Computational Intelligence in Data Mining

Janos Abonyi and Balazs Feil
University of Veszprem, Department of Process Engineering,
P.O. Box 158, H-8201 Veszprem, Hungary, abonyij@fmt.vein.hu
www.fmt.vein.hu/softcomp

Ajith Abraham
School of Computer Science and Engineering,
Chung-Ang University, Seoul, S. Korea, ajith.abraham@ieee.org
http://ajith.softcomputing.net

Keywords: KDD, Computational Intelligence, Soft Computing, Fuzzy Classifier System, Rule Base Reduction, Visualization

Received: December 20, 2004

This paper is aimed to give a comprehensive view about the links between computational intelligence and data mining. Further, a case study is also given in which the extracted knowledge is represented by fuzzy rule-based expert systems obtained by soft computing based data mining algorithms. It is recognized that both model performance and interpretability are of major importance, and effort is required to keep the resulting rule bases small and comprehensible. Therefore, CI technique based data mining algorithms have been developed for feature selection, feature extraction, model optimization and model reduction (rule base simplification). Application of these techniques is illustrated using the Wine data classification problem. The results illustrate that that CI based tools can be applied in a synergistic manner though the nine steps of knowledge discovery.

Povzetek:

1 Introduction

In our society the amount of data doubles almost every year. Hence, there is an urgent need for a new generation of computationally intelligent techniques and tools to assist humans in extracting useful information (knowledge) from the rapidly growing volume of data.

Historically the notion of finding useful patterns in data has been given a variety of names including data mining, knowledge extraction, information discovery, and data pattern processing. The term data mining has been mostly used by statisticians, data analysts, and the management information systems (MIS) communities.

The term knowledge discovery in databases (KDD) refers to the overall process of discovering knowledge from data, while data mining refers to a particular step of this process. Data mining is the application of specific algorithms for extracting patterns from data [1]. The additional steps in the KDD process, such as data selection, data cleaning, incorporating appropriate prior knowledge, and proper interpretation of the results are essential to ensure that useful knowledge is derived form the data.

KDD has evolved from the intersection of research fields such as machine learning, pattern recognition, databases, statistics, artificial intelligence, and more recently it gets new inspiration from computational intelligence.

When we attempt to solve real-world problems, like extracting knowledge from large amount of data, we realize that they are typically ill-defined systems, difficult to model and with large-scale solution spaces. In these cases, precise models are impractical, too expensive, or non-existent. Furthermore, the relevant available information is usually in the form of empirical prior knowledge and input–output data representing instances of the system's behavior. Therefore, we need an approximate reasoning system capable of handling such imperfect information. While Bezdek [2] defines such approaches within a frame called computational intelligence, Zadeh [3] explains the same using the soft computing paradigm. According to Zadeh "... in contrast to traditional, hard computing, soft computing is tolerant of imprecision, uncertainty, and partial truth." In this context Fuzzy Logic (FL), Probabilistic Reasoning (PR), Neural Networks (NNs), and Evolutionary Algorithms (EAs) are considered as main components of CI. Each of these technologies provide us with complementary **reasoning** and **searching** methods to solve complex, real-world problems. What is important to note is that soft computing is not a melange. Rather, it is a partnership in which each of the partners contributes a distinct methodology for addressing problems in its domain. In this perspective, the principal constituent methodologies in CI are complementary rather than competitive [4].

The aim of this paper is to illustrate how these elements of CI could be used in data mining. This special issue is

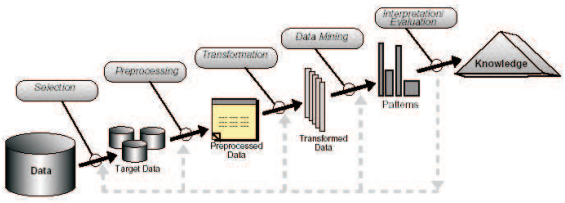

Figure 1: Steps of the knowledge discovery process.

focused on some of the theoretical developments and advances in this field.

Steps of Knowledge Discovery

Brachman and Anand [5] give a practical view of the KDD process emphasizing the interactive nature of the process. Here we broadly outline some of its basic steps depicted in Fig. 1 taken from [6], and we show the connections of these steps to CI based models and algorithms.

1. *Developing and understanding the application domain, the relevant prior knowledge, and identifying the goal of the KDD process.* The transparency of fuzzy systems allows the user to effectively combine different types of information, namely linguistic knowledge, first-principle knowledge and information from data. An example for the incorporation of prior knowledge into data-driven identification of dynamic fuzzy models of the Takagi-Sugeno type can be found in [7] where the prior information enters to the model through constraints defined on the model parameters. In [8] and [9] a different approach has been developed which uses block-oriented fuzzy models.

2. *Creating target data set.*

3. *Data cleaning and preprocessing*: basic operations such as the removal of noise, handling missing data fields.

4. *Data reduction and projection*: finding useful features to represent the data depending the goal of the task. Using dimensionality reduction or transformation methods to reduce the effective number of variables under consideration or to find invariant representation of data. Neural networks [10], cluster analysis [11], Markov blanket modeling [12], decision trees [13], evolutionary computing [14] and neuro-fuzzy systems are often used for this purpose.

5. *Matching the goals of the KDD process to a particular* **data mining method**: Although the boundaries between prediction and description are not sharp, the distinction is useful for understanding the overall discovery goal. The goals of knowledge discovery are achieved via the following data mining methods:

   - **Clustering**: Identification of a finite set of categories or clusters to describe the data. Closely related to clustering is the method of probability density estimation. Clustering quantizes the available input-output data to get a set of prototypes and use the obtained prototypes (signatures, templates, etc., and many writers refer to as codebook) and use the prototypes as model parameters.
   - **Summation**: finding a compact description for subset of data, e.g. the derivation of summary for association of rules and the use of multivariate visualization techniques.
   - **Dependency modeling**: finding a model which describes significant dependencies between variables (e.g. learning of belief networks).
   - **Regression**: learning a function which maps a data item to a real-valued prediction variable and the discovery of functional relationships between variables.
   - **Classification**: learning a function that maps (classifies) a data item into one of several predefined classes.
   - **Change and Deviation Detection:** Discovering the most significant changes in the data from previously measured or normative values.

6. *Choosing the* **data mining algorithm(s)**: selecting algorithms for searching for patterns in the data. This includes deciding which model and parameters may be appropriate and matching a particular algorithm with the overall criteria of the KDD process (e.g. the end-user may be more interested in understanding the model than its predictive capabilities.) One can identify three primary components in any data mining algorithm: model representation, model evaluation, and search.

   - **Model representation**: the language is used to describe the discoverable patterns. If the representation is too limited, then no amount of training time or examples will produce an accurate model for the data. Note that more powerful representation of models increases the danger of overfitting the training data resulting in reduced prediction accuracy on unseen data. It is important that data analysts fully comprehend the representational assumptions which may be inherent in a particular method.

     For instance, rule-based expert systems are often applied to classification problems in fault detection, biology, medicine etc. Among the wide range of CI techniques, fuzzy logic improves classification and decision support systems by allowing the use of overlapping class definitions and improves the interpretability of the

results by providing more insight into the classifier structure and decision making process [15]. In Section 2 a detailed discussion about the use of fuzzy techniques for knowledge representation in classifier systems will be given.

- **Model evaluation criteria**: qualitative statements or fit functions of how well a particular pattern (a model and its parameters) meet the goals of the KDD process. For example, predictive models can often judged by the empirical prediction accuracy on some test set. Descriptive models can be evaluated evaluated along the dimensions of predictive accuracy, novelty, utility, and understandability of the fitted model.

  Traditionally, algorithms to obtain classifiers have focused either on accuracy or interpretability. Recently some approaches to combining these properties have been reported; fuzzy clustering is proposed to derive transparent models in [16], linguistic constraints are applied to fuzzy modeling in [15] and rule extraction from neural networks is described in [17]. Hence, to obtain compact and interpretable fuzzy models, reduction algorithms have to be used that will be overviewed in Section 3.

- **Search method**: consists of two components: parameter search and model search. Once the model representation and the model evaluation criteria are fixed, then the data mining problem has been reduced to purely an optimization task: find the parameters/models for the selected family which optimize the evaluation criteria given observed data and fixed model representation. Model search occurs as a loop over the parameter search method [18].

  The automatic determination of fuzzy classification rules from data has been approached by several different techniques: neuro-fuzzy methods [19], genetic-algorithm based rule selection [20], hybrid combination of genetic algorithm and neural learning [21] and fuzzy clustering in combination with GA-optimization [22] [23]. For high-dimensional classification problems, the initialization step of the identification procedure of the fuzzy model becomes very significant. Several CI based tools developed for this purpose will be presented in Section 4.

7. *Data mining*: searching for patterns of interest in a particular representation form or a set of such representations: classification rules or trees, regression. Some of the CI models lend themselves to transform into other model structure that allows information transfer between different models. For example, in [24] a decision tree was mapped into a feedforward neural network. A variation of this method is given in [25] where the decision tree was used for the input domains discretization only. This approach was extended with a model pruning method in [26]. Another example is that as radial basis functions (RBF) are functionally equivalent to fuzzy inference systems [27, 28], tools developed for the identification of RBFs can also be used to design fuzzy models.

8. *Interpreting mined patterns*, possibly return to any of the steps 1-7 described above for further iteration. This step can also involve the visualization of the extracted patterns/models, or visualization of the data given the extracted models. Self-Organizing Map (SOM) as a special clustering tool that provides a compact representation of the data distribution, hence it has been widely applied in the visualization of high-dimensional data [29]. In Section 5 the theory and in Section 6 the application of SOM will be presented.

9. *Consolidating discovered knowledge*: incorporating this knowledge into another system for further action, or simply documenting and reporting it.

The remainder of this article is organized as follows. In the remaining sections, tools for visualization, knowledge representation, classifier identification and reduction are discussed. The proposed approaches are experimentally evaluated for the three-class Wine classification problem. Finally, conclusions are given in Section 7.

2 Effective Model Representation by Fuzzy Systems

2.1 Classifier Systems

The identification of a classifier system means the construction of a model that predicts whether a given pattern, $\boldsymbol{x}_k = [x_{1,k}, \ldots, x_{n,k},]$, in which $y_k = \{c_1, \ldots, c_C\}$ class should be classified. The classic approach for this problem with C classes is based on Bayes' rule. The probability of making an error when classifying an example $\boldsymbol{x}$ is minimized by Bayes' decision rule of assigning it to the class with the largest posterior probability:

$$\boldsymbol{x} \text{ is assigned to } c_i \Longleftrightarrow p(c_i|\boldsymbol{x}) \geq p(c_j|\boldsymbol{x})\, \forall j \neq i \quad (1)$$

The *a posteriori* probability of each class given a pattern $\boldsymbol{x}$ can be calculated based on the $p(\boldsymbol{x}|c_i)$ class conditional distribution, which models the density of the data belonging to the c_i class, and the $P(c_i)$ class prior, which represents the probability that an arbitrary example out of data belongs to class c_i

$$p(c_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|c_i)P(c_i)}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{x}|c_i)P(c_i)}{\sum_{j=1}^{C} p(\boldsymbol{x}|c_j)P(c_j)} \quad (2)$$

As (1) can be rewritten using the numerator of (2) we would have an optimal classifier if we would perfectly estimate the class priors and the class conditional densities. Of

course in practice one needs to find approximate estimates of these quantities on a finite set of training data $\{\boldsymbol{x}_k, y_k\}$, $k = 1, \ldots, N$. Priors $P(c_i)$ are often estimated on the basis of the training set as the proportion of samples of class c_i or using prior knowledge. The $p(c_i|\boldsymbol{x})$ class conditional densities can be modeled with non-parametric methods like histograms, nearest-neighbors or parametric methods such as mixture models.

2.2 Fuzzy Rules for Providing Interpretability of Classifiers

The classical fuzzy rule-based classifier consists of fuzzy rules that each describe one of the C classes. The rule antecedent defines the operating region of the rule in the n-dimensional feature space and the rule consequent is a crisp (non-fuzzy) class label from the $\{c_1, \ldots, c_C\}$ set:

$$r_i : \textbf{If } x_1 \text{ is } A_{i,1}(x_{1,k}) \textbf{ and } \ldots x_n \text{ is } A_{i,n}(x_{n,k}) \textbf{ then } \hat{y} = c_i, \quad [w_i] \tag{3}$$

where $A_{i,1}, \ldots, A_{i,n}$ are the antecedent fuzzy sets and w_i is a certainty factor that represents the desired impact of the rule. The value of w_i is usually chosen by the designer of the fuzzy system according to his or her belief in the accuracy of the rule. When such knowledge is not available, $w_i = 1, \forall\, i$ is used.

The **and** connective is modeled by the product operator allowing for interaction between the propositions in the antecedent. Hence, the degree of activation of the ith rule is calculated as:

$$\beta_i(\boldsymbol{x}_k) = w_i \prod_{j=1}^{n} A_{i,j}(x_{j,k}) \tag{4}$$

The output of the classical fuzzy classifier is determined by the *winner takes all* strategy, i.e. the output is the class related to the consequent of the rule that has the highest degree of activation:

$$\hat{y}_k = c_i^* \ , \ i^* = \arg\max_{1 \le i \le C} \ \beta_i(\boldsymbol{x}_k) \tag{5}$$

The fuzzy classifier defined by the previous equations is in fact a quadratic Bayes classifier when $\beta_i(\boldsymbol{x}_k) = p(\boldsymbol{x}|c_i)P(c_i)$.

As the number of the rules in the above representation is equal to the number of the classes, the application of this classical fuzzy classifier is restricted. In the [30], a new rule-structure has been derived to avoid this problem, where the $p(c_i|\boldsymbol{x})$ posteriori densities are modeled by $R > C$ mixture of models

$$p(c_i|\boldsymbol{x}) = \sum_{l=1}^{R} p(r_l|\boldsymbol{x})P(c_i|r_l) \tag{6}$$

This idea results in fuzzy rulebase where the consequent of rule defines the probability of the given rule represents the $c_1, \ldots, c_C$ classes:

$$r_i : \textbf{If } x_1 \text{ is } A_{i,1}(x_{1,k}) \textbf{ and } \ldots x_n \text{ is } A_{i,n}(x_{n,k}) \textbf{ then } \hat{y}_k = c_1 \text{ with } P(c_1|r_i) \ \ldots, \ \hat{y}_k = c_C \text{ with } P(c_C|r_i) \ [w_i] \tag{7}$$

The aim of the remaining part of the paper is to review some techniques for the identification of the fuzzy classifier presented above. In addition, methods for reduction of the model will be described.

3 Model Evaluation Criteria and Rule Base Reduction

Traditionally, algorithms to obtain best classifiers have been based either on accuracy or interpretability. Recently some approaches to combining these properties have been reported; fuzzy clustering is proposed to derive transparent models in [16], linguistic constraints are applied to fuzzy modeling in [15] and rule extraction from neural networks is described in [17].

3.1 Similarity-driven rule base simplification

The similarity-driven rule base simplification method [31] uses a similarity measure to quantify the redundancy among the fuzzy sets in the rule base. A similarity measure based on the set-theoretic operations of intersection and union is applied:

$$S(A_{i,j}, A_{l,j}) = \frac{|A_{i,j} \cap A_{l,j}|}{|A_{i,j} \cup A_{l,j}|} \tag{8}$$

where $|.|$ denotes the cardinality of a set, and the $\cap$ and $\cup$ operators represent the intersection and union of fuzzy sets, respectively. S is a symmetric measure in [0,1]. If $S(A_{i,j}, A_{l,j}) = 1$, then the two membership functions $A_{i,j}$ and $A_{l,j}$ are equal. $S(A_{i,j}, A_{l,j})$ becomes 0 when the membership functions are non-overlapping. The complete rule base simplification algorithm is given in [31].

Similar fuzzy sets are merged when their similarity exceeds a user defined threshold $\theta \in [0, 1]$ (θ=0.5 is applied). Merging reduces the number of different fuzzy sets (linguistic terms) used in the model and thereby increases the transparency. The similarity measure is also used to detect "don't care" terms, i.e., fuzzy sets in which all elements of a domain have a membership close to one. If all the fuzzy sets for a feature are similar to the universal set, or if merging led to only one membership function for a feature, then this feature is eliminated from the model. The method is illustrated in Fig. 2

3.2 Multi-Objective Function for GA based Identification

To improve the classification capability of the rule base, genetic algorithm (GA) optimization method can be applied [32] where the cost function is based on the model

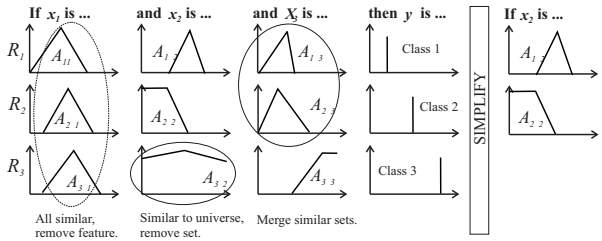

Figure 2: Similarity-driven simplification.

accuracy measured in terms of the number of misclassifications. Also other model properties can be optimized by applying multi-objective functions. For example in [33] to reduce the model complexity, the misclassification rate is combined with a similarity measure in the GA objective function. Similarity is rewarded during the iterative process, that is, the GA tries to emphasize the redundancy in the model. This redundancy is then used to remove unnecessary fuzzy sets in the next iteration. In the final step, fine tuning is combined with a penalized similarity among fuzzy sets to obtain a distinguishable term set for linguistic interpretation.

The GAs is subject to minimize the following multi-objective function:

$$J = (1 + \lambda S^*) \cdot Error\,, \tag{9}$$

where $S^* \in [0, 1]$ is the average of the maximum pairwise similarity that is present in each input, i.e., S^* is an aggregated similarity measure for the total model. The weighting function $\lambda \in [-1, 1]$ determines whether similarity is rewarded ($\lambda < 0$) or penalized ($\lambda > 0$).

3.3 Other Reduction Algorithms

The application of orthogonal transforms for reducing the number of rules has received much attention in recent literature [34]. These methods evaluate the output contribution of the rules to obtain an importance ordering. For modeling purpose Orthogonal Least Squares (OLS) is the most appropriate tool [35]. Evaluating only the approximation capabilities of the rules, the OLS method often assigns high importance to a set of redundant or correlated rules. To avoid this, in [36] some extension for the OLS method was proposed.

Using too many input variables may result in difficulties in the interpretability capabilities of the obtained classifier. Hence, selection of the relevant features is usually necessary. Others have focused on reducing the antecedent by similarity analysis of the fuzzy sets [33], however this method is not very suitable for feature selection. Hence, for this purpose, Fischer interclass separability method which is based on statistical properties of the data [37] has been modified in [38].

4 CI based Search Methods for the Identification of Fuzzy Classifiers

Fixed membership functions are often used to partition the feature space [20]. Membership functions derived from the data, however, explain the data-patterns in a better way. The automatic determination of fuzzy classification rules from data has been approached by several different techniques: neuro-fuzzy methods [19], genetic-algorithm based rule selection [20] and fuzzy clustering in combination with GA-optimization [22]. For high-dimensional classification problems, the initialization step of the identification procedure of the fuzzy model becomes very significant. Common initializations methods such as grid-type partitioning [20] and *rule generation on extrema* initialization [39], result in complex and non-interpretable initial models and the rule-base simplification and reduction step become computationally demanding.

4.1 Identification by Fuzzy Clustering

To obtain compact initial fuzzy models fuzzy clustering algorithms [22] or similar but less complex covariance based initialization techniques [38] were put forward, where the data is partitioned by ellipsoidal regions (multivariable membership functions). Normal fuzzy sets can then be obtained by an orthogonal projection of the multivariable membership functions onto the input-output domains. The projection of the ellipsoids results in hyperboxes in the product space. The information loss at this step makes the model suboptimal resulting in a much worse performance than the initial model defined by multivariable membership functions. However, gaining linguistic interpretability is the main advantage derived from this step. To avoid the erroneous projection step multivariate membership functions [40] or clustering algorithms providing axis-parallel clusters can be used [30]

4.2 Other Initialization Algorithms

For the effective initialization of fuzzy classifiers crisp decision tree-based initialization technique is proposed in [41]. DT-based classifiers perform a rectangular partitioning of the input space, while fuzzy models generate non-axis parallel decision boundaries [42]. Hence, the main advantage of rule-based fuzzy classifiers over crisp-DTs is the greater flexibility of the decision boundaries. Therefore fuzzy classifiers can be more parsimonious than DTs and one may conclude that the fuzzy classifiers, based on the transformation of DTs only [43], [44] will usually be more complex than necessary. This suggests that the simple transformation of a DT into a fuzzy model may be successfully followed by model reduction steps to reduce the complexity and improve the interpretability. The next section proposes rule-base optimization and simplification steps for this purpose.

5 Clustering by SOM for Visualization

The Self-Organizing Map (SOM) algorithm performs a topology preserving mapping from high dimensional space onto map units so that relative distances between data points are preserved. The map units, or neurons, form usually a two dimensional regular lattice. Each neuron i of the SOM is represented by an l-dimensional weight, or model vector $\boldsymbol{m}_i = [m_{i,1}, \ldots, m_{i,l}]^T$. These weight vectors of the SOM form a codebook. The neurons of the map are connected to adjacent neurons by a neighborhood relation, which dictates the topology of the map. The number of the neurons determines the granularity of the mapping, which affects the accuracy and the generalization capability of the SOM.

SOM is a vector quantizer, where the weights play the role of the codebook vectors. This means, each weight vector represents a local neighborhood of the space, also called Voronoi cell. The response of a SOM to an input $\boldsymbol{x}$ is determined by the reference vector (weight) $\boldsymbol{m}_i^0$ which produces the best match of the input

$$i^0 = \arg\min_i \ \|\boldsymbol{m}_i - \boldsymbol{x}\| \tag{10}$$

where i^0 represents the index of the Best Matching Unit (BMU).

During the iterative training, the SOM forms an elastic net that folds onto "cloud" formed by the data. The net tends for approximate the probability density of the data: the codebook vectors tend to drift there where the data are dense, while there are only a few codebook vectors where the data are sparse. The training of SOM can be accomplished generally with a competitive learning rule as

$$\boldsymbol{m}_i^{(k+1)} = \boldsymbol{m}_i^{(k)} + \eta \Lambda_{i^0,i}(\boldsymbol{x} - \boldsymbol{m}_i^{(k)}) \tag{11}$$

where $\Lambda_{i^0,i}$ is a spatial neighborhood function and η is the learning rate. Usually, the neighborhood function is

$$\Lambda_{i^0,i} = \exp\left(\frac{\|\boldsymbol{r}_i - \boldsymbol{r}_i^0\|^2}{2\sigma^{2(k)}}\right) \tag{12}$$

where $\|\boldsymbol{r}_i - \boldsymbol{r}_i^0\|$ represents the Euclidean distance in the output space between the i-th vector and the winner.

6 Case study: Wine Classification by CI techniques

6.1 Wine Data

The Wine data [1] contains the chemical analysis of 178 wines grown in the same region in Italy but derived from three different cultivars. The problem is to distinguish the three different types based on 13 continuous attributes derived from chemical analysis. : Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Non-flavanoid phenols, Proanthocyaninsm color intensity, Hue, OD280/OD315 of diluted wines and Proline (Fig. 3).

6.2 Fuzzy Classifier Identified by GA

An initial classifier with three rules was constructed by the covariance–based model initialization technique proposed in [38] using all samples resulting in 90.5% correct, 1.7% undecided and 7.9% misclassifications for the three wine classes. Improved classifiers are developed based on the GA based optimization technique discussed in Section 3.2. Based on the similarity analysis of the optimized fuzzy sets, some features have been removed from individual rules, while the interclass separability method have been used to omit some features in all the rules. The achieved membership functions are shown in Fig. 4, while the obtained rules are shown in Table 1.

6.3 Fuzzy Classifier Identified by Fuzzy Clustering

A fuzzy classifier, that utilizes all the 13 information profile data about the wine, has been identified by the clustering algorithm proposed in [30], where the obtained classifier is formulated by rules given by (7). Fuzzy models with three and four rules were identified. The three rule-model gave only 2 misclassification (98.9%). When a cluster was

[1] The Wine data is available from the University of California, Irvine, via anonymous ftp *ftp.ics.uci.edu/pub/machine-learning-databases.*

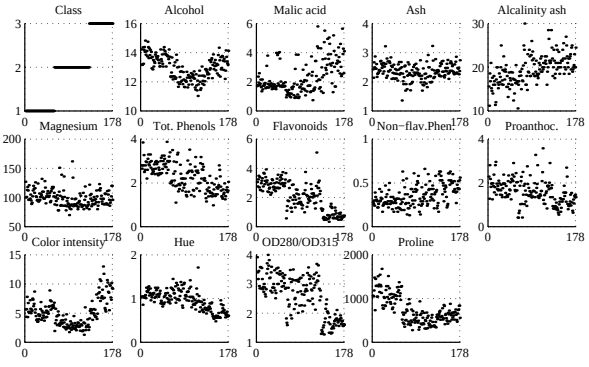

Figure 3: Wine data: 3 classes and 13 attributes.

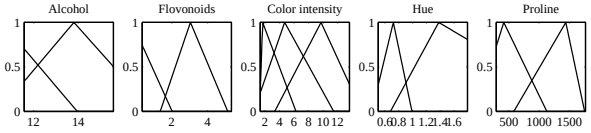

Figure 4: The fuzzy sets of the optimized three rule classifier for the Wine data.

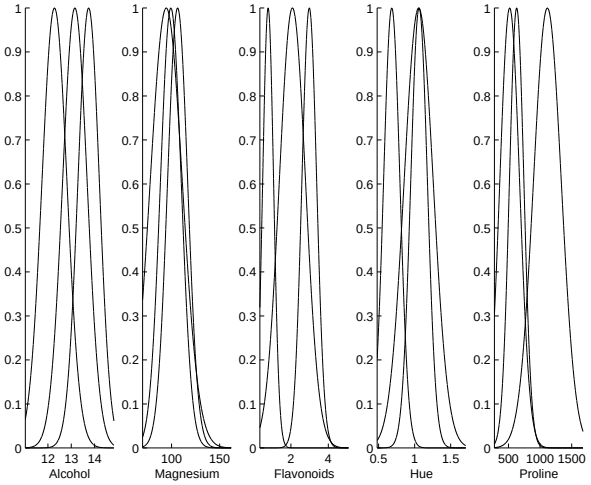

Figure 5: Membership functions obtained by fuzzy clustering.

added to improve the performance of this model, the obtained classifier gave only 1 misclassification (99.4%).

The classification power of the identified models is compared with fuzzy models with the same number of rules obtained by Gath-Geva clustering, as Gath-Geva clustering can be considered the unsupervised version of the proposed clustering algorithm. The Gath-Geva identified fuzzy model gives 8 (95.5%) misclassification when the fuzzy model has three rules and 6 (96.6%) misclassification with four rules. These results indicate that the proposed clustering method effectively utilizes the class labels.

The interclass separability based model reduction technique is applied to remove redundancy and simplify the obtained fuzzy models and five features were selected. The clustering has been applied again to identify a model based on the selected five attributes. This compact model with three, four and five rules gives four, two and zero misclassification, respectively. The resulted membership functions and the selected features are shown in Fig. 5.

6.4 Visualization by SOM

The SOM presented in Section 5. has been utilized to visualize the Wine data. SOM can be effectively used for correlation hunting, which procedure is useful for detecting the redundant features. It is interesting to note that the rules given in Table 1 can easily validated by the map of the variables given in Fig. 6

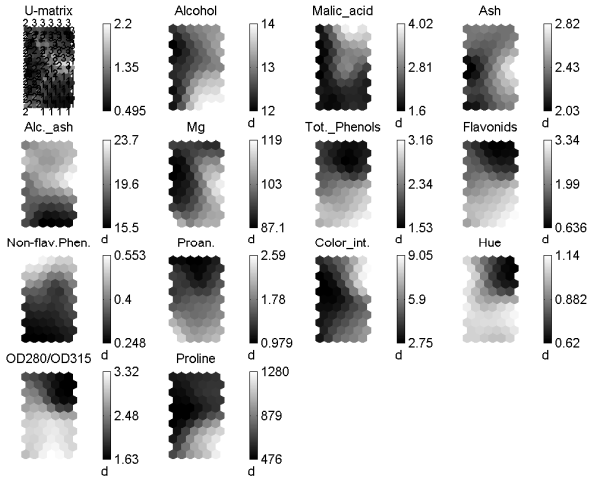

Figure 6: Self-Organizing Map of the Wine data

6.5 Discussion

The Wine data is widely applied for comparing the capabilities of different data mining tools. Corcoran and Sen [45] applied all the 178 samples for learning 60 non-fuzzy if-then rules in a real-coded genetic based-machine learning approach. They used a population of 1500 individuals and applied 300 generations, with full replacement, to come up with the following result for ten independent trials: best classification rate 100%, average classification rate 99.5% and worst classification rate 98.3% which is 3 misclassifications. Ishibuchi et al. [20] applied all the 178 samples designing a fuzzy classifier with 60 fuzzy rules by means of an integer-coded genetic algorithm and grid partitioning. Their population contained 100 individuals and they applied 1000 generations, with full replacement, to come up with the following result for ten independent trials: best classification rate 99.4% (1 misclassifications), average classification rate 98.5% and worst classification rate 97.8% (4 misclassifications). In both approaches the final rule base contains 60 rules. The main difference is the number of model evaluations that was necessary to come to the final result.

As can be seen from Table 2, because of the simplicity of the proposed clustering algorithm, the proposed approach is attractive in comparison with other iterative and optimization schemes that involves extensive intermediate optimization to generate fuzzy classifiers.

The results are summarized in Table 2. As it is shown, the performance of the obtained classifiers are comparable to those in [45] and [20], but use far less rules (3-5 compared to 60) and less features.

Comparing the fuzzy sets in Fig. 5 with the data in Fig. 3 shows that the obtained rules are highly interpretable. For example, the Flavonoids are divided in Low, Medium and High, which is clearly visible in the data. This knowledge can be easily validated by analyzing the SOM of the data given in Fig. 6.

7 Conclusion

The design of rule base classifiers is approached by combining a wide range of CI tools developed for knowledge representation (fuzzy rules), feature selection (class separability criterion), model initialization (clustering and decision tree), model reduction (orthogonal methods) and tuning (genetic algorithm). It has been shown that these tools can be applied in a synergistic manner though the nine steps of knowledge discovery.

References

[1] U. Fayyad, G. Piatestku-Shapio, P. Smyth, R. Uthurusamy, Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1996.

[2] J. Bezdek, Computational intelligence definedŰby everyone!, in: Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications, O. Kaynak et al. (Eds.), Springer Verlag, Germany, 1996.

[3] L. Zadeh, Roles of soft computing and fuzzy logic in the conception, design and deployment of information/intelligent systems, in: Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications, O. Kaynak et al. (Eds.), Springer Verlag, Germany, pp. 1-9, 1998.

[4] A. Abraham, Intelligent systems: Architectures and perspectives, Recent Advances in Intelligent Paradigms and Applications, Abraham A., Jain L. and Kacprzyk J. (Eds.), Studies in Fuzziness and Soft Computing, Springer Verlag Germany, ISBN 3790815381, Chapter 1, pp. 1-35 .

[5] R. Brachman, T. Anand, The process of knowledge discovery in databases, in: Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1994, pp. 37–58.

[6] U. Fayyad, G. Piatestku-Shapio, P. Smyth, Knowledge discovery and data mining: Towards a unifying framework, in: Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1994.

[7] J. Abonyi, R. Babuska, H. Verbruggen, F. Szeifert, Using a priori knowledge in fuzzy model identification, International Journal of Systems Science 31 (2000) 657–667.

[8] J. Abonyi, L. Nagy, F. Szeifert, Hybrid fuzzy convolution modelling and identification of chemical process systems, International Journal of Systems Science 31 (2000) 457–466.

[9] J. Abonyi, A. Bodizs, L. Nagy, F. Szeifert, Hybrid fuzzy convolution model and its application in predictive control, Chemical Engineering Research and Design 78 (2000) 597–604.

[10] J. Mao, K. Jain, Artificial neural networks for feature extraction and multivariate data projection, IEEE Trans. on Neural Networks 6(2) (1995) 296–317.

[11] S. Abe, R. Thawonmas, Y. Kobayashi, Feature selection by analyzing regions approximated by ellipsoids, IEEE Trans. on Systems, Man, and Cybernetics, Part. C 28(2) (1998) 282–287.

[12] C. A. I. Tsamardinos, A. Statnikov, Time and sample efficient discovery of markov blankets and direct causal relations, in: 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, pages 673-678, USA, 2003.

[13] A. A. S. Chebrolu, J. Thomas, Feature deduction and ensemble design of intrusion detection systems, Computers and Security <http://dx.doi.org/10.1016/j.cose.2004.09.008> .

[14] J. D. Y. Chen, B. Yang, A. Abraham, Time series forecasting using flexible neural tree model, Information Sciences <http://dx.doi.org/10.1016/j.ins.2004.10.005> .

[15] J. V. de Oliveira, Semantic constraints for membership function optimization, IEEE Trans. FS 19 (1999) 128–138.

[16] M. Setnes, R. Babuška, Fuzzy relational classifier trained by fuzzy clustering, IEEE Trans. on Systems, Man, and Cybernetics, Part. B 29 (1999) 619–625.

[17] R. Setiono, Generating concise and accurate classification rules for breast cancer diagnosis, Artificial Intelligence in Medicine 18 (2000) 205–219.

[18] A. Abraham, Meta-learning evolutionary artificial neural networks, Neurocomputing, Elsevier Science, Netherlands, Vol. 56c, pp. 1-38 .

[19] D. Nauck, R. Kruse, Obtaining interpretable fuzzy classification rules from medical data, Artificial Intelligence in Medicine 16 (1999) 149–169.

[20] H. Ishibuchi, T. Nakashima, T. Murata, Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems, IEEE Trans. SMC–B 29 (1999) 601–618.

[21] A. Abraham, Evonf: A framework for optimization of fuzzy inference systems using neural network learning and evolutionary computation, in: The 17th IEEE International Symposium on Intelligent Control, ISIC'02, IEEE Press, ISBN 0780376218, pp. 327-332, Canada, 2002.

[22] M. Setnes, J. Roubos, Rule-based modeling: Precision and transparency, IEEE Trans. FS. in press.

[23] A. Abraham, i-miner: A web usage mining framework using hierarchical intelligent systems, in: The IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'03, IEEE Press, ISBN 0780378113, pp. 1129-1134, USA, 2003.

[24] L. Sethi., Entropy nets: From decision trees to neural networks, Proc. IEEE 78 (1990) 1605–1613.

[25] I. Ivanova, M. Kubat, Initialization of neural networks by means of decision trees, Knowledge-Based Systems 8 (1995) 333–344.

[26] R. Setiono, W. Leow, On mapping decision trees and neural networks, Knowledge Based Systems 13 (1999) 95–99.

[27] J.-S. Jang, C.-T. Sun, Functional equivalence between radial basis function networks and fuzzy inference systems, IEEE Trans. NN 4 (1993) 156–159.

[28] L. T. Kóczy, D. Tikk, T. D. Gedeon, On functional equivalence of certain fuzzy controllers and rbf type approximation schemes, International Journal of Fuzzy Systems .

[29] T. Kohonen, The self-organizing map, Proceedings of the IEEE 78(9) (1990) 1464–1480.

[30] J. Abonyi, F. Szeifert, Supervised fuzzy clustering for the identification of fuzzy classifiers, Pattern Recognition Letters 24(14) (2003) 2195–2207.

[31] M. Setnes, R. Babuška, U. Kaymak, H. van Nauta Lemke, Similarity measures in fuzzy rule base simplification, IEEE Trans. SMC-B 28 (1998) 376–386.

[32] M. Setnes, J. Roubos, Transparent fuzzy modeling using fuzzy clustering and GA's, in: In NAFIPS, New York, USA, 1999, pp. 198–202.

[33] J. Roubos, M. Setnes, Compact fuzzy models through complexity reduction and evolutionary optimization, in: Proc. of IEEE international conference on fuzzy systems, San Antonio, USA, 2000, pp. 762–767.

[34] Y. Yam, P. Baranyi, C. Yang, Reduction of fuzzy rule base via singular value decomposition, IEEE Trans. Fuzzy Systems 7(2) (1999) 120–132.

[35] J. Yen, L. Wang., Simplifying fuzzy rule-based models using orthogonal transformation methods, IEEE Trans. SMC-B 29 (1999) 13–24.

[36] M. Setnes, R. Babuška, Rule base reduction: Some comments on the use of orthogonal transforms, IEEE Trans. on Systems, Man, and Cybernetics, Part. B 31 (2001) 199–206.

[37] K. Cios, W. Pedrycz, R. Swiniarski, Data Mining Methods for Knowledge Discovery, Kluwer Academic Press, Boston, 1998.

[38] J. Roubos, M. Setnes, J. Abonyi, Learning fuzzy classification rules from labeled data, International Journal of Information Sciences 150(1-2) (2003) 612–621.

[39] Y. Jin, Fuzzy modeling of high-dimensional systems, IEEE Trans. FS 8 (2000) 212–221.

[40] J. Abonyi, R. Babuška, F. Szeifert, Fuzzy modeling with multidimensional membership functions: Grey-box identification and control design, IEEE Trans. on Systems, Man, and Cybernetics, Part. B (2002) 612–621.

[41] J. Abonyi, H. Roubos, F. Szeifert, Data-driven generation of compact, accurate, and linguistically sound fuzzy classifiers based on a decision tree initialization, International Journal of Approximate Reasoning Jan (2003) 1–21.

[42] F. Hoppner, F. Klawonn, R. Kruse, T. Runkler, Fuzzy Cluster Analysis - Methods for Classification, Data Analysis and Image Recognition, John Wiley and Sons, 1999.

[43] O. Nelles, M. Fischer, Local linear model trees (LOLIMOT) for nonlinear system identification of a cooling blast, in: European Congress on Intelligent Techniques and Soft Computing (EUFIT), Aachen, Germany, 1996.

[44] J.-S. Jang, Structure determination in fuzzy modeling: A fuzzy cart approach, in: Proc. of IEEE international conference on fuzzy systems, Orlando, USA, 1994.

[45] A. Corcoran, S. Sen, Using real-valued genetic algorithms to evolve rule sets for classification, in: IEEE-CEC, Orlando, USA, 1994, pp. 120–124.

Table 1: Three rule fuzzy classifier (L=low, M=medium , H=high).

	1 Alc	2 Mal	3 Ash	4 aAsh	5 Mag	6 Tot	7 Fla	8 nFlav	9 Pro	10 Col	11 Hue	12 OD2	13 Pro	Class
R_1	H	-	-	-	-	-	H	-	-	M	L	-	L	1
R_2	L	-	-	-	-	-	-	-	-	L	L	-	H	2
R_3	H	-	-	-	-	-	L	-	-	H	H	-	H	3

Table 2: Classification rates on the Wine data for ten independent runs.

Method	Best result	Aver result	Worst result	Rules	Model eval
Corcoran and Sen [45]	100%	99.5%	98.3%	60	150000
Ishibuchi et al. [20]	99.4%	98.5%	97.8%	60	6000
Cluster + GA	99.4 %	varying schemes	98.3%	3	4000-8000
Gath-Geva clustering	95.5 %	95.5 %	95.5 %	3	1
Sup. cluster (13 features)	98.9 %	98.9 %	98.9 %	3	1
Sup. cluster (5 features)	100 %	100 %	100 %	5	2