AnfisImplementationRemarks

1- If one starts with grid partition a high number of fuzzy sets by dimension is needed, leading to a prohibitive number of rules. The computer will take long time.

2- Use preferable subtractive clustering that adapts well to the structure of the data. The four parameters in the GUI of Subtractive Clustering have the following meaning:

- **range of influence (cluster radius \( r_a \))** determines the range of influence of the cluster in each data dimension. The produced clusters are spherical meaning that the same cluster radius is used for all data space dimensions. If you use the script implementation different radius per dimension may be specified.

- **squash factor** \( s_q \) is used to multiply the radius \( r_a \) value to determine the neighbourhood of a cluster centre within which the existence of other cluster centres are to be discouraged. Os squash factor is low, cluster centers near to each other are produced.

- **accept ratio** \( \varepsilon_a \) sets the potential, as a fraction of the potential of the **first** cluster centre, above which another data point will be accepted as a cluster centre. A high value number produces

- **reject ratio** \( \varepsilon_r \) sets the potential, as a fraction of the potential of the **first** cluster centre, below which a data point will be rejected as a cluster centre. If there are some points that are outside the “main stream”, known as outliers, they will be ignored if this value is high. In our case there are small sets of points that are not outliers and must be considered. For this reason the reject ratio must be very low, like 0.001 or 0.0001.

The following figures illustrate the influence of squash factor and reject ratio.

Radius=0.25  accept ratio=0.25 reject ratio=0.001

![Radius=0.25  accept ratio=0.25 reject ratio=0.001](image1)

Squash factor=2  Squash factor=4

Radius=0.25  Squash factor=2  accept ratio=0.25:

![Radius=0.25  Squash factor=2  accept ratio=0.25:](image2)

Reject ratio =0.15  Reject ratio = 0.001

3- After creating the fis with Anfisedit, one can see the membership rules in the menu edit>membership function in **Anfis Editor** (not in fuzzy editor). The subtractive clustering creates, by default, first order Sugeno systems. In order to have zero order ones, one must **change in the membership editor menu**, one by one, the output membership functions from linear to constant, without leaving Anfis Editor.
After that, one is ready for training.
Next figure illustrates one result with 3 epochs. It is not not a good one, since training needs more epochs.

Radius=0.25  accept ratio=0.25 reject ratio=0.001 Squash factor=2

After 100 epochs one gets

Which is a better result from the numerical criteria point of view but is a worse one from the application side. In this case it is preferable to write a script and use

Anfis in the Command line:

\[
[fis, error, steps, chkFis, chkErr] = \text{anfis(trnData, numMFs, trnOpt, dispOpt, chkData, optMethod)}
\]

trnOpt: a vector of training options. When a training option is entered as NaN, the default options is in force. These options are as follows:
- trnOpt(1): training epoch number (default: 10)
- trnOpt(2): training error goal (default: 0)
- trnOpt(3): initial step size (default: 0.01)
trnOpt(4): step size decrease rate (default: 0.9)
trnOpt(5): step size increase rate (default: 1.1)

Augmenting the step size increase rate, one can improve the results. The needed number of epochs may also be rather big.

Sugeno first order 50 epochs Radius=0.25  accept ratio=0.25 reject ratio=0.001 Squash factor=2

In training, after a certain number of epochs, the error change remains practically unchanged, meaning that a more elaborated strategy is needed for the optimization iterative procedure. This is a challenge to the groups.

Sugeno zero order 2000 epochs =0.25  accept ratio=0.25 reject ratio=0.001 Squash factor=2

After 161 epochs the criteria does not change (0.25361). The solution is now better numerically

Conclusion: the MSE is not a good choice here, since the number of points where output is 1 is smashed by the number of points where it is 0. A better criteria is needed, allowing to give more importance to the points where output must be 1.

Probably the defuzzification method should also be changed. For example in a point with a high firing intensity, near to the center of a cluster, the output of the corresponding rule could be set to the target

From Matlab Help:

subclustFind cluster centers with subtractive clustering (used under ANFIS edit).

\[ [C,S] = \text{subclust}(X,\text{radii},\text{xBounds},\text{options}) \]

Description: This function estimates the cluster centers in a set of data by using the subtractive clustering method. The subtractive clustering method assumes each data point is a potential cluster center and calculates a measure of the likelihood that each data point would define the cluster center, based on the density of surrounding data points. The algorithm: Selects the data point with the highest potential to be the first cluster center Removes all data points in the vicinity of the first cluster center (as determined by radii), in order to determine the next data cluster and its center location Iterates on this process until all of the data is within radii of a cluster center The subtractive clustering method is an extension of the mountain clustering method proposed by R. Yager. The matrix X contains the data to be clustered; each row of X is a data point. The variable radii is a vector of entries between 0 and 1 that specifies a cluster center's range of influence in each of the data dimensions, assuming the data falls within a unit hyperbox.
Small radii values generally result in finding a few large clusters. The best values for radii are usually between 0.2 and 0.5. For example, if the data dimension is two (X has two columns), radii=[0.5 0.25] specifies that the range of influence in the first data dimension is half the width of the data space and the range of influence in the second data dimension is one quarter the width of the data space. If radii is a scalar, then the scalar value is applied to all data dimensions, i.e., each cluster center has a spherical neighborhood of influence with the given radius. xBounds is a 2-by-N matrix that specifies how to map the data in X into a unit hyperbox, where N is the data dimension. This argument is optional if X is already normalized. The first row contains the minimum axis range values and the second row contains the maximum axis range values for scaling the data in each dimension. For example, xBounds = [-10 -5; 10 5] specifies that data values in the first data dimension are to be scaled from the range [-10 +10] into values in the range [0 1]; data values in the second data dimension are to be scaled from the range [-5 +5] into values in the range [0 1]. If xBounds is an empty matrix or not provided, then xBounds defaults to the minimum and maximum data values found in each data dimension. The options vector can be used for specifying clustering algorithm parameters to override the default values. These components of the vector options are specified as follows: options(1) = squashFactor: This factor is used to multiply the radii values that determine the neighborhood of a cluster center, so as to squash the potential for outlying points to be considered as part of that cluster. (default: 1.25) options(2) = acceptRatio: This factor sets the potential, as a fraction of the potential of the first cluster center, above which another data point is accepted as a cluster center. (default: 0.5) options(3) = rejectRatio: This factor sets the potential, as a fraction of the potential of the first cluster center, below which a data point is rejected as a cluster center. (default: 0.15) options(4) = verbose: If this term is not zero, then progress information is printed as the clustering process proceeds. (default: 0) The function returns the cluster centers in the matrix C; each row of C contains the position of a cluster center. The returned S vector contains the sigma values that specify the range of influence of a cluster center in each of the data dimensions. All cluster centers share the same set of sigma values.

Examples

[C,S] = subclust(X,0.5)
This command sets the minimum number of arguments needed to use this function. A range of influence of 0.5 has been specified for all data dimensions.

[C,S] = subclust(X,[0.5 0.25 0.3],[2.0 0.8 0.7]) This command assumes the data dimension is 3 (X has 3 columns) and uses a range of influence of 0.5, 0.25, and 0.3 for the first, second, and third data dimension, respectively. The scaling factors for mapping the data into a unit hyperbox are obtained from the minimum and maximum data values. The squashFactor is set to 2.0, indicating that you only want to find clusters that are far from each other. The acceptRatio is set to 0.8, indicating that only data points that have a very strong potential for being cluster centers are accepted. The rejectRatio is set to 0.7, indicating that you want to reject all data points without a strong potential.