ICOT Technical Memorandum: TM-0726他

TM-0726他

人工知能学会全国大会

May, 1989

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

TM-0726 Knowledge Acquisition by Generating Explanation with Hypotheses
by H. Taki & K. Tsubaki

TM-0735 部分計算によるプロダクション・システムの高速化
吉川康一、藤田　博、新谷虎松

TM-0738 知識獲得支援システム
石川啓子、田中立二、山尾雅利

TM-0740 KBMS PHIの問合せ処理におけるルール変換の評価
阿比留幸展、羽生田博美、森田幸伯、宮崎収兄

TM-0741 KBMS PHIにおける知識ベース演算エンジンの評価
田中　勉、和田光教、山下祥司、宮崎収兄

TM-0744 設計問題向け制約指向知識コンパイラにおける制約解析および手順生成について
永井保夫、寺崎　智

TM-0752 知識獲得支援システムEPSILON/One
椿　和弘、瀧　寛和、大崎　宏

仮説を含む説明構造の生成による仮説的知識の獲得
Knowledge Acquisition by Generating Explanation with Hypotheses

滝 寛和、椿 和弘
Hirokazu Taki, Kazuhiro Tsubaki

(財)新世代コンピュータ技術開発機構
ICOT Research Center, Institute for New Generation Computer Technology.

Abstract: This paper describes an explanation-based learning system under incomplete knowledge. We propose an explanation method using abduction and induction. If the system fails to explain a goal and to explain examples, it makes a hypothesis by abduction. In this way, there are many candidate hypotheses. The system uses a few criteria to eliminate candidates. Selection of hypotheses that are in a consistent explanation is explained by induction using a background theory and has minimal generation cost. After explanation generation, the explanation tree is generalized according to operationality criteria. As a result, it generates assumptive macro knowledge.

1. Introduction

Knowledge acquisition bottleneck is one of most difficult problems in the building of knowledge bases of expert systems. There are two ways to build knowledge bases. The first way is interactive knowledge acquisition: a human expert is interviewed to extract his knowledge. The other way is knowledge building by understanding observed information [Taki 88]. One way to understand something is to try to explain it. EBL(Explanation-based learning)[Mitchell 85][Mitchell 86] is a learning method which uses explanation of examples. However, EBL makes only macro knowledge for effective inference. In order to use EBL, a complete domain theory must be prepared before learning. Normally, however, it is difficult to prepare complete knowledge before learning. Therefore, a learning system fails to explain examples. This situation is a trigger which makes other learning occur. Impasse situations of chunking in SOAR[Laird 86] and frustration situations of FBL[Suwa 89] are like failure situations of explanation. Abductive explanation is a kind of explanation method which uses hypotheses. In this explanation, a hypothesis is made by abduction[Charniak 86]. A hypothesis is new knowledge if it does not derive inconsistency. There is a shortcoming in abduction, which generates many candidates as hypotheses to explain examples. It is important to select meaningful hypotheses from all hypotheses by abduction. [Taki 89] This paper introduces a hypothesis selection method which considers the operation cost of generalization and specialization of a domain theory.

2. Abductive Explanation

When we want to explain something but there is insufficient information, we make hypotheses. These hypotheses are either uncertain knowledge or assumed knowledge. The following sections describes how to assume knowledge for explanation. In the discussion, we assume that background theory, a goal and examples are given in learning process. This background theory is not enough to explain the goal with examples.

2.1 Explanation-Based Learning and Incomplete Domain Theory

EBL is a learning about efficiency of knowledge. In EBL, complete domain theory, goal concept, learning examples and operationality criteria must be prepared. EBL explains a goal using examples and domain theory. An explanation tree is generalized by operationality criteria. EBL doesn't learn new knowledge. Before learning, it is hard to prepare a complete domain theory. If a framework of EBL under incomplete domain theory were developed, it could learn new knowledge and efficiency of knowledge usage. There is some incompleteness in the domain theory as follows.

(1) Completeness: There is some lack of knowledge, so some examples cannot be explained.

(2) Soundness: There is some wrong knowledge, so wrong examples are explained.

(3) Consistency: The domain theory contains inconsistency, so inconsistency is detected in an explanation.

(4) Over-generalization: Knowledge is too general, so

it explains negative examples.
(5) Over-specialization: Knowledge is too special to explain positive examples.

We aim to develop this framework using abduction and induction. We aim to develop an explanation system which deals with incomplete knowledge (1) and (5). It makes assumptive explanations and generalizes explanations. As a result, it generates assumptive macro knowledge which contains hypothesis as a new knowledge and usage of the hypothesis.

2.2 Explanation by Hypothetical Reasoning

In explanation with hypotheses, there may be inconsistency. Hypothetical reasoning deals with inconsistent reasoning during hypothesis selection. A formulation of a hypothetical reasoning[Pool 88] is as follows.

B: a background theory, H: a set of hypotheses, E: examples,
G: a goal, h: a subset of H

$$h \in H$$
$$B \cup E \nvdash G$$
$$B \cup E \cup h \vdash G$$
$$B \cup E \cup h \nvdash \bot$$

We eliminate the first formula and add the hypothesis generation formula.

$$B \cup E \nvdash G$$
$$B \cup E \cup G \mid\!\!< h \text{ (Induction)}$$
$$B \cup E \cup h \vdash G$$
$$B \cup E \cup h \nvdash \bot$$

This formula is shown in induction[Genesereth 86]. We use the latter formulation. We think that this formulation is suitable not only for induction but also for abduction and analogy. Figure 1 shows an explanation system using ATMS[de Kleer 86] as hypothetical reasoning.

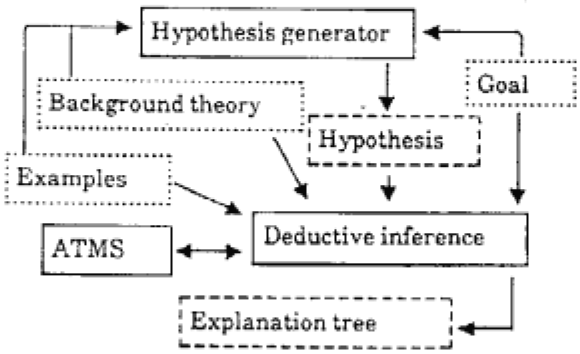

Figure 1 The explanation system

2.3 Generation of Hypothesis by Abduction

There are two kinds of candidates for hypotheses in this framework. They are fact from knowledge and rule form knowledge. We deal with the fact form knowledge as hypothesis. It is difficult to generate hypothesis deductively. This problem contains when, how and on what to generate hypotheses. When a reasoning process fails to make an explanation, this situation is a trigger to make a hypothesis. Abduction is the process that generates explanations.

Here is an example which makes an explanation by abduction.
(Example 1) Explanation about a bird, tweety
Background theory: bird(X) :- fly(X), has(X, wings), has(X, bill).
Examples: has(tweety, wings)
fly(tweety)
Goal: bird(tweety)

Given bird(X) :- fly(X), has(X, wings), has(X, bill)
Given bird(tweety)

Infer
has(tweety, wings), fly(tweety), has(tweety, bill)

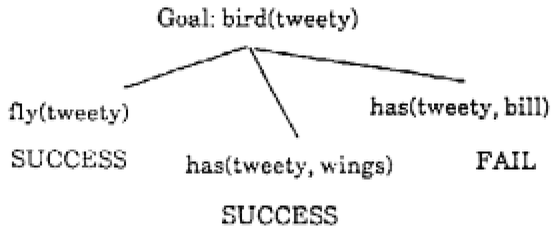

Figure 2 An explanation

"has(tweety, wings) and fly(tweety)" are known. Therefore a result of abduction is "has(tweety, bill)". If there is the hypothesis, "has(tweety, bill)", then the explanation of example 1 is established.

3. Selection Abductive Explanation

If we use abduction for any explanation, we can explain anything. It is necessary to select good hypotheses in order to generate better explanations. Selection of hypotheses means selection of explanations. This section describes an evaluation as selection criteria.

3.1 Selection Hypothesis

We define the following criteria to select hypotheses.
(1) An explanation has no inconsistency.
(2) A hypothesis made by abduction is explained from concepts of version spaces which are generated from facts and deductive conclusions of the domain theory.
(3) The explanation with the lowest total cost of hypothesis generation should be selected

The second criterion selects hypotheses which are explained by induction. If there are two hypotheses in an explanation, the total cost is the sum of both hypothesis generation costs.

3.1.1 Types of Hypothesis

To begin with, types of hypothesis in the explanation system are introduced. Necessary hypothesis is dependent on a sequence of explanation. In example 1, if a goal concept is bird(X), there is a difference between rule 1 "bird(X) :- fly(X), has(X, wings), has(X, bill)" and rule 2 "bird(X) :- has(X, bill), fly(X), has(X, wings)".

In an inference using the rule 1, when "has(X, bill)" is checked, the value "X" has been unified to "tweety" which was fixed at checking "fly(X)". Therefore, a result of abduction in this case is "has(tweety, bill)". On the other hand, when "has(X, bill)" is checked in the case of the rule 2, the value "X" is not fixed. The result of abduction is "has(X, bill)". After reasoning, both explanations are the same. In the second case, as a result, "has(X, bill)" is "has(tweety, bill)". This shows that necessary hypothesis is "has(tweety, bill)" in both cases. We call this result of abduction "has(tweety, bill)" a basic hypothesis and the result "has(X, bill)" a temporal hypothesis. If there are a basic hypothesis and a temporal hypothesis from an item by abduction in an explanation, the temporal hypothesis is more general than the basic hypothesis and the basic hypothesis is the most special hypothesis in the explanation.

3.1.2 Version Space and Hypothesis

Knowledge in the background theory can explain only concepts that are more specific than itself. Generalized knowledge is necessary to explain a hypothesis. Sets of generalized knowledge are represented by version spaces made from the background theory. Normally, the induction function decides an upper and lower boundary of a version space using positive and negative examples of a concept. If the background theory doesn't include negative examples, version spaces of concepts in the background theory have no upper boundaries. Therefore, the most generalized form of an item has its own predicate name and arguments that are represented by valuables. This means that the predicate name is not generalized. The lower boundaries of those version spaces are defined by facts that are deduced from the background theory. A concept in a version space which explains a basic hypothesis is called "a support concept for the basic hypothesis". The most specific concepts of support concepts is called "the least generalized support concept". The least generalized support concept is one of concepts made from the basic hypothesis and facts in the background theory by least generalization.

(Example 2) The least generalized support concept

In example 2, "has(fish, fins)" is the lower boundary of the version space, "has(X,Y)" is the upper boundary of it. "has(X, fins)" and "has(fish, Y)" are concepts in the version space. If "has(dolphin, fins)" is a basic hypothesis, "has(X,Y)" and "has(X, fins)" are support concepts for the hypothesis because these unify to it. A least generalized support concept is "has(X, fins)". If "has(beetle, horn)" is a basic

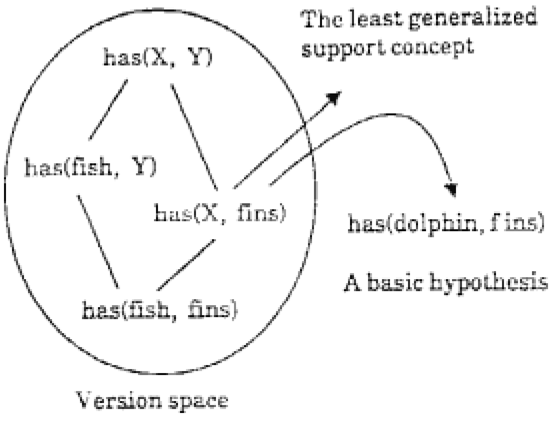

Figure 3 A version space and a hypothesis

hypothesis, a least generalized support concept is "has(X, Y)". If "eat(dolphin, fish)" is a basic hypothesis, there is no support concept for this hypothesis in this version space.

3.1.3 Hypothesis Generation Cost

This section describes operations for predicate modification. Each operation has the same cost and is identified as 1 point. Three operations are defined here. The first operation, argument generalization operation (G-op), exchanges a fixed argument to a variable. The second operation, argument disconnect operation (D-op), eliminates a dependency between two variables. The third operation, argument unification operation (U-op), unifies a instance and a variable.

(Example 3) Operations

G-op: has(dolphin, fins) -> has(X, fins) or has(dolphin, Y)

D-op: has(X,X) -> has(X,Y)

U-op: has(X, fins) -> has(fish, fins)

In example 2, total cost of generation "has(dolphin, fins)" from "has(fish, fins)" is 2 points. This generation process consists of "G-op: has(fish, fins) -> has(X, fins)" and "U-op: has(X, fins) -> has(dolphin, fins)".

4. Assumptive Macro Knowledge Generation

A basic hypothesis is very special for learned knowledge. A least generalized support concept is a support hypothesis for a basic hypothesis. Therefore, we use the least generalized support concept as a hypothesis to learn new knowledge. If the support concept is denied, the basic hypothesis is also denied. The support concept may explain other hypotheses different from the basic hypothesis. However, examples and an explanation are special cases. The learning system must learn more general inconsistent knowledge. We define assumptive macro knowledge shown in a following form.

Knowledge generated by EBL:
goal(X1, ..., Xn) :- p1(Xi, .., Xj, ..), , pm(...). --(1)

Hypothesis: h1(Xk, dolphin).
Assumptive macro knowledge (represented in default rule form):

$$\frac{\text{p1}(\ldots),\ \text{p2}(\ldots),\ \ldots,\ \text{pm}(\ldots)\ \text{M: h1}(\text{Xk, dolphin})}{\text{goal}(\text{X1}, \ldots, \text{Xn})}$$

Unless "h1(Xk, dolphin)" is not denied, a horn clause (1) is available.

5. Semantic Hypothesis Selection

The assumptive macro knowledge is generated syntactically. This section shows how we can express semantics about generalization and specialization.

5.1 Generalization Level of Hypothesis

The least generalized support concept is generated from the background theory and examples by generalization. However, there are some limits for generalization. These limits decide the upper boundaries of version spaces made from known knowledge such as negative examples. These constraints are dependent on application fields. For example, structured mapping theory[Falkenhainer 87] controls analogical mapping between two events. It maps structure information but not attributes of objects in these events. In order to control generalization levels of special predicates, this system allows the user to limit version spaces of predicts which are defined in a set of special names or have a fixed number of arguments.

5.2 Operationality

In order to generate operational macro knowledge, EBL generalizes an explanation tree about its structure. In our learning method, there are two relations between a hypothesis and an operationality criterion.

(1) An operationality criterion is more special than a hypothesis.
In this case, the hypothesis exists in macro knowledge. If this macro is used in an inference, this hypothesis must be explained by deduction. This type of macro knowledge is normal knowledge but not default knowledge.

(2) An operationality criterion is more general than a hypothesis.
In this case, the hypothesis doesn't exist in macro knowledge. It is eliminated from the explanation tree by an operatinality criterion. Therefore, this macro is independent of knowledge whose generalization level represents the hypothesis. The hypothesis is no longer necessary to explain the macro knowledge. However, if the hypothesis is denied, the macro knowledge loses its generation cause/reason. This type of knowledge is represented in assumptive macro knowledge form.

6. Summary

This paper discussed a learning method about how hypothetical knowledge is extracted. This method uses an advanced reasoning which generates abductive explanations and selects hypotheses according to checking costs of hypothesis generation. A basic hypothesis is made by abduction when an explanation is not generated. A least generalized support concept is also derived from the basic hypothesis and version spaces which are made by a background theory and examples. This support concept is explained by induction of known knowledge. A assumptive macro knowledge is made from the explanation with this support concept. This knowledge is available unless the support knowledge is not denied. A current version of this method deals with facts as hypotheses. It is necessary for us to develop rule generation methods to explain examples and an integrated knowledge acquisition system (EPSILON/2) which consists of an interview system and a learning system.

References

[Charniak 86] Charniak, E. and McDermott, D., Introduction to Artificial Intelligence, Addison Wesley, 1986

[de Kleer 86] de Kleer, J., An assumption-based TMS, Artificial Intelligence 28,pp.127-162, 1986

[Falkenhainer 87] Falkenhainer, B., Forbus, K.D. and Gentner, D, The Structured-mapping Engine: Algorithm and Examples, Technical Report, UIUCDCS-R-87-1361, Dep. of Computer Science, University of Illinois, 1987

[Genesereth 86] Genesereth, M. and Nilsson, N., Logical Foundations of Artificial Intelligence, Los Altos, CA: Morgan-Kaufmann, 1986

[Laird 86] Laird, J., Rosenbloom, P. and Newell, A., Universal Subgoaling and Chunking, Kluwer Academic Publishers, 1986

[Mitchell 77] Mitchell, T., Version spaces: A candidate elimination approach to rule learning, Proc. 5th IJICAI,pp.305-310, 1977

[Mitchell 85] Mitchell, T., Mahadevan, S. and Steinberg, L., LEAP: A Learning Apprentice for VLSI Design, Proc. 9th IJICAI, Los Angeles, 1985

[Mitchell 86] Mitchell, T., Keller, R., and Keder-Cabelli, S., Explanation-Based Generalization: A Unifying View, Machine Learning 1, 1986

[Poole 88] Poole, D., A Logical Framework for Default Reasoning, Artificial Intelligence 36, pp.27-47, 1988

[Suwa 89] Suwa, M. and Motoda, H., Understanding Metaphor by Frustration-Based Learning, Proc. SICE 9th Knowledge Engineering Symposium, 1989 (in Japanese)

[Taki 87] Taki, H., Tsubaki, K. and Iwashita, Y., Expert Model for Knowledge Acquisition, Proc. IEEE 3rd Annual Expert Systems in Government Conference, Washington D.C., 1987

[Taki 88] Taki, H, Knowledge Acquisition by Observation, Proceedings of FGCS'88, Tokyo, 1988

[Taki 89] Taki, H, Knowledge Acquisition by Abductive Explanation, Proc. of SICE'89, Matsuyama, 1989 (to appear)

[Uchihashi 89] Uchihashi, T. and Taki, H., An Interactive EBG System, ICOT Technical Report, 1989 (to appear)