ut optimally, according to the adopted
t famous algorithm of this category is
n [Lloy 82, Duda 73].
ithms, follow Bayesian classification
s assigned to the cluster C_i for which
robability) is maximum. These prob-
propriately defined optimization task.
where a vector belongs to a specific

ithms. In this case we measure the
x to belong to a cluster C_i.
s. Instead of determining the clusters
es, these algorithms adjust iteratively
where clusters lie. These algorithms,
a cost function optimization philoso-
e from the above algorithms. All the
luster representatives and the goal is
optimal way. In contrast, boundary
s of placing optimally boundaries bet-
o the decision to treat these algorithms
with algorithms to be discussed next.
me special clustering techniques that
ous categories. These include:
lgorithms. These algorithms provide
lustering without having to consider
ixed number m of clusters, and for
wever, their computational burden is

These algorithms use an initial popu-
and iteratively generate new popula-
in better clusterings than those of the
g to a prespecified criterion.
These are methods that guarantee,
ergence in probability to the globally
ect to a prespecified criterion, at the
ions.
ithms. These algorithms treat the fea-
ultidimensional) random variable x.
nly accepted assumption that regions
e correspond to regions of increased
ability density function (pdf) of x.
e pdf may highlight the regions where

—*Competitive learning algorithms*. These are iterative schemes that do not employ cost functions. They produce several clusterings and they converge to the most "sensible" one, according to a distance metric. Typical representatives of this category are the *basic competitive learning scheme* and the *leaky learning algorithm*.

—*Algorithms based on morphological transformation techniques*. These algorithms use morphological transformations in order to achieve better separation of the involved clusters.

12.3 SEQUENTIAL CLUSTERING ALGORITHMS

In this section we describe a basic sequential algorithmic scheme (BSAS) (which is a generalization of that discussed in [Hall 67]) and we also give some variants of it. First, we consider the case where all the vectors are presented to the algorithm *only once*. *The number of clusters is not known a priori in this case*. In fact, new clusters are created as the algorithm evolves.

Let $d(\boldsymbol{x}, C)$ denote the distance (or dissimilarity) between a feature vector $\boldsymbol{x}$ and a cluster C. This may be defined by taking into account either all vectors of C or a representative vector of it (see Chapter 11). *The user-defined parameters required by the algorithmic scheme are the threshold of dissimilarity* Θ *and the maximum allowable number of clusters*, q. The basic idea of the algorithm is the following: As each new vector is considered, it is either assigned to an existing cluster or assigned to a newly created cluster, depending on its distance from the already formed ones. Let m be the number of clusters that the algorithm has created up to now. Then the algorithmic scheme may be stated as:

Basic Sequential Algorithmic Scheme (BSAS)

- $m = 1$
- $C_m = \{\boldsymbol{x}_1\}$
- For $i = 2$ to N
  - —Find C_k: $d(\boldsymbol{x}_i, C_k) = \min_{1 \le j \le m} d(\boldsymbol{x}_i, C_j)$
  - —If $(d(\boldsymbol{x}_i, C_k) > \Theta)$ AND $(m < q)$ then
    - $m = m + 1$
    - $C_m = \{\boldsymbol{x}_i\}$
  - —Else
    - $C_k = C_k \cup \{\boldsymbol{x}_i\}$
    - Where necessary, update representatives.[2]
  - —End { if }
- End { For }

[2]This statement is activated in the cases where each cluster is represented by a single vector. For example, if each cluster is represented by its mean vector, this must be updated each time a new vector becomes a member of the cluster.

Different choices of $d(\boldsymbol{x}, C)$ lead to different algorithms and any of the measures introduced in Chapter 11 can be employed. When C is represented by a single vector, $d(\boldsymbol{x}, C)$ becomes

$$d(\boldsymbol{x}, C) = d(\boldsymbol{x}, \boldsymbol{m}_C) \tag{12.4}$$

where $\boldsymbol{m}_C$ is the representative of C. In the case in which the mean vector is used as a representative, the updating may take place in an iterative fashion, that is,

$$\boldsymbol{m}_{C_k}^{new} = \frac{(n_{C_k^{new}} - 1)\boldsymbol{m}_{C_k}^{old} + \boldsymbol{x}}{n_{C_k^{new}}} \tag{12.5}$$

where $n_{C_k^{new}}$ is the cardinality of C_k after the assignment of $\boldsymbol{x}$ to it and $\boldsymbol{m}_{C_k}^{new}$ ($\boldsymbol{m}_{C_k}^{old}$) is the representative of C_k after (before) the assignment of $\boldsymbol{x}$ to it (Problem 12.2).

Algorithms where each cluster is represented by a single vector are said to be based on *global clustering criteria* [Jain 88] and algorithms where all vectors are used for its representation are said to be based on *local clustering criteria.*

It is not difficult to realize that the order in which the vectors are presented to the BSAS plays an important role in the clustering results. Different presentation ordering may lead to totally different clustering results, in terms of the number of clusters as well as the clusters themselves (see Problem 12.3).

Another important factor affecting the result of the clustering algorithm is the choice of the threshold Θ. This value directly affects the number of clusters formed by BSAS. If Θ is too small, unnecessary clusters will be created. On the other hand, if Θ is too large a smaller than appropriate number of clusters will be created. In both cases, the number of clusters that best fits the data set is missed.

If the number q of the maximum allowable number of clusters is not constrained, we leave to the algorithm to "decide" about the appropriate number of clusters. Consider for example Figure 12.1, where three compact and well-separated clusters are formed by the points of X. If the maximum allowable number of clusters is set equal to two, the BSAS algorithm will be unable to discover three clusters. Probably, in this case the two rightmost groups of points will form a single cluster. On the other hand, if q is uncontrained the BSAS algorithm will probably form three clusters (with an appropriate choice of Θ), at least for the case in which the mean vector is used as a representative. However, constraining q becomes necessary when dealing with implementations where the available computational resources are limited. In the next subsection a simple technique is given for determining the number of clusters.[3]

[3] This problem is also treated in Chapter 16.

FIGURE 12.1: Three clusters are formed by the feature vectors. When q is constrained to a value less than 3, the BSAS algorithm will not be able to reveal them.

Remarks

- The BSAS scheme may be used with similarity instead of dissimilarity measures with appropriate modification; that is, the min operator is replaced by max.
- It turns out that BSAS, with point cluster representatives, favors compact clusters. Thus, it is not recommended if there is strong evidence that other types of clusters are present.
- The preceding algorithm is closely related to the algorithm implemented by the ART2 (adaptive reasonance theory) neural architecture [Carp 87, Burk 91].

12.3.1 Estimation of the Number of Clusters

In this subsection, a simple method is described for determining the number of clusters (see also Chapter 16). The method is suitable for BSAS as well as other algorithms, for which the number of clusters is not required as an input parameter. In what follows, BSAS(Θ) denotes the BSAS algorithm with a specific threshold of dissimilarity Θ.

- For $\Theta = a$ to b step c
  - —Run s times the algorithm BSAS(Θ), each time presenting the data in a different order.
  - —Estimate the number of clusters, m_Θ, as the most frequent number resulting from the s runs of BSAS(Θ).
- Next Θ

The values a and b are the minimum and maximum dissimilarity levels among all pairs of vectors in X, that is, $a = \min_{i,j=1,\ldots,N} d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $b = \max_{i,j=1,\ldots,N} d(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The choice of c is directly influenced by the choice of $d(\boldsymbol{x}, C)$. As far as the value of s is concerned, the greater the s, the larger the statistical sample

and, thus, the higher the accuracy of the results. In the sequel, we plot the number of clusters m_Θ versus Θ. This plot has a number of flat regions. We estimate the number of clusters as the number that corresponds to the largest flat region. It is expected that at least for the case in which the vectors form well-separated compact clusters, this is the desired number. Let us explain this argument intuitively. Suppose that the data form two compact and well-separated clusters C_1 and C_2. Let the minimum distance between two vectors in C_1 (C_2) be r_1 (r_2) and suppose that $r_1 < r_2$. Also let r ($> r_2$) be the minimum among all distances $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$, with $\boldsymbol{x}_i \in C_1$ and $\boldsymbol{x}_j \in C_2$. It is clear that for $\Theta \in [r_2, r - r_2]$, the number of clusters created by BSAS is 2. In addition, if $r >> r_2$, the interval has a big range, and thus it corresponds to a large flat region in the plot of m_Θ versus Θ. Example 12.2 illustrates the idea.

Example 12.2. Consider two 2-dimensional Gaussian distributions with means $[0, 0]^T$ and $[20, 20]^T$, respectively. The covariance matrices are $\Sigma = 0.5I$ for both distributions, where I is the 2×2 identity matrix. Generate 50 points from each distribution (Figure 12.2a). The number of underlying clusters is 2. The plot resulting from the application of the previously described procedure is shown in Figure 12.2b, with $a = \min_{\boldsymbol{x}_i, \boldsymbol{x}_j \in X} d_2(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $b = \max_{\boldsymbol{x}_i, \boldsymbol{x}_j \in X} d_2(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and $c \simeq 0.3$. It can be seen that the biggest flat region corresponds to the number 2, which is the number of underlying clusters.

In the foregoing procedure, we have implicitly assumed that the feature vectors do form clusters. If this is not the case, the method is useless. Methods that deal with the problem of discovering whether any clusters exist are discussed in Chapter 16. Moreover, if the vectors form compact clusters, which are not well separated, the

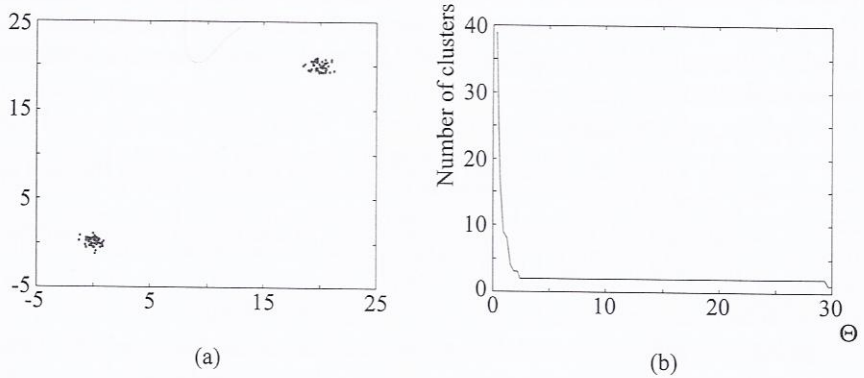
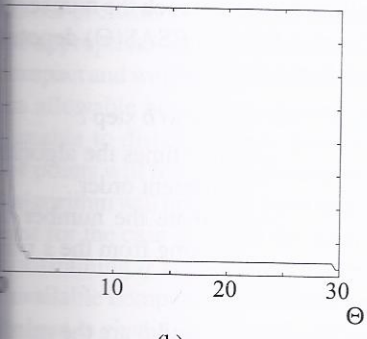


FIGURE 12.2: (a) The data set. (b) The plot of the number of clusters versus Θ. It can be seen that for a wide range of values of Θ, the number of clusters, m, is 2.

ts. In the sequel, we plot the number
ber of flat regions. We estimate the
onds to the largest flat region. It is
ectors form well-separated compact
explain this argument intuitively.
well-separated clusters C_1 and C_2.
rs in C_1 (C_2) be r_1 (r_2) and suppose
among all distances $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$, with
$[r_2, r - r_2]$, the number of clusters
r_2, the interval has a big range, and
plot of m_Θ versus Θ. Example 12.2

ssian distributions with means $[0, 0]^T$
es are $\Sigma = 0.5I$ for both distributions,
ts from each distribution (Figure 12.2a).
sulting from the application of the pre-
2.2b, with $a = \min_{\boldsymbol{x}_i, \boldsymbol{x}_j \in X} d_2(\boldsymbol{x}_i, \boldsymbol{x}_j)$,
seen that the biggest flat region corre-
derlying clusters.

tly assumed that the feature vectors
od is useless. Methods that deal with
rs exist are discussed in Chapter 16.
s, which are not well separated, the



(b)

ot of the number of clusters versus
es of Θ, the number of clusters, m,

procedure may give unreliable results, since it is unlikely for the plot of m_Θ versus Θ to contain wide flat regions.

In some cases, it may be advisable to consider all the numbers of clusters, m_Θ, that correspond to all flat regions of considerable size in the plot of m_Θ versus Θ. If, for example, we have three clusters and the first two of them lie close to each other and away from the third, the flattest region may occur for $m_\Theta = 2$ and the second flattest for $m_\Theta = 3$. If we discard the second flattest region, we will miss the three-cluster solution (Problem 12.6).

12.4 A MODIFICATION OF BSAS

As has already been stated, the basic idea behind BSAS is that each input vector $\boldsymbol{x}$ is assigned to an already created cluster or a new one is formed. Therefore, a decision for the vector $\boldsymbol{x}$ is reached *prior to the final cluster formation*, which is determined after all vectors have been presented. The following refinement of BSAS, which will be called modified BSAS (MBSAS), overcomes this drawback. The cost we pay for it is that the vectors of X have to be presented twice to the algorithm. The algorithmic scheme consists of two phases. The first phase involves the determination of the clusters, via the assignment of *some* of the vectors of X to them. During the second phase, the unassigned vectors are presented for a second time to the algorithm and are assigned to the appropriate cluster. The MBSAS may be written as follows:

Modified Basic Sequential Algorithmic Scheme (MBSAS)

Cluster Determination

- $m = 1$
- $C_m = \{\boldsymbol{x}_1\}$
- For $i = 2$ to N
  - —Find C_k: $d(\boldsymbol{x}_i, C_k) = \min_{1 \le j \le m} d(\boldsymbol{x}_i, C_j)$
  - —If $(d(\boldsymbol{x}_i, C_k) > \Theta)$ AND $(m < q)$ then
    - $m = m + 1$
    - $C_m = \{\boldsymbol{x}_i\}$
  - —End { if }
- End { For}

Pattern Classification

- For $i = 1$ to N
  - —If $\boldsymbol{x}_i$ has not been assigned to a cluster, then
    - Find C_k: $d(\boldsymbol{x}_i, C_k) = \min_{1 \le j \le m} d(\boldsymbol{x}_i, C_j)$
    - $C_k = C_k \cup \{\boldsymbol{x}_i\}$
    - Where necessary, update representatives.
  - —End { if }
- End { For }

The number of clusters is determined in the first phase and then it is frozen. Thus, the decision taken during the second phase for each vector takes into account all clusters.

When the mean vector of a cluster is used as its representative, the appropriate cluster representative has to be adjusted using Eq. (12.5), after the assignment of each vector in a cluster.

Also, as it was the case with BSAS, MBSAS is sensitive to the order in which the vectors are presented.

Finally, it must be stated that, after minor modifications, MBSAS may be used when a similarity measure is employed (see Problem 12.7).

12.5 A TWO-THRESHOLD SEQUENTIAL SCHEME

As already pointed out, the results of BSAS and MBSAS are strongly dependent on the order in which the vectors are presented to the algorithm, as well as on the value of Θ. Improper choice of Θ may lead to meaningless clustering results. One way to overcome these difficulties is to define a "gray" region (see [Trah 89]). This is achieved by employing two thresholds, Θ_1 and $\Theta_2 (> \Theta_1)$. If the dissimilarity level $d(\boldsymbol{x}, C)$ of a vector $\boldsymbol{x}$ from its closest cluster C is less than Θ_1, $\boldsymbol{x}$ is assigned to C. If $d(\boldsymbol{x}, C) > \Theta_2$, a new cluster is formed and $\boldsymbol{x}$ is placed in it. Otherwise, if $\Theta_1 \leq d(\boldsymbol{x}, C) \leq \Theta_2$, there exists uncertainty and the assignment of $\boldsymbol{x}$ to a cluster will take place at a later stage. Let $clas(\boldsymbol{x})$ be a flag that indicates whether $\boldsymbol{x}$ has been classified (1) or not (0). Again, we denote by m the number of clusters that have been formed up to now. In the following, we assume no bounds to the number of clusters (i.e., $q = N$). The algorithmic scheme is:

The Two-Threshold Sequential Algorithmic Scheme (TTSAS)

$m = 0$
$clas(\boldsymbol{x}) = 0, \quad \forall \boldsymbol{x} \in X$
$prev\_change = 0$
$cur\_change = 0$
$exists\_change = 0$

While (there exists at least one feature vector $\boldsymbol{x}$ with $clas(\boldsymbol{x}) = 0$) do

- For $i = 1$ to N
  - —if $clas(\boldsymbol{x}_i) = 0$ AND it is the first in the new while loop AND $exists\_change = 0$ then
    - $* \; m = m + 1$
    - $* \; C_m = \{\boldsymbol{x}_i\}$

* $clas(\boldsymbol{x}_i) = 1$
* $cur\_change = cur\_change + 1$

—Else if $clas(\boldsymbol{x}_i) = 0$ then

* Find $d(\boldsymbol{x}_i, C_k) = \min_{1 \le j \le m} d(\boldsymbol{x}_i, C_j)$
* if $d(\boldsymbol{x}_i, C_k) < \Theta_1$ then
  - $C_k = C_k \cup \{\boldsymbol{x}_i\}$
  - $clas(\boldsymbol{x}_i) = 1$
  - $cur\_change = cur\_change + 1$
* else if $d(\boldsymbol{x}_i, C_k) > \Theta_2$ then
  - $m = m + 1$
  - $C_m = \{\boldsymbol{x}_i\}$
  - $clas(\boldsymbol{x}_i) = 1$
  - $cur\_change = cur\_change + 1$
* End {If}

—Else if $clas(\boldsymbol{x}_i) = 1$ then

* $cur\_change = cur\_change + 1$

—End {If}

- End { For}
- $exists\_change = |cur\_change - prev\_change|$
- $prev\_change = cur\_change$
- $cur\_change = 0$

End {While}

The $exists\_change$ checks whether there exists at least one vector that has been classified at the current pass on X (i.e., the current iteration of the while loop). This is achieved by comparing the number of vectors that have been classified up to the current pass on X, cur_change, with the number of vectors that have been classified up to the previous pass on X, $prev\_change$. If $exists\_change = 0$, that is, no vector has been assigned to a cluster during the last pass on X, the first unclassified vector is used for the formation of a new cluster.

The first *if* condition in the *For* loop ensures that the algorithm terminates after N passes on X (N executions of the *while* loop) at the most. Indeed, this condition forces the first unassigned vector to a new cluster when no vector has been assigned during the last pass on X. This gives a way out to the case in which no vector has been assigned at a given circle.

However, in practice, the number of required passes is much less than N. It should be pointed out that this scheme is almost always at least as expensive as the previous two schemes, because in general it requires at least two passes on X. Moreover, since the assignment of a vector is postponed until enough information