International Journal of Computer Science and Applications,
Vol. 5, No. 3, pp 1 - 8


# FEATURE EXTRACTION USING FUZZY RULE BASED SYSTEM

NARENDRA S. CHAUDHARI and AVISHEK GHOSH

*School of Computer Engineering,*
*Nanyang Technological University, 50 Nanyang Avenue,*
*Singapore, 639929,Singapore*
*ASNarendra@ntu.edu.sg, nsc183@gmail.com, ghosh.avishek@gmail.com*
*http://www.ntu.edu.sg/home/asnarendra*

Data projection is an important tool in exploratory data analysis. Sammon's non linear projection method lacks predictability and is ineffective for large data sets. To introduce predictability we implement an extension of Sammon's algorithm using fuzzy logic approach. The fuzzy based rule model is implemented in the .Net framework using Microsoft Visual Studio with Visual C# as the programming language. The datasets used to test the system are stored in MS SQL Server databases and are programmatically linked with Visual Studio. The implemented algorithm is tested with a few datasets and is found to have good predictability and works well with large datasets.

*Keywords*: fuzzy rule based system, non linear projection, Sammon's method

## 1. Introduction

The goal of a projection algorithm is to map p-dimensional patterns to q-dimensional space such that the structure of the data is preserved, $q < p$. In exploratory data analysis, tbe value of q is two or three. Projection or mapping algorithms are extremely useful in understanding the structure of multidimensional data. Data projection can provide a better insight into the clustering tendency or intrinsic dimensionality of the data. Projection algorithms are also useful for feature extraction in pattern recognition. This paper deals with the extraction of lower dimensional data through the implementation of structure preserving algorithms.

In pattern recognition literature quite a few number of methods of data projection are available. These methods differ among themselves in terms of their mapping function $\Phi$, the training mechanism $\Phi$, and the optimization criterion being used. Mapping functions are either linear or non linear and can be trained through supervised or unsupervised methods.

Sammon's non linear projection method [Sammon (1969)] is efficient with relatively smaller data sets. the method lacks predictability and the computational overhead is significantly high for large data sets. While dealing with large data sets the inability to generalize significantly reduces the performance of the Sammon's algorithm. Some of the neural network (NN) implementations attempted in the past [Jain and Mao (1992)], to introduce the generalization capability in the Sammon's method, were inefficient with outliers and the use of hidden layer of neurons were not interpretable. Fuzzy rule based systems have predictability, are interpretable and are efficient with outliers and thus in

this paper we implement a proposed method which augments the Sammon's method with a fuzzy logic approach for data projection. The method combines the predictability of fuzzy rule based systems and structure preservation quality of the Sammon's algorithm. We also implement the Sammon's algorithm to enable us to compare the results of the fuzzy logic approach with the Sammon's data projection method. The fuzzy rule based algorithm produces similar results as the original Sammon's algorithm at a lower cost while introducing predictability. The performance for some of the data sets from the University of California Irvine (UCI) Machine Learning Repository has been found to be quite satisfactory when compared with the original Sammon's algorithm.

## 2. Implementation Framework

We implemented the fuzzy rule based algorithm in the .NET framework using Microsoft Visual Studio as the integrated development environment (IDE). We used Visual C# as the programming language for implementation. The data sets used to test the algorithm are stored in databases in the Microsoft SQL Server 2005. The link between the SQL server database and Visual Studio is made programmatically. The infrastructure of the framework can be seen in Fig. 1.

### 2.1. *.Net framework*

.Net Framework is a platform or development environment to seamlessly create applications that are web based or based on Windows Forms. The framework is platform independent and language independent. This means that .Net Framework allows us to use different programming languages such as VB.Net, J#, C#, Jscript, VBScript, and Managed C++ and run applications on different platforms such as UNIX, Macintosh, and Linux. Moreover, .Net Framework enables us to use various off-the-shelf libraries that help the development of applications faster, easier, and cheaper. .Net Framework now supports over 20 different programming languages. We implement the algorithm in the .Net framework because of the above mentioned features.

### 2.2. *Microsoft visual studio and visual C#*

Microsoft Visual Studio is the main Integrated Development Environment (IDE) from Microsoft. It can be used to develop console and GUI applications along with Windows Forms applications, web sites, web applications, and web services in both native as well as managed code. In built language supports C/C++ (via Visual C)m VB.NET (via Visual Basic.NET) and C# (via Visual C#). Visual Studio includes a code editor as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building GUI applications, web designer, class designer, and database schema designer. C# is an object-oriented programming language developed by Microsoft as part of the .NET initiative. The C# language has a procedural, object-oriented syntax based on C++ and

includes influences from aspects of several other programming languages with a particular emphasis on simplification. Implementation of the algorithms in the Visual C# makes the code portable and hence available for usage in different applications.

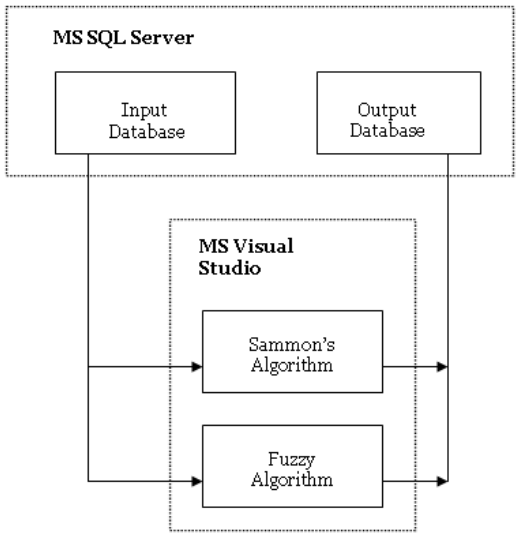



Fig. 1. Framework of implementation of algorithms Sammon's algorithm and fuzzy rule based system in the .NET Framework

### 2.3. *Microsoft SQL server 2005*

Microsoft SQL Server 2005 is a comprehensive integrated data management and analysis software that enables organizations to reliably manage mission-critical information and confidently run today's increasingly complex business applications. Creating a database in the SQL server, we use it to store different datasets and link it programmatically to the Microsoft visual studio. As the datasets are readily available to the program, they need not be stored in the dynamic memory of the program and as such memory requirements to run the application is thereby reduced.

## 3. Fuzzy Rule Based Model

To introduce predictability into Sammon's non linear projection method in [Pal et al. (2002)] Pal *et al.* proposed a fuzzy model. The basic logic behind the system was to use a set of fuzzy rules to discover the relation between input data and the projected output data so that new points could be projected easily. We assume a time invariant probability distribution for all data sets under consideration. Now if a rule base is derived from a

*representative* sample *X* then its output for a new data point $x_k$ can be expected to be the same as that of the system developed using the data set $X' = X \bigcup \{x_k\}$.

So the approach proposed by Pal consists of the three following basic steps

projecting *X* using Sammon's algorithm to project *Y*

Extracting a fuzzy rule base *R* from (*X*, *Y*)

Finally use R to project all new data points

Let the representative sample be $X = \{x_1, x_2, ..., x_n\} \subset R^p$ and projected output data set using the Sammon's algorithm be $Y = \{y_1, y_2, ..., y_n\} \subset R^q$. We define

$$X^* = \{x_i^* = \begin{pmatrix} x_i \in R^p \\ y_i \in R^q \end{pmatrix} \in R^{p+q}, i = 1,2,..., n\} \quad (1)$$

i.e. $x_i^*$ is $x_i$ augmented by $y_i$. Next step involves clustering the *X** by a suitable clustering algorithm producing a set of centroids

$$V^* = \{v_i^* = \begin{pmatrix} v_i^x \in R^p \\ v_i^y \in R^q \end{pmatrix} \in R^{p+q}, i = 1,2,..., C\} \quad (2)$$

and a fuzzy partition matrix. We extract the fuzzy rule base *R* from these cluster centroids.

Clustering is used to extract fuzzy rules for the reason that if the input and the output are assumed to have a smooth relationship then when we have a cluster with centroid $v_i^x$ in input space then the corresponding points in output space will likely form a cluster around corresponding centroid $v_i^y$. Thus the *i*th cluster can be translated into a rule of the form using the Mamdani-Assilian (MA) model [Mamdani, Assilian (1975)]:

If ***x*** is CLOSE to $v_i^x$ then ***y*** is CLOSE to $v_i^y$

The antecedent part of the fuzzy rule according to the MA model is conjunction of *p* atomic clauses: if $x_1$ is CLOSE to $v_{i1}^x$, $x_2$ is CLOSE to $v_{i2}^x$ and … and $x_p$ is CLOSE to $v_{ip}^x$.

Two important issues need to be considered for the clustering part of the algorithm.

**Choice of clustering domain**: The options available for clustering domain are clustering of representative sample *X*, clustering of corresponding output *Y*, clustering of *X** or the clustering of both *X* and *Y* separately. The reason for choosing *X** is that because in *X**, *x* and the corresponding *y* are tied together and as such the rule interpretation becomes easier.

**Use of clustering algorithm**: Although there could be many choices, Fuzzy C-Means (FCM) clustering algorithm for clustering *X** is used as we have no idea about the type of cluster structure that may be present in the data. FCM extracts hyper spherical clusters and any input–output relation, can be approximated by a reasonable number of hyper spherical clusters. Given $X = \{x_1, x_2, ..., x_n\} \subset R^p$, the FCM partition algorithm finds a partition matrix $U = [u_{ik}]_{c \times n}$ and set of centroids $V = \{v_1, v2, \ldots, v_c\}$ minimizing

$$J_m(U,V) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^m \left\| x_k - v_i \right\|_A^2 \quad (3)$$

where m is the weighting exponent typically greater than 1 and A is any $p \times p$ positive definite matrix.

### 3.1. *Rule identification scheme*

Let $v_i^*, i = 1,2,...,c$ be the centroids of the clusters obtained by FCM on X*. We translate the *i*th cluster into the rule using the Mamdani –Assilian model. For an input vector in input space $x_k$, the output $\hat{y}_k = (\hat{y}_{k1}, \hat{y}_{k2}, ..., \hat{y}_{kq})^T$ is computed using the height method of defuzzification for the sake of computational simplicity. Other methods such as the center of gravity can also be used.

### 3.2. *Antecedent memberships and tuning of parameters*

In order to implement the rule base it is important to define the membership function for "$x_j$ CLOSE to $v_{ij}^x$". In this paper we have implemented the asymmetric triangular functions having peak, $a_{ij}$ and left and right widths as $b_{ij}^L$ and $b_{ij}^R$ respectively.

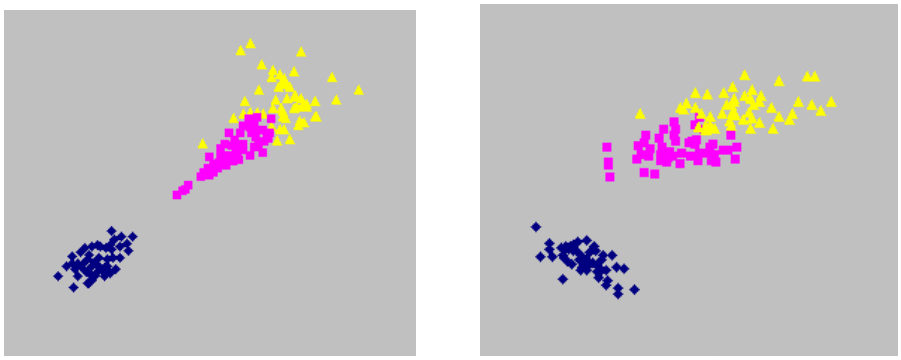
For each pair of attribute of the input vector $x$ and cluster centroid we define the asymmetric triangular functions. The parameters of the antecedent membership functions and the peak of consequent membership functions are optimized using gradient descent approach, to minimize error which is defined as

$$E = \sum_{k=1}^{n} \left\| \hat{y}_k - y_k \right\|^2 \tag{4}$$

### 3.3. *Implementation characteristics*

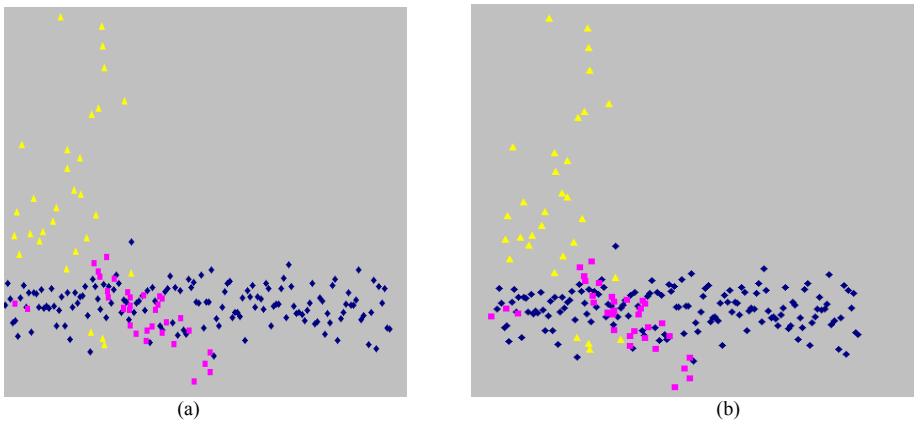
The representative sample used to train the fuzzy system is read into the main memory from the SQL server database where the different datasets are stored. The representative sample consists of randomly selected 30% of the whole data set. According to size and number of features in the representative sample two dimensional matrices are dynamically generated to store the sample. We use Sammon's algorithm to generate the output of the training sample. The FCM algorithm is applied to the input sample augmented with the corresponding output. This generates the set of required cluster centroids. The initial fuzzy partition matrix used by FCM is also generated with a random configuration.

For the extraction of fuzzy rules from the training sample and the set of centroids the MA model is implemented. Dynamic two dimensional arrays whose length and breadth depend on the number of clusters and features in the input space are used to store the parameters of the antecedent membership functions. The maximum *t-conorm* function is used to aggregate the *p* components of the antecedent clause of the MA model.



(a) (b)

Fig.2. (a) Sammon output on IRIS for entire data plot. (b) Fuzzy Model output on IRIS for entire data plot



(a) (b)

Fig.3. (a) Sammon output on New Thyroid for entire data plot. (b) Fuzzy Model output on New Thyroid for entire data plot

The gradient descent approach is implemented for which the error tolerance threshold and maximum number of iterations can be set according to characteristics of the data set being tested and level of accuracy desired.

## 4. Results

### 4.1. *Data sets and computational protocols*

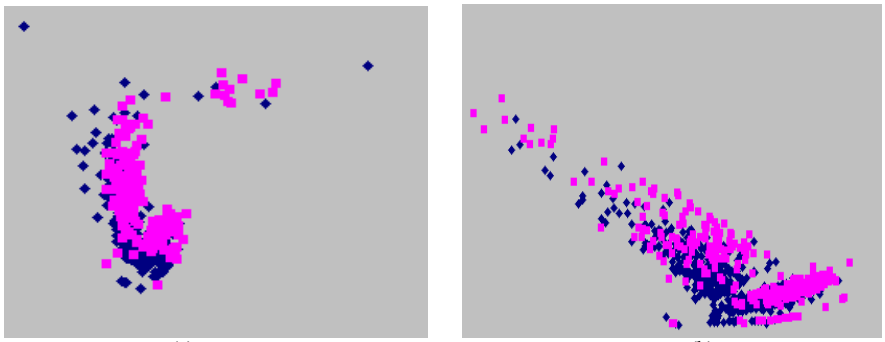
After the implementation of the Sammon's algorithm and fuzzy logic approach for feature extraction, the algorithms are tested on data sets from the UC Irvine (UCI) Machine Learning Repository. Both the algorithms are tested on three data sets named **Iris, New Thyroid,** and **Diabetes**.

Iris is a well-known data set consisting 150 points from three classes in a four-dimensional space. Each class has 50 points. One of the classes is well separated from the rest while the other two have some overlap.

New Thyroid data set consists of 215 points from three classes in a five-dimensional space. The three classes have 150, 35 and 30 points respectively. There is considerable overlap among the three classes.

Diabetes data set is comprised of 768 data points from two classes in a eight-dimensional space. Each class has 384 points. There is a major overlap among the two classes.

For the fuzzy logic approach using the MA model 30% of each data set is used as the training set whereas the whole data set is used as the testing set. As we want to compare the generalization capability of the fuzzy system, the Sammon's algorithm is applied on the entire data sets. For both algorithms the output dimension $q$ is set to be two as it is easier to analyze the scatter plots of the projected data.



(a) (b)

Fig.4.(a) Sammon output on New Thyroid for entire data plot. (b) Fuzzy Model output on New Thyroid for entire

### 4.2. *Results*

In this paper we present the two-dimensional scatter plots of the projected data which helps in easier visual assessment. The scatter plots for the IRIS data from each algorithm is shown in Fig. 2. As can be easily seen, Sammon's algorithm works effectively for Iris as the size of the data set is relatively small. Structure preservation is achieved as there is slight overlap between the two classes and the other is distinctly separate. For the fuzzy rule based model the results are similar to that of Sammon's algorithm and the projection is good.

Fig. 3 displays the two dimensional scatter plots for the New Thyroid data set. The results from Sammon's algorithm and fuzzy logic approach are nearly similar. Thus it is evident that the fuzzy rules extracted using the MA model provide good generalization capability. However for the diabetes data set the results from the fuzzy model are comparatively better than the Sammon's algorithm as can be seen from Fig. 4. Though there is significant overlap among the two classes the fuzzy model does a better job at classifying them. Hence for larger data sets, not only does the fuzzy model gives a better

result it also does so at a lower cost in terms of computational overhead as compared to the Sammon's algorithm.

## 5. Conclusions

In this paper a low cost data projection algorithm with prediction ability has been successfully implemented in the .NET framework. The algorithm integrates the structure preserving ability of the Sammon's algorithm and the generalization capability of rule based fuzzy systems. Different data sets from the UCI Repository were used and the comparison of the results makes it evident that the fuzzy model is significantly better than the Sammon's method as it has additional features of predictability and reduced computational overhead.

The programming was done Visual C# using the Microsoft Visual Studio IDE. The different data sets used for training and testing were stored in a SQL server database. The storage of the data sets in the SQL server database reduces the memory requirements of the fuzzy system as the whole database need not be stored in the main memory of the program. We also programmatically achieved the link between the Visual Studio and the SQL server.

## References

A. K. Jain and J. Mao. (1992): Artificial neural networks for nonlinear projection of multivariate data, Proc. IEEE Int. Joint Conf. Neural Networks, vol. 3, pp. 59–69.

A. K. Jain and R. C. Dubes.(1988): Algorithms for Clustering Data. Upper Saddle River, NJ: Prentice-Hall.

E. H. Mamdani and S. Assilian. (1975): An experiment in linguistic synthesis with a fuzzy logic controller, Int. J. Mach. Studies, vol. 7, no. 1, pp.1–13.

N. R. Pal and V. K. Eluri (1997): Neural networks for dimensionality reduction, in Progress in Connectionist-Based Information Systems, Proc.4th Int. Conf. Neural Inform. Processing, vol. 1, Kasabov, Ed., New Zealand, pp. 221–224

N. R. Pal, V. K. Eluri and G.K. Mandal (2002): Fuzzy Logic Approaches to Structure Preserving Dimensionality Reduction, in IEEE Trans. on Fuzzy Systems, vol. 10, pp. 277-286.

T. Takagi and M. Sugeno (1985): Fuzzy identification of systems and its application to modeling and control, IEEE Trans. Syst., Man, Cybern., vol. SMC-15, pp. 116–132.