Fuzzy Clustering Based on Dissimilarity Relations Extracted from Data

Mario G.C.A. Cimino, Beatrice Lazzerini, and Francesco Marcelloni

Dipartimento di Ingegneria dell'Informazione: Elettronica, Informatica, Telecomunicazioni, University of Pisa, Pisa, Italy

13.1 INTRODUCTION

Clustering algorithms partition a collection of data into a certain number of clusters (groups, subsets, or categories). Though there is no universally agreed definition, most researchers describe a cluster by considering the internal homogeneity and the external separation (Xu and Wunsch, 2005), i.e., patterns in the same cluster should be similar to each other, while patterns in different clusters should not (Jain, Murty, and Flynn 1999; Su and Chou, 2001). Thus, the correct identification of clusters depends on the definition of similarity. Typically, similarity (more often dissimilarity) is expressed in terms of some distance function, such as the Euclidean distance or the Mahalanobis distance. The choice of the (dis)similarity measure induces the cluster shape and therefore determines the success of a clustering algorithm on the specific application domain. For instance, the Euclidean and Mahalanobis distances lead clustering algorithms to determine hyperspherical-shaped or hyperellipsoidal-shaped clusters, respectively. Typically, when we apply a clustering algorithm, we do not know a priori the most natural and effective cluster shapes for the specific data-set. Each data-set is characterized by its own data distribution and therefore requires cluster shapes different from other data-sets. Nevertheless, we have to choose the dissimilarity measure before starting the clustering process. For instance, when we apply the classical fuzzy C-means (FCM) (Bezdek, 1981), which is one of the best known partitional fuzzy clustering algorithms, we decide a priori to use the Euclidean distance and therefore to identify hyperspherical-shaped clusters. To overcome this problem, in the literature, several approaches have been proposed. For instance, density-based clustering algorithms determine on-line the shape of clusters. In density-based clustering, clusters are regarded as regions in the data space in which the objects are dense. These regions may have an arbitrary shape and the points inside a region may be arbitrarily distributed. To determine if a region is dense, we need to define the concept of neighborhood, based on a priori defined proximity (see, for instance, DBSCAN, Ester, Kriegel, Sander, and Xu, 1996, or OPTICS, Ankerst, Breunig, Kriegel, and Sander, 1999). Though proximity can be defined in terms of any dissimilarity measure, applications of density-based clustering algorithms proposed in the literature adopt a distance function to determine spatial proximity. Thus, again, though the shapes of clusters may be different from each other, they still depend on the a priori choice of a distance.

When applying clustering to data with irregular distribution, as is often the case for image segmentation and pattern recognition (Jain and Flynn, 1993), distance functions cannot adequately model dissimilarity (Valentin, Abdi, O'Toole, and Cottrell, 1994; Kamgar-Parsi and Jain, 1999; Santini and Jain, 1999; Latecki and Lakamper, 2000). Consider, for example, the pixels of an image made up of distinguishable elements with irregular-shaped contours (for instance, bikes, cars, houses, trees). The dissimilarity between pixels should be small (large) when the pixels belong to the same image element (different image elements).

To solve this problem, some approaches have been proposed in the literature. For example, Jarvis and Patrick (1973) defined the dissimilarity between two points as a function of their context, i.e., the set of points in the neighborhood of each such point. Michalski, Stepp, and Diday (1983) used predefined concepts to define the "conceptual similarity" between points. Yang and Wu (2004) proposed adopting a total similarity related to the approximate density shape estimation as objective function of their clustering method. Jacobs, Weinshall, and Gdalyahu (2000) observed that classification systems, which can model human performance or use robust image matching methods, often exploit similarity judgement that is non-metric. Makrogiannis, Economou, and Fotopoutos (2005) introduced a region dissimilarity relation that combines feature-space and spatial information for color image segmentation.

Pedrycz (2005) suggested exploiting some auxiliary information (*knowledge-based hints*), which reflect some additional sources of domain knowledge, in order to guide the clustering process. He, first, proposed a general taxonomy of knowledge-based hints. Then, he discussed some clustering algorithms which partition the data-set guided by these hints. In particular, he considered a partially supervised version of the classical FCM algorithm (Pedrycz and Waletzky, 1997), which uses some labeled patterns as knowledge-based hints: these labeled patterns serve as reference elements in modeling the cluster shapes. Further, he discussed a proximity-based fuzzy clustering algorithm where knowledge-based hints are represented by proximity values between pairs of patterns (Pedrycz, Loia, and Senatore, 2004). Similarly, Lange, Law, Jain, and Buhmann (2005) or Law, Topchy, and Jain (2005) proposed exploiting a priori knowledge about a desired model via two types of pairwise constraints: must-link and must-not-link constraints. The two constraints correspond to the requirements that two objects should and should not be associated with the same label, respectively.

A different approach proposed extracting the dissimilarity relation directly from the data by guiding the extraction process itself with as little supervision as possible (Pedrycz *et al.*, 2001). Following this approach, Hertz, Bar-Hillel, and Weinshall (2004) suggested learning distance functions by using a subset of labeled data. In particular, they trained binary classifiers with margins, defined over the product space of pairs of images, to discriminate between pairs belonging to the same class and pairs belonging to different classes. The signed margin is used as a distance function. Both support vector machines and boosting algorithms are used as product space classifiers. Using some benchmark databases from the UCI repository, the authors showed that their approach significantly outperformed existing metric learning methods based on learning the Mahalanobis distance.

Recently, some methods have been proposed to exploit pairwise dissimilarity information for learning distance functions (Xing, Ng, Jordan, and Russell, 2003). Tsang, Cheung, and Kwok (2005), for instance, proposed learning distance metric from a subset of pairwise dissimilarity values by a kernelized version of the relevant component analysis method. Chang and Yeung (2005) formulated the metric learning problem as a kernel learning problem, which is efficiently solved by kernel matrix adaptation.

Similarly, in this chapter, we will discuss how the dissimilarity relation can be extracted directly from a few pairs of data with known dissimilarity values rather than from pairs of data with known labels. We will discuss the application of two different techniques based on, respectively, neural networks and fuzzy systems. More precisely, we use a multilayer perceptron (MLP) with supervised learning (Haykin, 1999) and a Takagi–Sugeno (TS) fuzzy system (Takagi and Sugeno, 1985). The rules of the TS are identified by using the method proposed by Setnes and Roubos (2000). Once the MLP has been trained and the TS has been identified, the two models can associate a dissimilarity value with each pair of patterns in the data-set. This relation, extracted from the data, can be exploited by a relational clustering algorithm to partition the data-set into a suitable number of clusters.

In real applications, clusters are generally overlapped and their boundaries are fuzzy rather than crisp. The identification of these clusters demands appropriate fuzzy relational clustering algorithms. To the best of our knowledge, fuzzy relational clustering algorithms proposed in the literature require dissimilarity relations which are symmetric and irreflexive (Bezdek, Keller, Krisnapuram, and Pal, 1999). On the other hand, the generalization performed by the MLP and by the TS may produce a relation that is neither symmetric nor irreflexive. For instance, a pattern not included in the training set may be judged slightly dissimilar to itself. Further, the dissimilarity value between pattern $\mathbf{x}_i$ and pattern $\mathbf{x}_j$ may be different from the dissimilarity value between pattern $\mathbf{x}_j$ and pattern $\mathbf{x}_i$. Thus, though fuzzy relational algorithms may work correctly on the relation produced by the two models, sometimes they converge to solutions which are not sound.

Actually, we observed that some of the best known fuzzy clustering algorithms, when applied to the dissimilarity relations extracted by the MLP and the TS, converge to a partition composed completely superimposed clusters, that is, each pattern in the data-set belongs to all clusters with the same membership grade. To overcome this unstable behavior, in a previous paper we proposed a new approach to fuzzy relational clustering: (Corsini, Lazzerini, and Marcelloni, 2005) starting from the definition of relational clustering algorithm, we transformed a relational clustering problem into an object clustering problem. This transformation allows us to apply any object clustering algorithm to partition sets of objects described by relational data. In the implementation based on the classical FCM algorithm and denoted ARCA (Corsini, Lazzerini, and Marcelloni, 2005), we verified that this approach produces partitions similar to the ones generated by the other fuzzy relational clustering algorithms, when these converge to a sound partition. On the other hand, as FCM has proved to be one of the most stable fuzzy clustering algorithms, ARCA is appreciably more stable than the other fuzzy relational clustering algorithms. In this chapter we show the effectiveness of the combinations MLP–ARCA and TS–ARCA using a synthetic data-set and the Iris data-set, respectively. We describe how these combinations achieve very good clustering performance using a limited number of training samples. Further, we show how the TS can provide an intuitive linguistic description of the dissimilarity relation. Finally, we discuss the performance obtained by the combination TS–ARCA on three real data-sets from the UCI repository.

We wish to point out that the combination of supervised and unsupervised learning discussed in this chapter is intended for use in all cases in which the dissimilarity relation can be learnt from a reasonably small portion of samples, which form the training set. The method works, in principle, with any kind of data-set. In fact, as it unfolds the entire data onto as many dimensions as the number of data points in order to transform the relational clustering to object-based clustering, it is more appropriate for moderate-size data-sets, typically containing up to a few hundreds of patterns.

13.2 DISSIMILARITY MODELING

Our approach to fuzzy clustering is based on extracting the dissimilarity measure that drives the clustering strategy from a small set of known similarities. Thus, we have to generate a model that, given a pair $(\mathbf{x}_i, \mathbf{x}_j)$ of input data, outputs the dissimilarity degree $d_{i,j}$ between $\mathbf{x}_i$ and $\mathbf{x}_j$. The generation of this model is a typical identification problem, which has been tackled by different techniques such as classical mathematical theory, support vector machines, neural networks, and fuzzy modeling. In this work, we discuss the application of two of these techniques: neural networks and fuzzy modeling. In particular, to model the dissimilarity relation, we used a multilayer perceptron (MLP) neural network with supervised learning and a Takagi–Sugeno (TS) fuzzy system. We assume that the patterns are described by numerical features (possibly, nonnumerical features are appropriately transformed into numerical ones) and the dissimilarity degrees[1] between a few pairs of patterns are known. Let $T = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ be the set of

[1] Actually, our method can deal with both similarity and dissimilarity relations.

known data, where $\mathbf{z}_n = [\mathbf{x}_i, \mathbf{x}_j, d_{i,j}] \in \Re^{2F+1}$. In the following two sections, we briefly describe the MLP and the TS used in our experiments.

13.2.1 The Multilayer Perceptron

We use a standard feedforward three-layer MLP neural network. Each neuron is equipped with a sigmoidal nonlinear function. The standard back-propagation algorithm with a dynamically decreasing learning rate is used as a learning scheme. Errors less than 0.001 are treated as zero. Initial weights are random values in the range $\lfloor -1/\sqrt{m}, 1/\sqrt{m} \rfloor$, with m being the number of inputs to a neuron. As described by Corsini, Lazzerini, and Marcelloni (2006), to determine the best structure of the neural network with respect to the generalization capability, we performed a number of experiments with two-layer and three-layer MLP and with a different number of neurons for each hidden layer. For the data-sets discussed in this chapter, we observed that the best generalization properties are obtained by using an architecture with 20 and eight neurons for the first and second hidden layers, respectively. Further, we experimentally verified that this result is quite independent of the size of the training set, at least for the sizes used in the experiments.

13.2.2 The Takagi-Sugeno System

The rules of the TS have the following form:

$$\begin{aligned} r_i\text{: If } & X_{1,1} \text{ is } A_{i,1,1} \text{ and} \ldots X_{1,F} \text{ is } A_{i,1,F} \text{ and } X_{2,1} \text{ is } A_{i,2,1} \text{ and} \ldots X_{2,F} \text{ is } A_{i,2,F} \\ & \text{then } d_i = \mathbf{a}_{i,1}^T \mathbf{X}_1 + \mathbf{a}_{i,2}^T \mathbf{X}_2 + b_i, \qquad i = 1..R \end{aligned}$$

where R is the number of rules, $\mathbf{X}_e = [X_{e,1}, \ldots, X_{e,F}]$, with e=1, 2, are the two input variables of F components that represent the pair of patterns whose dissimilarity has to be evaluated, $A_{i,e,1}, \ldots, A_{i,e,F}$ are fuzzy sets defined on the domain of $X_{e,1}, \ldots, X_{e,F}$, respectively, $\mathbf{a}_{i,e}^T = [a_{i,e,1}, \ldots, a_{i,e,F}]$, with $a_{i,e,f} \in \Re$, and $b_i \in \Re$. The model output d, which represents the dissimilarity between two input patterns, is computed by aggregating the conclusions inferred from the individual rules as follows:

$$d = \frac{\sum\limits_{i=1}^{R} \beta_i d_i}{\sum\limits_{i=1}^{R} \beta_i} \tag{13.1}$$

where $\beta_i = \prod\limits_{f=1}^{F} A_{i,1,f}(x_{j,f}) \prod\limits_{f=1}^{F} A_{i,2,f}(x_{k,f})$ is the degree of activation of the ith rule, when the pair $(\mathbf{x}_j, \mathbf{x}_k)$ is fed as input to the rule.

A TS model is built through two steps, called the *structure identification* and the *parameter identification* (Babuška, 1996). The structure identification determines the number of rules and the variables involved in the rule antecedents. The parameter identification estimates the parameters that define, respectively, the membership functions of the fuzzy sets in the antecedents and the consequent functions. The number of rules is generally computed by exploiting a clustering algorithm (Angelov and Filev, 2004; Abonyi, Babuška, and Szeifert, 2002). More precisely, the number of rules coincides with the number of clusters of the input-output space partition, which results to be the best with respect to an appropriate validity index. The parameter identification is obtained by first computing the fuzzy sets in the antecedent of the rules, and then estimating the parameters of the mathematical functions in the consequent (Angelov and Filev, 2004). One of the most used clustering algorithms to identify the structure of a TS is the classical FCM with Euclidean distance. As the FCM algorithm finds the fuzzy partition starting from a fixed number of clusters, and the number of clusters determines the number of rules that compose the fuzzy model, a criterion has to be adopted to determine the optimal number of clusters. The most common approach is to identify an interval of possible values of the number R of clusters and execute the FCM for each value in the interval. Each execution is therefore assessed against a validity index. Several different validity indexes have been proposed in the literature (Bezdek, Keller, Krisnapuram, and Pal, 1999). The most used among these indexes are the Xie and Beni's index (XB, Xie, and Beni, 1991), the Fukuyama and Sugeno's index (Pal and Bezdek, 1995), the Gath and Geva's index (Gath and Geva, 1989), and the Rezaee, Lelieveldt, and Reiber's index (Rezaee, Lelieveldt, and Reiber, 1998). As is well known in the literature, there does not exist a validity index which is good for each data-set (Pal and Bezdek, 1995). In order to choose the most reliable index for the data-sets used in the experiments, we compared the aforementioned validity indexes against the TS accuracy obtained with the number of rules determined by the indexes. We observed that the XB index guarantees the best results. The Xie–Beni index is defined as

$$XB(U,V,T) = \frac{\sum_{i=1}^{R}\sum_{n=1}^{N} u_{i,n}^{2} \parallel \mathbf{z}_n - \mathbf{v}_i \|^2}{N(\min_{j \neq k} \parallel \mathbf{v}_j - \mathbf{v}_k \|^2)},$$

where V is the vector of cluster prototypes $\mathbf{v}_i$ and U is the fuzzy partition matrix whose generic element $u_{i,n}^2$ represents the grade of membership of $\mathbf{z}_n$ to cluster i. The numerator of the fraction measures the compactness of the clusters while the denominator measures the degree of separation of the cluster prototypes. For compact and well-separated clusters we expect small values of XB. We execute the FCM algorithm with increasing values of the number R of clusters for values of the fuzzification constant m in $\{1.4, 1.6, 1.8, 2.0\}$ and plot the Xie–Beni index versus R. We choose, as the optimal number of clusters, the value of R corresponding to the first distinctive local minimum.

The combination of the FCM and the Xie–Beni index helps determining only the rules that describe important regions of the input/output space, thus leading to a moderate number of rules. Fuzzy sets $A_{i,e,f}$ are obtained by projecting the rows of the partition matrix U onto the fth component of the input variable $\mathbf{X}_e$ and approximating the projections by triangular membership functions $A_{i,e,f}(l_{i,e,f}, m_{i,e,f}, r_{i,e,f})$ with $l_{i,e,f} < m_{i,e,f} < r_{i,e,f}$ real numbers on the domain of definition of $X_{e,f}$. We computed the parameter $m_{i,e,f}$, which corresponds to the abscissa of the vertex of the triangle, as the weighted average of the $X_{e,f}$ components of the training patterns, the weights being the corresponding membership values. Parameters $l_{i,e,f}$ and $r_{i,e,f}$ were obtained as intersection of the $X_{e,f}$ axis with the lines obtained as linear regression of the membership values of the training patterns, respectively, on the left and the right sides of $m_{i,e,f}$. Obviously, if $l_{i,e,f}$ and $r_{i,e,f}$ are beyond the extremes of the definition domain of variable $X_{e,f}$, the sides of the triangles are truncated in correspondence to the extremes. The use of triangular functions allows easy interpretation of the fuzzy sets in linguistic terms. Once the antecedent membership functions have been fixed, the consequent parameters $[\mathbf{a}_{i,1},\ \mathbf{a}_{i,2},\ b_i]$, $i = 1..R$, of each individual rule i are obtained as a local least squares estimate.

The strategy used so far to build the TS is aimed at generating a rule base characterized by a number of interesting properties, such as a moderate number of rules, membership functions distinguishable from each other, and space coverage, rather than at minimizing the model error. We experimentally verified that this TS could show a poor performance, in particular for training sets composed of a high number of pairs. Thus, we apply a genetic algorithm (GA) to tune simultaneously the parameters in the antecedent and consequent parts of each rule in a global optimization. To preserve the good properties of the fuzzy model, we impose that no gap exists in the partition of each input variable. Further, to preserve distinguishability we allow the parameters that define the fuzzy sets to vary within a range around their initial values. Each chromosome represents the entire fuzzy system, rule by rule, with the antecedent and consequent parts (see Figure 13.1). Each rule antecedent consists of a sequence of $2 \cdot F$ triplets (l, m, r) of real numbers representing triangular membership functions, whereas each rule consequent contains $2 \cdot F + 1$ real numbers corresponding to the consequent parameters. The fitness value is the inverse of the mean square error (MSE) between the predicted output and the desired output over the training set.

We start with an initial population composed of 70 chromosomes generated as follows. The first chromosome codifies the system generated by the FCM, the others are obtained by perturbing the first chromosome randomly within the ranges fixed to maintain distinguishability. At each generation, the arithmetic crossover and the uniform mutation operators are applied with probabilities 0.8 and 0.6, respectively. Chromosomes to be mated are chosen by using the well-known roulette wheel selection method. At each generation, the offspring are cheeked against the aforementioned space coverage

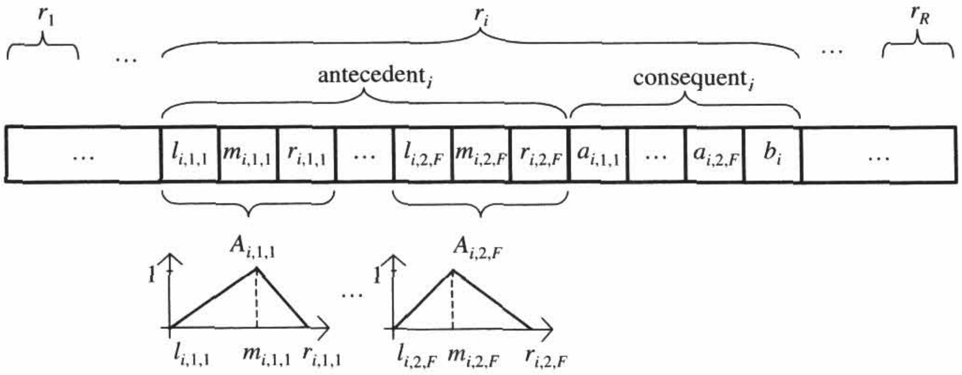

Figure 13.1 The chromosome structure.

criterion. To speed up the convergence of the algorithm without significantly increasing the risk of premature convergence to local minima, we adopt the following acceptance mechanism: 40 % of the new population is composed of offspring, whereas 60 % consists of the best chromosomes of the previous population. When the average of the fitness values of all the individuals in the population is greater than 99.9 % of the fitness value of the best individual or a prefixed number of iterations has been executed (6000 in the experiments), the GA is considered to have converged.

The fairly large size of the population and the mutation probability higher than usual have been chosen to counteract the effect of the strong exploitation of local linkages. Indeed, due to real coding (Wright, 1991) and to constraints imposed on the offspring so as to maintain distinguishability, exploitation could lead to a premature convergence to sub-optimal solutions. The values of the GA parameters used in the experiments reduce this risk. To strengthen this consideration, we observed in the experiments that, varying the data-set, the values of the GA parameters do not need to be changed.

13.2.3 MLP versus TS

To compare the two approaches, we used the synthetic data-set shown in Figure 13.2 and the Iris data-set (UCI, 2006). The first data-set was chosen because clustering algorithms, which measure the dissimilarity

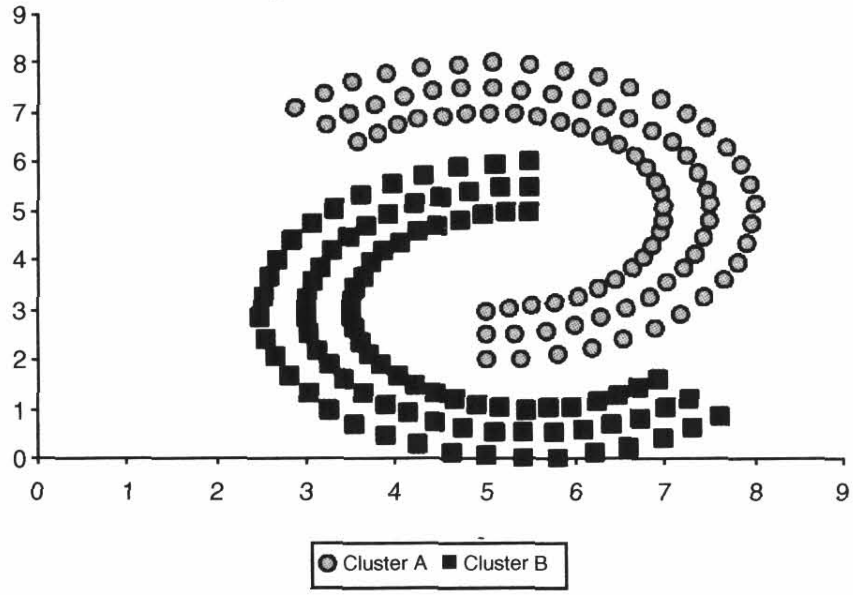

Figure 13.2 The synthetic data-set.

between two points as the distance between the two points, cannot partition it correctly. Indeed, both the Euclidean and the Mahalanobis distances that induce, respectively, spherical and ellipsoidal cluster shapes lead, for instance, the FCM algorithm and the GK algorithm (Gustafson and Kessel, 1979) to partition the data-set incorrectly (Corsini, Lazzerini, and Marcelloni, 2006).

For each data-set, we carried out five experiments. In these experiments, we aimed to assess how much the size of the training pool affected the performance of the MLP and the TS. For this purpose, we randomly extracted a pool of patterns (called the *training pool*) from the data-set. This pool was composed of 5 %, 10 %, 15 %, 20 %, and 25 % of the data-set, respectively, in the five experiments. Then, we built the training set by selecting a given number of pairs of patterns from the training pool. More precisely, assume that C is the number of clusters, which we expect to identify in the data-set. Then, for each pattern $\mathbf{x}_i$ in the training pool, we formed $q \cdot C$ pairs $(\mathbf{x}_i,\mathbf{x}_j)$, with $q \in [1..8]$, by randomly selecting $q \cdot C$ patterns $\mathbf{x}_j$ of the training pool as follows: q patterns were chosen among those with dissimilarity degree lower than 0.5 with $\mathbf{x}_i$, and the remaining $q \cdot (C\text{-}1)$ patterns were chosen among those with dissimilarity degree higher than 0.5.

This choice tries to provide the same number of training samples for pairs of points belonging to different clusters as for pairs of points belonging to the same cluster. Obviously, since we do not know a priori the membership of each point to a class, this choice is only an approximation. However, we experimentally verified that it provides reliable results. It is obvious that increasing values of q leads to better classification performance, but also to increasing execution times. Obviously, we assumed the dissimilarity degrees between all the pairs that can be built from patterns in the training pool were known. This assumption, which is not actually necessary, was made to test the effects of q on the performance of the MLP and the TS. For the two data-sets, we observed that $q = 5$ provides a good trade off between classification accuracy and execution time. Let $d_{i,j}$ be the degree of dissimilarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. We inserted both $[\mathbf{x}_i, \mathbf{x}_j, d_{i,j}]$ and $[\mathbf{x}_j, \mathbf{x}_i, d_{i,j}]$ into the training set.

We carried out the five experiments described above and, for each experiment, we executed 10 trials. For the sake of simplicity, in the experiments, we used only 0 and 1 to express the dissimilarity degree of two input points belonging to the same class or to different classes, respectively. Please note that we use the knowledge about classes just to assign dissimilarity degrees to pairs of points in the training pool.

To assess the generalization properties, for each trial and each experiment we tested the two models on all possible pairs of points in the data-set and measured the percentage of the point pairs with dissimilarity degree lower than (higher than) 0.5 for pairs of points belonging (not belonging) to the same class.

Tables 13.1 and 13.2 show the percentages of correct dissimilarity values obtained by applying the MLP to the synthetic and the Iris data sets. In the tables, the columns show, respectively, the percentage of points composing the training pool and the percentage (in the form (mean ± standard deviation)) of pattern pairs with correct dissimilarity.

Tables 13.3 and 13.4 show the percentages of correct dissimilarity values obtained by applying the TS system to the synthetic and the Iris data-sets. In the tables, the columns indicate, respectively, the percentage of points composing the training pool, the number of rules of the TS model (in the form (mean ± standard deviation)) and the percentage of correct dissimilarity values before and after the GA optimization. It can be observed that the application of the GA sensibly improves the percentage of correct dissimilarity values generated by the TS model independently of the cardinality of the training pool.

Table 13.1 Percentage of point pairs with correct dissimilarity values (MLP system on the synthetic data-set).

Training pool	Correct dissimilarity values
5%	70.1% ± 5.2%
10%	73.8% ± 4.5%
15%	81.5% ± 4.3%
20%	85.1% ± 3.3%
25%	89.8% ± 2.2%

Table 13.2 Percentage of point pairs with correct dissimilarity values (MLP system on Iris data-set).

Training pool	Correct dissimilarity values
5%	81.2% ± 3.2%
10%	85.5% ± 3.8%
15%	88.1% ± 3.4%
20%	90.4% ± 3.5%
25%	90.7% ± 2.7%

As shown in the tables, the two approaches have similar performance. Both the MLP and the TS achieve about 90 % of correct dissimilarity values with 25 % of the points. We have to consider that the percentage of total pairs of points included in the training set is much lower than the percentage of total points in the training pool. For instance, for the synthetic data-set, a training pool composed of 25 % of the points corresponds to a training set composed of 2.78 % of the dissimilarity values. Taking this into account, the percentages achieved by the two approaches are undoubtedly remarkable.

As regards the computational overhead, to achieve the results shown in Tables 13.1 and 13.2, the identification of the best performing architecture of the MLP has required several experiments. We used architectures with both two layers and three layers and with a different number of neurons for each hidden layer. For the two-layer architecture, we used 10, 20, 30, 40, 50, 60, and 70 neurons in the hidden layer and for the three-layer architecture, we used 12, 16, 20, 24, and 28 neurons in the first hidden layer and 4, 6, 8, 10, and 12 in the second hidden layer (Corsini, Lazzerini, and Marcelloni, 2006). For each architecture, we trained the MLP and evaluated the percentage of point pairs with correct dissimilarity. Similarly, to determine the structure of the TS system, we executed the FCM algorithm with increasing values of the number *R* of clusters for different values of the fuzzification constant *m* and assessed the goodness of each resulting partition using the Xie–Beni index. Since the execution of the FCM is generally faster than the learning phase of an MLP, the determination of the TS structure is certainly quicker than the identification of the MLP architecture.

Once the structure has been identified, the TS requires the execution of the GA for tuning the membership functions and the consequent parameters so as to minimize the mean square error. As is well known in the literature, GAs are generally computationally heavy. We verified, however, that the GA used in this work performs a good optimization after a reasonable number of iterations. As an example, Figure 13.3 shows the percentage of correct dissimilarity values versus the number of generations in five trials with the Iris data-set and a training pool of 25 %. We can observe that a thousand generations allow the genetic algorithm to achieve a good approximation of the dissimilarity relation. If we consider that, as discussed in the next section, we can obtain good clustering results with 70–75 % of correct dissimilarity values, we can stop the genetic algorithm after a few hundreds of generations. This solution provides the further advantage of preventing overfitting problems, which may occur for small and unrepresentative training sets. The results shown in Tables 13.3 and 13.4 were obtained by stopping the GA after 2000 generations. Thus, we can conclude that the generation of the TS requires less effort than the generation of the MLP. Indeed, the determination of the best MLP network requires iteration through a number of MLP architectures with a different number of hidden layers and of nodes for each layer. The different networks

Table 13.3 Percentage of point pairs with correct dissimilarity values (TS system on the synthetic data-set).

Training pool	Number of rules	Correct dissimilarity values before GA	Correct dissimilarity values after GA
5%	10.5 ± 3.3	61.8% ± 6.7%	69.6% ± 7.6%
10%	10.1 ± 3.2	66.3% ± 3.8%	75.7% ± 5.2%
15%	11.7 ± 3.2	65.6% ± 6.8%	82.6% ± 4.2%
20%	12.6 ± 2.9	67.8% ± 3.0%	85.3% ± 3.9%
25%	14.2 ± 1.5	69.7% ± 2.8%	90.4% ± 3.5%

Table 13.4 Percentage of pattern pairs with correct dissimilarity values (TS system on Iris data-set).

Training pool	Number of Rules	Correct dissimilarity values before GA	Correct dissimilarity values after GA
5%	8.9 ± 2.4	80.0% ± 4.1%	80.5% ± 4.5%
10%	6.4 ± 1.6	82.7% ± 4.8%	87.7% ± 3.1%
15%	4.8 ± 0.6	80.8% ± 3.0%	90.2% ± 2.2%
20%	4.4 ± 0.8	78.5% ± 7.0%	91.6% ± 2.0%
25%	4.7 ± 0.5	80.7% ± 4.7%	91.6% ± 1.8%

are compared against accuracy. Each architecture has to be trained and this operation generally requires a considerable amount of time, depending on the number of layers and neurons for each layer. On the contrary, the determination of the TS structure requires iteration of the execution of FCM with different values of the number of clusters. The execution of FCM is certainly faster than the training of the MLP network and also the number of executions of FCM needed is generally smaller than the number of MLP networks to be trained. On the other hand, the generation of the TS systems requires the execution of the GA, which is quite time consuming. We have to consider, however, that the GA is executed just one time.

Finally, unlike the MLP, the TS allows describing the dissimilarity relation intuitively. Figure 13.4 shows the antecedent and the consequent of the rules that compose a TS model (after the optimization performed by GA) generated with the training pool composed of 15 % of the synthetic data-set. Here, we have associated a label with each fuzzy set based on the position of the fuzzy set in the universe of definition.

Since each rule defines its fuzzy sets, which may be different from the other rules, we used the following method to assign a meaningful linguistic label to each fuzzy set. First, we uniformly partition the universes of discourse into G triangular fuzzy sets (denoted as *reference terms* in the following) and associate a meaningful label with each fuzzy set. In the example, labels *L*, *ML*, *M*, *MH*, and *H* denote, respectively, *low*, *medium-low*, *medium*, *medium-high*, and *high* (see Figure 13.5). Then, we compute the similarity between each fuzzy set used in the rules and the reference terms using the formula

$$S_{i,e,f,l} = \frac{\left|A_{i,e,f} \cap P_{l,e,f}\right|}{\left|A_{i,e,f} \cup P_{l,e,f}\right|},$$

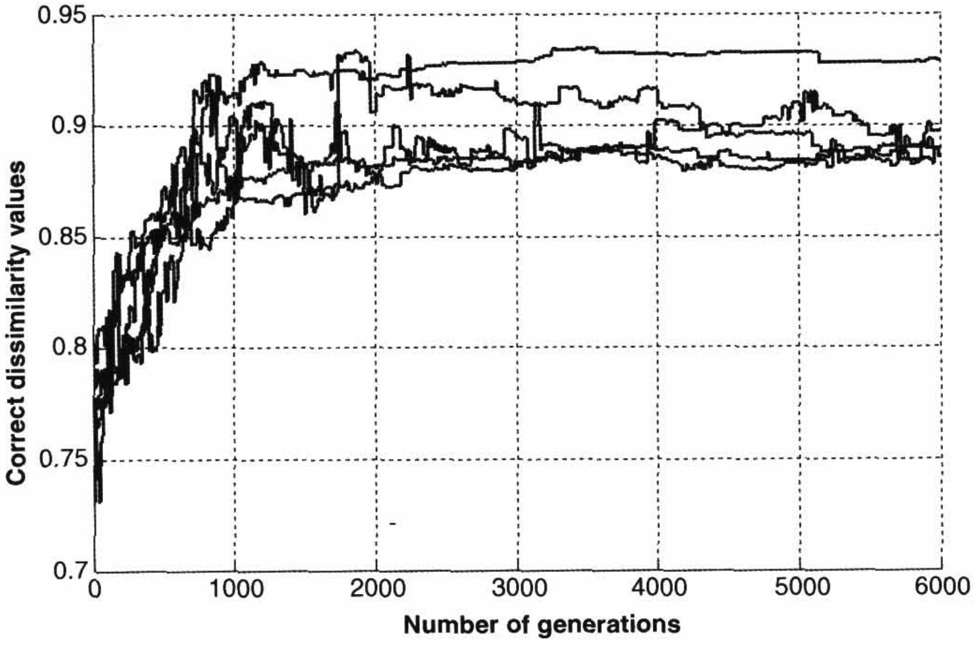

Figure 13.3 Percentage of correct dissimilarity values versus the number of generations.

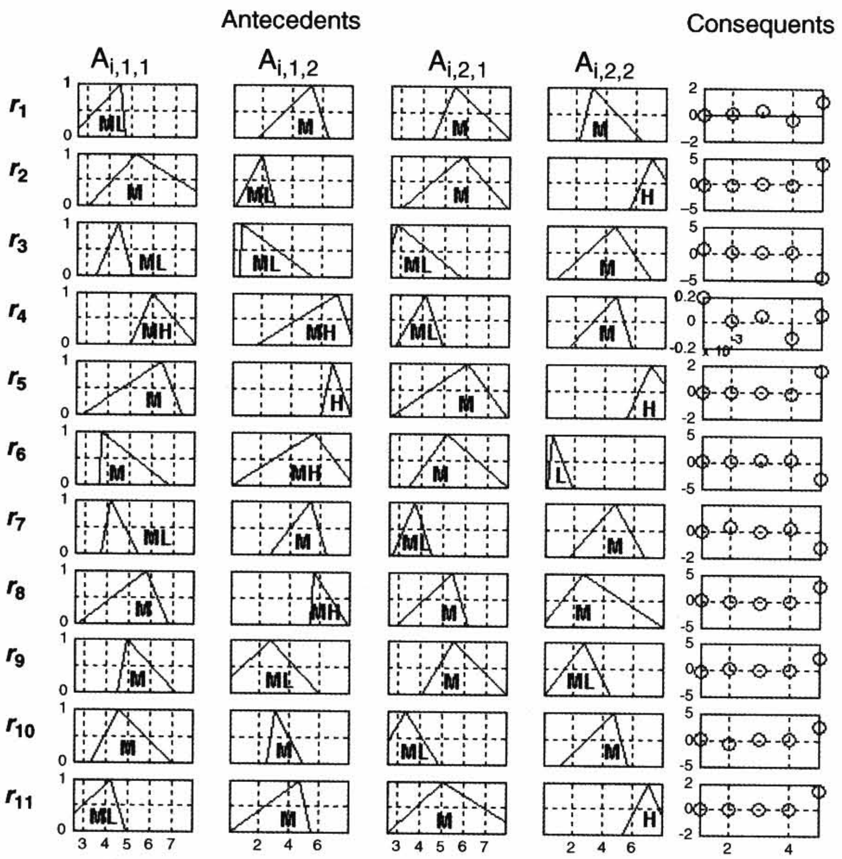

Figure 13.4 Rules after GA (synthetic data-set).

where $A_{i,e,f}$ and $P_{l,e,f}$ are, respectively, a fuzzy set and a reference term defined on the domain of input $X_{e,f}$ (Sugeno and Yasukawa, 1993). Finally, if there exists a value of $S_{i,e,f,l}$, with $l = 1..G$, larger than a fixed threshold τ, the reference term $P_{l,e,f}$ is associated with $A_{i,e,f}$ (if there exist more $P_{l,e,f}$ with $S_{i,e,f,l} > \tau$, then $A_{i,e,f}$ is associated with the $P_{l,e,f}$ corresponding to the highest $S_{i,e,f,l}$); otherwise, $A_{i,e,f}$ is added to the reference terms after associating a meaningful label with it. This association is carried out as follows. We first determine the reference term $P_{l,e,f}$ more similar to $A_{i,e,f}$. Then we generate four fuzzy sets. Two fuzzy sets are obtained by halving and doubling the support of $P_{l,e,f}$. We name the two fuzzy sets *very* $P_{l,e,f}$ and *more or less* $P_{l,e,f}$, respectively. The other two fuzzy sets are generated as $(P_{l,e,f} + P_{l,e,f-1})/2$, if $f \neq 0$, and $(P_{l,e,f} + P_{l,e,f+1})/2$, if $f \neq F$ (in the cases $f = 0$ and $f = F$, no fuzzy set is generated). The results of $(P_{l,e,f} + P_{l,e,f-1})/2$ and $(P_{l,e,f} + P_{l,e,f+1})/2$ are two triangular fuzzy sets defined as

$$\left(\frac{l_{l,e,f} + l_{l,e,f-1}}{2}, \frac{m_{l,e,f} + m_{l,e,f-1}}{2}, \frac{r_{l,e,f} + r_{l,e,f-1}}{2}\right)$$

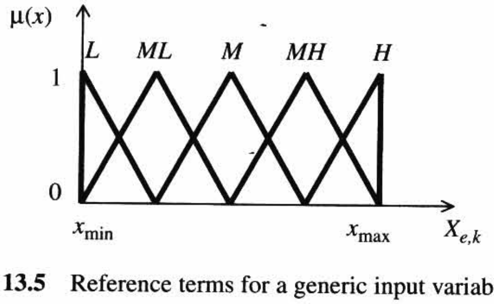

Figure 13.5 Reference terms for a generic input variable $X_{e,k}$.

Table 13.5 The qualitative model.

Rule	$\mathrm{X}_{1,1}$	$\mathrm{X}_{1,2}$	$\mathrm{X}_{2,1}$	$\mathrm{X}_{2,2}$	$\bar{d}_{i,j}$
r_1	*ML*	*M*	M	M	0.92
r_2	*M*	*ML*	*M*	H	0.73
r_3	*ML*	*ML*	*ML*	M	0.00
r_4	MH	MH	*ML*	M	1.00
r_5	*M*	*H*	*M*	H	0.00
r_6	*M*	MH	*M*	L	0.70
r_7	*ML*	*M*	*ML*	M	0.00
r_8	*M*	MH	*M*	M	0.73
r_9	*M*	*ML*	*M*	ML	0.32
r_{10}	*M*	M	*ML*	M	0.50
r_{11}	*ML*	*M*	M	H	0.38

and

$$\left(\frac{l_{l,e,f}+l_{l,e,f+1}}{2},\frac{m_{l,e,f}+m_{l,e,f+1}}{2},\frac{r_{l,e,f}+r_{l,e,f+1}}{2}\right),$$

respectively. We name these two fuzzy sets as $P_{l,e,f}-P_{l,e,f-1}$ and $P_{l,e,f}-P_{l,e,f+1}$, respectively. For instance, if $P_{l,e,f}=ML$, we obtain *very ML*, *more or less ML*, *L-ML*, and *ML-M*. Finally, we select the most similar among the four fuzzy sets to $A_{i,e,f}$ and assign the corresponding label to $A_{i,e,f}$. Once the fuzzy sets of all the rules have been examined, we again compute the similarity between each fuzzy set and the current reference terms in order to associate the most appropriate label with each fuzzy set. To generate the labels associated with the fuzzy sets shown in Figure 13.4, we have used a threshold $\tau = 0.5$. Note that no further reference term has been added.

To interpret the rules, we follow this procedure: for each pattern $\mathbf{z}_n = [\mathbf{x}_i, \mathbf{x}_j, d_{i,j}]$ in the training set, we feed as input the values of the coordinates of $\mathbf{x}_i$ and $\mathbf{x}_j$ to the TS model and measure the activation degree of each rule. We aim to discover whether there exists a relation between the activation of a rule and the values of dissimilarity. Table 13.5 shows, for each rule, the mean value $\bar{d}_{i,j}$ of dissimilarity $d_{i,j}$ of the pairs $(\mathbf{x}_i,\mathbf{x}_j)$ of patterns of the training set that activate this rule more than the other rules. This association between rules and dissimilarity values helps us interpret the meaning of the rules. From rule r_4, for instance, we can deduce that if the abscissa and the ordinate of the first point are, respectively, *MH* and *MH*, and the abscissa and the ordinate of the second point are, respectively, *ML* and *M*, then the dissimilarity is high. This rule can be easily verified by observing the data-set in Figure 13.2.

We note that rules are activated by pairs of points with either high or low dissimilarity. Indeed, the mean value of dissimilarity is close to 0 or 1. This means that the antecedents of the rules determine regions of the plane which contain points belonging either to the same class or to different classes. This observation confirms the results shown in Table 13.3: using 15 % of points in the training pool, we achieved 82.6 % of correct classification.

13.3 RELATIONAL CLUSTERING

Let $Q = [\mathbf{x}_1, \ldots, \mathbf{x}_M]$ be the data-set. Once the MLP has been trained or the TS has been generated and optimized, we compute the dissimilarity value between each possible pair $(\mathbf{x}_i, \mathbf{x}_j)$ of patterns in the data-set Q. Such dissimilarity values are provided as an $M \times M$ relation matrix $D = [d_{i,j}]$. The value $d_{i,j}$ represents the extent to which $\mathbf{x}_i$ is dissimilar to $\mathbf{x}_j$. Thus, the issue of partitioning patterns described through a set of meaningful features is transformed into the issue of partitioning patterns described through the values of their reciprocal relations. This issue is tackled by *relational clustering* in the literature. One of the most popular relational clustering algorithms is the sequential agglomerative hierarchical nonoverlapping clustering algorithm, which generates clusters by sequentially merging pairs of clusters which are the closest to each other at each step (Sneath and Sokal, 1973). Another well-known relational clustering algorithm partitions the data-set around a fixed number of representative objects, denoted *medoids*. The medoids are chosen from the data-set in such a way that the sum of the intra-cluster dissimilarity is minimized (Kaufman and Rousseeuw, 1987, 1990). Two versions of this algorithm aimed at handling large data-sets were proposed by Kaufman and Rousseeuw (1987) and by Ng and Han (1994), respectively. The aforementioned algorithms generate crisp clusters. As we are interested in finding a fuzzy partition of the data-set, in the following we discuss fuzzy relational clustering algorithms. The most popular examples of fuzzy relational clustering are the fuzzy nonmetric model (FNM, Roubens, 1978), the assignment prototype model (AP, Windham, 1985), the relational fuzzy C-means (RFCM, Hathaway, Davenport, and Bekdek, 1989), the non-Euclidean relational fuzzy C-means (NERFCM, Hathaway, and Bezdek, 1994), the fuzzy analysis (FANNY, Kaufman, and Rousseeuw, 1990), the fuzzy C-medoids (FCMdd, Krishnapuram, Joshi, Nasraoni, and Yi, 2001), and fuzzy relational data clustering (FRC, Davé, and Sen, 2002). All these algorithms assume (at least) that $D = [d_{i,j}]$ is a positive, irreflexive, and symmetric fuzzy square binary dissimilarity relation, i.e., $\forall i,j \in [1..M]$, $d_{i,j} \geq 0$, $d_{i,i} = 0$, and $d_{i,j} = d_{j,i}$. Unfortunately, the relation D produced by the two models may be neither irreflexive nor symmetric, thus making the existing fuzzy relational clustering algorithms theoretically not applicable to this relation. Actually, as shown by Corsini, Lazzerini, and Marcelloni (2002, 2004), these algorithms can be applied, but their convergence to a reasonable partition is not guaranteed (see, for instance, Corsini, Lazzerini, and Marcelloni, 2005). Indeed, in some data-sets used in our experiments, we observed that these algorithms tend to converge to a partition with completely superimposed fuzzy sets, that is, each object belongs to all clusters with equal membership value. To overcome this difficulty, we suggested transforming a relational clustering problem into an object clustering problem (Corsini, Lazzerini, and Marcelloni, 2005).

The basic idea of our approach arises from the definition of relational clustering algorithm itself: a relational clustering algorithm groups together objects that are "closely related" to each other, and "not so closely" related to objects in other clusters. Given a set of M patterns, and a square binary relation matrix $D = [d_{i,j}]$, with i,j in $[1..M]$, two patterns $\mathbf{x}_i$ and $\mathbf{x}_j$ should belong to the same cluster if the two vectors of the M strengths of relation between, respectively, $\mathbf{x}_i$ and all the patterns in the data-set Q, and $\mathbf{x}_j$ and all the patterns in Q, are close to each other. The two vectors correspond to the rows D_i and D_j of the matrix D. As the relation strengths are real numbers, the two vectors D_i and D_j can be represented as points in the metric space $\Re^M$. The closeness between D_i and D_j can be computed by using any metric defined in $\Re^M$; for instance, we could adopt the Euclidean or the Mahalanobis distance. Then, patterns $\mathbf{x}_i$ and $\mathbf{x}_j$ have to be inserted into the same cluster if and only if the distance between D_i and D_j is small (with respect to the distances between D_i (resp. D_j) and all the other row vectors). Based on this observation, the problem of partitioning M patterns, which are described by relational data, moves to the problem of partitioning M object data $D_k, k = 1..M$, in the metric space $\Re^M$. Thus, any clustering algorithm applicable to object data can be used. In particular, as proposed by Corsini, Lazzerini, and Marcelloni (2005, 2006), where the resulting clustering algorithm has been named ARCA, we can use the classical FCM. In the experiments, we used $m = 2$ and $\varepsilon = 0.001$, where ε is the maximum difference between corresponding membership values in two subsequent iterations. Moreover, we implemented the FCM algorithm in an efficient way in terms of both memory requirement and computation time, thanks to the use of the technique described by Kolen and Hutcheson (2002).

We executed ARCA with C ranging from two to five and chose the optimal number of clusters based on the Xie–Beni index. Tables 13.6 and 13.7 show the percentage of correctly classified points in the five experiments when $C = 2$ for the synthetic dataset and $C = 3$ for the Iris data-set, respectively. Here, the second and fourth columns indicate the percentage of correctly classified points for dissimilarity relations extracted by, respectively, the MLP and the TS, and the third and fifth columns the corresponding partition coefficients. The partition coefficient (PC) is defined as the average of the squared membership degrees. PC essentially measures the distance the partition U is from being crisp by assessing the fuzziness in the rows of U. PC varies in the interval $\left[\frac{1}{C},\ 1\right]$. Empirical studies show that maximizing PC leads to a good interpretation of data. Thus, the closer PC is to one, the better the partition is. As expected, the percentage of correctly classified points increases with the increase of

Table 13.6 Percentage of correctly classified points of the synthetic data-set in the five experiments.

Training pool	TS system: Correctly classified points	TS system: Partition coefficient	MLP system: Correctly classified points	MLP system: Partition coefficient
5%	84.4% ± 6.5%	0.84 ± 0.07	87.1% ± 2.8%	0.83 ± 0.10
10%	87.5% ± 5.4%	0.89 ± 0.05	88.9% ± 2.8%	0.86 ± 0.07
15%	93.7% ± 3.5%	0.90 ± 0.04	93.8% ± 1.5%	0.88 ± 0.04
20%	94.1% ± 2.9%	0.92 ± 0.02	94.6% ± 1.5%	0.91 ± 0.03
25%	97.0% ± 1.8%	0.94 ± 0.03	97.3% ± 1.3%	0.92 ± 0.02

points in the training pool. Just for small percentages of points in the training pool, the combinations MLP–ARCA and TS–ARCA are able to trace the boundaries of the classes conveniently. The quality of the approximation improves when the points of the training pool are a significant sample of the overall data-set. The tables show that the class shape is almost correctly identified just with 5 % of the points of the data-set. Note that, as reported in Tables 13.1–13.4, the MLP and the TS are able to output only 70.1 % and 69.6 % of correct dissimilarity values for the synthetic data-set, and 81.2 % and 80.5 % for the Iris data-set, when trained with training pools containing the same percentage of points. Finally, the high values of the partition coefficient highlight that the partition determined by the relational clustering algorithm is quite good.

Tables 13.8 and 13.9 show the number of clusters (in the form (mean ± standard deviation)) in the five experiments for, respectively, the synthetic and Iris data-sets when using the TS. It can be observed that the percentage of trials in which the number of clusters is equal to the number of classes increases very quickly (up to 100 %) with the increase of the percentage of points in the training pool.

As shown in Tables 13.6 and 13.7, ARCA achieves very interesting results and is characterized by a certain stability. As comparison, we applied some of the most popular fuzzy clustering algorithms to the same relations extracted by the TS and we observed a strong dependence of the results on the initial partition and on the fuzzification constant m. In several trials, we found out that the algorithms converge to a partition composed completely superimposed fuzzy sets. Anyway, since ARCA adopts the Euclidean distance, it suffers from the well-known curse of dimensionality problems: when the dimensionality increases, distances between points become relatively uniform, thus making the identification of clusters practically impossible. Actually, the curse of dimensionality problems could arise because the dimension of the space is equal to the number of objects in the data-set. Thus, for very large data-sets, we should adopt distance functions more suitable for high-dimensional spaces in place of the Euclidean distance. We did not adopt this solution in the examples simply because it was not strictly necessary. We performed, however, some experiments with the version of FCM proposed by Klawonn and Keller (1999), which adopts the cosine distance in place of the Euclidean distance. We used large dissimilarity relations

Table 13.7 Percentage of correctly classified points of the Iris data-set in the five experiments.

Training pool	TS system: Correctly classified points	TS system: Partition coefficient	MLP system: Correctly classified points	MLP system: Partition coefficient
5%	89.8% ± 5.5%	0.74 ± 0.08	90.8% ± 4.4%	0.78 ± 0.09
10%	92.5% ± 4.6%	0.86 ± 0.04	91.3% ± 3.7%	0.91 ± 0.07
15%	94.4% ± 3.0%	0.91 ± 0.04	94.1% ± 2.4%	0.88 ± 0.05
20%	95.2% ± 2.1%	0.93 ± 0.04	95.6% ± 2.7%	0.90 ± 0.05
25%	95.8% ± 1.6%	0.92 ± 0.03	96.0% ± 1.3%	0.91 ± 0.04

Table 13.8 Number of clusters in the five experiments (synthetic data-set).

Training pool	Number of clusters	Percentage of trials with number of clusters equal to number of classes
5%	2.1 ± 0.3	90%
10%	2.0 ± 0.0	100%
15%	2.0 ± 0.0	100%
20%	2.0 ± 0.0	100%
25%	2.0 ± 0.0	100%

(10 000 × 10 000) created artificially. We verified that, also in this case, the results obtained by the two versions of FCM are comparable. For instance, we generated a data-set composed of three clusters using uniform random distribution of points over three nonoverlapping circles centered in (700, 400), (400, 900), and (1000, 900), with radius equal to 530. The three clusters are composed of 3606, 3733, and 3606 points, respectively. Then, we generated a dissimilarity relation (10 945 × 10 945) using the Euclidean distance. We executed the FCM algorithm with the fuzzification coefficient m and the termination error ε equal to 1.4 and 0.01, respectively. We obtained 100 % classification rate for both the versions of FCM, with a partition coefficient equal to 0.95 and 0.98 for the version with the Euclidean distance and for the version with the cosine distance, respectively.

To further verify the validity of ARCA, we applied a well-known density-based algorithm, named OPTICS (Ankerst, Breuing, Kriegel, and Sander, 1999), to the dissimilarity relation produced by the TS. OPTICS is an extension of DBSCAN (Ester, Kriegel, Sander, and Xu, 1996), one of the best known density-based algorithms. DBSCAN defines a cluster to be a maximum set of density-connected points, which means that every core point in a cluster must have at least a minimum number of points (*MinPts*) within a given radius (*Eps*). DBSCAN assumes that all points within genuine clusters can be reached from one another by traversing a path of density-connected points and that points across different clusters cannot. DBSCAN can find arbitrarily shaped clusters if the cluster density can be determined beforehand and the cluster density is uniform. DBSCAN is very sensitive to the selection of *MinPts* and *Eps*. OPTICS reduces this sensitivity by limiting it to *MinPts*. To perform clustering, density-based algorithms assume that points within clusters are "density reachable" and points across different clusters are not. Obviously, the cluster shape depends on the concept of "density reachable" that, in its turn, depends on the definition of dissimilarity. Thus, we cannot consider adopting density-based algorithms to solve the initial problem, that is, to determine the most suitable dissimilarity measure and therefore the most suitable cluster shape. As an example, let us consider the data-set shown in Figure 13.6 (XOR problem). The points belong to two different classes: each class is composed of two compact clusters located on the opposite corners of a square, respectively.

A density-based clustering process performed in the feature space is not able to detect the correct structure, unless a specific proximity measure is defined. Indeed, the OPTICS algorithm finds four different clusters, i.e., it achieves 50 % classification rate. Figure 13.7 shows the output of the OPTICS algorithm.

Table 13.9 Number of clusters in the five experiments (Iris data-set).

Training pool	Number of clusters	Percentage of trials with number of clusters equal to number of classes
5%	2.5 ± 0.5	50%
10%	2.8 ± 0.6	60%
15%	3.3 ± 0.5	70%
20%	2.9 ± 0.3	100%
25%	3.0 ± 0.0	100%

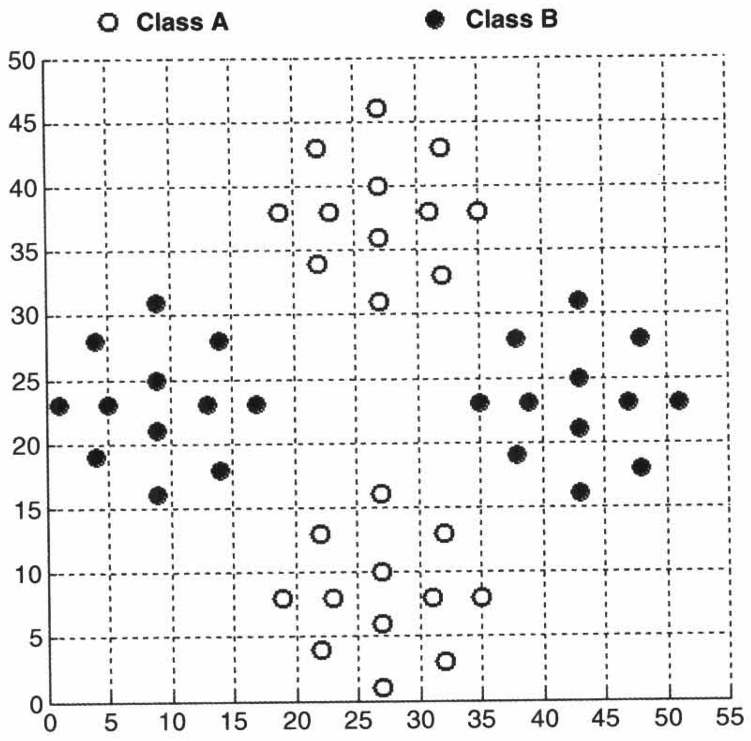

Figure 13.6 The XOR problem data-set.

On the contrary, our approach achieves 96.7 % ± 2.6 % classification rate using 20 % of points as training pool. Furthermore, we achieve a better classification rate than OPTICS even with 5 % points in the training pool. This example shows that our approach does not depend on the distribution of data and therefore on the concept of spatial density. Our method is certainly more time-consuming, but it has been introduced to solve clustering problems that are not automatically solvable with density-based clustering algorithms.

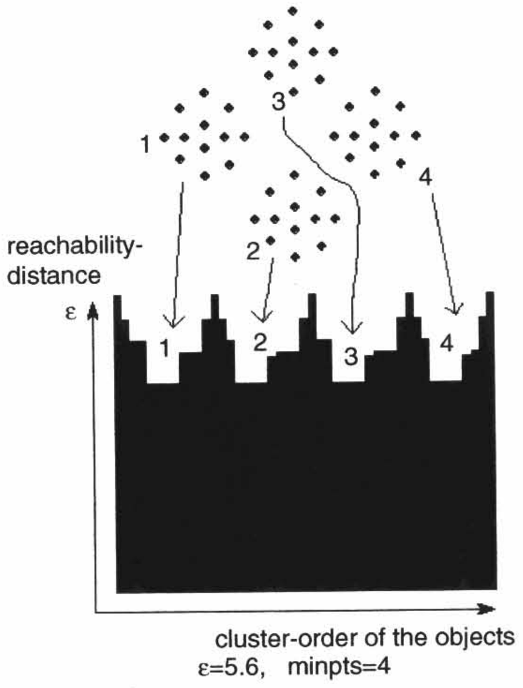

Figure 13.7 Output of the OPTICS algorithm.

On the other hand, since some density-based algorithms do not require distances but rather generic dissimilarity measures to determine the closeness of points, we can adopt OPTICS to cluster data described by the dissimilarity relations produced by the MLP and the TS. We performed different experiments and verified that the performance of OPTICS and ARCA are quite similar. In the XOR example, for instance, OPTICS achieves 95.6 % ± 2.3 % classification rate using 20 % of points as training pool.

13.4 EXPERIMENTAL RESULTS

In this section, we briefly discuss some results obtained by applying the combination TS–ARCA to some well-known data-sets provided by the University of California (UCI, 2006), namely the Wisconsin Breast Cancer (WBC) data-set, the wine data-set, and the Haberman's Survival (HS) data-set. We discuss only the TS approach because, as shown in Section 13.2, it is characterized by more interesting features.

The WBC data-set consists of 699 patterns belonging to two classes: 458 patterns are members of the "benign" class and the other 241 patterns are members of the "malignant" class. Each pattern is described by nine features: clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. Since 16 patterns have a missing value, we decided to use only 683 patterns in our experiments.

The Wine data-set contains the chemical analysis of 178 wines grown in the same region in Italy but derived from three different cultivars, which represent the classes. As known in the literature (Setnes and Roubos, 2000), only some of the 13 features are effective for classification. Thus, we performed a feature selection based on the correlation between classes and features, and selected the following four features: total phenols, flavanoids, color intensity, and OD280/OD315 of diluted wines.

The HS data-set contains 306 cases from a study on the survival of patients who had undergone surgery for breast cancer. The three attributes represent the age of the patient, the year of the operation, the number of the positive axillary nodes. The two classes represent the survival status after 5 years. In this data-set, features have a low correlation with classes and therefore the data-set is quite difficult for clustering algorithms.

We carried out the same experiments described in previous sections. Tables 13.10–13.12 show the percentage of correctly classified points of the WBC, Wine, and HS data sets in the five experiments. We

Table 13.10 Percentage of correctly classified points of the WBC data-set in the five experiments.

Training pool	Correctly classified points	Partition coefficient
5%	95.9% ± 0.5%	0.94 ± 0.03
10%	96.1% ± 0.3%	0.97 ± 0.00
15%	96.8% ± 0.7%	0.96 ± 0.00
20%	96.8% ± 0.3%	0.95 ± 0.01
25%	97.1% ± 0.1%	0.96 ± 0.01

Table 13.11 Percentage of correctly classified points of the wine data-set in the five experiments.

Training pool	Correctly classified points	Partition coefficient
5%	83.7% ± 4.6%	0.84 ± 0.05
10%	85.5% ± 3.4%	0.88 ± 0.03
15%	89.7% ± 3.2%	0.91 ± 0.02
20%	91.3% ± 3.2%	0.93 ± 0.02
25%	94.1% ± 1.4%	0.95 ± 0.02

Table 13.12 Percentage of correctly classified points of the HS data-set in the five experiments.

Training pool	Correctly classified points	Partition coefficient
5%	87.0% ± 2.9%	0.80 ± 0.02
10%	88.1% ± 3.0%	0.80 ± 0.05
15%	90.0% ± 3.4%	0.83 ± 0.04
20%	88.7% ± 4.7%	0.84 ± 0.05
25%	90.6% ± 5.0%	0.83 ± 0.04

can observe that the percentages of correct classifications are just quite high with training pools composed of only 5 % of patterns. These results compare favorably with several classification techniques proposed in the literature. Since our method is not a classification method because we do not suppose to know the labels of the classes, but rather some similarities between patterns, the results prove the effectiveness of the combination of learning algorithms and relational clustering algorithms.

13.5 CONCLUSIONS

Object clustering algorithms generally partition a data-set based on a dissimilarity measure expressed in terms of some distance. When the data distribution is irregular, for instance in image segmentation and pattern recognition where the nature of dissimilarity is conceptual rather than metric, distance functions may fail to drive the clustering algorithm correctly. Thus, the dissimilarity measure should be adapted to the specific data-set. For this reason, we have proposed extracting the dissimilarity relation directly from a few pairs of patterns of the data-set with known dissimilarity values. To this aim, we have used two different techniques: a multilayer perceptron with supervised learning and a Takagi–Sugeno fuzzy system. We have discussed and compared the two approaches with respect to generalization capabilities, computational overhead, and capability of explaining intuitively the dissimilarity relation. We have shown that the TS approach provides better characteristics than the MLP approach.

Once the dissimilarity relation has been generated, the partitioning of the data-set is performed by a fuzzy relational clustering algorithm, denoted ARCA, recently proposed by the authors. Unlike well-known relational clustering algorithms, this algorithm can manage the dissimilarity relations generated by the MLP and the TS, which are neither irreflexive nor symmetric. The experiments performed on some real data-sets have shown the good qualities of our approach. In particular, we have observed that just using a significantly low percentage of known dissimilarities, our method is able to cluster the data-sets almost correctly.

REFERENCES

Abonyi, J., Babuška, R., and Szeifert, F. (2002) 'Modified Gath–Geva fuzzy clustering for identification of Takagi–Sugeno fuzzy models', *IEEE Trans. on Systems, Man and Cybernetics-Part B*, **32**, 612–621.

Angelov, P.P. and Filev, D.P. (2004) 'An approach to online identification of takagi-sugeno fuzzy models'. *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, **34**, 484–498.

Ankerst, M., Breunig, M., Kriegel, H.-P. and Sander, J. (1999) 'Optics: ordering points to identify the clustering structure'. *Proc. ACM Int. Conf. Management of Data* (ACM SIGMOD '99), Philadelphia, PA, USA, pp. 49–60.

Babuška, R. (1996) *Fuzzy Modeling and Identification* PhD dissertation, Delft University of Technology, Delft, The Netherlands.

Bezdek, J.C. (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York, NY, USA, Plenum.

Bezdek, J.C., Keller, J., Krisnapuram, R. and Pal, N.R. (1999) *Fuzzy Model and Algorithms for Pattern Recognition and Image Processing*. Kluwer Academic Publishing, Boston, MA, USA.

Chang, H. and Yeung, D.Y. (2005) 'Semi-supervised metric learning by kernel matrix adaptation', *Proc. International Conference on Machine Learning and Cybernetics* (ICMLC '05), Guangzhou, China, Vol. 5, pp. 3210–3215.

Corsini, P., Lazzerini, B. and Marcelloni, F. (2002) 'Clustering based on a dissimilarity measure derived from data'. *Proc. Knowledge-Based Intelligent Information and Engineering Systems* (KES '02), Crema, Italy, 885–889.

Corsini, P., Lazzerini, B. and Marcelloni, F. (2004) 'A fuzzy relational clustering algorithm based on a dissimilarity measure extracted from data', *IEEE Trans. on Systems, Man, and Cybernetics – Part B*, **34**, pp. 775–782.

Corsini, P., Lazzerini, B. and Marcelloni, F. (2005) 'A new fuzzy relational clustering algorithm based on the fuzzy C-means algorithm'. *Soft Computing*, **9**, 439–447.

Corsini, P., Lazzerini, B. and Marcelloni, F. (2006) 'Combining supervised and unsupervised learning for data clustering'. *Neural Computing and Applications*, **15**, 289–297.

Davé R.N. and Sen, S. (2002) 'Robust fuzzy clustering of relational data'. *IEEE Trans. on Fuzzy Systems*, **10**, 713–727.

Ester, M., Kriegel, H.-P., Sander, J. and Xu, X. (1996) 'A density-based algorithm for discovering clusters in large spatial databases'. *Proc. Int. Conf. Knowledge Discovery and Data Mining* (KDD '96), Portland, OR, USA, pp. 226–231.

Gath, I. and Geva, A.B. (1989) 'Unsupervised optimal fuzzy clustering'. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **11**, 773–781.

Gustafson, D.E. and Kessel, W.C. (1979) 'Fuzzy clustering with fuzzy covariance matrix'. *Advances in Fuzzy Set Theory and Applications* (M.M. Gupta, R.K. Ragade and R.R Yager eds), North Holland, Amsterdam, The Netherlands, pp. 605–620.

Hathaway, R.J. and Bezdek, J.C. (1994) 'NERF C-means: non-Euclidean relational fuzzy clustering'. *Pattern Recognition*, **27**, 429–437.

Hathaway, R.J., Davenport, J.W. and Bezdek, J.C. (1989), 'Relational duals of the c-means clustering algorithms', *Pattern Recognition*, **22**, 205–212.

Haykin, S. (1999) *Neural Networks: A Comprehensive Foundation* (2nd Edition). Prentice Hall.

Hertz, T., Bar-Hillel, A. and Weinshall, D. (2004) 'Learning distance functions for image retrieval'. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE CVPR '04), Washington, DC, USA, Vol. II, pp. 570–577.

Jacobs, D.W., Weinshall, D. and Gdalyahu, Y. (2000) 'Classification with nonmetric distances: image retrieval and class representation'. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **22**, 583–600.

Jain, A.J. and Flynn, P.J. (eds) (1993) *Three dimensional Object Recognition Systems*. Elsevier Science, New York, USA.

Jain, A.K., Murty, M.N. and Flynn, P.J. (1999) 'Data clustering: a review'. *ACM Computing Surveys*, **31**, 265–323.

Jarvis, R.A. and Patrick, E.A. (1973) 'Clustering using a similarity method based on shared near neighbors'. *IEEE Trans. on Computers*, **C-22**, 1025–1034.

Kamgar-Parsi, B. and Jain, A.K. (1999) 'Automatic aircraft recognition: toward using human similarity measure in a recognition system'. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE CVPR '99), Fort Collins, CO, USA, pp. 268–273.

Kaufman, J. and Rousseeuw, P.J. (1987) 'Clustering by means of medoids'. *Statistical Data Analysis Based on the L1 Norm* (Y. Dodge ed.), North Holland/Elsevier, Amsterdam, The Netherlands, pp. 405–416.

Kaufman, J. and Rousseeuw, P.J. (1990) '*Findings Groups in Data: An Introduction to Cluster Analysis* John Wiley & Sons, Ltd, Brussels, Belgium.

Klawonn, F. and Keller, A. (1999) 'Fuzzy clustering based on modified distance measures'. *Advances in Intelligent Data Analysis* (D.J. Hand, J.N. Kok and M.R. Berthold, eds), Springer, Berlin, Germany, pp. 291–301.

Kolen, J.F. and Hutcheson, T. (2002) 'Reducing the time complexity of the fuzzy C-means algorithm'. *IEEE Trans. on Fuzzy Systems*, **10**, 263–267.

Krishnapuram, R., Joshi, A., Nasraoui, O. and Yi, L. (2001) 'Low-complexity fuzzy relational clustering algorithms for web mining'. *IEEE Trans. on Fuzzy Systems*, **9**, 595–607.

Lange, T., Law, M.H.C., Jain, A.K. and Buhmann, J.M. (2005), 'Learning with constrained and unlabelled data'. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE CVPR '05), San Diego, CA, USA, pp. 731–738.

Latecki, L.J. and Lakamper, R. (2000) 'Shape similarity measure based on correspondence of visual parts'. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **22**, 1185–1190.

Law, M.H., Topchy, A. and Jain, A.K. (2005) 'Model-based clustering with probabilistic constraints'. *Proc. SIAM International Conference on Data Mining* (SDM '05), Newport Beach, CA, USA, pp. 641–645.

Makrogiannis, S., Economou, G. and Fotopoulos, S. (2005) 'A region dissimilarity relation that combines feature-space and spatial information for color image segmentation'. *IEEE Trans. on Systems, Man and Cybernetics - Part B*, **35**, 44–53.

Michalski, R., Stepp, R.E. and Diday, E. (1983) 'Automated construction of classifications: conceptual clustering versus numerical taxonomy'. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **5**, 396–409.

Ng, R.T. and Han, J. (1994) 'Efficient and effective clustering methods for spatial data mining'. *Proc. International Conference on Very Large Data Bases* (VLDB '94), Santiago, Chile, pp. 144–155.

Pal, N.R. and Bezdek, J.C. (1995) 'On cluster validity for the fuzzy C-means model'. *IEEE Trans. on Fuzzy Systems*, **3**, 370–379.

Pedrycz, W. (2005) *Knowledge-based clustering: from Data to Information Granules*. John Wiley and Sons, Inc., Hoboken, NJ, USA.

Pedrycz, W. and Waletzky, J. (1997) 'Fuzzy clustering with partial supervision'. *IEEE Trans. on Systems, Man and Cybernetics – Part B: Cybernetics*, **27**, 787–795.

Pedrycz, W., Loia, V. and Senatore, S. (2004) 'P-FCM: a proximity-based fuzzy clustering'. *Fuzzy Sets and Systems*, **148**, 21–41.

Pedrycz, W. *et al.* (2001) 'Expressing similarity in software engineering: a neural model'. *Proc. Second International Workshop on Soft Computing Applied to Software Engineering* (SCASE '01), Enschede, The Netherlands.

Rezaee, M.R., Lelieveldt, B.P.F. and Reiber, J.H.C. (1998) A new cluster validity index for the fuzzy *C*-mean. *Pattern Recognition Letters*, **18**, 237–246.

Roubens, M. (1978) 'Pattern classification problems and fuzzy sets'. *Fuzzy Sets and Systems*, **1**, 239–253.

Santini, S. and Jain, R. (1999) 'Similarity measures'. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **21**, 871–883.

Setnes, M. and Roubos, H. (2000) 'GA-fuzzy modeling and classification: complexity and performance'. *IEEE Trans. on Fuzzy Systems*, **8**, 509–522.

Sneath, P.H. and Sokal, R.R. (1973) *Numerical Taxonomy – The Principles and Practice of Numerical Classification*. W.H. Freeman & Co., San Francisco, CA, USA.

Su, M.C. and Chou, C.H. (2001) 'A Modified version of the *K*-means algorithm with a distance based on cluster symmetry'. *IEEE Trans. of Pattern Analysis and Machine Intelligence*, **23**, 674–680.

Sugeno, M. and Yasukawa, T. (1993) 'A fuzzy logic-based approach to qualitative modeling'. *IEEE Trans. on Fuzzy Systems*, **1**, 7–31.

Takagi, T. and Sugeno, M. (1985) 'Fuzzy identification of systems and its application to modeling and control'. *IEEE Trans. on Systems, Man, and Cybernetics*, **15**, 116–132.

Tsang, I.W., Cheung, P.M. and Kwok, J.T. (2005) 'Kernel relevant component analysis for distance metric learning'. *Proc. IEEE International Joint Conference on Neural Networks* (IEEE IJCNN '05), Montréal, QB, Canada, **2**, pp. 954–959.

UCI Machine Learning Database Repository, http://www.ics.uci.edu/~mlearn/MLSummary.html, 2006.

Valentin, D., Abdi, H., O'Toole, A.J. and Cottrell, G.W. (1994) 'Connectionist models of face processing: a survey'. *Pattern Recognition*, **27**, 1208–1230.

Windham, M.P. (1985) 'Numerical classification of proximity data with assignment measures'. *Journal of Classification*, **2**, 157–172.

Wright, A.H. (1991) 'Genetic algorithms for real parameter optimization', *Foundations of Genetic Algorithms*, (G.J. Rawlins ed.); San Mateo, CA, USA, Morgan Kaufmann, pp. 205–218.

Xie, X.L. and Beni, G. (1991) 'A validity measure for fuzzy clustering'. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **13**, 841–847.

Xing, E.P., Ng, A.Y., Jordan, M.I. and Russell, S. 2003) 'Distance metric learning, with application to clustering with side-information'. *Advances in Neural Information Processing Systems* (S. Thrun, Becker, S. and K. Obermayer, (eds), Cambridge, MA, USA, **15**, pp. 505–512.

Xu, R. and Wunsch, D. II (2005) 'Survey of Clustering Algorithms'. *IEEE Trans. on Neural Networks*, **16**, 645–678.

Yang, M.S. and Wu, K.L. (2004) 'A similarity-based robust clustering method'. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **26**, 434–448.