Optimal Cascade Linguistic Attribute Hierarchies for Decision Making

Hongmei He *Member IAENG* and Jonathan Lawry *

Abstract— **A hierarchical approach, in which a high-dimensional model is decomposed into series of low-dimensional sub-models connected in cascade, has been shown to be an effective way to overcome the 'curse of dimensionality' problem. Information propagation through a cascade hierarchy of Linguistic Decision Trees (LDTs) based on label semantics forms a process of cascade decision making. In order to examine how a cascade hierarchy of LDTs works compared with a single LDT for multiple attribute decision making, we developed genetic algorithm with linguistic ID3 in wrapper to find optimal cascade hierarchies. Experiments have been carried out on the two benchmark databases, Pima Diabetes and Wisconsin Breast Cancer databases from the UCI Machine Learning Repository. It is shown that an optimal cascade hierarchy of LDTs has better performance than a single LDT. The use of attribute hierarchies also greatly reduces the number of rules when the relationship between a goal variable and input attributes is highly uncertain and nonlinear.**

Keywords: cascade linguistic attribute hierarchy, cascade decision making, Genetic algorithm in wrapper, Linguistic ID3

1 Introduction

For multiple attribute decision making, the underlying relationship between attributes and the classification or decision 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. One of the main drawbacks to fuzzy modeling of systems is known as the 'curse of dimensionality', which is the exponential growth in the number of possible fuzzy rules as a function of the dimension of model input space. A hierarchical approach in which the original high-dimensional model is decomposed into series of low-dimensional sub-models connected in cascade, has been shown to be an effective way to overcome this problem since it provides a linear growth in the number of rules and parameters as the input dimension increases [12]. Campello and Amaral presented a unilateral transformation that converts the proposed hierarchical model into a mathematically equivalent non-hierarchical one [2]. As a result of the uncertainty and non-linear relationship between different attributes and a goal variable, different cascade hierarchies will have different performance on decision making procedures. We have proposed

a general multiple attribute hierarchy embedded with Linguistic Decision Trees (LDTs) based on Label Semantics [7]. In this paper, we propose a cascade hierarchy approach embedded with LDTs representing transparent rules, and describe the process of information propagation through a cascade hierarchy. We then develop a genetic algorithm with the Linguistic ID3 (LID3) [9] algorithm in wrapper to optimise cascade hierarchies. The experiments are performed on benchmark databases from the UCI Machine Learning Repository.

2 Label Semantics

Label semantics [5, 6] 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 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 $m_x : 2^{\mathcal{L}} \to [0,1]$ on subsets of labels. The relationship between these two measures will be described below.

Unlike linguistic variables [14], 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

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

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* $m_x : 2^{\mathcal{L}} \to [0, 1]$ *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 $\lambda : LE \to 2^{2^{\mathcal{L}}}$ 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 $\lambda : LE \to 2^{\mathcal{F}}$ 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 we define the appropriateness measure 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 $\mu_{l \wedge \neg m}(x) = \sum_{F : l \in F, m \notin F} m_x(F) = m_x(\{l\})$.

3 A cascade linguistic attribute hierarchy

3.1 Definition of a cascade hierarchy

The process of aggregation of evidence in multi-attribute decision problems based on attributes $x_1, ..., x_n$ can be viewed as a functional mapping between a high level variable y and input attributes, $y = f(x_1, ..., x_n)$, which is often dynamic and nonlinear, and may be imprecisely defined. In some cases, the function f may be approximated by a composition of lower dimensional sub-functions, forming a cascade hierarchy (a binary tree). Each sub-function represents a new intermediate attribute. Figure 1 shows a simple cascade hierarchy. There are $n - 1$ intermediate attributes produced. The last intermediate attribute z_{n-1} corresponds to the goal variable y. The cascade relationship is expressed as following:

$$z_i = \begin{cases} F_1(x_1, x_2) & i = 1, \\ F_i(z_{i-1}, x_{i+1}) & n > i > 1. \end{cases} \qquad (2)$$

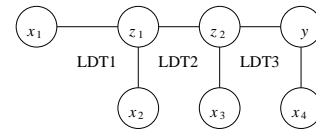As proposed in [7], in a linguistic attribute hierarchy, func-

Figure 1: A cascade hierarchy of LDTs

tion mappings between parent and child attribute nodes are defined in terms of weighted linguistic rules which explicitly model both the uncertainty and vagueness which often characterises our knowledge of such aggregation functions. These rules will be defined as conditional expressions in the label semantics framework [6] weighted by conditional probabilities. For each attribute, a set of labels and subsequent label expressions is defined. We assume that expressions describing a parent attribute can be (imprecisely) defined in terms of a description of its children. Let $\mathcal{L}_i$, θ_i and $\mathcal{F}_i$ denote the set of labels, a label expression and focal sets respectively, defined for attribute x_i for $i = 1, ..., n$. Similarly, let $\mathcal{L}_y$, θ_y and $\mathcal{F}_y$ denote the label set, a label expression and focal set for describing the goal variable y, respectively.

More precisely, the weighted conditional rules can take the form of an LDT. In an LDT, the nodes are attributes, and the edges are label expressions describing each attribute. The depth of an LDT with two input attributes is at most 2. A branch B is a conjunction of expressions $\theta_1 \wedge \theta_2$, where θ_1 and θ_2 are the label expressions of the two edges on the branch B, respectively. Each branch also is augmented by a set of conditional mass values $m(F|B)=P(C_x = F|B)$, for each output focal element $F \in \mathcal{F}_y$. Then the rules corresponding to the branch B would be: $\theta_1 \wedge \theta_2 \to F : m(F|B)$ for each focal element $F \in \mathcal{F}_y$.

3.2 Cascade decision making

Figure 1 shows the process of bottom-up information propagation through the cascade hierarchy. The only information available regarding the mappings F_1, F_2 and F_3 is in the form of decision trees LDT_1, LDT_2 and LDT_3, which define mapping functions for z_1 in terms of those for x_1 and x_2, for z_2 in terms of those for z_1 and x_3, and for y in terms of those for z_2 and x_4.

However, it is not easy to define the labels for intermediate attributes in terms of their children, as the intermediate attributes are not directly related to basic attributes in the system [2]. Therefore, we suppose all intermediate attributes are approximations of the decision variable y with the same domain and description labels. According to Jeffrey's rule [4], given an LDT, the mass assignment of the decision variable can be calculated by:

$$m_{z_i}(F_y) = \sum_{j=1}^{t_i} \mu_{\theta_1}(z_{i-1}) * \mu_{\theta_2}(x_{i+1}) * m(F_y|B_{i_j}), \quad (3)$$

where, B_{i_j} is the jth branch in the ith LDT, and $\mu_\theta(x)$ is appropriateness measure, quantifying the degree of our belief that label expression θ is appropriate for x [6]. The appropriateness measure can be calculated with mass assignments of attribute x according to Definition 2.5.

Information is propagated along the cascade LDTs from low level to high level. For the example in Figure 1, given mass functions m_{x_1}, m_{x_2}, m_{x_3}, and m_{x_4}, the mass function m_{z_1} is determined by propagating m_{x_1} and m_{x_2} through LDT_1, m_{z_2} is determined by propagating m_{z_1} and m_{x_3} through LDT_2, and finally, m_y is determined by propagating m_{z_2} and m_{x_4} through decision tree LDT_3 (see Figure 2). Here we con-
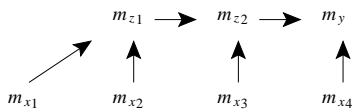

Figure 2: The cascade upwards information propagation

sider only classification problems where the goal variable y belongs to the finite set of classes $\{C_1, ..., C_t\}$. In this case, $\mathcal{F} = \{\{C_1\}, ..., \{C_t\}\}$, and for input vector $\vec{x}$, $m_y(\{C_i\}) = P(C_i|\vec{x})$.

In order to obtain a linguistic attribute hierarchy based on the LDTs, we need to train all cascade LDTs in turns from bottom to top. LID3 [9], an extension of well-known ID3 algorithm [11], is used to build the linguistic decision trees based on a given linguistic database.

4 GA in wrapper to optimise cascade hierarchies

4.1 Chromosomes and Reproduction

To learn a linguistic cascade hierarchy, we use a genetic algorithm as a search agent with the LID3 as an induction algorithm in wrapper. For the optimisation of cascade hierarchies with n attributes, the size of whole search space is $\frac{n!}{2}$. The performance of different hierarchies is judged on the basis of the accuracy for the given classification task.

Chromosomes: The purpose of a GA is to evolve a population of potential solutions each corresponding to the cascade hierarchies in a multiple-attribute space. Therefore, the GA in wrapper approach conducts a search in the space of possible cascade hierarchies. Different attribute orderings define different cascade hierarchies. So we define any possible permutation of all attributes, $\pi = \{x_1, ...x_n\}$, and $\pi \rightarrow \mathcal{H}$ as a genome of the genetic algorithm.

Reproduction: We use "*roulette-wheel*" selection, according to which, an individual with better fitness has higher probability of being selected. The probability that hierarchy $\mathcal{H}_i$ is selected is given by the nominalised fitness:

$$p_i = \frac{f_i(\mathcal{H}_i)}{\sum_{j=1}^{\Gamma} f_j(\mathcal{H}_j)}. \quad (4)$$

A one-elitism strategy is included since it keeps the current best individual in the next generation, and speeds up the convergence of the evolution process. On the other hand, in order to keep the diversity of solutions, a random hierarchy is generated in each generation.

We use two-point order crossover as follows (Figure 3): two parental permutations, π_1 and π_2, are chosen randomly depending on the probability chosen in 4. A continuous interval of the permutation π_1 is chosen, and also an interval starting at the same position and of the same length from π_2. The two parameters, '*starting position*' and '*length of interval*', are produced randomly. Two new permutations, π_1' and π_2', are created such that π_1' contains the interval from π_2 with the rest being the other elements of π_1 in the same order as they appeared in π_1. π_2' contains the interval from π_1 with the rest being the other elements of π_2 in the order as they were in π_2 (Figure 3). Mutation, which is the swapping of two randomly picked elements of a permutation, is carried out with some probability (m_rate) on each child in the population.

4.2 Evaluation and Termination Criteria

Here we only consider binary classification problem with two classes '+' and '-'. First, we investigate the ordinary accuracy on a threshold, which is the ratio of the number of correct classifications to the number of testing samples. When the estimated probability $p(C|\vec{x})$ (equivalent to $m_y(\{C\})$) that a sample with measurement vector $\vec{x}$ belongs to class C is larger than a threshold α, then that sample is classified as C. Con-
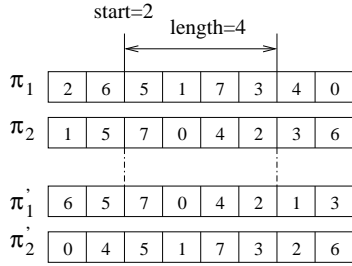
Figure 3: Two-point order crossover

ventionally we use 0.5 as a threshold. Here we consider two measures of accuracy, integrated accuracy and the area under ROC curve, which measures how well the classifier separates the two classes without reference to a decision threshold. The closer the ROC plot is to the upper left corner, the higher the ordinary accuracy of the test results.

For each possible threshold α for discriminating between the two classes, some positive cases will be correctly classified as positive (TP_α=number of True Positive), but some positive cases will be estimated as negative (FN_α=number of False Negative). On the other hand, some negative cases will be correctly classified as negative (TN_α=number of True Negative), but some negative cases will be classified as positive (FP_α=number of False Positive).

Accuracy: For a decision maker, the *Ordinary Accuracy* ($\mathcal{A}_\alpha$) over a threshold α can be calculated as below:

$$\mathcal{A}_\alpha(\mathcal{H}) = \frac{TP_\alpha + TN_\alpha}{\mathcal{M}}, \qquad (5)$$

where, $\mathcal{M}$ is the number of test examples. In order to reduce the sensitivity to the threshold α, we define the integrated accuracy to be the integration of accuracies for all $\alpha \in [0.5, 1)$ (Formula (6)):

$$\mathcal{A}_{\tilde{\alpha}}(\mathcal{H}) = \int_{0.5}^{1} \frac{\mathcal{N}_\alpha}{\mathcal{M}} d\alpha \approx \frac{\Delta(\alpha)}{\mathcal{M}} \sum_{i=1}^{m} \mathcal{N}_{\alpha_i}, \qquad (6)$$

where, the interval [0.5, 1) is divided into m subintervals with constant step length $\Delta(\alpha)$, and where $\mathcal{N}_{\alpha_i} = TP_{\alpha_i} + TN_{\alpha_i}$.

ROC curve: Receiver Operating Characteristic (ROC) analysis originated from signal detection theory and has been introduced to machine learning in recent years in order to evaluate algorithm performance in an imprecise environment. It is claimed [10] that ROC graphs can offer a more robust framework for evaluating classifier performance than traditional accuracy measure. The true positive rate is calculated with $\eta = \frac{TP}{P}$. The false positive rate is calculated with $\sigma = \frac{FP}{N}$. In a ROC curve, the true positive rate (η) is plotted as a function of the false positive rate (σ) for varing thresholds. Each point on a ROC plot represents a (η, σ) pair corresponding to a particular decision threshold.

Similarly, the integrated accuracy can be defined as the area under ROC curve, which measures how well the decision maker separates the two classes without reference to a decision threshold, as follows:

$$\mathcal{A}_{ROC}(\mathcal{H}) = \int_0^1 \eta d\sigma \qquad (7)$$

Let $p(+|\vec{x})$ be the estimation of the probability that an instance with measurement vector $\vec{x}$ is positive. If we rank test instances according to increasing positive probabilities, then the area under the ROC curve ($\mathcal{A}_{ROC}$) for a decision making problem with two classes +,- can be calculated [3] by:

$$\mathcal{A}_{ROC} = \frac{\sum_{i=1}^{P} r_i - P(P+1)/2}{PN}, \qquad (8)$$

where, P and N are the numbers of positive and negative samples, r_i is the rank of the ith positive instance in the rank list according to the probabilities of the positive class.

Termination criteria: Termination is an important parameter, which affects the running time and quality of solutions. Generally it heavily depends on the size of the chromosome. The maximum generations max_gen is linear function of the number of basic attributes. The evolution procedure will be repeated until the maximum number of generations is reached.

5 Experiments and Evaluation

All attributes are discretised using an entropy-based approach into three labels ($\mathcal{L} = \{small, medium, large\}$), respectively. Each label corresponds to a trapezoidal fuzzy set, which has 50% overlapping with neighbouring label fuzzy sets. A missing value of an attribute in an instance of the training database is replaced with the mean value of the attribute for the corresponding class.

The experiments are carried out using ten-fold cross validation. Data is split into 10 approximate equal partitions. Each one is used in turn for testing while the remainder is used for training i.e. 9/10 of data is used for training and 1/10 for testing. The whole procedure is repeated 10 times.

A trained hierarchy is evaluated using two types of accuracy measure described in Section 4.2. The ordinary accuracy is evaluated at threshold 0.5. The area under a ROC curve is calculated with Formula (8).

We examine the quality of cascade decision making and the cost of a hierarchy, i.e. the total number of branches from all decision trees in a cascade hierarchy, and compare the performance with that of a single LDT providing a direct mapping between input attributes and a classification variable.

5.1 On the Pima Diabetes database

The Pima database: The Pima Indian data set is a well-known benchmark problem from the UCI repository [1]. The problem relates to incidents of Diabetes mellitus in the Pima

Indian population living near Phoenix Arizona. The target attribute is a binary valued decision variable indicating whether or not the patient shows signs of Diabetes according to World Health Organisation criteria. The database of Diabetes includes 768 samples, in which, 268 positive instances (with Diabetes), 500 instances without Diabetes. There are 8 basic attributes.

Solutions and Fitness values: The two orders of attributes corresponding to the optimal cascade hierarchies ($\mathcal{H}_1$ and $\mathcal{H}_2$) obtained by the GAW with fitness values evaluated by $\mathcal{A}_a$ and $\mathcal{A}_{ROC}$ respectively, are: $\mathcal{H}_1$: 3, 4, 2, 5, 6, 7, 0, 1; $\mathcal{H}_2$: 2, 4, 0, 6, 3, 5, 7, 1. Table 1 lists the accuracies at threshold 0.5 ($\mathcal{A}_a$), the integrated accuracies ($\mathcal{A}_{\tilde{a}}$), the areas under ROC curves ($\mathcal{A}_{ROC}$) and the numbers of branches (β) for $\mathcal{H}_1$, $\mathcal{H}_2$ and the single LDT. It can be seen that $\mathcal{H}_1$ and $\mathcal{H}_2$ achieve similar performance in $\mathcal{A}_a$, $\mathcal{A}_{\tilde{a}}$ and $\mathcal{A}_{ROC}$. Their performance in $\mathcal{A}_a$ and $\mathcal{A}_{ROC}$ is better than that of a single LDT, while the single LDT has higher integrated accuracy than $\mathcal{H}_1$ and $\mathcal{H}_2$. The branch numbers for $\mathcal{H}_1$ and $\mathcal{H}_2$ are much less than for the single LDT.

Table 1: Evaluations of hierarchies obtained by GAW on the Pima database

$\mathcal{H}$	$\mathcal{A}_a$	$\mathcal{A}_{\tilde{a}}$	$\mathcal{A}_{ROC}$	β
$\mathcal{H}_1$	0.747396	0.188281	0.783776	115
$\mathcal{H}_2$	0.748698	0.189437	0.790649	115
LDT	0.713542	0.244922	0.769687	14845

Accuracy and ROC curves: Figure 4 (a) and (b) show the accuracy and ROC curves for the two hierarchies and the single LDT, respectively. From the accuracy curves in Figure 4 (a), it can be seen that $\mathcal{H}_1$ and $\mathcal{H}_2$ obtain approximately the same accuracy curves, and achieve higher ordinary accuracies at threshold 0.5 than the single LDT does. But the accuracies obtained by $\mathcal{H}_1$ and $\mathcal{H}_2$ decrease as thresholds increase, and become smaller than for the single LDT when thresholds are over 0.65. Figure 4 (b) shows that the two optimal cascade hierarchies obtain similar ROC curves to the single LDT, although they have different performance in accuracies.

5.2 On the Wisconsin Breast Cancer Database

The Wisconsin Breast Cancer Database: The Wisconsin Breast Cancer (WBC) database [1] was created by Dr. William H. Wolberg from the University of Wisconsin Hospitals, Madison [8]. There are 699 samples, in which 458 samples are Benign, and 241 samples are Malignant. There are nine basic attributes, and each attribute is with lower bound 1 and upper bound 10. There are 16 instances that contain a single missing (i.e., unavailable) attribute value. It is claimed that the best result is 93.7% trained on 200 instances and tested on the other 169 in the first group of 369 samples with the 1-nearest neighbor approach in [1].

Solutions and Fitness values: The two permutations of at-tributes corresponding to the two optimal cascade hierarchies are: $\mathcal{H}_3$:6,2,4,3,8,7,5,1,0; $\mathcal{H}_4$:6,4,3,8,1,7,5,2,0. Table 2 lists the accuracies at threshold 0.5 ($\mathcal{A}_a$), the integrated accuracies ($\mathcal{A}_{\tilde{a}}$), and the areas under ROC curves ($\mathcal{A}_{ROC}$) and branch numbers (β) for $\mathcal{H}_3$, $\mathcal{H}_4$ and the single LDT. The experiment results show that $\mathcal{H}_3$ and $\mathcal{H}_4$ have similar performance in ordinary accuracies for different thresholds, and the areas under ROC curves. They have ordinary accuracies at threshold 0.5 better than a single LDT, but they lose performance in the integrated accuracy. The best ordinary accuracy at threshold 0.5 is 96.7% obtained by $\mathcal{H}_3$. Both algorithms for learning a single LDT and a cascade hierarchy have computational complexity $O(n\beta)$, where n is the length of a branch and β is the total number of branches. Table 2 shows that the number of branches for the optimal cascade hierarchies $\mathcal{H}_3$ and $\mathcal{H}_4$ are close to that for the single LDT. However, for each LDT in a cascade hierarchy, there are only two input attributes, thus the length of a branch is at most 2. Therefore, the optimal cascade hierarchies have better computational complexity than the single LDT.

Table 2: Evaluations of hierarchies obtained by GAW on the WBC database

| $\mathcal{H}$	$\mathcal{A}_a$	$\mathcal{A}_{\tilde{a}}$	$\mathcal{A}_{ROC}$	β
$\mathcal{H}_3$	0.967096	0.409156	0.985831	100
$\mathcal{H}_4$	0.962804	0.408530	0.985867	100
LDT	0.934192	0.441863	0.932976	97

Accuracy and ROC curves: Figure 5 (a) and (b) show the accuracy and ROC curves for the two optimal cascade hierarchies and the single LDT, respectively. From the accuracy curves in Figure 5 (a), it can be seen that the ordinary accuracy at threshold 0.5 of $\mathcal{H}_3$ and $\mathcal{H}_4$ is better than for the single LDT, but their ordinary accuracies when the threshold is larger than 0.6 are worse than for the single LDT. The ROC curves of $\mathcal{H}_3$ and $\mathcal{H}_4$ are slightly better than for the single LDT.

6 Conclusion

In this paper, the process of information propagation through a cascade hierarchy of Linguistic Decision Trees for multi-attribute decision problems is investigated. We develop a genetic algorithm with the training algorithm LID3 in wrapper to optimise cascade hierarchies for decision making, and examine the performance of the optimal cascade hierarchies on two benchmark databases, Pima Diabetes and Wisconsin Breast Cancer databases from UCI machine learning repository. The experimental results show that an optimal cascade hierarchy can achieve better performance in the ordinary accuracy at threshold 0.5 and in the area under ROC curves than a single LDT. The number of rules induced by the optimal cascade hierarchy is much lower than a single LDT, when the relationship between a class and the input attributes is highly uncertain and nonlinear. However, accuracy tends to decrease with higher thresholds.

References

[1] Asuncion, A. and Newman, D.J., UCI Machine Learning Repository [http://www.ics.uci.edu/ mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, (2007).

[2] Campello, R. J. G. B. and Amaral, W. C., Hierarchical Fuzzy Relational Models: Linguistic Interpretation and Universal Approximation, *IEEE Transaction on Fuzzy Systems*, **14(3)**, (2006), pp. 446-453.

[3] Hand, D. and Hill, R. J., A simple generalisation of the area under the ROC curve for multiple class classification problems,*Machiine Learning*, **45**, (2001), pp. 171-186.

[4] Jeffrey, R. C., The Logic of Decision, Gordon and Breach, New York, (1965).

[5] J. Lawry, A Framework for Linguistic Modelling, *Artificial Intelligence*,**155**, (2004), pp. 1-39.

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

[7] J. Lawry and H. He, Multi-Attribute Decision Making Based on Label Semantics, the International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 16(2) supp, (2008), pp. 69 - 86.

[8] Mangasarian, O. L. and Wolberg, W. H., Cancer diagnosis via linear programming, *SIAM News*, **23 (5)**, September, (1990), pp. 1-18.

[9] Qin, Z. and Lawry, J., Decision Tree Learning with Fuzzy Labels, *Information Sciences*, **172**, (2005), pp. 91-129.

[10] Qin, Z., ROC analysis for predictions made by probabilistic classifiers, in Proc. of the International Conference on Machine Learning and Cybernetics, 18-21 August 2005, **5**, (2005), pp. 3119- 3124.

[11] Quinlan, J. R., Induction of Decision Trees, *Machine Learning*, **1**, (1986), pp. 81-106.

[12] Raju, G.U. and Zhou, J. and Kiner, R. A., Hierarchical Fuzzy Control, Int. J. Control, **54:55**, (1991), pp. 1201-1216.

[13] Zweig, MH. and Campbell, G., Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinic medicine. Clinical Chemistry, **39**, (1993), pp. 561-577.

[14] L.A. Zadeh, (1975), The Concept of Linguistic Variable and its Application to Approximate Reasoning Part I, *Information Sciences*, Vol. 8, pp199-249, Part II, *Information Sciences* Vol. 8 pp. 301-357, Part III, *Information Sciences* 9 pp 43-80.
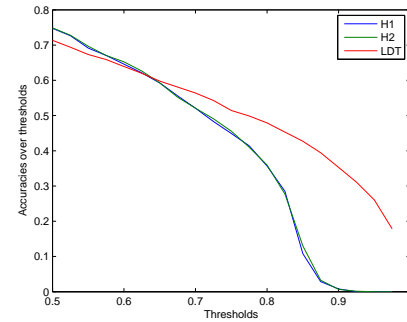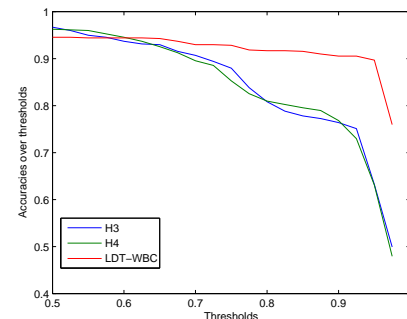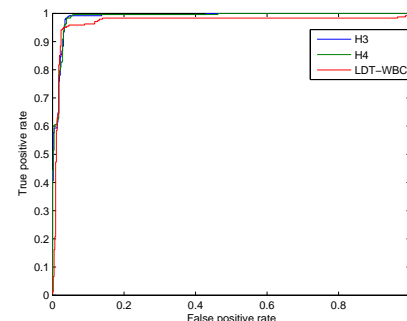
(a) Accuracies at threshold [0.5,1)
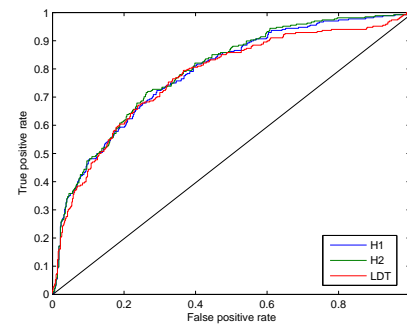
(b) ROC curves

Figure 4: Accuracy and ROC curve for $\mathcal{H}_1$, $\mathcal{H}_2$, and the single LDT on the Pima database

(a) Accuracies on threshold [0.5,1)

(b) ROC curves

Figure 5: Accuracy and ROC curve for $\mathcal{H}_3$, $\mathcal{H}_4$, and the single LDT on the WBC database