Linguistic CMAC for Multi-Attribute Decision Making

Hongmei He *Member IAENG* and Jonathan Lawry *

Abstract— **The multi-attribute decision making problem engages in the propagation of information, which often is highly uncertain or imprecise. Cerebellar Model Articulation Controller (CMAC) belongs to the family of feed-forward networks with a single linear trainable layer. CMAC has the feature of fast learning, and is suitable for modeling any non-linear relationship. Combining fuzzy linguistic semantics and CMAC, a linguistic CMAC based on Mass Assignment is proposed to map the relationship between attributes and a decision variable. We use mass assignment of attribute variables to calculate the appropriateness measure that is equivalent to the probability of the unit in the CMAC selected by the attributes. The state of decision variable is decided by the sum of weighted active units in CMAC. We then investigate the equivalence between the black box of the Linguistic CMAC and the transparent box of Linguistic Decision Tree.**

Keywords: Multi-Attribute Decision Making, Linguistic CMAC, Linguistic Decision Tree, Mass Assignment

1 Introduction

For multiple attribute decision making or classification, the underlying relationship between attributes and goal variable is often highly uncertain and imprecise. This requires an integrated treatment of uncertainty and fuzziness when modeling the propagation of information from low-level attributes to high-level goal variables. It is well recognized that the fuzzy measure plays a crucial role in the fusion of multiple attributes. Wang and Chen [16] used the Choquet fuzzy integral and the g-Lamda fuzzy measure to improve significantly the neural network classification accuracy. In recent work, Yang et al. [18] and Van-nam et al. [15] have proposed to aggregate evidence from different attributes on the basis of weighted combination rules in evidence theory, where the underlying idea is to use random set (mass assignment) to provide a unified model of probability and fuzziness.

Label semantics proposed by Lawry [5, 6], which is different with the paradigm of computing with words proposed by Zadeh [19], is a random set based semantics for modeling imprecise concepts where the degree of appropriateness of linguistic expression as a description of a value is measured in terms of how the set of appropriate labels for that value varies across a population. Based on this semantics, a tree-structured model, Linguistic Decision Tree (LDT) was proposed by Qin and Lawry [9]. In such an LDT, transparent label semantic rules of the LDT present an effective way for information propagation between low-level and high-level.

Neural networks have been well used for decision making or classification. The Cerebellar Model Articulation Controller (CMAC) [1, 2] is of that models the structure and function of the part of the brain known as the cerebellum, which is a special feed-forward neural network. CMAC has the unique property of quickly training areas of memory without affecting the whole memory structure due to local training property of CMAC. In a CMAC, each variable is quantized and the problem space is divided into discrete states. A vector of quantized input values specifies a discrete state and is used to generate addresses for retrieving information from memory at this state. Information is distributively stored. This property benefits the nonlinear multiple attribute decision making or classification. In this paper, a linguistic CMAC (LCMAC) based on Mass Assignment is proposed to map the relationship between the attributes and the decision variable. We investigate the equivalence between the black box of the LCMAC and the transparent box of an LDT.

2 Label Semantics

Fuzzy discretisation provides an interpretation between numerical data and linguistic data based on Label Semantics, which proposes two fundamental and inter-related measures of the appropriateness of labels as descriptions of an object or value.

Given a finite set of labels $\mathcal{L}$ from which can be generated a set of expressions LE through recursive applications of logical connectives, the measure of appropriateness of an expression $\theta \in LE$ as a description of instance x is denoted by $\mu_\theta(x)$ and quantifies the agent's subjective

*Department of Engineering Mathematics, University of Bristol, UK {H.He,J.Lawry}@bristol.ac.uk

belief that θ can be used to describe x based on his/her (partial) knowledge of the current labelling conventions of the population. From an alternative perspective, when faced with an object to describe, an agent may consider each label in $\mathcal{L}$ and attempt to identify the subset of labels that are appropriate to use. Let this set be denoted by $\mathcal{D}_x$. In the face of their uncertainty regarding labelling conventions the agent will also be uncertain as to the composition of $\mathcal{D}_x$, and in label semantics this is quantified by a probability mass function $m_x : 2^{\mathcal{L}} \rightarrow [0,1]$ on subsets of labels. The relationship between these two measures will be described below.

Unlike linguistic variables [20], which allow for the generation of new label symbols using a syntactic rule, label semantics assumes a finite set of labels $\mathcal{L}$. These are the basic or core labels to describe elements in an underlying domain of discourse Ω. Based on $\mathcal{L}$, the set of label expressions LE is then generated by recursive application of the standard logic connectives as follows:

Definition 2.1. *Label Expressions*
The set of label expressions LE of $\mathcal{L}$ is defined recursively as follows:
- *If $L \in \mathcal{L}$ then $L \in LE$*
- *If $\theta, \varphi \in LE$ then $\neg\theta, \theta \wedge \varphi, \theta \vee \varphi \in LE$*

A mass assignment m_x on sets of labels then quantifies the agent's belief that any particular subset of labels contains all and only the labels with which it is appropriate to describe x.

Definition 2.2. *Mass Assignment on Labels*
$\forall x \in \Omega$ a mass assignment on labels is a function $m_x : 2^{\mathcal{L}} \rightarrow [0,1]$ such that $\sum_{S \subseteq \mathcal{L}} m_x(S) = 1$

Now depending on labeling conventions there may be certain combinations of labels which cannot all be appropriate to describe any object. For example, *small* and *large* cannot both be appropriate. This restricts the possible values of $\mathcal{D}_x$ to the following set of focal elements:

Definition 2.3. *Set of Focal Elements*
Given labels $\mathcal{L}$ together with associated mass assignment $m_x : \forall x \in \Omega$, the set of focal elements for $\mathcal{L}$ is given by:

$$\mathcal{F} = \{S \subseteq \mathcal{L} : \exists x \in \Omega,\ m_x(S) > 0\} \qquad (1)$$

The appropriateness measure, $\mu_\theta(x)$, and the mass m_x are then related to each other on the basis that asserting 'x is θ' provides direct constraints on $\mathcal{D}_x$. For example, asserting 'x is $L_1 \wedge L_2$', for labels $L_1, L_2 \in \mathcal{L}$ is taken as conveying the information that both L_1 and L_2 are appropriate to describe x so that $\{L_1, L_2\} \subseteq \mathcal{D}_x$. Similarly, '$x$ is $\neg L$' implies that L is not appropriate to describe x so $L \notin \mathcal{D}_x$. In general we can recursively define a mapping $\lambda : LE \rightarrow 2^{2^{\mathcal{L}}}$ from expressions to sets of subsets of labels, such that the assertion 'x is θ' directly implies the constraint $\mathcal{D}_x \in \lambda(\theta)$ and where $\lambda(\theta)$ is dependent on the logical structure of θ. For example, if $\mathcal{L} = \{low,\ medium,\ high\}$ then $\lambda(medium \wedge \neg high) = \{\{low, medium\}, \{medium\}\}$ corresponding to those sets of labels which include *medium* but do not include *high*. Hence, the description $\mathcal{D}_x$ provides an alternative to Zadeh's linguistic variables in which the imprecise constraint 'x is θ' on x, is represented by the precise constraint $\mathcal{D}_x \in \lambda(\theta)$, on $\mathcal{D}_x$.

Definition 2.4. *λ-mapping $\lambda : LE \rightarrow 2^{\mathcal{F}}$ is defined recursively as follows: $\forall \theta,\ \varphi \in LE$*
- $\forall L_i \in \mathcal{L}\ \lambda(L_i) = \{F \in \mathcal{F} : L_i \in F\}$
- $\lambda(\theta \wedge \varphi) = \lambda(\theta) \cap \lambda(\varphi)$
- $\lambda(\theta \vee \varphi) = \lambda(\theta) \cup \lambda(\varphi)$
- $\lambda(\neg\theta) = \lambda(\theta)^c$

Therefore, based on the λ-mapping the appropriateness measure are defined as below:

Definition 2.5 (Appropriateness Measure)**.** *Appropriateness measure $\mu_{\theta(x)}$ is evaluated as the sum of mass assignment m_x over those subsets of labels in $\lambda_\theta(x)$, i.e. $\forall \theta \in LE, \forall x \in \Omega, \mu_{\theta(x)} = \sum_{F \in \lambda(\theta)} m_x(F)$.*

For example, if $\mathcal{L} = \{low, medium, high\}$ with focal sets $\{\{l\}, \{l, m\}, \{h\}\}$ and $\theta = low \wedge \neg medium$ then $\mu_{l \wedge \neg m}(x) = \sum_{F : l \in F, m \notin F} m_x(F) = m_x(\{l\})$.

The consonance assumption Appropriateness measures are not in general functional since m_x cannot be uniquely determined from $\mu_L(x) : L \in \mathcal{L}$. However, in the presence of additional assumptions the calculus can be functional. Based on the idea of ordering appropriateness measures on labels defined in multi-attribute models, an assumption is given as follows:

Definition 2.6 (Consonance in Label Semantics)**.** *Given non-zero appropriateness measure on basic labels $\mathcal{L} = \{L_1, L_2, ..., L_n\}$ ordered such that $\mu_{L_i}(x) \geq \mu_{L_{i+1}}(x)$ for $i = 1, ..., n$ then the consonant mass assignment has the form:*

$$m_x(\{L_1, ..., L_n\}) = \mu_{L_n}(x),$$
$$m_x(\phi) = 1 - \mu_{L_1}(x),$$
$$m_x(\{L_1, ..., L_i\}) = \mu_{L_i}(x) - \mu_{L_{i+1}}(x) \ \textit{for}\ i = 1, ..., n.$$

3 LCMAC based on mass assignment

3.1 Basic CMAC

The basic CMAC is a machine that is analogous to the process of cerebellum's work. In CMAC, the input vectors are spoken of as sensory cell firing patterns X, which may be either binary vector or R-ary vector. The appearance of an input vector X on the sensory cells produces an association cell vector A which also either binary or

R-ary. The association cell vector A multiplied by the weight matrix W produces a response vector P. There are two mapping in CMAC:

$$f : X \longrightarrow A,\ g : A \longrightarrow P$$

where, X is sensory input vectors, A is association cell vectors, P is response output vectors. The function f is generally fixed, but the function g depends on the values of weights which may be modified during the data storage (or training) process. When an input vector $X = (x_1, x_2, ..., x_N)$ is presented to the sensory cells, it is mapped into an association cell vector A. Define A^* to be a set of active or nonzero elements of A shown as in Figure 1. The response cell sums the values of the weights attached to active association cells to produce the output vector Y. Only the non-zero elements comprising A^* will affect this sum. The input vector X can be considered as an address. If for any input X, it is expected to change the contents Y, then we only need to adjust the weights attached to association cells in A^*.

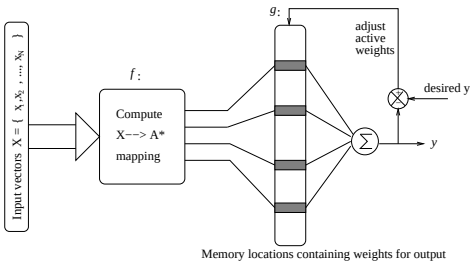

Figure 1: The structure of basic CMAC

3.2 Mapping with linguistic labels of input vectors

The new LCMAC is based on the mass assignments on the focal sets for each input attribute. In the LCMAC, the first mapping is the fuzzy discretisation of input attributes. Given appropriateness measure for each attribute, mass assignments on focal elements can be obtained according to the consonance assumption presented in Section 2.

Given input vector $X = (x_1, x_2, ..., x_N)$, for each attribute x_i, $i = 1, ..., N$, the label set $\mathcal{L} = \{L_1, L_2, ..., L_n\}$ is used to describe the attribute. The focal set for the attribute will be $\mathcal{F} = \{\{L_1\}, \{L_1, L_2\}, \{L_2\}, ..., \{L_{n-1}, L_n\}, \{L_n\}\}$. The size of the focal set is $f_n = 2n - 1$. F_{ij} denotes the j^{th} focal element of the i^{th} attribute. For example, Figure 2 illustrates an LCMAC with 2-dimension input space, where each focal element is associated to one unit of memory. Given a value of the input vector (x_1, x_2), where each attribute can be described with three labels, we can calculate the mass assignments $m_x(F_{1j})$ and $m_x(F_{2j}), j = 1, ..., 5$. For each attribute, usually there exist two neighbouring focal elements on which the mass assignments are not zero. Thus four units of memory are active. If $m_x(F_{1i}) \neq 0$, $m_x(F_{1(i+1)}) \neq 0$, $m_x(F_{2j}) \neq 0$, $m_x(F_{2(j+1)}) \neq 0$, then units M_{ij}, $M_{(i+1)j}$, $M_{i(j+1)}$, and $M_{(i+1)(j+1)}$ are active.

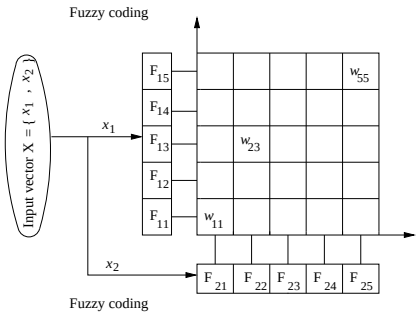

Figure 2: The structure of LCMAC

Theorem 3.1. *If every focal element of an attribute is associated to a unit of memory, for N-dimension input vector $X = (x_1, x_2, ..., x_N)$, the active space of memory is in an N-dimension hypercube with edge length 2 (i.e. 2^N units of memory).*

Proof. According to labeling conventions and consonance assumption between appropriateness and mass assignment in Section 2, for any pair of neighbouring focal elements F_i and F_{i+1}, $\exists x$, $m_{F_i}(x) \neq 0$ and $m_{F_{i+1}}(x) \neq 0$. In other words, $\forall x$, at most on one pair of neighbouring focal elements F_i and F_{i+1}, $m_{F_i}(x) \neq 0$ and $m_{F_{i+1}}(x) \neq 0$. There are two possible extreme cases: $m_{x_d}(F_{d_1}) \neq 0$, but $m_{x_d}(F_{d_2}) = 0$, and $m_{x_d}(F_{d_{(2n-2)}}) = 0$, but $m_{x_d}(F_{d_{(2n-1)}}) \neq 0$, where n is the number of labels that are used to describe x_d. If $m_{F_i}(x) \neq 0$, for non-neighbouring focal elements $F_{i\pm k}$, $k > 1$ and $2n - 1 \geq i \pm k > 0$, $m_{F_{i\pm k}}(x) = 0$. Therefore, the active space is in a N-dimension hypercube with edge length 2, which holds 2^N units of memory. □

3.3 Response mapping

3.3.1 Fine grain mapping

Each unit of memory is used to store a weight, which represents the probability that an input region described by label expressions occurs in the current database. The input region is constrained by mass assignments on focal elements of each attribute in the input vector (see Figure 2). A unit is addressed with the focal element indices $(a_1, ..., a_N)$ for all input attributes $(x_1, ..., x_N)$. For example, in Figure 2, vector $X = (x_1, x_2)$, the weight w_{11} responds to the unit whose address is $(1, 1)$, where the first '1' is indicated by the first focal element F_{11} of attribute x_1,and the second '1' is indicated by the first focal element F_{21} of attribute x_2. Given an input vector

$X = (x_1, ..., x_N)$, the probability $Pr(M|X)$ that a unit is located is the product of all mass assignments on focal elements of all input attributes, and formalized as below:

$$Pr(M|X) = \prod_{d=1}^{N} m_{x_d}(F_{d_i}), \quad (2)$$

where, M denotes a unit of memory, and F_{d_i} is a focal element for the d-th attribute, and the address of M is given by focal element indices $(d_1, d_2, ..., d_N)$. Assuming there are l possible labels $\mathcal{L}_y = \{L_{y_1}, ..., L_{y_l}\}$ to describe the goal variable Y. The output of the neural network is a vector $Y = \{y_1, ..., y_l\}$, where, y_k indicates how appropriate L_{y_k} is used to describe the goal based on the neural network given an input vector. The weight is a vector $W_a = \{w_1, w_2, ..., w_l\}$, which represents the distributed probability that the goal belongs to a class (label) in the unit of memory. Therefore, according to Jeffery's rule, the probability of a label L_{y_k} is the sum of probabilities in all active units of memory, and formalized as below:

$$P(y_k|X) = \sum_{a \in A^*} w_k Pr(M_a|X) = \sum_{a \in A^*} w_k \prod_{d=1}^{N} m_{x_d}(F_{d_i}). \quad (3)$$

3.3.2 Overlapping coarse grain mapping

Each active area of memory responses to a weight, which suggests the probability of occurrences of the input region described with label expressions in the current database. Obviously, each input region corresponds to an active area (See Figure 3). According to the formula in Defi-

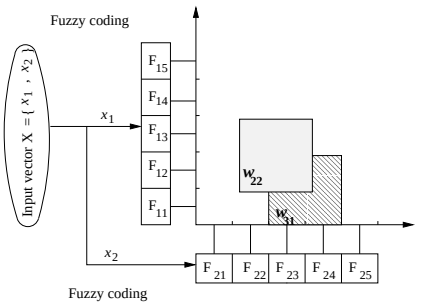

Figure 3: The overlapping map of LCMAC

nition 2.5, given an input vector X, the probability that an active area A^* is located can be calculated by:

$$Pr(A^*|X) = \prod_{d=1}^{N} \mu_\theta = \prod_{d=1}^{N} (m_{x_d}(F_{d_i}) + m_{x_d}(F_{d_{i+1}})), \quad (4)$$

where, X is N-dimension vector, F_{d_i} and $F_{d_{i+1}}$ are two neighbouring focal elements for attribute x_d, and the corresponding mass assignments of x_d on the two focal elements are not zero. Any pair of neighbouring active areas overlap. The probability that an active area is located is only related to the given input vector. Given input vector X, and the goal variable $Y = \{y_1, y_2, ..., y_l\}$, according to Jeffery's rule, the probability that a label L_{y_k} is appropriate to describe Y is the product of the weight in the active area and the probability that the active area is located. So, according to Formula (4), it can be written as:

$$\begin{aligned} P(y_k|X) &= w_k Pr(A^*|X) \\ &= w_k \prod_{d=1}^{N} (m_{x_d}(F_{d_i}) + m_{x_d}(F_{d_{i+1}})). \end{aligned} \quad (5)$$

4 The convergence of the LCMAC

The purpose of training the neural network is to adjust the weights to make the LCMAC approach the desired output. Sayil and Lee [12] compared 12 training algorithms, and suggested a hybrid maximum error algorithm [8] with neighborhood training for CMAC. We now investigate the convergence of the LCMAC. Hirsch [3] viewed a neural network as a nonlinear dynamic system called Neurodynamics, which presents a conceptual and eclectic methodological approach for understanding neural network activity. Assuming the dynamic system with N state variables $v_1, v_2, ..., v_N$, the network motion equation is $\frac{du_i}{dt} = -\frac{\partial E}{\partial v_i}$, where u_i and v_i are the input and output of the i-th neuron. Takefuji and Szu has proved [13]:

$$\begin{aligned} \frac{dE}{dt} &= \sum \frac{dv_i}{dt} \frac{\partial E}{\partial v_i} = \sum \frac{dv_i}{dt}(-\frac{du_i}{dt}) \\ &= -\sum (\frac{dv_i}{du_i})(\frac{du_i}{dt})(\frac{du_i}{dt}) = -\sum (\frac{dv_i}{du_i})(\frac{du_i}{dt})^2. \end{aligned}$$

Therefore, convergence of a neural network does not depend on the model. As long as the output v_i is the continuous, differentiable and monotonous increasing function of input u_i, namely, there exists the relationship between outputs and inputs of neurons $\frac{dv_i}{du_i} > 0$, the neural network always converges with a negative grade. Finally, the neural network arrives at a stable state with $\frac{dE}{dt} = 0$. Here we define the activation function as below:

$$v_i = \begin{cases} 1 & u_i \geq 1 \\ u_i & 0 \leq u_i < 1 \\ 0 & u_i < 0. \end{cases} \quad (6)$$

In LCMAC, each cell of response mapping represents a neuron. The weight in each cell indicates the state of each neuron. If the input u_i and output v_i of each neuron have the relationship u_i=v_i, then we have $\Delta W = -\frac{\partial E}{\partial W} \times \Delta t$. The Least Mean Square (LMS) algorithm is well-known for neural network training. Miller et al. used LMS to train the CMAC [7]. We can define the Mean Square Error as $E_k(t) = \gamma\ (D_k - y_k(t))^2/2$, where D_k indicates if the goal belongs to the class C_k. If the goal belongs to C_k, then $D_k = 1$, otherwise $D_k = 0$. Assuming $\Delta t = 1$, then we have:

$$\Delta w_k = \gamma (D_k - y_k(t)) \frac{\partial y_k(t)}{\partial w_k}, \quad (7)$$

where γ is the learning factor. Given a training sample X, we can calculate the value of Equation (7), which will

be as a correction to each of the memory cells activated by the input vector. For fine grain mapping, according to Equation (3) and (7), the motion function is:

$$\Delta w_k = \gamma(D_k - \sum_a w_k Pr(M_a|X))w_k Pr(M_a|X). \quad (8)$$

For coarse grain mapping, according to Equation (5) and (7), the motion function is:

$$\Delta w_k = \gamma(D_k - w_k Pr(A^*|X))Pr(A^*|X). \quad (9)$$

5 Equivalence to an LDT

5.1 Linguistic decision trees

In an LDT [9, 10], the nodes are attributes, such as $x_1, ..., x_N$, and the edges are label expressions describing each attribute. A branch B is a conjunction of expressions $\theta_1 \wedge ... \wedge \theta_N$, where θ_k is the label expression of an edge in branch B for $k = 1, ...N$. Each branch also is augmented by a set of conditional mass values $m(F|B)$, which is equivalent to $P(F|B)$, for each output focal element $F \in \mathcal{F}_y$.

5.1.1 A focal element linguistic decision tree

Qin and Lawry [9, 10] suggested to create Focal Element Linguistic Decision Trees (FELDTs) from database. In an FELDT, branches have the form $B = (F_{i1}, ..., F_{iN})$ where x_{id} is the attribute node at the depth d of B, and $F_{id} \in \mathcal{F}_{id}$ for $d = 1, ..., N$. If we use the LID3 algorithm [9, 10] to learning the FELDT, the probabilities $P(F_y|B)$ for a focal element $F_y \in \mathcal{F}_y$ conditional on a branch B can be evaluated from a database DB as below:

$$\begin{aligned} P(F_y|B) &= \frac{\sum_{r\in DB_{F_y}} m_{\langle x_{i_1}(r),...,x_{i_N}(r)\rangle}(F_{i_1}, ..., F_{i_N})}{\sum_{r\in DB} m_{\langle x_{i_1}(r),...,x_{i_N}(r)\rangle}(F_{i_1}, ..., F_{i_N})} \\ &= \frac{\sum_{r\in DB_{F_y}} \prod_{v=1}^{N} m_{x_{i_v}(r)}}{\sum_{r\in DB} \prod_{v=1}^{N} m_{x_{i_v}(r)}}. \end{aligned} \quad (10)$$

According to Jeffery's rule, the mass assignment of goal variable y on a focal element can be calculated as follows:

$$M_{F_y}(y) = \sum_{i=1}^{b}(\prod_{d=1}^{N}(m_{x_{i_d}}(F)))P(F_y|B_i), \quad (11)$$

where, b is the number of branches, and N is the number of attributes or the depth of a branch in the FELDT; Here we assume without the limitation of the depth, so the depth of all branches is the same as the number of attributes; x_{i_d} is the attribute incident to the edge at the d-th layer of branch B_i.

5.1.2 Dual-edge LDTs

Another kind of LDT is the one whose edge grain is two neighbouring focal elements. However, two neighbouring edges overlapping on a focal element. From each node there are $l - 1$ edges, where l the size of focal set. For an example, attribute x_1 in an LDT has the focal set $\{F_1, ..., F_9\} = \{\{vl\}, \{vl, l\}, \{l\}, \{l, m\}, \{m\}, \{m, h\}, \{h\}, \{h, vh\}, \{vh\}\}$. Then we have edges from node x_1, such as $\{F_1, F_2\}, \{F_2, F_3\}, ..., \{F_7, F_8\}, \{F_8, F_9\}$. We call the LDT as dual-edge LDT. The revised conditional probability of a focal element $F_y \in \mathcal{F}_y$ that is appropriate to describe a goal given the branch B can be evaluated from DB according to:

$$P(F_y|B) = \frac{\sum_{r\in DB_{F_y}} \prod_{d=1}^{N}(m_{x_{i_d}}(F_j) + m_{x_{i_d}}(F_{j+1}))}{\sum_{r\in DB} \prod_{d=1}^{N}(m_{x_{i_d}}(F_j) + m_{x_{i_d}}(F_{j+1}))} \quad (12)$$

This dual-edge LDT needs similar space as an FELDT does, but the calculation is based on a unique branch with Equation (13).

$$\begin{aligned} M_{F_y}(y) &= \prod_{d=1}^{N} \mu(\theta_d)P(F_y|B) \\ &= \prod_{d=1}^{N}(m_{x_{i_d}}(F_j) + m_{x_{i_d}}(F_{j+1}))P(F_y|B)), \end{aligned} \quad (13)$$

where, x_{i_d} is the attribute incident to the edge at the d-th layer of branch B, and $m_{x_{i_d}}(F_j)$ and $m_{x_{i_d}}(F_{j+1})$ are the non-zero mass assignments of attribute x_{i_d} on two neighbouring focal elements F_j and F_{j+1}, corresponding to the edge.

5.2 Comparing an LCMAC with an LDT

From Section 3.3, whether the response mapping is fine grain or coarse grain, the final output of the neural network is the distributed probabilities that the goal can be described with each label. From this point of view, an LCMAC has the same effectiveness as an LDT, presenting the mass assignments on labels of a goal variable.

Comparing the fine grain mapping LCMAC with FELDT, from Equation (3) and (11), we can see that the difference between the two equations lies in w_k for fine grain mapping LCMAC, which implies the probability that the goal belongs to a class or is described with a label conditional on a unit in the active area, and $P(F_y|B)$ for FELDT, which is the conditional mass assignment of the goal variable y on focal element F_y given branch B. Therefore, a unit in the active area in an fine grain LCMAC is equivalent to a branch in FELDT.

Similarly, comparing the coarse grain mapping LCMAC with the dual-edge LDT, from Equation (5) and (13),

there exists the difference as above, but w_k indicates the the probability that the goal belongs to a class in the active area. Therefore, an active area in coarse grain mapping LCMAC is equivalent to a branch in dual-edge LDT.

The large difference between an LCMAC and an LDT should be in the learning process. For an LCMAC, learning algorithms vary with different strategies based on the LMS algorithm, which uses the feedback of the error of desired output and calculated output to correct the state of a neuron, so that the neural network arrives at a stable state with least square error, and the training process only involves the neurons' state in the active area located by a given sample, while for an LDT, the learning algorithm LID3 proposed by Qin and Lawry [9, 10], is an extension of classic ID3 algorithm [11], the basic step of which is to calculate the conditional probability that a goal can be described with a label, then to decide which attribute is extended to current node in the tree according to the expected entropy.

6 Conclusion

For multiple attribute decision making or classification, we presented an LCMAC by combining the Label Semantics based on mass assignment of attributes, and investigated the convergence of the neural network. It is shown that an LCMAC and an LDT are functionally equivalent. A unit of memory in the fine grain mapping LCMAC is equivalent to a branch in an FELDT, while a unit of memory in coarse grain mapping LCMAC is equivalent to a branch in an dual-edge LDT. But they are different in their training processes. In order to validate the performance of an LCMAC, simulation of the model and experiments on some benchmark databases will be the further work. We will examine the performance of an LCMAC and an LDT through the further experiments.

References

[1] Albus, J. S., A new approach to manipulator control: the cerebellar model articulation controller(CMAC), *Journal of Dynamic Systems, Measurement and Control Transactions on ASME*, **97**, (1975), pp. 200-227.

[2] Albus, J. S., Data Storage in the Cerebellar Model Articulation Controller (CMAC), *Journal of Dynamic Systems, Measurement and Control Transactions on ASME*, **97**, (1975), pp. 228-233.

[3] Hirsch, M. W., Convergent activation dynamics in continuous time networks. Proc. Nat. Acad. Sci. **2**, (1989), pp. 331-349.

[4] Lawry, J., Appropriateness Measures: An Uncertainty Measure for Vague Concepts, to appear in *Synthese*, (2007).

[5] Lawry, J., *Modeling and Reasoning with Vague Concepts*, (Kacprzyk, J. Ed.), Springer, (2006).

[6] Lawry, J., A framework for Linguistic Modeling, *Artificial Intelligence*, **155**, (2004), pp. 1-39.

[7] Miller, W. T., Filson, H. G. and Craft, L. G., Application of A General Learning Algorithm To the Control of Robotic Manipulators, *The International Journal of Robotics Research*, **6**, (1982), pp. 123-147.

[8] Parks, P. C. and Militzer, J., A comparison of five algorithms the training of CMAC memories for learning control systems, *Automatica*, **28**, (1992), pp. 1027-1035.

[9] Qin, Z. and Lawry, J., A tree-Structured Classification Model Based on Label Semantics, *Proc. of the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based System* (IPMU), (2004), pp. 261-268.

[10] Qin, Z. and Lawry, J., Decision tree learning with fuzzy labels, *Information Sciences*, **172**, (2005), pp. 91-129.

[11] Quinlan, J. R., Induction of Decision Trees, *Machine Learning*, **1**, (1986), pp. 81-196.

[12] Sayil, S. and Lee, K. Y., A Hybrid Maximum Error Algorithm with Neiborhood Training for CMAC, IEEE, (2002).

[13] Takefuji, Y. and Szu H., Design of parallel distributed Cauchy machines [J], in *Proc of IJCNN Internal Joint Conference on Neural Networks*, (1989), pp. 529-532.

[14] Tang, Y. and Zheng, J., Linguistic Modeling Based on Semantic Similarity Relation among Linguistic Labels, *Fuzzy Sets and Systems*, **157**, (2006), pp. 1662-1673.

[15] Van-Nam, H., Nakamori, Y., Ho, T. and Murai, T., Multi-Attribute Decision Making Under Uncertainty: The evidence Reasoning Approach Revisited, *IEEE Transactions on Systems, Man and Cybernetics: Part A: System and Humans*, **36(4)**, (2006), pp. 804-822.

[16] Wang, X.-Z. and Chen, J.-F., Multiple Neural Networks Fusion Model Based on Choquet Fuzzy Integral, in *Proc. of the Third International Conference on Machine Learning and Cybernetics*, Shanghai, China, (2004), pp. 2024-2027.

[17] Williamson, T., *Vagueness*, Routledge, (1994)

[18] Yang, J. B. and Wang, Y. M. and Xu, D. L. and Chin, K. S., The evidential Reasoning Approach to MADA under both Probabilistic and Fuzzy Uncertainty, *European Journal of Operational Research*, **171(4)**, (2006), pp. 309-343.

[19] Zadeh, L. A., Fuzzy logic=computing with words, *IEEE Transaction on Fuzzy systems*, **4(2)**, (1996), pp. 103-111.

[20] Zadeh, L. A., The Concept of Linguistic Variables and its Application to Approximate Reasoning, Part 1, *Information Science*, **8**, (1975), pp. 199-249.