Random-Sets Clustering Based on Set-Similarities

J. Nuñez Garcia and O. Wolkenhauer
Control Systems Centre, UMIST
javier@csc.umist.ac.uk
olaf.wolkenhauer@umist.ac.uk
www.csc.umist.ac.uk

Control Systems Centre Report No. 878

March 1999

Abstract

The clustering algorithm described in this paper can be classified into the family of agglomerative hierarchic algorithms, i.e. the two most similar sets, agreeing with a predetermined criterion, are joined to form a bigger set at every step. After as many steps as there are initial sets minus one, we obtain a unique final set or cluster which groups all the initial sets. The algorithm can be stopped whenever the desired number of final clusters has been achieved. The fundamental objective is to investigate the geometric properties of a group of data. To achieve this, the algorithm must be able to generate clusters which fit the shape of the data. Through the features of these clusters (explained by some parameters) we will obtain knowledge about the form of the data in the space. A similarity measure based on the orientation and common information of two random sets will be used to determine the degree of closeness between them.

1 Introduction

We have a sample of data (size N) in a available n-dimensional space. Our aim is to look for a reduced group of clusters that characterise the shape of the data.

A point in the n-dimensional space has neither shape, size nor orientation. This can raise a problem at the beginning if we want to perform a criterion as described in the abstract. Taking every point as an initial set we will not be able to apply a similarity measure based on their shape or orientation. We propose as a solution that the initial sets are constructed by grouping the nearest neighbours for every point by using the Euclidean distance, so that we have as many initial sets as data there are, with few data inside. The choice of the number of neighbours will be discussed in another section of the paper. With this resolution the data is partitioned into N overlapping initial clusters (random sets) where two or more points which are very close will have a high number of nearest neighbours in common, sometimes all of them if they are close enough. In this last case we will maintain only one of these similar sets. Note that sets with the same data would have the same features in size, orientation, form, etc., i.e. they are the perfect candidates for merging.

The next step is to find a way to measure the shape of a group of data. A common approach is to describe a set of data, within a n-dimensional space, by hyperellipsoids. The mean is the centre and the eigenvectors of their covariance matrix specify the directions of the axes, The lengths of the axes are the square root of the corresponding eigenvalues. These hyperellipsoids will provide a way of characterising the the position, size and the orientation of a set of data (see figure 1). This knowledge (given through the parameters of the hyperellipsoids) must be taken into consideration to determine the *similarity* between sets.

The data points belonging to the intersection of two random sets will be understood as common information between them and will play a relevant role in measuring the proximity of two neigbouring random sets.

The similarity measure will then be formed by two concepts, the *orientation* of the random sets (described by their corresponding hyperellipsoids) and the *common information* mentioned before.

A set of data (as in figure 1, for the source see Appendix A) can present a general shape (in figure 1 there are three linear elements) and some local structures from groups of data which form isolated parts. In the beginning, our initial sets, or better the corresponding hyperellipsoids must be small enough to follow the general shape of the data, i.e. they should be formed by a small number of neighbours. For this reason these hyperellipsoids might follow other local structures and some of them may have a totally different orientation to the general shape, even if there is a lot of common information between close sets. Because of this at the first stages large differences in the orientation of the sets must not be an obstacle for joining since the general shape will still be preserved. The opposite happens at the final steps were the common information between sets is scarce (we will see forward) and it is important that the orientation of the sets follow the general shape of the data. All this implies that the combination between *orientation* and *common information* must be dynamic throughout the execution of the algorithm. We get this by introducing a function as a variable parameter at every step in the sum.

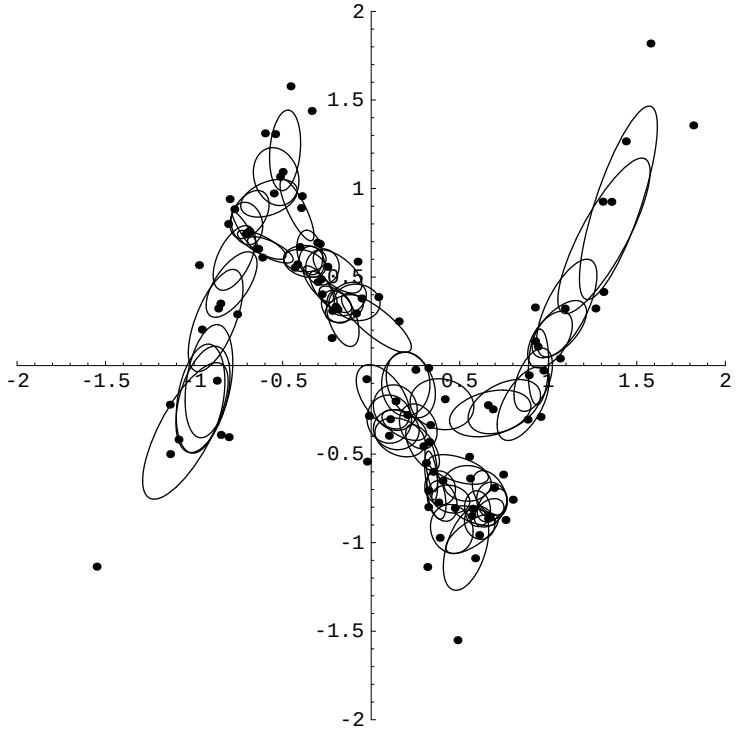

Figure 1: Initial random sets and their corresponding hyperellipsoids for the eight nearest neighbours.

2 Direction of a hyperellipsoid in a n-dimensional space

In this section we will discuss how the direction of two random sets can be compared by using their corresponding hyperellipsoids which are calculated by using the principal components method. Its aim is to describe the variation, among n correlated variables, in a set of multivariate data in terms of n uncorrelated (orthogonal) variables (principal components) each of which are a linear combination of the original variables, i.e. an orthogonal rotation in an n-space. The principal components are derived in decreasing order, so the first component explains the variation of the data as much as possible. The second explains the remaining variation as much as possible and so on. The usual objective is to look for a reduction in the dimension of the space in which the data is represented by eliminating the principal components which account for a few variations of the data. It can be useful in simplifying posterior analysis.

We will exploit this technique in our context. The idea is as follows; we will not take into consideration the smallest axis of the hyperellipsoids (the principal component corresponding with the smallest eigenvalue

of the covariance matrix). It should be no problem if we consider that it describes a few variations of the data. Then, we want to compare the direction of two $n-1$ dimensional hyperellipsoids or in other words, two hyperplanes in an $n-1$ dimensional space. The easier way is by comparing the vector as a result of the vectorial product among the $n-1$ directional vectors of the hyperplanes ($n-1$ principal components). Note how this vector is precisely the principal component which we did not take into consideration. For example in a 2-dimensional space, we would be comparing two lines. Checking the relative position in the space can be done by calculating the cosine of the angle of two perpendicular vectors at every line. They will be parallel or orthogonal depending on how close the cosine is to 0 or 90 degrees. These two vectors are the second principal component. The same argument can be extended for more dimensions. In three dimensions we would be comparing the direction of two planes by checking what the cosine of a normal vectors is to each plane. This is the third component.

Note that the cosine of the angle of two vectors verifies the properties of *similarity*. See Appendix A for the definition of similarity measure.

3 Common information as a measure of proximity

The angle formed by two eigenvectors cannot be considered as a sufficient criterion since two parallel sets (with the same orientation) which are far away from each other might be joined before two other almost parallel sets which are very close together. In the first case the result will be a set with very differing groups and the second gives the result of a more compact set. If we visualize the sets as hyperellipsoids, in the first case the central part of the hyperellipsoid will have no data, and this is not good if we are trying to fit the shape of the data. While in the second case, the empty area between both groups disappears as the closeness between both sets increases. Then, we need something to show how close the two sets are. We have multiple choices available among different *distances*, *similarities* and *dissimilarities* applicable to sets of data. We propose a similarity measure based on the *common points* of a pair of sets, defined as:

$$cp(A,B) = \frac{\text{Number of elements of } (A \cap B)}{\text{Min(Number of elements of A, Number of elements of B)}} \tag{1}$$

Note that the equation (1) satisfies the requirements of a similarity (see Appendix A for the definition of similarity measure). Also note that $cp(A,B) = 1 \Rightarrow cos(a,b) = 0$, where a and b are the eigenvectors corresponding to the smallest eigenvalues of the covariance matrix of the data of A and B respectively. This is clear since $cp(A,B) = 1 \Rightarrow A{=}B$.

The measure cp could be understood in the same way when we use the volume of the intersection of the two hyperellipsoids, generated from two sets, as measure of closeness between them. A problem arises since two hyperellipsoids can influence each other even when the volume of the intersection between both is zero. This is precisely because both sets have some data in common. So, we can find cases where the volume is zero but there is a weak influence between them, and cases where the volume is also zero and there is no influence (see figure 2). For this reason the use of the volume as a measure of closeness may produce wrong joins aside from the difficulty of its calculation.

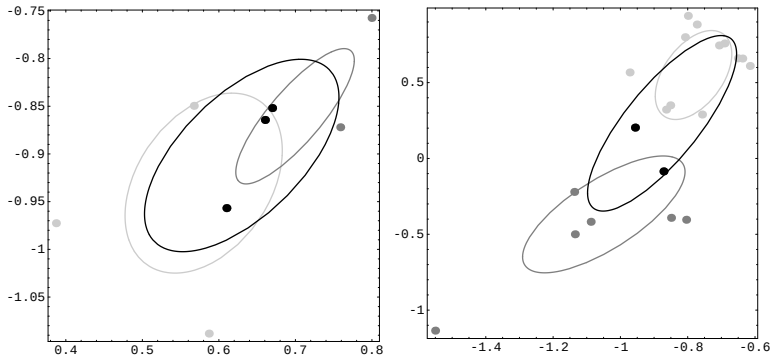

Figure 2: Common data between sets(black) and their ellipsoids.

4 Two similarity measures in one

Now our attention will concentrate on how to combine both similarity measures in only one measure which serves well to achieve our aim. By

$$s(A,B) = \mathcal{B} \cdot cp(A,B) + (1-\mathcal{B}) \cdot cos(a,b), \quad \mathcal{B} \in [0,1] \tag{2}$$

we obtain a similarity which can give similar or different weight to cp and cos.

Let us prove that s is a similarity measure(see Appendix A):

$$0 \leq s(A,B) \leq \mathcal{B} \cdot cp(A,B) + (1-\mathcal{B}) \cdot cos(a,b) \leq 1 \Rightarrow 0 \leq s(A,B) \leq 1, \quad \forall A,B \in U$$

$$s(A,A) = \mathcal{B} \cdot cp(A,A) + (1-\mathcal{B}) \cdot cos(a,a) = \mathcal{B} + (1-\mathcal{B}) = 1, \quad \forall A \in U$$

$$s(A,B) = \mathcal{B} \cdot cp(A,B) + (1-\mathcal{B}) \cdot cos(a,b) = \mathcal{B} \cdot cp(B,A) + (1-\mathcal{B}) \cdot cos(b,a) = s(B,A), \quad \forall A,B \in U$$

$$s(A,B) = \mathcal{B} \cdot cp(A,B) + (1-\mathcal{B}) \cdot cos(a,b) = 1 \Rightarrow$$

$$\mathcal{B} = \frac{cos(a,b)-1}{cos(a,b)-cp(A,B)} \Rightarrow \frac{cos(a,b)-1}{cos(a,b)-cp(A,B)} \leq 1 \Rightarrow$$

$$1 \leq cp(A,B) but cp(A,B) \in [0,1] \Rightarrow cp(A,B) = 1 \Rightarrow A = B, \quad \forall A,B \in U$$

Note that if $\mathcal{B} = 0.5$, cp and cos have the same weights.

5 How the algorithm works

To create the initial sets by using the nearest neighbours of each data we require:

- The hyperellipsoids generated by the initial sets following the shape of the data. This means that the number of neighbours cannot be too large so that we obtain hyperellipsoids with areas empty of data.
- The sets must overlap sufficiently to be able to apply *cp* as a similarity measure so the number of neighbours cannot be too small, otherwise *cp* will be zero for most of the pairs of sets and *cp* will not put weight on *s* and *cp* will be useless in *s*.

Both requirements will depend on the features of the data such as form, density or location. Giving a prior determined number may be risky, we therefore suggest another idea for this.

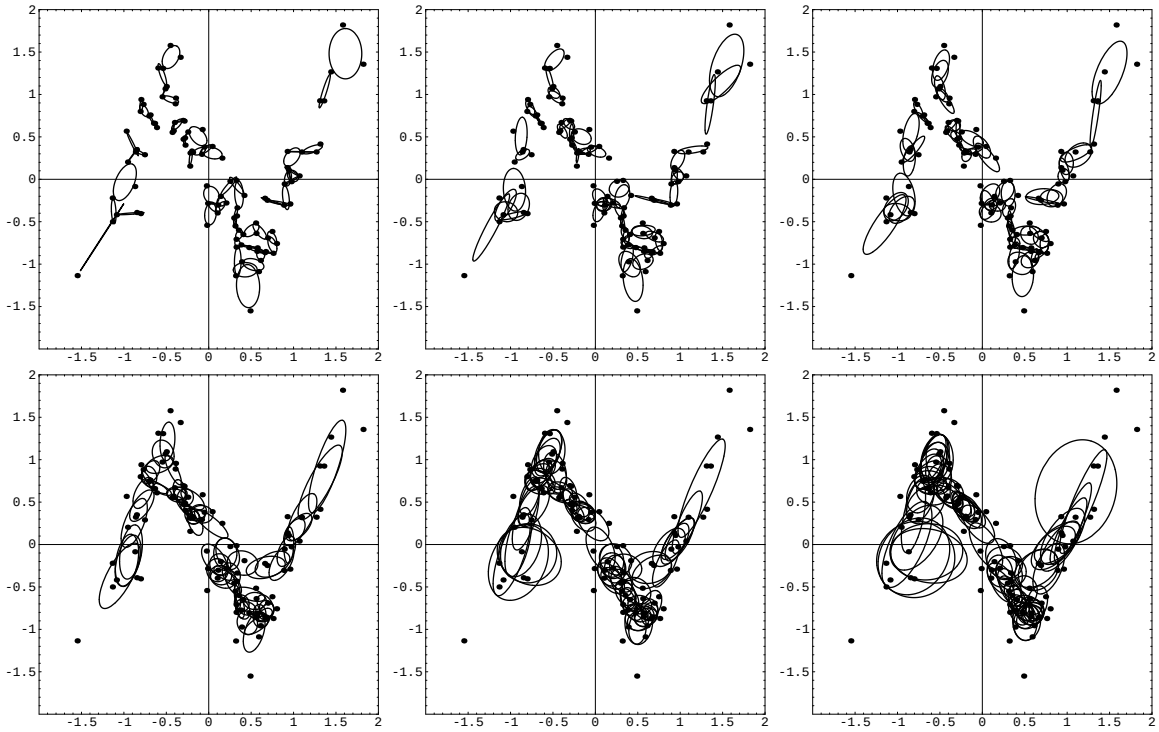

Figure 3: Initial random sets for 3, 4, 5, 8,12 and 15 nearest neighbours.

Note that in figure 3, for a small number of nearest neighbours, the sets are not overlapping sufficiently to apply *cp* (see figure 4 the finals clusters corresponding to these initial random sets), while for a large number of nearest neighbours the sets generate ellipsoids with empty areas. Between 5 and 8 nearest neighbours seems to be a good choice in this example.

Once the initial sets have been created by using an appropriate number of neighbours, it is obvious that the hyperellipsoids corresponding to sets of close data strongly overlap. They have a high percentage of common data and that implies that *cp* is close to one. This also implies that very often the orientation of their hyperellipsoids has a small variation. However this does not always happen. We must consider that in the first steps all the hyperellipsoids may fit the general shape of the data well and pairs of sets with *cp* close to 1 must be joined even if the angle of their hyperellipsoids has a strong variation since the resulting hyperellipsoids of the join will continue to fit the form of the data well. Then it would be interesting to give a bigger weight to *cp* rather than *cos* in the first steps.

The proposed similarity measure *s*, the angle aside, tends to generate independent clusters (without any data in common) since the criterion does the merging between sets with the highest percentage of common data, and step by step, it makes the common data among the sets decrease.

At the same time, step by step, the clusters increase in size and the orientation of their hyperellipsoids becomes more important in the joining process. This is because the common number of data between the pairs of sets is decreasing and joining two sets with a low *cp* without taking into consideration the *cos* can originate a new set with some areas in which its hyperellipsoids are empty, i.e. this new hyperellipsoid will not fit the shape of the data well. Then, whereas *cp* was more important than *cos* in *s*, in the last steps *cos* must be the principal criterion for joining.

Now, we need to think whether joining two independent clusters (without common data) makes sense. We will hold the idea that two independent clusters cannot be joined since a independent cluster alone explains a substructure of data differing from the general structure. We will add this idea to the algorithm as a restriction in the condition of searching the most similar clusters.

The outcome of using this restriction and the event mentioned before (the algorithm tends to generate a hard partition of clusters i.e. clusters without common data) is that if we use to small a number of nearest neighbours to create the initial sets, the common data between pairs will be small and the algorithm may be stopped if it finds a hard partition of the data, and maybe we have not obtained the requested number of final sets (see figure 3 and 4). To avoid this situation, we can take a bigger number of neighbours and so the initial set will have a stronger overlap. How big does this number need to be? If we want to obtain the hardest partition possible by using this algorithm, we propose to give a small value and to increase it until the algorithm does not stop before having the final number of requested clusters.

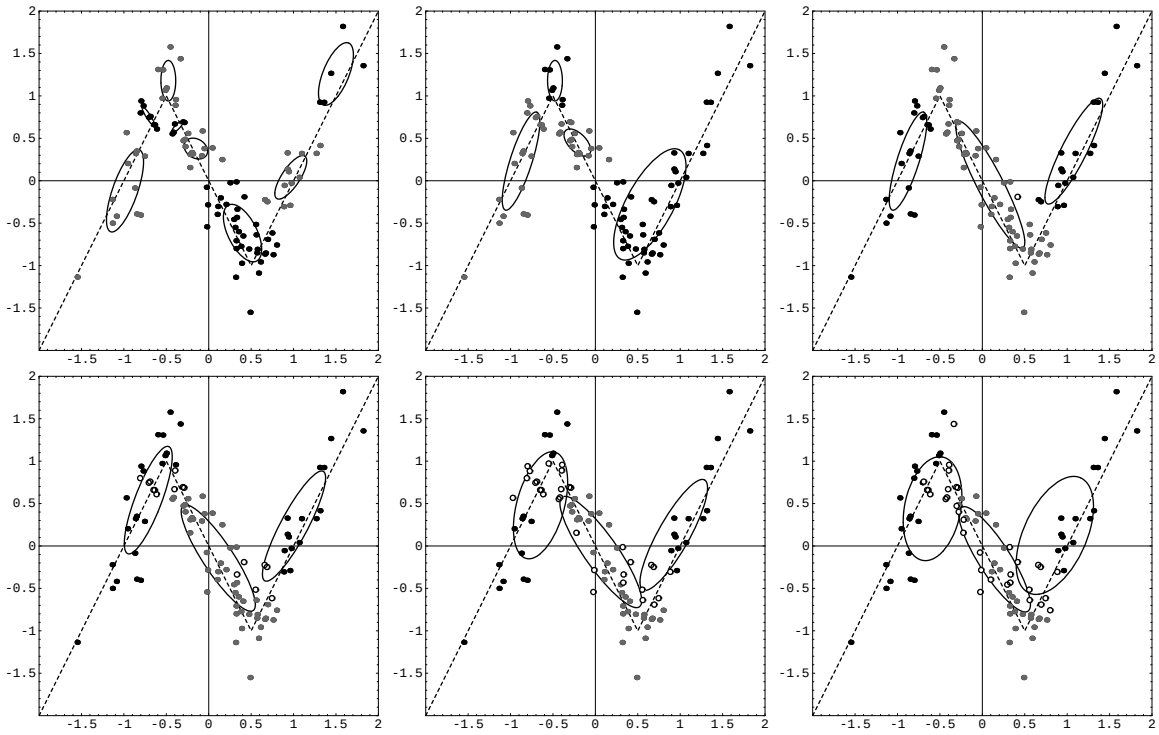

Figure 4: Final clusters for 3, 4, 5, 8, 12 and 15 nearest neighbours.

See in the figure 4 the final clusters by using different number of nearest neighbours to create the initial random sets and supposing that the final number of clusters is three. Note how with 3 or 4 nearest neighbours the algorithm, for these data, gets stop before getting the final 3 requested clusters, because it achieves a hard partition. Also, note how with 5 or more, the algorithm gets the final 3 clusters which form a weaker partition at the same time as the number of nearest neighbours is increased, having a higher number of common data (circles).

In the last steps, when the number of clusters and the number of common data between pairs of clusters is small, the restriction mentioned before, also avoids joining two clusters in parallel but without any common data (with hyperellipsoids far away from each other in terms of the Euclidean distance) (see figure 4).

6 $\mathcal{B}$ as a dynamic regulator of s

The introduction of the parameter $\mathcal{B}$ in s gives the opportunity to perform an algorithm as previously explained, giving *cp* more weight in the first steps and *cos* more weight in the final steps. As $\mathcal{B}$ is not constant, it must decrease with every step. We propose the following function:

$$\mathcal{B} : [1, \infty) \subset R \longrightarrow [0, 1] \subset R$$
$$\mathcal{B}(x) = 1 - \frac{1}{\sqrt{x}}$$

where x is the number of cluster at every step of the iterative algorithm. The idea is illustrated in figure 5.

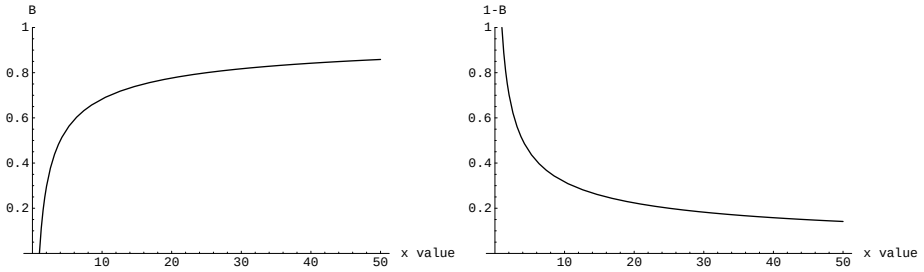

Figure 5: $\mathcal{B}$ and 1-$\mathcal{B}$ as functions of the number of clusters.

Note that when the number of clusters is high, $\mathcal{B}$ is close to one and 1-$\mathcal{B}$ is close to zero so *cp* gives the majority of the weight to s. When the number of clusters is low, $\mathcal{B}$ is close to zero and 1-$\mathcal{B}$ is close to one so *cos* gives the majority of the weight to s. Other choices of $\mathcal{B}$ can be tried.

7 Description of the algorithm

These are the steps that resume the algorithm:

1 Enter the number of final clusters and the number of nearest neighbours (nn)

2 Get initial sets for nn

3 Get matrices of *cp* and *cos*

4 Get the most similar pair of sets to agree with

$$Max\left\{ s(A,B) = \left(1 - \frac{1}{\sqrt{clusters}}\right) cp(A,B) + \frac{1}{\sqrt{clusters}} \quad cos(a,b), \quad \forall A, B \in U \right\}$$
$$and$$
$$cp(A,B) \leq 0$$

5 If *cp(A,B)=0* for all A,B, then increase nn and go to step 2

6 Join these two sets

7 If the number of clusters is equal to final clusters then END.

8 Update matrices of *cp* and *cos* and go to step 5

8 Conclusion

In this report hyperellipsoids were used as representations for the shape of data. We developed an algorithm which works particularly well when the shape of the data presents linear elements.

The introduction of $\mathcal{B}$ in s, as a function which equilibrates the influence of the orientation of the sets in the different stages of the clustering, lets us avoid, some local structures of data, generated by the existence of data with a large deviation, being taken into consideration in the first steps. This is fundamental if we are looking for more general structures. Refer to the figures and note how sometimes we can find neighbouring hyperellipsoids with different orientations.

We also want to note the relationship between the obtained clusters by using this algorithm and multiple linear regression analysis. Note that the $n-1$ principal components (in an n dimensional space) of a set of data generate the same hyperplane that we should obtain by applying multiple linear regression analysis.

In figure 4 all the graphs have included the original lines of the function that has been used to generate the data (see Appendix A). They are perfect regression lines. We can check the precision of the algorithm by comparing these lines with the largest axis of the ellipsoids. Note that the hardest partition corresponds to the initial sets with 5 nearest neighbours gives the best approximation.

In this way, once we have obtained the final clusters, we can then use the hyperplane, generated by $n-1$ principal components in every cluster, as a model to make predictions as if we are using a multiple linear regression. Always if we are supposing that the error or deviation of the data from the hyperplane follow a normal distribution.

Of course, this assumption of normality cannot always be taken. Other methods must be used to predict new values, although it can be used for cases where we do not have alternative methods of prediction available.

Finally it remains to be said the algorithm is sensitive to the number of data in terms of running time. This is because the increment in the dimension of the similarity matrix among sets, makes an exponential increment in the number of comparations that must be made in every step to find the closest sets.

Some variations of the algorithm have been proved to increase the speed. For example, by allowing more than two hyperellipsoids to join at every step by using a threshold of closeness. Better results have been obtained by using the original algorithm since we ensure that the joins are done in the correct order.

9 Appendix A

1 The Data set

The model generating the data in the examples was obtained from the book [1] and has been used in the paper [2]. The model was first described by Ikoma and Hirota in 1993. It consists of a nonlinear AR(1) dynamic system simulated as follows:

$$x(k+1) = f\left(x(k)\right) + \epsilon(k), \quad f(x) = \begin{cases} 2x - 2, & 0.5 \leq x, \\ -2x, & -0.5 < x < 0.5 \\ 2x + 2, & x \leq -0.5 \end{cases}$$

where $\epsilon(k) \sim N(0, \sigma^2)$ with $\sigma = 0.3$, $x(0) = 0.1$ and $k \in \{0, ..., 100\}$.

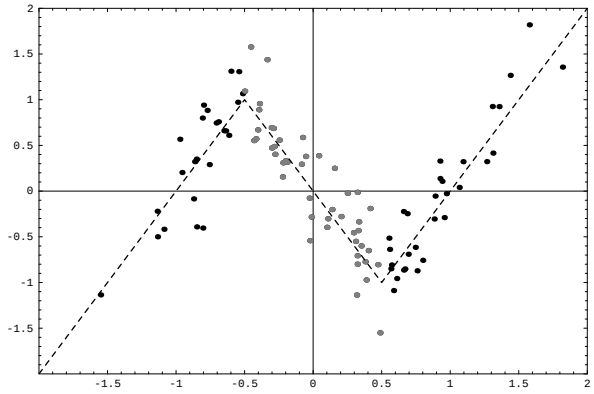

Figure 6: Nonlinear dynamic system. $x(k)$ against $x(k+1)$.

2 Definition of a Similarity Measure

Given a set of elements $\mathbf{U}$, a *similarity* measure between two elements is defined as function

$$\mathbf{s}\colon \mathbf{U} \times \mathbf{U} \to [0, 1]$$

which verifies the following properties:

$$\begin{aligned} &0 \leq s(A, B) \leq 1, && \forall A, B \in U \\ &s(A, A) = 0, && \forall A \in U \\ &s(A, b) = s(B, A), && \forall A, B \in U \\ &s(A, B) = 1 \Rightarrow A = B, && \forall A, B \in U \end{aligned}$$

References

[1] Robert Babuska. *Fuzzy modelling for control.* Kluwer, 1998.

[2] Olaf Wolkenhauer and Mario Garcia-Sanz. A random sets statistical approach to system identification. AIDA, 1999.