A Histogram-Based Rule Extraction Technique for Fuzzy Systems

Chong, A., Gedeon, T.D., Wong, K.W.
School of Information Technology
Murdoch University
South Street, Murdoch, WA, 6150 Australia

Koczy, L.T.
Department of Telecom & Telematics
Budapest University of Technology and Economics

Abstract

We propose a histogram-based rule extraction technique using straightforward histogram-based clustering that produces trapezoidal clusters that are well suited for the rule extraction purpose. Two experiments have been carried out to validate the feasibility and effectiveness of the proposed technique and show that the rule base generated by the proposed technique is reasonably accurate.

I. Introduction

Among the rule extraction techniques proposed in the literature, Sugeno and Yasukawa's technique [1] is one of the earliest works that emphasize the generation of a sparse rule-base. The SY approach clusters only the output data and induces the rules by computing the projections to the input domains of the cylindrical extensions of the fuzzy clusters. Thus, it produces only the necessary number of rules for the input-output data. In this paper, we propose an extension to the SY approach by introducing a histogram-based clustering tool that can improve the approach in several aspects.

II. Histograms for Clustering

One of the inexpensive ways to describe the frequency distribution of a set of data is by using a histogram. The X-axis of the graph shows a set of non-overlapping intervals, called bins. The bin counts (i.e. the number of data whose value falls in a particular bin) are shown in the Y-axis.

The use of histograms for clustering can be traced as early as the 60's [2]. It has been a useful clustering tool in the image processing literature [3, 4]. Theoretically, for each cluster in the data set, the histogram will reveal a peak. A peak suggests the existence of a cluster center where the density of the area is high. It is reasonable to assume that the data in a cluster has a normal distribution. If so, the density of data decreases as we move away from a peak. A set of consecutive bars with similar heights in the histogram suggests a certain level of cylindricity in the area (i.e., a cluster whose distribution is less affected by the input). Bars that are low in bin counts and significantly removed from the others are termed outliers. They can be caused by noise in the data samples and are potential candidates to be ignored or removed from the data set.

III. Bin Width Selection

A large bin width causes the histogram to obscure important details of the data. In this case, multiple clusters are likely to be merged and shown as a single peak. As the bin width grows smaller, the histogram becomes more sensitive to the noise in the data, and will begin to form false peaks and eventually the histogram will reveal more peaks than the number of actual clusters in the data. Often, the best choice of bin width is obtained through experimentation on a trial and error basis. Research has been devoted to designing techniques for histogram bin widths selection [5-7]. Unfortunately, a review of the methods suggests that most of them are either suitable for only certain types of data distribution, or too complex to implement efficiently.

The method used in this study to produce a rough estimation of a suitable bin width is as follows. An initial histogram is constructed with a relatively small, n *(e.g. $n=5$)* number of bins. The number of peaks formed in the histogram is then determined. The process is repeated with a slightly increased n (e.g. $n=n+1$) until n reaches a reasonable large value.

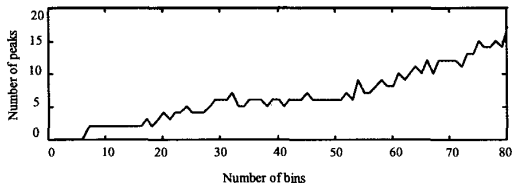

Fig. 3.1 Graph that illustrates the effect of the number-of-bins on the number-of-peaks in a dataset.

The stability region can be determined where the continuous increase of the number-of-bins in the x-axis causes minimum changes to the number-of-peaks in the y-axis. From figure 3.1, the stability region ranges from 7 – 16 bins.

IV. Histogram Smoothing

The goal of the histogram smoothing process is to remove or reduce the occurrence of small false peaks in the histogram, while maintaining the original shape of the histogram. A

0-7803-7293-X/01/$17.00 © 2001 IEEE

simple smoothing technique with the window size of 3 replaces the value, y_i of each bin, with the average $(y_{i-1} + y_i + y_{i+1})/3$. We propose the use of fuzzy smoothing which replaces each y_i with the weighted average:

$$\sum_{i-(n-1)/2}^{i+(n-1)/2} w_i y_i \Big/ \sum_{i=1}^{n} w_i \qquad (4.1)$$

where n is the window size and w_i is the weight. We observe that the use of fuzzy smoothing (as opposed to crisp) enables the histogram to reveal more details that is useful for the cluster approximation process (see section 4.0).

The parameters involved in the smoothing technique are the window size and weights. Increase in the window size results in the increase of the smoothness of the histogram. The smoothing process is further 'fine-tuned' by the use of different combinations of weights. Over-smoothed histograms tends to merge multiple peaks into one whereas under-smoothed histograms fail to eliminate false peaks due to noise in the data. In this study, a smoothing window size of 7 with the weights [1 2 4 6 4 2 1] is used.

V. Trapezoidal Cluster Approximation

Our algorithm for histogram based trapezoidal cluster approximation is:

1. *Peak finding*: We define peaks as bins in the histogram that are higher than their immediate neighboring bins. That is, *bin(i)* is a peak if *bin(i-1) < bin(i) > bin(i+1)*.
2. *Identify the plateau for each peak*: Neighboring bins of a peak are part of the plateau if the difference between their heights and the peak height is lower than the vertical threshold, *vt*. Given the peak at *bin(p)*, *bin(i)* is part of the plateau if *bin(p)-bin(i) < vt*. This is illustrated in figure 5.1(A) where the 2 neighboring bins on the right of the peaks are considered to be part of the plateau whereas the bins on the left are not.
3. *Trapezoidal bases approximation*: To identify the left base, we start with the leftmost bin, *i* in the plateau and move towards the left to examine the neighboring bins *i-1*, *i-2*, ...etc. We stop when we find a bin higher than its right neighboring bin, i.e. *bin (i - (n+1)) > bin (i – n)*. Thus, *bin (i – n)* will be the left trapezoidal base. Figure 5.1(A) illustrates this. A similar algorithm is applied to find the right base. Finally, the trapezoidal clusters are normalized to form convex normal fuzzy clusters.

4. We convert the rough estimations of our trapezoidal clusters by slope adjustment.
   - *Inputs:* Convert to Ruspini partitions [8]. We adjust the slopes for each pair of neighboring clusters according to the following Ruspini condition:

$$\sum_{i=1}^{n} A_i(x) = 1$$

Where x is an arbitrary value in the input range, n is the number of clusters in the dimension, and A_i is the membership function of the i^{th} fuzzy cluster. The physical slope adjustment is shown in figure 5.1(B).

   - *Output: prevent core overlapping*. Each pair of neighboring clusters is converted into Ruspini partitions only if the core of the cluster is overlapped by the other cluster.

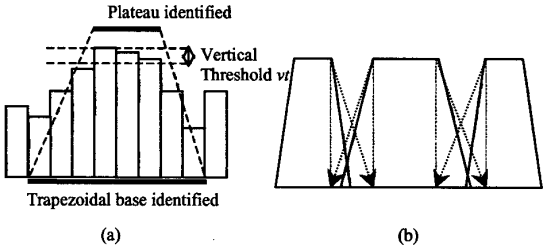

Fig. 5.1a Plateau identification using the vertical threshold vt and trapezoidal base identification; b. Adjusting the clusters according to the Ruspini condition

VI. Rule Extraction Using Histogram-Based Clustering

We propose a novel rule extraction technique that uses histogram-based fuzzy clustering. The technique extends the idea of Sugeno and Yasukawa's approach for fuzzy modeling [1] (abbreviated as SY method hereafter). The goal of the rule extraction technique is to automate the construction of a fuzzy rule base from a set of input-output sample data.

In SY modelling, the Regularity criterion [9] is used to identify input variables that have significant influence on the output. Other input variables are ignored for the rest of the process. The rule extraction process requires the partition of the output space. This is done by fuzzy c-means clustering [9]. The optimal number of clusters are determined by means of the criterion proposed in [9]. For each output fuzzy cluster B_i resulting from the fuzzy c-means clustering, a cluster in the input space A_i can be induced. The input cluster can be projected onto the various input dimensions to produce rules of the form:

If x1 is A_{i1} and x2 is A_{i2} and ... x_n is A_{in} then y is B_i

However, it is remarked in the paper [1] that there can be more than one fuzzy cluster in the input space which corresponds to the same fuzzy cluster B_i. In this case more than one rule is formed with the same consequent. Suppose that two input clusters (A_i and A_j) are induced from the output cluster B_i, we obtain the following two rules:

If x1 is A_{i1} and x2 is A_{i2} and ... x_n is A_{in} then y is B_i
If x1 is A_{j1} and x2 is A_{j2} and ... x_n is A_{jn} then y is B_i

Unfortunately, no concrete procedures for determining the number of input clusters to be induced from an output cluster is discussed in the paper [1].

One of the most important advantages of the rule extraction technique used in the SY approach is that it results in a sparse rule base. Hence we use it as the foundation of our rule extraction technique. As extensions, we propose the use of histogram-based clustering on both the input and output spaces. The use of histogram-based clustering complements the SY approach in the following ways:

1. Fuzzy C-means clustering was originally proposed for circumstances where the number of clusters in the set of data is known in advance. In the SY approach, the optimal number of clusters are determined by means of a criterion [9]. This step can be eliminated using the histogram-based clustering technique.
2. By applying the histogram-based clustering on the input space, the number of input clusters to be induced from an output cluster can be determined automatically.
3. The trapezoidal clusters approximated using the histogram approach gives an idea of the cylindricity of clusters. An input dimension that has cylindric clusters suggests that the corresponding input variable is less likely to be a significant input variable and could be ignored. Information about cluster cylindricity is also useful in the construction of a hierarchical fuzzy rule bases[10], which we will investigate in future work.

Our novel histogram-based rule extraction technique is:

1. Perform the histogram-based fuzzy clustering on the output space. We assume the output space only has one dimension. This is reasonable, since multi-dimensional output can be simply split multiple single output datasets.
2. For each trapezoidal output cluster B_i approximated, all the points belonging to the cluster are projected back to each of the input dimensions. For each dimension, the histogram-based clustering is applied to identify trapezoidal clusters. It is well known that the generation of histograms is not computationally intensive ($O(n)$). Therefore, the application of the clustering technique to the various input dimensions will not add significantly to the overall computational complexity.
3. The previous step results in multiple 1D trapezoidal clusters in each input dimension. Each of the n clusters ($C_{d1} - C_{dn}$) in the input dimension d, is a projection of the multi-dimensional input cluster to that input dimension. Next the clusters from individual dimensions are combined to form the multi-dimensional input cluster. This is done by projecting the individual points from the output cluster for the second time. The locations of the projected points in the multi-dimensional space show us the right combination of the 1D clusters to form the multidimensional input cluster. For an example of a fuzzy rule base with 3 input variables, suppose that most of the points projected from output cluster B_i fall into the clusters C_{11}, C_{23}, C_{34}, we obtain the following rule:

If x_1 is C_{11} and x_2 is C_{23} and x_3 is C_{34} then y is B_i

where C_{dn} is the n^{th} cluster of input dimension d. The cluster in the multi-dimensional space is the region where the number of projected points in the region exceeds threshold t. For regions that have the number of points < t, all points in the region are considered as noise. We propose t to be the average of the population. To merge the clusters in high dimensions, keeping track of the multi-dimensional points requires a large amount of memory. It is more memory efficient to merge two dimensions at a time. For example, to merge the clusters from the three input dimensions shown in 6.1(A), we first prepare 4 counters arranged in a grid layout like the one in figure 6.1(B). Points in the output cluster B_i are projected back to the input space and the relevant counter values are incremented according to the region where the projected points fall. This is followed by the comparisons of the counter values with the threshold t, calculated here to be 50.5 (see figure 6.1B), to determine the potential 2D clusters. From the figure, we obtain the potential 2D clusters as [C_{11} , C_{21}] and [C_{12} , C_{22}]. Next, the third input dimension is considered. A new grid layout of counters, as illustrated in figure 6.1C, is prepared and the process is repeated. Finally, the 3D clusters have been identified as clusters [C_{11} , C_{21} , C_{32}] and [C_{11} , C_{21} , C_{31}] and the following two rules are formed:

If x_1 is C_{11} and x_2 is C_{21} and x_3 is C_{32} then y is B_i

If x_1 is C_{11} and x_2 is C_{21} and x_3 is C_{31} then y is B_i

4. Repeat the steps 2 – 3 until all output clusters are examined.

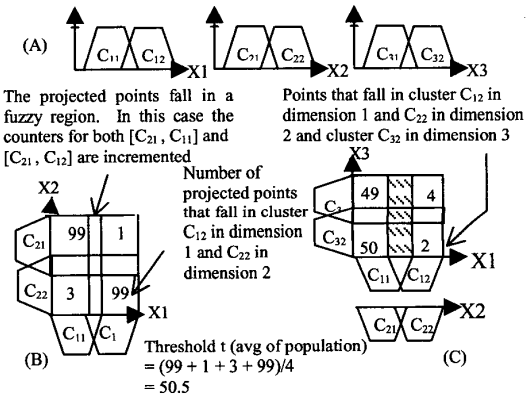

Fig. 6.1a Clusters at each dimension; b. Merging 1st and 2nd dimension; c. Merging results with 3rd dimension.

VII. Results and Discussion

To demonstrate the feasibility and effectiveness of the proposed approach, two experiments have been carried out. In each experiment, a fuzzy rule base is generated from a set

of input-output sample data using the proposed rule extraction technique. The same set of data is then used with the fuzzy rule base generated to produce a set of outputs. This is followed by the evaluation of the accuracy of the fuzzy rule base output. In this study, the mean square error of output is used as a performance index (equation 6.1).

$$PI = \sum_{i=1}^{m} (y^i - \hat{y}^i)^2 / m \qquad \text{(eqn 6.1)}$$

where m is the number of data, y^i is the i^{th} actual output and $\hat{y}^i$ is the i^{th} model output. The lower the performance index, the more accurate the fuzzy rule base.

As outlined in section 6.0, some of the points in the data set are treated as noise. These points fall outside the range of the multi-dimensional clusters identified. Hence, they are not covered by any of the fuzzy rules generated. We remark that this may not be a failure or weakness of the algorithm. After all, one of the ultimate goals of our technique is to generate a sparse rule base. For data points that can not find rules to fire, fuzzy rule interpolation techniques [11] and extrapolation [1] can be used to infer the output. These techniques have not been used here. In the experiments, when there are no rules to fire for a data point, the average of the range of each output variable is used as the default output.

For each experiment, two sets of results are presented. The first set shows the performance index of the whole data set whereas the second set shows the performance index on only those data that can find rules to fire. The latter gives a better idea of the potential of this fuzzy rule base to be used with fuzzy rule interpolation and extrapolation techniques.

Experiment 1: Modeling of a quadratic function

The method was used to model a quadratic function (equ 7.1).

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2, \quad 1 \le x_1, x_2 \le 5 \qquad \text{(eqn 7.1)}$$

Fifty input-output data were generated from the quadratic function. The output range is 1.3-5.1. The technique discussed in section 3.0 is used to select the bin width for the histogram construction. The stability region was identified at 8-16 bins. Nine bins was chosen for the histogram generation. At the output space, 2 clusters were identified and altogether 6 fuzzy rules were generated. During the evaluation, 14% of the data points cannot find rules to fire. The performance index of the data that can find rules to fire is 0.643. The overall performance index of the system is 0.966.

Experiment 2: Modeling of employee salary data

The data is a case study on employee salary based on age, experience and contacts. A simulated case is used to generate a total of 200 sample data (3 dimensional input). The output range is 25-75. The stability region was identified at 7 – 16 bins. Nine bins was chosen for the histogram generation and 2 clusters are identified in the output space. Altogether 7 fuzzy rules were generated. There are 17.5% of the data points that cannot find rules to fire. The performance index of the data that can find rules to fire is 93.40 and the overall performance index of the system is 174.02.

VIII. Conclusion

A novel histogram-based rule extraction technique has been proposed. Sugeno and Yasukawa's approach for fuzzy modeling [1] has been used as a foundation. As extensions to the SY approach, the use of a histogram-based clustering tool is proposed to simplify and improve several steps of the methodology. Experiments have been carried out to validate the feasibility and effectiveness of the proposed technique. Two sets of data have been used in the experiments. The sparse rule bases generated from the technique were evaluated by computing a performance index and the results were presented. It was shown that the rule base generated by the proposed technique is reasonably accurate.

References

[1] Sugeno, M. and Yasukawa, T., *A fuzzy-logic-based approach to qualitative modeling.* IEEE Transactions on Fuzzy Systems, 1993. **1**(1): p. 7-31.

[2] Prewitt, J.S.M. and Mendelsonn, M.L., *The analysis of cell images.* Annual New York Acad. Sci., 1966. **128**: p. 1035-1053.

[3] Glasbey, C.A., *An analysis of histogram-based thresholding algorithms.* Graphical Models and Image Processing, 1993. **55**(6): p. 532-537.

[4] Puzicha, J., Hoftmann, T., and Buhmann, J.M., *Histogram clustering for unsupervised segmentation and image retrieval.* Pattern Recognition Letters, 1999. **20**: p. 899-909.

[5] Sturges, H.A., *The choise of a class interval.* J. Amer. Statist. Assoc., 1926. **21**: p. 65-66.

[6] Simonoff, J.S., *Smoothing Methods in Statistics.* 1 ed. 1996, New York: Springer.

[7] Turkey, J.W., *Exploratory Data Analysis.* 1977, US: Addison Wesley.

[8] Ruspini, E.H., *A new approach to clustering.* Information and Control, 1969. **15**: p. 22-32.

[9] Bezdek, J.C., *Pattern Reconition with Fuzzy Objective Function Algorithms.* 1981, New York: Plenum Press.

[10] Koczy, L.T. *Approximative inference in hierarchical structured rule bases.* in *Fift IFSA World Congress.* 1993. Seoul: International Fuzzy Systems Association.

[11] Gedeon, T.D. and Koczy, L.T., *Conservation of fuzziness in rule interpolation.* Intelligent Technologies, 1996. **1**: p. 13-19.