



An Unsupervised Fuzzy Rule-Based Method for Structure Preserving Dimensionality Reduction with Prediction Ability

Suchismita Das, Nikhil R. Pal

► To cite this version:

Suchismita Das, Nikhil R. Pal. An Unsupervised Fuzzy Rule-Based Method for Structure Preserving Dimensionality Reduction with Prediction Ability. 15th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI), May 2019, Hersonissos, Greece. pp.413-424, 10.1007/978-3-030-19823-7_35 . hal-02331306

HAL Id: hal-02331306

<https://inria.hal.science/hal-02331306>

Submitted on 24 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

An Unsupervised Fuzzy Rule-based Method for Structure Preserving Dimensionality Reduction with Prediction Ability

Suchismita Das^[0000–0002–3952–3605] and Nikhil R. Pal^[0000–0001–6935–901X]

Electronics and Communication Sciences Unit
Indian Statistical Institute, Calcutta, 700108, India
{suchismitasimply,nrpal59}@gmail.com

Abstract. We propose an unsupervised fuzzy rule-based system to learn structure preserving data projection. Although, the framework is quite general and any structure preserving measure can be used, we use Sammon’s stress, an extensively used objective function for dimensionality reduction. Unlike Sammon’s method, it can predict the projection for new test points. To extract fuzzy rules, we perform fuzzy c-means clustering on the input data and translate the clusters to the antecedent parts of the rules. Initially, we set the consequent parameters of the rules with random values. We estimate the parameters of the rule base minimizing the Sammon’s stress error function using gradient descent. We explore both Mamdani-Assilian and Takagi-Sugeno type fuzzy rule-based systems. An additional advantage of the proposed system over a neural network based generalization of the Sammon’s method is that the proposed system can reject the test data that are far from the training data used to design the system. We use both synthetic as well as real-world datasets to validate the proposed scheme.

Keywords: Dimensionality reduction · fuzzy rules · predictability · Sammon’s mapping.

1 Introduction

We come across high dimensional datasets in various real-world problems related to finance, meteorology, computer vision, medical imaging, multimedia information processing, text mining, and many more [2, 15, 16, 18]. Handling high dimensional data is difficult as it demands more storage space, more processing time, and increased system complexity. For many applications, visualization of the structure in the data is important which is difficult for high dimensional data. Dimensionality reduction (DR) is a process that addresses these problems by representing a high dimensional data into a lower dimensional space, keeping the *relevant* information as intact as possible. DR schemes can be broadly categorized into two groups: feature selection (FS) and feature extraction (FE). FS chooses a small but important subset of features, whereas, FE constructs a smaller number of new features from the original features. Methods of FE are

more appropriate when the system does not bother about the physical meaning of the extracted features and focuses principally on preserving the local or global structure present in the original data.

We mathematically formulate FE as follows. Let the input data be represented by the set $\mathcal{X} = \{\mathbf{x}_i \in \mathbb{R}^D : i \in \{1, 2, \dots, n\}\}$. Here, n is the number of instances and D is the dimensionality of the data. The vector \mathbf{x}_i represents the i th instance of the data and x_{ik} represents the value of the k th feature associated with the i th instance. Then, $\forall \mathbf{x}_i \in \mathcal{X}$, the corresponding lower dimensional representation is $\mathbf{y}_i \in \mathbb{R}^{D'}$, $D' < D$. Let us represent the output dataset as $\mathcal{Y} = \{\mathbf{y}_i \in \mathbb{R}^{D'} : i \in \{1, 2, \dots, n\}\}$. FE is finding an implicit or explicit mapping, $\phi : \mathbb{R}^D \rightarrow \mathbb{R}^{D'}$, such that, an objective ψ is minimized/maximized. Usually, ψ is a function of \mathcal{X} and \mathcal{Y} , where $\mathbf{y}_i = \phi(\mathbf{x}_i)$, $\forall i$. Then, the obtained optimal mapping $\hat{\phi} = \arg \min_{\phi} \{\psi(\mathcal{X}, \mathcal{Y})\}$.

The Sammon's method [14] is a popular nonlinear FE technique that does not learn the mapping function ϕ explicitly. It aims to preserve certain geometric properties of the data by preserving the inter-point distances in the original higher dimensional space as the inter-point distances in the extracted lower dimensional space as much as possible. It reduces the following error function E (i.e., ψ):

$$E = \frac{1}{\sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij}^*} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{(d_{ij}^* - d_{ij})^2}{d_{ij}^*} \quad (1)$$

Here, d_{ij} and d_{ij}^* are the Euclidean distances between \mathbf{x}_i and \mathbf{x}_j ; and \mathbf{y}_i and \mathbf{y}_j ; respectively. The error function, E , is called Sammon's stress. To minimize E , Sammon's method employs the steepest descent method. Note that, for a fixed pair of points i and j , Sammon's stress multiplies a factor $1/d_{ij}^*$ to $(d_{ij}^* - d_{ij})^2$. As a result, the method pays more attention in preserving short pairwise distances, and hence, preserves the local structure.

The Sammon's method is a corner-stone method in structure preserving dimensionality reduction. However, it has a drawback: it does not learn the mapping function $\phi(\cdot)$ explicitly. After learning a mapping from n points, when a new point occurs, the Sammon's method needs to repeat the whole optimization process considering the $(n+1)$ points. From (1) we observe that, every iteration in the steepest descent search requires computation of $n(n-1)/2$ distances. When n is large, computations involved may be prohibitive. Another disadvantage of Sammon's method is that it optimizes a non-convex objective. So, it is possible that the method may get stuck at some local minima on the error surface. To avoid such a situation and to obtain a good solution the method needs to try several initializations. This again adds to the computational overhead.

To reduce the computational cost associated with Sammon's method, some attempts have been made to learn an explicit mapping with the Sammon's objective function. Jain and Mao [8, 9] have used a multilayer feed-forward network to learn Sammon's projection based explicit mapping. The number of input nodes and the number of output nodes of the neural network are set to D and D' , respectively. A pair of inputs \mathbf{x}_i and \mathbf{x}_j is applied one after another and their

corresponding outputs \mathbf{y}_i and \mathbf{y}_j are used to define the error function E . A problem with neural network based methods is that they predict outputs even when the test points are far away from the training data. Again, in [11], a supervised fuzzy rule-based model has been proposed to learn the Sammon’s projection parametrically. This method first generates the Sammon’s projection of the input data. Then, a fuzzy rule base is learnt using a supervised learning scheme, where the obtained output from Sammon’s projection is used as the target. An advantage of this method [11] over the neural network based methods [8,9] is that it can be used to detect test instances that are far away from the training data. However, it [11] has the following drawback. The computational cost for this method comprises the computational cost for obtaining Sammon’s projection and computational cost for learning the fuzzy system.

The objective of this paper is to propose a method for structure preserving dimensionality reduction. Although, any stress function can be used, we use here Sammon’s error, such that, it should avoid the additional computational overheads associated with the methods in [14,11]. Additionally unlike [8,9], it should decline to produce outcomes when the test points are far away from the training data. To achieve our objectives, here, we have contributed in the following ways:

1. We have proposed a fuzzy rule-based method which learns an explicit mapping function with the objective of Sammon’s method in an unsupervised fashion. It is unsupervised as there are no target values provided for the inputs. It is quite general and other objectives can also be used.
2. We have equipped the Sammon’s method with predictability, and hence, have cut down the computational cost associated with Sammon’s method to project new points.
3. The proposed system can indicate when its projection may not be reliable.

2 Proposed Method

The proposed method has the following major steps.

1. Form an initial fuzzy rule base \mathcal{R} .
2. Project input dataset \mathcal{X} with \mathcal{R} to obtain corresponding output \mathcal{Y} .
3. Modify \mathcal{R} to reduce the Sammon’s stress defined by \mathcal{X} and \mathcal{Y} .
4. Repeat steps 2 and 3 until convergence.
5. Use \mathcal{R} to project new points.

Next, we discuss the rule-based system used in this work, and then, we discuss the above steps with further details

2.1 The Rule-based System

We explore both Mamdani-Assilian (MA) and Takagi-Sugeno (TS) type fuzzy rule-based systems. Let there be r rules in the rule base. Let, for the i th sample \mathbf{x}_i , the m th output feature produced by the k th rule be y_{im}^k . Here, $m \in$

$\{1, 2, \dots, D'\}$; $k \in \{1, 2, \dots, r\}$; and $i \in \{1, 2, \dots, n\}$. For MA and TS models, the rules are denoted by \mathcal{R}_{km}^{MA} and \mathcal{R}_{km}^{TS} , respectively. They are defined as follows.

$$\begin{aligned} \mathcal{R}_{km}^{MA} : & \text{If } x_{i1} \text{ is } A_{k1} \text{ and } x_{i2} \text{ is } A_{k2} \text{ and } \dots x_{iD} \text{ is } A_{kD} \\ & \text{then } y_{im}^k \text{ is } B_{km}. \end{aligned} \quad (2)$$

$$\begin{aligned} \mathcal{R}_{km}^{TS} : & \text{If } x_{i1} \text{ is } A_{k1} \text{ and } x_{i2} \text{ is } A_{k2} \text{ and } \dots x_{iD} \text{ is } A_{kD} \\ & \text{then } y_{im}^k = u_{km}(\mathbf{x}_i). \end{aligned} \quad (3)$$

Here, A_{kl} is the k th fuzzy set (linguistic value) on the l th input feature and B_{km} is the k th output fuzzy set on the m th output variable; $u_{km}() : \mathbb{R}^D \rightarrow \mathbb{R}$ is a mapping function associated with the k th rule for the m th output feature.

Let, α_i^k be the firing strength of the k th fuzzy rule for the i th input \mathbf{x}_i . The rule firing strength is computed as the conjunction of D atomic clauses in (2)/(3) using a T -norm [7] over the fuzzy sets $A_{k1}, A_{k2}, \dots, A_{kD}$. We use the product operator here. For both MA and TS models, the output $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{iD'})^T$ is given by

$$y_{im} = \frac{\sum_{k=1}^r \alpha_i^k y_{im}^k}{\sum_{k=1}^r \alpha_i^k}. \quad (4)$$

We shall explain later how for the MA model we get the value of y_{im}^k from B_{km} . We note here that, for MA models, more involved method of defuzzification can be used [6].

2.2 Initial Rule Base Formation

When both input-output data are provided there are many ways of generating an initial rule base and its refinement [17, 5, 10–12, 4]. Some of the popular methods use clustering [11, 12, 4] of input-output data. The philosophy behind the use of clustering is that if there is cluster in the input space with centroid $\mathbf{v}_k \in \mathbb{R}^D$, for a smooth input-output relationship, the points in the output space are likely to form cluster around \mathbf{u}_k (which is the lower dimensional representation of \mathbf{v}_k). But here, we do not have any output data. We, however, use the clustering approach on the input data. We use the fuzzy c -means clustering algorithm [3] to find r clusters in the input data \mathcal{X} . The cluster centroids are denoted by $\mathbf{v}_k = (v_{k1}, v_{k2}, \dots, v_{kD})^T \in \mathbb{R}^D$; $k \in \{1, 2, \dots, r\}$. The k th cluster is translated into the antecedent of the k th rule as follows.

$$\begin{aligned} & \text{If } x_{i1} \text{ is close to } v_{k1} \text{ and} \\ & \quad x_{i2} \text{ is close to } v_{k2} \text{ and} \\ & \quad \vdots \\ & \quad x_{iD} \text{ is close to } v_{kD}. \end{aligned} \quad (5)$$

Here, x_{il} is close to v_{kl} is modeled using a Gaussian membership function centered at m_{kl} with spread σ_{kl} . The membership for x_{il} is

$$\mu_{ikl} = \exp \left\{ -\frac{(x_{il} - v_{kl})^2}{2\sigma_{kl}^2} \right\}. \quad (6)$$

Here, v_{kls} and σ_{kls} are the antecedent parameters of the rules.

For both MA and TS model, though we use the same fashion to define the rule-antecedents, we use different types of consequents. We discuss this next.

For MA model the rule-consequents are fuzzy sets in the output feature space. For each rule the output is a fuzzy set and the union of these fuzzy sets constitutes the fuzzy output which can be defuzzified using several methods [6]. For simplicity, here we use the height method of defuzzification [6]. Let B_{km} be either a symmetric triangular or Gaussian membership function with center at h_{km} . According to the height method the output of the rule R_{km}^{MA} is taken as h_{km} . Note that this becomes equivalent to the zeroth order TS model.

For the TS model, a consequent denotes an input-output mapping. For the first order TS model, this input-output mapping is a linear one and is described as follows.

$$y_{im}^k = u_{km}(\mathbf{x}_i) = c_{km0} + \sum_{l=1}^D c_{kml} x_{il} \quad (7)$$

Thus, the k th rule for TS model becomes

$$\begin{aligned} \mathcal{R}_{km}^{TS} : & \text{ If } x_{i1} \text{ is close to } v_{k1} \text{ and} \\ & x_{i2} \text{ is close to } v_{k2} \\ & \vdots \\ & x_{iD} \text{ is close to } v_{kD} \\ & \text{ then } y_{im}^k = c_{km0} + \sum_{l=1}^D c_{kml} x_{il}. \end{aligned} \quad (8)$$

For formation of the initial rule base, we randomly initialize the consequent parameters (h_{kms} for the MA model and c_{kmls} for the TS model).

2.3 Training the Rule Base

We estimate the rule base parameters that minimizes the Sammon's stress E in (1) using gradient-descent optimization. The learning rules for this are as follows.

$$v_{km}(t+1) = v_{km}(t) - \eta_v \frac{\partial E}{\partial v_{km}} \quad (9)$$

$$\sigma_{km}(t+1) = \sigma_{km}(t) - \eta_\sigma \frac{\partial E}{\partial \sigma_{km}} \quad (10)$$

$$h_{km}(t+1) = h_{km}(t) - \eta_h \frac{\partial E}{\partial h_{km}} \quad (11)$$

$$c_{kml}(t+1) = c_{kml}(t) - \eta_c \frac{\partial E}{\partial c_{kml}} \quad (12)$$

Here, t indicates the iteration number. $\eta_v, \eta_\sigma, \eta_h$ and η_c are learning rates for v, σ, h , and c , respectively.

2.4 Rejection of Outputs

An interesting feature of the proposed method is that it can reject test points for which it may not produce correct output. As shown in (4), the m th feature of i th output, y_{im} , is computed as a convex sum of the rule outputs y_{im}^k s. Each rule output (y_{im}^k) is multiplied by the firing strength (α_i^k) of the corresponding rule. The rule that fires with the maximum strength, primarily characterizes the output. The profile of the maximum rule firing strength on the training data may help in rejection of test points in a desired manner. For test points far away from the training data, no rule is expected to fire strongly. Possibly, none of the rules are appropriate to model those test points. Thus, we can reject a test point when the maximum value of the firing strength associated with a test point is less than a given threshold. We choose the value of this threshold based on only the training data.

3 Experimentation

3.1 Dataset Descriptions and Experimental Settings

We use two synthetic datasets: S-Curve and Swiss Roll shown in Figs. 1a and 2a, respectively. Both of them are three dimensional data consisting of 1000 points. They are generated using `scikit-learn` [13] package (version 0.19.2) of Python. We also use a real-world dataset, Iris. Iris has 150 datapoints from three classes in four dimensional space, such that, in each class there are 50 points.

We choose the number of fuzzy rules or the number of clusters as $0.05n$, where n is the number of points. In FCM clustering algorithm, the value of the fuzzifier and the maximum number of iterations are set to 2 and 200, respectively. We initialize the consequent parameters randomly in $[-0.5, 0.5]$. The spreads of the Gaussian membership functions in (6) are initialized as $0.2\beta_l$ and $0.3\beta_l$. Here, β_l

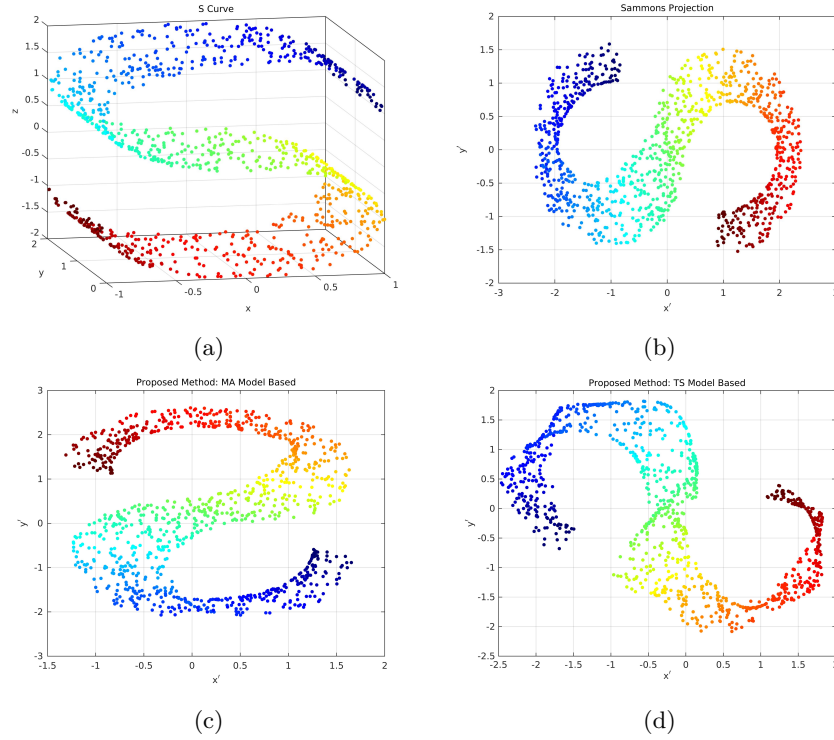


Fig. 1: Visualization of the S Curve data: (a) original, (b) Sammon's projection, (c) proposed method with MA model, (d) proposed method with TS model.

indicates the difference between the maximum and the minimum values of the l th feature. We repeat each experiment five times for each of these two initializations of the spreads. Out of these ten runs, the one with the minimum error (measured using (1)) is chosen as the output of the proposed algorithm. Note that, it is done automatically without any human intervention. To optimize the objective function in (1) using stochastic gradient descent, we use an optimizer, `train.GradientDescentOptimizer`, from a standard machine learning framework called `TensorFlow` [1]. The parameter `learning_rate` is set to 2.0 and 0.1, respectively, for the MA and TS models. Thus, in experiments with MA models, all the three involved learning rates η_v, η_σ , and η_h are equal to 2.0. Similarly for experiments with TS models, all the three involved learning rates η_v, η_σ , and η_c are equal to 0.1.

3.2 Results and Discussions

Figures 1b, 2b and 3a show the results with Sammon's method [14] on S-Curve, Swiss Roll and Iris datasets, respectively. The results of the proposed method with the MA model on the three datasets are illustrated in Figs. 1c, 2c, and 3b

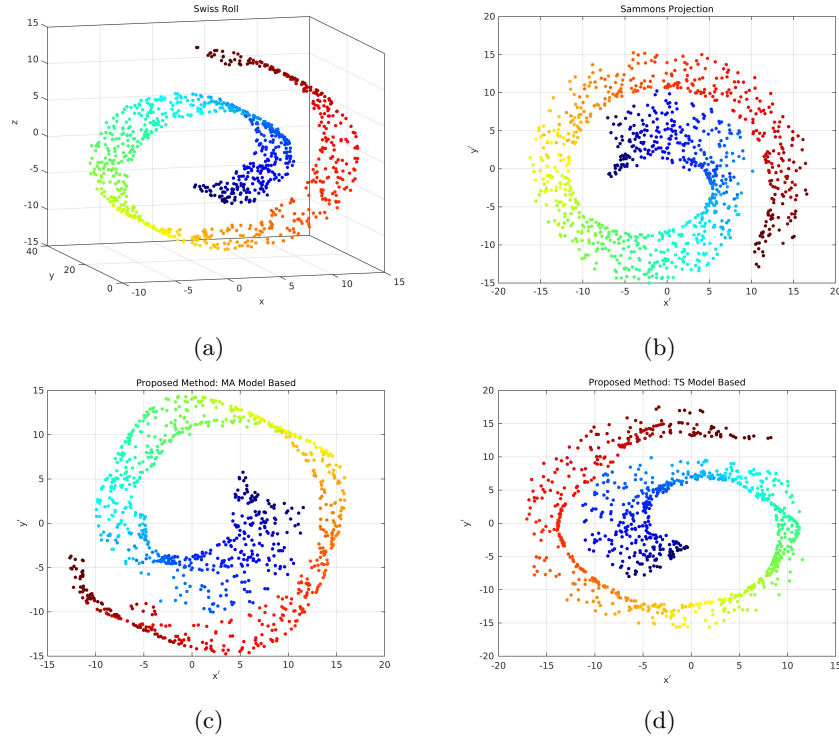


Fig. 2: Visualization of the Swiss Roll data: (a)original, (b)Sammon's projection, (c)proposed method with MA model, (d)proposed method with TS model.

respectively. Similarly, the outputs of the proposed method with the TS model are shown in Figs. 1d, 2d and 3c, respectively. From Figs. 1, 2, and 3, we observe that the proposed methods, both with MA and TS models, project the three datasets faithfully to a two dimensional space. For the synthetic datasets, we observe that the proposed method successfully preserves the underlying geometries as preserved in their Sammon's projection. We note that for S Curve, the MA model (with height method) produces somewhat better outcome than the TS model, whereas for Swiss Roll the TS model generates better projection than the MA model. In case of Iris dataset, separability of classes in the projected space is important. As seen in Figs. 3b and 3c the separability between the three classes are similar as in the case of Sammon's method (Fig. 3a). We note here that different initializations and use of different number of rules may produce different results. In addition to the graphical results, to assess the quality of the projection, we present the value of the Pearson's correlation coefficient between pairwise distances in the input space and in the output space in Table 1. For all the three data sets we obtained quite high value of the correlation coefficients.

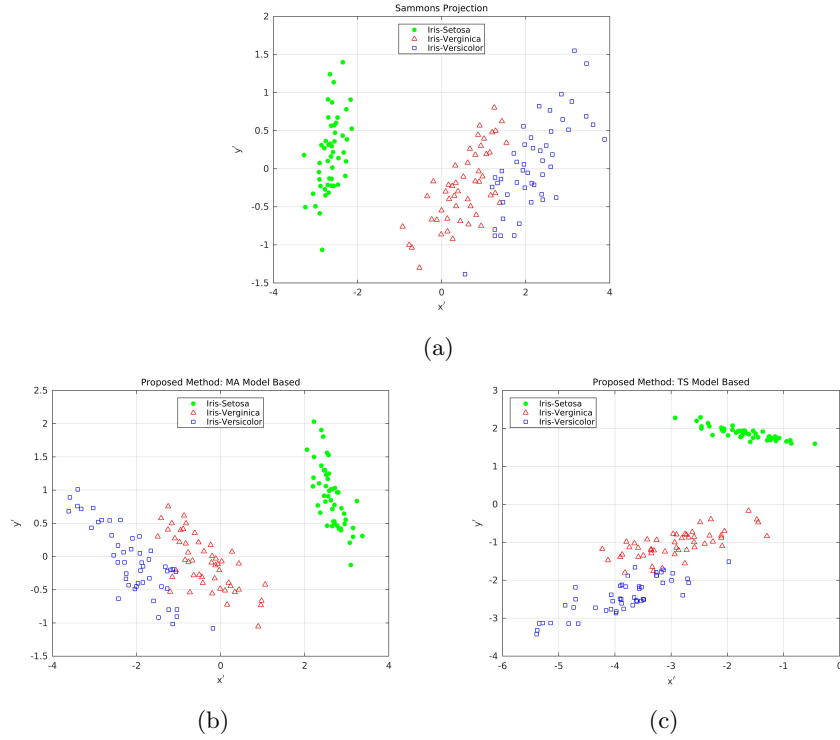


Fig. 3: Visualization of the Iris data: (a) Sammon's projection, (b) proposed method with MA model, (c) proposed method with TS model.

Table 1: Pearson's correlation coefficient between the pairwise distances in the input and output spaces

Dataset	MA Model	TS Model
S-Curve	0.9653	0.9620
Swiss Roll	0.8514	0.8594
Iris	0.9978	0.9969

This indicates that the relation among the pairwise distances of the inputs are mostly preserved in the obtained projections.

To illustrate the predictability of the proposed model, we perform an experiment with the S-Curve dataset. A contiguous portion of the dataset containing 50 points is removed and the rest of the points form a training dataset (see Fig. 4a). The removed 50 points serve as the test dataset. The MA model and the TS model based proposed schemes are used to learn the training data. The corresponding outputs are shown in Figs. 4c and 4e respectively. The full dataset

Table 2: Pearson’s correlation coefficient between the pairwise distances in the input and output spaces

Set	MA Model	TS Model
Train	0.9667	0.9611
Train and test together	0.9649	0.9565

containing both the training and the test points, is shown in Fig. 4b. To differentiate the training and the test points, we represent them by circle (o) and plus (+) signs, respectively. The predictions for the test points obtained by the MA model and the TS model based proposed method are shown in Figs. 4d and 4f, respectively. We observe that irrespective of the types of the model, the proposed method projects test points in the expected region. However, the prediction by the MA model based method seems more appropriate than that of the TS model based method. In Table 2, the values Pearson’s correlation coefficient between pairwise distances in input and output spaces for the training set and the training and test set together are presented. For both MA and TS models the correlations for the training data are similar to those for the cases when the training and test data are taken together.

4 Conclusion

In this work, we have proposed a fuzzy rule-based scheme for learning Sammon’s map explicitly. We have explored the MA model along with the height method of defuzzification. We have also used the TS model where consequents are a linear combination of input variables. For both the models the antecedent part is constructed by clustering the given input data. Antecedent and consequent parameters are modified using gradient descent search to obtain optimal values for minimizing the Sammon’s stress. The obtained results depict that the proposed system successfully produce the structure-preserving projection as that of the original Sammon’s method. It can also project the new (test) points in an expected manner. The framework is quite general and we can use objectives other than Sammon’s stress.

An issue with the proposed method is that the performance of the proposed method depends on the chosen number of clusters. Here, we have chosen the number of clusters based on our initial ad-hoc experiments. In future, we plan to investigate to address this issue. In our future works, we also plan to introduce regularizers to improve the quality of predictability. We have not investigated how the proposed method performs when it should reject the outputs. All these are left for our future work.

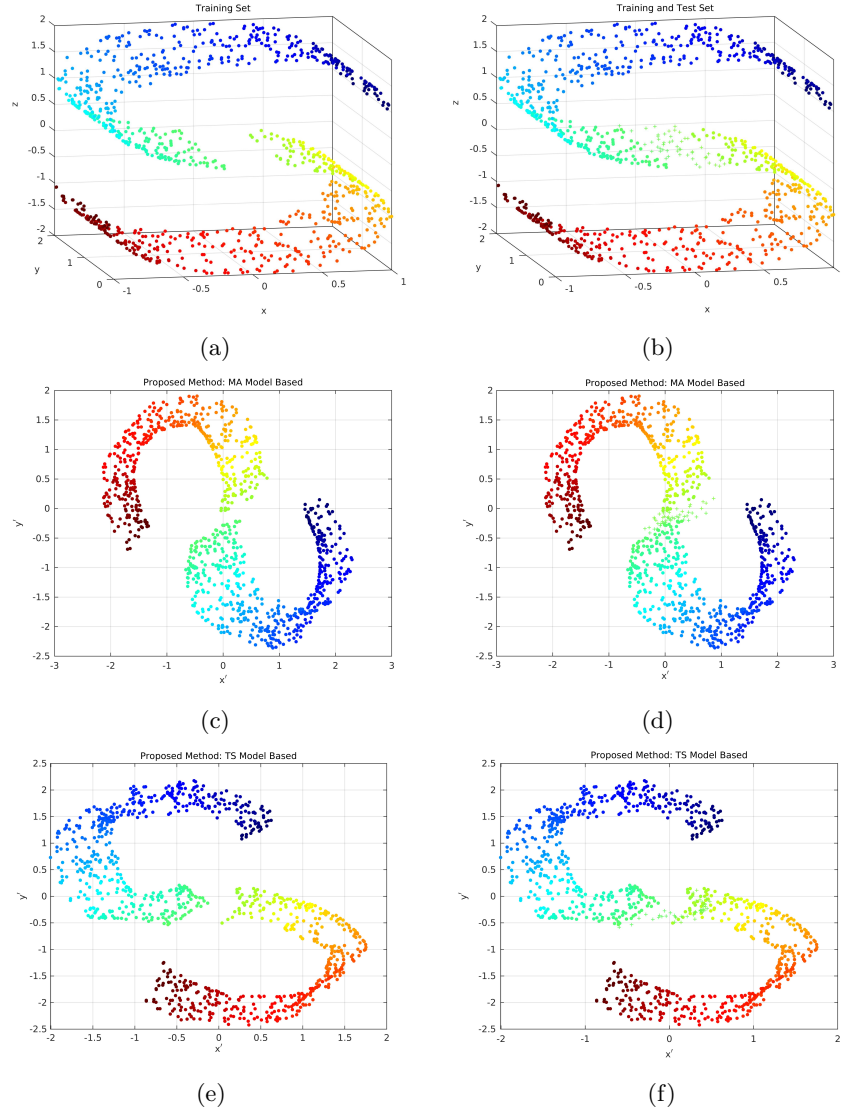


Fig. 4: Visualization in the original space (a) training data, (b) full data. Visualization using proposed method with MA model, (c) training data (d) full data with prediction. Visualization using proposed method with TS model, (e) training data, (f) full data with prediction.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J.,

- Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X.: Tensorflow: A system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). pp. 265–283 (2016), <https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf>
2. Álvarez-Meza, A.M., Lee, J.A., Verleysen, M., Castellanos-Dominguez, G.: Kernel-based dimensionality reduction using renyi’s α -entropy measures of similarity. *Neurocomputing* **222**, 36–46 (2017)
3. Bezdek, J.C., Ehrlich, R., Full, W.: Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences* **10**(2-3), 191–203 (1984)
4. Chen, Y.C., Pal, N.R., Chung, I.F.: An integrated mechanism for feature selection and fuzzy rule extraction for classification. *IEEE Transactions on Fuzzy Systems* **20**(4), 683–698 (2012)
5. Cordon, O., Herrera, F.: A two-stage evolutionary process for designing tsf fuzzy rule-based systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **29**(6), 703–715 (1999)
6. Driankov, D., Hellendoorn, H., Reinfrank, M.: An introduction to fuzzy control. Springer Science & Business Media (2013)
7. Gupta, M.M., Qi, J.: Theory of t-norms and fuzzy inference methods. *Fuzzy sets and systems* **40**(3), 431–450 (1991)
8. Jain, A.K., Mao, J.: Artificial neural network for nonlinear projection of multivariate data. In: *Neural Networks, 1992. IJCNN., International Joint Conference on*. vol. 3, pp. 335–340. IEEE (1992)
9. Mao, J., Jain, A.K.: Artificial neural networks for feature extraction and multivariate data projection. *IEEE transactions on neural networks* **6**(2), 296–317 (1995)
10. Mitra, S., Hayashi, Y.: Neuro-fuzzy rule generation: survey in soft computing framework. *IEEE transactions on neural networks* **11**(3), 748–768 (2000)
11. Pal, N.R., Eluri, V.K., Mandal, G.K.: Fuzzy logic approaches to structure preserving dimensionality reduction. *IEEE Transactions on Fuzzy Systems* **10**(3), 277–286 (2002)
12. Pal, N.R., Saha, S.: Simultaneous structure identification and fuzzy rule generation for takagi–sugeno models. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **38**(6), 1626–1638 (2008)
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011)
14. Sammon, J.W.: A nonlinear mapping for data structure analysis. *IEEE Transactions on computers* **100**(5), 401–409 (1969)
15. Talwalkar, A., Kumar, S., Rowley, H.: Large-scale manifold learning. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. pp. 1–8. IEEE (2008)
16. Van Der Maaten, L., Postma, E., Van den Herik, J.: Dimensionality reduction: a comparative. *J Mach Learn Res* **10**, 66–71 (2009)
17. Wang, L.X., Mendel, J.M.: Generating fuzzy rules by learning from examples. *IEEE Transactions on systems, man, and cybernetics* **22**(6), 1414–1427 (1992)
18. Wang, Y., Feng, K., Chu, X., Zhang, J., Fu, C.W., Sedlmair, M., Yu, X., Chen, B.: A perception-driven approach to supervised dimensionality reduction for visualization. *IEEE transactions on visualization and computer graphics* **24**(5), 1828–1840 (2018)