Belief-Evidence Fusion in a Hybrid Intelligent System

Nelson Marcos
De La Salle University
2401 Taft Avenue
Manila
Philippines
marcosn@dlsu.edu.ph

Arnulfo Azcarraga
De La Salle University
2401 Taft Avenue
Manila
Philippines
azcarragaa@dlsu.edu.ph

Abstract - *A hybrid intelligent system that is able to successively refine knowledge stored in its rulebase is developed. The existing knowledge (referred to as belief rules), which may initially be defined by experts in a particular domain, is stored in the form of rules in the rulebase and is refined by comparing it with new knowledge (referred to as evidence rules) extracted from data sets trained under a neural network. Based on measurement, assessment, and interpretation of rule similarity, belief rules existing in the rulebase may be found to be confirmed, contradicted, or left unsupported by new training data. New evidence rules may also be discovered from the training data set. This rule comparison is unique in the sense that rules are viewed and compared in a geometric manner. As rules evolve in existence in the rulebase during the belief-evidence fusion process, their bounds, strengths, and certainties are also revised. The hybrid intelligent system is tested with different data sets, including hypothetical data sets and actual data sets.*

Keywords: Successive rule refinement, belief-evidence fusion, hybrid intelligent system, rule extraction.

1 Introduction

The domain knowledge captured as a set of "beliefs" in a *rulebase* system (RBS) may not be perfect. The domain knowledge may be vague, imprecise, incorrect, or uncertain. There may be theories and heuristics which were not articulated well by experts, or there may be knowledge that were previously unknown to experts [1]. Real data, which serve as "evidence", can be used to refine and complete the belief structure of the system. Studies in data fusion [2-7] and evidence theory [8-10] are useful in this type of problem.

The set of belief rules, encoded in disjunctive normal form (DNF), are either confirmed, modified, challenged, or left unsupported by the "evidence" available. Certain new evidences that do not figure in any existing belief can be assimilated as new beliefs.

The fusion of belief and evidence is done through *successive rule refinement* in a *hybrid intelligent system* [11-16]. The available evidences in the form of raw data are converted into rule form so that they can be integrated with the existing beliefs about the domain in the RBS. Converting evidence into rule form is done through a *rule extraction* system that trains a *neural network* (NN) using the available evidence and extracts rules from the network once it has been sufficiently trained.

As illustrated in Fig. 1, the RBS in the hybrid intelligent system captures an existing belief structure. The RBS is enhanced by a rule refinement system that combines the belief structure in the RBS with evidence found and converted into rule form by an NN rule extraction system.

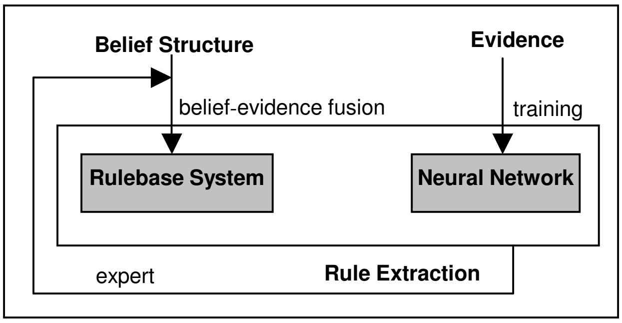

Fig. 1. Hybrid Intelligent System Model

In this work, we use the *Discretized Interpretable Multi Layer Perceptron* (DIMLP) [17-20] for training with actual data. DIMLP extracts rules with 100% fidelity (i.e., extracted rules exactly represent network responses). The key idea behind rule extraction in DIMLP is the precise localization of *discriminant frontiers* or *hyperplanes*, which are precisely determined by *weight values* connecting input and hidden neurons. The rule extraction algorithm checks whether a possible hyperplane frontier is effective or not in a given region of the input space. The algorithm uses decision trees with pruning. The algorithm is given in [17-20] as:

a. Determine discriminant hyper-planes using available examples and DIMLP classifications.
b. Build a decision tree according to the highest potential frontier criterion.
c. Prune rule antecedents according to the most enlarging pruned antecedent criterion.
d. Prune unnecessary rules, if any.
e. Modify antecedent thresholds according to the most enlarging criterion.
f. Go to step 4 if some antecedents have been modified in step 5.
g. Determine a default rule according to the maximum number of replaced antecedents.

During the fusion of RBS and NN rules, various changes to the belief rules are explored and evaluated with the help of the *expert*, although, in the experiments discussed in this paper, all fusion decisions are made automatically by the system. This enhancement of the belief structure is done in a successive manner as more and more evidence are submitted for *rule extraction* and *fusion* to calibrate the performance of the whole system.

From the experiments conducted to demonstrate the applicability of the approach, it can be seen that the system's set of beliefs becomes more and more refined and complete as increasing units of evidences are integrated in it.

2 Belief and Evidence Rules

Rules in the belief rule set $\mathfrak{R}_b$, and evidence rule set $\mathfrak{R}_e$ derived by the rule extraction system, are transformed into DNF defined by the nomenclature in Table 1.

Table 1: Nomenclature for Rules.

Nomenclature	Definition
r_i	Rule in the form: **If** (x_1 **and** x_2 **and** … x_k **and** … x_K) **Then** $v_o = c_t$
x_k	kth condition in the antecedent in the form $O_k(v_k, z_k)$
$O_k \in \{ \geq, \leq \}$	Relational operator
$v_k \in V$	Variable in the condition
$V = \{ v_p \mid 0 \leq v_p \leq 1 \}$	Set of input variables (or input attributes) in the application domain
z_k, $0 \leq z_k \leq 1$	Operand in the condition
v_o	Output variable
$c_t \in C$	Category for the output variable (or output class or category)

Examples of rules in this notation are:

- **If** $(X \geq 0.01)$ **and** $(X \leq 0.24)$ **and** $(Y \geq 0.01)$ **and** $(Y \leq 0.24)$
  Then Class = Black
- **If** (SepLen >= 0.09523812) **and** (SepLen <= 0.40476188) **and** (SepWid >= 0.19999999) **and** (SepWid <= 0.9000001) **and** (PetLen >= 0.046153847) **and** (PetLen <= 0.18461539) **and** (PetWid >= 0.035714287) **and** (PetWid <= 0.21428573) **and**
  Then Class = Setosa

The bounding values for each input variable in a rule are normalized between 0 and 1. These bounds can be viewed to specify a hyperspace occupied by a rule in a hyperplane.

2.1 Strength of Evidence

The *strength* of evidence of a rule is a measure of the quality and quantity of supporting data. A value ranging from 0 to 1, strength is a function of the *confidence* and *support* as shown in Eq. (1).

$$strength = \mathrm{f}(confidence, support) \qquad (1)$$

In the implementation of the system, strength of evidence is computed as the geometric mean of *confidence* and *support*.

The *confidence* of a rule **If** *X* **Then** *Y*, for a given data set *D*, is defined in data mining as the number of data instances in *D* that satisfy the premise *X* and has a conclusion *Y* over the number of data instances in *D* satisfying premise *X* [21-25]. This shows how often is the consequent true given that the premise is true. Statistically, confidence is defined as:

$$conf(X \rightarrow Y) = \mathrm{P}(X \cap Y) \,/\, \mathrm{P}(X) \qquad (2)$$

The *support* of a rule **If** *X* **Then** *Y*, for a given data set *D*, is defined in data mining as the number of data instances in *D* that satisfy the premise *X* and has a conclusion *Y* over the number of data instances in *D* [21-25]. This shows how often are both the premise and consequent true given the data set *D*. Statistically, support is defined as:

$$sup(X \rightarrow Y) = \mathrm{P}(X \cap Y) \qquad (3)$$

Confidence, support, and strength of evidence, each ranging from 0 to 1, are attached to each output class for each rule. The properties are attached to each of the categories because this allows observation of how each category plays in the hyperspace covered by a given rule. This gives a measure of how strong each category is (which we refer to as strength of evidence) in that hyperspace, a knowledge of which category is the strongest, and thus, a measure of how strong or certain is the rule.

In the sample rule:

If $(X \geq 0.01)$ **and** $(X \leq 0.24)$ **and** $(Y \geq 0.01)$ **and** $(Y \leq 0.24)$
Then Class = Black [v_o:$conf_{ct}$ = 1.0, v_o:sup_{ct} = 0.5, v_o:str_{ct} = 0.9670]
Class = White [v_o:$conf_{ct}$ = 0.0, v_o:sup_{ct} = 0.0, v_o:str_{ct} = 0.0]

all the data instances that satisfy the premise have output category Black. To be exact, it has 100% confidence for output category Black, and consequently, 0% confidence for White. In the same rule, 50% of the total number of data instances in the given training data set *D* satisfy the premise and has output category Black as v_o:sup_{ct} = 0.5 for Black, while none of the total number of data instances in *D* are found to have output category White with the given premise of the rule as v_o:sup_{ct} = 0.0 for White.

2.2 Rule Certainty

Since the expert system MYCIN was built, *certainty factor* has been used as a way of combining *belief* and *disbelief* into a single number [26-28]. In the hybrid model presented here, a certainty ($cert_{ri}$) is attached to each rule.

This certainty describes how much belief we have in a rule depending on how much evidence shows support for the rule. As the rule agrees with more data instances, there is stronger evidence to believe in the rule, thus, $cert_{ri}$ should increase. It should decrease otherwise. The certainty of a rule consists of all traces of theory and evidence that lend credence to the belief. This can be initially provided by human-users and subsequently modified (increased or decreased) as newer evidences are encountered.

In our model, the $cert_{ri}$ attached to any rule at any given time ranges from 0 to 1. The initial value set for all $cert_{ri}$ is either a value assigned by an expert if the rule was provided by the expert, or set by the hybrid intelligent system if the rule was extracted from data.

The hybrid intelligent system sets $cert_{ri}$ as a function of the NN learning accuracy and the strengths of the output categories. That is,

$$cert_{ri} = \mathrm{f}(acc_{train}, r_i{:}v_o{:}str_{ck\in \mathbf{C}}) \tag{4}$$

In the system, $cert_{ri}$ is computed as the product of acc_{train} and the square of max $_{ck\in \mathbf{C}}$ $\{ r_i{:}v_o{:}str_{ck} \}$.

Higher initial values for strengths give higher initial values for certainties. The factor acc_{train} is the learning accuracy of the whole rule set as extracted from DIMLP. This factor also gives an indication of how certain are the rules extracted. If the value of this factor is 100%, then the rule extraction was very reliable. It is less reliable otherwise.

Adding a rule certainty to the sample rule would give:

If $(X \geq 0.01)$ **and** $(X \leq 0.24)$ **and**
$(Y \geq 0.01)$ **and** $(Y \leq 0.24)$
Then Class = Black [$v_o{:}conf_{ct} = 1.0$,
$v_o{:}sup_{ct} = 0.5$, $v_o{:}str_{ct} = 0.9670$]
Class = White [$v_o{:}conf_{ct} = 0.0$,
$v_o{:}sup_{ct} = 0.0$, $v_o{:}str_{ct} = 0.0$]
[$cert_{ri} = 0.9$]

The output category properties and rule properties are used during comparison and fusion of sets of rules. As successive rule refinement is performed, the bound values of the input variables are adjusted. They can increase or decrease depending on how the new evidences confirm or contradict the rule.

Consequently, the output variable properties are changed and so are the rule certainties.

3 Belief-Evidence Fusion

During the *fusion* of the existing belief rule set and the new evidence rule set, rules from both sets are compared in terms of their *premise* (specifically, the *bounds* covered by the conditions in the antecedent), *consequents*, and their *properties*. That is, we compare each rule r_b in the belief rule set $\mathfrak{R}_b$ and each rule r_e in the evidence rule set $\mathfrak{R}_e$ as illustrated in Fig. 2.

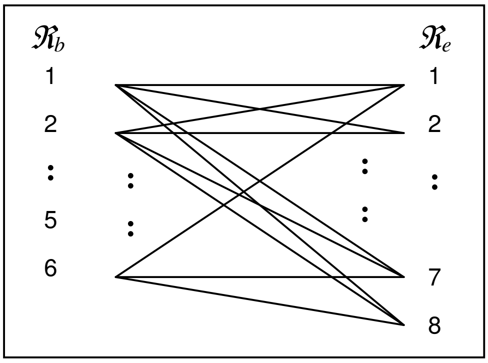

Fig. 2. Comparison of belief and evidence rules

3.1 Rule Similarity Measurement and Assessment

The interval for the condition of an input variable in a rule is a continuum of points or a set of points. Thus, in this research, a rule is viewed as occupying a certain region in the *hyperspace*, which in this case is a *hyperbox*. Each input variable in the domain concerned is treated as one dimension in the hyperspace. Thus, with P input variables in the domain, a rule is mapped to a P-dimensional hyperbox.

Each rule occupying a hyperbox covers a certain amount of space. Two rules then may or may not overlap in the hyperspace. The amount of *intersection* of space between two rules is a measure of how much they may affect each other. Two rules having no intersection do not affect each other at all. However, with an existence of such an intersection, similarity (or dissimilarity) between the two rules in terms of other rule properties must be measured.

Fig. 3 illustrates an example of comparing two rules r_a and r_b in terms of antecedents given two input variables U and V in the domain. In the figure, r_a has lower bound 0 and upper bound 0.8 for its input variable V and lower bound 0.4 and upper bound 1 for its input variable U. On the other hand, r_b has lower bound 0.2 and upper bound 0.9 for its input variable V and lower bound 0.2 and upper bound 0.9 for its input variable U. The intersection can easily be determined to be [0.2, 0.9] X [0.4, 0.9] or has an area of 0.7 * 0.5 or 0.35.

To determine the *degree* to which the intersection affects r_a, we simply divide the amount of intersection by the area occupied by rule r_a. That is, given that r_a has an area of 0.8 * 0.6, the degree of influence on r_a is 0.35/0.48 or 0.7292. The same line of reasoning can also be applied to the degree of influence over rule r_b.

Another measure for the degree of overlap between two rules is the quotient of the amount of intersection and the *smaller* of the two spaces. This gives us a notion of how one rule is absorbed by the other rule. Here, a value equal to 1 implies that one rule is completely inside the hyperbox of the other rule. A value very near 1 implies that one rule is almost completely absorbed by the other rule in terms of space.

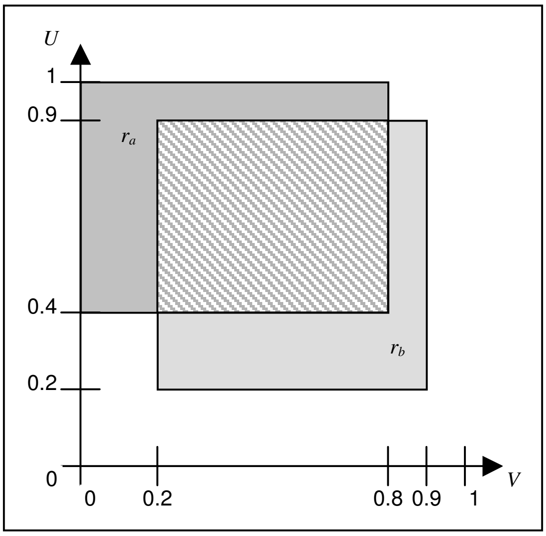

Fig. 3. Comparison of premises of rules r_a and r_b

Eq. (5-6) are formulas for computing the hyperspace occupied by rule r_b and r_e respectively. Eq. (7) is the formula for computing the intersection of the rules with respect to one input variable only.

$$r_b{:}\omega = \prod_{vp \in V} |[r_b{:}l_{vp},\ r_b{:}u_{vp}]| \tag{5}$$

$$r_e{:}\omega = \prod_{vp \in V} |[r_e{:}l_{vp},\ r_e{:}u_{vp}]| \tag{6}$$

$$I_{vp}(r_b{:}v_p,\ r_e{:}v_p) = |[r_b{:}l_{vp},\ r_b{:}u_{vp}] \cap [r_e{:}l_{vp},\ r_e{:}u_{vp}]| \tag{7}$$

Formulas for input variable degree of overlap (i.e., $S_{rb:vp}(r_b{:}v_p,\ r_e{:}v_p)$, $S_{re:vp}(r_b{:}v_p,\ r_e{:}v_p)$, $S_{min:vp}(r_b{:}v_p,\ r_e{:}v_p)$) are given in Eq. (8-10). Eq. (8) is the ratio of the intersection with respect to the length of the variable $r_b{:}v_p$ for a particular input variable while Eq. (9) is a ratio with respect to that of variable $r_e{:}v_p$ and Eq. (10) is with respect to the minimum between $r_b{:}v_p$ and $r_e{:}v_p$.

$$S_{rb:vp}(r_b{:}v_p,\ r_e{:}v_p) = I(r_b{:}v_p,\ r_e{:}v_p)\ /\ r_b{:}v_p \tag{8}$$

$$S_{re:vp}(r_b{:}v_p,\ r_e{:}v_p) = I(r_b{:}v_p,\ r_e{:}v_p)\ /\ r_e{:}v_p \tag{9}$$

$$S_{min:vp}(r_b{:}v_p,\ r_e{:}v_p) = I(r_b{:}v_p,\ r_e{:}v_p)\ /\ \min(r_b{:}v_p,\ r_e{:}v_p) \tag{10}$$

Hyperspace overlap between two rules' input variables is computed using Eq. (11). It is simply the product of the respective variable intersections or overlap.

$$I_\omega(r_b{:}\omega,\ r_e{:}\omega) = \prod_{\forall vp \in V} I_{vp}(r_b{:}v_p,\ r_e{:}v_p) \tag{11}$$

Various degrees of hyperspace overlap (i.e., $S_{rb:\omega}(r_b{:}\omega,\ r_e{:}\omega)$, $S_{re:\omega}(r_b{:}\omega,\ r_e{:}\omega)$, $S_{min:\omega}(r_b{:}\omega,\ r_e{:}\omega)$) between belief rules r_b in the belief rule set $\mathfrak{R}_b$ and evidence rules r_e in the evidence rule set $\mathfrak{R}_e$ are given in Eq. (12-14). The difference lies only in the denominator which is either the hyperspace occupied by r_b, or the hyperspace occupied by r_e, or the smaller of these two hyperspaces.

$$S_{rb:\omega}(r_b{:}\omega,\ r_e{:}\omega) = I(r_b{:}\omega,\ r_e{:}\omega)\ /\ r_b{:}\omega \tag{12}$$

$$S_{re:\omega}(r_b{:}\omega,\ r_e{:}\omega) = I(r_b{:}\omega,\ r_e{:}\omega)\ /\ r_e{:}\omega \tag{13}$$

$$S_{min:\omega}(r_b{:}\omega,\ r_e{:}\omega) = I(r_b{:}\omega,\ r_e{:}\omega)\ /\ \min(r_b{:}\omega,\ r_e{:}\omega) \tag{14}$$

Two rules that overlap in hyperspace may have their strengths of evidences for the output categories agree or disagree. The *degree* to which the *strength of evidences* of the categories of two rules r_a and r_b agree or disagree is computed using a function $S_{str}(r_a{:}\{\ str_{ci}\ \},\ r_b{:}\{\ str_{ci}\ \})$ whose value ranges from 0 to 1. The higher the value of the similarity function, the more similar the output categories of the rules are.

3.2 Rule Comparison Interpretations

The *similarity* (or dissimilarity) measurement between each pair of rules, when assessed, leads to various types of *interpretations* as to how the two rules in a rule pair relate. Rule comparison interpretations may be one of the following:

- *new* r_e rules; these are evidence rules that have no or very small antecedent similarity with any existing r_b rules, thus are considered unseen before; an example is rule r_e in Fig. 4

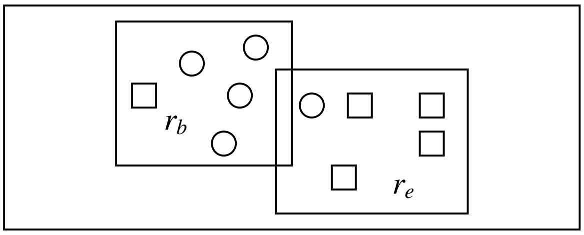

Fig. 4. New evidence rule

- *contradicted* or challenged r_b rules; these are rules that are being refuted by new evidences; an example is rule r_b in Fig. 5 which has instances different from those of rule r_e in their intersection

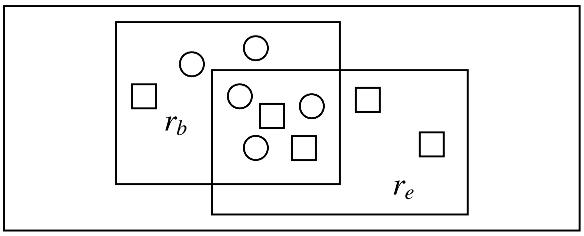

Fig. 5. Contradicted belief rule

- *confirmed* or modified r_b rules; these are rules that are supported by new evidences; an example is rule r_b in Fig. 6 which is supported by evidence rule r_e

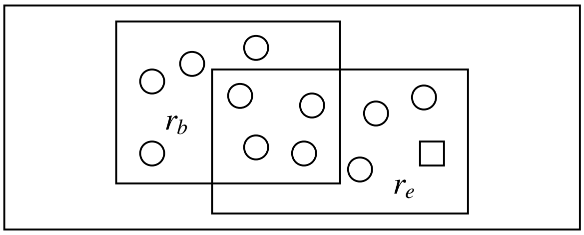

Fig. 6. Confirmed belief rule

- *unsupported* r_b rules; these are belief rules that did overlap significantly with any evidence rule; that is, it did not appear in the new evidence rule set; or it may be the case that the current evidences do not say much about the rule

3.3 The Fusion Process

The various forms of interpretations of the comparisons among rules call for appropriate actions to be performed in order to handle fusion of the belief rules in the belief rule set and evidence rules in the evidence rule set. These fusion actions should help in the refinement of existing belief rules in the rulebase. As more evidence rules are presented in the succeeding iterations, the rulebase of the expert system is expected to become more complete and more correct, and thus, should perform better.

Fusion basically involves the following steps:

- *processing* of the newly extracted evidence rules which involves:
  a. *removal* of *poor* evidence rules; such rules are possibly noise rules; this sub-step would be a means of *forgetting* or *unlearning* knowledge that are not valid or certain
  b. *joining* of *contiguous* or adjacent evidence rules whose consequents are the same or very similar; this would produce rules which have bigger bounds in the conditions in their premise or antecedents; this *expansion* of bounds makes this sub-step to be a means to make *generalizations*; this expansion may also result to *don't care values* or near don't care values for an input variable, and thus covers the case of *deletion of input variables* in the premise of a rule, which in itself is a generalization process
  c. *merging* of *hyperspace-redundant* evidence rules whose consequents are the same or very similar; this would eliminate rules which are redundant
- *modification* of belief rules
  a. *adjusting* the *bounds* of the input variables of belief rules vis-à-vis the evidence rules affecting or contradicting them; the bounds of belief rules are either *shrunk* or *retained*; shrinking of bounds or *contraction* allows the system to make *specializations*; contraction can cause don't care values to become non-don't care values for an input variable, and thus cover the case of *addition of input variables*
  b. *updating* the *certainty* values of belief rules in terms of *certainties* vis-à-vis the evidence rules affecting them; the certainty values of belief rules are either *increased* or *decreased*; certainties of *contradicted* rules are decreased and certainties of *unsupported* rules decay; certainties of *confirmed* rules are increased; this sub-step becomes a means for *learning* knowledge
- *combining* the set of belief rules and the set of evidence rules as the new rulebase; this sub-step allows *new* evidence rules to be accommodated into the new rulebase
- performing the first step (i.e., removal of poor rules, joining of contiguous rules, and merging of redundant rules in the new rulebase); the new rulebase must be inspected for rules that has degraded in quality, for rules that might have become contiguous, and for rules that might have become redundant; deleting rules that have degraded in quality is a means of *forgetting* or *unlearning* knowledge

4 Rule Refinement Experiments

The hybrid intelligent system was experimented with hypothetical data, and data from repositories usually used for machine learning algorithms.

4.1 Hypothetical Data

The testing of the system using the hypothetical *checker board* data set results to 98% accuracy.

This hypothetical domain consists of data instances with two input attributes X and Y both ranging from 0 to 1, and an output class which can be of category **B** (for *black*) or **W** (for *white*). The two-dimensional space of input variables X and Y are partitioned into n x m rectangular subspaces. For a 2 x 4 partitioning, the topmost left corner subspace will have boundaries $0 \leq x < 0.25$ and $0 \leq y < 0.5$. The categories of any two adjacent subspaces are different from each other and the topmost left corner subspace is labeled **B** (black).

An illustration of a sample experiment with this domain is presented next. The RBS starts with an existing belief structure that is incomplete and incorrect. As shown in Fig. 7, the initial belief of the expert consists of three rules only. These are: $0 \leq x < 0.5$ and $0 \leq y \leq 1$ believed to be more of **B**, $0.5 \leq x < 0.75$ and $0 \leq y \leq 1$ to be more of **W**, and $0.75 \leq x \leq 1$ and $0 \leq y \leq 1$ to be quite grey but still closer to **B**.

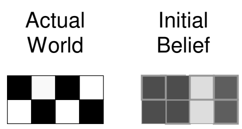

Fig. 7. Checker Board Experiment Initial Belief and Actual World

The starting set of belief provided by the expert obviously has some imperfections. By submitting a series of evidences for the system to learn, we can correct the erroneous belief structure. The series of evidences presented is shown in Fig. 8.

In iteration 0, as shown in Fig. 8(a), six rules can be extracted from the evidences provided. Of the six rules, three will match the region $0 \le x < 0.5$ and $0 \le y \le 1$. This increases the blackness of the matched region because there are more black evidence than white evidence presented for that region.

The original black region $0.75 \le x \le 1$ and $0 \le y \le 1$ changed very slightly only. This is because the evidence presented for that region has an almost equal amount of black and white evidence. On the other hand, for the region $0.5 \le x < 0.75$ and $0 \le y \le 1$ which is originally believed to be more of **W**, there is evidence that its upper portion must be **B**. This causes the upper region to be pulled towards being black. We can observe that at this point, the revised set of belief is still very far from the correct view of the world.

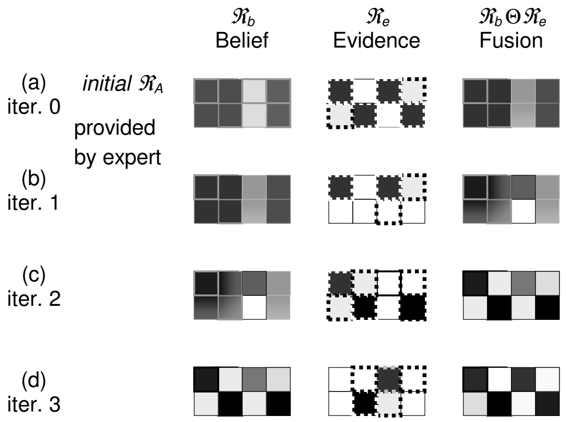

Fig. 8. Checker Board Successive Rule Refinement

In iteration 1, four rules appeared that again would challenge the existing belief $\Re_b$. The region $0 \le x < 0.5$ and $0 \le y \le 1$ has now been constrained or shrunk to a smaller region as there has been no evidence that the rest of that region is black. The black region $0.75 \le x \le 1$ and $0 \le y \le 1$, on the other hand, has been transformed to smaller region of grayish color as evidence presented tells that it should be white.

The evidences $0.5 \le x < 0.75$ and $0 \le y < 0.5$ (**B**) and $0.5 \le x < 0.75$ and $0 \le y \le 1$ (**W**) finally forces the upper portion of the region $0.5 \le x < 0.75$ and $0 \le y \le 1$ to be separately believed as more of black, while its lower portion as separately white. Notice in Fig. 8(b) that the view of the world has further improved.

The next iteration, iteration 2, presents six rules extracted from evidence as shown in Fig. 8(c). Of the six rules, only the rule $0 \le x < 0.25$ and $0 \le y < 0.5$ (**B**) would have a significant match with the shrunk region which was originally $0 \le x < 0.5$ and $0 \le y \le 1$. The three evidence rules adjacent to $0 \le x < 0.25$ and $0 \le y < 0.5$ (**B**) are assimilated as new rules for the belief structure as they have no or little significant match with the existing rules in $\Re_b$. Notice in Fig. 8(c) that this has greatly transformed the region $0 \le x < 0.5$ and $0 \le y \le 1$ to believe that it must consist of alternate black and white subspaces.

Since there were no evidence presented for rule $0.5 \le x < 0.75$ and $0 \le y < 0.5$, and rule $0.5 \le x < 0.75$ and $0 \le y \le 1$, their certainty for blackness and whiteness decayed a little bit.

The region $0.75 \le x \le 1$ and $0 \le y \le 1$, which has been shrunk and has turned grayish, was now clearly delineated into two different subspaces of different colors (upper portion as **W** and lower portion as **B**).

Iteration 2 results to a very good view of the real world as can be seen in Fig. 8(c). The last iteration in this experiment which introduces five new evidence rules further improves this view as shown in Fig. 8(d).

This sample experiment shows that through successive rule refinement, we are able to complete and correct an existing belief structure by introducing increasing amounts of evidence. The situation illustrated is a very common scenario in the real world, where a belief structure is partially incomplete or incorrect, although functioning properly to some extent. The belief structure may remain partially incomplete or incorrect for a long time. However, as new evidence is provided, the system is at least able to improve itself by fine-tuning some of its belief rules and deleting others that are consistently being negated by evidence.

This is akin to inexperienced medical doctors, for example, who may have a medical "belief structure" that is formed through medical training that may be far from being complete. With experience, as more and more clinical cases are encountered, the medical doctor gets to refine the belief structure and complete it with the various new cases encountered that were not learned while in medical school. Old beliefs may likewise be modified.

4.2 Machine Learning Repository Data

Three machine learning repository data sets available in public websites are used, namely the *Wisconsin breast cancer data set*, the *Iris data set*, and the *Echocardiogram data set*.

For each of these data sets, the available data instances are split randomly into the following:

- 10% for testing set
- 90% for training set

The training set was further divided into several partitions for the successive rule refinement. Since no existing set of rules from experts is available, the first training set was used to train the neural network for rule extraction. This set of extracted rules formed the initial rule set for the hybrid system. The remaining training partitions are used in succession for rule extraction and integration into the existing rule base.

In experimenting with the data sets, 10 folds were performed (10-fold cross validation). Under each fold, a different test set was used. This is to average out the biases of the test sets.

The hybrid system was able to reach 94% to 96% accuracy on the *Wisconsin breast cancer data set* (highest reported accuracy by other systems is 94%), 93% to 95% on the *Iris data set* (highest reported accuracy by other systems is 95%), and 92% to 94% on the *Echocardiogram data set* (highest reported accuracy by other systems is 90%). All of the experiments show that the hybrid system is able to recover from incomplete and incorrect initial knowledge and its performance increases with successive rule refinement.

5 Conclusion

Successive rule refinement is described in this paper as an effective method for belief and evidence *fusion*. The set of "beliefs" is encoded in rule-form (disjunctive normal form). These beliefs are then confirmed, modified, challenged, or left unsupported by all the "evidence" available. Certain new evidences that do not figure in any existing belief are assimilated as new belief.

The fusion of belief and evidence is done through *successive rule refinement* in a hybrid intelligent system model. The available evidence is in the form of raw data. These are first converted into rule-form so that they can be integrated with the existing beliefs about the domain. Converting evidence into rule-form is done through a hybrid system that trains a neural network using the available evidence and extracts rules from the network once it has been sufficiently trained. These are then submitted to the rulebase system for comparison with the existing belief rules. The rules in the rulebase system are then updated.

From the experiments conducted to demonstrate the applicability of the approach, it can be seen that the system's set of beliefs becomes more and more refined and complete as increasing units of evidences are integrated in it. The concepts of *strength of evidence* and *certainty of belief* are used in the various possible actions to be taken when modifying the belief structure when faced with new evidences.

Acknowledgements

DIMLP, the rule extraction system used in the system, was created by Dr. Guido Bologna.

References

[1] Bruce G. Buchanan, and D. C. Wilkins (eds.), *Readings in Knowledge Acquisition and Learning: Automating the Construction and Improvement of Expert Systems*, Morgan Kaufmann, USA, 1993.

[2] David L. Hall, *Mathematical Techniques in Multisensor Data Fusion*, Artech House, Massachusetts, 1992.

[3] Richard T. Antony, *Principles of Data Fusion Automation*. Artech House, Massachusetts, 1995.

[4] Irwin R. Goodman, P. S. Mahler, and H. T. Nguyen, *Mathematics of Data Fusion*, Kluwer Academic Publishers, Netherlands, 1997.

[5] Firooz A. Sadjadi, and B. J. Thompson (eds.), *Selected Papers on Sensor and Data Fusion*, SPIE, Washington, 1996.

[6] P. Korpisaari, and J. Saarinen, *Fusion '99: Second International Conference on Information Fusion*, The International Society of Information Fusion, California, 1999.

[7] Mongi A. Abidi, and R. C. Gonzalez (eds.), *Data Fusion in Robotics and Machine Intelligence,* Academic Press, London, 1992.

[8] Jurg Kohlas, and P. Monney, *A Mathematical Theory of Hints: An Approach to the Dempster-Shafer Theory of Evidence*, Springer-Verlag Berlin Heidelberg, Germany, 1995.

[9] Ronald R. Yager, M. Fedrizzi, and J. Kacprzyk (eds.), *Advances in the Demspter-Shafer Theory of Evidence*, John Wiley & Sons, Canada, 1994.

[10] Glenn Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, London, 1976.

[11] Geoffrey Towell, *Symbolic Knowledge and Neural Networks: Insertion, Refinement, and Extraction*, Doctoral dissertation, Computer Science Department, University of Wisconsin, 1991.

[12] Jhy-Shing Robert Jang, C. T. S. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice Hall, New Jersey, 1997.

[13] Fernando S. Osorio, B. Amy, and A. Cechin, Hybrid Machine Learning Tools - INSS: A Neuro-Symbolic System for Constructive Machine Learning, *Deep fusion of computational and symbolic processing,* Springer Verlag, Berlin, 2000.

[14] I. Cloete, and J. Zurada (Eds.). *Knowledge-Based Neurocomputing*. Cambridge, MA: MIT Press, 2000.

[15] Q. Song, and N. Kasabov. Dynamic Evolving Neuro-Fuzzy Inference System (DENFIS): On-line Learning and Application for Time-Series Prediction. In *Proceedings of the 6th International Conference on Soft Computing.* Iizuka, Japan, 2000.

[16] S. Wermter, and R. Sun (Eds.). *Hybrid Neural Systems.* Heidelberg, Springer, 2000.

[17] Guido Bologna, Rule Extraction from a Multi Layer Perceptron with Staircase Activation Functions, *International Joint Conference on Neural Networks*, Como, Italy, 2000.

[18] Guido Bologna, *Rule Extraction from Linear Combinations of DIMLP Neural Networks*, 2000.

[19] Guido Bologna. On the Validation of the DIMLP Neural Network. In *Proceedings of FLAIRS Conference*, 2002.

[20] Guido Bologna. Rule Extraction from Bagged Neural Networks. In *Proceedings of Hybrid Intelligent Systems*, 2002.

[21] Bing Liu, W. Hsu, and Y. Ma, Integrating Classification and Association Rule Mining, *Knowledge Discovery and Data Mining*, New York, 1998.

[22] Bing Liu, W. Hsu, H-S. Han, and Y. Xia. (2000). Mining Changes for Real-Life Applications. In *Proceedings of the 2nd International Conference on Data Warehousing and Knowledge Discovery.* London.

[23] Bing Liu, Yiming Ma, Ching-Kian Wong, and Philip S. Yu. Scoring the Data Using Association Rules. *Applied Intelligence*, Vol 18, No. 2, 119-135, 2003.

[24] Ming-Syan Chen, Philip S. Yu, and Bing Liu (Eds). Advances in Knoweldge Discovery and Data Mining. *6th Pacific-Asia Conference, PAKDD-2002 Proceedings*, Taipei, Taiwan, May 2002.
[25] J. Han, and M. Kamber. *Data Mining Concepts and Techniques.* USA: Academic Press, 2001.
[26] Joseph Giarratano, and G. Riley, *Expert Systems: Principles and Programming*, PWS Publishing Company, Boston, 1998.
[27] LiMin Fu, *Neural Networks in Computer Intelligence*, McGraw-Hill, USA, 1994.
[28] Bernadette Bouchon-Meunier, L. Valverde, and R. R. Yager (eds.), *Uncertainty in Intelligent Systems*, Elsevier Science Publishers, Amsterdam, 1993.