poor feature vector representation that will only yield acceptable performance with a complex clustering algorithm.

An aerial photo is used to illustrate the clustering of different land cover types in figure 4.1. In this

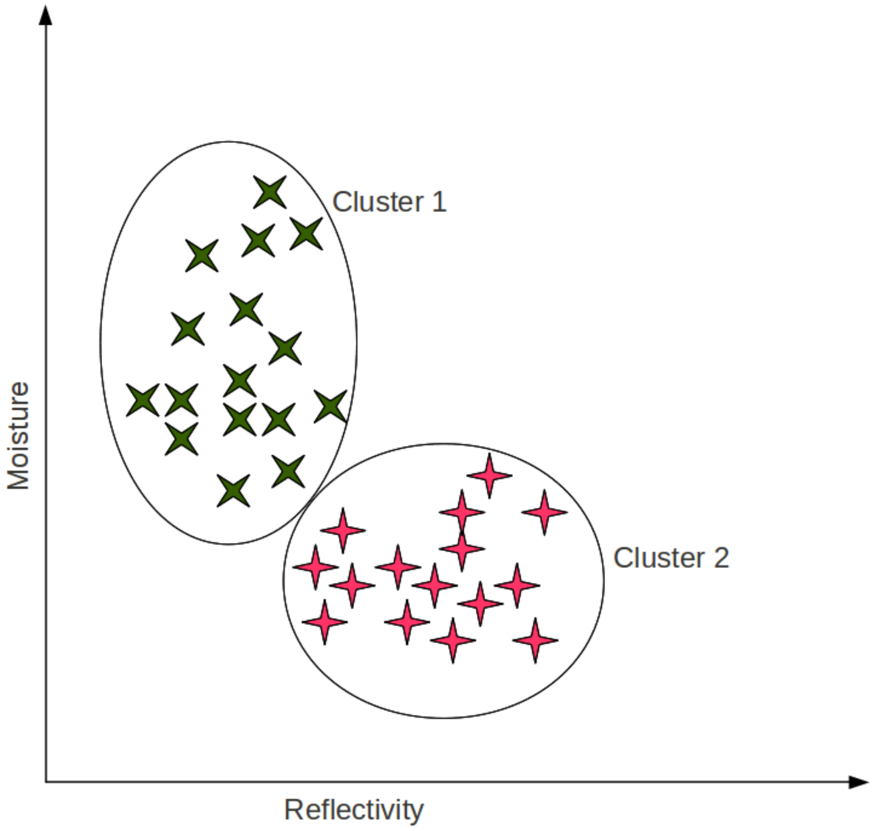

FIGURE 4.2: A two-dimensional illustration of feature vectors within the feature space. The green feature vectors represent the natural vegetation class and the red feature vectors represent the human settlement class.

image two land cover types are of interest: natural vegetation and human settlement.

Land cover example: In the case of the land cover example shown in figure 4.1, domain knowledge is used for feature extraction and selection. Let it be assumed that the domain knowledge provided information that the feature vector given in equation (4.2) will provide better separability between the two categories.

$$\vec{x} = [(\text{Moisture})\ (\text{Reflectivity})]. \tag{4.2}$$

The natural vegetation segments have feature vectors with low reflectivity and high moisture levels, while the human settlement segments have feature vectors with high reflectivity and low moisture levels. This is illustrated in a two-dimensional plot shown in figure 4.2. When natural clusters exist in the feature space and the number of clusters is set to K=2, a well-designed clustering algorithm will produce two perfect clusters, as shown in figure 4.2. □

Domain knowledge in many fields is incomplete or unavailable. Verifying the domain knowledge from actual (non-synthetic) data sets is extremely resource-expensive and is difficult to relate to the feature space. The most practical approach for designing an unsupervised learning algorithm is to *learn from example* [171]. The *learning from example* approach requires that the clustering algorithm be subjected to an external evaluation process. The external evaluation is hampered by the fact that thousands of different clustering algorithms have been developed and evidence suggests that none of them is superior to any other [172]. This is addressed in the *impossibility theorem*, which states three criteria which no clustering algorithm can satisfy [172]. The three criteria to satisfy in the *impossibility theorem* are:

1. Scale invariance; the scaling of the feature vectors should not change the assigned cluster labels.
2. Richness; the clustering algorithm must be able to achieve all possible partitions in the feature space.
3. Consistency; the change in distance within all clusters will not change the assigned cluster labels.

Based on the *impossibility theorem*, each clustering application is different and requires an unique design to obtain good clustering results. This emphasises the importance of obtaining *acceptable performance* in the search for a clustering algorithm, as it is infeasible to search through all the permutations of clustering designs. The admissibility criterion is a more practical approach to consider when applying external evaluation to a clustering algorithm [170]. The admissibility criterion comprises three important design considerations:

1. The manner in which the clusters are formed.
2. The intrinsic structure of the feature vectors.
3. The sensitivity of the clusters created.

4.2.3 Challenges of clustering

Humans cluster with ease in two and three dimensions, while a machine learning method is required to cluster in higher dimensions. Several design implications arise when clustering in higher dimensions [171]:

- Determining the number of clusters K (section 4.6).
- Determining whether the feature vectors carry representative information to produce clusters that will hold a relation to the desired classes for the application (section 4.2.2).

- Deciding which pairwise similarity metric should be used to evaluate the feature space (section 4.3).

- Determining how the feature vectors should be evaluated to form clusters. Clustering algorithms are broadly divided into hierarchical and partitional clustering approaches [40, 170]. The first approach is hierarchical clustering, which produces a nested hierarchy of clusters of discrete groups (section 4.4). The second approach is partitional clustering, which creates an unnested partitioning of the data points with K clusters [173] (section 4.5).

4.3 SIMILARITY METRIC

A clustering algorithm defines clusters with feature vectors that are similar to one another, and separate them from feature vectors that are dissimilar. This similarity between feature vectors is usually measured using a distance function.

Let $\{\vec{x}\}$, $\vec{x} \in \mathbb{R}^N$ denote a set of N-dimensional feature vectors. Let $D : \mathbb{R}^N \to \mathbb{R}_+$ denote the distance function that calculates the distance between the vector $\vec{x}^p$ and $\vec{x}^q$. The function D is said to return the distance (similarity metric) between the two feature vectors.

The properties of the distance function D are:

- Non-negative, $D(\vec{x}^p, \vec{x}^q) \geq 0$.

- Identity axiom, $D(\vec{x}^p, \vec{x}^q) = 0$, iff $p = q$.

- Triangle inequality, $D(\vec{x}^o, \vec{x}^p) + D(\vec{x}^p, \vec{x}^q) \geq D(\vec{x}^o, \vec{x}^q)$.

- Symmetry axiom, $D(\vec{x}^p, \vec{x}^q) = D(\vec{x}^q, \vec{x}^p)$.

The non-negative and identity axioms produce a positive definite function. The distance metric is as important in the design as the clustering algorithm itself. Proper selection of a distance metric will result in the distance between feature vectors of the same cluster being smaller than the distance between the feature vectors of other clusters.

Choosing a distance function opens a broad class of distance metrics. The first to consider is the general Minkowski distance, which is used to derive some of the most common distance functions used in clustering applications. The Minkowski distance D_{mink} is expressed as

$$D_{\text{mink}}(\vec{x}^p, \vec{x}^q) = \left(\sum_{n=1}^{N} |x_n^p - x_n^q|^m \right)^{\frac{1}{m}}. \tag{4.3}$$

The variable $m, m \in \mathbb{N}$, is the Minkowski parameter that is used to adjust the nature of the distance metric. The Minkowski distance simplifies to the popular Euclidean distance D_{ed} if the Minkowski parameter m is set to 2 in equation (4.3). The Euclidean distance is computed as

$$D_{\text{ed}}(\vec{x}^p, \vec{x}^q) = \sqrt{\sum_{n=1}^{N} |x_n^p - x_n^q|^2}. \tag{4.4}$$

The advantage of the Euclidean distance is that it is invariant to translation or rotation of the feature vector $\vec{x}$. The Euclidean distance however does vary under an arbitrary linear transformation.

The squared Euclidean distance is an alteration to the Euclidean distance, as it places a greater weight on a set of vectors that are considered to be outliers in the vector space. The squared Euclidean distance is expressed as

$$D_{\text{sq}}(\vec{x}^p, \vec{x}^q) = \sum_{n=1}^{N} |x_n^p - x_n^q|^2. \tag{4.5}$$

If the Minkowski parameter is set to m=1, equation (4.3) simplifies to the Manhattan distance. The Manhattan distance is the sum of the absolute difference between vectors. The Manhattan distance is expressed as

$$D_{\text{man}}(\vec{x}^p, \vec{x}^q) = \sum_{n=1}^{N} |x_n^p - x_n^q|. \tag{4.6}$$

The Mahalanobis distance metric is used in statistics to measure the correlations between multivariante vectors. The Mahalanobis distance metric D_{mahal} is expressed as

$$D_{\text{mahal}}(\vec{x}^p, \vec{x}^q) = \sqrt{(\vec{x}^p - \vec{x}^q) G_{\text{mahal}}^{-1} (\vec{x}^p - \vec{x}^q)}, \tag{4.7}$$

where G_{mahal} denotes the covariance matrix.

4.4 HIERARCHICAL CLUSTERING ALGORITHMS

A clustering algorithm uses a set of feature vectors $\{\vec{x}^p\}$, cluster parameters and a similarity metric to construct a mapping function $\mathcal{F}_{\mathcal{C}}$. Let $\vartheta = (\cup_{q=1}^{Q} \vartheta^q)$ denote the set of cluster parameters that the clustering algorithm needs to determine when constructing $\mathcal{F}_{\mathcal{C}}$.

As stated previously, clustering algorithms are broadly divided into either a hierarchical or partitional clustering approach [40, 170]. The hierarchical clustering approach produces a nested hierarchy of clusters of discrete groups according to a certain linkage criterion. The nested clusters are recursively linked in either an agglomerative mode or divisive mode. The second approach to clustering is partitional clustering, which creates an unnested partitioning of the vectors into K clusters [173]. In hierarchical clustering using an agglomerative mode, the clustering parameter set $\{\vartheta\}$ is determined iteratively in four steps:

Step 1: The clustering algorithm starts by allocating each feature vector to its own cluster. The initialisation phase is defined as

$$\vartheta_I^p = \vec{x}^p, \qquad \forall p \text{ and } I = 0. \tag{4.8}$$

The variable ϑ_I^p denotes the p^{th} set of cluster parameters at epoch I, with I set to zero for the initialisation phase. The vector $\vec{x}^p$ denotes the p^{th} feature vector.

Step 2: The similarity between two clusters is defined by a linkage criterion. The linkage criterion evaluates two clusters using a similarity metric (section 4.3) to compute the dendrogrammatic distance $T(\vartheta_I^l, \vartheta_I^k)$. The dendrogrammatic distance is computed as

$$T(\vartheta_I^l, \vartheta_I^k) = \beta(\vartheta_I^l, \vartheta_I^k), \tag{4.9}$$

where the linkage criterion is denoted by the function β, $\beta \in \{T_{\text{sing}}, T_{\text{com}}, T_{\text{ave}}, T_{\text{ward}}\}$.

This expression states that all the feature vectors in cluster y^l must be compared to all the feature vectors in cluster y^k using a predefined argument. The linkage criterion's function β returns a dendrogrammatic distance between the two clusters.

Step 3: Select the shortest dendrogrammatic distance $T(\vartheta_I^l, \vartheta_I^k)$ between all pairs of clusters. Let $\vartheta_I^{l^*}$ and $\vartheta_I^{k^*}$ be selected such that

$$[\vartheta_I^{l^*}, \vartheta_I^{k^*}] = \underset{l,k \in [1,K]; l \neq k}{\text{argmin}} \; T(\vartheta_I^l, \vartheta_I^k). \tag{4.10}$$

Step 4: Merge the two clusters with index l^* and k^* as

$$\vartheta_{(I+1)}^{l^*} = \left(\vartheta_I^{l^*} \cup \vartheta_I^{k^*}\right), \tag{4.11}$$

$$\vartheta_{(I+1)}^{k^*} = \emptyset. \tag{4.12}$$

Steps 2–4 are repeated until all the clusters are merged into a single cluster. The sequence of merging clusters can be graphically presented by a tree diagram, called a dendrogram. The dendrogram is a multi-level hierarchy with two clusters merging at each level.

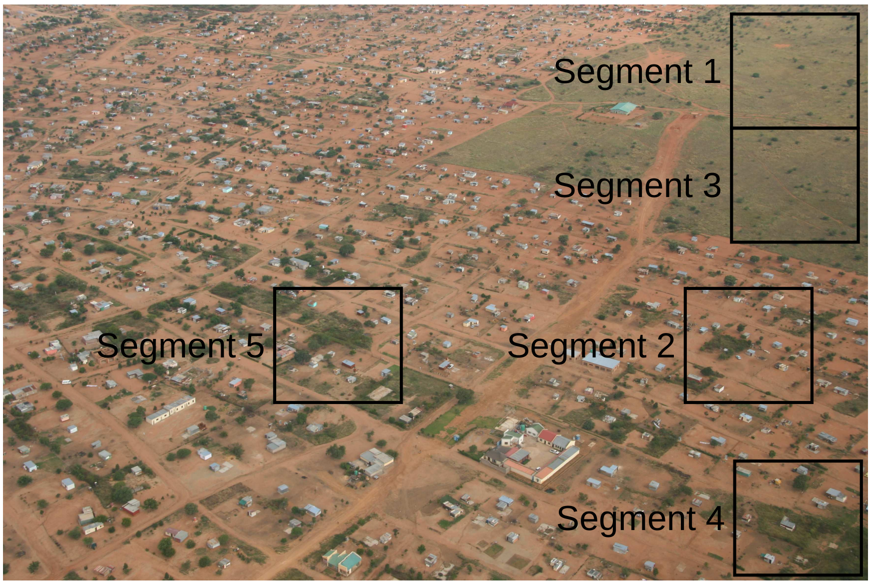

FIGURE 4.3: An alternative selection of five new segments of the aerial photo taken in the Limpopo province which indicates different types of land cover types.

Land cover example: Five new segments are defined in figure 4.3. A hierarchical clustering algorithm operating in agglomerative mode creates a dendrogram shown in figure 4.4 when applied to the five segments. In the first iteration the similarity between segment 4 and segment 5 is the highest (shortest dendrogrammatic distance). These segments are merged to form a new cluster. The dendrogrammatic distances between the merging clusters are indicated on the vertical axis. The shorter the distance on the vertical axis, the more similar the two joining clusters. In the second iteration, segment 1 and segment 3 are joined as being the next most similar clusters. These two newly formed clusters are joined together, as they are more similar to each other than to segment 2. Segment 2 is joined to form a single cluster containing all segments, which completes the dendrogram.

In the divisive mode, the clustering algorithm starts by placing the entire feature vector set in a single cluster. In this mode, a comparison is made between all the feature vectors within the cluster to determine which feature vectors are the most dissimilar and split the cluster into two separate clusters. This process is repeated until every single cluster retains a single feature vector. The sequence of separating the clusters is also represented on a dendrogram. Only the agglomerative mode was considered, as it is a bottom-up approach and the concept could easily be derived for a divisive mode with the same methodology in a top-down approach.

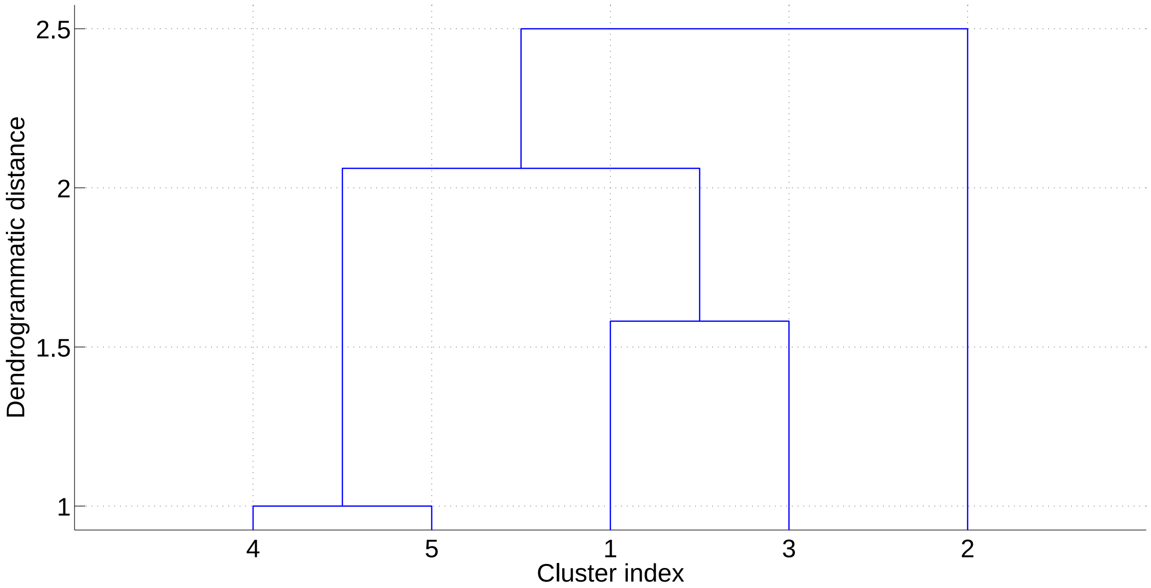

FIGURE 4.4: An illustration of an hierarchical clustering approach operating in agglomerative mode.

4.4.1 Linkage criteria

4.4.1.1 Single linkage criterion

The merging of clusters is based on the dendrogrammatic distance between clusters. The dendrogrammatic distance is computed using a linkage criterion. The single linkage criterion is the first linkage criterion that is considered, as it searches for the shortest distance between two feature vectors; each residing in two different clusters. The single linkage criterion $T_{\text{sing}}(\vartheta_I^l, \vartheta_I^k)$ is expressed as

$$T_{\text{sing}}(\vartheta_I^l, \vartheta_I^k) = \min\{D(\vec{x}^p, \vec{x}^q)\} \quad \forall \vec{x}^p \in \vartheta_I^l,\ \vec{x}^q \in \vartheta_I^k \text{ and } l \neq k. \tag{4.13}$$

The variable $\vec{x}^p$ denotes the p^{th} feature vector and $\vec{x}^q$ denotes the q^{th} feature vector. The similarity metrics shown in section 4.3 (equation (4.3)–(4.7)) or any other distance metric found in the literature can be used as the distance metric $D(\vec{x}^p, \vec{x}^q)$. The single linkage criterion has a chaining effect as a characteristic trait when forming clusters. This results in clusters that are straggly and elongated in shape [174]. The advantage of elongated clusters is that they can extract spherical clusters from the feature space.

4.4.1.2 Complete linkage criterion

The complete linkage criterion computes a dendrogrammatic distance by finding the maximum possible distance between two feature vectors that reside in different clusters. The complete linkage criterion $T_{\text{com}}(\vartheta_I^l, \vartheta_I^k)$ is expressed as

$$T_{\text{com}}(\vartheta_I^l, \vartheta_I^k) = \max\{D(\vec{x}^p, \vec{x}^q)\} \quad \forall \vec{x}^p \in \vartheta_I^l,\ \vec{x}^q \in \vartheta_I^k \text{ and } l \neq k. \tag{4.14}$$

The variable $\vec{x}^p$ denotes the p^{th} feature vector and $\vec{x}^q$ denotes the q^{th} feature vector. The complete linkage criterion has the characteristic trait of forming tightly bounded compact clusters. The complete linkage criterion creates more useful clusters in many actual (non-synthetic) data sets than the single linkage criterion [170, 175].

4.4.1.3 Average linkage criterion

The average linkage criterion is the most intuitive linkage criterion, as it calculates a dendrogrammatic distance between two clusters by finding the average distance among all pairs of feature vectors residing in different clusters. The average linkage criterion $T_{\text{ave}}(\vartheta_I^l, \vartheta_I^k)$ is expressed as

$$T_{\text{ave}}(\vartheta_I^l, \vartheta_I^k) = \frac{1}{|\vartheta_I^l||\vartheta_I^k|} \sum_{\vec{x}^p \in \vartheta_I^l} \sum_{\vec{x}^q \in \vartheta_I^k} D(\vec{x}^p, \vec{x}^q), \quad l \neq k. \tag{4.15}$$

$|\vartheta_I^l|$ denotes the number of feature vectors in cluster ϑ_I^l and $|\vartheta_I^k|$ denotes the number of feature vectors in cluster ϑ_I^k. The average linkage criterion is a compromise between the complete linkage criterion's sensitivity to outliers and the chaining effect produced by the single linkage criterion.

4.4.1.4 Ward criterion

The Ward criterion computes a dendrogrammatic distance between clusters by finding the clusters that will maximise the coefficient of determination R^2 [176]. The Ward criterion $T_{\text{ward}}(\vartheta_I^l, \vartheta_I^k)$ is expressed as

$$\begin{aligned} T_{\text{ward}}(\vartheta_I^l, \vartheta_I^k) \quad = \quad & \sum_{p \in \left(\vartheta_I^l \cup \vartheta_I^k\right)} \left\| \vec{x}^p - E\left[\vartheta_I^l \cup \vartheta_I^k\right] \right\|^2 - \sum_{p \in \vartheta_I^l} \left\| \vec{x}^p - E\left[\vartheta_I^l\right] \right\|^2 - \\ & \sum_{p \in \vartheta_I^k} \left\| \vec{x}^p - E\left[\vartheta_I^k\right] \right\|^2. \end{aligned} \tag{4.16}$$

The expected value of the feature vectors in the cluster is denoted by $E[\vec{x}^p]$. The Ward criterion attempts to minimise the variance between the K clusters and only uses the Euclidean distance. Most linkage criteria in the literature are variants of the single linkage, complete linkage, average linkage or Ward criterion.

4.4.2 Cophenetic correlation coefficient

A dendrogram is created iteratively as the function $\mathcal{F}_\mathcal{C}$ is derived with a hierarchical clustering algorithm. The dendrogram illustrates the dendrogrammatic distances obtained with the linkage criterion (section 4.4.1). The cophenetic correlation coefficient is a statistical measure of correlation between the dendrogrammatic distances and the similarity distances for all pairs of feature vectors [177]. The cophenetic correlation coefficient is computed as

$$D_{\mathrm{cc}} = \frac{\sum_{q=2}^{P}\sum_{p=1}^{q}(D(\vec{x}^{p},\vec{x}^{q}) - E[D(\vec{x}^{p},\vec{x}^{q})])(T(\vartheta_0^l,\vartheta_0^k) - E[T(\vartheta_0^l,\vartheta_0^k)])}{\sqrt{\sum_{q=2}^{P}\sum_{p=1}^{q}(D(\vec{x}^{p},\vec{x}^{q}) - E[D(\vec{x}^{p},\vec{x}^{q})])^2(T(\vartheta_0^l,\vartheta_0^k) - E[T(\vartheta_0^l,\vartheta_0^k)])^2}}, \tag{4.17}$$

with $\vec{x}^{p} \in \vartheta_0^l$ and $\vec{x}^{q} \in \vartheta_0^k$. The function $D(\vec{x}^{p},\vec{x}^{q})$ denotes the distance between the feature vector $\vec{x}^{p}$ and $\vec{x}^{q}$ as shown in section 4.3. The $T(\vartheta_0^l,\vartheta_0^k)$, $\vec{x}^{p} \in \vartheta_0^l$, $\vec{x}^{q} \in \vartheta_0^k$, denotes the dendrogrammatic distance between the feature vector $\vec{x}^{p}$ and $\vec{x}^{q}$ as shown in equation (4.9). The higher the correlation, the better the dendrogram preserves the information of the feature space when using a particular linkage criterion. The cophenetic correlation coefficient is used to evaluate several different distance metrics and linkage criteria that will best retain the original distances of the feature space in the dendrogram [177].

4.5 PARTITIONAL CLUSTERING ALGORITHMS

A partitional clustering algorithm operates on the actual feature vectors, which significantly reduces the required space and computations to operate, which makes it more suitable for larger data sets when compared to hierarchical clustering [173].

Let $\{y^k\}$, $k \in \mathbb{N}$, $1 \leq k \leq K$ denote the set of cluster labels. Let $\mathcal{F}_\mathcal{C} : \mathbb{R}^N \rightarrow \{y^k\}$ denote the function that maps feature vectors $\{\vec{x}\}$, $\{\vec{x}\} \in \mathbb{R}^N$, onto the clusters. Then $\mathcal{F}_\mathcal{C}$ is said to cluster $\vec{x}$ into K clusters.

In a general case of partitional clustering, a set of clustering parameters is determined when constructing the mapping function $\mathcal{F}_\mathcal{C}$. Let $\{\vartheta_I^k\}$, $\{\vartheta_I^k\} \in \Omega_\vartheta$, denote the set of clustering parameters. The variable k, $1 \leq k \leq K$, denotes the index in the set $\{\vartheta_I^k\}$ which refers to the cluster label y^k. The variable I denotes the current epoch. The partitional clustering algorithm uses a distance metric $D(\vec{x}^{\mathrm{p}},\vartheta_I^k)$ to measure the distance between the p^{th} feature vector $\vec{x}^{\mathrm{p}}$ and cluster y^k. The feature vector $\vec{x}^{\mathrm{p}}$ is then mapped onto $\{y^k\}$ using the function $\mathcal{F}_\mathcal{C}$, such that

$$\mathcal{F}_{\mathcal{C}}(\vec{x}^{\,\mathrm{p}}) = \underset{y^k \in \{y^k\}}{\operatorname{argmin}} \left\{ D(\vec{x}^{\,\mathrm{p}}, \vartheta_I^k) \right\}. \tag{4.18}$$

Intuitively, the function $\mathcal{F}_{\mathcal{C}}$ maps a vector $\vec{x}^{\,\mathrm{p}}$ to the nearest cluster.

The function $\mathcal{F}_{\mathcal{C}}$ is constructed by determining the set of cluster parameters $\{\vartheta_I^k\}$ to minimise the overall distance between a given set of feature vectors $\{\vec{x}\}$ and the K corresponding clusters. One possible definition of this process is

$$\{\vartheta_I^{k*}\} = \underset{\{\vartheta_I^k\} \in \Omega_\vartheta}{\operatorname{argmin}} \left\{ \sum_{p=1}^{P} D\big(\vec{x}^{\,\mathrm{p}}, \vartheta_I^{\mathcal{F}_{\mathcal{C}}(\vec{x}^{\,\mathrm{p}})}\big) \right\}. \tag{4.19}$$

The clustering algorithm simultaneously determines the parameters ϑ_I^k of each cluster, as well as the cluster assignment of each feature vector $\vec{x}^{\,\mathrm{p}}$.

4.5.1 K-means algorithm

The first partitional clustering algorithm explored is the popular K-means algorithm [178]. The K-means algorithm attempts to find the center points of the natural clusters. The K-means clustering algorithm accomplishes this by partitioning the feature vectors into K mutually exclusive clusters.

K-means is a heuristic, hill-climbing algorithm that attempts to converge to the center mass point of the natural clusters. It can be viewed as a gradient descent approach which attempts to minimise the sum of squared error of each feature vector to the nearest cluster centroid [179]. The clusters created with the K-means algorithm are compact and isolated in nature.

Minimising the SSE has been shown to be a NP-hard problem, even for a two-cluster problem [180]. This gives rise to a variety of heuristic approaches to solving the problem for practical applications. The most common method of implementing the K-means algorithm is the Lloyd's approach. The Lloyd's approach is an iterative method which comprises three steps:

Step 1: Initialise a set of K centroids $\{\vartheta_I^k\}$.

Step 2: Assign each feature vector to its closest centroid. This is accomplished by creating K empty sets $\vec{s}^{\,k} = \emptyset, k = 1, 2, \ldots, K$, for each of the corresponding centroids $\{\vartheta_I^k\}$. The assignment step is expressed as

$$\vec{s}^{\,k} = \left\{ \{\vec{x}^{\,p}\} : D(\vec{x}^{\,p}, \vartheta_I^k) < D(\vec{x}^{\,p}, \vartheta_I^l), \forall l \neq k \right\}. \tag{4.20}$$

The vector $\vec{x}^{\,p}$ denotes the p^{th} feature vector and D denotes the distance function.

Step 3: The update step adjusts the centroids' position to minimise the sum of distance given in equation (4.19). The adjustment is made for each centroid as

$$\vartheta^k_{(I+1)} = \frac{1}{|\vec{s}^k|} \sum_{\vec{x}^p \in \vec{s}^k} \vec{x}^p, \qquad \forall\, k. \tag{4.21}$$

$|\vec{s}^k|$ denotes the number of elements in the set.

Steps 2–3 are repeated until all the feature vectors within each cluster remain unchanged or a predefined stopping criterion is reached.

The performance of the K-means algorithm is dependent on the density distribution of the feature vectors in the feature space. K-means will minimise the SSE with high probability to the global minimum if the feature vectors are well separated [181]. The ability of the K-means algorithm to handle a large number of feature vectors enables the parallel execution of multiple replications with different initial seeds to avoid local minima. The K-means clustering algorithm is usually used as a benchmark against other algorithms, and has been used successfully in many other fields [171].

4.5.2 Expectation-maximisation algorithm

The Expectation-Maximisation (EM) algorithm is another partitional clustering algorithm, which attempts to fit a mixture of probability distributions on the set of feature vectors [182]. The EM algorithm was designed on the assumption that the feature vectors are extracted from a feature space with a multi-modal distribution.

Given a set of observable vectors $\{\vec{x}\}$ and unknown variables $\{y^k\}$, the EM algorithm finds the maximum likelihood or maximum *aposterior* estimates for the parameters $\vec{\omega}$, $\vec{\omega} \in \Omega$. The maximum likelihood estimation of the parameters $\vec{\omega}_{\mathrm{ML}}$ is expressed as

$$\vec{\omega}_{\mathrm{ML}} = \underset{\vec{\omega}\in\Omega}{\operatorname{argmax}} \left\{ \log p(\vec{x}|\vec{\omega}) \right\} = \underset{\vec{\omega}\in\Omega}{\operatorname{argmax}} \left\{ \mathcal{J}(\vec{\omega}) \right\}. \tag{4.22}$$

The log-likelihood of the conditional probability in equation (4.22) is expanded to incorporate the unknown variables y^k as

$$\mathcal{J}(\vec{\omega}) = \log p(\vec{x}|\vec{\omega}) = \log \sum_k p(\vec{x}, y^k|\vec{\omega}) = \log \sum_k q(y^k|\vec{x},\vec{\omega}) \frac{p(\vec{x}, y^k|\vec{\omega})}{q(y^k|\vec{x},\vec{\omega})}. \tag{4.23}$$

The function $q(y^k|\vec{x},\vec{\omega})$ is an arbitrary density over y^k. Considering the following lower bound inequality to equation (4.23) as

$$\log \sum_k q(y^k|\vec{x},\vec{\omega}) \frac{p(\vec{x}, y^k|\vec{\omega})}{q(y^k|\vec{x},\vec{\omega})} \geq \sum_k q(y^k|\vec{x},\vec{\omega}) \log \frac{p(\vec{x}, y^k|\vec{\omega})}{q(y^k|\vec{x},\vec{\omega})}, \tag{4.24}$$

which for convenience is rewritten as

$$\mathcal{J}(\vec{\omega}) \geq \sum_k q(y^k|\vec{x},\vec{\omega}) \log \frac{p(\vec{x},y^k|\vec{\omega})}{q(y^k|\vec{x},\vec{\omega})}. \tag{4.25}$$

It is easier if the EM algorithm instead attempts to maximise the lower bound shown in equation (4.25). The EM algorithm iteratively adjusts the parameters of the distributions in two steps. The first step is the expectation step (E-step) which calculates the log likelihood function, with respect to the conditional distribution of y^k given $\vec{x}$ with the current estimate of the parameter $\vec{\omega}$ as

$$q(y^k|\vec{x},\vec{\omega})^{\text{new}} = \underset{q(y^k|\vec{x},\vec{\omega})}{\operatorname{argmax}} \left\{ \sum_k q(y^k|\vec{x},\vec{\omega}) \log \frac{p(\vec{x},y^k|\vec{\omega})}{q(y^k|\vec{x},\vec{\omega})} \right\}. \tag{4.26}$$

Calculating the E-step requires the vector $\vec{\omega}$ to be fixed while attempting to optimise over the space of distributions. The second step is the maximisation step (M-step), which tries to maximise the vector $\vec{\omega}$ using the result from equation (4.26). The M-step is computed as

$$\vec{\omega}^{\text{new}} = \underset{\vec{\omega}}{\operatorname{argmax}} \left\{ \sum_k q(y^k|\vec{x},\vec{\omega})^{\text{new}} \log \frac{p(\vec{x},y^k|\vec{\omega})}{q(y^k|\vec{x},\vec{\omega})^{\text{new}}} \right\}. \tag{4.27}$$

The EM algorithm iterates through both steps until it converges to a local maximum. The feature vector is assigned to a cluster that maximises the *aposterior* probabilities of a given distribution.

The disadvantage of the EM algorithm is that even though the probability of the feature vectors does not decrease, it does not guarantee that the algorithm will converge to the global maximum for a multi-modal distribution. This implies that the EM algorithm can converge to a local maximum. This can be avoided with multiple replications of the algorithm executed with different initial seeds. The EM algorithm is well suited to operate on data sets that contain missing vectors and data sets with low feature space dimensionality.

4.6 DETERMINING THE NUMBER OF CLUSTERS

The most difficult design consideration is to determine the correct number of clusters that should be extracted from the data set. Hundreds of methods have been developed to determine the number of clusters within a data set. The choice in determining the number of clusters K is always ambiguous and is a distinct issue from the process of actually solving the unsupervised clustering problem.

The problem if the number of clusters K is increased without penalty in the design phase (which defeats the purpose of clustering), is that the number of incorrect cluster assignments will steadily decrease to zero. In the extreme case; each feature vector is assigned to its own cluster, which results in zero incorrect clustering allocations. Intuitively this makes the choice in the number of clusters a balance between the maximum compression of the feature vectors into a single cluster and complete accuracy by assigning each feature vector to it own cluster.

The silhouette value is used as a measure of how close each feature vector is to its own cluster when compared to feature vectors in neighbouring clusters [183]. The silhouette value $\mathcal{S}(\vec{x}^p, K)$ for the feature vector $\vec{x}^p$ is computed as

$$\mathcal{S}(\vec{x}^p, K) = \frac{\min\{\mathcal{S}_{\mathrm{BD}}(\vec{x}^p, l) - \mathcal{S}_{\mathrm{WD}}(\vec{x}^p)\}}{\max\{\mathcal{S}_{\mathrm{WD}}(\vec{x}^p), \min\{\mathcal{S}_{\mathrm{BD}}(\vec{x}^p, k)\}\}}, \quad \forall k, l. \tag{4.28}$$

The function $\mathcal{S}_{\mathrm{WD}}(\vec{x}^p)$ denotes the average distance for the feature vector $\vec{x}^p$ to the other feature vectors in the same cluster. The cluster index is denoted by $k, k \in \mathbb{N}, 1 \leq k \leq K$, and $\mathcal{S}_{\mathrm{BD}}(\vec{x}^p, k)$ denotes the average distance for the feature vector $\vec{x}^p$ to the feature vectors in the k^{th} cluster. The average distance within the same cluster $\mathcal{S}_{\mathrm{WD}}(\vec{x}^p)$ for the feature vector $\vec{x}^p$ is computed as

$$\mathcal{S}_{\mathrm{WD}}(\vec{x}^p) = \left\{ \sum_{q=1}^{|\vartheta^{\mathcal{F}_{\mathcal{C}}(\vec{x}^p)}|} \frac{D(\vec{x}^p, \vec{x}^q)}{|\vartheta^{\mathcal{F}_{\mathcal{C}}(\vec{x}^p)}| - 1} : \forall \vec{x}^q \in \vartheta^{\mathcal{F}_{\mathcal{C}}(\vec{x}^p) \backslash \vec{x}^p} \right\}. \tag{4.29}$$

The variable $|\vartheta^{\mathcal{F}_{\mathcal{C}}(\vec{x}^p)}|$ denotes the number of feature vectors in the cluster where $\vec{x}^p$ reside. The average distance between the feature vector $\vec{x}^p$ and the k^{th} cluster is computed as

$$\mathcal{S}_{\mathrm{BD}}(\vec{x}^p, k) = \left\{ \sum_{q=1}^{|\vartheta^{\mathcal{F}_{\mathcal{C}}(\vec{x}^q)}|} \frac{D(\vec{x}^p, \vec{x}^q)}{|\vartheta^{\mathcal{F}_{\mathcal{C}}(\vec{x}^q)}|} : \forall \vec{x}^q \in \vartheta^{\mathcal{F}_{\mathcal{C}}(\vec{x}^q)}, \vec{x}^q \notin \vartheta^{\mathcal{F}_{\mathcal{C}}(\vec{x}^p)}, \mathcal{F}_{\mathcal{C}}(\vec{x}^q) = y^k \right\}. \tag{4.30}$$

The variable $|\vartheta^{\mathcal{F}_{\mathcal{C}}(\vec{x}^q)}|$ denotes the number of feature vectors within the k^{th} cluster.

The silhouette value $\mathcal{S}(\vec{x}^p, K)$ ranges from -1 to 1. A silhouette value $\mathcal{S}(\vec{x}^p, K) \to 1$ indicates that the feature vector $\vec{x}^p$ is very distant from the neighbouring K clusters. A silhouette value $\mathcal{S}(\vec{x}^p, K) \to 0$ indicates the feature vector $\vec{x}^p$ is close to the decision boundary between two clusters. A silhouette value $\mathcal{S}(\vec{x}^p, K) \to -1$ indicates that the feature vector $\vec{x}^p$ is probably in the wrong cluster.

A silhouette graph is a visual representation of the silhouette values and is a visual aid used to determine the number of clusters. The x-axis denotes the silhouette values and the y-axis denotes the cluster labels. The silhouette graph shown in figure 4.5 was created from a larger set of segments defined in the example of land cover classification (figure 4.3). In this silhouette graph; cluster 3 has high silhouette values present, which implies that the current feature vectors within cluster 3 are well separated from the other two clusters. Cluster 1 also has high silhouette values, but with a few feature vectors considered to be ill-positioned. Cluster 2 has significantly lower silhouette values and most of its feature vectors are closely positioned at the boundary between clusters. This might suggest that

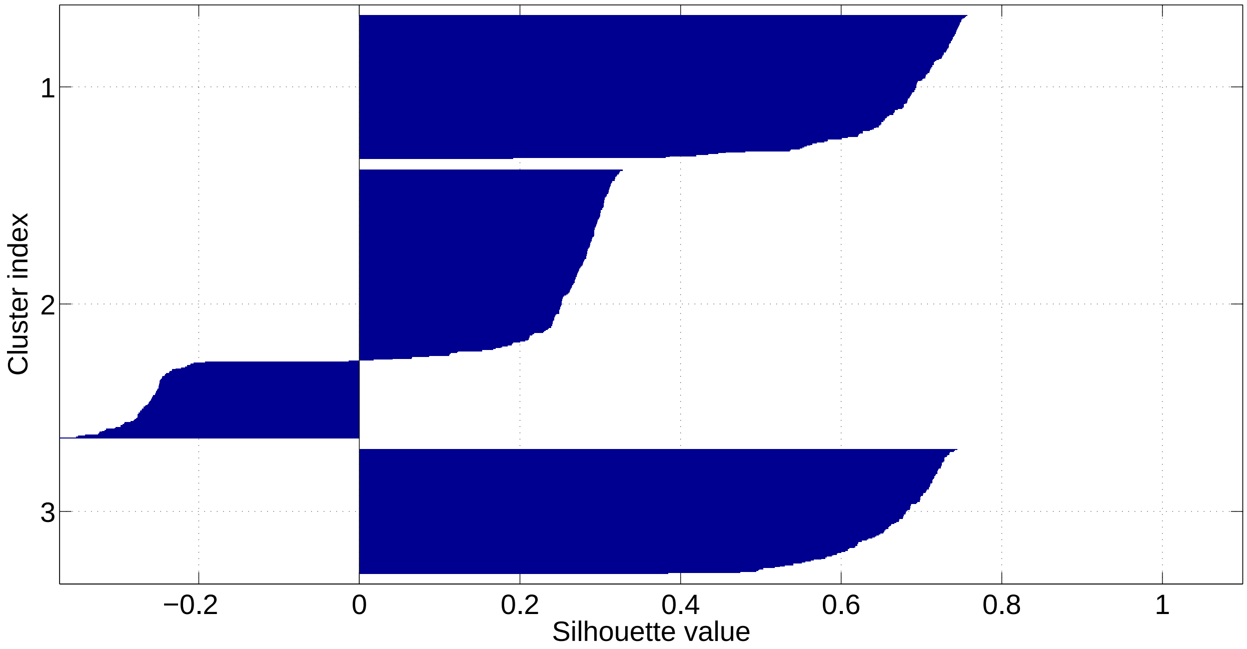

FIGURE 4.5: A silhouette plot of 3 clusters formed of example given in figure 4.3.

cluster 2 can be subdivided into two separate clusters.

An analytical method of deciding on the correct number of clusters K, is the computation of the average of the silhouette value. The average silhouette value is calculated as

$$\mathcal{S}_{\text{ave}}(\{\vec{x}\}, K) = \sum_{p=1}^{P_{\max}} \mathcal{S}(\vec{x}^p, K), \tag{4.31}$$

where $P_{\max}$ denotes the total number of feature vectors in set $\{\vec{x}\}$. A range of K can be evaluated without any prior knowledge to determine the performance of the clustering algorithm. The number of clusters K that produces the highest average silhouette value is then selected.

4.7 CLASSIFICATION OF CLUSTER LABELS

Clusters typically encapsulate properties of the feature vector set and this homogeneous property motivates the assignment of class labels to the clusters. The class labels are assigned using a supervised classifier, which assigns a set of class labels $\{\mathcal{C}_k\}$ to the K cluster labels [171].

The supervised classifier assigns a class label to a cluster with the most frequently occurring class label from the labelled training data set. Assigning the class labels to the cluster labels with a supervised classifier is expressed as

$$\mathcal{C}_k = \mathcal{Z}(y^k). \tag{4.32}$$

Owing to the fact that there is no *one cluster represents one class* property, feature vectors of a certain class might end up in the incorrect cluster and therefore be assigned the wrong class label.

Land cover example: The clustering algorithm uses a function $\mathcal{F}_C$ to assign a cluster label to each of the two segments in figure 4.1. The supervised classifier is then used to assign a class label to each of the clusters. In this example the number of clusters K is set to two and the supervised classifier will assign either the natural vegetation class or the human settlement class to the cluster label. This is accomplished by mapping the cluster label y^k, as

$$\mathcal{C}_k = \begin{cases} \mathcal{C}_1(\text{natural vegetation}) & \text{if } y^k = 1 \\ \mathcal{C}_2(\text{human settlement}) & \text{if } y^k = 2. \end{cases} \tag{4.33}$$

The cluster label y^k is classified as natural vegetation when the label is in the first cluster and human settlement when the label is in the second cluster. □

4.8 SUMMARY

In this chapter a methodology was presented to aid in the design process of an unsupervised classifier. The way in which a clustering method tends to find clusters in the feature space irrespective of whether any real clusters exist was discussed. This shows that proper design criteria must be adhered to and the most practical approach to designing a clustering method is to *learn from example* [171].

The design of the clustering method requires the simultaneous optimisation of the:

- feature extraction and feature selection,
- clustering algorithm, and
- similarity metric.

Six popular clustering algorithms were explored. These algorithms are based on basic concepts, which explore the properties of the feature vectors. Thousands of clustering algorithms have been developed in the last couple of decades and most of them only use different permutations and combinations of the concepts defined in these six clustering algorithms. These basic concepts will provide insight into the intrinsic properties of the feature vectors that populate a high-dimensional feature space.