

# 誤差逆伝播を利用した重み付き仮説推論の教師あり学習

山本 風人<sup>1,a)</sup> 井之上 直也<sup>1,b)</sup> 渡邊 陽太郎<sup>1,c)</sup> 岡崎 直観<sup>1,2,d)</sup> 乾 健太郎<sup>1,e)</sup>

**概要:** 本稿では、重み付き仮説推論のパラメタを教師あり学習によって自動調整する手法を提案する。仮説推論は、与えられた観測に対して評価関数を最大化する最良の説明を求める推論であり、自然言語処理において文章に明示されていない情報の顕在化を行うことに対して有用な枠組みとして注目を浴びている。しかしその一方で、仮説推論の評価関数の学習手法が未だ提案されておらず、評価関数のパラメタの調整は、人手による調整やヒューリスティックな手法に頼らざるを得ないという問題があった。そこで我々は、仮説推論の拡張のひとつである重み付き仮説推論を対象として、仮説の証明木におけるリテラル間のコストの関係をフィードフォワードニューラルネットワークの形で表現することで、誤差に対する各パラメタの勾配を求め、評価関数のパラメタの識別学習を実現する。また、提案手法によって評価関数を学習できていることを確かめるために、既存のデータセットを用いて実験した結果についても報告する。

**キーワード:** 仮説推論、機械学習、誤差逆伝播、ニューラルネットワーク

## Backpropagation Learning for Weighted Abduction

KAZETO YAMAMOTO<sup>1,a)</sup> NAOYA INOUE<sup>1,b)</sup> YOTARO WATANABE<sup>1,c)</sup> NAOAKI OKAZAKI<sup>1,2,d)</sup> KENTARO INUI<sup>1,e)</sup>

### Abstract:

We explore a discourse processing framework for discovering implicit information in texts, based on Hobbs et al.'s weighted abduction [5]. Abduction is inference to the best explanation. In weighted abduction, the best explanation is defined as the explanation that minimizes a parametrized cost function. However, less attention has been paid to how to tune the parameters of the cost function automatically. In this paper, we propose a discriminative approach to learning parameters in weighted abduction. We represent the transition of costs in a proof tree as feed-forward neural networks, and calculate the gradients of parameters in a background knowledge base. Our experiments show that our method correctly learns parameters on the existing dataset of plan recognition.

**Keywords:** Abduction, Machine Learning, Error Backpropagation, Neural Networks

## 1. はじめに

文章に明示されていない情報(照応関係、登場人物の意図など)を顕在化する処理(*Making the implicit explicit*)は、情報抽出やQAなどの多くの自然言語処理のタスクにおいて重要な技術である。我々は、情報の顕在化を行う枠組み

として、仮説推論を用いた言語理解の枠組みの構築に取り組んでいる。

仮説推論とは、与えられた観測に対して、評価関数を最大化する最良の説明を求める推論である。仮説推論を言語理解に適用する試みは1980年代から1990年代に多く見られたが、そのうち最も代表的な枠組みがHobbsら[5]の*Interpretation as Abduction (IA)*である。Hobbsらは、照応解析や談話関係などの談話理解に関する種々のタスクを、仮説推論(アブダクション)のタスクとして自然に定式化できることを示した。たとえば、“*John went to the bank. He got a loan.*”という文章に対して解釈を行うとしよう。IA

<sup>1</sup> 東北大学

<sup>2</sup> 科学技術振興機構 さきがけ

a) kazeto@ecei.tohoku.ac.jp

b) naoya-i@ecei.tohoku.ac.jp

c) yotaro-w@ecei.tohoku.ac.jp

d) okazaki@ecei.tohoku.ac.jp

e) inui@ecei.tohoku.ac.jp

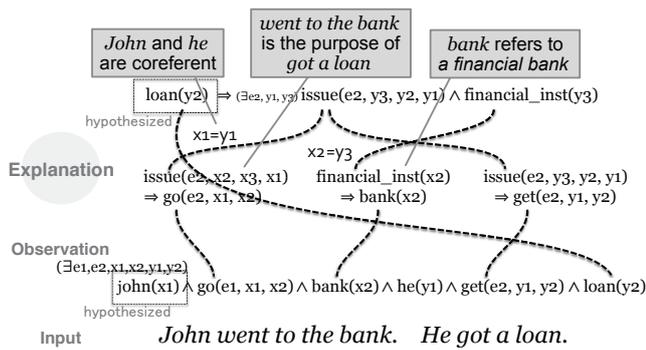


図1 仮説推論による言語理解の例

では、世界知識を背景知識、解釈の対象となる文章を観測として仮説推論を行い、最良の説明を解釈の結果として出力する。図1に、仮説推論による文の解釈の様子を示す。解釈の結果から、「JohnとHeが共参照関係である」「went to the bankの目的はgot a loanであった」といった、文章に明示されていない情報が顕在化されていることがわかる。

仮説推論による言語理解のモデル化には、各サブタスク間の相互依存関係を考慮し、それぞれのタスクにおいて一貫した解釈のうち、最も良い解釈を出力する処理を自然にモデル化できるという利点がある。情報の顕在化における最新の研究プロジェクトであるMachine Readingにおけるいくつかの推論モデルも、仮説推論を用いて自然にモデル化できると考えられる[10]。

IAの枠組みは、1990年代-2000年代を経てふたたび注目を浴び始めている[1], [9], [16], [17]。これには、大きく2つの背景があると考えられる。ひとつめに、仮説推論を実用規模の問題に適用できる程度の、大規模な世界知識を取り揃える技術が昔に比べて大幅に成熟してきたことが挙げられる[2], [3], [12], [13]。例えば文献[9]では、WordNet[3]とFrameNet[12]を用いて約数十万の推論規則からなる背景知識を構築し、含意関係認識のタスクにIAを適用している。ふたつめの背景には、効率的な仮説推論エンジンが提案されたことが挙げられる[6]。例えば文献[17]では、約数十万の推論規則からなる背景知識を用いて含意関係認識のデータセットに対して推論を行い、先行研究より大幅に高速な推論を行えたことが報告されている。

しかしながら、IAを言語理解に適用するには、クリアすべき課題がもうひとつある。それは、仮説評価関数の学習手法の開発である。我々が意図する解釈を出力するためには、仮説の評価関数を適切に選択する必要があるが、我々の知る限り、仮説推論の評価関数を学習する手法は提案されていない。先に挙げた先行研究では、仮説推論の評価関数のパラメータを手入、またはヒューリスティックな手法に基づいて与えているが、人手によるチューニングを毎回行うのは現実的でない。

そこで本稿では、仮説推論の評価関数のパラメータを、教

師あり学習により識別的に学習する手法を提案する。本稿では、Interpretation as Abductionのなかで用いられている**重み付き仮説推論**の評価関数を仮説評価関数として用いることを想定し、そのパラメータを学習する手法について述べる。具体的には、仮説推論の出力をフィードフォワードニューラルネットワークに変換し、誤差逆伝播法を用いて最適なパラメータを求める。

本稿では、まず重み付き仮説推論について説明したあと(2節)、重み付き仮説推論の評価関数の教師あり学習手法について説明する(3節)。次に、提案手法による重み学習の動作を確認する実験の結果について報告したあと(4節)、最後に関連研究との関係について述べる(5節)。

## 2. 背景

### 2.1 重み付き仮説推論

仮説推論とは、与えられた観測に対して最良の説明を求める推論である。形式的には次のように定義される。

**Given:** 背景知識  $B$ 、観測  $O$ 。ただし  $B, O$  は一階述語論理式の集合。

**Find:** 仮説  $H$ 。ただし  $H$  は  $H \cup B \models O, H \cup B \not\models \perp$  を満たす一階述語論理式の集合。

本稿では、上の条件を満たす  $H$  を**候補仮説**と呼び、そのなかで最良の仮説を**解仮説**と呼ぶこととする。候補仮説の中から解仮説を選択する際の評価方法は様々な手法が存在するが、我々は重み付き仮説推論の枠組み[5]を用いる。重み付き仮説推論では、観測  $O$  と仮説  $H$  は存在限量されたリテラルの連言として表現され、各リテラルは正の実数の**コスト**を持つ。直感的には、リテラルのコストとはそのリテラルの不確かさであり、コストが高いリテラルを説明するためにはより多くの根拠が必要となる。また、背景知識  $B$  はホーン節:  $P_1^{w_1} \wedge P_2^{w_2} \wedge \dots \wedge P_n^{w_n} \Rightarrow Q$  の形式を持つ一階述語論理式(**公理**)の集合である。前件のリテラルには正の実数の**重み**が割り当てられている。直感的には、背景知識の重みはその背景知識による推論のこじつけ度合いを表し、重みが高い背景知識によって仮説されたりテラルは高いコストを持つこととなる。

重み付き仮説推論では、以下の二つの操作を観測に対して逐次適用することで、候補仮説を生成する:

(i) **後ろ向き推論**は、背景知識  $P_1(x)^{w_1} \wedge P_2(x)^{w_2} \wedge \dots \wedge P_n(x)^{w_n} \Rightarrow Q(x) \in B$  と、観測または仮説リテラル  $Q(a)$  を前提として、新しいリテラルの集合  $\{P_i(a)\}_{i=1}^n$  を導き、候補仮説に追加する処理である。このとき、生成されたりテラル  $P_i(a)$  のコスト  $cost(P_i(a))$  を、リテラル  $Q(a)$  のコスト  $cost(Q(a))$  と、背景知識の重み  $w_i$  の掛け算により計算し ( $cost(P_i(a)) = w_i \cdot cost(Q(a))$ )、リテラル  $Q(a)$  のコストを0とする。例えば、 $O = \exists x(q(x)^{S10})$ 、 $B = \forall x(p(x)^{1.2} \Rightarrow q(x))$  が与えられた場合は、 $H_1 = \exists x(q(x)^{S10})$ 、 $H_2 = \exists x(q(x)^{S0} \wedge p(x)^{S12})$  の二つの候補仮説を生成する。(ii) **単一化**は、同一の述語

を持つ2つのリテラルを合成し、互いの項が等価であるという仮説を置く操作である。この際、2つのリテラルが持つコストのうち大きい方のコストを0にする。例えば、 $O = \exists x, y(p(x)^{S10} \wedge p(y)^{S20} \wedge q(y)^{S10})$  が与えられた場合は、 $H = \exists x, y(p(x)^{S10} \wedge q(y)^{S10} \wedge x = y)$  を候補仮説として生成する。

背景知識の重み  $W$  のもとでの、観測  $O$  に対する解仮説  $H_{O,W}^*$  は、候補仮説の集合  $\mathbb{H}$  の中で最小のコストを持つ候補仮説、すなわち最も蓋然的な仮説と定義される。

$$H_{O,W}^* = \arg \min_{H \in \mathbb{H}} C_{O,W}(H) \quad (1)$$

ここで、 $C_{O,W}(H)$  は候補仮説のコストである。候補仮説のコストは、候補仮説に含まれるリテラル (要素仮説) のコストの和として定義され、次の式により表現できる。

$$C_{O,W}(H) = \sum_{p \in H} cost(p) \quad (2)$$

$$= \sum_{p \in P_H} \left[ \prod_{w \in chain(p)} w \right] cost^*(obs(p)) \quad (3)$$

ここで、 $P_H$  は仮説  $H$  の要素仮説のうち、単一化されていない、かつ説明されていない要素仮説の集合を表す。 $cost(p)$  はリテラル  $p$  が持つコスト、 $cost^*(p)$  はリテラル  $p$  が後ろ向き推論や単一化によってコストが0になる前に持っていたコスト (以下**仮説時コスト**と表記する) を表す。 $chain(p)$  はリテラル  $p$  を仮説するために用いられた背景知識の重みの集合、 $obs(p)$  はリテラル  $p$  を仮説するために用いられた観測を表す。背景知識はホーン節であるので、 $p$  から  $obs(p)$  へ辿る道筋は枝分かれせず一つに定まる。

重み付き仮説推論の仮説評価指標を用いる利点としては、仮説の特殊性を評価できることが挙げられる。仮説推論では、後ろ向き推論が適用されるごとに仮説の特殊性は高くなるが、一般には仮説が特殊になるほど仮説を支持する証拠は少なくなるため、信頼度が高い範囲で推論を停止できることが望ましい。重み付き仮説推論では、仮説コストの伝播と単一化の操作によって、これが自然に実現できているといえる。

しかしその一方で、評価関数のパラメータである、観測のコストと背景知識の重みの割当てについて、機械学習手法が確立されていないために人手による調整を余儀なくされている実情がある。そこで本稿では、これらのパラメータを教師あり学習によって調整することを目指す。

### 3. 誤差逆伝播を使用した重み付き仮説推論の教師あり学習

本論文では、重み付き仮説推論の評価関数のパラメータ推定問題を、隠れ変数付きの構造識別学習問題として捉え、最適なパラメータを学習する手法を提案する。具体的には、入出力のペア、すなわち観測と解仮説 (または、その一部)

のペア  $(O_i, Y_i)$  が訓練事例の集合として与えられたとき、すべての訓練事例  $i$  について、 $O_i$  を観測とした候補仮説の集合の中で、 $Y_i$  を含む仮説のコストが最も小さくなるよう、パラメータを学習する。以後、 $Y_i$  を**正解**と呼ぶ。

つぎに、図2を用いて提案する手法の概要を説明する。本手法では、各訓練事例  $(O_i, Y_i)$  について、まず現在のパラメータ  $W$  のもとで入力  $O_i$  に対して重み付き仮説推論を行う。出力された解仮説  $H_{O_i,W}^*$  が正解  $Y_i$  を含むならば、パラメータ更新を行わずに次の訓練事例に移る。一方、解仮説  $H_{O_i,W}^*$  が正解  $Y_i$  を含まない場合には、“正解  $Y_i$  を含む候補仮説の集合の中で、最もコストの小さい候補仮説  $H_{O_i,W}^+$ ” を求め、 $H_{O_i,W}^+$  に対する  $H_{O_i,W}^*$  のコストの誤差を得る。つぎに、パラメータ  $W$  の更新を行うために、仮説に含まれるリテラルのコストの遷移関係と等価となるようなフィードフォワードニューラルネットワークを生成し、誤差逆伝播を行う。以降の各セクションでは、以上の手続きについての詳細を述べていく。

#### 3.1 問題設定

本学習手法は、以下の2つの状況においてパラメータを学習することを想定している：

##### (i) 仮説の正解判定関数を定義可能

識別学習を行うための条件として、観測に対して得られた解仮説が正しい仮説であるかどうかを判別できなければならない。しかし、“仮説が意味的に正しいかどうか”は人間にとっても判断が曖昧な場合が存在し、これをモデル化することは一般的に考えて非常に難しい。そこで本稿では、何らかのタスクを仮定し、タスクベースの学習に持ち込むことで、解仮説に対する正解判定を可能にする。このとき仮定するタスクは、次の要件を満たす関数  $J(O, H)$  が定義可能なタスクでなければならない。

- (1) 観測  $O$  に対して推論された仮説  $H$  がそのタスクにおいて正解かどうかの真偽値を返す
- (2) 出力値が観測と仮説の組  $(O, H)$  に対して一意に定められる

##### (ii) 正解を仮説するのに十分な背景知識がある

与えられた問題の正解を含む仮説を候補仮説として生成するために必要な背景知識は、十分に揃っていると仮定する。理由は次のとおりである。仮説推論において、正解を含む候補仮説を解仮説として得るために必要な要素は、次の2つだと考えられる：

- (1) 仮説の評価関数のパラメータ割当てが適切であること
- (2) 観測から正解を生成するのに必要な背景知識が存在する

これらのうち、前者が満たされていない場合は、正解を含む仮説  $H^*$  が候補仮説に含まれているが、解仮説でない (コストが候補仮説の中で最小でない) という状況にあたる。この場合、 $H^*$  が解仮説として出力されるよう、パラメータを

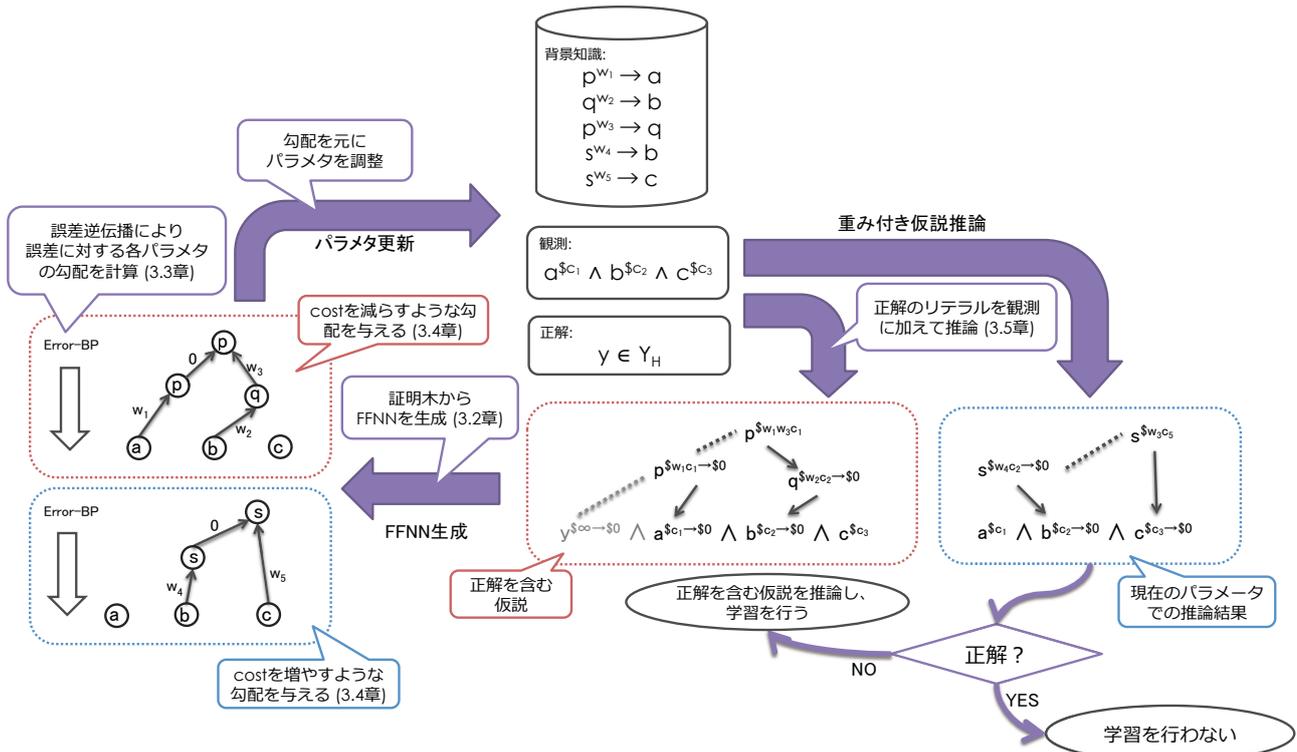


図2 提案手法の概要

適切な方向に調整することで対応できる。一方、後者が満たされていない場合は、正解となる仮説がそもそも候補仮説に含まれないため、どのようにパラメタ調整を行っても正解を含む仮説を解仮説として得ることは出来ない。よって、本稿ではそのような場合については対象外とする。

### 3.2 フィードフォワードニューラルネットワークへの変換

ここでは、重み付き仮説推論の仮説における各リテラルのコストの遷移関係を、フィードフォワードネットワーク (FFNN) を用いて表現することを考える。

まず、後ろ向き推論について考えてみよう。FFNNにおいて、ノード  $j$  の出力は次の式で計算される。

$$z_j = h(a_j) \tag{4}$$

$$a_j = \sum_{i \in \{i | e_{i \rightarrow j} \in E\}} z_i \times w_{i \rightarrow j} \tag{5}$$

$z_i$  はノード  $i$  の出力、 $a_i$  はノード  $i$  の活性、 $h(a)$  は活性化関数、 $E$  はネットワーク中の全てのエッジの集合、 $e_{i \rightarrow j}$  はノード  $i$  からノード  $j$  へ向かう有向エッジ、 $w_{i \rightarrow j}$  はエッジ  $e_{i \rightarrow j}$  が持つ重みを表す。以上から、重み付き仮説推論における各リテラルを FFNN のノードとみなし、リテラルの仮説時コストをノードの出力値に、後ろ向き推論の操作とそれに用いられた背景知識  $j \rightarrow i$  の重み  $w_{j \rightarrow i}$  をそれぞれエッジとその重みとして、活性化関数を  $h(a) = a$  とすれば、上式は

$$cost^*(j) = \sum_{i \in \{i | e_{i \rightarrow j} \in E\}} cost^*(i) \times w_{j \rightarrow i} \tag{6}$$

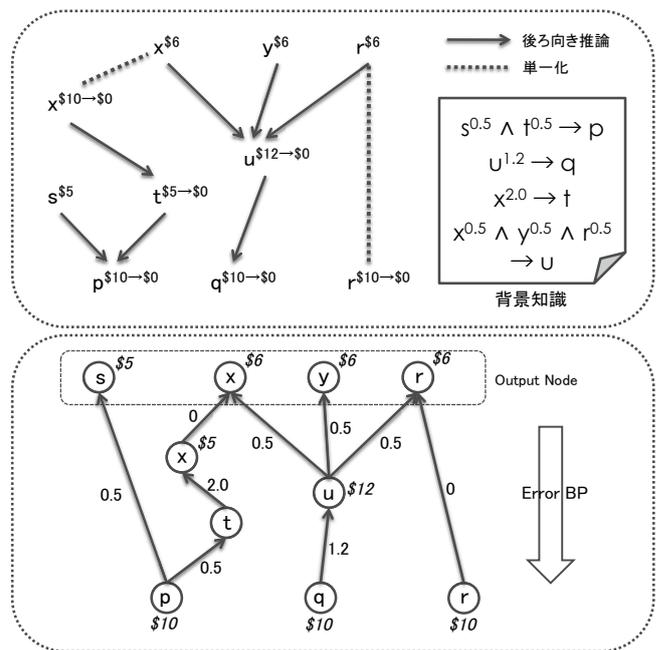


図3 推論グラフから FFNN への変換例

となり、重み付き仮説推論における後ろ向き推論のコストの計算式と等価になる。

次に、単一化について考える。単一化は、二つのリテラルのうちコストが大きい方のコストを 0 にする操作である。これは FFNN 上では、コストが大きい方のリテラルから小さい方のリテラルへ、重み 0 の有向エッジを置くことによって表すことができる。

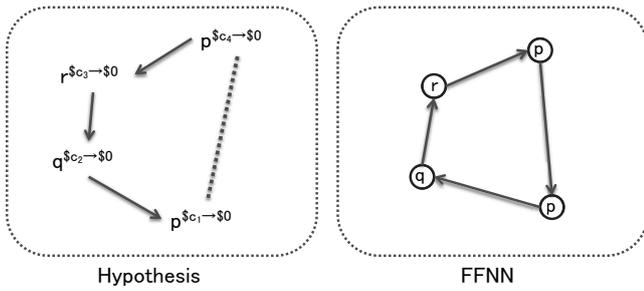


図4 ループが生じてしまうような場合の例

図3に例を示すように、以上の方法により、重み付き仮説推論によって得られた仮説におけるリテラルのコスト遷移は、等価なFFNNに変換することが可能であることが示された。これは、FFNNにおける種々のテクニックが、重み付き仮説推論の学習においても利用可能であることを示唆している。

ところで、FFNNはループが存在しないネットワークであるため、FFNNに変換される仮説は、変換した際にループを持たないものでなければならない。仮説をFFNNに変換した際にFFNNにループが存在するような場合とは、図4に示すようにリテラルがその子孫と単一化する場合である。そこで本稿では、リテラルに対して“自身の子孫であるリテラルと単一化してはならない”という制限を加えることによって対応する。そのような推論を許した場合、あるリテラルについてその仮説元を説明の根拠としても使うことに対応することから、推論に対してこのような制限を加えることは重み付き仮説推論の枠組みの上でも妥当であると考えられる。

### 3.3 誤差逆伝播

観測  $O$ 、重み  $W$  のもとで得られる解仮説  $H_{O,W}^*$  に対する誤差を  $E_{O,W}$  とする。ここで  $E_{O,W}$  は解仮説のコスト  $C_{O,W}(H_{O,W}^*)$  によって微分可能な関数であるとする。リテラル  $q$  のコスト  $cost^*(q)$  について、背景知識  $p_1^{w_1} \wedge \dots \wedge p_n^{w_n} \rightarrow q$  を用いて後ろ向き推論した場合を考える。このとき、リテラル  $q$  の仮説時コスト  $cost^*(q)$  の誤差  $E_{O,W}$  に対する勾配は、ニューラルネットワークにおける誤差逆伝播の考え方から、リテラル  $p_1, \dots, p_n$  の仮説時コスト  $cost^*(p_1), \dots, cost^*(p_n)$  を用いて次のように表せる。

$$\frac{\partial E_{O,W}}{\partial cost^*(q)} = \sum_{i=1}^n \frac{\partial E_{O,W}}{\partial cost^*(p_i)} \frac{\partial cost^*(p_i)}{\partial cost^*(q)} \quad (7)$$

$$= \sum_{i=1}^n \frac{\partial E_{O,W}}{\partial cost^*(p_i)} w_i \quad (8)$$

また、同様にして  $E_{O,W}$  に対する背景知識の重み  $w_1, \dots, w_n$  の勾配は次の式で表せる。

$$\frac{\partial E_{O,W}}{\partial w_i} = \frac{\partial E_{O,W}}{\partial cost^*(p_i)} \frac{\partial cost^*(p_i)}{\partial w_i} \quad (9)$$

$$= \frac{\partial E_{O,W}}{\partial cost^*(p_i)} cost^*(q) \quad (10)$$

以上から、コストが0でないリテラルから勾配を観測側へ伝播させていくことにより、仮説中のリテラルおよび背景知識の重みに対する勾配が求められる。あとは、それぞれについて求められた勾配を用いてパラメータの更新を行えばよい。一般に、あるパラメータ  $x$  について更新後のパラメータを  $\hat{x}$  とすると、 $\hat{x}$  は次のような式で計算される。

$$\hat{x} = x - \eta \frac{\partial E_{O,W}}{\partial x} \quad (11)$$

ここで  $\eta$  は学習率を表す。

### 3.4 誤差関数

教師あり学習を行うためには教師信号が必要となる。しかし重み付き仮説推論において、正解となる仮説の証明木そのものを教師信号として与えることは、問題それぞれに推論木を用意することの人的コストと、そもそも正しい仮説というものが複数通り存在し得ることの2点から、あまり現実的ではない。

そこで本稿では、正しい推論の中でコスト最小のものと、誤った推論の中でコストが最小のものをそれぞれ導出した上で、正しい推論の方がコストが小さくなるように重みを調整する。つまり、観測  $O$  に対して考えられる仮説の集合を  $\mathbb{H}_O$  とし、観測  $O$  を入力として与えて得られる仮説  $H$  がそのタスクにおいて正解か不正解かを判別する関数  $J(O, H)$  について、 $J(O, H)$  が真となるものの中で最もコストが小さいものを  $H_{pos}$ 、 $J(O, H)$  が偽となるものの中で最もコストが小さいものを  $H_{neg}$  とする。これらについて  $C_{O,W}(H_{pos}) < C_{O,W}(H_{neg})$  となればよいので、 $C_{O,W}(H_{pos}) - C_{O,W}(H_{neg})$  に対して単調増加であるような関数  $E(O, W) = f(C_{O,W}(H_{pos}), C_{O,W}(H_{neg}))$  を誤差関数として、この誤差が最小になるように学習すれば、正しい推論のコストがより小さくなるように学習が行われる。なお、本稿の実験では誤差関数として

$$E(O, W) = \frac{C_{O,W}(H_{pos}) - C_{O,W}(H_{neg})}{C_{O,W}(H_{pos}) + C_{O,W}(H_{neg})} \quad (12)$$

を用いている。

ところで、本稿における学習は、証明木を隠れ変数とみなすことにより、隠れ変数を含む識別学習とみることができる。そのような学習に対する手法としては、構造化SVMに対する適用 [15] などが提案されているが、本稿の手法は、重み付き仮説推論に対する適用例といえる。

### 3.5 正解を含む推論の導出

候補仮説の数は観測から仮説される要素仮説の数に対して指数的に増加する。そのため一般に仮説推論では、計算量の観点から、動的な手法によって解仮説を導出しており、考えられる候補仮説を全て列挙するなどといったことは行わない。そのような状況で、“正解を含む仮説の中

で最もコストの小さい仮説”を得るためにはどうすれば良いだろうか。ここでは、判定関数  $J(O, H)$  をリテラルの集合  $Y$  が仮説  $H$  の部分集合に含まれる時に真となる関数とし、その上で  $H_{pos}$  を求める方法を述べる。

観測  $\hat{O}$  を次のように定める。

$$\hat{O} = O \cup \{y^{s\infty} \mid y \in Y\} \quad (13)$$

$\hat{O}$  を観測とした時の解仮説  $H_{\hat{O}, W}^*$  は、次の条件を満たす限りにおいて  $H_{pos} = H_{\hat{O}, W}^*$  と等価である。

- リテラル  $y \in Y$  が持つコストが十分に大きい
- $Y$  を含む仮説が観測  $O$  の候補仮説の集合の中に存在する
- リテラル  $y \in Y$  に対して後ろ向き推論が行われない
- 観測  $O$  の候補仮説の集合のなかで、コストが  $C_{O, W}(H_{pos})$  となる仮説がひとつだけである

以上の条件が成り立つとき、 $y \in Y$  のコストが十分に大きいことから、 $H_{\hat{O}, W}^*$  として選ばれる仮説は、より多くの  $y \in Y$  が他のリテラルと単一化しているような候補仮説となる。よって、 $Y$  を含む仮説が  $O$  の候補仮説として存在することより、 $\hat{O}$  の候補仮説集合の中に、全ての  $y \in Y$  が単一化されているような候補仮説が存在し、 $H_{\hat{O}, W}^*$  としてもそのような仮説が選ばれる。このとき  $y \in Y$  のそれぞれのコストは、 $O$  から仮説されうどのリテラルよりもコストが大きいとすれば、 $O$  から仮説されたリテラルと  $y$  が単一化したときにコストが 0 になるのは必ず  $y$  の側となるため、 $C_{(\hat{O}, W)}(H_{\hat{O}, W}^*) = C_{(O, W)}(H_{O, W}^*)$  が成り立つ。よって、 $O$  の候補仮説集合のうちコストが  $C_{(O, W)}(H_{O, W}^*)$  となるものは一つに定まることから、 $H_{\hat{O}, W}^* = H_{O, W}^* \cup Y$  といえる。

なお、 $Y$  を含む仮説が仮説候補中に存在しない場合は、解仮説において  $y^{s\infty} \in \hat{O}$  が全て単一化されずにいずれかのコストが 0 にならずに残るため、仮説候補が条件を満たしているかどうかは、解仮説のコストを確認することで調べることができる。

### 3.6 学習の流れ

学習はアルゴリズム 1 に示すような手順で行われる。ここで、 $O$  は観測の集合、 $W$  は背景知識の重み、 $Y_O$  は観測  $O$  に対する正解のリテラルの集合を表す。

## 4. 実験

提案手法によって重み付き仮説推論のパラメタが学習されていることを確認するために、評価実験を行った。この章では、この評価実験の結果を報告する。

### 4.1 データセット

今回の実験では、Singla らが用いたものと同様のデータセットを用いた上で、彼らの結果をベースラインとした。このデータセットはプラン認識を対象タスクとしている。

### Algorithm 1 学習の流れ

```

1: repeat
2:   for all  $O \in \mathcal{O}$  do
3:      $H \leftarrow \text{Inference}(O, W)$ 
4:     if  $Y_O \not\subseteq H$  then
5:        $H_{neg} \leftarrow H$ 
6:        $\hat{O} = O \cup \{y^{s\infty} \mid y \in Y_O\}$ 
7:        $H_{pos} \leftarrow \text{Inference}(\hat{O}, W)$ 
8:        $E_{O, W} \leftarrow \text{ErrorFunction}(H_{pos}, H_{neg})$ 
9:       for all  $p \in P_{H_{pos}} \cup P_{H_{neg}}$  do
10:        if  $\text{cost}(p) > 0$  then
11:          assign gradient  $\frac{\partial E_{O, W}}{\partial \text{cost}(p)}$ 
12:        end if
13:      end for
14:      do backward-propagation
15:      do update parameter
16:    end if
17:  end for
18: until convergence

```

このタスクでは、例えば “Bill went to the liquor-store. He pointed a gun at the owner”、のような談話文に対して、そこから得られる観測を上手く説明するような登場人物の意図を推論することが目的である。例えばこの例では、“Bill が強盗する” のような意図が得られることを目指す。このとき、談話から得られる観測の論理表現は、それぞれの例で既に与えられているものとしている。データセットは、開発データ 25 個とテストデータ 25 個から成り、開発データは [4]、テストデータは Ng ら [8] によって作成されたものである。本稿では、開発データを訓練データとして用いて学習を行った。

背景知識は、Ng ら [8] が上記の開発データの推論を行うために作成したものを用いる。この背景知識は、高位の意図 (shopping, robbing など) とその格関係から、それらが含意する情報を推論するような形式で記述されている。背景知識中に含まれる意図は、shopping, robbing, restaurant dining, traveling in a vehicle (bus, taxi, or plane), party, jogging などがある。それぞれの意図は、それらを構成するような部分的な意図 (サブプラン) が存在し、そのうちいくつかは、再帰的にサブプランを含んでいる。例えば、“traveling in a plane” は、“(飛行機に乗るために) 空港へ (何らかの交通機関を用いて) 移動する” というサブプランを持っている。本実験ではこれらを我々の用いている推論エンジンに利用可能な形に変形し、その結果として 220 個の Horn 節からなる背景知識を用いて実験を行った。

図 1(a) に訓練データに含まれる談話の例とそこから得られる観測を、図 1(b) にこの談話の正解として与えられている論理式を示す。また、図 1(c) に背景知識の一部を示す。この例では、図 1(a) の観測に対して図 1(c) の背景知識を用いて仮説推論を行うことによって、ある shopping というイベントが存在して、その動作主が Bill であり、買ったものが Milk であることを推論することが求められる。

(a)  
“Bill went to the store. He paid for some milk”  
*instance\_going(GO1)*  
*goer(GO1, BILL)*  
*destination\_go(GO1, STORE)*  
*instance\_paying(PAY1)*  
*payer(PAY1, BILL)*  
*thing\_paid(PAY1, MILK)*

(b)  
*instance\_shopping(s)*  
*shopper(s, BILL)*  
*go\_step(s, GO1)*  
*pay\_step(s, PAY1)*  
*thing\_shopped\_for(s, MILK)*

(c)  
*instance\_shopping(s) ∧ go\_step(s, g) →*  
*instance\_going(g)*

*instance\_shopping(s) ∧ go\_step(s, g) ∧*  
*shopper(s, p) → goer(g, p)*

*instance\_shopping(s) ∧ go\_step(s, g) ∧*  
*store(s, str) → destination\_go(g, str)*

*instance\_shopping(s) ∧ pay\_step(s, pay) →*  
*instance\_paying(pay)*

*instance\_shopping(s) ∧ pay\_step(s, pay) →*  
*payer(pay, p)*

*instance\_shopping(s) ∧ pay\_step(s, pay) ∧*  
*thing\_shopped\_for(s, t) → thing\_paid(pay, t)*

表1 データセットおよび背景知識の例

## 4.2 実験手順

推論には井之上ら [17] の推論エンジンを用いた。これは、重み付き仮説推論を整数線形計画問題として解くことで、従来の仮説推論エンジンと比べて遥かに高速な推論を実現している。

正解の条件としては、正解に含まれるプランが解仮説に過不足なく含まれていれば正解とする。また、評価指標としては recall、precision、および F 値を用いた。recall は (正解したプラン数/正解に含まれるプラン数) として定義し、precision は (正解したプラン数/解仮説に含まれるプラン数) として定義する。これは Singla らの評価指標と同様のものである。

学習率等のパラメタは開発データに対するクロズドテストの結果に基づいて人手で調整した値を用いた。

## 4.3 結果

実験結果の反復回数ごとの推移を図 5 に、Singla らの手

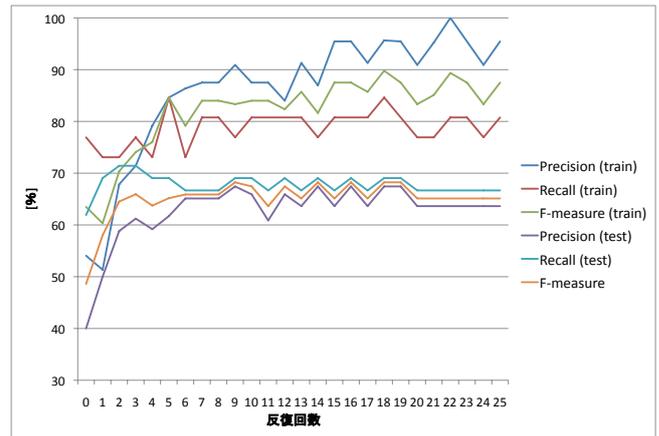


図5 反復回数ごとの実験結果

	Precision	Recall	F-measure
Singla+,2011	69.13	75.32	72.10
提案手法	65.12	66.67	65.88

表2 ベースライン手法との比較

法との比較を表 2 にそれぞれ示す。図 5、2 より、Singla らの結果には若干劣るものの、仮説推論のパラメタに対して学習が行われていることが確認できた。

今回の実験では、観測の仮説コストおよび背景知識の重みは、そのそれぞれに固有のコストおよび重みを割り当てていた。しかし、実問題に対する適用を考えると、背景知識や観測それぞれに固有のパラメタを割り当てていたのではパラメタの次元数があまりに多くなってしまうため、データセットの規模に応じてパラメタを汎化する必要があると考えられる。この点について今後は、観測や背景知識をそれぞれいくつかのタイプにグループ分けした上で共通した重みを持たせたり、パラメタを重みベクトルと特徴ベクトルの内積と定義した上で重みベクトルの学習に持ち込むなどといった方法を将来的には考えている。

## 5. 関連研究

冒頭で述べたように、仮説推論を談話解析に用いる試みはいくつもあるが [1], [9], [16], [17]、いずれの研究も人手により評価関数のパラメタを与えている。我々が知る限り、仮説推論の評価関数のパラメタを学習する手法は提案されていない。最も近いもののうちのひとつに、Markov Logic Networks (MLNs) [11] により仮説推論を表現し、仮説の評価関数を MLNs のパラメタ学習手法を用いて学習する試みがある [14]。しかしながら、MLNs は本来演繹推論のための枠組みであり、仮説推論の評価関数が間接的にモデル化されているため、厳密には正しく学習できていないと考えられる。一方で、我々の枠組みは仮説推論を直接モデル化しているため、より正確にパラメタ推定できていると考えられる。さらに仮説推論を直接モデル化していることのメリットとして、学習に必要な時間が少なくなることが期待

できる [7]。実際に、仮説推論を演繹推論に変換して仮説推論を行う手法は、大きなオーバーヘッドを伴うことが報告されている [14]。

## 6. 結論

これまで仮説推論は、自然言語処理において文章に明示されていない情報の顕在化を行うことに対して有用な枠組みと考えられている一方、仮説推論の評価関数の学習手法がこれまでは提案されておらず、評価関数のパラメタの調整は、人手による調整やヒューリスティックな手法に頼らざるを得ないという問題があった。本稿ではその問題の解決として、重み付き仮説推論の証明木からフィードフォワードニューラルネットワークを作成し、誤差逆伝播を用いることによって重み付き仮説推論のパラメタを機械学習によって調整する手法を提案した。また、既存のデータセットを用いて学習実験を行い、提案手法により重み付き仮説推論の評価関数が学習できていることが確かめられた。

今後は、実問題への適用へ向けて、本稿の手法をより拡張していく。ひとつは評価関数について、観測や背景知識をそれぞれいくつかのタイプにグループ分けした上で共通した重みを持たせたり、パラメタを重みベクトルと特徴ベクトルの内積と定義した上で重みベクトルを学習するなど、パラメタの汎化を行っていく必要がある。

また、重み付き仮説推論の枠組みそのものの拡張を行っていく予定である。例えば、談話解析では共参照解析が重要であるが、仮説推論においては共参照関係は単一化に対応する。そのため、仮説推論において共参照解析を行うためには、単一化に対する尤度を扱えるようにする必要があり、単一化に対してコストを課すなどの拡張を行い、それらに対応して学習の枠組みも拡張していく。

## 謝辞

本研究は、文部科学省科研費課題 22・9719、課題 23240018、および JST 戦略的創造研究推進事業さきがけの一環として行われた。

## 参考文献

- [1] Blythe, J., Hobbs, J. R., Domingos, P., Kate, R. J. and Mooney, R. J.: Implementing Weighted Abduction in Markov Logic, *Proceedings of IWCS*, Oxford, UK, pp. 55–64 (2011).
- [2] Chambers, N. and Jurafsky, D.: Unsupervised Learning of Narrative Schemas and their Participants, *Proceedings of ACL*, pp. 602–610 (2009).
- [3] Fellbaum, C.(ed.): *WordNet: an electronic lexical database*, MIT Press (1998).
- [4] Goldman, R. P.: A Probabilistic Approach to Language Understanding, PhD Thesis, Department of Computer Science, Brown University (1990).
- [5] Hobbs, J. R., Stickel, M., Martin, P. and Edwards, D.: Interpretation as Abduction, *Artificial Intelligence*, Vol. 63, pp. 69–142 (1993).

- [6] Inoue, N. and Inui, K.: ILP-Based Reasoning for Weighted Abduction, *Proceedings of AAAI Workshop on Plan, Activity and Intent Recognition* (2011).
- [7] Inoue, N. and Inui, K.: An ILP Formulation of Abductive Inference for Discourse Interpretation, Technical Report 3 (2011-09-09).
- [8] Ng, H. T. and Mooney, R. J.: Abductive Plan Recognition and Diagnosis: A Comprehensive Empirical Evaluation, *KR*, pp. 499–508 (1992).
- [9] Ovchinnikova, E., Montazeri, N., Alexandrov, T., Hobbs, J. R., McCord, M. and Mulkar-Mehta, R.: Abductive Reasoning with a Large Knowledge Base for Discourse Processing, *Proceedings of IWCS*, Oxford, UK, pp. 225–234 (2011).
- [10] Penas, A. and Hovy, E.: Filling knowledge gaps in text for machine reading, *Proceedings of COLING: Posters*, pp. 979–987 (2010).
- [11] Richardson, M. and Domingos, P.: Markov logic networks, *Machine Learning*, pp. 107–136 (2006).
- [12] Ruppenhofer, J., Ellsworth, M., Petruck, M., Johnson, C. and Scheffczyk, J.: *FrameNet II: Extended Theory and Practice*, Technical report, Berkeley, USA (2010).
- [13] Schoenmackers, S., Davis, J., Etzioni, O. and Weld, D.: Learning First-order Horn Clauses from Web Text, *Proceedings of EMNLP*, pp. 1088–1098 (2010).
- [14] Singla, P. and Domingos, P.: Abductive Markov Logic for Plan Recognition, *Proceedings of AAAI-11*, pp. 1069–1075 (2011).
- [15] Yu, C.-N. J. and Joachims, T.: Learning structural SVMs with latent variables, *ICML* (2009).
- [16] 杉浦 純, 井之上直也, 乾健太郎: 説明生成に基づく談話構造解析の課題分析, 言語処理学会第 18 回年次大会論文集, pp. 115–118 (2012).
- [17] 井之上直也, 乾健太郎, Ovchinnikova, E., Hobbs, J. R.: 大規模世界知識を用いた仮説推論による談話解析の課題と対策, 言語処理学会第 18 回年次大会論文集, pp. 119–122 (2012).