Fuzzy Rule Extraction for Determining Creditworthiness of Credit Applicants

by

Steve G. Romaniuk

Department of Information Systems and Computer Science

National University of Singapore

10 Kent Ridge

Singapore, 0511

AI topic: Fuzzy Logic, Rule Extraction, Machine Learning, Hybrid Symbolic/Connectionist Representation.

Domain area: Creditworthiness of credit seeking applicants.

Language/Tool: C-language, SC-net tool.

Status: Prototypical.

Abstract

The main objective of this research paper is to provide an empirical analysis of the hybrid symbolic/connectionist expert system development tool SC-net to act as a viable system for acquiring expert system knowledge by means of learning. The task to be studied is the prediction of creditworthiness for credit seeking applicants. The creditworthiness domain - unlike many other domains studied by the machine learning community - contains both uncertainties in the inputs and outputs. Apart from showing SC-net's ability to derive human acceptable models for this data, strong emphasis is placed on deriving rules that can adequately describe the imprecision inherent in such domains. No a priori domain knowledge, such as pre-defined fuzzy membership functions or pre-selection of important input features is required. The affect of training set size on number of rules and attributes per rule is addressed and a sample set of extracted rules with derived membership functions is provided. In all cases acceptable models for determining creditworthiness are derived. The herein described experimental results should further strengthen SC-net's ability to act as a knowledge acquisition tool for obtaining acceptable expert knowledge in uncertain domains.

1 Introduction

1.1 Knowledge Acquisition for Fuzzy Domains

In light of the ever increasing popularity of expert system being applied to the human decision making process - which can be harnessed by industries and businesses -, the problem of acquiring the vast knowledge required for these systems has materialized in form of increased interview sessions between knowledge engineers and human domain experts. Attempts to curb this excessive requirement in man hours has lead researchers to develop methods for easing the knowledge acquisition process. Interviewing methods based on repetory grids and automated tools for simplification and graphical representation of knowledge have been added to the knowledge engineer's toolbox. Nonetheless, some of the basic problems which directly relate to the interaction of the knowledge engineer with one or more human domain experts have remained. Communication, lack of knowledge and sometimes subconscious fear (computers intrusion and possible domination into an area of human expertise) have not disappeared and prompted the development of methods to further cut down on the interaction of the two. Connectionist systems and their ability to learn have made it possible to incorporate parts of the knowledge acquisition phase into an expert system development tool [4] [5]. Explaining and expanding such a system to allow for direct incorporation of knowledge (representational form is rules), and supporting knowledge extraction after learning, to allow for knowledge refinement makes it possible for domain experts to directly prototype expert systems, or under other circumstances directly support the knowledge engineer in obtaining information from raw data, which may then be presented to the domain expert for further refinement [1]. Besides reducing the reliance of the knowledge engineer on the domain expert to develop a knowledge base, two additional motivations can be identified for automating the knowledge acquisition phase: First, it allows the synthesis of new knowledge from existing knowledge through learning. Second, it provides a dynamic environment that allows modifying knowledge whenever it becomes necessary, and thus further supports the knowledge refinement cycle. Applying machine learning to the task of knowledge acquisition has been suggested for quite some time now [2] [3], making it applicable for domains containing uncertainty is just getting started. Learning the salience of input features - training examples are composed of,- is essential to develop compact and meaningful fuzzy rules. The research described in this paper will address the above stated problems by examining the applicability of hybrid learning systems to the problem of determining creditworthiness of credit seeking applicants.

1.2 Overview of SC-net

SC-net has been developed as a tool to support the knowledge engineer in the difficult task of knowledge acquisition for expert systems. Machine learning and knowledge representation in a hybrid/symbolic connectionist environment, together with uncertainty management through means of fuzzy logic, form the corner stones of the system. Learning is analogous to instance-based learners in that only a single pass through the training data is required and examples are encoded through Recruitment of Cells Algorithm (RCA). During RCA pass an instance is either identified (Difference in actual and expected output of network within ϵ), the bias of one or more cells - representing encoded instances - is modified (Difference within 5ϵ), or a new cell is recruited to encode the presented instance. For more detail on the RCA algorithm see [3] [5]. After RCA generates a network from a given training set, the network can be pruned (optional). Pruning is achieved through use of the Global Attribute Covering Algorithm (GAC). It is SC-net's foremost means to learn the importance of input features and can result in simple but powerful rules for describing learned knowledge. Due to space constraints the interested reader is referred to [3] for an in depth description of GAC and its properties.

The choice of a connectionist architecture was motivated by the following three desirable features:

1. A highly parallel and uniform representation of knowledge (the SC-net network).

2. Fault tolerance and noise resistance.

3. A built-in ability to deal with non-crisp inputs and outputs.

On the other hand it proved beneficial to incorporate some of the strong points of symbolic machine learning. From the symbolic side we can identify:

1. The ability to encode rules to support knowledge refinement.

2. Allow for rule extraction as a direct means to elicit learned knowledge and support the implementation of expert system standards such as, consulation and explanation facilities.

3. Provide a means to represent symbolic constructs such as variables, comparators and quantifiers. This leads to a more powerful language for describing knowledge and augmenting the learning process through use of domain specific meta knowledge.

The quintessence of SC-net is to combine the virtues of both symbolic and connectionist representations, that is to yield a more powerful environment for constructing expert systems by providing means to overcome the knowledge acquisition bottleneck.

2 Fuzzy Variables and their Adaption through Learning

SC-net supports the representation of fuzzy variables, which allow either the user or the system itself to divide the numerical range of a variable into its fuzzy equivalent. In general fuzzy variables are described by a set of membership functions, where each function is associated with a linguistic hedge such as *high*, *small*, etc. These membership functions correlate a given numerical value with a degree of membership indicating the strength (membership) of the numerical value being a member of the predefined fuzzy sets. In SC-net only pi-shaped membership functions are supported. An extension to more complicated membership functions is forthcoming [7]. A linguistic hedge in SC-net is defined by the 4 quantities:

$$< HedgeId >: Bound_{Lower} \ldots Bound_{Upper}(Plateau_{Lower}, Plateau_{Upper}) \tag{1}$$

Whenever the value of a given fuzzy variable lies between $Bound_{Lower}$ and $Bound_{upper}$ $< HedgeId >$ takes on a membership value of 1. If the value falls outside the $Plateau_{upper}$ and $Plateau_{lower}$ range a membership value of 0 is assigned. In every other case a graded membership response is returned which in turn is described by a linear function (arms of pi-shaped membership function). Figure 1 shows the general network structure used by SC-net to represent a fuzzy variable (labeled attribute) and a linguistic hedge (labeled attribute[value]). Cells labeled with the numerical value of -1 return the minimum of the incoming activations, whereas cells with a 0 label return the strong negation $(1-Activation)$ of the incoming activation.

The weights are calculated as follows:

$$\begin{aligned}
&Weight-1 = \tfrac{1}{Plateau_{Upper}}\\
&Weight-2 = 1\\
&Weight-3 = \tfrac{Plateau_{Upper}}{Plateau_{Upper}-Bound_{Upper}}\\
&Weight-4 = \tfrac{1}{1-Plateau_{Lower}}\\
&Weight-5 = \tfrac{1-Plateau_{Lower}}{Bound_{Lower}-Plateau_{Lower}}
\end{aligned} \tag{2}$$

Finally, SC-net allows the arms of fuzzy membership functions to be dynamically adapted through use of the Dynamic Plateau Modification Algorithm (DPM for short) [3]. Initially $Plateau_{Lower}$ and $Plateau_{Upper}$ are set to the smallest and the largest variable range value, respectively. By presenting encoded instances of examples learned by RCA and represented into a network structure (the SC-net network) the membership arms of the pi-shaped functions are modified. The central idea of the algorithm is to place constraints on the degree of generalization provided by each of the arms. If the degree of membership

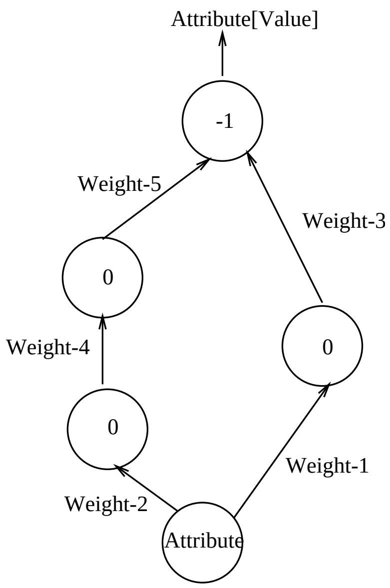


Figure 1: SC-net Network Representing Fuzzy Attribute

calculated by either arm is too high (for a given encoded example), it is lowered by appropriately moving either the $Plateau_{Lower}$ value closer to the $Bound_{Lower}$ or the $Plateau_{Upper}$ value closer to $Bound_{Upper}$. The amount of adjustment is determined by comparing the actual and the expected output response of a cell. For further detail on fuzzy variables, activation functions used by SC-net and the Dynamic Plateau Modification algorithm refer to [3, 5].

3 Experiments

The purpose of the experimental section is to test SC-net's ability to deal with fuzzy domains (uncertainty in inputs and outputs) both during learning and testing. Two questions are posed and answered:

1. Can SC-net adequately learn the prediction of creditworthiness for applicants, that is, derive an acceptable model and if so, how many training patterns are required?

2. Keeping the first question in mind, can representative but simple fuzzy rules be extracted, - can salient features be identified for forming concept descriptions?

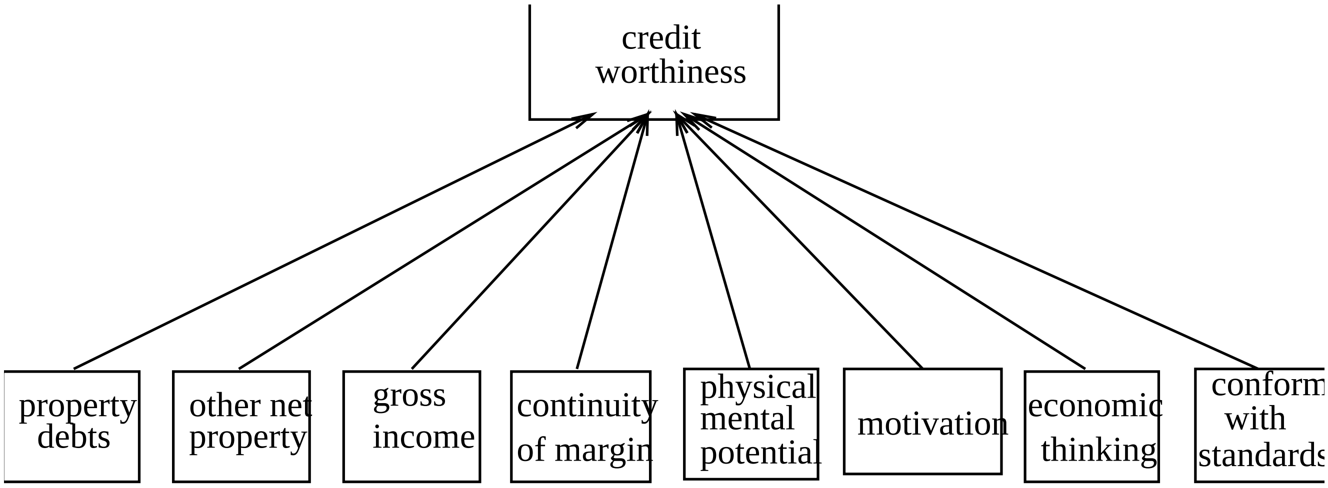


Figure 2: Description of Creditworthiness Domain

3.1 Domain of Study

In this section we provide a brief introduction of the credit domain. The main objective is to identify the creditworthiness (its degree) of credit seeking applicants by means of learning individual records obtained from [8]. There are a total of 50 credit records where each applicants record is described by 8 fuzzy features (continuous in range [0,1]). The output is a membership value reflecting the degree of creditworthiness of a person. The relation of the input features to the output are described in Figure 2. Note, that the original problem was described [9] by a hierarchical dependency network of 3 layers. In order to study SC-net's ability to learn the salience of certain input features, it was deemed necessary to slightly recast the original problem description by removing the 2 intermittent layers and directly mapping the input features to the output creditworthiness, as shown in Figure 2. Furthermore, the original data was rounded from 4 digit to 2 digit precision after the decimal point. Table 1 contains a few sample records of the original 50 credit examples. The first record represents an individual with a high credit rating, whereas the decision for the second is unclear and the third receives an overall bad rating. Note, that the order of input features is identical to the one in Figure 1.

Table 1. Sample creditworthiness records for different train set partitions

	PD	ONP	GI	COM	PMP	M	ET	CWS	CW
Record 1	0.85	0.53	0.67	0.68	0.66	0.66	0.57	0.80	0.83
Record 2	0.00	0.54	0.47	0.50	0.57	0.56	0.53	0.63	0.58
Record 3	0.01	0.16	0.51	0.43	0.47	0.40	0.17	0.32	0.23

3.2 Description of Experiments

Prior to presenting the experimental results, we briefly outline the measure used to determine the quality of SC-net's performance. In all the forthcoming experiments the average prognostic error (s_p^2) between the actual and expected output of the 50 credit examples was calculated. The error is given by

$$s_p^2 = \frac{\sum_{i=1}^{N}(\mu_{Oi} - \mu_{Ti})}{2(N-1)} \tag{3}$$

where, μ_{Oi} is the actual response obtained for example i and μ_{Ti} is the expected response for the same example. In order to evaluate the overall performance of the learned models, the intolerable error (s_e^2) is set to 0.005. This particular assignment was chosen in accordance with the error threshold used in [9]. Also in [9] the intolerable error is used as a criterion of acceptance or rejection of a model. It is acceptable if $s_p^2 < s_e^2$, otherwise it is rejected. Furthermore, it is beneficial to define a model of the goodness of the ratio of variance.

$$F_{comp} = \frac{s_e^2}{s_p^2} \tag{4}$$

We now turn our attention to the second experiment, conducted to test SC-net's ability to perform on the problem domain of determining creditworthiness of credit seeking applicants. In the first experiment we attempt to determine the quality of the models generated for various randomly chosen variable sized training sets. For this purpose 10 random train partitions were generated from the original 50 examples with partition sizes of 40, 30, 20, and 10 examples. Table 2 displays the obtained average prognostic error and corresponding F_{comp} for $\epsilon = 0.0001$. As the results clearly point out, SC-net is capable of deriving an acceptable model in all cases, even when the size of the train set is as small as 10 examples. As expected, as the number of training patterns is reduced the overall performance decreases, indicating that the complete set of examples contains many unique patterns. This is further substantiated by observing the drastic drop in F_{comp} when the training set size is reduced from 50 to 40 patterns. Removal of 20% of the patterns has a significant impact on the quality of the derived creditworthiness models. Nonetheless, all models are acceptable. The findings of these results corresponds to those reported in [6] were FUZZNET - a pre-cursor to SC-net - was applied to a similar task. In all cases F_{comp} is decisively higher for SC-net over FUZZNET.

Table 2. Average Prognostic Error and Variance Ratio for various test partitions

	50 Ex.	40 Ex.	30 Ex.	20 Ex.	10 Ex.
s_p^2	0.000082	0.0007	0.0016	0.0028	0.0039
F_{comp}	61.0	7.1	3.1	1.8	1.3

Next, we draw our attention to the second experiment. Using the same random partitions of the first experiment, we additionally apply network pruning to learn the important input features, which allows us to derive simple compact rules. To measure the overall quality we determine the maximum, minimum and average number of rules extracted after pruning and the average number of fuzzy attributes required within the extracted rules. Table 3 shows the results obtained by SC-net with pruning for $\epsilon = 0.01$. We again note, that all the generated models are acceptable with regard to the intolerable error. In other words, the generated rules for the various variable sized train sets are all acceptable. The average number of rules generated drops quite generously as the train set size decreases, as does the average number of attributes. Using train sets of size 10 generates on the average the smallest rule set with the least number of attributes in the rules premise.

Table 3. Performance Statistics for SC-net ($\epsilon = 0.01$) for different train set partitions

	40 Ex.	30 Ex.	20 Ex.	10 Ex.
s_p^2	0.0016	0.0021	0.0031	0.0037
Max #Rules	24	20	14	9
Min #Rules	15	9	9	4
Avg. #Rules	20.2	17.7	11.2	5.7
Avg. #Attributes	2.3	2.1	2.1	1.4

It is expected that once the value of ϵ is modified results will change, and on the average a similar number of rules with similar sized premises will be obtained for different partition sizes. This will rectify the results shown in the previous table, which may lead one to conclude rule quality and quantity is a function of training set size only. Table 4 shows the same quality measures for the generated rules as the previous table did, except that ϵ was adapted for different partition sizes. As expected the results found in Table 3 did not carry over. Overall the rule sets derived are less than 5 for various training set sizes. Also, the average number of attributes in the premise of the extracted rules is about the same. These results clearly show that by appropriately modifying the ϵ learning parameter very small and compact rule sets can be learned regardless of the training set size. This is an important finding for knowledge engineers, since it allows them to inspect and judge what an SC-net network actually has learned for domains containing uncertainty.

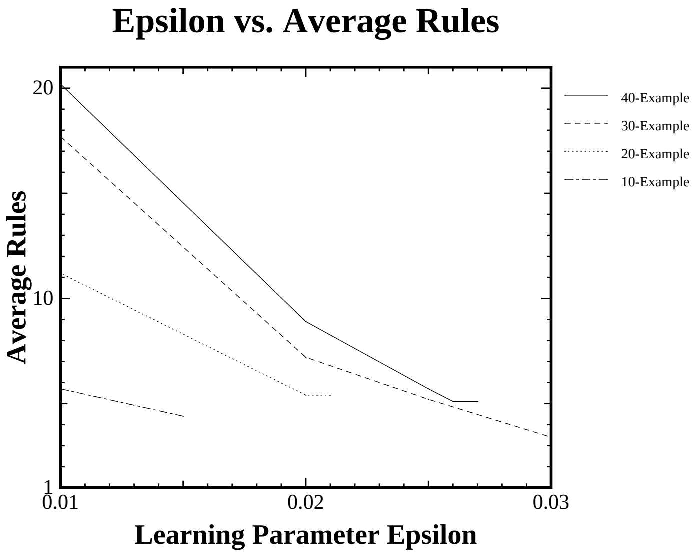
Table 4. Performance Statistics for SC-net for different train set partitions

	40 Ex.	30 Ex.	20 Ex.	10 Ex.
ϵ	0.027	0.03	0.02	0.015
s_p^2	0.0044	0.0049	0.0037	0.005
Max #Rules	9	7	10	7
Min #Rules	3	2	1	2
Avg. #Rules	5.1	3.4	5.4	4.4
Avg. #Attributes	1.3	1.2	1.5	1.4

In Figure 3 we shed light on the impact the learning parameter ϵ has on the number of rules generated and the number of attributes in the premise of these rules. We can make the following observations: As the number of training examples is reduced the initial size of the rule sets generated by SC-net also decreases. Reduction in rule set size (for fixed size train sets) is smooth and almost linear as the learning parameter ϵ is relaxed. Recall, ϵ determines during the RCA phase the degree by which cells are recruited to represent new instances. As the parameter is decreased, the number of instances stored in the network increases, whereas when the parameter is increased more of the existing network cells biases are adjusted. Finally, turn around points for the parameter ϵ can be identified at the end points of the represented functions. Increasing ϵ beyond these points results in a decrease of the rules performance. In fact, the derived rule sets do not represent acceptable models and are therefore not shown.

3.3 Extracted Fuzzy Rules

To provide a flavor of the types of rules generated by SC-net one complete rule set derived for the 30 example train set for $epsilon = 0.025$ is displayed in Table 5.



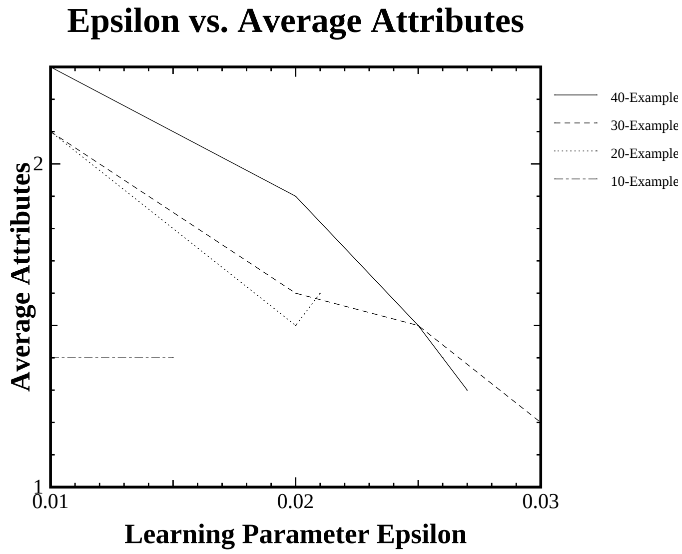


Figure 3: Learning Parameter Epsilon vs. (a) Average #Rules (b) Average #Attributes

Table 5. Fuzzy Rules Extracted by SC-net for Creditworthiness

Rule 1:

if and(fuzzy(gross_income[p2]) = 1.0, fuzzy(property_debts[p1]) = 1.0) then credit_worthiness (0.887).

Rule 2:

if (fuzzy(economic_thinking[p1]) = 1.0) then credit_worthiness (0.867).

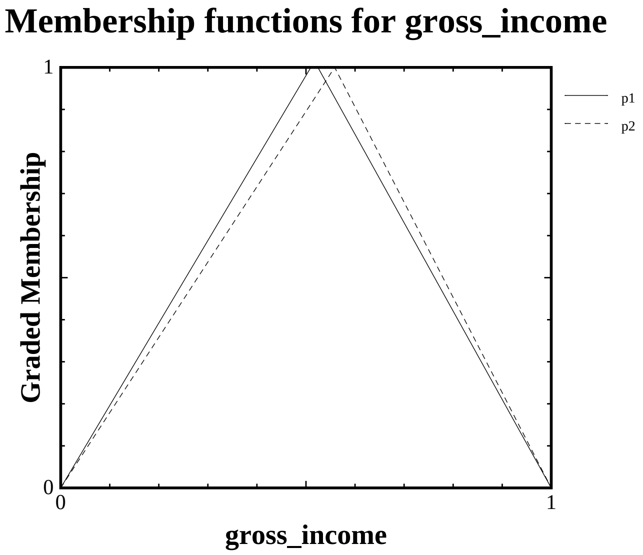
Rule 3:

if and(fuzzy(gross_income[p1]) = 1.0, fuzzy(property_debts[p1]) = 1.0) then credit_worthiness (0.827).

Rule 4:

if (fuzzy(gross_income[p1]) = 1.0) then credit_worthiness (0.753).

In order to provide a better understanding of the semantic content of these rules Figures 4-6 show the initial and DPM generated membership functions for the fuzzy inputs gross_income, property_debts, and economic_thinking, respectively. For features gross_income and economic_thinking we can identify a strong adaption of the left arms of the membership functions, whereas the right arms and feature property_debts are left untouched. The fact that only a total of 3 input features were selected (of the original 8) and 2 of these features are described by a single membership function (2 membership functions for gross_income) further substantiates the fact that SC-net is capable of deriving simple and compact rules which display overall good performance for the creditworthiness domain. Simplicity of rules allows for quicker and better understanding of the rules semantic content by the knowledge engineer. Paired with the systems ability to allow pre-encoding of rules prior to training supports knowledge refinement capability through human intervention. Furthermore, simpler rules in terms of the size of their premise and the number of membership functions required to model input features provides the opportunity for high speed implementations in the event that neural hardware to emulate SC-net's network behavior is unavailable.



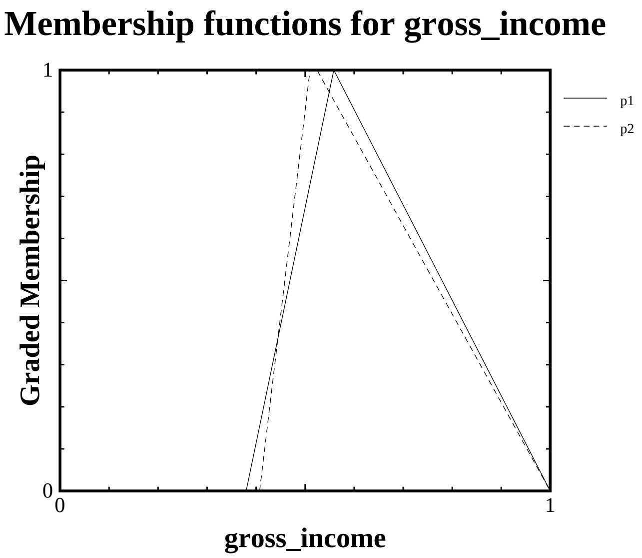
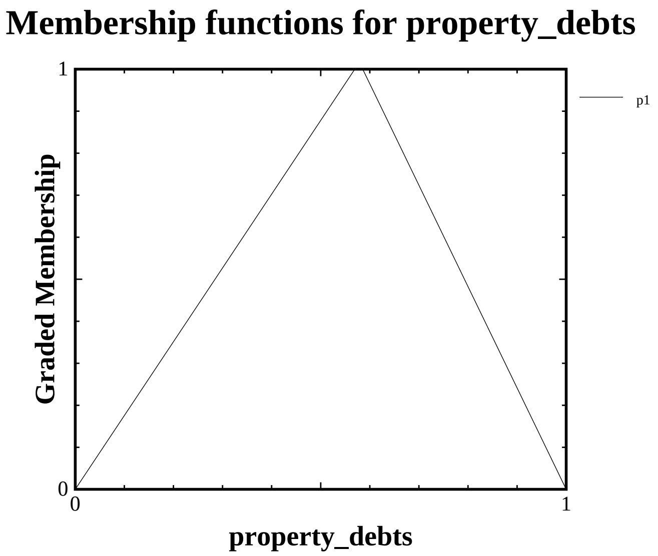


Figure 4: Membership Functions of Attribute gross_income (a) before DPM (b) after DPM



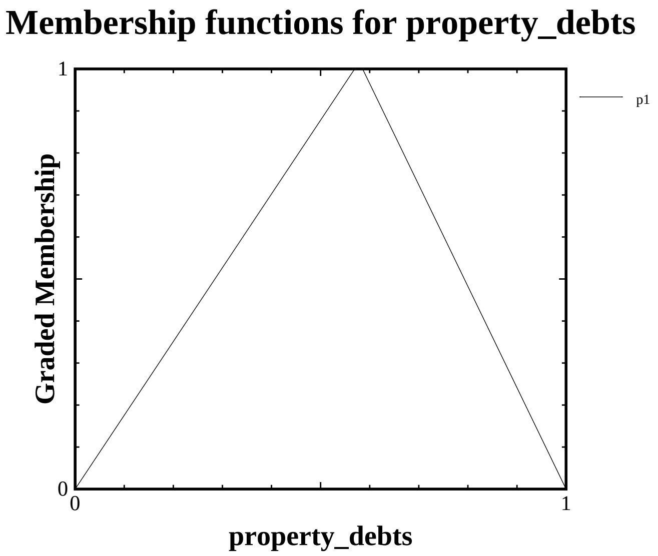
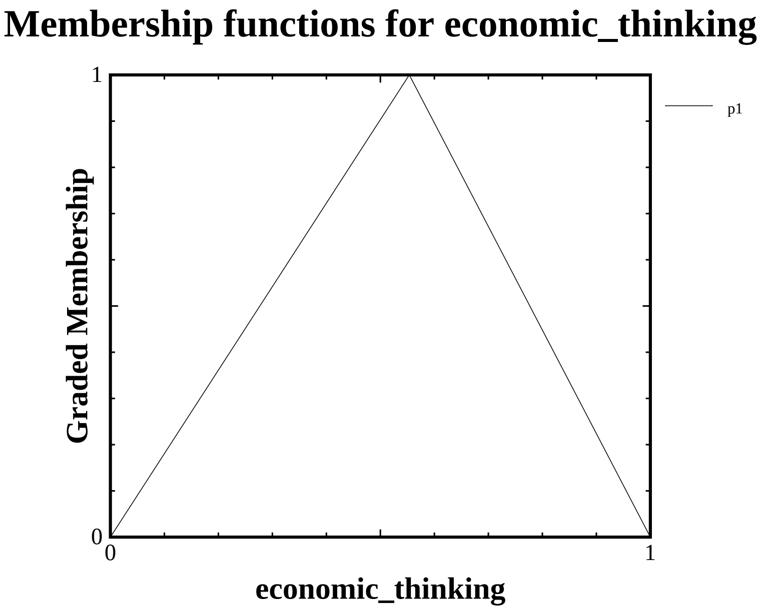


Figure 5: Membership Functions of Attribute property_debts (a) before DPM (b) after DPM



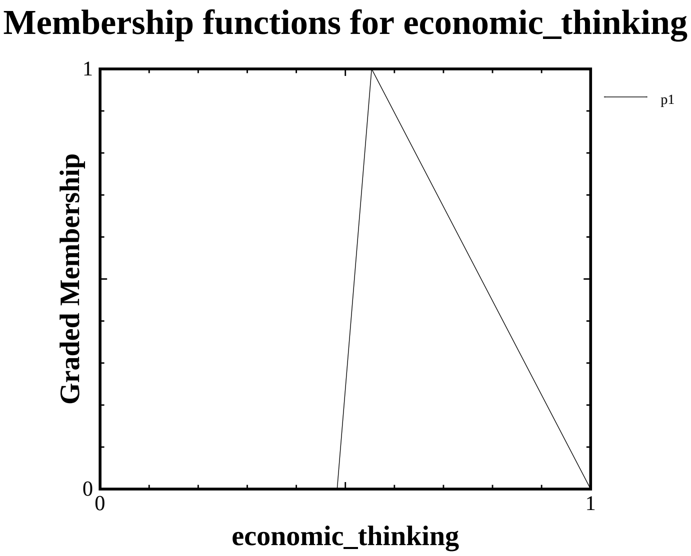


Figure 6: Membership Functions of Attribute economic_thinking (a) before DPM (b) after DPM

4 Summary

This paper provided a series of experiments targeted to investigate the salience of the prototypical hybrid symbolic/connectionist expert system development tool SC-net to extract expert like knowledge for a real world domain concerned with analyzing creditworthiness of credit seeking applicants. Emphasis was placed both on deriving human acceptable models of decision making in conjunction with deriving compact and readily comprehensible sets of rules that could be used in expert system like environments. The effectiveness of SC-net to derive acceptable models of creditworthiness across variable sized training sets was underlined, and its dependence on the learning parameter ϵ was pointed out. A sample set of extracted rules together with derived fuzzy membership functions for input features deemed important was provided. In light of the positive results obtained, it would seem justified to conclude the viability of knowledge acquisition for fuzzy domains by means of machine learning and warrant to continue investigating SC-net's applicability to the domain of creditworthiness and possibly other domains containing uncertainty.

References

[1] Perez, R.A., Hall, L.O., Romaniuk, S.G., Lilkendey, J,T. (1992), Inductive Learning For Expert Systems In Manufacturing, HICCS-25, Hawaii International Conference on Systems Sciences, Koloa, Hawaii, January.

[2] Quinlan, J. R. 1979, Discovering rules by induction from large collections of examples, In D. Mitchie (Ed.), Expert systems in the micro electronic age. Edingurgh University Press.

[3] Romaniuk, S.G. (1991) Extracting Knowledge from a Hybrid Symbolic, Connectionist Network, PhD Dissertation, University of South Florida.

[4] Romaniuk, S. G., Hall, L. O. (1990) FUZZNET: Towards a Fuzzy Connectionist Expert System Development Tool, In Proceedings of IJCNN, Washington D.C., January.

[5] Romaniuk, S.G., Hall, L.O. (1993) SC-net: A Hybrid Connectionist, Symbolic Network, To appear: Information Sciences.

[6] Romaniuk, S.G., Hall, L.O. (1992) Decision making on creditworthiness, using a fuzzy connectionist model, Fuzzy Sets and Systems, 48, pp. 15-22.

[7] Romaniuk, S.G., (1993) Representing Complex Fuzzy Membership Functions in a Connectionist network, International Fuzzy Systems and Intelligent Control Conference, Kentucky.

[8] Zimmermann, H.-J., Zysno, Thole, U., P. (1980) Keditwuerdigkeit - Empirische Ueberpruefung einer im Rahmen der Theorie der unscharfen Mebngen modellierten Hierachie gewichteter Kriterien, Technical report, University of Aachen.

[9] Zimmermann, H.-J., Zysno, P. (1983) Decisions and evluation by hierarchical aggregation of information, Fuzzy Sets and Systems, 10, pp. 243-260.