

Aberystwyth University

Fuzzy Rule Interpolation with K-Neighbours for TSK Models

Zhang, Pu; Shang, Changjing; Shen, Qiang

Published in:

IEEE Transactions on Fuzzy Systems

DOI:

[10.1109/TFUZZ.2021.3136359](https://doi.org/10.1109/TFUZZ.2021.3136359)

Publication date:

2021

Citation for published version (APA):

Zhang, P., Shang, C., & Shen, Q. (2021). Fuzzy Rule Interpolation with K-Neighbours for TSK Models. *IEEE Transactions on Fuzzy Systems*. <https://doi.org/10.1109/TFUZZ.2021.3136359>

Document License

CC BY

General rights

Copyright and moral rights for the publications made accessible in the Aberystwyth Research Portal (the Institutional Repository) are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Aberystwyth Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Aberystwyth Research Portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

tel: +44 1970 62 2400
email: is@aber.ac.uk

Fuzzy Rule Interpolation with K-Neighbours for TSK Models

Pu Zhang, Changjing Shang and Qiang Shen

Abstract—When a fuzzy system is presented with an incomplete (or sparse) rule base, fuzzy rule interpolation (FRI) offers a useful mechanism to infer conclusions for unmatched observations. However, most existing FRI methodologies are established for Mamdani inference models, but not for Takagi-Sugeno-Kang (TSK) ones. This paper presents a novel approach for computing interpolated outcomes with TSK models, using only a small number of neighbouring rules to an unmatched observation. Compared with existing methods, the new approach helps improve the computational efficiency of the overall interpolative reasoning process, while minimising the adverse impact on accuracy induced by firing those rules of low similarities with the new observation. For problems that involve a rule base of a large size, where closest neighbouring rules may be rather alike to one another, a rule-clustering-based method is introduced. It derives an interpolated conclusion by first clustering rules into different groups with a clustering algorithm and then, by utilising only those rules that are each selected from one of a given, small number of closest rule clusters. Systematic experimental examinations are carried out to verify the efficacy of the introduced techniques, in comparison with state-of-the-art methods, over a range of benchmark regression problems, while employing different clustering algorithms (which also shows the flexibility in ways of implementing the novel approach).

Index Terms—TSK models, Fuzzy rule interpolation, Rule clustering, K-nearest rules, Nearest rule clusters.

I. INTRODUCTION

One successful methodology to implement practically effective knowledge-based systems is through the use of fuzzy rule based reasoning. The underlying idea of a fuzzy rule-based system is to represent domain knowledge in the form of fuzzy “if-then” rules by employing fuzzy sets to describe imprecise variable values. These rules may be directly provided by domain experts or obtained via data-driven learning. A fuzzy system works generally by checking whether a novel input matches the antecedent of any rule in the set of collected rules (namely, the rule base), with the output computed as the corresponding rule consequent [1, 2]. Particularly, supported with fuzzy logic and fuzzy set theory, imprecise and linguistic terms can be readily described and their associative relationships explicitly represented, enabling the inference process to be performed that resembles human reasoning.

Different types of fuzzy rule inference system exist in the literature. Takagi-Sugeno-Kang (TSK) models [3] are one of the conventional that are the most applied. Within such

a model, fuzzy sets are used to depict the values of rule antecedents and polynomials to describe the consequents. Firing a rule of this form results in crisp conclusions. As such, TSK models are very suitable for solving regression and prediction problems, over continuously-valued domains.

Whilst being generally powerful, conventional rule based systems all suffer from an important limitation, be they fuzzy or not. That is, if the input domain is not completely covered, a novel observation may not always match any rule in the given rule base. In this case, they are unable to produce any conclusion by applying any of the classical rule-firing methods. Such rule bases are termed sparse ones (although this is often taken to simply imply an incomplete rule base) in the literature. To rectify, or at least to reduce the adverse impact of this limitation, fuzzy rule interpolation (FRI) has been introduced [4]. If a newly presented input or observation does not match any of the rules available, FRI can help by generating an intermediate rule through the approximation of those rules close to the observation, from which a potentially relevant conclusion may then be obtained. FRI has been widely applied in performing practical pattern recognition problems and has obtained satisfactory results, such as computer vision and image processing [5]; medical diagnosis [6] and risk analysis [7]; and cyber security and network security [8].

A good number of FRI approaches have been established over the past few decades. Just considering the popular family of transformation-based fuzzy rule interpolation (T-FRI) techniques [9] that generally follow the seminal work on linear interpolation [10], there have been many distinct FRI mechanisms reported in the literature, including: adaptive interpolation [11], higher-order interpolation [12], and weighted FRI techniques [13]. However, these exemplified techniques are all developed for Mamdani models rather than for TSK models.

TSK inference extension (TSK+) [14] is a recently proposed fuzzy interpolative reasoning method for fuzzy systems employing TSK models. Instead of relying upon computing the matching degrees, it uses a distance metric-based similarity measure to perform interpolative reasoning, by manipulating all rules contained within the rule base. In so doing, when an input matches no rule, a certain output is still obtained. Although TSK+ offers a useful means for innovative inference, it has its own shortcomings. Particularly, it is not sufficiently efficient for many practical applications given its nature of fundamentally requiring the use of all given rules, incurring significant computational overheads. Besides, redundant or possibly, irrelevant rules are also included in any attempt to compute the output. This may introduce undesirable biases into

P. Zhang, C. Shang and Q. Shen are with the Department of Computer Science, Aberystwyth University, Aberystwyth SY23 3DB, U.K. (e-mail: {puz, cns, qqs}@aber.ac.uk)

This work is partly supported by the British Council under grant number: 623718881.

the final interpolated outcomes, thereby reducing the system accuracy.

To address these limitations, a different approach, interpolation with (just) K closest neighbours, is proposed in this paper. The work presents two advancements in developing FRI methodologies for TSK models, through two novel implementations of the approach: (i) interpolation with K Closest Rules (KCR) for sparse rule bases of a small size, and (ii) interpolation with K Closest Rule Clusters (CRC) for those of a large size. The underlying principle for both is to perform interpolation using a small number, K of distinctive rules close to an unmatched observation. This follows the common practice as with the state-of-the-art techniques developed for Mamdani models. In so doing, rules with low similarities to an unmatched observation are not fired and hence, their adverse impact on model accuracy minimised, while incurring less computation. In addition, the problem of lacking diversity of rules being involved in subsequent interpolation, caused by the situation where a large-sized sparse rule base is present but the K nearest rules may be very similar with one another, is resolved. To provide flexibility in implementing the proposed approach, ensuring that it does not rely on any specific clustering algorithm, the implementation for the second method is herein systematically evaluated using five different clustering techniques.

To have a fair comparison over different methods, a range of experimental studies are systematically carried out on different benchmark datasets. Statistic analyses of the results demonstrate that KCR improves the performance over TSK+ and CRC for cases involving sparse rule bases of a small size, and that for cases involving rule bases of a large size, CRC outperforms TSK+ and KCR. Of course, both KCR and CRC offer superior results over the existing approach TSK+. The most recent work as reported in [15] provides a novel approach that makes it possible to automatically select fuzzy rules for interpolation. Having recognised this, experimental studies are also conducted to compare KCR and CRC with this existing approach, with favourable results.

The remainder of this paper is structured as follows. For academic completeness, Section II briefly reviews the inference process of the conventional fuzzy systems that use a TSK model, the procedure of TSK+, and the five popular clustering algorithms that are subsequently used for rule clustering. Sections III and IV detail the two aforementioned methods, respectively introduced for interpolation with K closest rules and that with K closest rule clusters. Section V presents and discusses the experimental results, in comparison with state-of-the-art alternatives. Finally, Section VI concludes the paper and points out interesting future work.

II. BACKGROUND

This section presents the directly relevant background material, including an outline of TSK fuzzy inference systems and TSK+, and a brief description of five popular clustering algorithms each of which may be used to facilitate the rule clustering-based interpolation method.

A. TSK Model-Based Fuzzy Reasoning Systems

The TSK fuzzy systems were originally presented by Takagi, Sugeno, and Kang in 1985 [3]. A TSK model uses fuzzy sets to represent rule antecedents and polynomials to do the consequents, with the computed conclusions represented in crisp values. As presented in Fig. 1, for an unknown observation, the TSK model-based system first calculates the matching degrees between the observation and the rule antecedent in each rule. The weight of a certain rule is then determined by an integrating operation (usually implemented by a minimum operator) on the resulting individual matching degrees. The final outcome is computed by the weighted average of the corresponding rule consequents. A more detailed description of a TSK system and its working is outlined in Alg. A1, as provided in Appendix A.

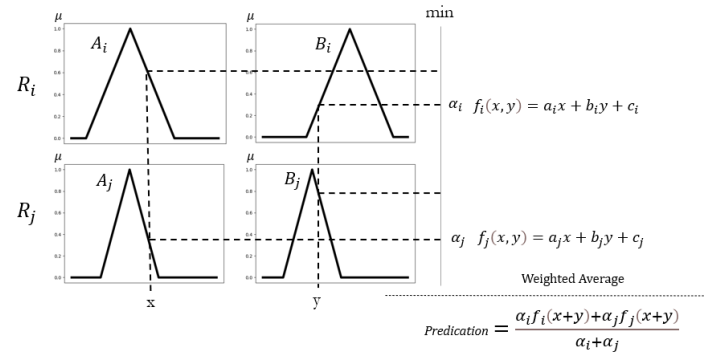


Fig. 1. Illustrative example of TSK inference process.

If a new observation does not match any rule, the weight α_i that is calculated for each rule will be 0. This implies that neither sub-conclusion nor final result can be derived using traditional approximate reasoning. Thus, any attempt to utilise the conventional TSK model will fail. This is due to the fact that the system utilises a sparse rule base, or more precisely, an incomplete fuzzy rule base, namely it does not cover the full problem domain. As indicated previously, FRI has been developed to compute the corresponding conclusions for such unmatched observations, by exploiting the approximation of the rules close to them, although the substantial majority of the existing techniques are aimed at interpolative reasoning with Mamdani models [4].

B. TSK Inference Extension (TSK+)

TSK+ [14] represents the current state of the art, providing a fuzzy inference mechanism that extends the original TSK inference to cope with sparse rule bases. Different from the conventional TSK approach that works based on calculating the matching or overlapping degrees with an observation, TSK+ works by employing a similarity measure modified from the Euclidean distance metric [16] to assess any potential relationships between the observation and every given rule. The similarities so measured between the observation and each rule are always positive (with a similarity degree larger than zero). Therefore, each and every rule in the given rule base will be involved in the computation of the inferred conclusion.

In so doing, if an observation matches no rules at all in the rule base, an approximate conclusion is still estimated. The inference procedure of TSK+ is summarised in Alg. A2, in Appendix A.

Fundamentally, TSK+ applies all given rules. This clearly is not efficient, especially for problems that involve a large (however incomplete) rule base. More importantly, redundant or even possibly irrelevant rules are also included in every attempt to compute the final output. This may introduce a certain undesirable bias or noise into the computation process for the generation of the conclusion, further to the introduction of significant computational overheads.

C. Clustering Algorithms

A brief introduction to five popularly used and readily available clustering algorithms is given here. Each of these can be adopted to implement the rule clustering procedure that is required in the subsequent development when facing a large-sized rule base. Any one of these may be employed to carry out the intended task, but these are collectively reviewed to facilitate comparison, in an effort to make an informed choice of the potentially most suitable.

1) *K-means* [17]: As one of the most widely used fuzzy clustering algorithms, it clusters instances into K groups by iteratively updating cluster centres and assigning instances to their closest centres. The underlying objective function (known as the inertia or within-cluster sum-of-squares error) is defined as follows:

$$J_w(U, V) = \sum_{j=1}^N \sum_{i=1}^K \|x_{ji} - v_i\|^2 \quad (1)$$

where U denotes the set of data instances, x_{ji} expresses that the j -th instance belongs to the i -th cluster, V stands for the set of cluster centres and $v_i \in V$, $\|x_j - v_i\|^2$ denotes the Euclidean distance between the object x_{ji} and the centre v_i , K is the number of cluster centres, and N is the number of instances.

This algorithm requires the number of clusters to be pre-specified. Many methodologies have been introduced in the literature to determine the number of clusters, K , such as those through the use of: the Elbow method [18], the Silhouette-based technique [19], and the Bayesian Information Criterion [20]. The Elbow method is fast and effective, as it determines the value of K simply based on the criterion that adding another cluster does not lead to much improved modelling outcome (with respect to the above objective function), and hence it is employed in this paper.

2) *Gaussian Mixture Models (GMM)* [21]: Being another classical clustering algorithm, this method works by presuming that the distribution of instances conforms to the linear combination of multiple Gaussian distribution functions. This combination function is defined by:

$$p(x_j) = \sum_{i=1}^K \pi_{ji} \mathcal{N}(x_j | \mu_i, \sigma_i) \quad (2)$$

where x_j denotes an instances to be clustered, π_{ji} represents the probability that the instance x_j belongs to the i -th cluster,

and μ_i and σ_i stand for the mean and standard deviation of the i -th Gaussian model respectively.

Theoretically, this model can fit any type of data distribution. The expectation maximisation (EM) algorithm [22] is the most commonly used algorithm to construct GMM. In particular, to identify the optimal partitions of GMM, the log-likelihood function to be maximised is given by

$$\log\left(\prod_{j=1}^N p(x_j)\right) = \sum_{j=1}^N \log\left(\sum_{i=1}^K \pi_{ji} \mathcal{N}(x_j | \mu_i, \sigma_i)\right) \quad (3)$$

The corresponding parameters are updated iteratively such that:

$$\begin{aligned} \mu_i &= \frac{1}{N_i} \sum_{j=1}^N \gamma_{ji} x_j \\ \sigma_i &= \frac{1}{N_i} \sum_{j=1}^N \gamma_{ji} (x_j - \mu_i)(x_j - \mu_i)^T \\ \pi_{ji} &= \frac{N_i}{N} \end{aligned} \quad (4)$$

where $N_i = \sum_{j=1}^N \gamma_{ji}$ and γ_{ji} represents the conditional probability which is calculated by:

$$\gamma_{ji} = \frac{\pi_{ji} \mathcal{N}(x_j | \mu_i, \sigma_i)}{\sum_{i=1}^K \pi_{ji} \mathcal{N}(x_j | \mu_i, \sigma_i)} \quad (5)$$

3) *Fuzzy C-means (FCM)* [23]: This conventional clustering algorithm has been popularly applied in dealing with various problems (e.g., for fuzzy rule base generation [13] and social network modelling [24]). Unlike any crisp clustering method (say, K-means and GMM), FCM allows a data instance to belong to different clusters at the same time with different membership degrees. It works by assigning each instance a membership degree to every cluster, based on the measurement of the distance between the instance and an individual cluster centre. The closer an instance to a cluster centre, the higher the membership degree. To identify the optimal partitions of FCM, the objective function to be minimised is defined by

$$J_w(U, V) = \sum_{j=1}^N \sum_{i=1}^K (u_{ij})^w \|x_j - v_i\|^2 \quad (6)$$

where x_j denotes an instance, w is a parameter that signifies the weight of each element, K stands for the number of cluster centres, N is the number of instances, V is the set of cluster centres with $v_i \in V$, U is the matrix of membership degree and $u_{ij} \in U$ represents the membership degree of the instance x_j belonging to the cluster with the centre v_i , and $\|x_j - v_i\|^2$ expresses the similarity between the instance x_k and the centre v_i .

The procedure of this algorithm is summarised in Alg. A3, in Appendix A. Particularly, the membership u_{ik} and the centre v_i are updated iteratively such that

$$v_i = \frac{\sum_{j=1}^N (u_{ij})^w x_j}{\sum_{j=1}^N (u_{ij})^w} \quad (7)$$

$$u_{ij} = \left(\sum_{n=1}^K \left(\frac{\|x_n - v_i\|}{\|x_j - v_n\|}\right)^{2/(w-1)}\right)^{-1} \quad (8)$$

4) *Kernel Fuzzy C-means (K-FCM) [25]*: Being an extension to the standard FCM, it works by applying a kernel-induced distance measure to replace the original Euclidean distance. Kernel function is a nonlinear mapping that transforms a low dimensional input data space into a feature space with a much higher dimension, aiming at turning the original nonlinear problem into a potentially linear one so as to facilitate problem solving [26]. The following Gaussian radial basis function (GRBF) is one of the commonly used kernel functions, and the one employed in this paper, because no additional parameters are required:

$$KN(x_j, v_i) = \exp \frac{-\|x_j - v_i\|^2}{\sigma^2} \quad (9)$$

where x_j denotes an instance, v_i stands for a cluster centre and σ^2 represents the variance of the instance.

As it is a direct extension of FCM, the algorithm procedure is omitted here. However, note that adapted from its original, the underlying objective function is now:

$$J_w(U, V) = \sum_{j=1}^N \sum_{i=1}^K (u_{ij})^w (1 - KN(x_j, v_i)) \quad (10)$$

with Eqn. 7 and Eqn. 8 respectively transformed to:

$$v_i = \frac{\sum_{j=1}^N (u_{ij})^w KN(x_j, v_i) x_j}{\sum_{j=1}^N (u_{ij})^w KN(x_j, v_i)} \quad (11)$$

$$u_{ij} = (\sum_{n=1}^K \left(\frac{1 - KN(x_j, v_i)}{1 - KN(x_j, v_n)} \right)^{1/w-1})^{-1} \quad (12)$$

5) *Suppress Fuzzy C-means (S-FCM) [27]*: This is another extension of FCM, following the motivation for improving on its convergence speed [28]. It is developed on the basis of the rival-checked fuzzy c-means clustering algorithm (RCFCM) [29] that speeds up FCM with competitive learning capacity. The underlying mechanism of this approach is to magnify the largest membership degree, u_{pj} while suppressing the others. In order to achieve such an objective, a membership modifying mechanism is added after iteratively updating the membership degrees U , such that

$$u_{pj} \leq 1 - \alpha \sum_{i \neq p} u_{ij} = 1 - \alpha + \alpha u_{pj} \quad (13)$$

$$u_{ij} \leq \alpha u_{ij}$$

where $u_{pj} > u_{ij}, i \neq p$ and $0 \leq \alpha \leq 1$.

III. INTERPOLATION WITH K CLOSEST RULES (KCR)

From the specification as well as the application of TSK+, it is easy to reveal that those rules nearest to an unmatched observation generally have a much higher similarity degree than others. This indicates that the interpolated outcomes may be mainly determined by those closest rules, with the rest typically contributing substantially less. Note that in the FRI literature, it is often assumed that the interpolated conclusion is estimated by a certain aggregation of those neighbouring rules to the observation [4]. That is, the nearest rules are (normally correctly) considered to contain the most relevant information

whilst those rules far away from the observation are less relevant. Indeed, distant rules may introduce adverse biases into the results, with their use becoming counter-productive. As far-away rules generally have relatively smaller similarity measures against the observation, such biases do not necessarily impose much influence upon the interpolated results, but they do induce significant computational overheads if there are many rules in the rule base (despite its incompleteness). Thus, such biases or the use of remote rules should be minimised, for both the efficiency and the effectiveness of the interpolative reasoning process.

To address the aforementioned issue, a significantly revised inference procedure, termed *interpolation with K closest rules (KCR)* is introduced in this section. It employs the same similarity measure (17) as the one used in TSK+. The underpinning idea is that only K nearest neighbouring rules to a given unmatched input are exploited in performing the interpolation, rather than involving all the rules in the sparse rule base.

A. KCR Algorithm

Without losing generality, let a TSK sparse fuzzy rule base consist of m rules with each involving n antecedent variables and being defined by

$$R_i : \text{if } x_1 \text{ is } A_{i1}, \dots, x_n \text{ is } A_{in}, \quad (14)$$

$$\text{then } f_i(x_1, \dots, x_n) = a_{i0} + a_{i1}x_1 + \dots + a_{in}x_n$$

where A_{i1}, \dots, A_{in} are the fuzzy sets respectively taken by the rule antecedent variables x_1, \dots, x_n and $a_{i0}, a_{i1}, \dots, a_{in}$ are the parameters specifying the polynomials of a rule's consequent.

Given an observation $O(B_1, \dots, B_n)$, the KCR algorithm can be summarised as shown in Alg. 1.

Algorithm 1: Interpolation with K Closest Rules

Input: Rule base $\{R_j\}$; Unmatched observation O

Output: Interpolated conclusion $f(B_1, \dots, B_n)$

- 1) Computation of Euclidean distances between individual rules and given observation O .
- 2) Selection of K nearest rules with Quickselect [30] (which is exploited purely for efficiency).
- 3) Computation of similarity measures between O and each R_i that is one of selected K nearest rules: $S(A_{i1}, B_1), \dots, S(A_{in}, B_n)$
- 4) Determination of weight of each such rule R_i : $\alpha_i = S(A_{i1}, B_1) \wedge \dots \wedge S(A_{in}, B_n)$
- 5) Integration of resultant K similarities to obtain interpolated rule with consequent parameters:

$$a_0 = \frac{\sum_{i=1}^K \alpha_i a_{i0}}{\sum_{i=1}^K \alpha_i}, \dots, a_n = \frac{\sum_{i=1}^K \alpha_i a_{in}}{\sum_{i=1}^K \alpha_i} \quad (15)$$

- 6) Execution of interpolated rule to yield interpolated consequent: $f(B_1, \dots, B_n) = a_0 + a_1 B_1 + \dots + a_n B_n$
 - 7) Return: $f(B_1, \dots, B_n)$.
-

B. Similarity Measurement

Algorithm 1 works via the employment of similarity measures. To offer a flexibility in practical applications, as well as to provide an opportunity for comparative evaluation of the algorithm, two distinct similarity measurement methods are introduced here.

1) *Similarity with Distance Factor (Similarity-DF)*: This is the same as with similarity measurement means utilised in TSK+. It is a revised version of the measurement presented in [16], with a distance factor (*DF*) employed to reinforce its sensitivity. Note that as empirically shown in the literature (e.g., [31]), the use of what type of membership function has little impact upon the outcomes of fuzzy rule-based inference provided that the membership functions are appropriately tuned with training data. From this observation and also, for computational simplicity, this work utilises triangular membership functions to represent fuzzy values unless otherwise stated.

For illustration, suppose that two normalised fuzzy sets are given, represented by triangular membership function $A = (a_1, a_2, a_3)$ and $A' = (a'_1, a'_2, a'_3)$ [32]. Then, the similarity degree between these two fuzzy values $S(A, A')$ is defined and computed by

$$\begin{aligned} S(A, A') &= (1 - d(A, A')) \cdot DF \\ DF &= 1 - \frac{1}{1 + e^{-s\alpha + \beta}} \end{aligned} \quad (16)$$

where $d(A, A') = \frac{\sum_{i=1}^3 |a_i - a'_i|}{3}$ and α represents the Euclidean distance between the gravity centres (namely, the representative values [9]) of the two fuzzy values, s is an adjustable parameter that determines the sensitivity of the similarity measure to the distance measure (the larger the value of s the more sensitive DF to α), and β is a sufficiently large integer (which is empirically set to 5) to ensure that DF is approximately normalised as 1 when d becomes 0.

2) *Similarity Based on Distance Measures (Similarity-d)*: As one of the most widely applied similarity measurement method [33], this measures the similarity between two fuzzy sets directly based on the inverse of the distance metric between them:

$$S(A, A') = \frac{1}{1 + d(A, A')} \quad (17)$$

where, again, $d(A, A') = \frac{\sum_{i=1}^3 |a_i - a'_i|}{3}$.

Following the above specifications of these similarity measurement methods, it is clear that the larger the value of $S(A, A')$, the nearer and hence, the more similar the two fuzzy sets A and A' . In particular, $S(A, A')$ reaches the maximum value if and only if A and A' are identical. Owing to their generality, both can be effective and applicable to capture and reveal the similarities between fuzzy sets (as to be experimentally verified later, while Similarity-DF generally performs better than Similarity-d, at the cost of a little extra computation).

C. KCR Complexity

In Alg. 1, two distance metrics are exploited. The first is for the use of Euclidean metric to efficiently determine nearest rules, without resorting to the more complicated similarity measurement. The second utilisation takes place to find the similarity measures. However, in this latter use, it only plays a small part in helping capture the essential relationships between an observation and the rules. Importantly, the similarity measurement and hence, the second round of application of distance metric is only applied K times for K selected rules rather than for all the rules in the rule base. Typically, K is substantially smaller than the number of the rules available. As such, the proposed approach significantly reduces the running time that would otherwise have to be taken if TSK+ is applied. Furthermore, the employment of Quickselect helps reduce the effort required to select the nearest rules.

Indeed, given m rules each involving n antecedent attribute, the time complexity of KCR is $O(mK + nK)$, where $O(mK)$ is the time complexity taken to implement the selection of K closest rules. In comparison, TSK+ has a complexity of $O(mn)$ since all rules are fired to derive the final conclusion. Note however that generally, K is much smaller than m or n . Thus, the proposed approach has a significantly lower time complexity.

IV. INTERPOLATION WITH K CLOSEST RULE CLUSTERS (CRC)

KCR is efficient. However, when KCR is applied to solve problems that involve a large-sized sparse rule base, the K nearest rules with the greatest similarity degrees may appear to be rather more similar amongst themselves than the rest. This may be expected intuitively, as illustrated in Fig. 2. Thus, if only these K rules are taken to implement interpolation, the results will be also similar to the linear combination of their consequents regardless of what similarity measures may be. Of course, this potential problem is not unique to KCR, but it can arise in TSK+ as well, despite that all rules are involved in rule interpolation there. This is simply because the similarities of the selected K rules are much larger than those measured over the rest. That is, the final interpolated result in TSK+ is also mainly determined by those nearest ones.

The above analysis prompts the need to extend the diversity of rules selected for use in the rule interpolation process, in an effort to avoid the involvement of far too many similar rules. Driven by this consideration, a clustering-aided interpolative reasoning process is presented here, termed *interpolation with K closest rule clusters (CRC)* hereafter.

In CRC, to maximise computational efficiency, all fuzzy values appearing in the rule antecedents are approximately represented using their representative values, which are firstly clustered into different groups, by the use of a clustering method. Those in the same cluster can be intuitively regarded as containing similar information about the mappings between the antecedents and the consequents; after all, they have been deemed to belong to the same cluster. Then, K clusters nearest to an unmatched observation are selected, with K being a small number, where the distance between a cluster

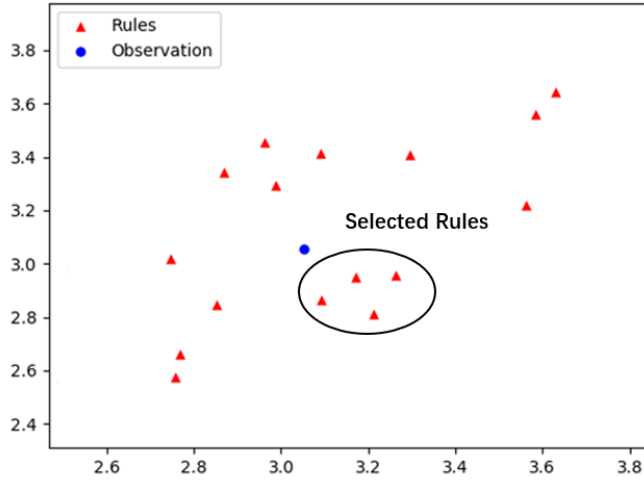


Fig. 2. Rules selected with respect to unmatched observation using KCR, where rule antecedents and observation are represented by their representative values (implemented as centres of gravity).

and an observation is determined by the use of the Euclidean measurement between the cluster centre and the observation. From each selected cluster, the rule that is the closest to the observation is then taken as an element of a set of K nearest rules to be used for interpolation. Thus, rules measured without necessarily having the higher similarity measures are able to contribute to the creation of the final interpolated consequent.

Note that through the afore-described rule selection process, the rule that is of the overall closest distance to the observation is always included to participate in rule interpolation. This is obvious as it always is the representative of a certain cluster of rules since it has the greatest similarity to the observation amongst the entire rule base. Fig. 3 presents an illustrative example of rule clusters and rules selected from the closest clusters that are produced by the CRC algorithm.

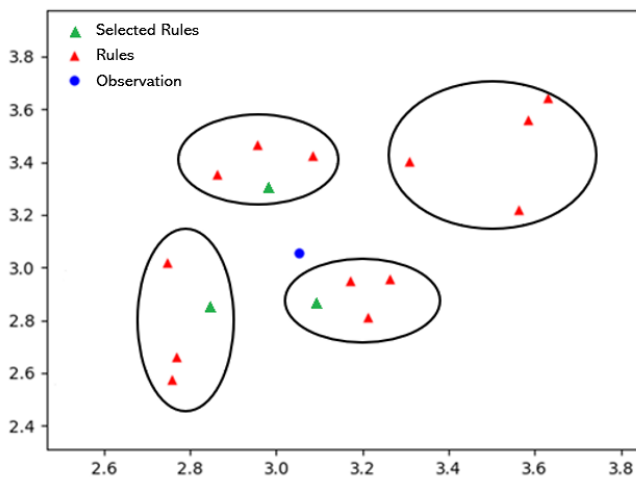


Fig. 3. Rules selected with respect to unmatched observation using CRC, where rule antecedents and observation are represented by their representative values (implemented as centres of gravity).

A. CRC Algorithm

The above-described intuition for the development of the CRC procedure is summarised as given in Alg. 2, where the similarity measurements are obtained in the same way as with KCR. In this algorithm, it is assumed, without losing generality, that a sparse rule base contains m rules (with each specified as per Eqn. 14) and an observation $O(B_1, \dots, B_n)$ (which does not match any of the rules) are given. Note that any of the five different clustering algorithms outlined in Section II-D (and indeed many other clustering methods if preferred) can be employed here to perform rule clustering.

Algorithm 2: Interpolation with K Closest Rule Clusters

Input: Rule base $\{R_i\}$; Unmatched observation O

Output: Interpolated conclusion $f(O)$

- 1) Clustering all rules into C different clusters using representative values of involved fuzzy values throughout.
- 2) Computation of Euclidean distances between observation O and all centres of the C clusters and selection of K nearest clusters (with K being normally much smaller than C).
- 3) Calculation of distances between O and each rule of cluster C_i that is taken from K clusters and that has not been considered so far, $i \in \{1, \dots, K\}$.
- 4) Selection of nearest rule R_i in C_i as its representative.
- 5) Determination of weight of rule R_i via similarity measures: $\alpha_i = S(A_{i1}, B_1) \wedge \dots \wedge S(A_{in}, B_n)$
- 6) Reiteration of Steps 3, 4 and 5 for all K selected clusters, returning K rules and corresponding similarities.
- 7) Integration of K similar rules to obtain interpolated rule, resulting in consequent parameters:

$$a_0 = \frac{\sum_{i=1}^K \alpha_i a_{i0}}{\sum_{i=1}^K \alpha_i}, \dots, a_n = \frac{\sum_{i=1}^K \alpha_i a_{in}}{\sum_{i=1}^K \alpha_i} \quad (18)$$

- 8) Execution of interpolated rule with O , resulting in final consequent outcome:

$$f(B_1, \dots, B_n) = a_0 + a_1 B_1 + \dots + a_n B_n$$

- 9) Return: $f(B_1, \dots, B_n)$.
-

B. CRC Complexity

The time complexity of the proposed CRC procedure is $O(KC + KG + nK)$, where $O(KC)$ and $O(KG)$ represent the complexity incurred for K clusters selection and that for K rules selection with one from each cluster, respectively; and G is the largest number of the rules contained within any single cluster. Compared with KCR, whose time complexity is $O(mK + nK)$, CRC can also help to reduce the computation cost incurred to perform similarity measurement. CRC does not require the computation of the distances between the observation O and all given rules, but only those from O to the centres of clusters and those to the rules in the K selected clusters. In so doing, $O(KC + KG)$ is in general, smaller than

$O(mK)$. In other words, the time complexity can generally be lower than that of CRC. Thus, the complexity of the proposed approach is further reduced in comparison to that of TSK+.

C. Integration of KCR and CRC

Both KCR and CRC procedures can be integrated into a single algorithm, in conjunction with the conventional inference mechanism for TSK models. This integration is straightforward, as shown in Fig. 4. It works by the use of just a small number of closest rules to infer the final conclusion while the conventional method is stuck when an observation matches no rules.

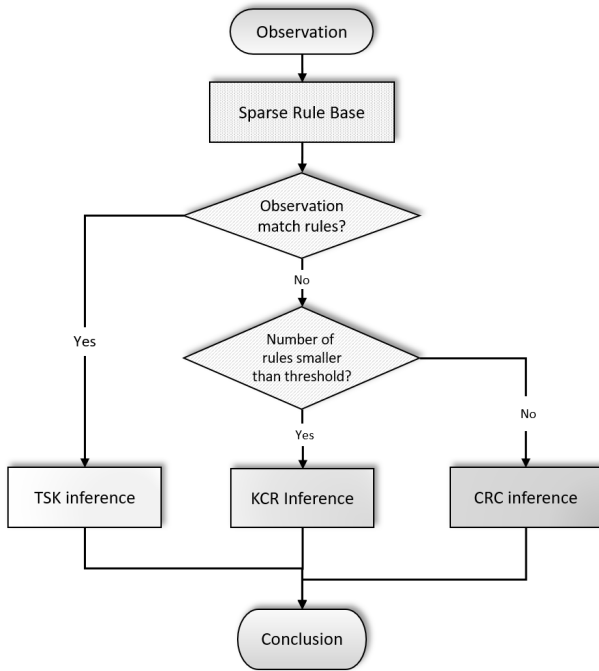


Fig. 4. TSK fuzzy inference enhanced with two rule interpolation procedures.

V. EXPERIMENTAL EVALUATION

The performance of the above-introduced novel FRI approach for TSK models is experimentally evaluated, in comparison to the state-of-the-art techniques, including the aforementioned TSK+ and the automated rule selection (AutoRS) based method [15], over ten benchmark datasets. The robustness and effectiveness of the presented approach are also demonstrated by observing the consistency and efficiency of utilising different clustering methods in supporting CRC.

A. Experimental Setup

1) *Datasets used:* The datasets run include one nonlinear mathematical model and nine real-world benchmark datasets (for regression problems) that have been taken from the UCI machine learning [34], function approximation [35] and evolutionary learning repositories [36]. The details of these employed benchmark datasets are summarised in Table I. For illustration the threshold for determining large-sized sparse

rule bases is empirically set to 95, in order to help evaluate the performance of KCR and CRC in relation to the sizes of the sparse rule bases concerned. Note that the Polynomial dataset in the table is produced by randomly sampling from the following 3-dimensional nonlinear function:

$$F(x, y) = \sin\left(\frac{x}{\pi}\right) \cdot \sin\left(\frac{y}{\pi}\right) \quad (19)$$

This nonlinear function has been used to produce a benchmark dataset in [14] and [37], and the random sampling method has been frequently employed in the literature (e.g., [38] and [39]).

TABLE I
DETAILS OF BENCHMARK DATASETS

Dataset	Feature no.	Instance no.	Output Domain	Rule no.
Dee	6	365	[0.765853, 5.11875]	20
AutoMPG6	5	392	[9.0, 46.6]	40
Stock	9	950	[34, 62]	90
Laser	4	993	[0.0, 255.0]	100
Friedman	5	1200	[0.6640, 28.5903]	120
Polynomial	2	2000	[-1, 1]	200
Quake	3	2178	[5.8, 6.9]	200
Delta_ail	5	7129	[-0.0021, 0.0022]	500
Delta_elv	6	9517	[-0.014, 0.013]	1000
Pole	26	14998	[0.0, 100.0]	1500

2) *Performance Evaluation Criteria:* To enable thorough evaluation and fair comparison, experimental results are reported using the average obtained from 10×10-fold cross-validation per dataset. Training sets are used to create sparse fuzzy rule bases (see next) and testing sets to assess the performance, in terms of RMSE (root-mean-square error, in relation to the ground truth). The smaller the value of RMSE is, the higher accuracy the approach has.

3) *Sparse Rule Base Generation:* In this experimental study, sparse fuzzy rule bases are artificially created from the dense fuzzy rule bases that are induced from the original datasets. This enables a challenging comparison of the FRI results (against those attainable using a full set of rules, although in real applications the full coverage of the problem domain is not assumed). In particular, a sparse rule base is generated by randomly removing a number of rules from the original dense rule base that has been learned by employing a data-driven learning method. To emphasise on the sparsity of the knowledge available, in order to compare against conventional approximate reasoning and state-of-the-art FRI mechanisms (both running on TSK models), only 80% rules are retained to form the sparse fuzzy rule base for each problem case.

The following simple data-driven fuzzy rule learning procedure is employed to generate the original dense rule base: The data instances in a given training dataset are clustered into different categories using fuzzy c-means [23]. Since fuzzy c-means allows an instance to belong to more than one cluster with different membership values, the worst rule-learning assumption is made here, with the least biased threshold of 0.5 membership value used to determine whether an attribute is taking on a certain fuzzy set as its value. The polynomial consequent of an emerging rule is learned through the popular linear regression approach as described in [40].

For computational simplicity apart from fairness in comparison, as indicated previously, only triangular membership functions are used throughout to represent fuzzy values. The three parameters of a triangular membership function are implemented by the infimum, centre and supremum of the corresponding cluster. Note that, if fine-tuned membership functions are available and employed, improved performance attainable by all interpolation approaches examined can be expected. The number of rules in the created sparse fuzzy rule base for each dataset is listed in Table I.

For easy illustration, consider the sparse rule base generated from the polynomial dataset. The following are two (randomly taken) examples of the learned rules:

- **If** x_1 is $A_1 = (0.092, 0.159, 0.227)$ and x_2 is $A_2 = (0.727, 0.767, 0.808)$,
then $f_i(x_1, \dots, x_n) = -1.759x_1 + 0.468x_2 + 0.029$
- **If** x_1 is $A_1 = (0.437, 0.455, 0.472)$ and x_2 is $A_2 = (0.388, 0.413, 0.438)$,
then $f_i(x_1, \dots, x_n) = -1.214x_1 - 0.764x_2 + 1.441$

Results from running the TSK model consisting of 200 such rules (taken from the sparse rule base) without FRI are illustrated in Fig. 5. The two sub-figures reflect the results viewed from a different inspection angle of the same inference process. In particular, Fig. 5(a) gives a sideview of the outcomes of running on the entire sparse rule base, and Fig. 5(b) shows a bird’s-eye view. As there are substantial amounts of space that are not covered by the learned rules, plenty of observations have matched no rule, resulting in missing values in the output domain. These two sub-figures collectively demonstrate the poor outcome of just exploiting the incomplete knowledge in the given problem domain, without the support of FRI.

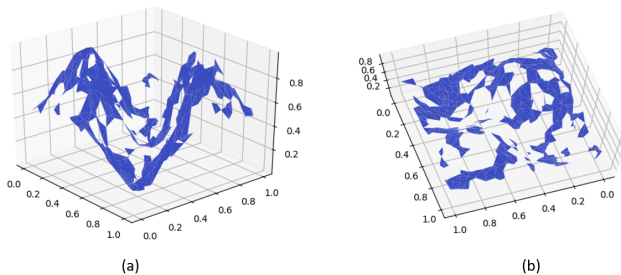


Fig. 5. Inference results running TSK model on sparse rule base without FRI on Polynomial dataset: (a) sideview and (b) bird’s-eye view.

4) *Algorithmic Parameters*: For completeness, the parameters used to implement KCR and CRC for the experiments on different datasets are listed in Table II. Note that the number of selected rules, K for both KCR and CRC is determined by a trial and error process. Particularly, the processes of determining the K for KCR and CRC are exemplified on the polynomial dataset, as shown in Fig. 6 and Fig. 7 respectively. For CRC, the Elbow method is employed to determine the number of clusters required.

5) *Automated Rule Selection (AutoRS)*: A key issue concerned within this work is the determination of the number of

TABLE II
SETTING FOR KCR AND CRC ON DIFFERENT DATASETS

Dataset	KCR		CRC	
	Selected rules (K)	Rule clusters	Selected rules (K)	
Dee	3	5	2	
AutoMPG6	3	5	2	
Stock	3	5	2	
Laser	3	10	3	
Friedman	4	10	3	
Polynomial	3	20	3	
Quake	3	20	4	
Delta_ail	5	50	5	
Delta_elv	9	80	5	
Pole	4	100	5	

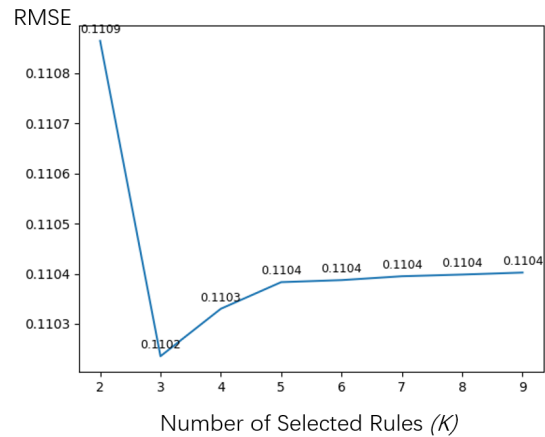


Fig. 6. Model RMSE vs. number of rules K on KCR.

neighbourhood rules, K used to perform FRI. In the implementation, K may be a fixed number empirically justified by a trial and error process as indicated above. However, automated rule selection (AutoRS) [15] provides a novel approach that makes automatic selection of fuzzy rules useful for subsequent rule interpolation. Therefore, the proposed approach is also compared with AutoRS which represents the most recent development in the area of FRI. The details of AutoRS are out of the scope of this paper, but its main procedure can be summarised as shown in Alg. A4, presented in Appendix A.

B. Results and Discussion

1) On Use of Different Similarity Measurement Methods:

Table III presents the means and standard deviations of the interpolation results produced by KCR and CRC (with FCM employed as the clustering method). Each FRI mechanism is supported by the use of either of the two similarity measurement methods introduced previously. As can be seen, the FRI algorithms employing Similarity-DF can produce better results on all but one dataset than those using Similarity-d. In particular, the effectiveness of utilising Similarity-DF is more obvious on more complex datasets. However, note that KCR with Similarity-d leads to the best inference outcome on the dataset Dee. One possible reason for this is that selected rules have similar weights concerning the conclusion whilst their distances to the observation are different. Having taken notice of the generally superior results of using Similarity-DF, in the

TABLE III
PERFORMANCE OF EMPLOYING DIFFERENT SIMILARITY MEASUREMENT METHODS IN KCR AND CRC (WITH FCM AS REPRESENTATIVE)

Dataset	KCR		CRC with FCM	
	Similarity-d	Similarity-DF	Similarity-d	Similarity-DF
Dee	0.367 ± 0.130	0.539 ± 0.204	0.498 ± 0.207	0.623 ± 0.304
AutoMPG6	4.010 ± 2.716	3.419 ± 1.384	3.627 ± 3.161	3.667 ± 1.503
Stock	2.703 ± 2.458	1.099 ± 0.362	8.095 ± 6.604	1.144 ± 0.402
Laser	14.98 ± 10.20	12.45 ± 6.51	22.76 ± 20.30	11.29 ± 5.95
Friedman	2.909 ± 1.712	3.400 ± 2.110	1.654 ± 1.033	1.615 ± 1.204
Polynomial	0.083 ± 0.011	0.088 ± 0.034	0.245 ± 0.091	0.037 ± 0.011
Quake	1.414 ± 0.764	1.152 ± 0.897	1.638 ± 3.129	0.575 ± 0.479
Delta_ail	1.516 ± 0.582	0.860 ± 0.630	1.477 ± 0.357	0.358 ± 0.243
Delta_elv	2.880 ± 2.012	2.175 ± 1.872	2.420 ± 1.247	0.327 ± 0.280
Pole	101.67 ± 42.99	90.26 ± 23.48	50.34 ± 21.75	13.70 ± 12.75

TABLE IV
PERFORMANCE OF DIFFERENT RULE SELECTION METHODS ON SMALL-SIZED SPARSE RULE BASES

Dataset	Mean ± Standard deviation							
	TSK+	AutoRS	KCR	CRC with Kmeans	CRC with GMM	CRC with FCM	CRC with K-FCM	CRC with S-FCM
Dee	0.572±0.222	0.554±0.201	0.539±0.204	0.620±0.302	0.623±0.302	0.623±0.304	0.597±0.207	0.597±0.258
AutoMPG6	3.467±1.419	3.267±1.313	3.419±1.384	3.681±1.551	3.645±1.470	3.667±1.503	3.579±1.477	3.642±1.514
Stock	1.141±0.390	1.149± 0.385	1.099±0.362	1.131±0.402	1.116±0.411	1.144±0.402	1.137±0.435	1.129±0.409

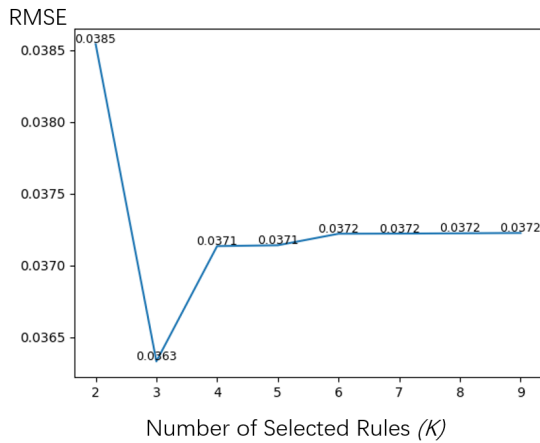


Fig. 7. Model RMSE vs. number of rules K on CRC.

following presentation of the experimental investigations, only Similarity-DF is adopted to measure similarity.

2) *On Rule Bases of a Small Size:* Table IV shows the means and standard deviations of FRI results, which are averaged outcomes over 10×10 -fold cross-validation, for each of the eight compared approaches on the datasets that involve the number of rules being less than or equal to 90. In this table, the notion of “CRC with C” stands for a procedure whose employed clustering method is “C” to create rule clusters, with “C” indicating one of the following: Kmeans, GMM, FCM, K-FCM or S-FCM. The comparison with the conventional TSK models is not included herein, due to the fact that TSK models alone cannot derive any conclusion when an observation does not match any of the rules in the rule base. Naturally, if compared, all rule interpolation algorithms would significantly outperform the direct utilisation of the TSK models involving a sparse rule base across all problems.

As can be seen from Table IV, regarding those models

comprising a sparse rule base of a small size, on average, the overall best results are obtained by KCR. AutoRS also works well; and TSK+ has slightly lower accuracies (despite having a higher computational complexity), which is due to the fact that all rules are involved, bringing forward adverse biases.

Independent of which of the five clustering methods is used, the CRC algorithm does not work well on small sized sparse fuzzy rule bases. This can be expected as those potentially highly relevant rules are most likely to have been clustered into one single cluster whilst the other clusters of rules offer little useful information to the conclusion. Thus, all rules bar one contribute misleading information to the calculation of the final results, leading to inaccurate interpolated outcomes.

3) *On Rule Bases of a Large Size:* Table V presents the means and standard deviations of interpolation results returned by each of the eight compared methods, regarding the five TSK models that involve sparse rule bases of a large size. All algorithms are ranked according to their means and standard deviations of the results on each dataset, while those of the same results are deemed to have the same rank. The lower the ranking value, the higher the model accuracy. In summary, the bottom row of this table provides the total rank values calculated by the sum of individual ranks across all seven benchmark datasets.

Clearly, CRC outperforms the rest consistently for such more complex datasets. Examining the experimental results more closely, CRC supported by K-FCM has highly impressive results on the Quake, Delta_ail and Delta_elv datasets, significantly improving the accuracy of the existing methods. Interestingly, this indicates that the kernel function (Eqn. 9) applied in K-FCM has the ability to make the clustering results more suitable for the distribution of the corresponding sparse rule bases.

As reflected by these experimental outcomes, the utilisation of any of the five clustering algorithms enables CRC to

outperform the other algorithms that do not involve clustering on the sparse rule bases, when these rule bases are of a significant size. This demonstrates the significance of clustering-aided FRI. Indeed, these results positively reflect the intuition that similar-rule crowds do have negative impact upon the accuracy of the final interpolated conclusions. With the support of clustering, CRC successfully avoids involving otherwise far too many similar rules and extends the diversity of rules used for subsequent interpolation. These results also confirm that whilst all (the sparse) rules are exploited to derive the final outcome, TSK+ does not resolve the problem well. Importantly, the narrow-banded standard deviation values as shown in Table V demonstrate the robustness of the proposed approach.

4) *Further Examination on Performance:* In general, CRC performs excellently for large-sized sparse rule bases, as demonstrated above. However, to systematically exploit the proposed approach as per Fig. 4, a threshold is required to determine the threshold at which to decide on whether KCR or CRC is to be used (although in the event when it is unrealistic to identify such a threshold, both methods may be applied to provide suggestions that are still useful for interpolative decision-making). According to Table I, on the stock and laser datasets, the sparse rule bases include 90 or 100 rules. Consider Tables IV and V, by comparing against the outcomes achieved on the other datasets, algorithms with and without clustering have more similar results on these two datasets. Therefore, borrowing the underlying idea of the Elbow method, the threshold of large-sized sparse rule bases can be determined as at least including 95 rules (the average number between 90 and 100). This is the empirical basis upon which the preceding experiments have been conducted.

Apart from the issue of determining the threshold, there are occasional situations where the generality of CRC outperforming KCR for large datasets does not necessarily hold. This is evident by considering the cases where CRC supported by K-FCM or S-FCM is utilised to carry out interpolative reasoning with the sparse rule bases which are induced from the polynomial dataset. Fig. 8 depicts the distribution of the results in a box-plot, across all methods investigated. As revealed by this figure, CRC with K-FCM or S-FCM is significantly underperformed than its peers, in comparison to CRC with K-means, GMM or the original version of FCM. To analyse the causes of such occasional performance deviation, the rules selected for a given observation on the polynomial dataset are

outlined in Table VI, represented by their indices. It can be seen that rule 36 is the nearest to the particular observation and is always selected by all rule selection methods. Rules selected by K-FCM and S-FCM are clearly distinct from the ones that lead to satisfactory results. The likely reason for this is that these two clustering algorithms fail to derive appropriate rule clusters on this dataset, adversely affecting the subsequent rule selection.

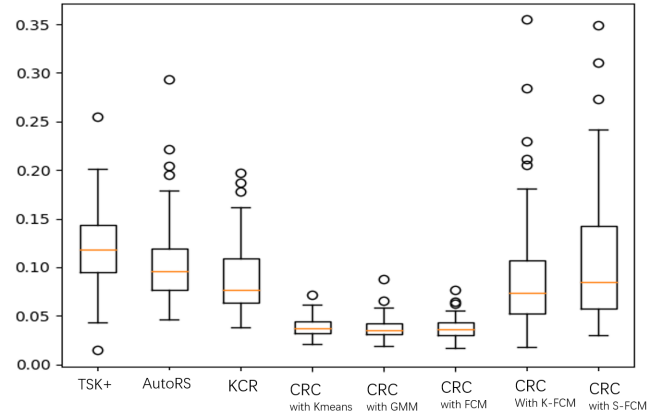


Fig. 8. Box-plot of interpolative results on polynomial dataset.

Another interesting observation is that what KCR selected are the top three rules also selected by AutoRS. However, with another two rules selected (and employed) on top of these three, AutoRS performs relatively worse than KCR. Note that rule 21 is selected by AutoRS, which is also taken by the top performers like CRC with GMM or FCM (both of which happen to utilise the same selected rules). However, AutoRS underperforms in comparison to the two CRC implementations. This is probably due to the fact that those three rules (of indices 36, 21 and 57) jointly offer the best information for producing accurate outcomes. Although rule 21 is also taken by AutoRS, it is being treated as the one of the lowest weight (as it is taken the last amongst the five rules selected in order). Thus, its potential contribution is delimited, whilst adding more computation costs to reach the (less well) interpolated result.

To qualitatively visualise the strengths of the proposed approach, Fig. 9 illustrates the best inference results produced by the method without clustering (KCR) and that with clustering

TABLE V
PERFORMANCE OF DIFFERENT RULE SELECTION METHODS ON LARGE-SIZED SPARSE RULE BASES

Dataset	Mean \pm Standard deviation							
	TSK+	AutoRS	KCR	CRC with Kmeans	CRC with GMM	CRC with FCM	CRC with K-FCM	CRC with S-FCM
Laser	13.858 \pm 6.210	12.970 \pm 6.686	12.456 \pm 6.510	11.218\pm6.050	11.513 \pm 6.046	11.292 \pm 5.949	12.033 \pm 6.262	11.790 \pm 6.840
Friedman	3.325 \pm 2.02	3.632 \pm 1.948	3.400 \pm 2.110	1.753 \pm 1.151	1.567 \pm 1.0002	1.615 \pm 1.204	1.528 \pm 1.168	1.522\pm1.112
Polynomial	0.119 \pm 0.040	0.103 \pm 0.040	0.088 \pm 0.034	0.038 \pm 0.009	0.037\pm0.011	0.037\pm0.011	0.088 \pm 0.055	0.105 \pm 0.062
Quake	1.155 \pm 0.750	1.147 \pm 0.895	1.152 \pm 0.897	0.663 \pm 0.474	0.683 \pm 0.523	0.575 \pm 0.479	0.408\pm0.319	0.497 \pm 0.396
Delta_ail (e-4)	1.788 \pm 1.449	0.951 \pm 0.604	0.860 \pm 0.630	0.357 \pm 0.230	0.359 \pm 0.241	0.358 \pm 0.243	0.243\pm0.161	0.359 \pm 0.246
Delta_elv (e-4)	4.778 \pm 1.582	7.216 \pm 3.256	2.175 \pm 1.872	0.525 \pm 0.472	0.602 \pm 0.208	0.327 \pm 0.280	0.299\pm0.230	0.360 \pm 0.422
Pole	100.97 \pm 73.54	75.50 \pm 26.15	90.26 \pm 23.48	39.24 \pm 13.06	45.59 \pm 15.90	13.70\pm12.75	15.50 \pm 14.75	19.31 \pm 18.98
Rank	52	47	48	24	27	16	16	24

TABLE VI
INDICES OF SELECTED RULES FOR ONE OBSERVATION ON POLYNOMIAL

Approach	Indices of selected rules
TSK+	All rules in sparse rule base
AutoRS	[36, 39, 41, 45, 21]
KCR	[36, 39, 41]
CRC with Kmeans	[36, 22, 59]
CRC with GMM	[36, 21, 57]
CRC with FCM	[36, 21, 57]
CRC with K-FCM	[36, 41, 21]
CRC with S-FCM	[36, 45, 56]

(CRC with FCM), over the polynomial dataset. Together with Fig. 5, it can be seen that the novel approach introduced herein enables appropriate conclusions to be generated for data instances unmatched by the sparse rules. Additionally, for this particular dataset, CRC with FCM can produce smoother results than KCR.

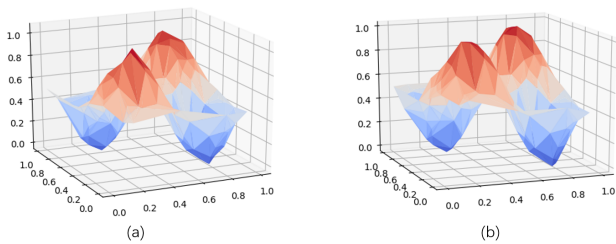


Fig. 9. Inference results using KCR (a) and using CRC with FCM (b), on Polynomial dataset.

VI. CONCLUSION AND FUTURE WORK

This paper has presented a novel approach for performing fuzzy rule interpolation (FRI) with TSK models. The work has been motivated by the observation that existing FRI approaches are almost completely devised for reasoning with Mamdani models, whilst the very few developed for TSK models are inefficient. It makes two particular contributions to the FRI literature: (i) For models involving a sparse rule base of a small size, the implementation of the approach (KCR) derives accurate interpolation using only a small number of closest rules to an unmatched input. (ii) For models comprising a sparse rule base of a large size, the implementation (CRC) first employs a clustering method to categorise the rules into groups and then, performs interpolation by just utilising one closest rule from each of a small number of the resulting rule clusters. The paper has presented results of systematic comparative experimental studies over a range of benchmark datasets, demonstrating the efficacy of both implementations.

This original work has offered many opportunities for further development. For example, whilst five clustering methods have been considered to support the realisation of CRC, they may not be the best to generate the most appropriate categories for all sparse rule bases. More clustering algorithms, particularly modified fuzzy c-means algorithms (e.g., the global fuzzy c-means [41] and the possibilistic fuzzy c-means [42]) may be adopted as the alternative to strengthening

the performance. Another point is regarding the specification of the required algorithmic parameters, such as the number of closest rules and that of the nearest clusters. Currently, they are empirically set; creating an automated method to determine these parameters from the training data requires significant further research. Also, AutoRS offers a means for automated selection of the number of closest rules for interpolation, how it may be integrated with the proposed approach to minimise human intervention is worth investigating.

Another important issue is the determination of whether a given rule base is a large or a small one, in order to facilitate an informed choice of which FRI method to use. While the size of a rule base is typically related to that of the datasets concerned, it may also be affected by a number of other factors (e.g., the number of domain features and the distribution of data objects). An automated mechanism to decide on a rule base's size is clearly desirable. In the present implementations, all rule antecedent variables are treated equally. This gives rise to a further interesting piece of active research, aiming at extending weighted representations as per the most recent work of [7, 13] to accommodating interpolation with TSK models. Furthermore, for many real-world problems, the inputs are usually time-dependent, the requirements of fuzzy systems may change over time. Therefore, designing a novel system that can dynamically maintain and enrich the sparse fuzzy rule base is also desirable.

APPENDIX A

The procedures of fuzzy rule inference methods and their associated algorithms are outlined here for easy reference.

Algorithm A1: Inference with TSK models

Input: Rule base $\{R_i\}$; Observation O

Output: Conclusion $f(O)$

- 1) Calculation of matching degrees between individual antecedent variables of each rule R_i and their counterparts in observation O :
 $D(A_{i1}, B_1), \dots, D(A_{in}, B_n)$
- 2) Determination of weight of R_i via aggregating all matching degrees:
 $\alpha_i = D(A_{i1}, B_1) \wedge \dots \wedge D(A_{in}, B_n)$
 where \wedge is implemented by minimum (or alternative if preferred).
- 3) Computation of rule consequent polynomial with O as input for each of k matched rules, resulting in sub-conclusions:
 $f_i(B_1, \dots, B_n) = a_{i0} + a_{i1}B_1 + \dots + a_{in}B_n$
- 4) Integration of all sub-conclusions, obtaining final outcome for consequent by computing weighted average:

$$f(B_1, \dots, B_n) = \frac{\sum_{i=1}^k \alpha_i f_i(B_1, \dots, B_n)}{\sum_{i=1}^k \alpha_i} \quad (20)$$

- 5) Return: $f(B_1, \dots, B_n)$.
-

Algorithm A2: TSK extension**Input:** Rule base $\{R_i\}$; Observation O **Output:** Interpolated conclusion $f(O)$

- 1) Calculation of similarities between observation O and each rule R_i : $S(A_{i1}, B_1), \dots, S(A_{in}, B_n)$
- 2) Determination of weight of rule R_i :
 $\alpha_i = S(A_{i1}, B_1) \wedge \dots \wedge S(A_{in}, B_n)$
- 3) Integration of all similarity measures, obtaining interpolated rule with following consequent parameters:

$$a_0 = \frac{\sum_{i=1}^m \alpha_i a_{i0}}{\sum_{i=1}^m \alpha_i}, \dots, a_n = \frac{\sum_{i=1}^m \alpha_i a_{in}}{\sum_{i=1}^m \alpha_i} \quad (21)$$

- 4) Computation of interpolated outcome with observation O as input to interpolated rule:
 $f(B_1, \dots, B_n) = a_0 + a_1 B_1 + \dots + a_n B_n$
- 5) Return: $f(B_1, \dots, B_n)$.

Algorithm A3: Fuzzy C-means**Input:** Instances x_j ; Number of cluster K **Output:** Instances with label; Cluster centres V ;
Matrix of membership degree U

- 1) Initialisation of $U^{(0)}$ randomly.
- 2) Specification of maximum iteration number L , termination condition ε , and counter k , $k = 0, 1, \dots, L$
- 3) Computation of $V^{(k)}$ with $U^{(k)}$ and Eqn. 7.
- 4) Updating of $U^{(k+1)}$ by $V^{(k)}$ and Eqn. 8.
- 5) Evaluation of $\|U^{(k+1)} - U^{(k)}\|$, if it is less than ε or $k = L$, stop; otherwise, set $U^{(k)} = U^{(k+1)}$ and return to Step 3.
- 6) Return: Labeled instances; $V(k+1)$; $U(k+1)$.

Algorithm A4: AutoRS for interpolation**Input:** Rule base $\{R_i\}$; Observation O **Output:** Selected rule set for interpolation $U(R)$

- 1) Initialisation of candidate rule set, $U'(R_k)$: for each antecedent attribute, iteratively adding rules from nearest to furthest until emerging rule set satisfies either of following conditions:
 - a) $\exists R_i, R_j \in U'(R_k), Rep(A_{ik}) > Rep(B_k)$ and $Rep(A_{jk}) < Rep(B_k) (i < j)$
 - b) $\exists R_i, R_j \in U'(R_k), Rep(A_{ik}) < Rep(B_k)$ and $Rep(A_{jk}) > Rep(B_k) (i < j)$
 - c) $\exists R_i \in U(R), Rep(A_{ik}) = Rep(B_k)$

where $Rep(A_{ik})$ stands for representative value of k -th antecedent fuzzy set of i -th fuzzy rule, and $Rep(B_k)$ for that of k -th feature of unmatched observation O .

- 2) Assignment of initial $U(R)$ to largest candidate rule.
- 3) Verification if $U(R)$ satisfies either of conditions indicated in Step 1 for all other features, if so, stop; otherwise, update $U(R) = U(R) \cup U'(R_k)$.
- 4) Post pruning of $U(R)$: for each $R_i \in U(R)$, if $U(R) - \{R_i\}$ still satisfies any one condition, update $U(R) = U(R) - R_i$; otherwise, retain R_i .
- 5) Iteration of Step 4 with $i = i + 1$.
- 6) Return: $U(R)$.

REFERENCES

- [1] S.-H. Liao, "Expert system methodologies and applications—a decade review from 1995 to 2004," *Expert Systems with Applications*, vol. 28, no. 1, pp. 93–103, 2005.
- [2] C. Grosan and A. Abraham, "Rule-based expert systems," in *Intelligent Systems*, pp. 149–185, Springer, 2011.
- [3] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 1, pp. 116–132, 1985.
- [4] F. Li, C. Shang, Y. Li, J. Yang, and Q. Shen, "Approximate reasoning with fuzzy rule interpolation: background and recent advances," *Artificial Intelligence Review*, vol. 54, pp. 4543–4590, 2021.
- [5] J. Yang, C. Shang, Y. Li, F. Li, and Q. Shen, "Anfis construction with sparse data via group rule interpolation," *IEEE Transactions on Cybernetics*, vol. 51, no. 5, pp. 2773–2786, 2021.
- [6] S. Nagy, B. Sziová, and L. T. Kóczy, "The effect of image feature qualifiers on fuzzy colorectal polyp detection schemes using kh interpolation—towards hierarchical fuzzy classification of coloscopic still images," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–7, IEEE, 2018.
- [7] F. Li, C. Shang, Y. Li, and Q. Shen, "Interpretable mammographic mass classification with fuzzy interpolative reasoning," *Knowledge-Based Systems*, vol. 191, p. 105279, 2020.
- [8] N. Naik, R. Diao, and Q. Shen, "Dynamic fuzzy rule interpolation and its application to intrusion detection," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 4, pp. 1878–1892, 2018.
- [9] Z. Huang and Q. Shen, "Fuzzy interpolation and extrapolation: A practical approach," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 1, pp. 13–28, 2008.
- [10] L. Kóczy and K. Hirota, "Approximate reasoning by linear rule interpolation and general approximation," *International Journal of Approximate Reasoning*, vol. 9, no. 3, pp. 197–225, 1993.
- [11] L. Yang and Q. Shen, "Adaptive fuzzy interpolation," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 6, pp. 1107–1126, 2011.
- [12] C. Chen, N. MacParthlain, Y. Li, C. Price, C. Quek, and Q. Shen, "Rough-fuzzy rule interpolation," *Information Sciences*, vol. 351, pp. 1–17, 2016.
- [13] F. Li, C. Shang, Y. Li, J. Yang, and Q. Shen, "Fuzzy rule-based interpolative reasoning supported by attribute ranking," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 5, pp. 2758–2773, 2018.
- [14] J. Li, L. Yang, Y. Qu, and G. Sexton, "An extended takagi–sugeno–kang inference system (tsk+) with fuzzy interpolation and its rule base generation," *Soft Computing*, vol. 22, no. 10, pp. 3155–3170, 2018.
- [15] T. Chen, C. Shang, J. Yang, F. Li, and Q. Shen, "A new approach for transformation-based fuzzy rule interpolation," *IEEE Transactions on Fuzzy Systems*, vol. 28,

- no. 12, pp. 3330–3344, 2019.
- [16] S.-J. Chen and S.-M. Chen, “Fuzzy risk analysis based on similarity measures of generalized fuzzy numbers,” *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 1, pp. 45–56, 2003.
- [17] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, Oakland, CA, USA, 1967.
- [18] R. Thorndike, “Who belongs in the family?,” *Psychometrika*, vol. 18, no. 4, pp. 267–276, 1953.
- [19] T. M. Kodinariya and P. R. Makwana, “Review on determining number of cluster in k-means clustering,” *International Journal*, vol. 1, no. 6, pp. 90–95, 2013.
- [20] C. Bouveyron and C. Brunet-Saumard, “Model-based clustering of high-dimensional data: A review,” *Computational Statistics and Data Analysis*, vol. 71, pp. 52–78, 2014.
- [21] J. A. Bilmes *et al.*, “A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models,” *International Computer Science Institute*, vol. 4, no. 510, p. 126, 1998.
- [22] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.
- [23] J. C. Bezdek, R. Ehrlich, and W. Full, “Fcm: The fuzzy c-means clustering algorithm,” *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.
- [24] P. Zhang and Q. Shen, “Fuzzy c-means based coincidental link filtering in support of inferring social networks from spatiotemporal data streams,” *Soft Computing*, vol. 22, no. 21, pp. 7015–7025, 2018.
- [25] D.-Q. Zhang and S.-C. Chen, “A novel kernelized fuzzy c-means algorithm with application in medical image segmentation,” *Artificial Intelligence in Medicine*, vol. 32, no. 1, pp. 37–50, 2004.
- [26] K.-R. Müller, S. Mika, K. Tsuda, and K. Schölkopf, “An introduction to kernel-based learning algorithms,” *Handbook of Neural Network Signal Processing*, pp. 4–1, 2018.
- [27] J.-L. Fan, W.-Z. Zhen, and W.-X. Xie, “Suppressed fuzzy c-means clustering algorithm,” *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1607–1612, 2003.
- [28] R. L. Cannon, J. V. Dave, and J. C. Bezdek, “Efficient implementation of the fuzzy c-means clustering algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, pp. 248–255, 1986.
- [29] L.-M. Wei and W.-X. Xie, “Rival checked fuzzy c-means algorithm,” *Acta Electronica Sinica*, vol. 28, no. 7, pp. 63–66, 2000.
- [30] C. A. Hoare, “Algorithm 65: find,” *Communications of the ACM*, vol. 4, no. 7, pp. 321–322, 1961.
- [31] P.-C. Chang and C.-Y. Fan, “A hybrid system integrating a wavelet and tsf fuzzy rules for stock price forecasting,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 6, pp. 802–815, 2008.
- [32] Q. Shen and R. Leitch, “Fuzzy qualitative simulation,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 4, pp. 1038–1061, 1993.
- [33] S.-M. Chen, “New methods for subjective mental workload assessment and fuzzy risk analysis,” *Cybernetics & Systems*, vol. 27, no. 5, pp. 449–472, 1996.
- [34] D. Dua and C. Graff, “UCI machine learning repository,” 2017.
- [35] H. A. Guvenir, I. Uysal, and F. A. Repositor, “Function approximation repository,” *Bilkent University*. URL <http://funapp.cs.bilkent.edu.tr/DataSets>, 2000.
- [36] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, “Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework.,” *Journal of Multiple-Valued Logic & Soft Computing*, vol. 17, 2011.
- [37] B. Rezaee and M. F. Zarandi, “Data-driven fuzzy modeling for takagi–sugeno–kang fuzzy system,” *Information Sciences*, vol. 180, no. 2, pp. 241–255, 2010.
- [38] J. Li, Y. Qu, H. P. Shum, and L. Yang, “TSK inference with sparse rule bases,” in *Advances in Computational Intelligence Systems*, pp. 107–123, Springer, 2017.
- [39] H. Bellaaj, R. Ketata, and M. Chtourou, “A new method for fuzzy rule base reduction,” *Journal of Intelligent & Fuzzy Systems*, vol. 25, no. 3, pp. 605–613, 2013.
- [40] D. A. Freedman, *Statistical models: theory and practice*. Cambridge University Press, 2009.
- [41] W. Wang, Y. Zhang, Y. Li, and X. Zhang, “The global fuzzy c-means clustering algorithm,” in *2006 6th World Congress on Intelligent Control and Automation*, vol. 1, pp. 3604–3607, IEEE, 2006.
- [42] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, “A possibilistic fuzzy c-means clustering algorithm,” *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 4, pp. 517–530, 2005.



Pu Zhang received the B.Sc. degree in computer science and technology from Northwestern Polytechnical University, China, in 2015, and the M.Sc. degree in intelligent systems from Aberystwyth University, U.K., in 2016. He is currently working toward the Ph.D. degree in computational intelligence with the Department of Computer Science, Aberystwyth University, U.K. His research interests include fuzzy interpolative reasoning, machine learning, and nonlinear regression.



Changjing Shang received the Ph.D. degree in computing and electrical engineering from Heriot-Watt University, U.K., in 1995. She is currently a University Research Fellow with the Department of Computer Science, Aberystwyth University, U.K. Prior to joining Aberystwyth, she was with Heriot-Watt, Loughborough and Glasgow Universities. She has published 160 peer-reviewed papers, and supervised more than twenty Ph.Ds/PDRAs in the areas of pattern recognition, data mining, intelligent robotics, and image analysis



Qiang Shen received the Ph.D. degree in computing and electrical engineering from Heriot-Watt University, U.K. in 1990, and the D.Sc. degree in computational intelligence from Aberystwyth University, U.K. in 2013. He holds the Established Chair in Computer Science and is Pro Vice-Chancellor for Faculty of Business and Physical Sciences, Aberystwyth University. He has authored two research monographs and more than 410 peer-reviewed papers, including one receiving an Outstanding Transactions Paper Award from IEEE.