UNDERSTANDING RULE GENERATION SUPPORTING SYSTEM FOR DRAWING UNDERSTANDING USING INTERACTION WITH USER

Shin'ichi SATOH†and Masao SAKAUCHI‡

†National Center for Science Information Systems(NACSIS)
3-29-1 Otsuka, Bunkyo-ku, Tokyo 112, Japan
‡Institute of Industrial Science, The University of Tokyo
7-22-1 Roppongi, Minato-ku, Tokyo 106, Japan

Abstract

It is necessary to achieve efficient and powerful drawing understanding in order to realize large scale practical drawing image databases. Besides, a drawing understanding system is hard to build, so this can be a bottle-neck of realizing these databases. Thus automatic generation of drawing understanding system is desirable.

We have proposed a drawing understanding system using state transition models. This system performs understanding process complying with understanding rules which have declarative form to be handled easily by machines.

Here we present an example of drawing understanding system generation system using man-machine interaction. This system stores previous session with user and avoids repeating same query using computer learning method. In this paper, after brief introduction of the system is presented, embodiment of an experimental system and successful experiments are shown to reveal effectiveness of the system.

1 Introduction

We have proposed a drawing understanding system using state transition models[1, 2, 3]. This system performs understanding process complying with understanding rules. These rules have declarative form to be obtained relatively easily, and they can be handled easily also by machine. So, on realizing a generating system of drawing understanding systems, we use this framework for drawing understanding.

In our approach, instead of generating the understanding system itself, understanding rules for the understanding system are generated automatically. Thus whole system is constituted of two subsystems. One is a drawing understanding system using state transition models, and another is an understanding rule generation system. The understanding system is given drawing image and generates understanding results according to understanding rules. The rule generation system reads understanding results, refers to understanding rules and modifies them. Understanding results are transferred to the rule generation system at once, and this system modifies rules to make understanding results much more proper. This modification is reflected to the understanding system immediately.

The rule generation system uses man-machine cooperation. This interaction is performed with visual interface using graphical display and pointing device mouse. The system shows understanding results, given from the understanding system, graphically to user, and is given 'oracle' from user. Then the system infers user's intention from this oracle, generalizes it, and generates appropriate rules using inductive inference algorithm. This algorithm is based on Shapiro's Model Inference System[4] with extension to handle numerical data. After several sessions with user, the system stores many oracles into its internal database, and comes to be able to generate correct rules. The system's behavior deeply depends on user's oracles, but they are obtained with simple interface using mouse so user can make it with as few errors as possible.

This system is implemented and evaluated with practical map drawings, and as an additional experiment, with clustered sports scene images. These experiments proved successful results. In the paper, after explanation of the background, outline of this system is given, learning algorithm is explained briefly, and some experiments are described.

2 Outline of the system

Fig. 1 shows an outline structure of the understanding rule generation supporting system. The system is composed mainly of two parts. One is a state transition type drawing understanding system, and another is state transition rule learning system.

The drawing understanding system part is constructed based on state transition models[1, 2, 3]. We assume that drawings are composed of symbols, and an understanding process is realized as proper labeling to these symbols. In this model, each of symbols which are called tokens has its own internal state, inspects surroundings spontaneously, and makes transition of its state independently and successively. Final states of tokens express understanding results. This state transition is performed complying with state transition rules.

The rule learning system is the core part of the whole system. The system generates state transition rules (understanding rules) referring to the understanding results with man-machine interactions. First, the system shows the understanding results to the user graphically,

and wait for indication of adequacy of the results. If the understanding results contains erroneous results, the user indicates these mistaken points (misunderstood tokens) to the system. Following that, the system modifies state transition rules to make understanding results much more proper for the indication from user. At that time, the system stores these sessions with user to the example database, to avoid repeating same mistakes and interactions.

Thus the system repeats interactions with user, modifies erroneous understanding results with modifying understanding rules also, and gradually comes to generate correct understanding rules automatically.

As described in [1], the state transition type drawing understanding system perform understanding process as bottom-up process and top-down process. This system should be given understanding rules for both bottom-up process and top-down process, which are bottom-up rules and top-down rules respectively. The rule learning system has a restriction that it can generate only bottom-up rules. Besides that, the system cannot generate these rules without anything. It requires outlined rules to a certain extent, and accomplishes rule generation using them.

3 The rule learning system

3.1 General flow

As stated above, the rule learning system generates bottom-up rules for the state transition type drawing understanding system. Generally, bottom-up process of state transition models performs sequence of processes as "measurement of parameters" $\longrightarrow$ "inspection of situations" $\longrightarrow$ "state transition" $\longrightarrow$ "measurement of parameters"... This sequence is shown as fig. 2. Bottom-up rules which had been written for concrete drawing understanding systems also perform this sequence. On writing these bottom-up rules, though an outlined framework is determined easily, it is not so easy to adjust rules in detail. Assuming that bottom-up processes are made of sequence described above, we can think that the outlined framework of bottom-up process is constituted of that sequence without "inspection of situations", and the detailed adjustment of bottom-up process is adjustment of how to test parameters on "inspection of situations". It is desired to realize the supporting system for understanding rule generation that this detailed adjustment can be executed easily using interaction with user etc., even if we permit that the system is given the outlined framework of bottom-up process by user in advance.

Thus we think the bottom-up rule generation supporting system acting as following.

(1) The system gets indication from user what state a token should become.

(2) If there exists rule which perform transition to that state, using parameters the token has, consistent with previous indications, the system outputs this rule.

(3) If there is no such rule, the system decides that parameters of that token are insufficient for "inspection of situations", and ask user to give rules which generate new parameters.

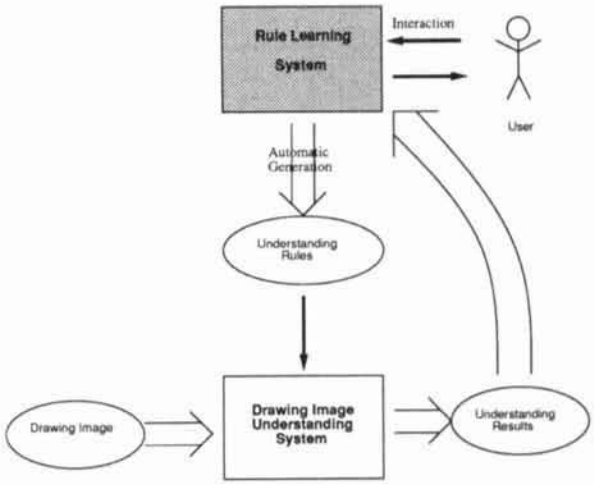

Figure 1: Outline structure of the system

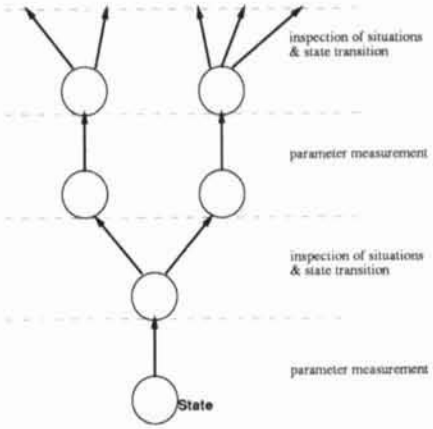

Figure 2: Sequence of bottom-up process

Repeating this sequence, the system can generate bottom-up rules. This rule generation supporting system is given rules for "parameter measurement" from user, and generate rules for "inspection of situations" automatically using learning from given examples.

3.2 Learning algorithm

The learning algorithm of our system is based on Shapiro's Model Inference System[4], extended to be able to handle numerical comparison and types of parameters. Our algorithm is a inductive inference algorithm which is intended to generate bottom-up process rules efficiently. It can handle list structures and nu-

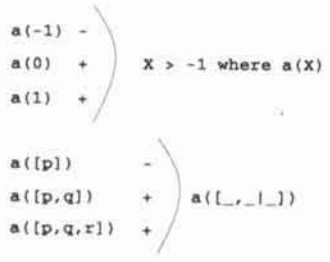

Figure 3: Examples of rule generation

merical data as types of parameters.

Basic behavior of the algorithm is that it generates a series of conditional test rule with possibility which fits to parameter types of given status as example, applies given examples to this rule, and seeks for the rule which is consistent with all examples. Generating conditional rules in the manner of more general ones to more specific ones gradually, the system will generate most general rule which is consistent with all examples. It performs a kind of pruning of a search tree for efficiency.

On our implementation described in section 4, states of tokens are represented as a Prolog's term. Fig. 3 shows examples of rule generation. Plus signs represent positive examples, and minus signs represent negative ones. These examples show this algorithm can generate rules using numerical data and list structures. Practically, the algorithm can deal more complicated states which are combination of numerical data and list structures as their parameters.

3.3 Interaction with user

On man-machine interaction systems, it is desirable to restrain number of interactions as few as possible. Our system uses learning algorithm to avoid repeating same questions again to restrain interactions.

Besides, it is preferable that interaction is as easy for a user as possible. To archive this, our system uses graphical user interface (GUI) to show the user intermediate understanding results visually, and it inquires user whether there are misunderstanding tokens. The user has only to pick up questioned tokens with mouse. These interactions is very easy for users.

4 Experiment

4.1 Implementation

We embodied an experimental system on Sun Sparc workstations, and let the system generate state transition rules. Within the whole system, we used the system described in [2] as the state transition type drawing understanding system, and created the rule learning system newly. This is described in SICStus Prolog and is about 800 lines in scale. Then the system was evaluated with several experiments.

4.2 Classification of colored segments

On early stage of color image analyses, colored segments composing images are classified according to their colors. Color is represented by number using color systems e.g. RGB, LAB, HIS etc. While classification of colored segments is done referring these values, it is not obvious to which extent of values of color system a certain color corresponds.

As the first experiment, we applied the system to generate color classification rules. Using our system, all the user had to do to generate color classification rules is to pick up segments of actual color images as positive or negative examples. We used sports scene images for color images, segmented into colored segments labeled with HIS (Hue, Intensity, Saturation) color system values.

We let the system to generate a rule to distinguish green segments. It was given indications of 6 positive example segments and 8 negative ones and generated a rule:

$$\begin{aligned} &\texttt{\$chain}(H, I, S) \longrightarrow \\ &\quad -125 \leq H, H \leq -56, \\ &\quad I \leq 432, \\ &\quad 743 \leq S, \\ &\quad \texttt{\$transform}(\texttt{\$green}). \end{aligned} \tag{1}$$

This rule expresses hue is between $-125°$ and $-56°$, intensity is under 432 (regularized to 768), and saturation is over 743 (regularized to 10,000). This rule could distinguish all green segments without any other colors in sufficient number of images properly. This was done with very plain interactions.

4.3 Map drawing processing

The next experiment is generating rules for understanding map drawings. Fig. 4 shows a sample of map drawings used for this experiment. Series of small dots in the map forms land usage boundaries. The system was evaluated to generate rules which could distinguish dots composing these land usage boundaries from others.

Fig. 5 shows how to extract land usage boundaries. At first, the system easily generated rules to distinguish small dots which might constitute land usage boundaries. Next, it should inspect surroundings of each dots and extract dots around which there are two dots except for itself within a "certain distance". Besides, it is no easy to determine this distance, for example, if this distance is too large, even ideal dots series are perceived over concentrated area; if this distance is too small, every dots seem to be disconnected at all.

In this experiment, we used the system to determine this "certain distance" interactively. At first, we gave the system following rule to inspect surroundings within multiple distances.

$$\texttt{\$dot} \longrightarrow$$
$$\texttt{\$get\_near\_loop}(100, P_1), \texttt{\$select}(P_1, \texttt{\$dot}, Q_1),$$
$$\texttt{\$get\_near\_loop}(105, P_2), \texttt{\$select}(P_2, \texttt{\$dot}, Q_2),$$
$$\texttt{\$get\_near\_loop}(110, P_3), \texttt{\$select}(P_3, \texttt{\$dot}, Q_3),$$
$$\texttt{\$get\_near\_loop}(115, P_4), \texttt{\$select}(P_4, \texttt{\$dot}, Q_4),$$
$$\texttt{\$get\_near\_loop}(120, P_5), \texttt{\$select}(P_5, \texttt{\$dot}, Q_5),$$
$$\texttt{\$transform}(\texttt{\$dot}(Q_1, Q_2, Q_3, Q_4, Q_5)). \quad (2)$$

Parameters Q_1 through Q_5 represent surrounding situations within the distance 100, 105, 110, 115, 120 respectively, each of them is a list of tokens at the state $dot without itself. After applying this rule to every tokens, the system selected an appropriate parameter and generated proper parameter checking rule automatically. As a result of this experiment, the system generated a following rule after accepting one positive example and four negative examples:

$$\texttt{\$dot}(\_, \_, \_, [A, B], \_) \longrightarrow$$
$$\texttt{\$transform}(\texttt{\$dot}(A, B)). \quad (3)$$

This rule expresses "there exist two dots except for itself within a distance 115." This rule is sufficiently proper, so the system can generated proper rules easily using only five interactions.

Figure 4: Sample of map drawing

5 Conclusion

We discussed the understanding rule generation supporting system which can generate effective rules from plain interactions with user using computer learning method. The system was evaluated with successful experiments to generate rules for classification of colored segments and understanding of map drawings, revealing that it can generate practical rules using only several plain interactions. This system can deal with only parametric data such as list structures and numerical data. However, to handle spatial data like drawings effectively, it comes to be important to use spatial properties. This point is now examined.

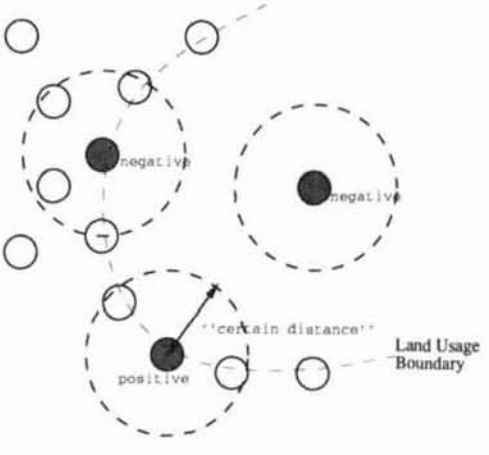

Figure 5: Organization of land usage boundaries

References

[1] S. Satoh, Y. Ohsawa, and M. Sakauchi, "A proposal of drawing image understanding system applicable to various target drawings (in Japanese)," *Transactions of Information Processing Society of Japan*, vol. 33, pp. 1092–1102, September 1992.

[2] S. Satoh, Y. Ohsawa, and M. Sakauchi, "Drawing image understanding framework using state transition models," in *Proc. of 10th ICPR*, pp. 491–495, 1990.

[3] S. Satoh and M. Sakauchi, "Descriptive ability of drawing image understanding framework using state transition models," in *Proc. of MVA '90*, pp. 199–202, 1990.

[4] E. Y. Shapiro, "Inductive inference of theories from facts," Tech. Rep. TR192, Yale University, Dept. of Computer Science, 1981.