ISSN1000-9825 Journal of Software 软件学报 2000,11(4):453~460

A Hybrid Multi-Concept Acquisition System*

GAO Yang LIU Hai-tao ZHOU Zhi-hua CHEN Zhao-qian

(State Key Laboratory for Novel Software Technology Nanjing University Nanjing 210093)

E-mail: gaoy@aiake1.nju.edu.cn

Abstract In this paper, a hybrid multi-concept acquisition system HMCAS is proposed. HMCAS can perform incremental supervised learning on arbitrary sequences composed of analog or binary inputs. The kernel algorithm of HMCAS, named HMCAP, which integrates symbolic and neural learning based on the probability of instance space, has the ability of generating concept descriptions in the form of hybrid decision tree. The prototype system of HMCAS has been applied to the field of typhoon forecasting and achieved successful result.

Key words Machine learning, neural network, knowledge acquisition, hybrid model, decision tree.

In recent years, the research of the hybrid learning method that combines symbolic with neural learning has become one of the hotspots of machine learning. More and more researchers focus on this field, and fruitful results have been achieved.

Gallant proposed a Connectionist Expert System[1], which uses a large amount of instances to train a dependency network via Pocket algorithm, and regards the trained network as the knowledge base of an expert system. Peschl developed a method combining neural network with symbolic processing and applied it to rule application and robotics[2]. In his method, the trained neural network is regarded as a heuristic function that is necessary for symbolic processing. Towell and Shavlik proposed a method named KBANN[3], which combines explanation based learning with empirical learning. KBANN utilizes rough domain knowledge as heuristics to aid the construction of the neural network, and uses the trained neural network to refine domain knowledge. McMillan *et al.* proposed RuleNet[4], which exploits a variant of Backpropagation[5] to induce symbolic rules for string mapping field through setting constraints on the change of connective weights. It uses a network to find out the classification of symbols, and uses another network to generate corresponding rules. Behnke and Karayiannis proposed CNeT that combines LVQ[6] with m-ary tree[7]. It could solve the uncertainty problem associated with the representation of the feature vectors by creating a structured partition of the feature space.

---

* This research is supported by the National Natural Science Foundation of China (国家自然科学基金, No. 69875006). **GAO Yang** was born in 1972. He is a Ph. D. student at Department of Computer Science and Technology, Nanjing University. He received a B. S. degree in mechanics from Nanjing University of Science and Technology in 1996. His research interests are distributed artificial intellgence and machine learning. **LIU Hai-tao** was born in 1975. He is a Ph. D. student at Department of Computer Science and Technology, Nanjing University. He received a B. S. degree in computer science from Harbin Institute of Technology in 1998. His research interests are data mining and machine learning. **ZHOU Zhi-hua** was born in 1973. He is a Ph. D. student at Department of Computer Science and Technology, Nanjing University. He received a B. S. degree in computer science from Nanjing University in 1998. His research interests are machine learning, neural network and data mining. **CHEN Zhao-qian** was born in 1940. She is a professor and doctoral supervisor of the Department of Computer Science and Technology, Nanjing University. Her current research areas include machine learning, knowledge engineering and neural network.

Manuscript received 1998-10-22, accepted 2000-02-29.

Above brief review incompletely demonstrates the achievements attained in hybrid learning field in the last two decades. In this paper, a hybrid learning system named HMCAS is proposed. The kernel of HMCAS is the hybrid multi-concept acquisition algorithm HMCAP[8], which combines decision tree with neural networks. Given both a concept set and an instance set where each instance belongs to a certain concept and is expressed as a string composed of discrete attributes and continuous attributes, HMCAS can generate the description for each concept. This system has many characteristics: it is a multi-concept acquisition system; the symbolic processing is coupled with the neural network, so that it could deal with continuous attributes as well as discrete attributes.

The rest of this paper is organized as follows. In Section 1, we introduce the hybrid learning algorithm HMCAP. In Section 2, we describe the system architecture of HMCAS. In Section 3, we apply the prototype system of HMCAS to the field of typhoon forecasting. Finally in Section 4, we conclude the paper.

1 Hybrid Learning Algorithm HMCAP

1.1 Hybrid learning strategy

The hybrid multi-concept acquisition algorithm HMCAP is designed to generate concept descriptions from training examples composed of attribute set. Given some instances, it will generate a hybrid decision tree that embeds specific tree leaves in neural networks. After all the leaves are trained, the hybrid tree can make prediction.

The input of HMCAP is a set of instances where each instance is described in the form of concept-attribute value. Each attribute is defined by attribute name and value's range. The entire attributes compose the attribute set, which may involve not only discrete attributes but also continuous ones. On the contrary, most inductive learning algorithms based on attribute description, such as ID3[9], can only deal with discrete attributes. And although some algorithms, such as C4.5[10], can deal with continuous attributes as well as discrete ones, they must perform discretization before utilizing the continuous attributes. So, HMCAP is superior to them in this aspect.

The knowledge representation of HMCAP is based on the attribute value. This style of representation can make the relation between symbolic learning and neural network closer, and decrease the complexity of symbolic inductive process. The reason is that the attributes of instances can be easily transformed into input-output patterns, which are needed by neural network training.

Before training the NN nodes, it is necessary to transform the learning instances into input-output patterns for neural network. We manage to do so by mapping the attribute values into real values within [0,1] by Sigmoid function:

$$f(x)=\frac{1}{1+e^{-(\frac{x-b}{a})}}, \tag{1}$$

where a is the sensitivity of the specific attribute, b is the median of the attribute values. a and b are different for each attribute. In order to get the outputs of the NN nodes, we assume that each output node exports either 0 or 1. If a node's output is less than 0.5, we take it as 0. Similarly, if the output is greater than or equal to 0.5, we take it as 1. Therefore, we can express the outputs of all nodes in binary code, where each bit represents one concept. In this way, the number of nodes in the neural network is less than that of the concepts, so that the network complexity is reduced.

1.2 Search for the best attribute

In order to generate the decision tree at the least cost, a choice among the combinations of the discrete attributes with concept $(\langle a,v\rangle,c)$ is offered by HMCAP for guiding the inductive process. a is a discrete attribute, v is one of the values of attribute a, and c is a concept. This search process is called Best-State (BS). The best attribute value $\langle a,v\rangle$ covers the most instances of concept t_c, and excludes the most instances of other concepts. The probability of the division of the instances is computed according to the following formula:

$$\frac{M_c^2 * N}{T^2 * M * (N+1)} \quad i=1,2,\ldots,n, \tag{2}$$

T is the total number of instances. M and N are respectively the number of instances that $\langle a,v\rangle$ covers and excludes. M_c is the amount of instances that subordinate to concept c in M.

1.3 Algorithm description

We give out two concepts before describing the HMCAP algorithm, that is, the Inductive-Precision and the Pruning-Factor. The purpose of introducing these two concepts is two-sided. First, they are helpful to eliminate the influence of noisy data during learning. Second, they are useful in controlling the depth of the decision tree.

Definition 1. Assume S is the expression set of the attributes from the root to the current node when the decision tree is divided to node T_k. The Inductive-Precision of S for concept t_i is defined as:

$$Pr_i(s)=(|P'_{k,i}|+|NP'_{k,i}|)/N \quad i=1,2,\ldots,n, \tag{3}$$

N is the amount of instances, $|P'_{k,i}|$ is the amount of instances in instance set $P_{k,i}$ that subordinate to concept t_i. $|NP'_{k,i}|$ is the amount of instances that do not subordinate to concept t_i.

Equation (3) can also be expressed as below:

$$Pr_i(s)=1-\left(\sum_{i=1}^{n}Pc_i+C_i+C'_i\right)/N \quad i=1,2,\ldots,n, \tag{4}$$

Pc_i indicates how many instances have been abandoned on each node. C_i is the amount of instances that subordinate to concept t_i but have not been included during dividing node T_k. For concept t_i, if S's Inductive-Precision is greater than the given value P_r, S can be considered as the general description of concept t_i.

Definition 2. Assume $D_{k,i}$ is the Pruning-Factor on node T_k for concept t_i. The value of $D_{k,i}$ is computed according to the following equation:

$$D_{k,i}=1\ \left(\sum_{i=1}^{n}Pc_i+C_i\right)/N \quad i=1,2,\ldots,n, \tag{5}$$

Then the Pruning-Factor of T_k is computed as below:

$$D_k=\min(D_{k,i}) \quad i=1,2,\ldots,n, \tag{6}$$

The HMCAP algorithm is described as a recursion procedure, whose pseu-code is as below:

Tree(node, P_k, NominalAttrs, P_c)

/* node is the object of decision tree's node; NominalAttrs is an array of discrete attribute objects; P_k is the array of instance set objects, each element of P_k is a separate instance set subordinating to a concept; P_c is an array of integers, it is a global variable and represents the number of instances discarded from the root to the current node (not including the neural network nodes). */

[Initialization]

NominalAttrs = discrete attribute set;
P_k = instance set of all concepts;
P_r = learning precision; /* P_r is a constant
node.condition = NULL;
$P_c[i]=0$; $i=1,2,\ldots,n$;
Tree(node, P_k, NominalAttrs, P_c);

[Step 1]

if (the current node only covers the instance set of one concept)
then
$P_k[i]$.examples ≠ null;

```
        node.type=leaf;                    /* the current node doesn't need division
        node.conceptIndex=P_k[i].conceptIndex;
        return;
    else goto Step 2;
[Step 2]
    if (NominalAttrs is empty)
    then
        node.type=NN;
          /* It is needed to use NN to continue learning for having no discrete attribute values */
        node.NN=create neural network;
        Transform P_k to learning pattern set of neural networks;
        node.NN.StartTrain();
        return;
    else goto Step 3;
[Step 3]
    call BS, get the best discrete attribute value (<a,v>,c);
    node.type=nonleaf;          /* the current node is not a leaf
    P_{2k}[i].examples={x:x∈P_k[i].examples, and <a,v>∈x}, i=1,2,...,n;
    P_{2k+1}[i].examples={x:x∈P_k[i].examples, and <a,v>∉x}, i=1,2,...,n;
    node.leftChild=allocate node from memory;
    node.rightChild=allocate node from memory;
    node.leftChild.condition=<a,v,true>;
    node.rightChild.condition=<a,v,false>;
    goto Step 4;
[Step 4]
    Pr_k[c]=1-( Σ_{i=1}^{n} P_c[i]-P_{2k+1}[c].instanceNumber+ Σ_{j=1, j≠i}^{n} (P_{2k}[j].instanceNumber) )/N;
                                   /* compute the Inductive-Precision
if (Pr_k[c]≥Pr)
then
    node.leftChild.type=leaf;          /* the left subnode doesn't need division
    node.leftChild.conceptIndex=P_{2k}[c].conceptIndex;
                                 /* the left subnode subordinates to concept c
    P_c[i]=P_c[i]+P_{2k}[i].instanceNumber, i=1,2,...,n and i≠c;
else
    node.leftChild.type=nonleaf;
    set NominalAttrs_{2k}=NominalAttrs;
    delete NominalAttrs_{2k}[a] from NominalAttrs_{2k};
    Tree(node.leftChild, P_{2k}, Nominal-Attrs_{2k}, P_c);
    goto Step 5;
[Step 5]
D_k[i]=1-(TotalP_c+C[i])/N, i=1,2,...,n;          /* compute the Pruning-Factor
if (min(D_k[i])≥Pr)
then
    delete node.rightChild;                    /* the right sub-tree can be deleted
        P_c[i]=P_c[i]+P_{2k-1}[i].instanceNumber, i=1,2,...,n;
else
    node.rightChild.type=nonleaf;
    set NominalAttrs_{2k-1}=NominalAttrs;
    delete NominalAttrs_{2k-1}[a].list[v] from NominalAttrs_{2k-1}[i].list;
    if (there is only one element in NominalAttrs_{2k-1}[i].list)
    then
        delete NominalAttrs_{2k-1}[a] from NominalAttrs_{2k-1};
        Tree(node.rightChild, P_{2k-1}, NominalAttrs_{2k-1}, P_c);
        return;
```

The predictive mechanism of HMCAP is based on the characteristic of hybrid decision tree. The process is performed as follows. First, the input attribute values are compared with the partition conditions from the tree root to the leaves. The result of the current comparison decides which node to be compared next. Then, if arrives at a leaf node, the instance is labeled with the concept of that leaf. Or else the instance is transferred into a neural network input-output pattern. After the determination of the neural network, the instance gets its classification.

2 Architecture of HMCAS

The HMCAS system manages the background knowledge and the rules obtained through learning by Database Management System. KnowledgeBase and RuleBase are considered and defined as in Fig. 1, in a way that allows users to add, update and delete the background knowledge, which includes concept set, attribute set and instance set, and controls the description of each concept. Therefore, it is convenient for users to add new instances to implement incremental learning and extend the system. When it is necessary to refine or optimize the concepts' descriptions, or to infer with the gained descriptions, the system gets data from the RuleBase. When users manipulate the data, the Knowledge Acquisition module and KnowledgeBase can check the validation of the new data and maintain the integrity of the database.

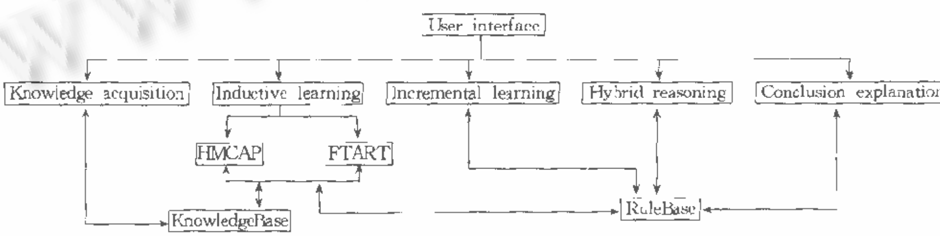

Fig. 1 Architecture of HMCAS

Inductive learning is the kernel of the system. It employs symbolic method and neural network algorithm to obtain the descriptions of the concepts represented by a hybrid decision tree. The symbolic learning part exploits HMCAP algorithm to get the descriptions and generate the optimized decision tree in which each NN node is related to a neural network. The NN nodes are trained by the NN learning of the system. Users can choose either FTART, which is a field theory[11] based adaptive resonance algorithm[12], or standard BP[5] to implement NN learning. Within a given accuracy, the NN nodes can get correct outputs of the networks.

Inductive learning module gets data as training instances from KnowledgeBase and saves the result as rules in RuleBase. When the description of a concept cannot explain some instances correctly, or it can be changed by some instances, the user is required to add instances to KnowledgeBase module and make the description more accurate through utilizing Incremental Learning module.

After a training cycle, Hybrid Reasoning module can generate classifications for unknown instances. During the inference process, the given instance is matched with the partition conditions of the decision tree. The result of the previous match decides the path of the next match. This process will not stop until it arrives at a leaf or a neural network node. If the end is a common leaf, the instance is labeled as the leaf's class. Or else the neural network module is started to produce corresponding classification.

3 Simulation: Typhoon Forecasting

Typhoon is an important weather phenomenon that often influences Jiangsu Province of China. At present the observatory forecasts typhoon with some experience of experts, and the forecasting is performed manually. Applying HMCAS to develop a practical system in typhoon forecast field could increase the forecasting accuracy and result in great economical reward.

The data we used to train HMCAS were provided by Jiangsu Observatory, P.R. China. There are 7,251 instances in the training set. Each instance has 12 attributes and a concept label. Table 1 shows the background knowledge of this task.

Table 1 Background knowledge of typhoon forecasting

Name	Value Range
Type	High pressure inshore, subtropical pressure, west wind chamfer, no chamfer
South pressure greater than north	no, yes
588 line west ridge longitude	west of 116E, 116E～120E, 120E～127E, east of 127E
West wind chamfer beyond 35N	None, 104E～120E, out of 104E～120E
Central value of the nearest H-P ring	>5920 gpm, ≤5920 gpm
Latitude span from center to eastern 588 line	0～13
Central latitude change in 24 hours	0～40
Latitude span from center to subtropical ridge	0～15
Subtropical pressure center latitude	0～50
Subtropical pressure center longitude	90～179
Longitude begin	113.5～123
Latitude begin	21.5～33
North bound of subtropical pressure 588 line	22～34.1

From Table 1 we can see that there are 4 discrete attributes and 8 continuous attributes, and the classification label is the type of typhoon. Figure 2 shows the hybrid decision tree generated by HMCAS.

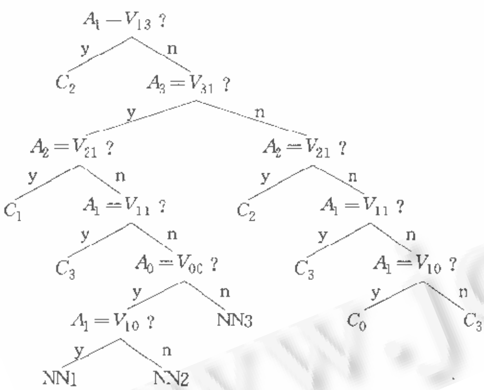

Fig. 2 Hybrid decision tree generated by HMCAS

There are respectively 5,2,3 instances falling into the three neural network nodes in Fig. 2. For the purpose of comparing the function of FTART, we also provide standard BP[5] to be used at the same time. Figure 3 shows the final structure of NN1.

As a comparison, we also apply C4.5[10] to this task. The reason we did't use other hybrid learning methods for comparison is that nearly all those algorithms are not fit for this field. For example, KBANN[3] needs domain knowledge to aid the construction of the neural networks, but there are no such symbolic rules available. RuleNet[4] could deal with continuous attributes, but it is designed for string mapping field that has characteristics different from typhoon forecasting.

After the training process, we use another instance set composed of 1,200 instances to test those models. The testing results are shown in Table 2. It is obvious that HMCAS achieves higher testing set accuracy than C4.5 whatever neural network is adopted. When employing FTART, the training time of HMCAS is very close to that of C4.5, and the training set accuracy is higher than that of C4.5 as well as the testing set accuracy. So, HMCAS with FTART is a good choice at least in the typhoon forecasting field.

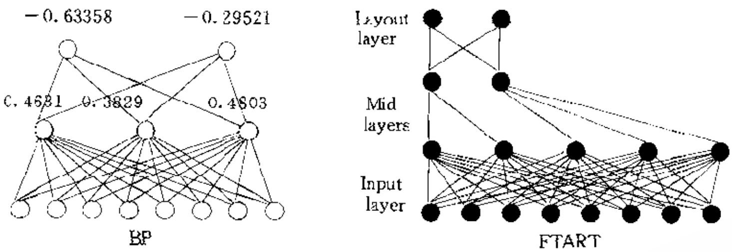

Fig. 3 The structure of NN1

Table 2 Comparison of testing results

	Model	Training time(sec)	Training set accuracy(%)	Testing set accuracy(%)
	C4.5	2.7	98.5	78.33
HMCAS	Employing FTART	3.4	100	86.92
	Employing BP	542	94.17	94.17

4 Conclusions

As an alternative to the symbolic methods and neural network methods, we suggested the use of hybrid learning. In this paper, we propose a hybrid multi-concept acquisition system named HMCAS, which combines symbolic induction with artificial neural network learning. Compared with previous inductive learning algorithms, the kernel algorithm of HMCAS has many characteristics. It could acquire multi-concept, and could deal with continuous attributes as well as discrete ones.

A neural network algorithm FTART could be embedded in HMCAS. FTART is a field theory based adaptive resonance algorithm, which could change the classifications generated during learning according to the probability of the samples in instance space. The reason is that FTART employs a unique way to resolve the conflict between the instances and dynamically extend the area of classifications. In our experiments, HMCAS employing FTART has got the best results. It achieves the highest accuracy through fast learning.

The conclusion we can draw from the experiment is that the hybrid learning method can be used in real world tasks, such as weather forecasting. In particular, this approach can compete quite well with other methods, such as solo symbolic methods or neural networks in domains where structural knowledge is relevant and where both discrete and continuous attributes exist.

Acknowledgements The authors wish to thank Mr. Sun Rong-jian and Mr. Yin Sheng for their fruitful work. Jiangsu Observatory, P. R. China, assisted this project. The National Natural Science Foundation of China and the Science Foundation of Jiangsu Province supported this research.

References

1 Gallant S I. Connectionist expert systems. Communications of the ACM, 1988,31(1):152～169

2 Peschl M F. Hybrid processing: a combination of 'Intelligent' symbol manipulation and parallel distributed processing. In: Gardin F, Mauri G eds. Computational Intelligence 2, North-Holland, BV: Elsevier Science Publisher, 1989. 69～76

3 Towell G G, Shavlik J W. Knowledge-based artificial neural networks. Artificial Intelligence, 1994,70(1,2):119～165

4 McMillan C, Mozer M C, Smolensky P. Dynamic conflict resolution in a connectionist rule-based system. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence. Morgan Kaufmann, Chamborg, France, 1993,1366～1371

5 Rumelhart D, Hinton G, Williams R. Learning representations by backpropagating errors. Nature, 1986,323(9):318～

362
6 Karayiannis N B. Learning vector quantization: a review. International Journal of Smart Engineering System Design, 1997,1:33～58
7 Behnke S, Karayiannis N B. Competitive neural trees for vector quantization. Neural Network World, 1996,6(3):263～277
8 Chen Zhao-qian, Liu Hong, Zhou Rong *et al*. A hybrid algorithm for multi-concept acquisition and its application. Chinese Journal of Computers, 1996,19(10):753～760
(陈兆乾,刘宏,周戎等.一种混合型多概念获取算法HMCAP及其应用.计算机学报,1996,19(10):753～760)
9 Quinlan J R. Induction of decision trees. Machine Learning, 1986,62(1):81～106
10 Quinlan J R. C4.5: Programs for Machine Learning. Morgan Kaufmann, Chamborg, France, 1993
11 Wasserman P D. Advanced Methods in Neural Computing. Van Nostrand Reinhold, New York, 1993
12 Chen Zhao-qian, Zhou Rong, Liu Hong *et al*. A new adaptive resonance theory algorithm. Journal of Software, 1996,7(8):458～465
(陈兆乾,周戎,刘宏等.一种新的自适应谐振算法.软件学报,1996,7(8):458～465)

一种混合型多概念获取系统

高阳 刘海涛 周志华 陈兆乾

(南京大学计算机软件新技术国家重点实验室 南京 210093)

摘要 文章实现混合型多概念获取系统HMCAS(hybrid muiti-concept acquisition system).无论在离散值或连续值输入下,HMCAS系统都可以实现增量式教师学习.HMCAS的核心算法HMCAP基于事例空间的概率分布,结合了符号学习和神经网络学习,能够以混合型判定树形式产生概念描述.HMCAS的原型系统已经成功应用于台风预测领域.

关键词 机器学习,神经网络,知识获取,混合模型,判定树.

中图法分类号 TP181