Predefining Numbers of Fuzzy Sets for Genetically Generated Fuzzy Knowledge Bases Using Clustering Techniques: Application to Tool Wear Monitoring

S. Achiche
Department of Mechanical Engineering
École Polytechnique de Montréal
C.P. 6079, Succ. Centre-Ville, Montréal, H3C 3A7, Canada
sofiane.achiche@polymtl.ca
M. Balazinski
Department of Mechanical Engineering
École Polytechnique de Montréal
C.P. 6079, Succ. Centre-Ville, Montréal, H3C 3A7, Canada
marek.balazinski@polymtl.ca

A. Przybylo
Department of Mechanical Engineering
École Polytechnique de Montréal
C.P. 6079, Succ. Centre-Ville, Montréal, H3C 3A7, Canada
aleksander.przybylo@polymtl.ca
L. Baron
Department of Mechanical Engineering
École Polytechnique de Montréal
C.P. 6079, Succ. Centre-Ville, Montréal, H3C 3A7, Canada
luc.baron@polymtl.ca

Abstract - One of the problems surrounding fuzzy knowledge base generation using genetic algorithms is finding an optimal number of fuzzy sets for each premise. A Genetic algorithm developed by the authors for the automatic generation of fuzzy knowledge bases uses a multi-objective method combining error minimization and simplification. This paper proposes solutions based on cluster analysis and validation indices for the numbers of clusters used in predefining the numbers of fuzzy sets. Two different validation indices as well as a combination of one of these with the multi-objective method are compared to the original multi-objective method on both synthetic and experimental data. Results obtained with the proposed techniques showed a considerable improvement over the multi-objective method on both data sets.

I. INTRODUCTION

A Fuzzy knowledge base (FKB) is composed of a factual base and a rule base. The factual base is composed of premises (inputs) and conclusions (outputs) along with fuzzy sets distributed on both. The rule base is composed of a finite number of fuzzy rules. One of the numerous problems surrounding FKB learning using genetic algorithms is finding an optimal number of fuzzy sets for the factual base. Furthermore, one has to know that the quality of an FKB is very sensitive to the quality of the factual base [8]. The simplest genetic algorithms start with a random distribution of fuzzy sets and optimize their location and/or shape, but cannot determine the number of fuzzy sets [12]. The authors have developed a real binary-like coded genetic algorithm (RBCGA) [1]; a more “evolved” method that uses a multi-objective (MO) approach combining error minimization and simplification. Simplification of the FKBs is performed by starting with an over estimated number of fuzzy sets and then randomly eliminating fuzzy sets one by one along with the corresponding fuzzy rules. The main drawback of the MO method is that solutions often remain too complex (large sizes of the factual base and the rule base). Moreover, since the evolutionary algorithm starts with very complex solutions, the computation time necessary to evaluate each solution is too long for some applications. In this paper we describe a technique for finding an optimal number of fuzzy sets for each premise based on the application of the k-means clustering algorithm to optimize learning time and FKB quality. A validation of the algorithms is performed using synthetic data obtained from a 3D surface followed by a factory-floor application; tool wear monitoring. The fuzzy decision support system (FDSS) Fuzzy-Flou developed by the authors at École Polytechnique de Montréal and University of Silesia (Poland) [3] is used as the prediction tool.

II. AUTOMATIC GENERATION OF FKBs

GAs are powerful stochastic optimization techniques based on the analogy of the mechanics of biological genetics and imitate the Darwinian survival of the fittest approach [11]. As shown in Fig. 2, each individual of a population is a potential member of the FDSS Fuzzy-Flou knowledge base. Four basic operations of the developed GA learning software are performed; reproduction, mutation, evaluation and natural selection. The RBCGA developed by the authors is a combination of a real coded genetic algorithm and a binary coded genetic algorithm.

A. Coding

The genotype of an FKB is generated by coding its parameters into chromosomes and corresponds to several independent sets of real numbers for the factual base (non-symmetrical-overlapping triangular fuzzy sets for premises
and symmetrical triangular fuzzy sets for the conclusions) and a set of integers for the rule base.

B. Multi-Crossover Mechanisms

The evolution of a population of FKBs for each generation is achieved by the reproduction of the “best” individuals, based on their abilities to survive natural selection. Reproduction is performed by combinations of the genotypes of the parents to obtain the genotype of an offspring using a multi-crossover which is composed of a premises/conclusion crossover and a fuzzy rules crossover. These mechanisms are governed by the initiating probability \( p_c \).

C. Premises/Conclusion Crossover

The mechanism used is called a blending crossover \( \alpha \) (BLX-\( \alpha \)) [25], where \( \alpha \) determines the exploitation/exploration level of the offspring. The parameter \( \alpha \) is set to 1.0 for the first third of the generations (exploration), to 0.5 for the second third (relaxed exploitation) and finally to 0.1 for the last third of the evolution (exploitation) [2].

D. Fuzzy Rules Crossover

This mechanism is applied to the fuzzy rule base. The operation is performed by exchanging the end part of the sets (containing the fuzzy rules) of the parents at a randomly selected crossover site [2].

E. Fuzzy Set Reducer

The fuzzy set reducer mechanism is used to reduce the complexity of the FKBs by selecting a fuzzy set on each premise and erasing it from the respective sets together with the corresponding fuzzy rules. This mechanism is governed by the initiating probability \( p_r \). In this paper, \( p_r \) is set to 0.15 while testing the MO method and to 0.0 while using predefined numbers of fuzzy sets, since the optimal number of fuzzy sets is supposed to be reached in the initial population. The 0.15 value for \( p_r \) generated the most reliable results on the validation data used in section IV. When using the MO method, the initial number of fuzzy sets was set to 7.

F. Mutation

The probability \( p_m \) governs the occurrence of this mechanism. Uniform mutation is used [15]. The probability \( p_m \) is set to 0.1 when testing the MO technique and to 0.01 when predefined numbers of fuzzy sets are used. Again, the 0.1 value for \( p_m \) was obtained after several tests on the validation data. It has been lowered for predefined numbers of fuzzy sets to decrease the chance of having two fuzzy sets overlapping, causing one of them to be eliminated.

G. Natural Selection

Natural selection is performed on the population by keeping the “most promising” individuals based on their fitness. In this paper the population size is set to 100 and the number of generations is set to 200.

H. Performance Criterion of the RBCGA

The performance criterion allows the computation of the rating of each FKB used by the RBCGA in order to perform natural selection. In this paper, the performance criterion is the combination of the accuracy level of an FKB (approximation error) in reproducing the outputs of the learning data and the simplicity of the FKB. The approximation error \( \Delta_{RMS} \) is measured using the RMS error method:

\[
\Delta_{RMS} = \frac{1}{N} \sum_{i=1}^{N} (RBCGA_{output} - data_{output})^2
\]

where \( N \) represents the size of the learning data. The RMS fitness value \( \phi_1 \) is evaluated as a percentage of the output length of the conclusion \( L \), i.e.:

\[
\phi_1 = \frac{L - \Delta_{RMS} \times 100}{L}
\]

The second objective function \( \phi_2 \) evaluates the complexity of an FKB based on the number of rules (compared to the maximum possible). The resulting objective function (3) combines these two contradictory objectives in a weighted sum of \( \phi_1 \) and \( \phi_2 \) based on the weight \( \omega_0 \) associated with \( \phi_1 \).

\[
\phi = \omega_0 \phi_1 + (1 - \omega_0) \phi_2
\]

The weight \( \omega_0 \) is set to 0.9, in order to put the emphasis on accuracy rather than simplicity.

III. CLUSTERING

A. The k-means Algorithm

The \( k \)-means algorithm [23] was selected for the computation of clusters as it is arguably the most popular clustering algorithm. The version implemented by the authors in the software used in this paper starts with a distribution based on the maximization of distance between clusters. First, the sum of distances between each object and all the other objects is computed. The object with the largest sum of distances constitutes the first cluster centroid. Next cluster centroid is the furthest object from the first centroid. The subsequent cluster centroids are those with the largest values for the minimum distance to the previous centroids. This method ensures very fast and accurate results in the case of outlier-free data. In presence of outliers, another initial distribution or clustering method should be used. In this paper an assumption has been made that the data contains no outliers.
Convergence is obtained by selecting an object and moving it to the first cluster closer than the cluster it belongs to. After each swapping, the centroids are recalculated. Such a method usually gives better results than searching for the closest centroid. The loop stops when each object belongs to the closest cluster.

The number of clusters \( k \) must be specified \textit{a-priori} since the \( k \)-means algorithm can only swap objects between clusters. It cannot merge or divide existing clusters. This is why additional techniques must be used in order to find the optimal number of clusters \( (k_{opt}) \) for each data set. Fig. 4 shows a screenshot of the clustering software developed for optimization of FKB automatic generation using clustering techniques.

![Fig. 4 Screenshot of the clustering software](image)

**B. Number of Clusters**

Several techniques for finding the optimal number of clusters \( k_{opt} \) are available. Most of these techniques require the computation of clusters for every possible \( k \) and then propose a distribution based on several parameters such as within-cluster similarity, between-cluster dissimilarity, etc. Such is the case for the five most popular indices suggested by several research papers [7, 14, 18, 20, 27]. Sugar and James [26] provide quite extensive descriptions and experimental results for these techniques. Several methods have been proposed that do not require computation of clusters for every possible \( k \), but none of them is completely satisfactory [10]. Jain and Dubes [17] provide a general overview of such methods.

In this paper, two different methods have been selected. The first method, proposed by Hartigan [14], is based on the within cluster sum of square distances (tr\( W \)). This method advises the distribution with the smallest \( k \) such that \( Hart_{k} < 10 \) (4).

\[
Hart_{k} = \left( \frac{W_k}{W_{k+1}} \right) \times (n-k-1)
\]

where \( n \) is the size of the data.

The Silhouette method proposed by Kaufman and Rousseeuw [18] advises the distribution with the largest average Silhouette. The Silhouette of an object \( i \) is defined by (5).

\[
Silhouette(i) = \frac{b_i - a_i}{\max(a_i, b_i)}
\]

where \( a_i \) is the average dissimilarity between object \( i \) and all other objects in the cluster to which it belongs and \( b_i \) is the smallest average \( d_{ij} \), where \( d_{ij} \) is the distance of object \( i \) to all objects of cluster \( j \) to which it does not belong.

These two methods have been selected because they usually give relatively distinct results, whereas the other methods often provide results close to the Silhouette method.

The results obtained with these techniques, as well as those obtained with the original multi-objective method, are compared in section 5. Since, the \( Hart \) index tends to overestimate the number of clusters in a data set, a combination of this technique with the MO method has also been included in the tests.

**C. Global Algorithm Overview**

The learning process has been divided into two stages. First, the data is analyzed by the clustering software in order to find \( k_{opt} \) for each variable. The different dimensions are treated independently as several sets of one-dimensional data. The distribution obtained for each premise is then memorized and used as input to the genetic algorithm software.

The second stage starts by creating the desired number of knowledge bases with the appropriate number of fuzzy sets, defined by the clustering process for each premise. The first 80% of the knowledge bases are created randomly and the remaining 20% are created using the centroid positions found by the clustering algorithm. The 80/20 proportion has been obtained from performance and stability tests on several data sets. It is noteworthy that the first and last clusters are ignored, since the first and last premises are defined by the limits of the data. This is necessary to preserve a valid range of the premises in the FKBs. Due to this restriction, when any premise has only two fuzzy sets, the centroid positions for this particular premise are totally ignored. Another version of the algorithm uses all the centroid positions and adds two additional fuzzy sets for the limits, thus obtaining \( (k_{opt} + 2) \) fuzzy sets. Creating 2 additional fuzzy sets increases the complexity of the FKB and the results obtained with this version didn’t give satisfactory results. Hence, they were omitted in the validation results in the next section.

Note that no preliminary tests are required when predefining the number of fuzzy sets, contrary to the case when finding the optimal values for \( p_c, p_r \), and \( p_m \) for each new data set. This makes the automatic generation less fastidious and more reliable.

**IV. Validation Results**

Two series of validation tests have been performed. The first one is based on a synthetic 3D surface and the second one on experimental 5D data obtained from a tool condition monitoring application.
A. 3D Surface Data

The synthetic data was obtained from a 3D surface defined by (6) in the interval of \([-3; 3] \times [-3; 3]\). The output range for this input interval is \([-54.0; 54.0]\).

\[
f(x, y) = 3x^2y - y^3
\]

Data samples containing 100 entries have been randomly generated in 9 Gaussian clusters with 10 different standard deviations in the interval of \([0.05, 0.5]\). A slight Gaussian noise (standard deviation of 1.0) has also been added. A set of 30 samples have been generated for each of the 10 cluster spreads (standard deviations), which ensures reliable estimates of performance and stability of the employed techniques. The obtained FKBs were all tested on the same sample of 3721 uniformly distributed theoretical data without any noise. Fig. 5 shows the resulting validation errors obtained with each technique.

![Validation errors on synthetic data](image)

Fig. 5 Validation errors on synthetic data

Fig. 5 demonstrates that both Silhouette and the combination of the Hartigan and the MO technique (Hart + MO) give quite noticeable improvement on the validation error. The Hartigan technique alone gives slightly worse results. As previously mentioned, this can be explained by the fact that the Hartigan technique tends to overestimate the number of clusters in a data set. This increases the complexity of the obtained FKBs causing overfitting of the training data and hence diminishing its generalization capabilities. A summary of the results is shown in table 1.

<table>
<thead>
<tr>
<th>Cluster Spread</th>
<th>0.05</th>
<th>0.15</th>
<th>0.25</th>
<th>0.35</th>
<th>0.45</th>
</tr>
</thead>
<tbody>
<tr>
<td>Silhouette</td>
<td>11.85</td>
<td>10.98</td>
<td>9.65</td>
<td>8.12</td>
<td>7.23</td>
</tr>
<tr>
<td>Improvement over MO</td>
<td>16 %</td>
<td>14 %</td>
<td>12 %</td>
<td>15 %</td>
<td>21 %</td>
</tr>
<tr>
<td>Improvement over MO</td>
<td>-3 %</td>
<td>-4 %</td>
<td>-6 %</td>
<td>-7 %</td>
<td>-3 %</td>
</tr>
<tr>
<td>Hart + MO</td>
<td>13.13</td>
<td>12.09</td>
<td>9.91</td>
<td>8.91</td>
<td>8.70</td>
</tr>
<tr>
<td>Improvement over MO</td>
<td>7 %</td>
<td>5 %</td>
<td>10 %</td>
<td>7 %</td>
<td>4 %</td>
</tr>
</tbody>
</table>

Table 1 indicates a maximum improvement over the original MO method of 21% in the case of the method based on the Silhouette index and 10% in the case of the method based on the combination of the Hartigan index and the MO method. An approximate 5% increase on the validation error was observed for the method based on the Hartigan index alone.

B. Application to Tool Wear Monitoring

Experimental data for a tool wear monitoring application were collected by Bombinski and Jemielniak [6]. Four signals were collected (cutting force, feed force, passive force and acoustic emissions) out of which numerous signal features (SFs) were computed: average value, RMS, several distribution parameters etc. Based on their correlation with the tool wear, four SFs were selected for the learning process: variance of the feed force \((F_{\text{Var}})\), variance of the feed force from the 2nd seconds of cuts \((F_{\text{Var, 2nd}})\), RMS of the acoustic emissions from the 2nd seconds of cuts \((AE_{\text{RMS, 2nd}})\) and the average value of the feed force \((F_{\text{av}})\).

![Selected signal features](image)

Tool 1 – Tool 2 – Tool 3 – Tool 4

Tool wear has been approximated as the ratio of the cutting time performed versus the overall tool life \((\Delta T)\). In total 94 operations were performed during which 9 tools were
worn out. For more precise information on how the data were collected and on the signal features that were chosen, please refer to [6]. Fig. 6 shows the selected signal features versus the used up portion of the tool life from 4 different tools. Since the amount of data was quite small, a leave-one-out cross-validation technique was used. During each of the 9 training sessions data collected from one tool was hidden and later used for validation, whereas data collected from the 8 other tools were used for training. Fig. 7 shows the mean validation errors for each technique, where similar order of validation errors to that on Fig. 5 can be noticed. Again, results from the Silhouette and Hart + MO techniques were clearly superior to the multi-objective method except that this time the Hart technique generated better results than the Hart + MO. It is noteworthy that a 28% improvement has been obtained. The contents of Fig. 7 can also be analyzed by their relationship with the complexity of the FKBs, i.e. the mean number of rules (the product of the numbers of fuzzy sets in the particular premises) which are presented on Fig. 8.

It can be seen on Fig. 7 and Fig. 8 that the prediction accuracy of the FKBs is directly linked to their complexity. The weaker performance of the FKBs obtained with the MO method can thus be explained by the inability of the MO method to accurately reach the optimal complexity for that particular data set.

Fig. 9 shows representative examples of FKBs: one obtained using the MO method and one obtained with predefined numbers of fuzzy sets using the Silhouette technique. Slight differences can be seen between the two knowledge bases. In these particular examples (not all FKBs obtained with the same method have the same complexity) the FKBs obtained with the MO method and with the Silhouette technique have respectively 64 and 40 rules.

V. CONCLUSION

Two different methods of predefining the numbers of fuzzy sets for automatic generation of FKBs using GAs were developed and compared with the initial MO method. A combination of one of these techniques with the MO method was also analyzed. Validation with synthetic and experimental data show considerable improvement in prediction accuracy for FKBs obtained with predefined numbers of fuzzy sets. On the synthetic data set, an average improvement of 15.4% (and a maximum of 21%) was obtained with the Silhouette technique. On the experimental data set, an improvement of 28% was obtained with that same technique. Moreover, use of a predefined number of fuzzy sets removes the necessity to pre-analyze the training data. An optimal complexity can be determined automatically by the clustering process, thus removing the need for human supervision.
gratefully acknowledged. Financial support from the Natural Sciences and Engineering Research Council of Canada under grants of RGPIN-203618, RGPIN-105518 and STPGP-269579 is gratefully acknowledged.

Authors would also like to thank Sebastian Bombinski and Krzysztof Jemielniak for providing us with very valuable experimental data.

ACKNOWLEDGMENT

Fig. 10 Predicted tool wear versus experimental data on four different tools

REFERENCES