博士学位申請論文

仮説推論法と文書主題抽出法に関する研究

指導教官 石塚 満 教授

東京大学大学院 工学系研究科
電子情報工学専攻 博士課程 97124

松尾 豊

目 次

第1章　序論

1.1　はじめに

1956年のダートマス会議[1]で人工知能 (Artificial Intelligence) という語が生まれて以来，数多くの研究者が人工知能の研究に取り組んで来た．1960年代には探索を中心課題とした「推論中心の時代」，1980年代にはエキスパートシステムを中心とした「知識と推論の時代」，1990年代は大量のデータを処理する「データの時代」であった．最近ではインタラクションによる知性，身体性，知的増幅器 (Intelligent Amplifier) としてのAIなどの新しい方向性が生まれている．

本論文では，まず記号の処理による推論の枠組である仮説推論法と，その高速推論法について述べる．仮説推論は，事象の原因を探る推論である．人間は，事象の原因を探るような思考を行う．「道が濡れている．ゆうべは雨が降ったのかな?」「今日はよく怒っている．おなかが空いているのかな?」 大量の知識が与えられていれば，このような推論も可能である．しかし，大量の知識を処理する際には，その推論速度が大きな問題になる．本論文の前半では，仮説推論の高速化というテーマを中心に，仮説推論問題の分析や仮説推論を含んだシステムについて考察する．

一方で，大量の知識を人間が記述するのは困難である．何より，知識とはある視点，目的のための事象の記述であるから，一般的に用いることのできる知識を大量に記述するのは少なくとも非常に難しい．最近では，電子的な文書が大量に存在し，その中から知識を発見するテキストマイニングの研究も盛んである．本論文の後半では，文書からキーワードを抽出する手法について述べる．キーワードを抽出することは，今後，文書からの知識抽出のひとつの基礎技術となるだろう．

[1] M. Minsky, J. McCarthy, H. Simon, A. Newell, C. Shannon らが参加した．

1.2 本論文の構成

本論文は大きく，仮説推論に関する研究と，自然言語文からのキーワード抽出に関する研究の2つの部から構成される．大量の知識があったときにいかに高速に推論するかと，その大量の知識をどのように獲得するかという密接に関連した2つの問題を扱っている

1章：序論

第I部

2章：人工知能と仮説推論 人工知能研究の概観と仮説推論について述べる．特に，仮説推論が他の知識表現とどのような関係にあるか，そして，高速推論法としてどのような手法が提案されてきたかを述べる．

3章： 仮説推論の最適化問題への置き換え 現在まで，仮説推論を最適化問題に置き換える方法がいくつか提案されてきたが，それらは基本的に2つの置き換え法にまとめることができる．本章では，この2つの置き換え法について述べる．

4章： SL法：線形計画法と非線形計画法の併用による高速推論法 仮説推論を線形計画法に変換し，探索の初期点を得た後，非線形関数の勾配法によって解を探索するアルゴリズム (SL法) について述べる．

5章： 仮説推論の問題例に対する分析 これまで仮説推論の問題に対する分析はあまり行われてこなかった．本章では，問題を解く前に実行可能解が存在するかどうかを見積もる方法と，制約の付加による問題の難易度の変化について述べる．

6章： 2種の置き換え法の協調による推論法 仮説推論から制約への2つの置き換え法には，それぞれ特徴がある．拡張ラグランジュ法を用い，2つの置き換え法をうまく協調させることで，柔軟で高速なアルゴリズムを構築する．

7章： システムとしての仮説推論 使いやすい仮説推論システムのためには，どのような拡張が必要か，その可能性と問題点について議論する．

第II部

8章： **自然言語文からの知識獲得** 近年，大量の電子的な文書から知識を抽出するテキストマイニングやWebマイニングの技術が進展している．本章では，テキストマイニング，Webマイニング，そしてキーワード抽出の諸手法を概観する．

9章： **語の共起情報を用いたキーワード抽出** 自然言語のテキストからキーワードを抽出する方法を述べる．語の共起を統計的に処理し，共起の偏りを検出することで，単独の文書から高い性能でキーワードを抽出する．

10章： **Small World構造に基づくキーワード抽出** 語の共起は全体としてグラフを形成する．このグラフはSmall Worldと呼ばれる特徴を持つ．本章では，このSmall Worldの特徴に基づいて，構造上重要な位置にある語をキーワードとして抽出する．

11章： **仮説推論を利用した複数文書要約** 複数の文書の語の共起グラフを組み合わせると，より大きな語の共起グラフができる．このグラフを利用した複数文書の要約技術について述べる．辺被覆問題を解く際に，仮説推論を利用する．

第III部 結論

11章： **おわりに** 本論文の結論と，今後の研究課題について述べる．

第I部

仮説推論法

第2章　人工知能と仮説推論

2.1　仮説推論とは

人工知能研究が目指す知的機能として、通常の演算・検索や、知識ベースの簡単な処理としての演繹の枠を越えた、認識・連想・類推、帰納・学習、発想・創造など、高次人工知能技術のさまざまな枠組みが研究されている．その中でも仮説推論（またはアブダクション）は、あることが観測されたときに、その原因を見い出す（生成する）推論形態であり、発想や創造といった高次人工知能へ向けての共通の基盤技術となることが期待される．後に否定される可能性のある不完全な知識を扱うことができる仮説推論は、より柔軟な知識処理が可能であり、設計・診断問題など実用的問題に直接適用できるという特長がある．また、仮説推論は、自然科学をはじめ、広い範囲の学問領域での基本的な推論過程であり、近代工学においてもその重要性が指摘されている [吉川].

仮説推論は非単調推論である．つまり，知識ベースに新たな知識を追加することにより、導出される帰結が単調に増加しない．以前導けていた帰結が導けなくなることがあるような推論系である．したがって，仮説推論は組み合わせ的な探索を必要とし，その高速化が実用上最大の問題となる．この高速化が仮説推論に関する重要なテーマのひとつである．

2.1.1　仮説推論の枠組み

仮説推論 (hypothetical reasoning) は、真か偽か不明な事柄をとりあえず真と考えて（仮説を立てて）推論を進め、矛盾なくうまく与えられた問題が解決できたり、矛盾なくゴール（観測された事実）が説明できれば、立てた仮説は正しかったと考える形式の推論である．仮説推論と類似する言葉として、**アブダクション** (abduction: **仮説生成**) という言葉が用いられることも多い．仮説推論が、あらかじめ定められた生成可能な仮説の候補

から適切な仮説を選ぶ推論であるのに対し、アブダクションは仮説を生成する場合を含む、より広い範囲を指す [石塚 96].

アブダクションという用語を最初に用いたのは、哲学者の Peirce(1839-1914) であると言われている．Peirce は、彼の研究の初期段階で、推論の三分法として演繹、帰納、仮説推論を区別しその違いを指摘した [井上 92].

例：
（大前提）電気系の学生は、今日、論文輪講がある．
（小前提）太郎は電気系の学生である．
（結論）太郎は、今日、論文輪講がある．

演繹 (deduction) は、大前提と小前提から結論を導く．用いる知識が正しければ、得られる結論は必ず正しい、安全な推論系である．**帰納** (induction) は小前提と結論から大前提を求める．いろんな学生を観察していると、電気系の学生は、今日、論文輪講[1]がある人が多い．どうやら電気系の学生は、今日、論文輪講があるのではないか、と推論するのが帰納である．サンプルが多くなればなるほど、その信頼性は上がる．

一方、仮説推論は、結論と大前提から、小前提を求める問題である．電気系の学生は、今日、論文輪講があり、太郎も今日輪講がある．もしかして、太郎は電気系の学生ではないか、と考える．

帰納と仮説推論の違いは分かりにくいが、Peirce によれば、帰納は観測された事例からそれらの事例が属するクラス全体について一般化を行う推論である．それに対し、仮説推論は、観測原因とは違う種類の原因を追求し、事例の中から直接的に観測できない現象の可能性について推論する．例えば、内陸部から魚の化石が発見されたときに、そこが過去に海であったという仮説を立てることは仮説推論である．しかし、太古の状態を現実に観測することは不可能であり、帰納からは説明できない．仮説推論は、「新しい概念を提供する唯一の形態であり、その意味で唯一の綜合的な推論形態」である．

仮説推論は、論理的には後件肯定の虚偽とよばれる妥当でない (invalid) 推論である．上の例では、太郎は、電気系の教授かもしれないし、たまたま論文輪講を聞きにきた人かも

[1]東京大学大学院の電気系各学科では，毎週金曜日の午前中に論文輪講という必修科目がある．

しれない．このように、ある観測を説明するのにいくつもの仮説が考えられるのが一般的である．そして、仮説が正しいことは、仮説から演繹により得られた帰結が、事実に適合するかどうかを帰納的に検証することによって確かめられる．この仮説生成–演繹–検証というサイクルの形成は、科学における探求方法でもある．

2.1.2 仮説推論の基本的な推論動作

知識ベースには、対象とする領域で常に成り立つ**背景知識**の集合Σと、矛盾の可能性を有する**生成可能な仮説の集合**Hを記述する．

基本的な推論動作は以下のようになる．

ゴールGが与えられたとき、まずΣからGが演繹的に証明できるか確かめる．Σからだけでは証明できないとき、仮説の集合Hの中から次の条件を満たすhを切り出す（図2.1参照）．

$$\Sigma \cup h \vdash G \tag{2.1}$$

$$\Sigma \cup h \text{は無矛盾} \tag{2.2}$$

これを満たすhを、Gの**説明 (explanation)** もしくは**解仮説**という．また、Gの説明hが極小 (minimal) であるとは、hのいかなる真部分集合も説明でないことをいう．

例えば、「電気系の学生は論文輪講に出る」「教授は当番だと出る」「論文輪講に出ると仮説推論を知っている」「AIの研究者は仮説推論を知っている」をΣとして次のように表す．

$$\begin{aligned}
know_abduction(X) &\leftarrow attend_rinko(X).\\
know_abduction(X) &\leftarrow ai_reserchar(X).\\
attend_rinko(X) &\leftarrow student(X).\\
attend_rinko(X) &\leftarrow professor(X) \wedge duty(X).\\
false &\leftarrow student(X) \wedge professor(X).
\end{aligned} \tag{2.3}$$

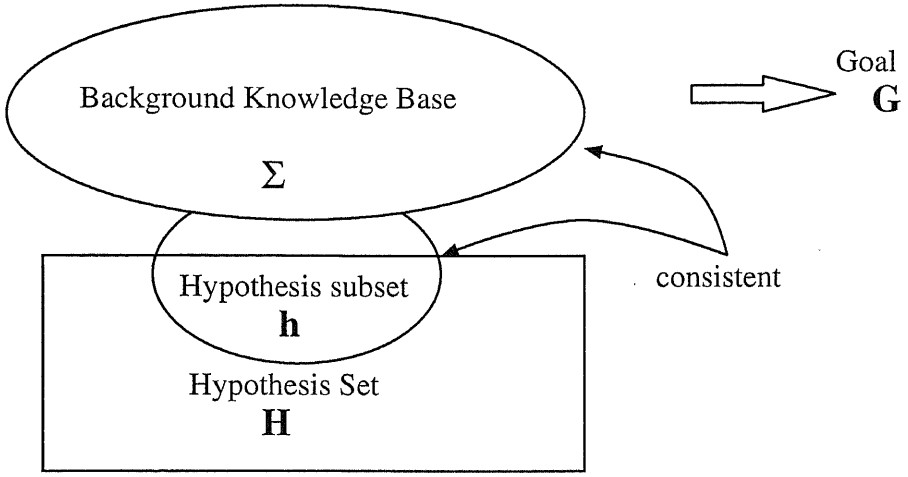


図 2.1: 仮説推論

H は、以下である.

$$\{ \quad student(X), \quad professor(X), \quad duty(X), \quad ai\_resercher(X)\}$$

ここで、$know\_abduction(X)$ は、「X が仮説推論を知っている」ということを表す. 式 (2.3) は、「X が電気系の学生であって、しかも電気系の教授であることは矛盾している」ことを表している.

さて、太郎が仮説推論について知っていたとしよう. つまり

$$know\_abduction(Taro) = true$$

が与えられたとする. この観測を説明する仮説、すなわち $know\_abduction(Taro)$ の説明は、例えば、$\{student(Taro)\}$ や $\{ai\_resercher(Taro)\}$、さらに、$\{professor(Taro), duty(Taro)\}$ などとなる. これらはすべて、極小の説明である.

このように、仮説推論において、ゴールを説明する仮説は１つであるとは限らず、いくつもの可能性が考えられるのが普通である. そこで、複数の仮説の中から合理的なものを選択する**仮説選択**の問題も重要である. 例えば、

- 生成できる仮説を限定する. また、解となる仮説が、一定の制約 (integrity constraint: 整合性制約) を満たすことを要請する.

- 最もシンプルな仮説を優先する.(Occam の剃刀と呼ばれる.)

などの基準がある.これらを一般化した仮説推論の枠組みが、解仮説を定量的に評価する**コストに基づく仮説推論** (cost-based abduction)[Charniak 90] である.

コストに基づく仮説推論では、ゴールを満たす解仮説の中でコストの小さいものを優先する.あらかじめ、各仮説、ルールに重みを設定しておき、用いる仮説、ルールのコストの和を解仮説のコストとする.仮説 h_i が真となるコストを $cost(h_i)$ とおくと、解コストは以下のようになる[2].

$$C = \sum_{i \in h} cost(h_i) \tag{2.4}$$

例えば $cost(student(X)) = 3$, $cost(duty(X)) = 2$,$cost(professor(X)) = 10$ とすると、解 $\{student(Taro)\}$ のコストは3、$\{professor(Taro), duty(Taro)\}$ のコストは12となり、前者が望ましい説明であるということになる.

確率的な観点から見ると、仮説 h_i が真となる確率を p_i とすると、重みは、

$$cost(h_i) = -\log p_i$$

で表されることになる.

仮説推論は、知識ベースに新たな知識が加わったとき、以前導けていた帰結が導けなくなることもありうるという、非単調推論の一種である.このような不完全な知識を扱う非単調推論系には、デフォルト推論 (default logic)、サーカムスクリプション (circumscription: 極小限定) などがあり、これらの関係も明らかにされている.仮説推論は、その中でも、最もシンプルな形式を有している.

ここでは,引数を持つ述語記号を用いた述語論理の仮説推論を紹介した.しかし,本論文で主に扱うのは引数を持たない命題論理の仮説推論である.また,扱う節は,ホーン節形式とする.ホーン節とは、たかだか1つの正のリテラルから成る節,つまり、p(ファクト型) や $p \leftarrow q \wedge r.$、また $false \leftarrow q \wedge r.$ などのタイプのルールである.

以下,述語論理を命題論理に変換する手法について概説する.

[2]「あるルールを適用する」という仮説も考えることができる.このルール型の仮説も,条件部に新たな仮説を追加することで実現できる.

2.1.3 述語論理仮説推論から命題論理仮説推論への変換

命題レベルの推論は，多くのAI問題の基盤となるが，知識表現能力の点では必ずしも十分ではなく，述語論理が必要になることが多い．述語論理の変数はエルブラン領域に展開すれば，定数(基礎項)とすることができるので，述語論理表現を命題論理表現へと展開できることになる．例えば，$student(X)$という述語に対し，Xとして取る可能性のある具体値が$\{taro, jiro, ichiro, saburo\}$であれば，$student(taro)$, $student(jiro)$, $student(ichiro)$, $student(saburo)$と表記し命題と考えればよい．

しかし，引数が複数ある場合や，引数が異なる述語による論理式の場合には，例え簡単な述語論理表現でも，命題論理式に直すと数100，数1000倍と膨大な数になってしまう．そのため，述語論理を効率的に命題論理に変換することが必要となる．

棚橋らは，述語論理式を適切に変形することで，生成される命題論理式を減らす手法を提案した[棚橋 99]．述語ホーン節のリフォメーションを行い，QSQR法[3]によりゴールの証明に関与する部分のみを抽出して命題化し，命題論理版仮説推論問題に変換する．例えば，次のような知識ベースがあるとする．

$$g(X,Z) \leftarrow p(X,Y), q(Y,Z)$$
$$q(Y,Z) \leftarrow r(Y), s(Z)$$

各変数が3種類の値を取る可能性があるとすると，第1の論理式に関しては命題論理式が$3^3 = 27$種類，第2のルールに関しては$3^2 = 9$，合計36種類生成される．一方，この2つの式をunfoldし，

$$g(X,Z) \leftarrow p(X,Y), r(Y), s(Z)$$

とすると，命題論理式の数は27となる．さらにゴールが$g(a,b)$である場合には，$X = a, Y = b$と束縛することができるので，命題論理式の数は3個になる．他にも，論理式を分離した方が生成される命題の数が少なくなる場合には，分離するfoldingフェーズも行う．

[3]ボトムアップに推論する状況を考える．推論の過程で真とされた述語を状態Qとして保持する．状態Lには，IDB述語間の補題を保持する．すなわち，サブゴールを保持することで，不要な推論を抑制する手法である．

また，Prendingerらは，まずゴールに関連した部分だけを抜き出すRelevance Reasoiningという処理を行い，その後，次のような規則を適用する手法を考案した[Prendinger 00].

Block ブロックにより論理式を分割することにより，具体化される節の数を減らす．例えば，

$$q(X,Y) \leftarrow p1(X,Z1,Z2) \wedge p2(Z1) \wedge p3(Z2,Z)$$

という論理式の述語を$p1(X,Z1,Z2), p2(Z1)$と$p2(Y,Y1)$の2つのブロックに分け，

$$\begin{aligned} newq(X,Z2) &\leftarrow p1(X,Z1,Z2) \wedge p2(Z1) \\ q(X,Y) &\leftarrow newp(X,Z2) \wedge p3(Z2,Y) \end{aligned}$$

Chain 変数のつながりを見つけて，論理式を分割する．例えば，

$$q(X,Y) \leftarrow p1(X,Z1) \wedge p2(Z1,Z2) \wedge p3(Z2,Y)$$

という論理式があったとき，これを

$$q(X,Y) \leftarrow newp(X,Z2) \wedge p3(Z2,Y)$$

$$newp(X,Z2) \leftarrow p1(X,Z1) \wedge p2(Z1,Z2)$$

に分ける．

Isolated blockpart 孤立した変数があるとき，論理式を分割する．例えば，

$$q(X,Y) \leftarrow p1(X,Z) \wedge p2(Z,Y) \wedge p3(Z,Z1)$$

は，

$$\begin{aligned} newp(Z) &\leftarrow p3(Z,Z1) \\ q(X,Y) &\leftarrow p1(X,Z) \wedge p2(Z,Y) \wedge newp(Z) \end{aligned}$$

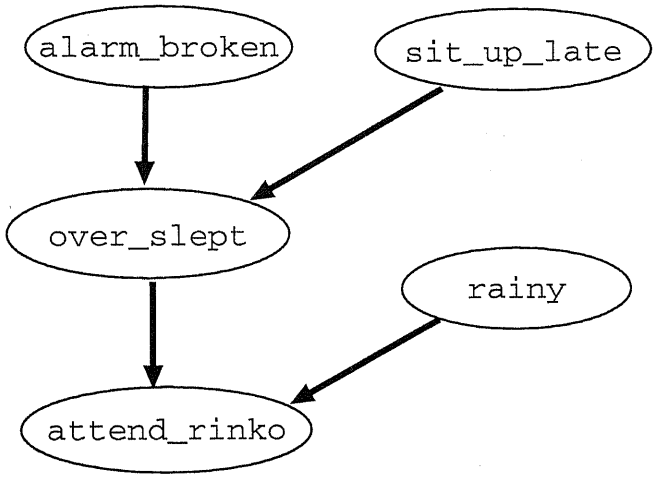
に分ける．

このような方法を用いることで，特定の述語論理のルールに関しては，効率的に命題論理に変換できる．しかし，このような効率化を図っても，一般的には定数の数の多項式オーダの命題論理節が生成されることになる．

2.2 他の知識表現との関係

不完全な知識を扱う仮説推論は，他の知識表現とも密接に関係がある．以下では，Bayesian ネットワークおよびデフォルト推論との関係を述べる．

2.2.1 Bayesian ネットワークとの関係



CPT for over_slept

alarm broken	sit_up late	P(over_slept)
T	T	0.6
T	F	0.3
F	T	0.1
F	F	0.01

T...True, F...False,
P()is the probability to be true.

図 2.2: ベイジアンネットワーク

Bayesian ネットワーク[4]は、確率変数の因果的な関係を表現するための有向非循環グラフ (DAG) である．確率変数がネットワークのノードを形成し、ノードＸからノードＹのリンクは、ＸがＹに直接的影響を及ぼすということを表す．各ノードは、親ノードがある値を取ったときに、真となる確率を示す条件付き確率表 (Conditional Probability Table: CPT)

[4]信念ネットワーク (belief network) とも呼ばれる．

を持つ.

各ノードは親ノードのみに依存し、他の祖先ノードからは条件付き独立 (conditional independence) であると仮定することにより、知識をコンパクトに表すことができる. 確定的な論理とは異なり、確率を自然な形で扱うことができるため、実用性に優れており、特に海外で研究が盛んである.

例を図 2.2 に示す. 目覚まし時計が壊れている (*alarm _broken*)、遅くまで起きていた (*sit_up_late*)、寝過ごした (*over_slept*)、雨が降っている (*rainy*)、論文輪講に出席する (*attend_rinko*) の間の確率的な因果関係を示している.

Bayesian ネットワークにおいて、証拠をもとに条件付き確率 P(質問 | 証拠) を求めることにより、推論システムを構成することができる. 証拠、質問として、どのようなものを選んでもよいことが Bayesian ネットワークにおける推論の優れた点であるが、このうち、証拠をある観測とおき、その原因を求める推論は、仮説推論と類似の働きをする. ある観測が得られたときに、原因の最大事後確率解 (MAP 解: maximum a posteriori) を求める問題は、コストに基づくアブダクションと密接な関連がある.

仮説推論と Bayesian ネットワークは異なる表現形式であるが, その関係を考えることにより、仮説推論のコストの確率的な意味を明確にする、相互のアルゴリズムを共有できるようになる, などのメリットがある. また, Bayesian ネットワーク上の MAP 解を見つける問題は, コストに基づく仮説推論問題に帰着することができる. 仮説推論の方がヒューリスティック手法を用いやすいため, その高速解法により Bayesian ネットワークの MAP 解を高速に得ることもできる.

D. Poole は、ホーン節述語論理による仮説推論を、確率を含めて拡張した、確率つき仮説推論という新たな言語を提案し、Bayesian ネットワークと相互に変換できることを示した [D.Poole 93]. 確率つき仮説推論の要点は以下である.

- 一組の仮説に確率を割り当てる. 例えば仮説 h1 が真となる確率が 0.8、仮説 h2 が真となる確率が 0.2 であるとすると、disjoint([h1:0.8, h2:0.2]) と記述する. 同じ組の仮説は、排他的であると仮定する. つまり $false \leftarrow h1 \wedge h2$. である.

- さらに、ホーン節知識において、仮説がヘッド部にこないこと、cycle がないこと、知識は完備化が行われていることなどの仮定を置く.

このように設計された言語上では、確率つき仮説推論と Bayesian ネットワークに写像が存在し，述語論理表現を Bayesian ネットワークで表現することができる．しかし、そのノード数は問題の大きさに関して指数的に増加してしまい、本質的な問題の難しさは変わらない.

また、E. Charniak は、Bayesian ネットワークを、仮説推論問題のグラフ表現の拡張である Boolean 関数 DAG(WBFDAG) に変換し、MAP 解を見つける問題と WBFDAG 上の最小コストを見つける問題が等価であることを示した [Charniak 94]．そして、最良優先探索を用いて、MAP 解を見つける手法を示している.

これらの手法は、確率を扱う Bayesian ネットワークと仮説推論と結び付けようと、仮説推論を確率を扱える形で設計し直したものである．そのため、もとのコストに基づく仮説推論と Bayesian ネットワークの直接の等価性はない.

一方、Abdelbar は、Bayesian ネットワークを、命題論理のコストに基づく仮説推論に変換する（一方向の）線形時間アルゴリズムを提案している [A.M.Abdelbar 98]．ここでは、図 2.2 に示した例を用いて、仮説推論問題に変換する方法を示す．まず、*over_slept* に値 T または F が割り当てられていることを、$over\_slept_T$ または $over\_slept_F$ で表す．いずれかの値が割り当てられていることを、$over\_slept_D$ と表す．すると、

$$over\_slept_D \leftarrow over\_slept_T \vee over\_slept_F.$$

である．次に、このような変数を使って、CPT を表現する．以下の式は図 2.2 の CPT の第 1 行目である.

$$over\_slept_T \leftarrow alarm\_broken_T \wedge sit\_up\_late_T \wedge o_T a_T s_T.$$

ここで、$o_T a_T s_T$ は形式上の仮説であり、その重みは、

$$cost(o_T a_T s_T) = -log(0.6)$$

である．同様に、

$$over\_slept_T \leftarrow alarm\_broken_T \wedge sit\_up\_late_{\mathbf{F}} \wedge o_T a_T s_{\mathbf{F}}.$$

$$cost(o_T a_T s_F) = -log(0.3)$$

などと、順次CPTの各行をルールに変更していく．また、すべてのノードの値が決定されていなければいけないので、

$$true \leftarrow over\_slept_D \wedge alarm\_broken_D \wedge ... \wedge attend\_rinko_D.$$

である．これで、Bayesianネットワーク上の知識はすべて論理式で表現することができた．あとは、ある証拠が与えられたとき、例えば、$attend_rinko = false$が与えられたとき、コストを最小とする$attend_rinko_F$の説明を求めれば、もとのBayesianネットワーク上でのMAP解、すなわち、各確率変数への最も確からしい値の割り当てが得られる．

2.2.2 デフォルト推論との関係

デフォルト推論 (default reasoning) は1980年にR. Reiterによって提案された非単調論理による推論系のひとつであり、常識的な推論に用いられる一般的な枠組みである．述語論理を拡張した論理であり、仮説推論と密接な関係を持っている．

例えば、ある角度から自動車を見たとき、3つのタイヤしか見えないとしよう．それにも関わらず、我々は、タイヤは4つあると確信している．もし、反対側に回って、車がジャッキで持ち上げられているのを見れば、今度はタイヤは3つであることを確信するだろう．デフォルト推論は、このように、矛盾を生じない限り、デフォルトと呼ばれる一般法則を適用していく推論法である．

デフォルトは形式的には次のように表される．

$$\frac{\alpha(x) : \mathbf{M}\beta_1(x) \ldots \beta_n(x)}{\gamma(x)}$$

これは、「もし、$\alpha(t)$が成立しており、$\beta_1(t) \ldots \beta_n(t)$が矛盾しなければ、$\gamma(t)$が成立する」と解釈する．デフォルト理論は、このようなデフォルトの集合Dと、一般的な論理式で表された事実の集合Σで構成される．Σと矛盾を生じないDの極大の部分集合を切り出し、これとΣから帰結できる結論の集合が**拡張 (extension)** と呼ばれる．仮説推論における説

明と対照的なのは、極大である点（仮説推論における「説明」では極小性が望ましい）、およびΣとの帰結を拡張と呼ぶ点（仮説推論における「説明」は仮説のみからなる）である.

デフォルト$\alpha(x)$が$true$であるとき無前提とよび、デフォルトが$\alpha(x) : M\gamma(x)/\gamma(x)$の形のとき正規という.

ここで無前提正規デフォルトによる例を考えてみる.「私は、仮説推論に関する文書は興味がある．しかし、難しいのは嫌いだ.」をデフォルトを用いて表すと次のようになる.

$$\Sigma = \{interesting(X) \leftarrow abduction(X) \wedge int\_abduction(X).$$

$$false \leftarrow difficult(X) \wedge interesting(X).\}$$

$$D = \{int\_abduction(X)\}$$

ある文書Aが、abductionについて書かれていた場合、得られる拡張は、

$$\{abduction(A), int\_abduction(A), \mathbf{interesting}(A)\}$$

となる．したがって、文書Aが面白い、ということが推論できる．また、ある文書Bが、abductionについての難しい文書であるとき、得られる拡張は、

$$\{abduction(B), difficult(B)\}$$

となる．$int_abduction(B)$も真にすると矛盾が発生するので、この2つが帰結できるすべてである．したがって、文書Bは面白くない、ということが分かる.

仮説推論では、gが与えられているときに、それを説明する仮説を選択したが、デフォルト推論では、gの真偽が分からないときにそれを決定する．デフォルト推論は、何かが例外であるという事実を付け加えると、以前帰結していたものが帰結できなくなる可能性があるため非単調推論である.

デフォルト推論と仮説推論の最も大きな違いは、デフォルト推論は、ある証拠が与えられたときに原因となる仮説（デフォルト）を選定するだけでなく、ある事実が与えられたときにそれから何が予測(predict)できるか、という問いに答えることができる点である[D.Poole 98]．ある仮説がすべての拡張に含まれれば、その仮説を真と考えることをskeptical

default reasoningといい、あるひとつの拡張に含まれれば仮説を真と考える推論をcredulous default reasoningという．Pooleらによる非単調論理推論システムであるTheorist[Poole 87]は、仮説推論による説明の生成とデフォルト推論による予測の両方を用いたシステムである．

さて、無前提正規デフォルト理論と仮説推論には深い関係があり、gが仮説推論(Σ, H)からの説明を持てば、かつそのときに限り、gは対応する正規デフォルト理論(Σ, D_H)のある拡張に含まれる[5]、という定理が知られている[井上 92]．

一方、Selmanはデフォルト理論と仮説推論の計算複雑度は、同じ問題に由来することを明らかにした[Selman 96]．Support Set Selectionと呼ばれるサブタスクが、デフォルト推論および仮説推論の核心部であり、NP完全性を引き起こしているとしている．

ここで、Support Selection Taskとは、ゴールをGとしたとき、仮説推論における極小性を要求しないGの説明(support set)を求める問題であり、この問題自体がNP完全である．そして、この結果得られたsupport setを極小化すれば、仮説推論における説明が得られ、極大化すれば（さらにΣとの帰結を得れば）デフォルト推論におけるGを含む拡張が得られることになる．極大化、極小化の処理は多項式オーダで実行できる．すなわち、仮説推論における説明やデフォルト推論における拡張を得る際に本質的であるのは、このsupport setを見つけることといえる．

2.2.3 SATとの関係

命題論理の仮説推論より一般的な枠組みとして、与えられたすべての節を充足させるような変数への真偽の割り当てを求める、SAT(satisfiability)問題がある．例えば，次のような節が与えられたとき，

$$a \vee b$$

$$\bar{a} \vee c$$

$$\bar{b} \vee c$$

$$\bar{a} \vee \bar{b} \vee \bar{c}$$

[5] D_Hは、仮説推論の仮説集合Hを、無前提正規デフォルトで置き換えたデフォルト理論である．

解は，$a = true, b = false, c = true$ または $a = false, b = true, c = true$ であり，このときすべての節が真となる．

仮説推論は，ホーン節で表されるから，SAT として表現することも可能である．主な違いは，

- SAT は，ホーン節だけではなく、一般的な節を扱う．
- すべての変数の真偽を定めることができる．
- 解の間の選好順序は（普通は）考えない[6]．

である．このうち，コストに基づく仮説推論との最も大きな違いは，3 番目の解の選好順序があるかどうかである．Support Selection Task を SAT で表現し，SAT のアルゴリズムを用いることは可能だが，そこから，例えばコストの低い解をどのように見つけるかは難しい．

SAT 問題は、推論、各種計画、プランニングなど、幅広い用途を持つことから、海外では研究がさかんである．さらに SAT を一般化した問題として CSP(Constraint Satisfaction Problem) がある．これは、各変数がとれる値を多値にし、制約を 2 つの変数間だけで表したものである．

SAT や CSP については，さまざまな高速解法が提案されているため，参考になるものが多い．これらについては，2.4 節で述べる．

2.3 仮説推論の応用事例

仮説推論は、観測した事実の原因を求める枠組みであるから、その自然な応用として、診断や設計問題がある．

診断では、観測の結果得られたシステムの入出力をゴールとし、どの部分が壊れているかという解を見つける．医療における診断では患者の症状から原因 (病気) を推論する．例

[6]充足する節の数を最大にするという MAX-SAT という枠組みもある．

えば，予測された症状と観測された症状が異なっている場合に，それを説明するような仮説を見つける [Reiter 87].

また、設計問題では、設計仕様をゴール、要素を仮説とし、要素をどのように組み合わせればよいかを求める．診断では、直接観測はできないが、実際にある事象を見つけるのが目的であり、コストの確率的な意味が重要となるが、設計では、ユーザの好みを反映するようにコストを設定すればよい．電子回路などの設計問題への適用例には，[Finger 87, 牧野 90] がある．また，決められた条件を満たすようなスケジュールを作成する例 [河原 98] もある．

仮説推論を自然言語理解に応用した例 [Hobbs 93] もある．ここでは，文を手がかりに、筆者の意図を推論する．また，画像認識では，画像をもとにそれが何であるかを推論する [辻野 97],

文書検索への適用例としては，次のようなものがある．[大澤 98b] では、ユーザからのキーワードを観測とし、それを説明するようなユーザの興味を推論し、その興味に適した文書を提示するシステムを構成している．[松村 99] では，ユーザの求める知識をうまく満たすような文書の組み合わせを仮説推論を用いて求めている．

2.4 高速推論法に関する従来研究

真とすることのできる仮説の集合 H があらかじめ与えられている命題論理表現の仮説推論で、ゴールを説明する仮説を見つける問題はNP完全[7]である．また、コストが最小の仮説を見つける問題はNP困難となる [Eiter 95]．すべてのリテラルが生成可能でも、矛盾を表すルール (inc(inconsistency) ルール) を含む場合はNP完全である．したがって、コストに基づく仮説推論の最適解を見つけるには、最悪ケースで問題規模に対して指数オーダの推論時間がかかる．

このため，仮説推論を実際に用いる際には，推論速度が大きな問題になる．現在までに

[7]アルゴリズムに選択分岐があり、その分岐がうまく行われたときに有限回の選択で必ず解に到達できるようなアルゴリズムを非決定性アルゴリズムという．さらに、問題の規模 n に対し、多項式オーダの計算量で解を得ることができるアルゴリズムを多項式オーダのアルゴリズムという．多項式オーダの非決定性アルゴリズムによって解を得られる問題のクラスを NP(Non-deterministic Polynomial) という．問題 X が、(a) クラス NP に属し、(b) かつ NP に属するどのような問題も X に多項式時間で帰着させることができるとき、問題 X は NP 完全であるという．条件 (b) だけのとき、問題 X は NP 困難であるという．NP 困難な問題は、NP 完全問題を包含するクラスであり、より難しい．

知識構造を利用したさまざまな解法が提案されてきた．しかし，厳密に最適解を求めようとする限り、指数オーダの計算時間がかかることは避けられないので，実用上十分である最適解に近い準最適を高速に得る方法が研究されている．

ここでは，まず，記号処理的なアプローチを述べ，その後，数理計画法により準最適解を求めるアプローチを述べる．

2.4.1 知識ベース操作による高速推論法

記号処理的なアプローチとしては，推論パスネットワークを用いたKICK-SHOTGANと呼ばれるシステム[伊藤 91]をはじめ，さまざまなアプローチが試みられている．

推論パスネットワーク [伊藤 91]

非効率的なバックトラックを回避し，かつ計算コストの高い仮説合成(データベースの結合演算に相当する)の回数を最小にすることを実現したもので，Prologによるシステムに比べ，1000倍以上の推論測度を達成している．

推論パスネットワーク形成フェーズと仮説合成フェーズという2つのフェーズから成る．まず，推論パスネットワーク形成フェーズでは，仮説を除外した背景知識により，ゴール指向の推論パスネットワークを形成し，そのリーフノードに仮説を関連づける．各ノードは，“真”，“偽”，“仮説により真”のいずれかの状態をとり，また各ノードは“仮説により真”である状態を成立させるために必要な仮説の集合を格納する仮説ボックスを有する．仮説合成フェーズでは，形成された推論パスネットワークに沿って，前向きにゴールの証明に必要な仮説を合成する．

また，述語論理に対して，QSQR法を用い高速化を図ったものがKICK-HOPE[近藤 94]である．無駄な処理を省くことでこれを高速化した研究[越野 01]も行われている．

類推による高速化 [阿部 92]

類推による高速化を図る方法 [阿部 92] では，過去に成功した類似性のある問題の推論を以降の推論で利用することで，推論速度の向上を実現するる手法がある．この基本的な考え方は以下の通りである．過去にゴール G' に対して解仮説 $(h_1 + h_2)$ が求められたとする．つまり

$$\Sigma \cup (h_1 + h_2) \vdash G'$$

ということで，G' に対する解仮説集合 $(h_1 + h_2)$ として登録する．新たなゴール G が G' と似ている場合 G の解仮説集合は G' の解仮説集合と共通部分をもつと予想されるのでこの部分を (h_1) とし

$$\Sigma \cup (h_1 + h_3) \vdash G$$

となる新たな仮説 (h_3) を求める問題に帰着する．このようにすることで h_1 の間で無矛盾性の確認に要していた時間を削減できる．また，新たに生成すべき仮説数が減少するため，これによっても推論時間を短縮する事ができる．この手法では適当な例を前もって与え学習させておく事により，同種の問題に対し大幅な推論時間の短縮をはかる事ができる．

知識コンパイルによる高速化 [堂前 94]

通常の仮説推論ではゴールが与えられてから推論が開始される．これに対し，知識コンパイルの考え方は，ゴールが与えられる前に可能な推論をあらかじめ行ってしまおうというものである．知識コンパイルにはある程度時間がかかってしまっても，ゴールが与えられてからの推論時間はできるだけ短縮する．この手法では $\Sigma \cup h \vdash G$ となるような最小の h を求めるのだが，コンパイル法ではまず Σ を Prime Implicates(PI) に展開しておく．PI とは，Σ から導出される節のうち他に包摂されない節である．命題表現の場合には，$A \subset B$ のとき，節 A は節 B を包接しているという．PI が計算されていれば，もしゴールが PI に含まれれば，$(PI - G)$ が解仮説 h となる．したがって，何も推論する必要がなく，単に PI を格納するメモリ探索で推論を置き換える事ができる．PI の計算自体には指数オーダの時間がかかるがこれは推論を行う前にあらかじめ実行しておく事ができるため問題はない．だ

がPI自体が膨大な量になってしまう問題がある.

知識ベースのリフォメーション [高間 95]

何度も用いるような知識を扱うには，PIを保持しておけばよいが，一回しか用いないような知識も保存しておくには，メモリや計算時間の面から無駄が多い．そこで，この論文では，ゴールが与えられてからの知識ベースのリフォメーションを行う手法を提案している．大きく3つのフェーズからなり，それぞれ知識ベースの部分コンパイル，冗長性の除去，そして多段化による冗長性の除去を行う.

2.4.2 数理計画法を利用した高速推論法

コストに基づく仮説推論は一種の最適化問題である．最適化問題もしくは数理計画問題は、オペレーションズリサーチ (OR) の分野で長く研究されてきた．人工知能の諸問題においても、制約に基づく問題解決のアプローチが重要性を増してきており、知識表現としてのAIと，探索の方法としてのORのそれぞれの良い面を取り入れることが重要である [石塚 97, Selman 97].

命題論理表現の仮説推論の問題は，0-1整数計画法の問題に置き換えることができる．0-1整数計画法の計算複雑度はNP完全であるから，厳密解法を使う限り，やはり計算時間は問題サイズに対して指数オーダになる．しかし，最適解を得るという目的を準最適解を得ることに譲歩すれば，指数オーダの推論時間の壁を克服することができる．掃出し補数法 [Balas 80] は0-1整数計画問題の準最適解を多項式オーダで得る有効な方法であるが，岡本らはこの掃出し補数法を用いて仮説推論問題が高速に解けることを示した [岡本 93]．なお，本論文における準最適解とは，解仮説に含まれる要素仮説のコストの和が必ずしも最適 (最小) ではないという意味であり，ゴールを証明するのに十分な無矛盾な仮説の組という論理的制約は満たす.

0-1整数計画法への変換では，例えば以下の例のようにホーン節を置き換えることで問題を解く.

1. $p = q_1 \vee \cdots \vee q_n$

2. $p = (q_1 \wedge \cdots \wedge q_n) \vee r$

$$\Downarrow$$

1. $\dfrac{x_{q_1} + \cdots + x_{q_n}}{n} \leq p \leq x_{q_1} + \cdots + x_{q_n}$

2. $\dfrac{x_{q_1} + \cdots + x_{q_n} + nr - (n-1)}{2n} \leq p \leq \dfrac{x_{q_1} + \cdots + x_{q_n} + nr}{n}$

このような置き換え法には，さまざまなものが提案されているが，それについては，次章で詳しく議論する．

大澤らは掃き出し補数法を，ネットワーク上での振舞いという点に着目し，ネットワーク化バブル伝搬法 (NBP 法) を開発した [大澤 94, 大澤 95a, 大澤 95b, Ohsawa 97]．仮説数に対し $N^{1.4}$ 以下のオーダの計算時間で，準最適解の探索を可能としている．

しかし，これらの準最適解を求める手法は，解の質と探索時間のトレードオフがある．つまり，探索時間を短くすれば得られる解のコストは大きくなり，探索時間を長くかければコストの小さい解が得られる．したがって，どのくらいの探索時間でどの程度の質の解が得られるかが重要である．

また，対象とする問題に依存して，アルゴリズムの実行結果も異なる．したがって，どのような問題に適用しているのかという分析も不可欠である．この話題に関しては，5 章で詳しく述べる．

2.4.3 SAT における高速解法

仮説推論と関係の深い SAT 問題では、以前から GSAT を中心とした解法が大きな成果を上げている．GSAT は、山登り法とランダムな再スタートを組み合わせた手法であるが、一回の山登りでより探索可能性を高めるには、**違反した制約の重みを上げていく** GSAT extension [Selman 93] や、これと同様な breakout 法 [Morris 93]、heuristic repair method [Minton 92] などの手法が有効である．breakout 法のアルゴリズムの概要は以下である．

1. 初期点を決める.

2. 総コスト（違反している制約の重み和）を最も下げるような変数の値のフリップ (0と 1 の入れ換え) を行う．そのような変数がなければ (局所最適解に陥れば)4 へ.

3. すべての節を満たしていれば終了．そうでなければ 2 へ.

4. 違反している制約の重みを増やす．2 へ.

制約の重みの更新は、探索が局所解に陥ったときだけに行われる.

同様の手法で，制約充足だけでなく目的関数も考慮したものとして、DLM(discrete Lagrangian based search method)[Wah 97, Wu 00] や GLS(Guided Local Search)[Voudouris 95] が挙げられ、近年 MAX-SAT 問題などで良好な成果を挙げている．これらに共通するのは、図 2.3 のように，山登り法を基本としながら、探索が局所解に陥った場合に違反している制約の重みを上げることで探索空間の形を変え、局所解から脱出することである.

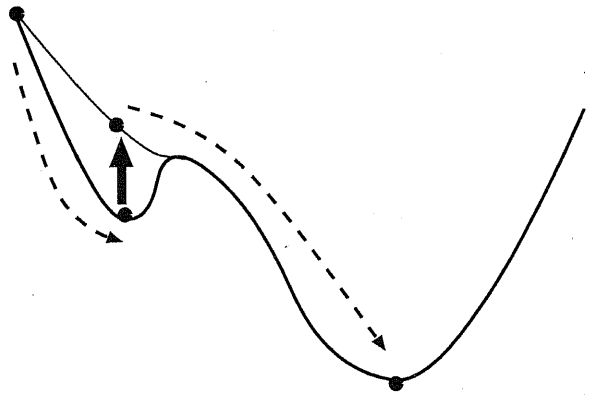


図 2.3: Breakout 法の概念図

ここでは，簡単に DLM のアルゴリズムを紹介しよう．DLM は，各変数値 x（0-1）と各節の持つ重み λ（実数値）を反復的に更新していくことにより解を得る．まず，m 変数 n 節からなる SAT 問題を次のような最適化問題に置き換える.

$$\begin{aligned} \min_{x\in\{0,1\}^m} \quad & N(x) = \sum_{i=1}^{n} U_i(x) \\ \text{subject to} \quad & U_i(x) = 0 \ \ \forall i \in \{1,2,\ldots,n\}, \end{aligned}$$

ただし，$U_i(x)$ は，x が節 i を満たすとき 1，そうでなければ 0 を取る関数である．ラグランジュ関数は，

$$L(x,\lambda) = N(x) + \lambda^T U(x)$$

で表される．次のように x および λ を更新していけば，鞍点となる (x^*, λ^*) が得られる．

$$x^{k+1} = x^k - \Delta_x L(x^k, \lambda^k) \tag{2.5}$$

$$\lambda^{k+1} = \lambda^k + U(x^k) \tag{2.6}$$

ただし，$\Delta_x L(x,\lambda)$ は discrete gradient operator と呼ばれ，x の近傍（ハミング距離 1）の中で，最も $L(x,\lambda)$ を下げる x' を指すベクトルである．もし，$L(x,\lambda)$ を下げるものがなければ，$\Delta_x L(x,\lambda) = 0$ である．

これを実現する最も簡単なアルゴリズムは以下の通りである．

1. x と λ の初期化
2. x が解でなければ，$x^{k+1} \leftarrow x^k - \delta_x L(x^k, \lambda^k)$ にしたがって x を更新．
3. もし，λ を更新する条件（局所解に陥る，もしくは規定の反復回数に達する）を満たした場合，$\lambda \leftarrow \lambda^k + U(x^k)$ にしたがって λ を更新．
4. 2 へ．

すなわち、DLM は、ラグランジュ関数（つまり，もとの目的関数＋違反している節の重み和で表される関数）の降下方向に探索点を進める．局所解に陥った場合、違反している節の重みを上げることによって、局所解から脱出する．人間がパズルを解くときでも、試行錯誤を繰り返すうちに制約に違反しやすい部分、どうしてもうまくいかない部分が分かってくる．そういった部分により注意を払う（重みを上げる）ことにより、より効果的な試行錯誤が繰り返され、パズルが効率的に解けるだろう．したがって、変数値（パズルの配置）だけでなく、ラグランジュ乗数（どの制約がどれくらい重要か）の空間も同時に探索することは、直観的にも効率的なアプローチである．実際、DLM は SAT 問題のベンチマー

クで今まで解けなかった問題のいくつかを解いており [Wu 00]、SAT 問題に対する最も効果的な解法のひとつである.

一方、Gu は，SAT 問題を制約なし非線形計画問題に置き換えて解く手法 [Gu 93, Gu 94] を提案した．UniSAT と呼ばれるこのモデルでは，例えば，

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

を

$$f = (x_1 - T)^p (x_2 + F)^p + (x_1 + F)^p (x_2 - T)^p (x_3 - T)^p$$

(ただし T, F, p は定数）と置き換える．この関数を最小値 0 とするような値を見つければ，SAT 問題の解が得られたことになる．DLM が離散的な空間を探索しているのに対し，この手法は連続空間を探索するアプローチとなっている.

さて，次章からは，仮説推論の数理計画問題への置き換え法，そして高速推論法について順次述べていくことにする.

第3章　仮説推論の最適化問題への置き換え

命題論理表現のコストに基づく仮説推論を最適化問題に置き換える方法は，今までさまざまなものが提案されてきた．本章では，それらを最も基本的な2つの置き換え法にまとめ，それらの変種として他の方法が表せることを述べる．

本章を通じて、命題論理変数 x_i の真/偽は1/0に対応する．また，目的関数 f を以下のようにおく．

$$f = \sum_{i \in H} w_i x_i \tag{3.1}$$

(ただし、w_i は要素仮説 i の重み、H は要素仮説の集合)

この f を最小化するコスト最小の解仮説を求めることになる．

以下では、ホーン節を変数間の制約に置き換えるための前処理、置換法について、順に述べる．

3.1　ホーン節知識の前処理

知識ベースKB中のホーン節を効率的に制約に変換するため、以下の操作を行う．

(1) 真（もしくは偽）であることが自明なファクトノードに真（もしくは偽）の値を代入する．不必要となったホーン節は除去する．

(2) ゴールに関連した知識だけを取り出す．

(3) 完備化を施す．

(4) トップダウンのルールを取り出す．

(1) は、SAT における Unit clause（リテラルが 1 つの節）の除去 [Davis 60] に相当する．例えば，$a.$　$b \leftarrow a.$ という節があれば，a も b も真となるので，これらの節は除去する．

(2) は、Relevant reasoning[Levy 97] と呼ばれる処理である．まず、ゴールをヘッド部とするルールを取り出し、さらにそのボディ部に出現するアトムをヘッド部とするルールを取り出す．これを順次行うことによって、ゴールの証明に必要なルールだけを取り出すことができる．取り出したルールのいずれにも出現しない仮説は、ゴールの証明に無関係な仮説であり、このような仮説を含む矛盾制約は除去することができる．推論パスネットワーク法 [伊藤 91] では (1) と (2) を合わせて、推論パスネットワーク形成フェーズと呼んでいる．

(3) では、知識ベースに対し完備化を施す．完備化は、“$q \leftarrow p$” を” ”$p \leftrightarrow q$” とする操作であり、$\{q$ if $p\}$ を $\{q$ if and only if $p\}$ とすることに対応する[1]．この完備化によって、例えば、「ゴールノードは真、他のノードはすべて偽」という自明な解が得られるのを避けることができる．

完備化の手順は、次のようになる．

(a) あるアトムが複数のホーン節のヘッド部に出現するなら、各節ごとにその名前を付けかえる．さらに、付けかえた複数のアトムの選言をボディ部に、元のアトムをヘッド部にもつ新たな OR ルールを知識ベースに加える．

(b) 各ルールに対して、逆方向のルールを知識ベースに加える．

例えば、以下の知識ベース

$$a \leftarrow b \wedge c. \tag{3.2}$$

$$a \leftarrow c \wedge d. \tag{3.3}$$

は、(a) の処理により、次のようにヘッド部のアトムの名前の付けかえが行われる．

$$a1 \leftarrow b \wedge c. \tag{3.4}$$

$$a2 \leftarrow c \wedge d. \tag{3.5}$$

$$a \leftarrow a1 \vee a2 \tag{3.6}$$

[1]なお、矛盾制約には完備化の必要はない．

さらに (b) の処理により、以下のルール（下線部）が付け加えられる．

$$a1 \leftarrow b \wedge c. \tag{3.7}$$

$$\underline{b \leftarrow a1.} \tag{3.8}$$

$$\underline{c \leftarrow a1.} \tag{3.9}$$

$$a2 \leftarrow c \wedge d. \tag{3.10}$$

$$\underline{c \leftarrow a2.} \tag{3.11}$$

$$\underline{d \leftarrow a2.} \tag{3.12}$$

$$a \leftarrow a1 \vee a2. \tag{3.13}$$

$$\underline{a1 \vee a2 \leftarrow a.} \tag{3.14}$$

したがって、得られた知識ベースはホーン節でないものも含まれることになる．以下では、下線部のルールをトップダウンのルールと呼ぶことにする．

(4) の処理は仮説推論に特有の処理である．仮説推論はゴールを証明する仮説を見つけるので、探索の過程でゴールの証明に必要なノードは真となる．しかし、ゴールの証明に不必要な要素仮説が真となると解コストが大になるため、コストの小さな解を求めようとすると、必然的に不必要な要素仮説は除去される．そのため、トップダウンのルールだけを考慮して推論を行えばよい．

以上の操作により、仮説推論問題から、以後の探索に必要な少ない数の節（非ホーン節も含む）を取り出すことができる．この節からなる知識ベースを KB' とする．

3.2 線形不等式制約への変換

さて、知識ベース中に以下の節があるとする．

$$p1 \vee \neg p2 \vee \neg p3 \tag{3.15}$$

この節は3つのリテラルのうちどれかが真になればよいという要請を表わすので、以下の不等式制約と等価である.

$$x_{p1} + (1 - x_{p2}) + (1 - x_{p3}) \geq 1$$

即ち、節から以下のように等価な不等式制約を作ることができる.

< 置換 L >

- リテラル p, $\neg p$ をそれぞれ x_p, $(1 - x_p)$ に置き換える.
- $\vee$ を+に置き換え、左辺とする.
- (左辺) ≥ 1 とする不等式制約を作る.

ホーン節を不等式制約に置き換える方法はいくつか提案されていたが[石塚 96, Santos, Jr. 94]、ホーン節の前処理+節の置換 L による制約への変換として説明することができる. この置換 L は、節を充足する 0-1 点を内部に含む線形不等式制約のうちで最小のものである. 変数の 0-1 制約を考慮しない場合、この制約を満たすことは、元の節を満たすための必要条件となる.

変数の 0-1 制約を緩和することで、次の問題 LP が得られる.

問題 LP:

$$\text{Minimize} \quad f = \sum_{i \in H} w_i x_i$$

$$\text{subject to} \quad g_j(\boldsymbol{x}) \geq 0 \quad (j \in C)$$

$$0 \leq x_i \leq 1 \quad (i \in N) \tag{3.16}$$

(ただし、$g_j(x)$ は、KB' に含まれる各節を置換 L により線形不等式制約に変換したもの. また、C は制約集合、N は変数集合.)

この線形計画問題は、シンプレックス法により高速に解 (実数最適解) を得ることができるが、そのメリットは以下のようにまとめることができる.

- 実数最適解が 0-1 値なら、仮説推論の最適解が得られたことになる.

- そうでなくとも、コストの低い0-1解（準最適解）への指針となる.
- 実数最適解のコストは、0-1最適解の解コストの下界値を示す.
- 問題LPが実行不可能であれば、実行可能な0-1解も存在しない.

仮説推論では、問題LPで得られた実数最適解の近傍の0-1解を探索することにより、コストの低い解を見つける手法が幾つか提案されてきた[大澤 94, 松尾 98]. しかし、仮説推論に比べ制約の厳しいSAT問題に対しては、すべての値が0.5となってしまうことがしばしば生じ、よい指針とならないことが分かっている[Selman 97].

3.3 等式制約への変換

式3.15の節は、

$$\neg(\neg p1 \wedge p2 \wedge p3)$$

と変形できるので、$\neg p1 \wedge p2 \wedge p3$が偽となればよい. したがって、以下のように等式制約でも表現することができる.

$$(1 - x_{p1})x_{p2}x_{p3} = 0 \tag{3.17}$$

これは一般化すると以下の置換となる.

< 置換NL >

- リテラルp, $\neg p$をそれぞれ$(1 - x_p)$, x_pに置き換える.
- $\vee$を$\times$に置き換え、左辺とする.
- （左辺）$= 0$とする等式制約を作る.

置換Lが"$\vee$"を"$+$"で置き換えていたのに対し、置換NLは各節をAND形式で表した上で"$\wedge$"を"$\times$"で置き換えている.

置換NLにより、次の問題NLPが得られる.

問題NLP:

$$\text{Minimize} \quad f = \sum_{i \in H} w_i x_i$$

$$\text{subject to} \quad h_j(\boldsymbol{x}) = 0 \quad (j \in C)$$
$$0 \leq x_i \leq 1 \quad (i \in N) \tag{3.18}$$

(ただし、$h_j(x)$ は、KB' に含まれる各節を置換 NL により等式制約に変換したもの．また、C は制約集合、N は変数集合.)

問題 NLP を、制約付き非線形最適化の手法であるペナルティ法で解くと、以下の関数の制約なし最小化問題に帰着される．

$$f' = \sum_{i \in H} w_i x_i + k \sum_{j \in C} |h_j(x)|^2 \tag{3.19}$$

この手法は、SAT 問題に対して Gu が提案した手法 [Gu 94] に相当する．また、前章で述べた DLM は、問題 NLP をラグランジュ法で解くことにより得られる手法である．

3.4 2つの変換法について

節を不等式または等式制約に置き換える方法にはどのようなものがあるのだろうか．"∧" を"×" に、"∨" を"+" に置き換えるのは解釈として自然であるので、これらについて考えてみると以下の 4 通りの方法が挙げられる．

1. ∨ で結ばれたリテラルが真となる制約: 各項の和が 1 以上.

2. ∨ で結ばれたリテラルが偽となる制約： 各項の和が 0.

3. ∧ で結ばれたリテラルが真となる制約： 各項の積が 1(以上).

4. ∧ で結ばれたリテラルが偽となる制約： 各項の積が 0.

各節の否定を取ることによって、(1) と (4)、(2) と (3) はそれぞれ等価なものとして変換することができる．このうち、(2) と (3) は、それぞれ各項が 0、1 であるという制約に分解することができ、問題を解く上ではその方が扱いやすい．(1) と (4) は相互に変換できるが、変換した制約式の形は異なる．この (1) と (4) が、前節までに述べた置換 L と置換 NL

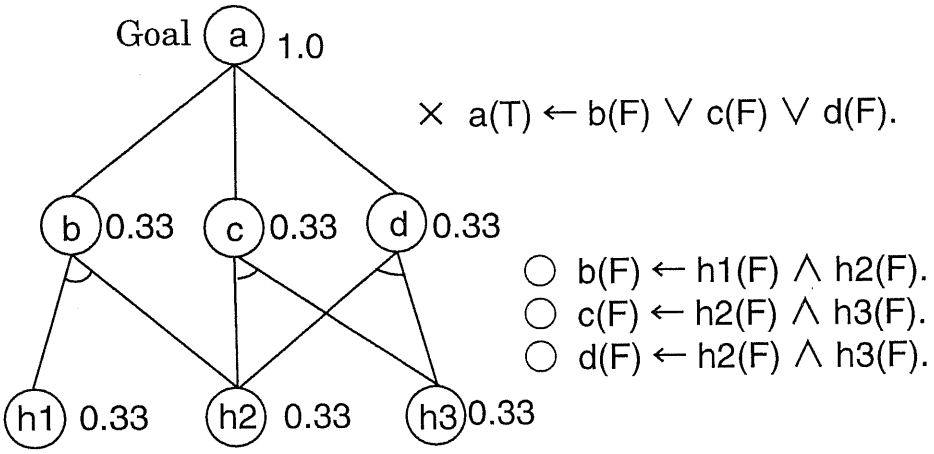


図 3.1: 置換Lの解がもとの節を満たさない例：図中の数字は変数値を表す．また、a(T),a(F)はaがそれぞれ真、偽であることを示し、○、×はそれぞれ充足、非充足を示す．

に相当する．したがって、ここで述べた2つの置換法L、NLが基本的な2つの置換法と考えることができよう．

次に、この2つの置換法の特徴について考える．まず、置換Lは、変数が節を満たすための必要条件である．したがって、問題LPを解いても、元のホーン節を満たす解が得られないかもしれない．3.1にそのような状況が起こる典型的な例を示す．しかしながら、実行可能な領域を線形に切り出しているため（3.2左)、コストの低い方向に進むことができる．

一方、置換NLは、変数が節を満たすための必要十分条件である．節を満たす変数が制約式を満たすことは明らかである．逆に、制約式を満たす変数値が得られたとき、1ならば真に、0ならば偽に、どちらでもなければdon't careなので真偽いずれかに決めると、節を充足することになる．したがって，問題NLを解いて得られた解は必ず実行可能解となるが、実行可能領域の形は凸凹しており（3.2右)、コストの勾配方向に進むのは難しい．

次章では，LとNLという2つの置換法を利用しながら，その高速解法を探っていく．

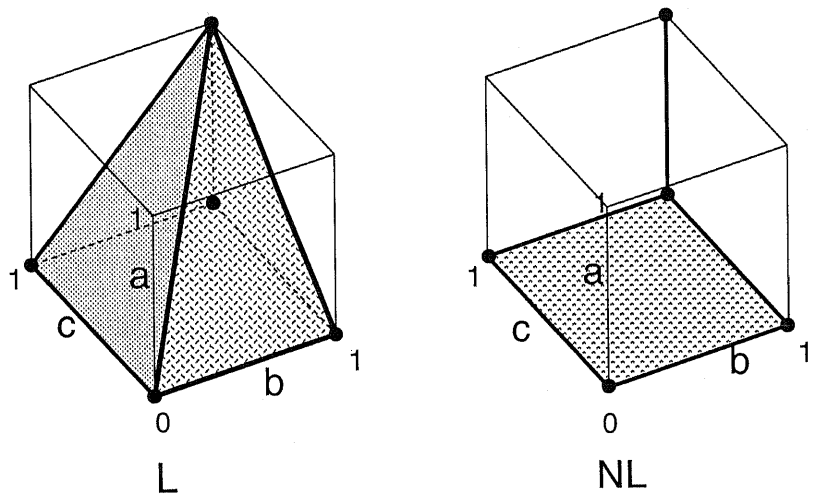


図 3.2: 実行可能領域の違い ($a \leftarrow b \wedge c.$ を変換したもの)

第4章　SL法：線形計画法と非線形計画法の併用による高速推論法

本章では，仮説推論を線形計画問題（問題L）に置き換え，初期探索点となる実数最適解を得た後，非線形関数（問題NL）の探索を行う手法について述べる．非線形関数の探索では，局所最適点へ陥ることが多く発生するため，変数の固定化という有効な脱出法を示す．固定化は，局所的に極小となる点において仮説推論問題のローカルな誤りを正し探索を再開するというもので，従来の探索点のランダムな再設定による脱出法に比べ，問題の知識構造を利用したよりシステマティックな方法となっている．

本手法は，SL法（Slide-down and Lift-up method）と呼ばれるが，非線形関数の谷に向かう降下（Slide down）と，固定化による極小点からの脱出を図る変数への値の設定による床上げ操作（Lift up）を交互に繰り返すためである．また，SL法による探索動作を視覚化して示す．

4.1　アルゴリズムの概要

SL法は，仮説推論問題を置き換えて得られた問題Lをまず単体法で解く．その後，実数最適解の近傍を探索するために，問題NLをペナルティ法を用いて解く．この方法は，GuがSAT問題を制約なし非線形計画問題に変換して解く方法 [Gu 93, Gu 94] に準拠している．この変換法は以下のようになる．

- （命題）変数の真／偽をそれぞれ1／-1に対応させる．
- 変数x，$\neg x$をそれぞれ$(x-1)^2$，$(x+1)^2$に書き換える．
- 連言（$\wedge$），選言（$\vee$）をそれぞれ算術記号$+$，$\times$に置き換える（各節は論理的に連言

で結合されていることに注意）.

例として以下の知識ベースで表される仮説推論問題を考える.

$1 \leftarrow g.$

（g はゴールで満たされることが要請される）

$g \leftarrow a \wedge b.,\ a \leftarrow h1 \wedge c.,\ b \leftarrow h2 \wedge c.$

$c \leftarrow h3 \wedge h4.,\ c \leftarrow h5 \wedge h6.$

$inc \leftarrow h1 \wedge h4.$（$inc$ は矛盾を表し空節と等価）

完備化によって生成される論理式は次のようになる.

$1 \leftarrow g.$

$g \leftarrow a \wedge b.,\ a \leftarrow g.,\ b \leftarrow g.,$

$a \leftarrow h1 \wedge c.,\ h1 \leftarrow a.,\ c \leftarrow a.,$

$b \leftarrow h2 \wedge c.,\ h2 \leftarrow b.,\ c \leftarrow b.,$

$c1 \vee c2 \leftarrow c.,\ c \leftarrow c1.,\ c \leftarrow c2.,$

$c1 \leftarrow h3 \wedge h4.,\ h3 \leftarrow c1.,\ h4 \leftarrow c1.,$

$c2 \leftarrow h5 \wedge h6.,\ h5 \leftarrow c2.,\ h6 \leftarrow c2.,$

$inc \leftarrow h1 \wedge h4.$

そして最小値 0 となる状態を見出だす非線形関数は次のようになる.

$$\begin{aligned} f \ = \ & (g-1)^2 \\ & +(g-1)^2(a+1)^2(b+1)^2 \\ & +(g+1)^2(a-1)^2+(g+1)^2(b-1)^2 \\ & +(a-1)^2(h1+1)^2(c+1)^2 \end{aligned}$$

$$
\begin{aligned}
&+(a+1)^2(h1-1)^2+(a+1)^2(c-1)^2\\
&+(b-1)^2(h2+1)^2(c+1)^2\\
&+(b+1)^2(h2-1)^2+(b+1)^2(c-1)^2\\
&+(c+1)^2(c1-1)^2(c2-1)^2\\
&+(c-1)^2(c1+1)^2+(c-1)^2(c2+1)^2\\
&+(c1-1)^2(h3+1)^2(h4+1)^2\\
&+(c1+1)^2(h3-1)^2+(c1+1)^2(h4-1)^2\\
&+(c2-1)^2(h5+1)^2(h6+1)^2\\
&+(c2+1)^2(h5-1)^2+(c2+1)^2(h6-1)^2\\
&+(h1+1)^2(h4+1)^2
\end{aligned}
\tag{4.1}
$$

ここでは各要素仮説の重みは考慮されておらず，$f=0$ となる解が見つかったとしても解仮説のコスト最小性は達成されない．しかし線形計画の単体法によって得られる実数最適解を初期探索点としているので，その近傍の 0-1 解はおそらくコストが低いもの（最適に近い準最適解）であると考えられる．

上記のようにして生成される非線形関数の特徴的な点は，１つの変数について高々２次式であることである．これは１つの命題ホーン節中に同一変数が２回出現しないことによる．

4.2 非線形関数への置き換え法の改善

このように得られた非線形関数を探索し，最小値 0 となる点を求めればよいのだが，大きな問題に対しては局所最適点に陥り，最小値の 0 に達しない状況が頻繁に発生する．そこで，まず，非線形関数への置き換え法の改善を行う．

次の例を考えよう．(以下で ( ) 内はある状態で変数のとる真理値を表す.)

$$\cdots$$

$$a(\text{false}) \leftarrow b(\text{true}) \wedge c(\text{false}) \wedge d(\text{true}) \tag{4.2}$$

$$c(\text{false}) \leftarrow e(\text{false}) \vee p(\text{true}) \tag{4.3}$$

...

この例では，式 (4.3) が非充足な状態であるため，全体の非線形関数の値が 0 にならない．式 (4.2) と式 (4.3) は完備化を施され前述の非線形関数に置き換えると，それぞれ以下の式 (4.4)，(4.5) のように表される．

$$\begin{aligned}&(a-1)^2(b+1)^2(c+1)^2(d+1)^2\\&\quad+(a+1)^2(b-1)^2+(a+1)^2(c-1)^2\\&\quad+(a+1)^2(d-1)^2\end{aligned}\tag{4.4}$$

$$\begin{aligned}&(c+1)^2(e-1)^2(p-1)^2\\&\quad+(c-1)^2(e+1)^2\\&\quad+(c-1)^2(p+1)^2\end{aligned}\tag{4.5}$$

変数 c はこの部分だけに出現するとし，c を変動させる場合について考える．c についての偏微分はそれぞれ次のようになる．

$$\begin{aligned}&(-1-1)^2(1+1)^2(2(c+1))(1+1)^2\\&\quad+(-1+1)^2(2(c-1))=128(c+1)\end{aligned}\tag{4.6}$$

$$\begin{aligned}&2(c+1)(-1-1)^2(1-1)^2\\&\quad+2(c-1)(-1+1)^2\\&\quad+2(c-1)(1+1)^2=8(c-1)\end{aligned}\tag{4.7}$$

これらは式 (4.1) のように加算されて $\partial f/\partial c$ となるので，c についての偏微分は $136c+120$ となる．一変数について関数 f は 2 次式なので，$c=-120/136$ の時に f は最小となる．したがって，c は偽から動くことができない．

したがって，Gu の方法に準拠する非線形関数への変換を用いると，子ノード[1]の多い親ノード[1]で，ある子ノードの真偽によって親ノードの真偽が決まるときに問題が生じる．すなわち，その子ノードの変数の係数が 2 の累乗になってしまうため，この影響が強くなり

[1]ホーン節命題論理表現は出現する命題変数をノードとする AND／OR ネットワークで表すことができる．この場合，ホーン節頭部変数が親ノード，本体変数が子ノードとなる．

すぎて，他のホーン節の頭部に現れる同一変数の真偽が誤った値に固定されてしまうことになる．ここで問題になるのは，pが真であるときに$(p+1)^2$が2^2になるためであり，ホーン節本体のアトム数により，係数が2の累乗のオーダで大きさが異なってくることである[2]．

そこで，真$=1$，偽$=-1$と置き換えるのではなく，真$=1$，偽$=0$と置き換え，x，$\neg x$を$(1-x)^2$，x^2と書き換える[3]．これによって，各項は最大で1となり各項間の係数のアンバランスがなくなる．

このとき，上記の例では非線形関数のcについての偏微分は$4c-2$となり，$c=0.5$でcについて最小値となる．すなわち，非線形関数の最小化プロセスで式(4.2)，(4.3)の両者を充たす方向として，cには中立的な位置に向かうような力が働く．そして，他の変数との関係で，cとpが偽，あるいはcとaが真になる位置へと向かうようになる．Guの方法に準拠の非線形関数では局所最適点に陥るような場合でも，そのような事態が生じないですむ．

整理して記すと，ここで用いる非線形関数への新しい置き換え法は次のようになる．

- 変数の真／偽をそれぞれ1／0に対応させる．
- 変数x，$\neg x$をそれぞれ$(1-x)^2$，x^2に書き換える．
- 連言（$\wedge$），選言（$\vee$）をそれぞれ+，$\times$に置き換える．

このような非線形関数を用いることで，各ホーン節本体のアトム数に関係なく，関数の各積項が0から1の間の値をとるようになる．これによって，全体的なバランスが良くなり，非線形関数最小化プロセスが片寄った動きをせず，局所最適点への捕捉も減少する．

4.3 局所最適点からの脱出法

改善した非線形関数を用いても，探索が局所最適点に陥り，関数値0の最適点に到達しないことがある．変数値が正であるノードやゴールノードなどは，問題を表すネットワー

[2]Guは，1と-1を用いることに限定はしておらず，$(x1-T)^p$，$(x1+F)^p$と一般化している．ただし，例題中では$T=F=1$，$p=2$を用いている．

[3]論文[松尾 98]では，真=0.5，偽=-0.5としているが，平行移動しただけであり違いはない．真=1，偽=0の方が他の議論と整合がよいので，本章ではこの値にしている．

クが整合するように周りのノードを正に向かわせ，変数値が負であるノードや違反している矛盾制約は周りのノードを負に向かわせるため，これらが全て釣り合った点が極小点となるからである．

例えば，次の問題は非常に単純な問題であるが，初期値を全て偽 (すなわち 0) とすると，局所最適に陥る．負である $h1$，$h2$，$h3$，$h4$ が a，b を負に向かわせているために，ゴールノード g は真になることができない．

$$g \leftarrow a \wedge b.$$

$$a \leftarrow h1 \wedge h2.$$

$$b \leftarrow h3 \wedge h4.$$

このような簡単な問題でも局所最適に陥ることから，より複雑な問題ではこの非線形関数は非常に多くの 0 でない極小点を持つことが理解できるだろう．

この解決法を考えるために，再び局所最適に陥った例を挙げる．

$$\cdots$$

$$a(\text{false}) \leftarrow b(\text{true}) \wedge c(\text{true}) \wedge d(\text{false}). \tag{4.8}$$

$$d(\text{false}) \leftarrow e(\text{true}) \wedge p(\text{true}) \wedge q(\text{true}). \tag{4.9}$$

$$\cdots$$

関数値が 0 にならず極小に陥るということは，必ずいずれかのホーン節を満たしていない訳であるが，この例では d に関して，上位ノードからの要求（d が偽であること）と下位ノードからの要求（d が真であること）が異なるため，式 (4.9) を満たしていない．このような場合，a のさらに上位ノードも a が偽であることを要求していたり，e，p，q の下位ノードもそれらが真であることを要求していたりするので，問題は単純ではない．

そこで，d を真とするか，もしくは e，p，q のいずれかを偽とすることによって，強制的に注目したホーン節を満たし，局所最適点からの脱出を図る．問題が解けない場合には，解の要素仮説が不足でゴールが充たされない場合と，矛盾制約に違反する場合の 2 つがあるが，前者を解消させる方向を優先させる．(実験的にも単体法の実数最適解を初期点とす

る探索では，真とする要素仮説が不足の状態の要素が強いので，この方策が有効となる.)
この場合，上位ノードからの要求と下位ノードからの要求が異なる場合，そのノードを真とするようにする.

さらに，あるノードを真としたいとき，その変数に初期値として1を代入し探索を再開すると，この関数は1変数に関しては2次式であるから，多くの場合その変数は元の負の値に戻ってしまい同じ局所最適点に陥る．そのために，1を代入しその後の探索では定数とみなすという操作が必要になる．これは，より強い意味での1の代入であり，以後**固定化**と呼ぶことにする．ある変数の固定化の影響は非線形関数最小化のプロセスで他変数にも及ぶことになる.

固定化の対象となるのは，次のような条件を満たすノードである.(なお，ここでの固定化は真であるノードを増やす操作であるので，矛盾制約にだけ関係する部分は固定化の対象としない.)

(1). 偽である場合のゴールノード

(2). AND関係の子ノードが全て真であるのに，偽である親ノード

(3). OR関係の子ノードが真であるのに，偽である親ノード

(4). 親ノードが真であるのに，偽であるAND関係の子ノード

(5). あるノードが真で，そのOR関係の子ノードが全て偽であれば，その内の1つの子ノード

局所最適点に陥ったら(1)から順に調べてみて，該当する変数が1個あれば，固定化し探索を再開する．同時に何個か固定化するよりも，1つの変数を固定化し探索を再開する方が，単体法による実数最適解から離れることを避けることができる.

(2)，(3)は下位ノードからの要求であり，ゴールの証明に必ずしも必要であるとは限らないが，矛盾制約も考慮して値が選ばれている訳であるので，矛盾制約に違反する恐れは少ない．しかし，(4)，(5)は，ゴールの証明には必要であるかもしれないが，下位ノード

が偽であるということは，これを真にすると矛盾の制約に違反する可能性も大きい．これらの理由から，上記の優先順位を採用した．

(5) は一義的に決まらないが，変数値の大きいノードほど，そのノードが真である場合の下位ノードの許容度が高いと考えられるので，最も変数値の大きいノードを選ぶ．この処理を**ORノード選択フェーズ**と呼ぶ．なお色々実験を行ったが，(2)，(3)，(4) の順番を入れ換えても，それほど大きな差はない．

固定化により，探索が固定化前の方向に向かうことがなく，また変数が１つ減るので必ず有限回の処理でアルゴリズムは停止する．また，極小に陥っても順次真である変数が増えていくため，矛盾制約のない問題に関しては全て解を得ることが保証される．解が得られた後に解コスト改善フェーズを行い，不必要な要素仮説を除去することで準最適解となる．

しかし，間違ったノード（変数）を固定化したため，矛盾制約に違反し，問題が解けなくなることがある．その原因として，次の２つが考えられる．

(i). 途中で偽となるべきノードが真に固定化されたため，矛盾が発生した．

(ii). ORノード選択で，真となるべきノードと異なるノードを真に固定したため，矛盾が発生した．

(i) の場合に対応するには，最後に得られた解仮説 (矛盾制約には違反しているが，問題のネットワークの他の部分は整合) から，矛盾制約に違反しなくなる様に要素仮説を減らせないか試す（**冗長判定フェーズ**）．

(ii) の場合に対応するには，ORノード選択フェーズにおいて，実際に選んだのと異なるOR子ノードを仮に選ぶ場合をストックしておき，探索が失敗したら，ORノード選択フェーズから再スタートするという処理が必要になる．なお，このようなアルゴリズムを用いるとバックトラックを認めることになるが，実験結果からこの処理が行なわれる確率は小さく，効率上の大きな問題にはならない．

もちろん，ORノードは普通は非線形関数最小化の探索によって選択されるが，その際誤ったORノードが選択されても対応する手段がない．しかし，ORノードを非線形関数

を用いる探索で決定するところにこの手法の要点があり，このような場合不都合となるのは仕方がない．局所最適点から脱出する固定化により探索の次の状態へと進むことになる．

なお，このような固定化という処理は，改善した置き換え法を用いることにより効果的になる．Gu の非線形関数への置き換え法では，探索による OR ノード選択の間違いが多く生じ，各項の大きさの違いも極小の原因となり，誤ったノードを固定化する確率が大となってしまう．

4.4 極小点の判定と解コスト改善フェーズ

SL 法のように非線形関数の大域的最適化問題においては，極小点における停止規則が重要である．極小点であることは有限回のループでは保証できないため，常にある程度誤差を含んだ極小点判定の必要があり，手間と信頼性のトレードオフが重要な課題となる．通常，このような停止規則は問題に応じて決められる．

我々の SL 法では，非線形関数 f に対し ∇f の大きさは次のようになる．

$$|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x_1}\right)^2 + \cdots + \left(\frac{\partial f}{\partial x_n}\right)^2} \tag{4.10}$$

ここで，ルートの中の項は各変数についての f の偏微分の 2 乗の和であるが，1 変数についての f の偏微分は，f においてその変数が出現する項の偏微分の和であり，各項が -2 から 2 の間の値をとることを考えると，その大きさは項の数によって決まる．言い換えれば，対応するノードに隣接する子ノードと親ノードの数によって決まり，これは問題の規模（総ノード数）には関係がなく，ネットワークの複雑さに関係する．ルートの中の項の数は，総ノード数であり，これは問題のノード数 n に比例して増える．

そこで，SL 法において ∇f を用いて停止規則を決めるには，次のように正規化することが妥当であるとした．

$$if \quad \frac{|\nabla f|}{\sqrt{n}} < |\nabla f_{min}| \quad then \text{ 極小点と判定}$$

なお，∇f_{min} は閾値で定数である．

また，探索が終了して解が得られた後にコストの改善を行う．そのアルゴリズムは，解仮説の中の要素仮説をコストの高いものから順にチェックし，一時的に偽にしてもゴールが証明できるなら解仮説を外す，というものである．0-1 整数計画法の近似解法である掃出し補数法 [Balas 80, 石塚 96] でも用いられているものである．簡単なアルゴリズムであるが，計算時間と効果のトレードオフを考えると，この方法が最も優れていると思われる．

4.5 アルゴリズム

以下に SL 法のアルゴリズムを整理して示す．なお，非線形関数の最小化の探索には最急降下法を用いる．

1. **＜初期フェーズ＞** 与えられた制約から 0-1 整数条件を外した線形計画問題を単体法によって解く．これを探索フェーズの初期点とする．

2. **＜非線形関数への置き換え＞**与えられた制約を改善した置き換え法により非線形関数に置き換える．

3. **＜探索フェーズ＞** 非線形関数の最小値 0 を見出だす探索を行なう．0／1 への丸めにより関数値が 0 になれば終了し 6 へ．極小点に陥った場合には 4 へ．

4. **＜固定化＞** 次の条件を満たすノードを 1 つ固定化し 3 へ．固定化の対象がなければ 5 へ．

   (a) 偽である場合のゴールノード

   (b) AND 関係の子ノードが全て真であるのに，偽である親ノード

   (c) OR 関係の子ノードが真であるのに，偽である親ノード

   (d) 親ノードが真であるのに，偽である AND 関係の子ノード

   (e) **＜OR ノード選択フェーズ＞** あるノードが真で，その OR 関係の子ノードが全て偽であれば，その内最も値の大きい子ノード

5. **<冗長判定フェーズ>** 4で固定化の対象がないということは，矛盾制約を除く全てのホーン節が整合していることを意味する．従って，ゴールノードを真に保ったまま矛盾制約も違反しないように，真となっている要素仮説を減らすことができるか試みる．成功したら6へ．失敗したら4eから再スタート．再スタートの候補もなければ探索失敗．

6. **<改善フェーズ>** 真となっている要素仮説を一時的に偽としてもゴールの証明が可能か試みる．成功したらこの仮説と証明試行中に偽となるノードを偽とする．

4.6 動作の図示

単体法による実数の初期探索点からスタートし，非線形関数の最小化プロセスを基本とするSL法は，動作が理解しやすいことが大きな特徴である．SL法における探索の動作を視覚的に分かりやすく表すために，探索空間の図示を行う．

図示するに当って，多変数による多次元空間の状態をどのように表示するかが問題であり，単に2変数を軸にしたような表示法は不十分である．探索，固定化という各ステップの前の点と現在の点を結ぶ方向と，現在の点と解を結ぶ方向の2つのベクトルを軸とし，関数値を高さにとり図示すると分かりやすくなる．図示は各ステップの前の点を$(0,1)$，現在の点を$(0,0)$，解の点を$(1,0)$にとり表示する．

図4.1は1回の固定化で解にたどり着くことのできる例である．初期点から出発し，図の中程のわずかに盛り上がった部分のために解にたどり着くことができず極小点に陥る（1）．次に，固定化によって探索点を山の上に引き上げることで，最小化の探索プロセスにより解にたどり着くことができることを示している（2）．

これは1回の固定化で解にたどり着いた例であるが，もう少し複雑な例を示す．図4.2は解にたどり着くまでに4回の固定化が必要であった例である．最小化の探索（Slide-down）と固定化（Lift-up）を繰り返していることが良く分かると思う．

なお表示にはMathematicaを使用した．

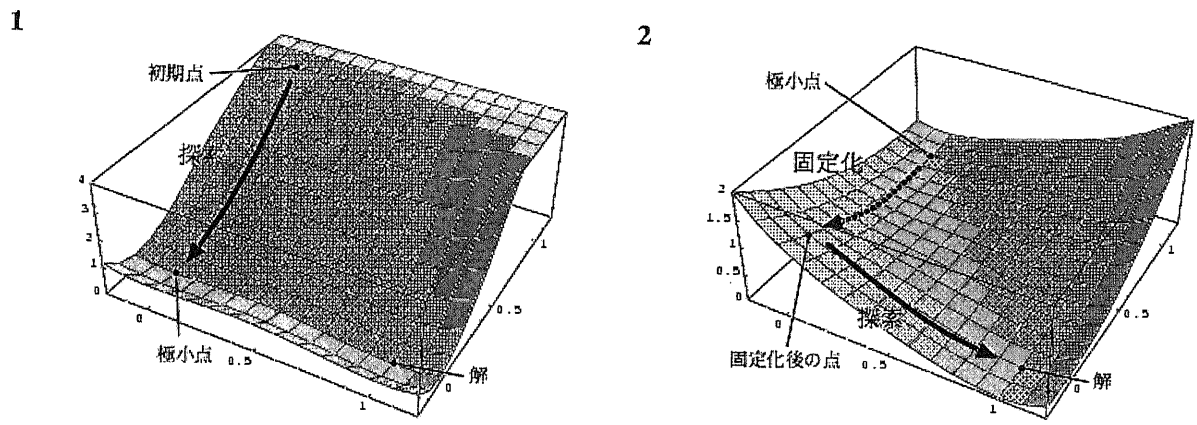


図 4.1: 探索の様子 (例 1)

4.7 評価

SL 法のシステムは C 言語で記述し，SGI Onyx 上で実行した．各ホーン節本体のアトム数は 2 ～ 7 個，どのアトムも知識ベース上での出現数の上限は 1 0 としている．極小点判定の閾値である ∇f_{min} は 5.0×10^{-3} とした．

改善前，改善後の非線形関数への置き換え法において，それぞれ単体法，非線形関数の探索（固定化なし），固定化，OR ノード選択フェーズからの再スタートのどのプロセスまでを用いて解が得られたかを表 4.1 に示す．問題 111 問中，改善前の置き換え法において，非線形関数の探索で初めて解が得られた 9 問は，改善後の置き換え法においても全て非線形関数の探索で解を得ることができた．同様に，改善前の置き換え法において，固定化を用いて初めて解が得られた 38 問は，改善後の置き換え法において，1 問（再スタートを用いて解が得られた）を除き全て固定化までで解を得ることができた．表 4.1 より改善した非線形関数への置き換え法の効果が理解できよう．また，改善後の置き換え法で再スタートまで用いる SL 法では探索失敗は 111 問中 1 問であり，得られた解コストは NBP 法（ネットワーク化バブル伝播法 [大澤 94,95,Ohsawa 97]，ほぼ全ての問題で最適コストから 3 番目以内の解が得られることが分かっている）と同程度であった．

表 4.3 に SL 法による推論速度の実験結果を示す．SL 法はノード数個の探索空間上で探索を行なうため，横軸にはノード数をとった．また，単体法は多項式オーダの計算時間で

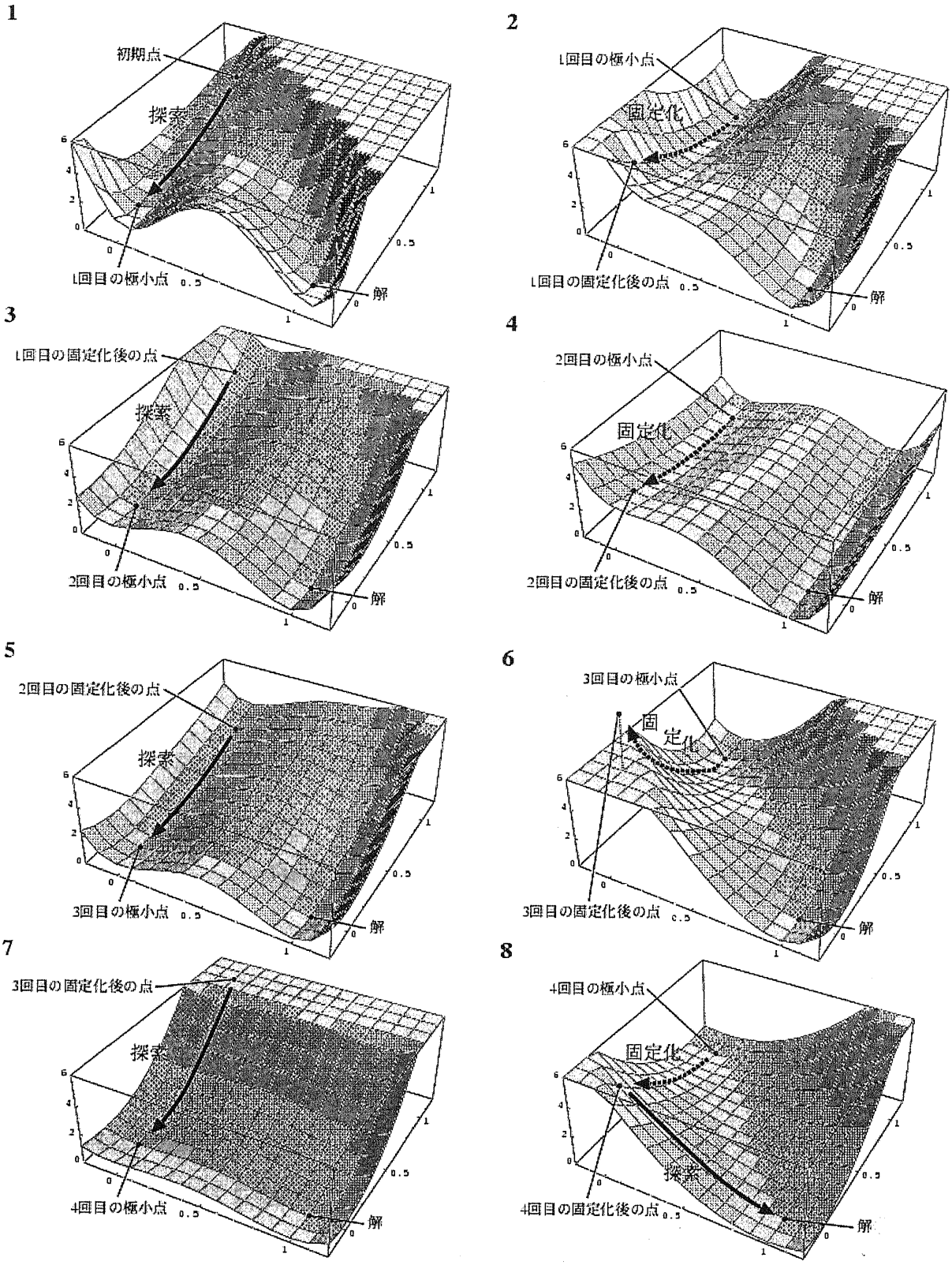

1
初期点
探索
1回目の極小点
解
2
1回目の極小点
固定化
1回目の固定化後の点
解
3
1回目の固定化後の点
探索
2回目の極小点
解
4
2回目の極小点
固定化
2回目の固定化後の点
解
5
2回目の固定化後の点
探索
3回目の極小点
解
6
3回目の極小点
固定化
3回目の固定化後の点
解
7
3回目の固定化後の点
探索
4回目の極小点
解
8
4回目の極小点
固定化
探索
4回目の固定化後の点
解

図 4.2: 探索の様子 (例 2)

表 4.1: 非線形関数への置き換え法の改善による効果（数字は解の得られた問題数）

非線形関数への置き換え法	単体法	+探索	+固定化	+再スタート	失敗
改善前	41	9	38	0	23
改善後		20	41	8	1

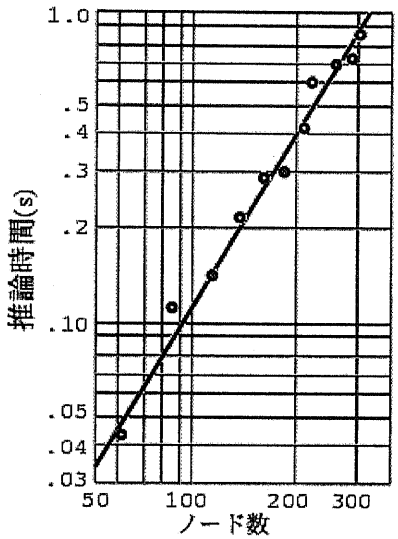


図 4.3: SL 法によるコストに基づく仮説推論の推論時間

あることが分かっているので，単体法のみで計算が終了しないケースだけについて平均をとり，縦軸には単体法の計算時間を除いた推論時間をとった．実験結果を近似的に評価すると，ノード数 n の約 1.8 乗（$n^{1.8}$）程度の計算時間を達成している．

なお，SL 法は大域的最適化の手法を用いているため，極小点の判定が計算時間に大きな影響を与え，その理論的推論速度の計算は困難であるため，実験による評価にとどめる．

本章では，非線形関数の探索により，仮説推論の準最適解を得る手法について述べた．良いと思われる方向に考えを進め，行きづまったらひとつひとつのルールのどこが間違っているかチェックし修正するというアルゴリズムは，多少なりとも人間の思考を思わせるものである．

次章では，視点を変えて，仮説推論の問題例に焦点を当ててみよう．

第5章　仮説推論の問題例に対する分析

アルゴリズムの優劣を実験的に評価する際に、用いる問題によって結果が異なったものとなる．アルゴリズムを正しく評価するには、用いる問題に対する考察が不可欠である．例えば，[Mitchell 96] には，「ランダムに生成した問題」を用いる実験結果のいい加減さが痛烈に批判されている．多くの研究は極めて簡単なクラスの問題を用いることで，パフォーマンスの良さを強調すると述べられている．

ここでは，問題例に関して2つの考察を行う．1つ目は、ある仮説推論問題に対して、解がいくつくらいありそうか計算することによって問題の難しさを表す指標を提案する．この指標により、仮説推論問題を実際に解く前に、解が得られる可能性が高いかどうか判定することができる．2つ目は，特定の問題に対して，問題のパラメータを変化させることで仮説推論の難しさがどのように変化するかという考察である．

5.1　問題の難しさの分析

5.1.1　関連研究

仮説推論の問題に対する理論的な考察は、以前にも行われていた．なかでも、コストに基づく仮説推論の最適解を多項式時間で求めるための十分条件を示した報告 [大澤 98a] は、仮説推論問題が多項式時間で解ける条件を拡大したという点で意義深い．そこでは、仮説推論問題の各命題変数をノード、ホーン節をアークとして表したグラフ上で、すべての閉路について、or ルールと inc ルールが偶数個含まれる場合には、その問題は多項式時間で最適解を得ることができることが示されている．

命題論理版の仮説推論と SAT(satisfiability) 問題は関連が深い．ランダムに生成した 3-SAT 問題では、以前から phase transition という現象が知られていた．変数の数を N、節

の数を L とすると、一般に L/N が小さい問題は解を見つけるのが簡単であり、L/N が大きい問題は充足不能である確率が高いが、これが $L/N \simeq 4.3$ で急激に変化するというものである．この場合には、L/N という値が問題の充足可能性を表すよい指標になっているといえる．

3-SAT の phase transition に関する初期の理論的研究には、次のようなものがある．

N 個の変数があるとき、その解空間は 2^N（個）である．一方、例えば、

$$a \vee \bar{b} \vee c.$$

という節を加えたとき、解空間全体のうち、$\{ a = false, b = true, c = false \}$ という可能性が取り除かれる．a,b,c だけに着目すると、$2^3 = 8$ つの候補のうち 1 つだけ実行不可能となるので、解空間全体は 7/8 になると見なすことができる．したがって、節が L 個あれば、解空間全体は $(7/8)^L$ 倍となると考える．

したがって、実行可能解の個数は、

$$2^N \left(\frac{7}{8}\right)^L$$

であると見積もることができ、この値が 1 を越えれば、実行可能解が存在する可能性が高い．

$$2^N \left(\frac{7}{8}\right)^L > 1$$

$$L/N > 5.19$$

この 5.19 という数字が、大雑把ではあるが、実際に phase transition の起こる L/N の値の理論的な予測値のひとつである [Franco 83]．その後も研究は進んでおり、例えば、Kamath は $L/N > 4.758$ なら高い確率で解を持たないことを、Frieze-Suen は phase transition の起こる L/N の下限が少なくとも 3.003 であることを示している [Kamath 94]．

5.1.2 $K1$：仮説推論の難しさの指標

SAT の分析の拡張 $K1$

ここでは、3-SAT における議論をもとに、仮説推論の難しさの指標を見つけることを試みる．基本的な考え方は、実行可能解をいくつくらい持つらしいか、という見積もりを行

うことである．3-SAT とは以下の点で異なるため、工夫が必要である．

- ひとつの節に含まれる変数の数が 3 と決まっていない．
- 真とすることのできるのは仮説だけである．中間ノードは受動的に真偽が決まる．

まず、仮説推論問題の解空間全体は、仮説の数を H とすると 2^H である．

次に inc ルールについて考える．

$$inc \leftarrow h1 \wedge h2$$

というルールを加えた場合、$\{h1 = true, h2 = true\}$ の場合にだけ実行不可能となるので、解の個数は 3/4 倍になる．一般的にボディ部の変数の数を b とすると、inc ルールによって解の個数は、$(2^b - 1)/2^b$ 倍になる．

さらに、and ルール（正のリテラルを 1 つもつ節）を考える．この場合には、各リテラルが仮説ではなく中間ノードである場合も考慮しなければならない．

$$a \leftarrow b \wedge c$$

というルールがあるとする．仮に、a, b, c が全く独立に真偽の値を取ると仮定すると、このルールがない場合には a が真となるのは、a,b,c の真偽の組み合わせ 8 通り中 4 通りであるが、このルールが加わった場合には 8 通り中 5 通りとなる．($\{a = false, b = true, c = true\}$ がこのルールを考慮することで $\{a = true, b = true, c = true\}$ となる．) 上位ノードが真となる組み合わせが p 倍になると、単純に考えてゴールが真となる組み合わせも p 倍になると考えられる．一般化して、各ルールのボディ部の数を b とすると、解の個数は $(2^b + 1)/2^b$ 倍になると考えられる．

以上をまとめると、解の個数は

$$K1 = 2^H \times \prod_{i \in and\ ルール} \frac{2^{b_i} + 1}{2^{b_i}} \times \prod_{j \in inc\ ルール} \frac{2^{b_j} - 1}{2^{b_j}}$$

であると見積もることができる．

ここで、乗除を加減に置き換えるるために $K1' = \log_2 K1$ とする． $K1'$ は大きいほど実行可能解を持つ可能性が高い、すなわち易しい問題であり、小さいほど実行可能解を持つ可能性が低い、すなわち難しい問題である．

この計算は、ルール数に対して線形なオーダの時間で行うことができる．

表 5.1: $K1'$ によりうまく分類できている例（問題群 P1）

$K1'$	Unsolved	
⇑ *hard*	数	割合
41.0	8/ 12	0.667
56.9	16/ 25	0.640
72.9	8/ 37	0.216
88.9	14/ 66	0.212
104.8	12/ 77	0.156
120.8	10/ 74	0.135
136.8	9/ 59	0.153
152.8	9/ 81	0.111
168.7	9/ 87	0.103
184.7	6/ 61	0.098
200.7	2/ 57	0.035
216.6	1/ 33	0.030
232.6	0/ 42	0.000
248.6	0/ 30	0.000
264.6	0/ 31	0.000
280.5	0/ 8	0.000
296.5	0/ 8	0.000
312.5	0/ 10	0.000
328.4〜	0/ 7	0.000
⇓ *easy*	113/1000	

評価

以上の議論はあくまでも計算上のものであり、指標 $K1'$ が問題の難しさを表しているかどうか確かめる必要がある．

ここでは、いくつかの方法でランダムに問題群を生成し、それらが解をもつかどうかを、[松尾 00] に述べられている breakout タイプのアルゴリズムで実際に問題を解くことにより

表 5.2: $K1'$ で境界がはっきりしない例 (問題群 P2)

$K1'$	Unsolved	
⇑ *hard*	数	割合
-166.6	10/ 13	0.769
-155.5	10/ 13	0.769
-144.3	13/ 17	0.765
-133.1	11/ 13	0.846
-122.0	12/ 16	0.750
-110.8	7/ 19	0.368
-99.7	8/ 14	0.571
-88.5	9/ 13	0.692
-77.3	12/ 16	0.750
-66.2	14/ 15	0.933
-55.0	9/ 16	0.562
-43.9	7/ 14	0.500
-32.7	2/ 17	0.118
-21.5	6/ 14	0.429
-10.4	6/ 14	0.429
0.8	5/ 13	0.385
12.0	1/ 16	0.062
23.1	4/ 16	0.250
34.3	1/ 13	0.077
45.4	2/ 17	0.118
56.6	0/ 1	0.000
⇓ *easy*	151/300	

調べた. このアルゴリズムは、各ノードの真理値を関連するホーン節の違反度が最小になるようにフリップさせ、違反しているホーン節は違反度の重みを重くしていくという処理を反復することにより、探索を行う. ループの回数を 5000 回とし、解が得られなければ探索失敗とした.

用いた問題群は以下の通りである.

P1 仮説数 23〜287 のツリー状[1]の問題.

P2 パラメータにより、片寄ったツリー状か、バランスの良いツリー状か調整することが

[1]もちろん、閉路もある問題であるが、仮説推論の問題は一般的にゴールを頂点としたツリーの形態のグラフとなることから、ここではツリー状と呼んでいる.

できる問題．P2は片寄ったツリー状、P2′はバランスの良いツリー状の問題である．仮説数50.

P3 完全なバランス木にincルールをランダムに追加した問題．仮説数27.

問題群P1, P2についての結果を表5.1,5.2に示す．ランダムに問題を生成し、$K1'$の値をもとに、区間ごとにその問題が解けたかどうかの統計をとったものである．

表5.1では、$K1' \geq 72.9$でUnsolvedの割合は約2割以下であり、$K1' < 72.9$で6割以上の問題がUnsolvedである．したがって、ある程度、問題が解けるかどうかを見分けるられていると言えよう．しかし、表5.2では、問題をあまりうまく分類できていない．

5.1.3 $K2$：より詳細な指標

指標$K1'$では、あまりうまく問題を分類できないケースがあり、また、指標のどのあたりで問題の解きやすい領域と解きにくい領域が移り変わるのかが一定していない．次のような点が原因として挙げられる．

- ゴールと関係あるかないかに関わらず、すべての節に対して計算を行う．
- orルールを考慮していない．(つまり、複数個のandルールのヘッド部が同じであるかどうかを斟酌していない．)

本節では、仮説推論ではルールがホーン節であることを利用し、指標$K1$をより詳細にした指標$K2$について述べる．

$K2$は、$K1$と同様、充足可能解をいくつくらいもつらしいかを表す指標であるが、仮説推論の問題構造に着目し、各ノード間の親子関係を利用して計算を行う．その際、解全体の個数に対して、各ノードを真とする解の個数の割合を計算していく．基本的な考え方は、子ノードを真とする解の割合をもとに、親ノードを真とする解の割合を計算する．この処理を反復的に繰り返すことにより、ゴールノードを真とする解の数の見積もりが得られればよい．ノードaを真とする解の割合の見積もりを$estimate(a)$で表すことにする．

まず、解全体の個数は 2^H である．以下、h ではじまるアトムは仮説、それ以外は中間ノードを表す．

and ルール

$$b \leftarrow h1 \wedge h2$$

というルールがあるとする．b を真とするのは解空間全体のうち 1/4 ($\{h1 = true, h2 = true\}$ のとき) である．よって、$estimate(b) = 1/4$ とする．さらに

$$a \leftarrow b \wedge h3$$

というルールがあるとすると、$estimate(a) = \frac{1}{4} \times \frac{1}{2} = \frac{1}{8}$ となる．つまり、a を真とするのは解空間全体のうち 1/8 である．一般化すると、and ルールについては、

$$estimate(head) = \prod_{i \in body} estimate(body_i) \tag{5.1}$$

により計算する．

or ルール

$$b \leftarrow h1 \vee h2$$

という or ルールがあるとする．b を真とするのは解空間全体のうち $1-(1/2)(1/2) = 3/4$ ($\{h1 = false, h2 = false\}$ 以外) である．$estimate(b) = 3/4$ となる．さらに、

$$a \leftarrow b \vee h1$$

というルールがあるとすると、a を真とするのは解空間全体のうち、$estimate(a) = 1 - (1 - \frac{3}{4})(\frac{1}{2}) = \frac{7}{8}$ である．一般化すると、or ルールについては、

$$estimate(head) = 1 - \prod_{i \in body} (1 - estimate(body_i)) \tag{5.2}$$

である．

inc ルール inc ルールの寄与は、指標 K1 と同様である．

まとめると、すべての仮説 $h \in H$ に対して $estimate(h) = 0.5$ とし、式 (5.1),(5.2) によりボトムアップに $estimate()$ を計算していく.

そして、

$$K2 = 2^H \times estimate(\text{ゴールノード}) \times \prod_{j \in inc\ \text{ルール}} \frac{2^{b_j} - 1}{2^{b_j}}$$

により、解の個数の見積もりが与えられる. 対数をとって、$K2' = \log_2 K2$ とする. $K2'$ も $K1'$ と同じく、小さいほど問題が難しい、つまり実行可能解が存在しにくい.

なお、$K2$ の計算は、ルール数に対して線形なオーダの時間で行うことができる.

指標の性質

仮説推論問題を表現したグラフがすべて木状で表されるときには、指標 $K2$ により解の数を正確に数えることができる.

しかし、グラフに閉路がある場合には、実際の値とは違ってくる. 例えば、

$$\begin{aligned} 1 &\leftarrow g \\ g &\leftarrow a \vee b \\ a &\leftarrow h1 \wedge h2 \\ b &\leftarrow h2 \wedge h3 \end{aligned}$$

という場合（g, a, h2, b が閉路になっている）には、$K' = 8 \times \frac{15}{16} = 7.5$ であるが、実際の解の個数は 7 である.

この閉路こそ、コストに基づく仮説推論を難しくしている要因であるが、この問題に対して分析を詳細化することによって対応しようとすると、指標の計算時間の問題が発生する. 例えば上の例で、a を真とする場合は $h1$ が真である場合と $h2$ が真である場合の 2 通りあって…、ということまで考え始めると、全解探索を行うのと同じであり、指数オーダの計算時間がかかることになる.

評価

問題群 P1, P2, P3 について、指標 $K2'$ の値と探索結果を示したものを表 5.3, 5.4, 5.5, 5.6 に示す.

この結果から、次のことがいえる.

- 急激な phase transition 現象は、あまり見られないようである. これは、仮説推論問題が、ランダムに生成した SAT に比べ構造を持っているからであると思われる.
- 指標が負となる（問題が難しい）領域では、どこで問題がほとんど解けなくなるかは問題群によりさまざまである. これは、問題のツリーがどの程度片寄っているかなどにも依存し、たまたま簡単に解ける場合があるからであろう.
- 指標が正となる（問題が易しい）領域では、だいたい 30 から 50 を越えるとほとんどの問題で解が得られるようである. つまり、解があるであろう問題は予測することができる.
- 問題の探索時間は、$K2'$ が小さくなると長くなっている. したがって、探索をいつ打ち切るかの判断にも指標 $K2'$ を利用できそうである.

5.1.4 構造のある問題に対して

構造のある問題、例えば、割り当て問題や経路探索問題を仮説推論で表現し、この指標が有効であるか実験を行ったが、現在のところ、よい結果は得られていない. このような問題では、問題に解があるにも関わらず、指標 $K2'$ が非常に小さくなる状況が頻繁に発生し、問題が解をもつかどうかをうまく予測することができない.

これは、もちろん、大きな解空間のごく一部だけが実行可能解となるように問題が作られているためである. [Gomes 97] では、構造がある問題に対して一部の変数にランダムな値を割り当てると、phase transition 現象が見られることが述べられているが、割り当て問題や経路探索問題でも、一部の変数の真偽をランダムに割り当ててしまうと、指標 $K2'$ と問題の難しさとの隠れた相関が現れるかもしれない.

表 5.3: 指標 $K2'$（問題群 P1）

$K2'$	Unsolved	
⇑ *hard*	数	割合
～-361.0	8/ 14	0.571
-322.6	6/ 8	0.750
-284.3	3/ 5	0.600
-245.9	7/ 12	0.583
-207.6	8/ 12	0.667
-169.2	8/ 14	0.571
-130.9	10/ 22	0.455
-92.5	12/ 29	0.414
-54.2	12/ 42	0.286
-15.8	17/116	0.147
22.5	14/136	0.103
60.9	6/177	0.034
99.2	1/161	0.006
137.6	1/125	0.008
176.0	0/ 82	0.000
214.3	0/ 25	0.000
252.7～	0/ 20	0.000
⇓ *easy*	113/1000	

構造のある問題に対しての検討は、今後の課題のひとつである.

5.2 単体法の有効性について

仮説推論では，問題Ｌを解くことで準最適解を高速に得るアプローチの研究が多く行われてきた．この節では,、簡単な例題によって、問題Ｌを解くことが有効な領域とそうでない領域が存在することを示す．さらに，対比的に問題 NL を解く手法はこのような顕著な有効性の違いがないことを示す.

5.2.1 置換Ｌの有効性

まず、図 5.1 のような問題を考える．これは最小被覆問題を仮説推論で表したものである．中間ノードを全て真とするようなコスト最小の仮説の真偽の割り当てを求める．中間

表 5.4: 指標 $K2'$（問題群 P2）

$K2'$	Unsolved		探索時間 [s]		
			Solved		Unsolved
⇑ *hard*	数	割合	平均	MIN/MAX	平均
-180.0	9/ 9	1.000		0.00/0.00	20.24
-168.4	10/ 16	0.625	0.98	0.40/1.79	20.31
-156.8	12/ 14	0.857	0.53	0.45/0.61	19.09
-145.2	10/ 13	0.769	0.76	0.44/1.08	18.28
-133.7	15/ 18	0.833	0.80	0.34/1.44	17.66
-122.1	6/ 15	0.400	0.65	0.38/1.20	16.28
-110.5	8/ 16	0.500	0.76	0.29/1.87	15.57
-98.9	9/ 14	0.643	0.67	0.39/1.05	14.66
-87.3	15/ 20	0.750	0.50	0.35/0.66	14.13
-75.8	13/ 15	0.867	0.40	0.38/0.42	12.96
-64.2	8/ 13	0.615	0.43	0.33/0.57	11.97
-52.6	7/ 13	0.538	0.62	0.30/0.98	11.05
-41.0	3/ 19	0.158	0.59	0.25/1.54	7.23
-29.4	8/ 16	0.500	0.47	0.18/0.68	8.99
-17.9	6/ 15	0.400	0.47	0.21/0.84	7.83
-6.3	2/ 13	0.154	0.30	0.17/0.57	6.99
5.3	5/ 20	0.250	0.44	0.15/1.89	5.12
16.9	2/ 14	0.143	0.48	0.12/2.45	4.83
28.4	1/ 16	0.062	0.35	0.14/1.32	3.84
40.0	0/ 10	0.000	0.20	0.11/0.28	
51.6	0/ 1	0.000	0.15	0.15/0.15	
⇓ *easy*	149/300				

表 5.5: 指標 $K2'$ (問題群 P2')

$K2'$	Unsolved		探索時間 [s]		
			Solved		Unsolved
⇑ *hard*	数	割合	平均	MIN/MAX	平均
-131.1	1/ 1	1.000		0.00/0.00	6.00
-122.2	1/ 1	1.000		0.00/0.00	57.70
-113.3	3/ 3	1.000		0.00/0.00	55.33
-104.4	11/ 11	1.000		0.00/0.00	58.01
-95.6	9/ 11	0.818	22.75	21.10/24.40	51.21
-86.7	15/ 17	0.882	10.48	8.96/12.00	56.75
-77.8	7/ 8	0.875	9.32	9.32/9.32	53.49
-68.9	6/ 9	0.667	26.20	5.00/39.70	51.50
-60.0	16/ 16	1.000		0.00/0.00	54.67
-51.2	9/ 11	0.818	46.55	40.50/52.60	50.60
-42.3	13/ 13	1.000		0.00/0.00	44.12
-33.4	10/ 12	0.833	14.23	9.37/19.10	50.05
-24.5	11/ 13	0.846	16.35	15.60/17.10	46.81
-15.7	8/ 11	0.727	25.83	12.50/37.70	42.77
-6.8	9/ 12	0.750	25.30	16.50/33.80	47.12
2.1	7/ 10	0.700	22.25	7.66/34.90	46.54
11.0	2/ 10	0.200	18.37	7.43/35.70	46.00
19.9	2/ 13	0.154	17.05	5.66/32.20	44.15
28.7	2/ 15	0.133	11.46	4.58/23.50	53.60
37.6	0/ 2	0.000	7.71	5.21/10.20	
46.5	0/ 1	0.000	4.16	4.16/4.16	
⇓ *easy*	162/200				

表 5.6: 指標 $K2'$（問題群 P3）

$K2'$	Unsolved		探索時間 [s]		
			Solved		Unsolved
⇑ *hard*	数	割合	平均	MIN/MAX	平均
-15.8	1/ 1	1.000		0.00/0.00	8.59
-13.9	37/ 40	0.925	0.61	0.26/1.16	8.11
-12.0	18/ 20	0.900	0.55	0.43/0.66	7.67
-10.0	16/ 20	0.800	1.70	0.50/3.80	7.52
-8.1	15/ 20	0.750	0.54	0.23/0.73	6.93
-6.2	14/ 20	0.700	1.56	0.32/5.53	6.54
-4.3	13/ 20	0.650	1.27	0.42/2.93	6.63
-2.3	6/ 20	0.300	0.99	0.30/3.91	5.70
-0.4	7/ 20	0.350	0.77	0.19/1.99	5.32
1.5	2/ 20	0.100	0.50	0.21/1.59	5.05
3.4	5/ 20	0.250	0.74	0.22/1.75	4.64
5.4	1/ 20	0.050	0.50	0.20/1.56	4.36
7.3	1/ 20	0.050	0.63	0.16/2.90	4.09
9.2	0/ 20	0.000	0.26	0.15/0.53	
11.1	0/ 20	0.000	0.24	0.14/0.46	
13.1	0/ 20	0.000	0.27	0.13/0.66	
15.0	0/ 20	0.000	0.17	0.12/0.37	
16.9	0/ 20	0.000	0.15	0.11/0.21	
18.9	0/ 20	0.000	0.12	0.09/0.19	
20.8	0/ 0				
22.7	0/ 19	0.000	0.10	0.08/0.13	
⇓ *easy*	136/400				

ノード数を増やしていくと、満たすべき条件が増えるわけであるから、厳しい問題になる（解の数が少なくなる）．しかし、全ての仮説を真とすれば必ずゴールは証明されるので少なくとも１つの解は持つことになる．

横軸に中間ノード数を、縦軸にシンプレックス法により解が得られたかどうかを取ったものが図5.2、5.3である．それぞれ、仮説数50、500の問題である．図5.2の場合、中間ノード数が80以下の場合にはほとんど全ての問題が線形計画法だけで解ける．しかしながら、80を越えたあたりから一気にその割合が低下し、120近くを越えると、ほぼ全ての問題で線形計画法だけでは解を得ることができなくなる．これらのケースでは、ほとんどの変数が0.5の値を取り、何の情報も得られていない．図5.3の場合には、変化がより急激に起こっている．この現象は、SAT問題に対するphase transitionと似ている．3-SATの場合には、節の数/変数の数が約4.3を越えると、問題が解を持つ確率が急に下がるのであるが、この場合にはシンプレックス法の有用性が急に落ちているので、その意味するところは異なる．

従来、仮説推論に対しては線形計画法を用いる手法が有効であるとされてきた．[Santos, Jr. 94]では97％以上のランダムに生成した問題が線形計画法だけで解けたと延べられているが、おそらく、制約のあまり厳しくない問題に適用したのだと思われる．問題の制約が厳しくなると線形計画法によるパフォーマンスが悪くなることは、線形計画法により多項式時間で解が得られる十分条件 [大澤 95b] が満たされにくくなることからも自明である．

5.2.2 置換NLの有効性

一方、置換NLにより得られた問題をSAT問題に対する手法であるDLMを適用し解いた．DLMを用いた理由は、制約なし非線形関数を用いる方法は局所解から脱出するヒューリスティックによってパフォーマンスに大きな影響があるのに対し、DLMは比較的シンプルな枠組みであり，調整すべきパラメータが少ないためである．

さて、上述の最小被覆問題をDLMにより解いたところ、全ての問題を解くことができた．（この問題は、SAT問題としては極めて簡単な問題である．）図5.4は、DLMによる探

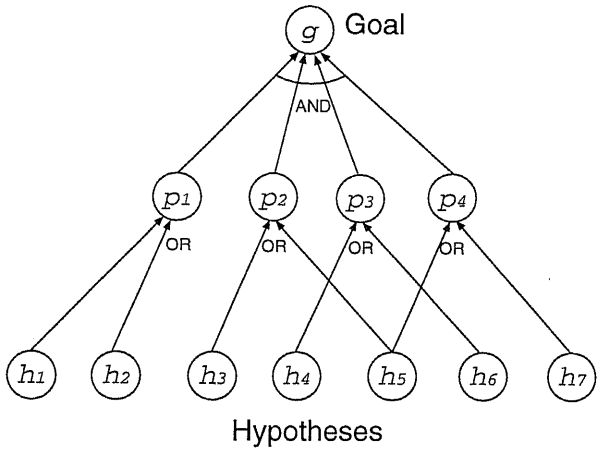


図 5.1: 最小被覆問題

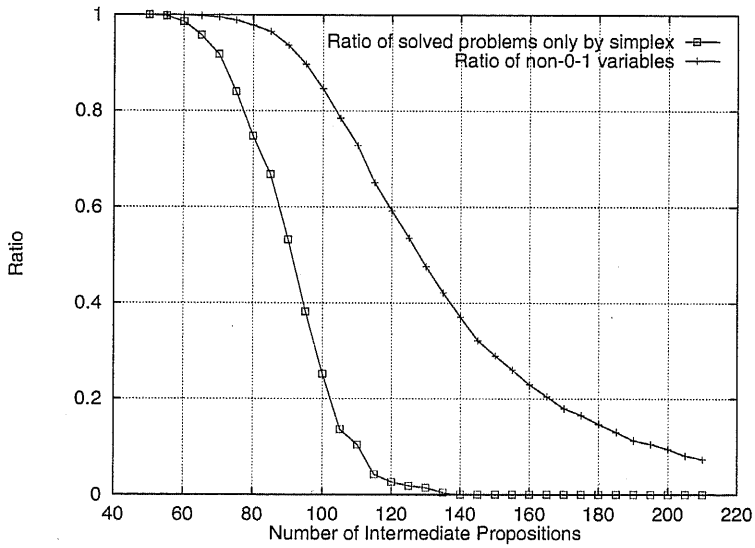


図 5.2: シンプレックス法だけで解が得られた問題の割合（仮説数 100）

索時間を取ったものであり、図中の各プロットは 100 問題の平均である．これによると、置換 L で phase transition のような現象が見られたあたりでも、他の部分と変わらず、中間ノード数が増えると実行時間が増えていることが分かる．したがって、置換 NL を利用する手法 (少なくとも DLM) は、問題の難しさに対して置換 L を利用するシンプレックス法ほど敏感ではないことが分かる．

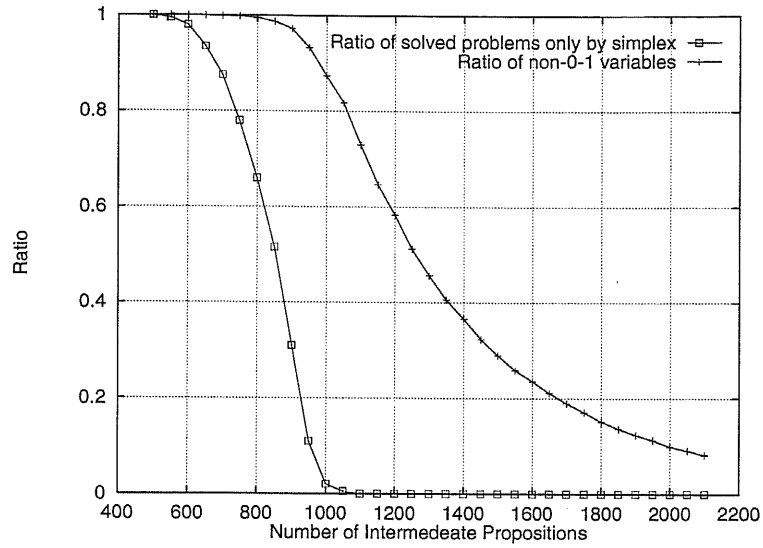


図 5.3: シンプレックス法だけで解が得られた問題の割合（仮説数 1000）

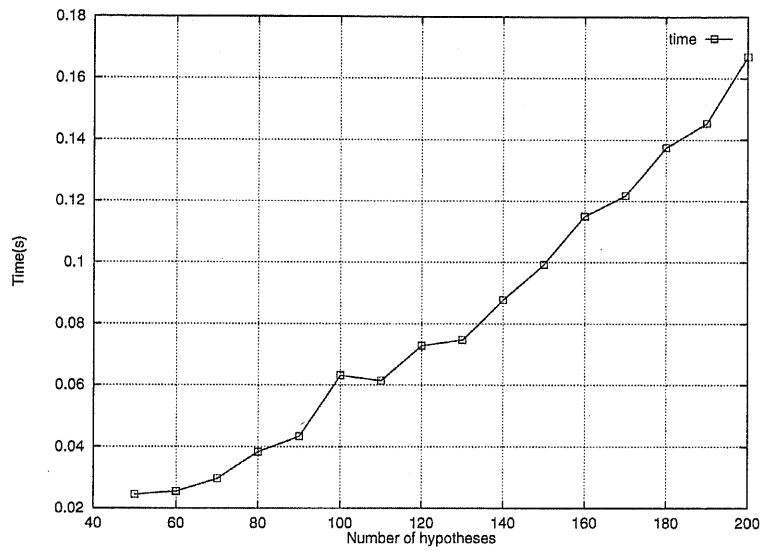


図 5.4: DLM の探索時間（仮説数 100）

5.3 議論とまとめ

本章では，アルゴリズムではなく，仮説推論の問題側からみた分析を行った．前半では，仮説推論の難しさ（解があるかどうか）をどのように予測すればよいかという問題を扱った．後半では，同じ仮説数の問題でも，単体法が極めて有効な領域からほとんど役に立たない領域まであることを示した．

アルゴリズムの評価には，どのような問題に適用するのかを無視することはできない．こ

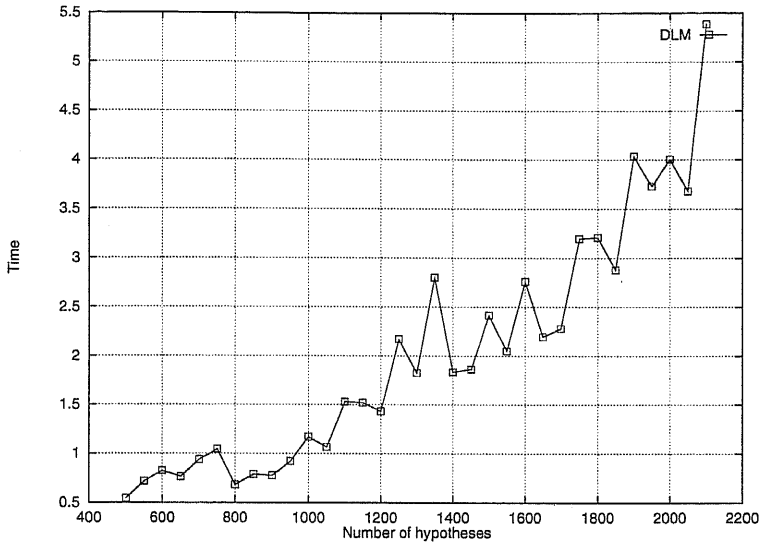


図 5.5: DLM の探索時間（仮説数 1000）

こでの議論がアルゴリズム評価の参考になれば幸いである．

さて，次章では，制約の緩い問題には極めて有効な置換 L と，難しい問題にも安定して有効な置換 NL をうまく統合するアプローチについて述べる．

第6章　2種の置き換え法の協調による高速推論法

前章では，置換Lと置換NLの問題例に対するパフォーマンスの違いについて触れた．本章では，この2つの置換法をうまく併用することで，より柔軟で高速な推論法を構築する．

6.1　2種類のプロセッサの協調による推論法

置換Lと置換NLはそれぞれ、コストの低い方向に探索を進める、及び実行可能解を見つけるという特徴がある．したがって、それぞれの特徴を生かすように両方を同時に扱いたい．

ここでは、並行計算の手法を用いることによりこれを実現する．並行計算による拡張ラグランジュ法[Bertsekas 89]では、各変数、各制約をそれぞれプロセッサと考える．つまりn個の変数とm個の制約があれば、$n+m$個のプロセッサのインタラクションにより探索を行う．したがって、この枠組では制約を付け加えたり取り除いたりすることが、制約に対応するプロセッサを付け加えたり取り除いたりすることに相当する．もちろん、線形計画問題ならシンプレックス法、制約つきの非線形計画問題でも逐次2次計画法などのニュートン法に基づく方法を用いればよいが、直接的にプロセッサ間のインタラクションにより探索が進む様子が捉えにくいので、ここでは、拡張ラグランジュ法を用いることとした．

以下では、まず拡張ラグランジュ法を用いた並行計算の手法について概説した後、具体的にどのようなアルゴリズムを構築することができるかを述べる．

なお、ここでの並行計算の目的は複数のプロセッサを用いて推論速度を上げることではなく、2種類の制約の協調をより分かりやすい形で捉えることにある．

6.1.1 拡張ラグランジュ法

以下の制約つき最適化問題を考える.

$$\text{Minimize} \quad f(\boldsymbol{x})$$

$$\text{subject to} \quad h_j(\boldsymbol{x}) = 0 \quad (j \in C)$$

$$\boldsymbol{x} \in P, \tag{6.1}$$

(ただし, C は制約集合, P は x の定義域.)

ラグランジュ法に基づくと, 次の最適性の必要条件を満たす点を見つけることが目的となる.

$$\begin{cases} \nabla_x L(\boldsymbol{x}, \boldsymbol{\lambda}) = \nabla f(\boldsymbol{x}) + \nabla h(\boldsymbol{x})\boldsymbol{\lambda} = \boldsymbol{0} \\ \nabla_\lambda L(\boldsymbol{x}, \boldsymbol{\lambda}) = h(\boldsymbol{x}) = \boldsymbol{0} \end{cases} \tag{6.2}$$

ここで L はラグランジュ関数で,

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{j \in C} \lambda_j h_j(\boldsymbol{x})$$

で定義され, λ_j はラグランジュ乗数と呼ばれる. ただし, 拡張ラグランジュ法では, 計算上のメリットから通常のラグランジュ関数にペナルティ項を加えた

$$L_c(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}' h(\boldsymbol{x}) + \frac{c(t)}{2} h(\boldsymbol{x})^2$$

($c(t)$ は反復 t に対して非減少な正の関数) [1]

が用いられる.

式 (6.2) を勾配法によって解くには, 以下のように $\boldsymbol{x}$ と $\boldsymbol{\lambda}$ を交互に更新していく.

$$\boldsymbol{x}^{(k)} := \arg\min_{\boldsymbol{x}} L_c(\boldsymbol{x}, \boldsymbol{\lambda}^{(k)}) \tag{6.3}$$

$$\boldsymbol{\lambda}^{(k+1)} := \boldsymbol{\lambda}^{(k)} + c(t) h(\boldsymbol{x}^{(k)}) \tag{6.4}$$

式 (6.3) は, L_c を最小にするような $\boldsymbol{x}$ の値を $\boldsymbol{x}^{(k)}$ に代入することを表す. 即ち, 違反している制約のラグランジュ乗数を大きくしながら, 現在のラグランジュ関数を x について最小化するというプロセスを繰り返す.

[1] ここでは, $c(t)$ は反復 30 回ごとに 2 倍としている. 一般的には数十回の反復ごとに 2〜10 倍するのがよいとされている [Bertsekas 89].

さて，仮説推論問題は制約と変数が局所的に結び付いた構造をしている．したがって，1つの変数もしくはラグランジュ乗数は，近傍のラグランジュ乗数もしくは変数値が分かれば，更新することができる．

ここで，1つの変数の値の更新や受渡しを行うプロセッサを**変数プロセッサ**，ラグランジュ乗数の値の更新・受渡しを行うプロセッサを**制約プロセッサ**と呼ぶことにする．各変数プロセッサはまわりの制約プロセッサ j から $\partial L/\partial \lambda_j^{(k)}$ を受取り，それをもとに，以下により変数値を更新する．

$$x_i^{(k+1)} := \arg \min_{0 \le x_i \le 1} L_c(\boldsymbol{x}^{(k)}, \boldsymbol{\lambda}^{(k)}) \tag{6.5}$$

一方，各制約プロセッサはまわりの変数プロセッサ i から変数値 $x_i^{(k)}$ を受取り，それをもとに，以下によりラグランジュ乗数を更新する．

$$\lambda_j^{(k+1)} := \lambda_j^{(k)} + c(t) h_j(\boldsymbol{x}^{(k)}) \tag{6.6}$$

すべての制約プロセッサと変数プロセッサは，値の更新と伝達を繰り返し，いずれの値も変わらなくなれば(収束すれば)，問題 (6.1) の局所最適解が得られたことになる．なお，問題 (6.1) は等式制約だけを考えているが，不等式制約にも拡張することができる．

以上が拡張ラグランジュ法による並行計算の概要であるが，この手法を用いるメリットとして，置換 L による制約 $g(\boldsymbol{x}) \ge 0$ と置換 NL による制約 $h(\boldsymbol{x}) = 0$ を同時に扱うことが可能である点[2]，各プロセッサはローカルな値の受渡しだけを行うので，探索の途中でプロセッサの構成を変更 (新たなプロセッサの追加，削除) が行えるという点が挙げられる．

6.1.2 アルゴリズム

前節の拡張ラグランジュ法により，複数の制約を動的に扱うための準備ができた．以下では，置換 L による制約を扱うプロセッサを**制約プロセッサ L**，置き換え NL による制約を扱うプロセッサを**制約プロセッサ NL** とよぶことにする．

まず，制約プロセッサ L と変数プロセッサにより探索を行った場合には，シンプレックス法と同じく線形計画問題を解いていることになる．得られる解は実数最適解となる．一

[2] $g(\boldsymbol{x}) \ge 0$ を扱うには、$\max(0, -g(\boldsymbol{x})) = 0$ と考える．

方，制約プロセッサNLと変数プロセッサで探索を行った場合，得られる解は0-1値の実行可能解となる．

したがって，まず制約プロセッサLにより探索を行い，0-1解が得られない場合，制約プロセッサNLに切替える，もしくは制約プロセッサNLを追加する方法がよいと思われる．

ここでは，以下の2種の戦略を試みた．

プロセッサの協調による推論法 [全節戦略]

はじめに制約プロセッサL，その後制約プロセッサLと制約プロセッサNLを併用する方法である．

1. **初期フェーズ** 制約プロセッサLと変数プロセッサで探索を行う．0-1解が得られれば終了．
2. **構成変更** すべての制約に制約プロセッサNLを加え，探索を再開する．
3. **終了条件** 実行可能解が得られれば(4)へ．規定の反復数を経ても解が得られなければ探索失敗．
4. **解改善フェーズ** 不必要な仮説が含まれていれば除去し，終了．

プロセッサの協調による推論法 [全違反節戦略]

収束ごとに，違反している制約に順次制約プロセッサNLを加えていく方法である．

1. **初期フェーズ** 制約プロセッサLと変数プロセッサで探索を行う．0-1解が得られれば終了．
2. **構成変更** 違反している制約すべてに制約プロセッサNLを取りつけ，探索を再開する．
3. **終了条件** 探索の途中で実行可能解が得られれば(4)へ．収束しても実行可能解が得られなければ(2)へ．規定の反復数を経ても解が得られなければ探索失敗．
4. **解改善フェーズ** 不必要な仮説が含まれていれば除去し，終了．

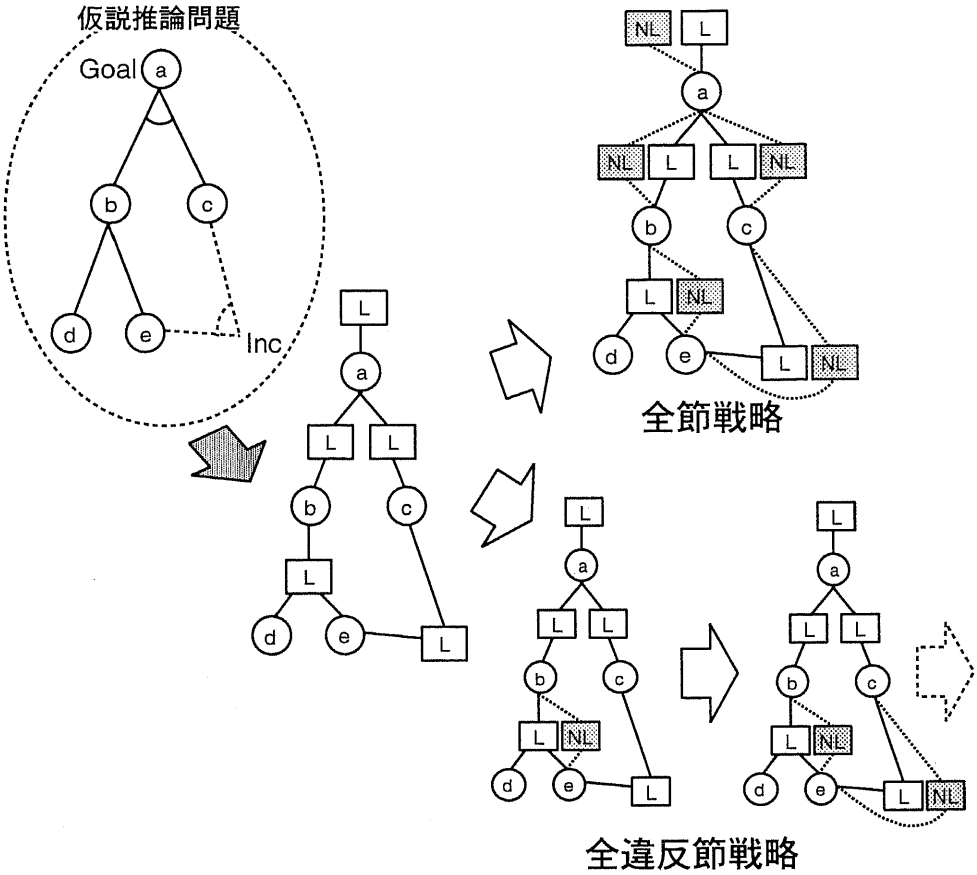


図 6.1: 制約プロセッサ NL を加えていく 2 種の戦略

これらの戦略を図 6.1 に示す．全節戦略は構成変更が 1 回なのに対し，全違反節戦略では反復的に構成変更を行う．制約プロセッサ NL を必要以上に追加しないので，よりコストの低い解を探索する可能性が高い．

この他の戦略として，全 OR 節戦略 (すべての OR ルールと仮説間の矛盾を表す Inc ルールに制約プロセッサ NL を追加する)，最大違反節戦略 (最も違反度の高い制約にだけ制約プロセッサ NL を追加する)，全節取り換え戦略 (すべての制約に対して，制約プロセッサ L を制約プロセッサ NL で取り換える) なども試みたが，実験的にここに挙げた 2 つが探索時間と解の質の面から代表的であったので，他は省略する．

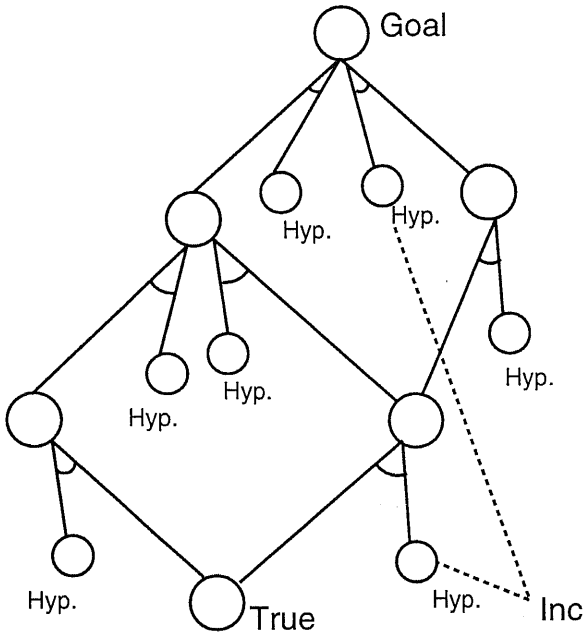


図 6.2: 最短経路問題の構造

6.2 評価

本手法に対し評価実験を行った．システムはC言語で記述し，SUN Enterprise 400MHz上で実行した．

ここでは2種類の問題を用いた．1つは，NP完全問題である集合被覆問題を仮説推論問題により表したものである (図5.1)．いくつかの代替要素から仕様を満足しコストを最小化する設計問題を単純化した問題であり，与えられた概念を説明する適切な文書集合を見つける [松村 99] などの応用例がある．問題構造としては，横長のグラフ構造となる．もう1つは最短経路問題である．ある2点間の経路を通るかどうかを仮説で表し，ある地点に到達した状態を中間ノードの真偽で表す (図6.2)．最短経路問題はクラスPに属するが，仮説間の矛盾まで考えると組合せ的な問題となる．あるノードの真偽がその祖先ノードの真偽に影響する縦長のタイプの問題となっており，回路の故障診断問題などがこのような縦長のグラフ構造をしている．

これらの問題のスケールを変えて，プロセッサの協調による推論法の全節戦略と全違反節戦略に対して推論時間を測定したものが，図6.3，図6.4である．また，得られた解コス

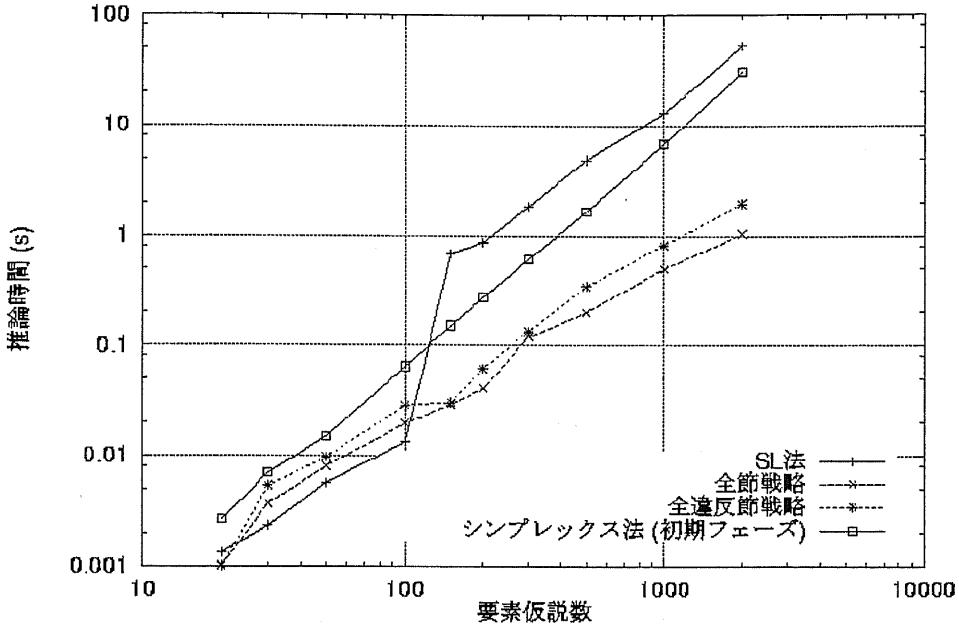


図 6.3: 集合被覆問題の推論時間

トを表 6.1，表 6.2 にそれぞれ示す．比較のために SL 法の推論結果も記した．なお，図中の推論時間は，初期フェーズ以降の推論時間であり，初期フェーズのシンプレックス法による計算時間は図中に別に記した．

まず，集合被覆問題に対しては，全節戦略，全違反節戦略ともに，推論時間，解コストともに SL 法よりもよい．しかし，実際には初期フェーズのシンプレックス法の計算時間が支配的となるため，推論時間に関しては大きな差はないことになる．なお，SL 法が要素仮説数 100 を越えたところから急に時間がかかっているのは，局所解に陥ることが頻繁に発生するようになるためである．

一方の最短経路問題に対しては，逆に SL 法の推論時間が最も速い．全節戦略はそれよりもやや遅く，全違反節戦略は非常に遅くなる．しかし，解の質では，全違反節戦略が非常に良く，次いで全節戦略，SL 法の順となっている．即ち，簡単な解はすぐに見つかるが，コストの低い解を見つけるのが難しい問題であるといえる．

以上をまとめると，推論時間に関しては初期フェーズのシンプレックス法の計算時間の

表 6.1: 集合被覆問題の解コスト

	SL 法	全節戦略	全違反節戦略	最適解
平均コスト	107.3%	101.7%	101.6%	100.0%

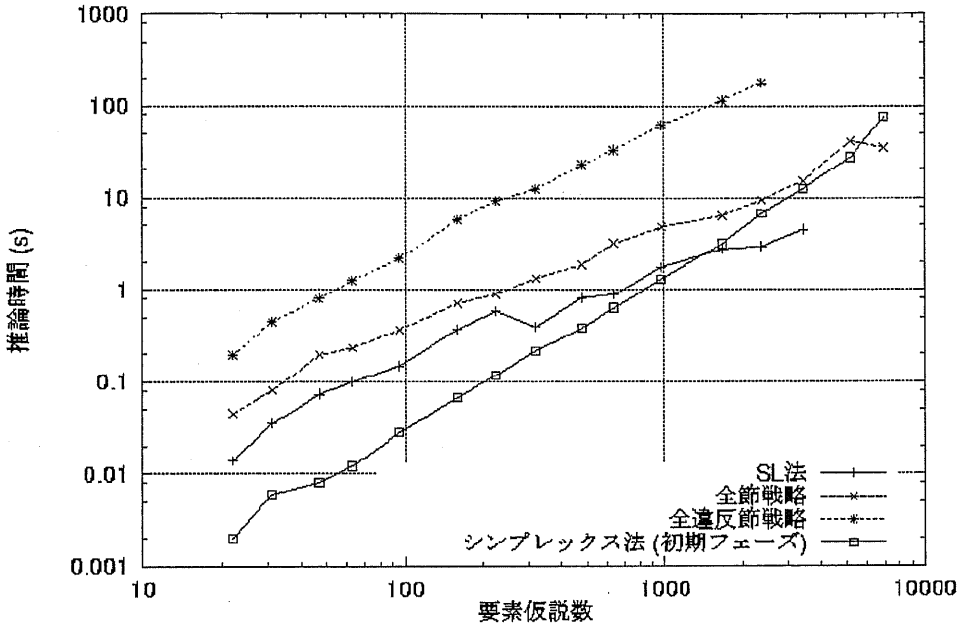


図 6.4: 最短経路問題の推論時間

表 6.2: 最短経路問題の解コスト

	SL 法	全節戦略	全違反節戦略	最適解
平均コスト	136.3%	133.6%	116.6%	100.0%

オーダが大きく，手法ごとの差はあまりない．解コストはSL法よりもよい解が得られており，単純に実数最適解の近傍を探索するSL法と比べ，制約プロセッサLと制約プロセッサNLが協調することで，よりコストの低い方向に探索がガイドされていることが示されている．

また当然，推論時間と解コストのトレードオフがあるが，パラメータだけが変更可能なアルゴリズムに比べて，プロセッサの構成を変更できる本手法は，問題の性質やユーザの

希望に応じて，より柔軟な対応が可能である．実際，11 章では文書の要約問題に対して，仮説推論問題から得られた制約だけでなく，問題に依存する制約を付与し，解を見つけるという用い方をする．

6.3 議論とまとめ

本章では，2 種類のプロセッサを用いることにより，柔軟かつ高速に仮説推論の解を得るアルゴリズムについて述べた．ローカルに探索を行う操作の集合として，全体で解を探索するというしくみは，大規模問題に対して効率的に解を得るためには不可欠な要素ではないかと考えている．

ここまでは，主に仮説推論を数理的な問題と置き換えながら，その高速解法を探るという方向で話を進めてきた．次章では，より大きな視点から，このような仮説推論システムをどのように利用すればよいかについて議論する．

第7章　システムとしての仮説推論

前章までは，主に仮説推論の高速化を中心に話を進めてきた．本章では，仮説推論を情報処理システムの一要素として位置付けたときの可能性と問題点について考察する．

7.1　仮説推論システムの可能性と限界

これまで見てきたように，仮説推論は、与えられたゴールを説明するのに必要な無矛盾な仮説の集合を求める．しかし、実際に仮説推論を用いて設計・診断問題などを解く際には、仮説の重み和を最小化するという簡単な選好基準では不十分なことも多い．例えば、ある地点から別の地点までの経路を探索するような問題では，時間と料金という少なくとも2種類のコストを考える必要があるだろう．従来のコストに基づく仮説推論では，単一のコストの和の最小化を行うが，ユーザの目的に合わせた解の選好基準を用いることができれば便利であろう．

また、知識ベースを記述する際には，ユーザがゴールと関連していると思われる知識を選んで記述する．人間は多くの知識を持っており，その中から問題の解決に必要と思われる属性を抜きだして思考している．したがって，新しい仮説を発見するということは，今まで気づいていなかった新しい属性に着目することとも言えるだろう．そのための研究として，欠落知識を補間する方法 [阿部 98, グエン 01] や新たな仮説を見つけ出す方法 [砂山 99] なども提案されている．また，ユーザが新たな知識を簡単に書き加えられるように、現在の知識ベースから得られる推論結果を分かりやすく提示することも重要であろう [堀 99]．情報をどのように提示するかという問題は，情報可視化の分野でも盛んに研究が行われている [Shneiderman 98]．

ここまで扱ってきたような仮説推論の枠組みでは，与えられた要素仮説の組み合わせで

ゴールを証明することはできるが，与えられていない仮説を使ってゴールを証明することはできない．また，与えられていない仮説を生成することもできない．既存の命題の論理式としての新たな仮説を生成する [Reiter 87] のが精いっぱいであろう．仮説推論を，情報システムの一要素としてみると，このような限界をうまく補ってやる必要があることに気づく．記述されていない仮説は生成することができないのであるから，ユーザにいかに新たな仮説を記述させるか，また，仮説推論の解から，いかにユーザに新たな仮説に気づかせるかという視点も重要であろう．

D. Poole は，1990 年の論文でこのように述べている [Poole 90]．

> 非単調推論システムを構築する人は，結局，実用問題をどのように解けるかについて言及せざるを得ない．また，これらが AI にとって有用なことを示すには，適切なプログラム方法を考えなければならない．それらは大概，どんな計算可能な関数をも表すことのできるとても強力な論理であり，「このシステムはよい問題表現システムである」と経験的に述べるには，どのように表現システムで用いられるかを示さなければならない．このようなシステムをどのように用いるかを明確に意識していなければ，前に進めるとは思わない．我々は，「正しい」知識をただ投げ入れ，適切な答えが返ってくることを期待することはできない．

仮説推論システムもまた然りであろう．つまり，システムとしてどのようなことができるかによって評価されなければならない．以下では、仮説推論システムをより柔軟で使いやすいものにするいくつかの試行について述べる．

7.2 仮説推論システムの拡張

使いやすい仮説推論システムを構築するためには、(i) 知識、ゴールが表現しやすい、(ii) 結果が理解しやすい、ことが必要である．ここでは、(i) の例として複数のコストを用いて問題を柔軟に表現した例、(ii) の例として解以外の有用な情報を表示する例について述べる．

7.2.1 複数のコストを用いる例

6章では、仮説推論の2種の置き換え法の協調による高速推論法について述べた．これは、1つの変数やホーン節をそれぞれ1つのプロセッサと考え、近傍の制約を充足するようなインタラクションを行うことにより解を得るアルゴリズムである．一般的な最適化問題の解法である拡張ラグランジュ法に基づいてアルゴリズムを構築しているので、目的関数を自由に設定することができる．例えば、

- 各仮説に複数のコストを設定する．複数のコストの関数を目的関数とし、これを最小とする仮説が最も好ましいと考える．
- 非線形な目的関数を用いる．また、離散的（組み合わせ的）な目的関数を用いる．

もちろん、これは仮説候補のうちどれを選ぶかという選好基準であって、仮説候補は背景知識とともにゴールを演繹的に導くことができなければならない．

図7.1は、東京大学から新宿へ行く経路の知識を表した知識ベースである．各経路には料金、かかる時間という2つのパラメータがあり、一般的なコストに基づく仮説推論では表現できない．ここでは、各仮説に料金、時間の2種類のコストを持たせ、さらに目的関数を、例えば、

$$Min.\ f = \sum_{h_i \in H} time_i h_i \times \sum_{h_i \in H} price_i h_i \tag{7.1}$$

ただし H は仮説集合、仮説 h_i が真のとき $h_i = 1$、偽のとき $h = i = 0$、仮説 h_i のコストを $time_i$ および $price_i$ とする．

とする．このような変更は、推論法として並行プロセスによる高速仮説推論法を用いると簡単であり、実際に、次のような解が得られる．

$$\{walk(uTokyo, Hon3), marunouchi(Hon3, Shinjuku)\}$$

$$\text{コスト1} = 190(\text{円}),\ \ \text{コスト2} = 35(\text{分})$$

ここで用いた知識ベースは、東京大学から新宿へ行く場合だけに用いることができるが、例えば、区間ごとの乗車距離を仮説に付加し、乗車距離に応じて料金を計算するような関数を用いて目的関数を設定すれば、より簡潔に知識ベースを記述することができる．

11章でも，このように仮説推論の枠組みを拡張する応用例が示されている．

```
1 ← at(Shinjuku).
at(Hon3) ← at(uTokyo), walk(uTokyo,Hon3).
at(Nezu) ← at(uTokyo), walk(uTokyo,Nezu).
at(Todaimae) ← at(uTokyo), walk(uTokyo,Todaimae).
at(Shinjuku) ← at(Hon3), marunouchi(Hon3,Shinjuku).
at(Ikebukuro) ← at(Hon3), marunouchi(Hon3,Ikebukuro).
at(Shinjuku) ← at(Ikebukuro), JR(Ikebukuro,Shinjuku).
at(Ocha) ← at(Hon3), marunouchi(Hon3, Ocha).
at(Shinjuku) ← at(Ocha), JR(Ocha, Shinjuku).
at(ShinOcha) ← at(Nezu), chiyoda(Nezu, ShinOcha).
at(Ocha) ← at(ShinOcha), walk(Ocha, ShinOcha).
at(Yotsuya) ← at(Todaimae), nanboku(Todaimae,Yotsuya).
at(Shinjuku) ← at(Yotsuya), marunouchi(Yotsuya,Shinjuku).
at(uTokyo).
$walk(uTokyo,Hon3)/0/10, walk(uTokyo,Nezu)/0/5,
$walk(uTokyo,Todaimae)/0/4, walk(Ocha, ShinOcha)/0/4,
$marunouchi(Hon3,Shinjuku)/190/25,
$marunouchi(Hon3,Ikebukuro)/160/8,
$JR(Ikebukuro,Shinjuku)/150/8,
$marunouchi(Hon3,Ocha)/160/2,
$JR(Ocha,Shinjuku)/160/10,
$chiyoda(Nezu,ShinOcha)/160/4,
$nanboku(Todaimae,Yotsuya)/160/9,
$marunouchi(Yotsuya,Shinjuku)/160/7,
コストの定義は '$命題/コスト1/コスト2.'
```

図 7.1: コストが2種類定義される問題

7.2.2 解の表示の例

仮説推論システムにおいて、推論の結果を分かりやすく表示することも重要である．解以外の情報も同時に提示し、解が得られる過程を示すことで、推論結果を利用しやすく、また新たに知識を付け加えやすくすることができる．

図7.2は、7人の学生を4つの研究室に割り当てる問題を仮説推論で表現したものである．満たすべき条件として、各研究室に1人以上2人以下の学生が配属されなければいけない．

仮説 $SiLj$ は、学生 i が研究室 j に配属されることを示している.

これを推論システムを使って解くと、

$$\{S1L3, S2L1, S3L2, S4L4, S5L1, S6L4, S7L3\}$$

コスト 20

という解が得られる. しかし、これだけでは、なぜこのような解が得られるのか分からない.

そこで、解だけでなく、並行プロセスによる仮説推論法において、最終的にラグランジュ乗数の高いルールを提示する. これは、解の導出の際に大きく考慮された（手を煩わせた）ルールである.

$$9.0: \quad inc \leftarrow S1L3[T], S2L3[F], S7L3[T].$$

$$8.0: \quad inc \leftarrow S2L4[F], S4L4[T], S6L4[T].$$

これらのルールを見ると、各研究室に 2 人までしか学生を配属できないために、学生 2 が研究室 3 にも 4 にもいけない、という状況が分かる.

また、

$$10.1: \quad G1 \leftarrow 1$$

$$4.5: \quad G2 \leftarrow 1$$

から、「すべての学生に研究室を割り当てる」という制約の方が「すべての研究室に学生を割り当てる」という制約よりも充足しにくいことを表している.

以下は、"$G1 \leftarrow stu1, \ldots, stu7$." に完備化を施した際のトップダウンのルールである. これらのラグランジュ乗数

$$10.0: \quad stu1 \leftarrow G1, \quad 2.1: \quad stu5 \leftarrow G1$$

$$10.1: \quad stu2 \leftarrow G1, \quad 8.9: \quad stu6 \leftarrow G1$$

$$1.0: \quad stu3 \leftarrow G1, \quad 9.9: \quad stu7 \leftarrow G1$$

$$8.9: \quad stu4 \leftarrow G1,$$

から、学生3と学生5に対しては配属を決めるのは簡単であるが、それ以外の学生については、配属を決めるのが難しいことが分かる.

この例では問題の規模が小さいため、ここで述べたことはほとんど自明であるが、より複雑な問題を解く際には、このような情報をうまく利用することで、なぜその解が出るのかを分かりやすく示すことができるだろう.

1 ← G1, G2.
G1 ← lab1, lab2, lab3, lab4.
lab2 ← S1L1 ∨ S2L1 ∨ S3L1 ∨ S4L1 ∨ S5L1 ∨ S6L1.
lab2 ← S1L2 ∨ S2L2 ∨ S3L2 ∨ S4L2 ∨ S5L2 ∨ S6L2.
...
G2 ← stu1, stu2, stu3, stu4, stu5, stu6, stu7.
stu1 ← S1L1 ∨ S1L2 ∨ S1L3 ∨ S1L4.
stu1 ← S2L1 ∨ S2L2 ∨ S2L3 ∨ S2L4.
...
inc ← S1L1, S1L2.
inc ← S1L1, S1L3.
...
inc ← S1L1, S2L1, S3L1.
inc ← S1L1, S2L1, S4L1.
...
$S1L1/10, S1L2/10, S1L3/ 1, S1L4/ 2,
$S1L1/10, S1L2/10, S1L3/ 1, S1L4/ 2,
$S1L1/10, S1L2/ 2, S1L3/10, S1L4/ 1,
$S1L1/10, S1L2/10, S1L3/ 2, S1L4/ 1,
$S1L1/ 2, S1L2/10, S1L3/ 1, S1L4/10,
$S1L1/10, S1L2/10, S1L3/ 2, S1L4/ 1,
$S1L1/10, S1L2/10, S1L3/ 1, S1L4/ 2,
labi = 研究室 i に学生が配属になることを表す命題
stuj = 学生 j がいずれかの研究室に配属になることを表す命題

図 7.2: 研究室の配属を決める問題

7.3 今後の仮説推論システム

仮説推論は，人間の思考のひとつの側面をうまく表しているのかもしれない．しかし，人間の思考過程，情報処理過程にとって必要不可欠であろう大量のデータの入力，それからの学習と行動という要素がない．人間の情報処理の全てをカバーしていないことは確実である．そもそも，知識は事象のある側面を抜きだして抽象化された記述であるから，さまざまな記述の仕方があり，それをいつどのように使うかは状況に依存する．記号 (デジタル情報) と実世界を結びつけ，意味や価値を与えるグラウンディング [Harnad 90][中島 01] の研究も行われており，記号と意味，そして状況依存性をうまく捉えていかなければならないだろう．

仮説推論には2章で述べたようにさまざまな応用対象がある．しかし，応用を目的としたモデルとしては限界もある．診断を目的とした実用システムの多くは，統計的な根拠に基づく処理を行っており，コストの和の最小化といった簡単な解の選好基準では表せないことも多い．例えば，ネットワークにおけるアラームからの故障箇所の推定 [Hashimoto 00] は，指数分布の畳み込みによる計算が必要となる．これを仮説推論として捉えるには，大幅なモデルの拡張が必要になる．情報処理システムで用いるひとつのモデルとしては，仮説推論の他にもさまざまな選択肢がある．他の知識表現かも知れないし，数学モデルかもしれないし，統計モデルかもしれないし，言語モデルかもしれない．

仮説推論などの人工知能モデルの依拠するところは，「人間の知能に近いように思える」点であろう．しかし，情報処理がシステム全体として語るべきものである以上，ひとつのモデルだけをとってこれが良い悪いということは難しいであろう．ユーザがどのような情報を求め，それに対してどのようなモデルで何を処理すればよいのか，それがそのユーザにどのようなメリットをもたらすのかというシステム全体から見たモデルの優劣こそが議論されるべきであろう．

以下の第 II 部では，文書からのキーワード抽出というテーマで話を進める．大量の電子的な文書は，コンピュータにとってはまさに暗号である．この中の何に着目するのか．どうやって意味を紐解いていくのか．どうやって人が，そしてコンピュータが，システムの

外にあるものに気づくことができるのだろうか.

第II部

文書主題抽出法

第8章　自然言語文からの知識獲得

自然言語のテキストから情報を発見／抽出するのがテキストマイニング，WWWから情報を発見／抽出するのがWebマイニングである．これらの技術が着目されるようになったのはごく最近で、text miningという語が最初に使われたのは1995年の第1回KDD（the First International Conference on Knowledge Discovery and Data Mining）であった．Webマイニングも96年に[Etzioni 96]で定義され、[Kosala 00]でその大きな分類がされている．

テキストマイニングは自然言語で自由に書かれた文書を扱う．したがって、データが構造化されておらず、生のデータをどのように処理するかを考えなければならないところが、データマイニングとの大きな違いである．したがって、自然言語で書かれた文を対象とし、データ選択、前処理、データマイニングを含んだ知識発見プロセスがテキストマイニングであると言えるだろう．

一方、Webマイニングは、Webを対象としたデータマイニングであるが、大きく次の3つに分けられる．ひとつは、Webページの内容のマイニングで、自然言語で書かれた文だけを取り出せばテキストマイニングと同じである．(実際には、他にもタグ情報、ハイパーリンクなども含まれているので、それらを利用することもできる.）2つ目は、Webページ間のグラフ構造からのマイニングであり、Webページのランキングなどが目的となる．3つ目は、ユーザのインタラクションからのマイニングで、ユーザの閲覧履歴を用いてユーザプロファイルを得るなどの目的がある．

8.1　テキストマイニング

テキストマイニングとは、文章データから役に立つ知識・情報を取り出す技術である[Tkach 98]．ある特定の情報を持つ文書の検索を行うのではなく、膨大な文書の中に記述さ

れている内容の傾向や相関関係などを分析することで、既存の知識ではない有用な知識・情報を得ることを目的としている．同じ目的をもつ研究にデータマイニングがあるが、データマイニングで扱うデータはデータベース・スキーマによってきれいに整理されているという前提があるのに対し、テキストマイニングは、形式化されていないテキストを目的としている．

たとえばアンケートで顧客の声を集めることを考えよう．その場合、顧客の意見にどのような内容がありそうかを最初から予想し、選択肢を用意したり表を埋めるような形式にしておけば集計が楽になる．反面、最初から仮定できる内容であれば、新たな知見は得難いというジレンマが生じる．そのため、アンケートには、選択肢形式の回答欄だけでなく自由回答形式の記述欄が含まれている場合が多い．そして、回答結果の中で貴重なのが、この自由回答部分と考えられる．ところが実際には集計の困難さから、アンケートの規模が大きくなるほど自由回答部分の活用度が低くなる傾向が見られる．したがって、このような文章形式のデータを分析し、その中から有効な知識を獲得する技術が重要となる [那州 99]．

データマイニングの技術を用いて、テキストの情報をマイニングできないかという発想はデータマイニングが注目を集め始めた当初から存在し、JUMAN や茶筅のような形態素解析ツールを用いれば、名詞句を中心としたキーワードを文書から抽出することが比較的容易に実現できる．しかし、単に名詞句を中心としたキーワードの集合に対し、一般的なデータマイニングの技術を適用しても、新たな知識を発見するという点ではあまり有効な結果は得られない．たとえば、語と語の相関関係を分析しようとすれば、『「データ」と「分析」は共起しやすい』などといった、実用的に価値のない結果が得られることが多い．すなわち、文書中から単なる文字列としてのキーワードを抽出し、その結果に対して一般的なデータマイニングの手法を適用しても有効な結果はなかなか得られない．

したがって、文書データを本格的にマイニングするには、概念を的確に抽出する、重要な単語・文・相関関係などを抽出するマイニングを行う、結果を表示するという３つの機能を適切に構成する必要がある．

一般的なデータマイニングでは、通常個々のデータの値は大きな意味を成さないのに対し、文書中では単語ごとに目的に応じた重要度が存在して、出現頻度が低くても重要視し

なければならない概念も存在する．このような重要視すべき概念をどのように認識するかも大きな課題となる．

8.2 Webマイニング

Webマイニングは、Webを用いたマイニングの総称であり、次のようなサブカテゴリに分類される [Kosala 00].

Web content mining

Web上のコンテンツ、データ、文書から、有用な情報を発見することである．対象とするコンテンツは、例えば、政府機関の情報、企業の新製品情報、財務指標、デジタルライブラリなど非常に多岐に渡る．また、データのタイプも、テキスト、画像、audio、video、メタデータ、ハイパーリンクなど、さまざまである．一般的には、構造化されていないテキストデータからのマイニングを指す．したがって、テキストマイニングは、Webコンテンツマイニングのひとつであるとも考えられる．

例えば，[Craven 98] では，Webページから人の学部や専攻，プロジェクトなどの情報を抽出する．オントロジーを用い，語の間の関連を計算することで情報を抽出している．また，著者は，電子掲示板から面白い話題を取り出す研究を行っている [松尾 02c]．談話構造における新情報，旧情報といった概念に着目し，語の継承をもとに，談話の構造を壊さずにしかも分かりやすく表示する．

Web structure mining

Web structure miningは、Webのリンク構造に隠されたモデルを発見する．Webページをノード、ハイパーリンクをエッジと考えたWebグラフを用いて、その性質を分析する．Webページの分類・ランキングや、異なるWebページ間の類似性を得るのに用いられる．例えば、検索エンジンとして評価の高いGoogle[goo]にはPageRankというアルゴリズム [Brin 98] が使われている．これはWebグラフの構造をもとにWebページをランキングする

手法である．また，ある領域で権威の高いauthorityページと，authorityページにリンクを張っているhubページをグラフ構造から見つけ出すKleinbergらの研究[Kleinberg 99]も有名である．hubとauthorityを効率的に見つける確率モデルも提案されている[Lempel 00]．

Webのリンク構造から，コミュニティを発見する研究も行われている．完全2部グラフを発見することによりコミュニティを発見する[Kumar 99]、参照の共起性によりコミュニティを抽出する[村田 01]などである．また，[Flake 00]では，最大流量最小カットの定理を用いて，コミュニティとそれ以外を分ける適切なカットを見つけている．

Webのマクロ構造に着目したものとしては，Webが蝶ネクタイ構造をしており，in-degreeとout-degreeがPower Lawにしたがうという研究[Broder 00]や，Webのトポロジーが Small World（後述）であることを述べたもの[Adamic 99]などがある．

また，著者は，リンク構造とユーザの心理的な距離間に着目し，average-clickという尺度で計ったページ間の関連性が，ユーザの心理的な距離間と密接な相関を持つことを明らかにした[Y.Matsuo 01, 松尾 02a]．他にも，Refferal Webという Web上の情報から人間関係を見つけ出す研究[Kautz 97]である．このように，Webには社会的なネットワークとしての側面も大きい．

Web usage mining

Web上のユーザの振る舞いのデータを利用する技術である．主に、Webサーバのログ、プロキシサーバのログ、クライアントログ（ブラウザのキャッシュ、クッキーなど）のいずれかを用いて、Webページとユーザのインタラクションという2次的なデータを用いる．前処理を行った後に、ログデータをデータマイニング手法を用いて解析することになる．

Web usage miningの利用法は大きく2つあり、ひとつは、ユーザプロファイルを得て、適応的な個人指向のインタフェースを実現することである．もうひとつは、情報提供者側がユーザの振る舞いに合わせてWebページのデザインや構成を変えるのに使われる．その他、ネットワーク／システムの改善、ビジネスの知見を得る、ユーザの振る舞いを知るなどの目的に使われる．

8.3 キーワード抽出の従来研究

さて，本論文の以後の章では，いくつかのキーワード抽出アルゴリズムについて述べる．ここでは，キーワード抽出に関する従来研究を概観しておこう．

キーワードとは，文書中で重要な意味を担う語という意味である．文書から重要な語を抽出するのがキーワード抽出である．「キーワード抽出」という語は，主にテキストマイニングで使われる語であり（例えば [Rajman 98]），情報検索の分野では自動索引づけ (automatic indexing) とか自動キーワード抽出 (automatic keyword extraction) と呼ばれる．主に，文書を検索するための適切な索引語を見つける目的である．また，計算機言語学では，自動用語抽出 (automatic term recognition) と呼ばれる [Kageura 96]．日々，膨大な新語が創出されるなかで，どのように専門用語を抽出して，辞書の作成等に役立てるかという目的がある．

いずれにしても，文書における語の重要性を計るのは，本質的である．テキストマイニングや発想支援という立場からも，適切なキーワードを自動的に抽出することができれば，読むべき文書を選択しやすくなったり，文書間の関係を把握することが容易になるなどのメリットがある．さらに，文書の傾向をつかむ，特徴的な意見を見つける，新しい知見を得るといった用途に必要不可欠である．実際に，多くの情報検索／テキストマイニング手法が何らかの形でキーワード抽出技術を用いている．

さて，キーワード抽出，または語の重みづけは，情報検索の分野で1950年代から行われていた．最も単純な重みづけ，すなわち文書中の語の頻度による重みづけは，1957年のLuhnの研究 [Luhn 57] にまで遡る．その語，より詳細化した手法 [Sparck-Jones 72, Noreault 77] も提案されたが，基本的には，語の頻度を数えるものである．また，「要するに」などの手がかり語をもとにキーワードを抽出する方法 [Edmundson 69][木本 91] などもある．しかし、頻度による方法は単純すぎて一般的な語も抽出してしまうし、手がかり語による方法は汎用性がない．

索引づけという観点からは，語が文書に書かれていることをどのくらい網羅しているかという網羅性 (exhaustivity) と，他の文書と区別するのにどのくらい役に立つかという特

定性 (specificity) の両面を考えることが必要である [徳永 99]．特定性を高くするには，その文書には現われるが，他の文書には現われないような索引語を選択すればよい．極端な例では，その文書にしか現われないような索引語を選べば，その索引語を検索質問で用いられれば，その文書のみを選択することができる．しかし，このように文書にあまりに特化した索引語だけを選ぶと，検索質問でその索引語が用いられる可能性も低くなる．一方，一般に文書によく用いられる語を索引語として用いると，多くの文書の索引語となってしまう．したがって，特定性と網羅性は相互排他的な関係にある．

また，自動用語抽出の文脈では，語がどのくらいひとつの単位として用いられるかという unithood と，単独でどのくらい文書の内容を表すかという termhood の 2 つの面を考慮しなければならない．例えば，文書中で “digital computer” という語が用いられていれば，“digital” という語は，unithood の点で “digital computer” に劣る．

現在，情報検索システムで最もよく用いられている手法が，tfidf である．tfidf は，その文書にはよく出現するが，他の文書にはあまり出現しない語を高く評価する．一般的に，文書 d における索引語 t の重みは，次のような式で与えられる．

$$tfidf(t,d) = tf(t,d) \times idf(t).$$

頻度 (term frequency) $tf(t,d)$ は，語 t の文書 d における出現回数である．idf (inverse document frequency) はさまざまなものがあるが，例えば，

$$idf(t) = \log / fracNdf(t) + 1$$

がよく用いられる．tfidf の他にも，ある文書集合にだけ偏って出現する語は特徴的である [長尾 76][Dunning 93]，文書集合中で共起する語が少ないほど特徴的である [寺本 99]，共起する単語分布の偏りが大きい語ほど特徴的である [Hisamitsu 00] などの手法も提案されている．これらは，文書集合中の語の分布をもとに，統計的／経験的な尺度を用いてある文書（集合）を代表する語彙を自動識別する方法である．

影浦は，語の重要度の計算の方法として次のような 5 つの基本的なタイプに分類した [Kageura 96].

- その文書に現われる語は，索引語になりやすい．
- その文書に頻繁に現われる語は，索引語になりやすい．
- 限られた数の文書に現われる語は，それらの文書の索引語になりやすい．
- 全体の文書よりもある文書に頻繁に現われる語は，その文書の索引語になりやすい．
- 全体の文書で特徴的な分布を示す語は，全体の文書の索引語になりやすい．

第2部の以下の章では，主に，単独の文書からのキーワード抽出を行う．従来は，「頻繁に出現する」ことでしか判断できなかった重要性を，語の共起やネットワークといった側面から重要性を判断する枠組みである．

第9章　語の共起情報を用いたキーワード抽出

本章では，語の共起をもとに統計的な指標を用いキーワードを抽出する手法について述べる．コーパスを必要とせず単一の文書から *tfidf* に匹敵する精度でキーワードを抽出することができる[1]．

9.1　語の文内共起と重要語

文書中に出現する単語は、文毎に句点やピリオドによって区切られている．以下では、同文中に出現する2つの語は1回共起していると考える．すなわち、各文をひとつの「バスケット」として捉え、（n-gram 処理以外の）順序関係については考慮しない．

さて、ひとつの文書が与えられたとき、単語の出現頻度を数えることで、頻出語を取出すことができる．例として、コンピュータ科学の分野で非常に有名なアラン・チューリングの論文 “Computing machinery and intelligence”[Turing 50] を取り上げよう．表9.1に、頻出語上位10個の出現頻度と出現確率（全体が1になるように正規化したもの）を示す．語の出現頻度は、経験的に Zipf の法則 (順位 r と頻度 f の積が定数 C になる (第1法則))[徳永 99] に従うことが知られている．

次に、語の共起の頻度を集計することにより、表9.2 のような共起行列を作ることができる．(表形式で表している.) この行列は、例えば語 a と語 b は22文で共起していることを示している．共起行列は、文書中に出現する語の数を N とすると $N \times N$ の対称行列であるが、ここでは頻出語上位10語 (G とする) に対応する列だけを抜きだし、$N \times 10$ 行列としている．対角成分は、ここでは定義しない．

[1]なお，本手法は，「データからの特徴アイテム抽出方法」(特願 2001-254905，出願日 平成13年8月24日) として特許出願を行っている．

表 9.1: 文書全体における頻度と確率分布

頻出語	a	b	c	d	e	f	g	h	i	j	合計
頻度	203	63	44	44	39	36	35	33	30	28	555
確率	0.366	0.114	0.079	0.079	0.070	0.065	0.063	0.059	0.054	0.050	1.0

a: *machine*, b: *computer*, c: *question*, d: *digital*, e: *answer*, f: *game*, g: *argument*, h: *make*, i: *state*, j: *number*

表 9.2: 共起行列

	a	b	c	d	e	f	g	h	i	j	Total
a	—	30	26	19	18	12	12	17	22	9	165
b	30	—	5	50	6	11	1	3	2	3	111
c	26	5	—	4	23	7	0	2	0	0	67
d	19	50	4	—	3	7	1	1	0	4	89
e	18	6	23	3	—	7	1	2	1	0	61
f	12	11	7	7	7	—	2	4	0	0	50
g	12	1	0	1	1	2	—	5	1	0	23
h	17	3	2	1	2	4	5	—	0	0	34
i	22	2	0	0	1	0	1	0	—	7	33
j	9	3	0	4	0	0	0	0	7	—	23
...	...	...	...	...	...	...	...	...	...	...	...
u	6	5	5	3	3	18	2	2	1	0	45
v	13	40	4	35	3	6	1	0	0	2	104
w	11	2	2	1	1	0	1	4	0	0	22
x	17	3	2	1	2	4	5	0	0	0	34

u: *imitation*, v: *digital computer*, w:*kind*, x:*make*

さて、仮に、語 w が頻出語 $g \in G$ と全く独立に生起するなら、語 w と語 $g \in G$ が共起する確率は表 9.1 の確率と同様の分布になるはずである．一方、語 w と頻出語 $g \in G$ の間に何らかの意味的なつながりがあれば、この確率は偏ることになる．

図 9.2、図 9.1 に、いくつかの語と語 $g \in G$ との共起確率[2]の分布を示す．図中に標準として、語 $g \in G$ の単独での出現頻度の分布（表 9.1）を示している．“kind” や “make” などの語は、どの頻出語 $g \in G$ とも偏りなく用いられるのに対し、“imitation” や “digital computer” などの語は特定の頻出語と選択的に多く共起している．このような偏りは、筆

[2]合計が 1 になるように正規化したもの

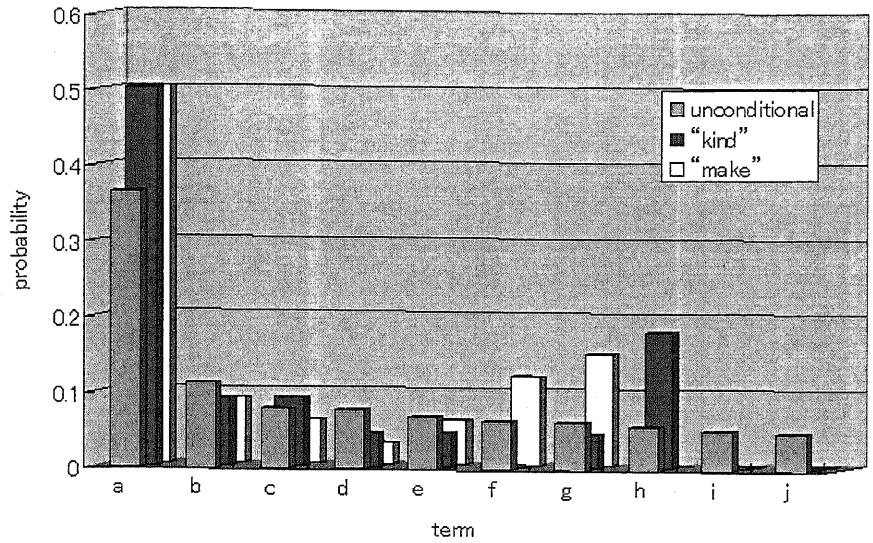


図 9.1: 語 *"kind"*, *"make"* の頻出語との共起の確率分布

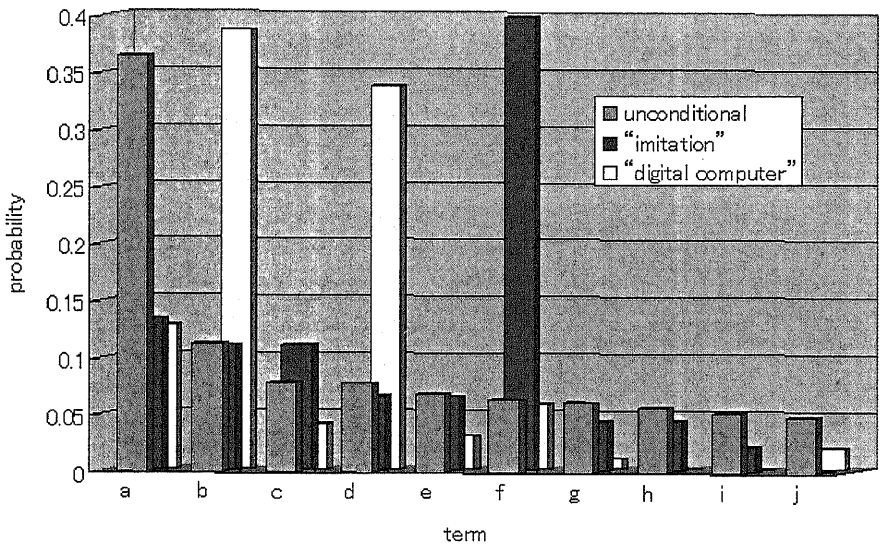


図 9.2: 語 *"imitation"*, *"digital computer"* の頻出語との共起の確率分布

者が意味的なつながりを考慮し文書を書き進めていく上で生まれたものであり、分布が偏っている語は文書中において何らかの重要な意味を担っている語であると考えられる．実際に、このチューリングの論文は、よく知られているように「機械は思考できるかという問いを、imitation game によって置き換える」ことを提案しており、"imitation" や "digital

computer" などの語は、論文中で重要な語である.

このように、ある語の頻出語 $g \in G$ に対する共起確率が、頻出語単独での出現確率から大きくずれていれば、その語は特徴的な語であり、論文中で意図的に用いられている重要な語である可能性が高いと考えることができる. すなわち、ひとつひとつの単語について、各頻出語との共起頻度を標本値とし、「$g \in G$ の出現する確率は単語 w の出現いかんに関わらず等しい」を帰無仮説として検定を行えばよい.

ここでは、χ^2 検定を用いる. 頻出語単独での生起確率 (表 9.1) を理論確率 $p_g(g \in G)$ とし、語 w と頻出語群 G の共起の総数を n_w、語 w と語 $g \in G$ の共起頻度を $freq(w, g)$ とすると、統計量 χ^2 は以下の式で与えられる.

$$\chi^2(w) = \sum_{g \in G} \frac{(freq(w, g) - n_w p_g)^2}{np_g}$$

$\chi^2(w) > \chi^2_\alpha$ であれば、帰無仮説が有意水準 α で棄却される. (χ^2_α は通常 χ^2 分布表より得る.) したがって、$\chi^2(w)$ の大きな語 w が理論確率分布からのずれが大きな語である. なお、本手法では χ^2 値を検定法としてではなく、単純に偏りの程度を示す指標、度合いとして用いている.

表 9.3 に、前述の例 [Turing 50] に対する χ^2 値の高い語と低い語を示す. (出現頻度の上位 10 語を G としている.) 表から分かる通り、χ^2 値の高い語は論旨に関係の深い語が並んでおり、χ^2 値の低い語は一般的な語である傾向が強い.

すなわち、本手法はまず、頻出語を取り出すことによって文書自身の全体的な傾向を求め、この傾向から大きく逸脱する特徴を持つ語をキーワードとして取り出す.

9.2 手法の詳細とアルゴリズム

本手法の大略は上述の通りであるが、単一の文書からキーワードを抽出するわけであるから、キーワードの精度を上げるために、さまざまな工夫が必要である.

表 9.3: χ^2 値による語のソート

Rank	χ^2	Term	Frequency
1	593.7	digital computer	31
2	179.3	imitation game	16
3	163.1	future	4
4	161.3	question	44
5	152.8	internal	3
6	143.5	answer	39
7	142.8	input signal	3
8	137.7	moment	2
9	130.7	play	8
10	123.0	output	15
⋮	⋮	⋮	⋮
551	1.0	slowness	2
552	1.0	unemotional channel	2
553	0.8	Mr.	2
554	0.8	sympathetic	2
555	0.7	leg	2
556	0.7	chess	2
557	0.6	Pickwick	2
558	0.6	scan	2
559	0.3	worse	2
560	0.1	eye	2

9.2.1 χ^2 値の計算について

文書中の文の長さは様々であり、長い文に出現する語は他の語と共起しやすく、短い文に出現する語は他の語と共起しにくい．つまり、短い文で共起しているときには、その関係はより強いと考える方が自然であろう．したがって、以下のような変更を行う．

- p_g を、(g が出現する文数)/(G 中の語が出現する延べ文数) ではなく、(g が出現する文の語数の合計)/(文書全体の語数の合計) とする．
- n_w を、語 w が出現する文の語数の合計とする．

これにより、文の長さを考慮した、より正確な計算結果が得られる．なお、G に含まれる語についても χ^2 値の計算を行うが、その際には自分自身との共起は計算に含めない．

表 9.4: Transposed two columns.

	a	b	c	d	e	f	g	h	i	j	...
c	**26**	**5**	—	**4**	*23*	**7**	**0**	**2**	**0**	**0**	...
e	**18**	**6**	*23*	**3**	—	**7**	**1**	**2**	**1**	**0**	...

表 9.5: Clustering of the top 49 frequent terms.

C1: game, imitation, imitation game, play, programme
C2: system, rules, result, important
C3: computer, digital, digital computer
C4: behaviour, random, law
C5: capacity, storage
C6: question, answer
... ...
C26: human
C27: state
C28: learn

頻出語中の特定の 1 語 $g \in G$ とだけ共起する語は χ^2 値は高くなるが、重要な語であるというより、語 g に付随する語である場合がほとんどである. 例えば、前述の例では、"future" や "internal" は "state" とだけ選択的に共起するため、χ^2 値は高くなる. これは、論文中で "future state"、"internal state" という決まった形で使われるためであるが、"future" や "internal" が重要な語かというと、そうではない. こういった語は、仮に "node" が G に含まれないとすると、χ^2 値は急に低くなる. そこで、分布の偏りをロバストに算出する目的で、χ^2 値の最大の項を除いた値

$$\chi'^2(w) = \chi^2(w) - \max_{g \in G} \left\{ \frac{(freq(w,g) - n_w p_g)^2}{n_w p_g} \right\}$$

を重みづけの関数として用いた.

9.2.2 語のクラスタリング

本来、共起行列は $N \times N$ 行列であるが、本手法では、そこから基準となる語の集合 G に対応する列を抜きだしている. G との共起が特徴的な語 (分布の特異な行) を選び出すの

で、G を適切に定めることがキーワード抽出の性能をあげる上で極めて重要である.

文書中に多く登場する頻出語を G として選ぶという方針は、頻出語の方がそうでない語よりも文書のコンテキストを表す可能性が高い点、標本数が多いため数値がより信頼できるという 2 点から適切であると考えられる. しかし、言い替えを行っていたり、近い概念を表す語は、ひとつのクラスタとする方がよい.

文書中の語をクラスタリングする研究は数多く行われているが、大別すると次の 2 つに分けられる [Hofmann 98].

類似性 語 w_1 と語 w_2 の他の語との共起の分布が似ていれば、同じクラスタとする.

共起 語 w_1 と語 w_2 が頻繁に共起していれば、同じクラスタとする.

表 9.4 は共起行列から 2 つの列を抜き出したものであるが、前者はこの列の太字部分に着目し、後者は斜体字部分に着目していることに相当する.

類似性によるクラスタリングでは、例えば"Sunday"、"Monday"、"Tuesday"、... や、"build", "establish", "found" など、同じような働きをする語が同じクラスタとなる. 我々の予備実験では、言い替えを行っている語や、"digital computer" と "computer" のように、フレーズとその要素語をひとつのクラスタとする傾向が多く見られた. 2 つの分布の類似性は、Kullback-Leibler divergence や Jensen-Shanon divergence といった統計量により計ることができる [Dagan 99][Pereira 93]. Jensen-Shanon 統計量は, 次の式で表される.

$$\begin{aligned} J(w1, w2) &= \log 2 + \frac{1}{2} \sum_{w' \in C} \{h(P(w'|w_1) + P(w'|w_2)) \\ &\quad - h(P(w'|w_1)) - h(P(w'|w_2)\} \end{aligned}$$

ただし、$h(x) = -x \log x$、$P(w'|w_1) = freq(w', w_1)/freq(w_1)$ である.

一方、共起によるクラスタリングは、関連のある語が同じクラスタとなる. ひとつひとつの語の意味は異なることが多いが、クラスタ全体としてひとつの概念を表しやすい. 例えば、"doctor", "nurse", "hospital" などがクラスタとなる [Tanaka-Ishii 96]. 共起頻度 $freq(w_1, w_2)$ や相互情報量を用い、関連の強さを計る [Church 90][Dunning 93]. 語 w_1 と語

w_2 の相互情報量は，次のように表される．

$$\begin{aligned} M(w_1, w_2) &= \log \frac{P(w_1, w_2)}{P(w_1)P(w_2)} \\ &= \log \frac{Nfreq(w_1, w_2)}{freq(w_1)freq(w_2)} \end{aligned}$$

さて、我々の目的は、ひとつには言い替えや同義語を同じクラスタにまとめてしまうこと、もうひとつは、関連した概念も同じクラスタにまとめることである．したがって、2つのクラスタリング手法をともに用いる必要がある．ここでは、類似性によるクラスタリングでは Jensen-Shannon divergence を用い、共起性によるクラスタリングには相互情報量を用いた．適切なクラスタ化を行うことで、共起行列の列がマージされ行列中の標本が多くなる．図 9.5 に前述の例におけるクラスタリングの例を示す．

9.2.3 キーワードの提示

さらに、語の重みづけと、どの語をキーワードとして提示するかは別の問題である．例えば、ある文書に何が書かれているか簡単に把握したいときに、“digital”, “computer”, “digital computer” などが全てキーワードとして表示されることは望ましくない．そこで、同じクラスタに属する複数の語が出力結果に含まれるときは、そのうち χ^2 値が最も高い語だけを提示する．

9.2.4 アルゴリズム

具体的なアルゴリズムを示す．

1. 前処理：英語の場合には、stemming を行う[3]．日本語の場合には、形態素解析を行い[4]、分かち書きをする．さらに，フレーズを取り出す[5]．*stop word*[6]が与えられてい

[3]語幹の形を得る処理．”···ing”,”···ed”, 三単元の s などを取り除く．ここでは，Porter の方法 [Porter 80] を用いる．

[4]取り出す品詞は，基本的に，名詞，動詞，形容詞，副詞，未知語である．ただし，代名詞や非自立語は除くなどの設定を行っている．

[5]Apriori 的な手法により，出現回数が 3 回以上の 4-gram までのすべてのフレーズを取り出す [Fürnkranz 98]．

[6]“a”,”the”,”that” などのあらかじめ決められた不要語．ここでは，Salton の SMART System のリスト [Salton 88] を用いた．

る場合には、これを取り除く.

2. 頻出語の選択：文書中の語の延べ総数 N_{total} の 30%に達するまで頻出語の上位語を取り出す.

3. 頻出語のクラスタリング：頻出語間の類似度の特徴量（Jensen-Shanon divergence）が閾値 $(0.95 \times \log 2)$ を越えるものは、クラスタとしてまとめる. 共起の相互情報量が閾値（$\log(2.0)$）を越えるものも、クラスタとしてまとめる. 得られたクラスタ群を C とする. 以下、クラスタ $c \in C$ との共起とは、クラスタ中のいずれかの語との共起を指す.

4. 理論確率の計算：クラスタ $c \in C$ と同文中で共起する語の延べ総数 n_c を調べ、理論確率 $p_c = n_c/N_{total}$ を求める.

5. χ^2 値の計算：すべての語 w について、w と $c \in C$ との共起頻度 $freq(w,c)$ と、w が出現する文の語の総数 n_w を求める. χ'^2 値を以下により求める.

$$\chi'^2(w) = \sum_{c \in G} \left\{ \frac{(freq(w,c) - n_w p_c)^2}{n_w p_c} \right\} - \max_{c \in G} \left\{ \frac{(freq(w,c) - n_w p_c)^2}{n_w p_c} \right\}$$

6. キーワードの提示： χ'^2 値の上位語を一定数提示する.

9.3 評価

本手法の実行例を示す. 分かりやすいように、対象とする文書はこの章とほぼ同内容の論文 [松尾 02b] を取り出す. 頻出語の上位は表 9.6 の通りである. 頻出語の上位 18 語（累計で総語数の 30%に達する）をクラスタリングすると、図 9.7 となる. これらのクラスタに対して、χ^2 値を計算したものが図 9.8 である. 表中の "+" はフレーズを表している.「キーワード抽出」や「χ^2 値」といった語が、上位にきていることが分かる.

さて、本手法のキーワード抽出の精度を評価するために、評価実験を行った. 評価実験

表 9.6: 論文 [松尾 02b] に対しての頻出語

順位	頻度	ラベル
1	147	語
2	50	共起
3	36	文書
4	29	出現
5	25	キーワード
6	23	値
7	23	手法
8	22	頻出
9	21	中
10	21	χ^2

は，人工知能の分野の 7 著者 20 論文に対して行い，tf，tfidf[7]，KeyGraph[8] と比較した．各手法でキーワード 15 個を出力し，各手法から得られたキーワードの上位語を混ぜてシャッフルし，著者に「論文を構成する重要な概念を表すと思う語にチェックをして下さい」という質問を行った．各手法による出力語中でキーワードであると判定された割合が precision である．さらに,「提示した全ての語（提示した以外の語でも覚えているものがあれば含めてよい）のうち，論文中で不可欠な概念を表す語 5 つ以上を選び A，B，C，D，E と印をつけ，それと同義の語にも同じ印をつけてください」という指示を行った．5 つ（以上）の概念のうち各手法で提示した語にいくつ含まれているかで coverage を測定した．

結果を表 9.9 に示す．どの手法も *precision* が 0.5 前後であるが、*coverage* は tf や KeyGraph よりも高く、大量のコーパスを必要とする tfidf に匹敵する性能が得られている．また、tf や tfidf は文書中でよく出てくる語の重みを大きくするので、出てくる語は当たり前の語が多い．それに対し、本手法では出現頻度が少なくても重要な語を取り出している．それを数値化したものが frequency index で、これは提示した語の出現頻度の平均を表している．tf は文書中に平均して 30 回近く出現する語を提示しているのに対し、本手法は平均 11.5 回出現する語を提示しており、それでいて同程度の precision を得ているところは評価

[7]コーパスは JAIR(Journal of Artificial Intelligence Research) の 93 年 (Vol.1) から 2001 年 (Vol.14) までの論文全文 166 篇とした．また，語 v に対する idf の重みづけは $\log(D/df(w))+1$ とした．ただし D は全文書数，$df(w)$ は語 w が出現する文書数である．

[8]本手法と同様に構造的な特徴からキーワードを抽出するため，コーパスは不要である．

表 9.7: 頻出語上位 18 個のクラスタリング

C1: 語、共起
C2: 文書
C3: 出現
C4: キーワード、抽出
C5: χ^2、値
C6: 手法
C7: 頻出、語+共起、頻出+語
C8: 中
C9: 確率
C10: 用いる
C11: 行う
C12: クラスタ
C13: 頻度

表 9.8: 論文 [松尾 02b] に対しての χ^2 値上位の語

順位	χ^2 値	頻度	ラベル
1	126.1	147	語
2	81.2	14	キーワード+抽出
3	68.1	12	χ^2+値
4	45.6	20	確率
5	42.7	5	頻出+語
6	40.7	5	文書+キーワード+抽出
7	38.7	29	出現
8	35.0	5	分野
9	34.6	5	低い
10	34.3	17	語+共起

できるだろう．また，上位 15 位までの語を対象とした場合に本手法の precision は 0.51 だが，これを上位 10 語までとすると precision は 0.52 に，上位 5 語では 0.60 に，上位 2 語では 0.72 となる．したがって，χ^2 値の値をキーワードの優先度とすることが可能である．

本手法では，フレーズもキーワードとして抽出するが，表 9.10 にフレーズの含まれる割合とフレーズを除いた場合の結果について示す．精度や再現率は下がるものの，tfidf に準ずる結果となっている．

表 9.9: Precision と Coverage

	tf	KeyGraph	**本手法**	tfidf
precision	0.53	0.42	**0.51**	0.55
coverage	0.48	0.44	**0.61**	0.61
frequency index	28.6	17.3	**11.5**	18.1

表 9.10: 得られた結果におけるフレーズの内訳

	tf	KeyGraph	**本手法**	tfidf
フレーズの割合	0.11	0.14	**0.33**	0.33
フレーズを除いた precision	0.42	0.36	**0.42**	0.45
フレーズを除いた recall	0.39	0.36	**0.46**	0.54

結果を定性的に評価すると、本手法は頻出語を基準とするが、tf による頻出語ですでに十分よいキーワードになっている場合には、本手法の提示する語は概念を特定しすぎた語になっているケースが多かった．逆に、頻出語が一般的な語で情報量が少ない場合には、本手法の提示する語が適切なキーワードとなっているケースが多かった．したがって、本手法は tf に置き換わるような手法ではないが、組み合わせて用いることで、より適切なキーワードの抽出ができると考えられる．

本手法の大きな特徴のひとつは，大規模なコーパスを必要としない手軽さにある．実験で用いた論文および JAIR の各論文についての時間と語の数のプロットを図 9.3 に示す．プログラムは C++で記述し，Celeron 333MHz の Linux OS 上に実装している．処理時間は，ほぼ語数に対して線形なオーダで増えており，20000 語程度なら数秒で処理が終了する．

9.4 関連研究との比較・考察

本章では、語の共起関係によりキーワードを抽出するが、語の共起に着目した研究は非常に多く行われている．[Pereira 93] では、ニュース記事 44 万語から、語を複数のクラスタに分割している．[Even-Zohar 99] や [Tanaka-Ishii 96] は、複数のクラスタに属するような同義語を適切に処理する方法を示している．[Tanaka 96] では、2 言語の共起行列を用いて、コンテキストを考慮した訳語の割り当てを行っている．

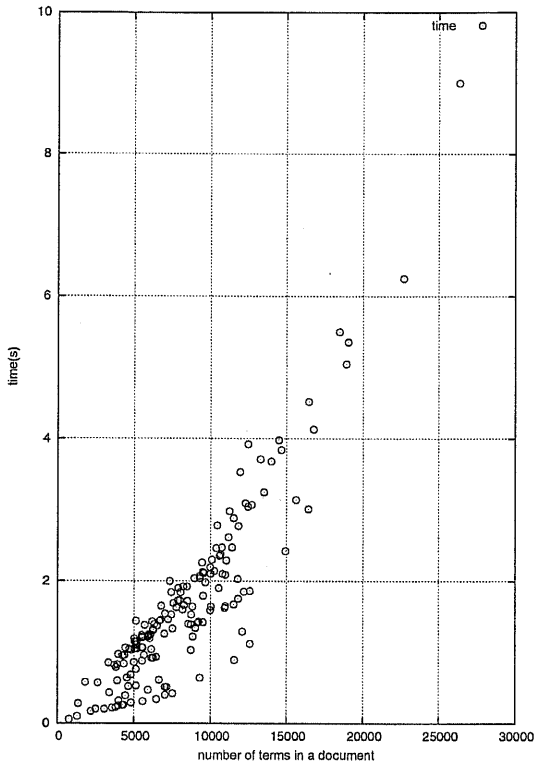


図 9.3: 語数と処理時間

また、確率的な視点からの研究も行われており [Dagan 99]、一回も出現していないフレーズの理論的な確率をどのように推測するかという zero frequency に関する議論もされている [Witten 91, Dagan 93]. しかし、これらの研究は大量のコーパスを用い、シソーラス作成のための語のクラスタリングや翻訳が目的となっている.

語と文書の共起関係に関する研究も多く行われている [相澤 00]. これらは基本的には大量のコーパスを背景とした特徴語抽出である. 例えば、[長尾 76] では、χ^2 検定により一般語と重要語を区別している. 全文献をいくつかの分野に分けて、分野ごとの単語の頻度を数え、ある単語が各分野に偏りなく出現すれば一般語で、少数の分野に偏って出現すれば重要語としており、χ^2 値を用いている. また、文書をカテゴライズするための重要語抽出に χ^2 検定を用いる方法もいくつか提案されている [Schutze 95, Ng 97, 大平 99]. また、「特徴的な語は共起する語の種類が少ない」という本手法と類似の考え方を用いた語の重みづけも最近、提案されている [Hisamitsu 00]. しかし、いずれもコーパスを背景とした方法であり、単一の文書からキーワードを取り出すものではない.

9.5 キーワード提示によるブラウジング支援

ここまで述べた頻出語は，当該文書に出現する頻出語であり，「その文書における」重要語であった．しかし，重要語とは読む人にとって変わるものではないだろうか? 読む人にとっての重要語とは何だろうか?

文書を読む人にとっての重要語とは，その人にとって情報量の多いような語であろう．では，その人にとって情報量の多い語とは何だろうか? おそらく，興味のある分野，自分に親しい話題でありながら，まだ知らないような情報であろう．例えば，「モーニング娘。」に興味のある人は、その語と偏って共起する「ハロープロジェクト」をキーワードと思うかもしれない。しかし、「モーニング娘。」に興味のない人には，それほど情報量は多くないだろう。

興味のある分野や自分の親しい話題というのは，その人が見てきた文書の履歴をとれば、その文書集合における頻出語をとることによって表すことが可能だろう．したがって、その履歴の文書中における頻出語と偏って共起するような語をキーワードとして提示すればよいと考えられる。

このような仮定にたち，Web をブラウズするユーザにキーワードを提示するシステムを構築した [福田 02a]．これは，ブラウジングする際に，Proxy を仲介することによってユーザが見た文書の頻度情報を保存し，閲覧中の文書の中でその頻出語との偏りが大きな語をハイライトして提示するというものである．

図 9.4 にシステム構成を示す．ユーザは，Proxy を通してインターネットにアクセスすることになるが，その際，語の出現情報がデータベースに保存される．(このデータベースは，かなり個人情報を含むことになるので，その取り扱いには注意する必要がある．) そして，ユーザが現在閲覧しているページにおいて，頻出語 (履歴における頻出語) は青でハイライトされ，頻出語と偏って共起している語は赤でハイライトされる (図 9.5)．

定性的な評価としては、青で表示される頻出語は、ユーザがよく目にするような語であり、ごく当たり前の情報量の少ない語が多い。しかし、赤で表示される語は、その文書のユーザの興味を引くような語となっている。また、履歴情報を用いずにキーワードを提示

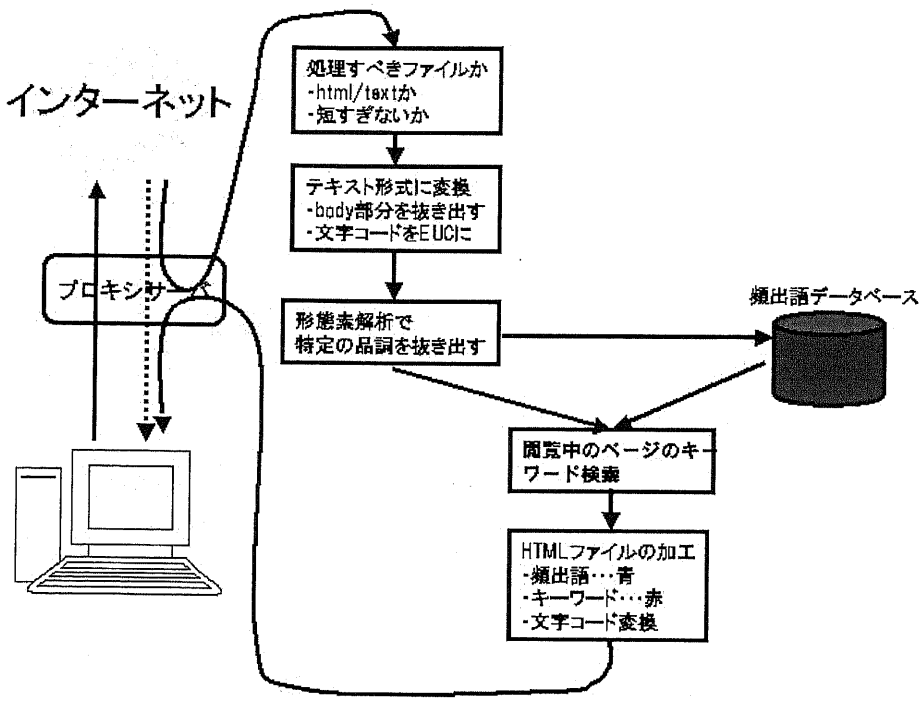


図 9.4: キーワード抽出によるブラウジング支援システム

した場合よりも、履歴を用いた方がよい結果が得られている。定量的な評価については、現在調査中であり、[福田 02b] で報告する予定である。

ここでのキーワードの抽出と提示システムは、ユーザにとって興味のある語を履歴から判断しようという野心的な試みであり、的確な評価を行うのは難しいが、少なくとも重要語に関して従来にない視点を提案していると考えている。

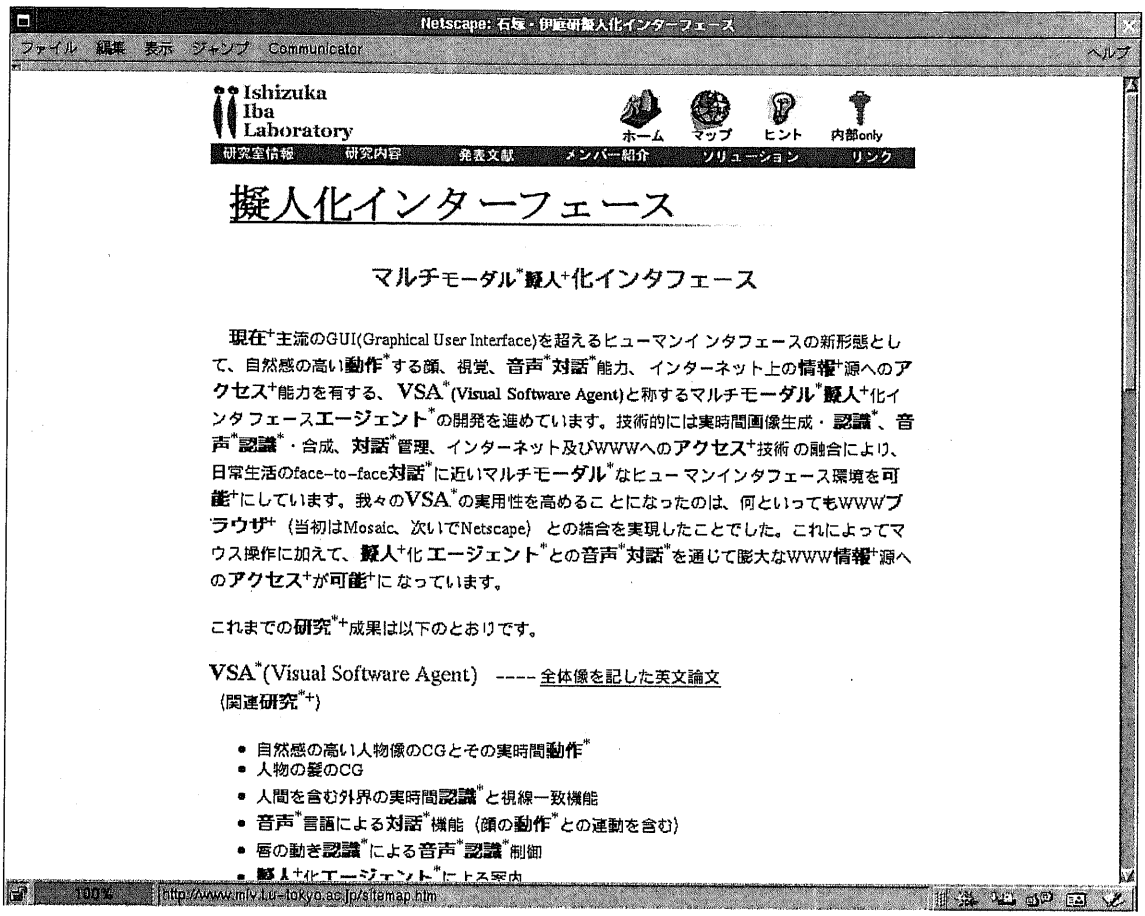


図 9.5: キーワード抽出によるハイライト例

色によるハイライトなので，図では分かりづらいが．青 (ユーザにとっての頻出語) には+マークが，赤 (ユーザにとってのキーワード) には*マークがついている．

第10章　Small World構造に基づくキーワード抽出

前章のキーワード抽出法は、語の共起に着目した統計的な情報に基づくキーワード抽出法であった．語の共起は，2つの語の間のなんらかの関係性を意味するという仮定に基づき，共起の偏りのある語を抽出した．では，語の共起全体は，どのような全体像を構成するのだろうか?

本章では，語をノード，共起をノード間のリンク関係ととらえ，ひとつの文書からグラフを形成する．そのグラフの中で重要な位置を占める語をキーワードとして抽出する．

10.1　語の共起グラフ

文書中に出現する語の共起関係は、全体として構造を持つ．各ノードが語を表し、共起関係によりリンクを張ることで語の共起グラフを構成することができる．

具体的には、以下のような手順で構成する．

1. stemming を行う．stop word を取り除く．n-gram によりフレーズを抽出する．これらの処理は，前章の処理と同じである．

2. 頻出数の上位語（フレーズも含む）をノードとして決められた数だけ選ぶ．

3. リンクの生成：2つのノードに対応する語の，同一文中での共起が多ければリンクを張る．共起はJaccard係数を用いて計り，この上位から順にkが既定値(k_0)に達するまでリンクを張る．

Jaccard係数とは，(両方の語を含む文の数)/(少なくとも一方の語を含む文の数)である．例えば[Kautz 97]で，Web上の文書から人名の共起グラフを構成する際に用いられている．

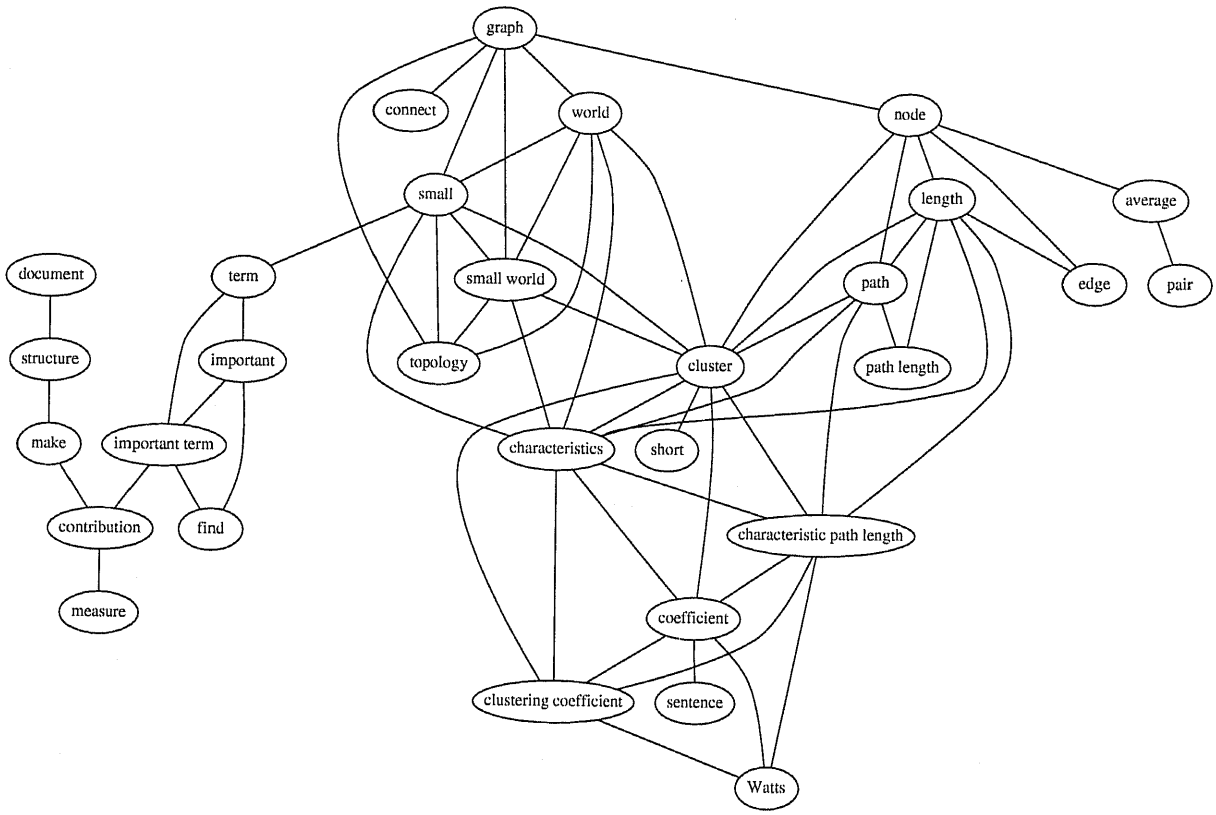


図 10.1: 論文 [Matsuo 01] の語の共起グラフ

図 10.1 は，本章とほぼ同じ内容の英語の論文 [Matsuo 01] から得た共起グラフである．密接にリンクのある語の集まり (クラスタ) があり，それらがつながれている様子が分かる．このようなグラフはどのような特徴を持つのだろうか．次節では，最近着目を集めている Small World というグラフのトポロジを紹介する．

10.2 Small World とは

Small World は，例えば飛行機でとなりに乗り合わせた人が共通の友人を発見し，「せまい世界ですね (It's a small world)」と話し合うといった現象に端を発する．このような経験は誰しもよくあるが，いったいこのようなことが起こる確率はどのくらいなのだろうか．世界の任意の 2 人は，いったい何人の仲介者を経由すれば互いに知り合いだろうか．

1960 年代にこの問題に対して興味深い実験を行ったのが著名な社会心理学者である Stanley Milgram である．その実験とは次のようなものである [Milgram 67]．

Aという人物からZという人物へのリンクを考えよう．Aをstart person、Zをtarget personと呼ぶ．start personとtarget personをランダムに選ぶ．start personは手紙をtarget personに向けて、自分の友達か知り合いに転送する．手紙はfirst-nameを互いに知っている人にだけ転送される．starting personとして参加してくれる人には、文書の入った封筒が送られる．文書には以下の事項が書かれている．

- target personの名前とその人の情報．
- target personに到達する方法．最も重要なのは以下であろう．「target personを個人的に知らないのであれば、彼と連絡をとろうとしないで下さい．そのかわり、target personを自分より良く知っていそうなfirst-name basisの知り合いにこの封筒を送ってください．」
- chainの参加者名簿．これにより誰が誰に転送するのか分かるし、ループを避けることができる．

さらに、封筒には15枚の”tracer card”が入っている．受け取った人は、tracer cardの一枚に書き込んで、実験者まで送り返してもらう．封筒は次の人に送ってもらう．

Milgramは，start personをKansan州Wichita（最初の実験）とNebraska州Omaha（2番目の実験）から選んだ．最初の実験のtarget personはCambridgeに住んでいて、神学校の学生の妻である．２番目の実験では、target personは、Bostonに勤務し、Massachusetts州Sharonに住むstockbrokerである．さて、本当にKansasから始まったchainがMassachusettsまで届くのだろうか？その結果は意外に早く分かった．封筒を何人かのstarting personに送ってから４日後、神学の講師が通りで最初のtarget personに声をかけ、“アリス，プレゼントだよ”と封筒を手渡した．封筒の中の名簿を見てみると、Kansasの小麦農家から始まって、Kansasの牧師に渡り、Cambridgeの神学の講師に渡り、target personに渡ったことが書かれてあった．何と、starting personとtarget personのリンクは２であった！

このケースは、最も短いchainのひとつであったことが後に分かったが、chainはだいたい２から10の中継者を経由しており、中間値は５であった．この５という数字は大変な

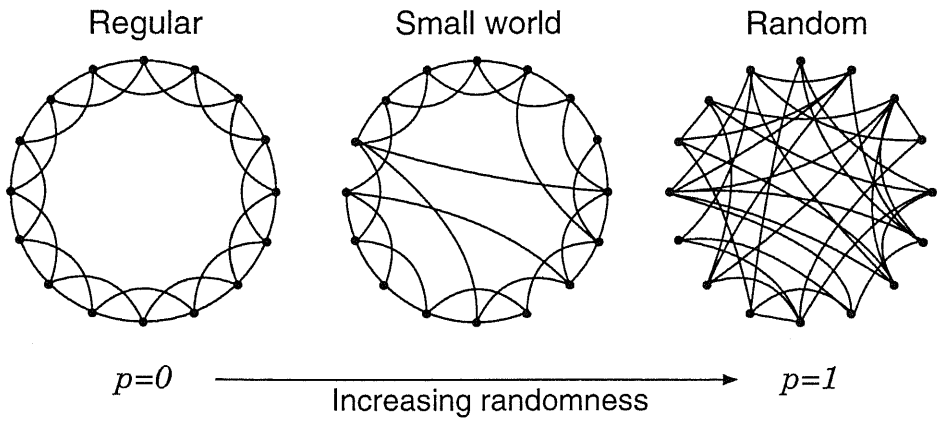


図 10.2: regular ring lattice のランダムなつなぎかえ.

驚きである．米国の人口と比較してこの5人という数字は驚くほど小さく，"six-degrees of separation" として米国では広く知られている [Guare 90]. [1].

長い間，社会心理学的な考察対象であった Small World は，1998 年に Duncan Watts らがグラフの2つの特徴量として定式化を行って以来，コンピュータサイエンスの分野でも注目を集めるようになった．2つの特徴量とは，以下である [Watts 98][Collins 98].

- L (characteristic path length) : すべてのノードの組についてのパス長の平均．パス長とは，最短パスの長さである.
- C (clustering coefficient) : 近傍の cliqueness(派閥度) を表す．ひとつのノードが k 個のノードと隣接しているとき，この k 個のノード間に存在するリンク数を ${}_kC_2$ で割ったものを，すべてのノードについて平均をとったものである．(リンクを友だち関係と考えると，自分の友だち同士が友だちである確率である.)

Watts によると，Small World は $L \geq L_{rand}$ (または $L \approx L_{rand}$) かつ $C \gg C_{rand}$ であるようなグラフとして定義される．ここで，L_{rand}，C_{rand} は，同じノード数，リンク数のランダムグラフにおける L と C である．図 10.2 に，規則的なグラフから，リンクをランダムにつなぎ変えていくと，途中の段階では，規則的でもランダムでもないグラフが現われる．これが Small World である.

[1] "Six-degrees of separation" (邦題：私に近い 6 人の他人) という映画も 1993 年に公開されている.

T. Walshは，さらにsmall-worldlinessという指標μを考案した[Walsh 99].

$$\mu = (C/L) / (C_{rand}/L_{rand}) \tag{10.1}$$

つなぎかえの確率pが0から増えていくと，少数のショートカットがLを急激に減らす．これらのショートカットはCにはほとんど影響をおよぼさない．結果としてμが急激に大きくなり，グラフはSmall Worldの特徴を持つようになる．さらにpが1に近づくと，つなぎかえがLにほとんど影響を及ぼさなくなり，Cとμが低下する．こうして，再びSmall Worldの特徴を失うことになる．表10.1にSmall World構造を持つ様々なグラフの例と各特徴量の値を示す．俳優の人間関係のネットワーク，線虫のニューロンのネットワーク，送電網のネットワークのいずれもμは1より大きく，Small Worldの特徴を備えている．

では，なぜSmall Worldが自然界や人工物に遍在するだろうか．これについては，局所的かつ大域的な情報伝達効率が高い[Marchiori 00]，グラフの連結性の最大化(maximal connectivity)とコストの最小化(minimal cost)のトレードオフである[Mathias 01]などの指摘がされている．ノード間の距離には，物理的な距離（例えば航空路でいえば，New YorkとL.A.の物理的な距離）とグラフ上の距離（飛行機を何回乗り継がなければいけないか）の2つがある．信号や情報，物質などの伝達効率からは，グラフ上の距離は短い方が望ましい．すなわち，Lは短い方が望ましい．一方，リンクを張るコストはノード間の物理的な距離に比例すると考えると，リンクの物理的な長さの平均Wは短い方が望ましい．LとWはトレードオフの関係にあるが，[Mathias 01]では関数

$$E = \lambda L + (1 - \lambda)W$$

（ただし，λは0以上1以下の重みを表すパラメータ）を想定し，これを最小化するようなリンクの張り方を求めると，λが0のときにはregular latticeが，λが1のときはランダムグラフが得られ，その中間ではSmall Worldが出現することを明らかにした．多くのグラフでは，連結性の最大化（Lが小さいこと）とコストの最小化（Wが小さいこと）という相反する両方の要求があるため，Small Worldが遍在すると考えられている．

[Amaral 00]では，Small Worldネットワークを，

表 10.1: Small World 構造を持つ様々なグラフの L,C,μ [Walsh 99]

	L	L_{rand}	C	C_{rand}	μ
Film actor	3.65	2.99	0.79	0.00027	2396
Power grid	18.7	12.4	0.080	0.005	10.61
C. elegans	2.65	2.55	0.28	0.05	4.755

Film actor は，Hollywood の俳優についてのグラフで，2 人の俳優が同じ映画で共演していればリンクが張られる．Power grid はアメリカ西部の送電網についてのグラフで，ノードは発電機，変電所等を表し，リンクは高圧送電線を表す．*C. elegans* は，*Caenorhabditis elegans* という線虫のニューロンのネットワークで，シナプスもしくはギャップ結合でつながれたニューロン間にリンクが張られる．

(a) scale-free なネットワーク（ノードの degree（連結度）を横軸に，確率分布を縦軸に取ると，power law に従う．つまり，極めて degree の大きいノードもいくつか存在するネットワーク），

(b) broad-scale なネットワーク（power law だがカットオフがある．ある程度以上 degree の大きいノードはないネットワーク），

(c) single-scale なネットワーク（ノードの degree の分布が指数的に小さくなるネットワーク）

という 3 種類に分けて議論を行っている．また，Small World は疫病や流行が伝わりやすいこと [Watts 99][Zanette ar]，エラーや攻撃に強いネットワークであること [Albert 00] なども示されている．[大澤 01] では，Web コミュニティを L と C を用いて特徴づける試みを行っている．

10.3 共起グラフの Small World 性

文書から得られた語の共起グラフは，いくつかのクラスタとその間をつなぐリンクから構成されていた．では，語の共起グラフは Small World の特徴を持つのであろうか?

前述の例 (図 10.1) を再び考えてみよう．右側の大きなクラスタには Small World に関連する語が集まっており，左側には “important term” “document” などキーワード抽出に関する語

が集まっている．そして2つのクラスタを“small”と“term”のリンクがつないでいる．また右側のクラスタの中にも，“small world”“topology”などの語の集まり，“node”“path”“length”などの語の集まり，“characteristic path length”“clustering coefficient”“Watts”などの語の集まりが見てとれる．関連する概念の語がリンクで結ばれながら，全体としてもまとまりのある構造になっている．実際，このグラフに対して$L = 3.63$，$C = 0.524$（$L_{rand} = 2.40$，$C_{rand} = 0.0856$）であり，LがL_{rand}と同程度でありながらCがC_{rand}より非常に大きいというSmall Worldの特徴を備えていることが分かる．

では，一般的に論文から生成した語の共起グラフはSmall Worldの特徴を持っているのだろうか? そこで，WWW9[2]の論文57篇およびJAIR[3]の論文166篇を用いて検証した．結果を表10.2，10.3に示す．WWW9の論文は，平均でLは5.60でL_{rand}の1.5倍程度だが，CはC_{rand}と比べて15倍以上大きい．JAIRの論文はWWW9のものと比べて，論文の長さのばらつきが大きいが，やはりLはL_{rand}の1.4倍程度であるのに対してCはC_{rand}の14倍である．したがって，LがL_{rand}と同程度で，CがC_{rand}よりも非常に大きいというSmall Worldの性質を備えていることがわかる．なお，同じ規模のregular latticeに対して，Lは20以上，Cは0.6程度である．なお，著者らの知る限り，C_{rand}と比べCがどの程度大きければSmall Worldといえるのかという定量的な報告はされていない．表10.1では，非常にnの大きいFilm actorsでCはC_{rand}の約3000倍，Power gridでは16倍，*C. elegans*では5.6倍である．WWW9やJAIRの論文では，CはC_{rand}の十数倍であり，グラフの規模が同程度である*C. elegans*がSmall Worldであると認められていることから考えて，十分Small Worldといえると判断した．

論文から得た語の共起グラフがSmall Worldの性質を持っている理由は，次のように説明することができる．論文の一文中で同時に用いる語は関連が強い方が分かりやすい．例えば,「ノード」や「パス」という語は，グラフ構造に関する語であるので同時に用いても分かりやすいが,「パス」と「文書」は関連が弱く，同時に用いると分かりにくい．一方で，なぜ「パス」と「文書」がひとつの論文に出現するのか明らかであることも重要である.「パ

[2]9th International World Wide Web Conference. http://www9.org/.
[3]Journal of Artificial Intelligence Research の93年 (Vol.1) から2001年 (Vol.14) までの論文.

表 10.2: WWW9 の論文 57 篇についての L と C

	L	L_{rand}	C	C_{rand}
Max.	7.67	4.21	0.509	0.0432
Ave.	5.60	3.71	0.355	0.0211
Min.	4.17	3.03	0.196	0.0113

全ての論文に対して，各項目それぞれの最大値，平均，最小値を示している．共起グラフは，$f_0 = 3$, $k_0 = 4.0$ として生成し，最大連結サブグラフ（平均 79%のノードをカバーする）だけに着目した．得られたグラフは，平均で $n = 275$, $k = 5.04$ であった．(k が k_0 と異なるのは，最大連結サブグラフにだけ着目しているためである．)

表 10.3: JAIR の論文 166 篇についての L と C

	L	L_{rand}	C	C_{rand}
Max.	9.22	3.97	0.588	0.0931
Ave.	4.73	3.38	0.326	0.0230
Min.	3.07	2.41	0.149	0.0129

$f_0 = 3$, $k_0 = 4.0$ とした．最大連結サブグラフ（平均 88%のノードをカバーする）だけに着目し，得られたグラフは平均で $n = 196$, $k = 4.81$ であった．

ス」「ノード」などのグラフに関する語と「文書」「キーワード」などの文書に関する語が，「共起グラフ」という語によって結びついていることが明らかであれば，筆者の主張が伝わりやすいだろう．したがって，読者に分かりやすく，しかもまとまりのある論文を推敲しながら書き上げるという作業は，コストの最小化と連結性の最大化の両方を考慮していると考えられる．

10.4 Small World 構造を利用したキーワード抽出

文書から得られた語の共起グラフが Small World であるとすると，いくつかのノードは L を減少させるのに大きく貢献しているはずである．このような語は，互いに関連のうすい語のクラスタ同士をつないでいるのであるから，文書の論旨において重要な意味を担ったキーワードであると考えることができる．本節では，まず L の定義を非連結グラフに拡張した後，ひとつのノードの Small World 構造に対する貢献を示す contribution という指標について述べる．

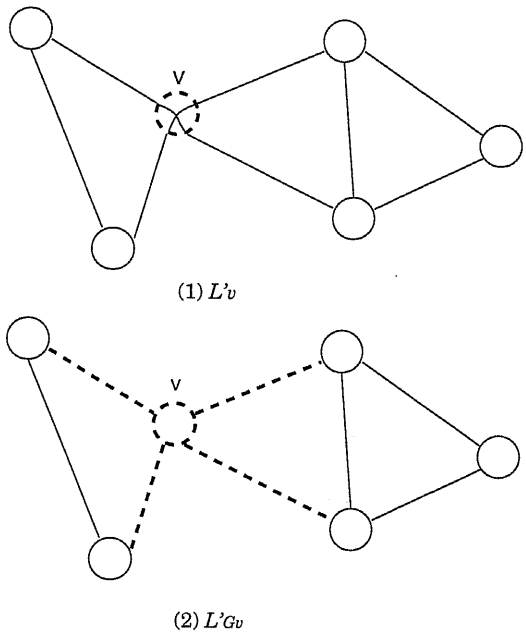


図 10.3: L'_v と L'_{G_v} の例

定義 10.4.1

ノード i，ノード j に対する *extended* path length $d'(i,j)$ を次のように定義する．

$$d'(i,j) = \begin{cases} d(i,j), & \text{if } (i,j) \text{ are connected,} \\ w_{sum}, & \text{otherwise.} \end{cases} \tag{10.2}$$

ただし，$d(i,j)$ は，連結したグラフにおけるノード i とノード j のパス長である．w_{sum} は定数で，すべての連結していないサブグラフの幅の和である．グラフの幅とはグラフ中の2ノード間のパス長の最大値であり，w_{sum} は，サブグラフが新たなリンクにより連結されたときの2ノード間のパス長の上限を与えている．

この定義を用いて，L を自然に拡張することができる．

定義 10.4.2

extended characteristic path length L' は，すべてのノードの組についての extended path length の平均である．

さらに，ひとつのノードの L に対する寄与を計るために，次の定義を行う．

定義 10.4.3

L'_v は，ノード v 以外のすべてのノードの組についての extended path length の平均である．

表 10.4: 論文 [Matsuo 01] の頻出語

Term	Frequency
graph	39
small	37
world	37
term	34
small world	30
node	29
paper	21
length	21
document	19
edge	19

L'_{G_v} は，ノード v を取り除いたグラフにおける extended characteristic path length である．

これを図 10.3 に簡単に図示する．L'_v の計算ではノード v はグラフに接続されているが平均には含めない．L'_{G_v} の計算ではノード v と v を含むリンクはグラフから除外される．この差をとることで，ノード v が L の減少にどれくらい影響を与えているかを求めることができる．

定義 10.4.4

ノード v の *contribution* CB_v は，次のように定義される．

$$CB_v = L'_{G_v} - L'_v \tag{10.3}$$

すなわち，contribution はノード v の Small World 構造への貢献を計る指標である．この値が大きいノードは，離れたクラスタをつなぐ重要なノードであると考えられる．

10.5 評価

10.5.1 具体例と評価

前述の論文 ([Matsuo 01]) に対して，頻出語の上位と contribution の高い語の上位を表 10.4, 10.5 に示す．頻出語は，“graph”, “small world” など，論文中に多く出現する一般的な

表 10.5: 論文 [Matsuo 01] における CB_v が上位の語

Term	CB_v	Frequency
small	3.05	37
term	2.80	34
important term	1.93	7
contribution	1.64	6
node	1.00	29
make	0.82	6
cluster	0.57	15
graph	0.54	39
coefficient	0.52	8
average	0.50	8

表 10.6: 論文 [Matsuo 01] における CB_e が上位のリンク

Pair	CB_e
small – term	3.07
important term – contribution	1.93
make – contribution	1.22
node – average	0.98
structure – make	0.82
cluster – short	0.55
graph – connect	0.53
coefficient – sentence	0.52
average – pair	0.49
contribution – measure	0.48

語が得られているのに対し，contribution の高い語は “important word” や “contribution”, “cluster” など，出現回数は多くないものの論文中で重要な役割を果たす語が得られている．

一方，あるリンクが存在するときとしないときでの L の変化を同様に比較することで，Small World 構造へのリンクの貢献を計ることができる．これを表 10.6 に示す．しかし，語と語の関係は様々なものがあり，どのように利用すればよいかは難しい．

次に，contribution の上位語がキーワードとしてどの程度優れているか評価実験を行っ

た．実験は，人工知能の分野の7著者20論文に対して行い，tf，tfidf[4]，KeyGraph[5]と比較した．各手法でキーワード15個を出力し，各手法から得られたキーワードの上位語を混ぜてシャッフルし，著者に「論文を構成する重要な概念を表すと思う語にチェックをして下さい」という質問を行った．各手法による出力語中でキーワードであると判定された割合がprecisionである．さらに，「提示した全ての語（提示した以外の語でも覚えているものがあれば含めてよい）のうち，論文中で不可欠な概念を表す語5つ以上を選びA，B，C，D，Eと印をつけ，それと同義の語にも同じ印をつけてください」という指示を行った．5つ（以上）の概念のうち各手法で提示した語にいくつ含まれているかでcoverageを測定した．

表 10.7: Precision と Coverage

	tf	KeyGraph	**本手法**	tfidf	**本手法・idf**
precision	0.53	0.42	**0.47**	0.55	**0.73**
coverage	0.48	0.44	**0.52**	0.61	**0.60**
frequency index	28.6	17.3	**13.8**	18.1	**11.1**

結果を表10.7に示す．precision，coverageの他にfrequency indexという指標を表示しているが，これは各手法が提示した語の出現頻度の平均を示しており，これが高いほど「当たり前」の語を出力しているということになる．本手法は，precision，coverageともに50%程度と，tfと同程度の性能しか得られておらず，tfidfより悪い結果となっている．しかしながら，frequency indexは低く，出現頻度が低いにも関わらず重要な語を，tfに近い割合で取り出しているという点では評価できる．

本手法の性能がそれほど良くない理由として，出力の上位にキーワードに混ざって，表10.5における“make”のような非常に一般的な語も含まれる場合が多いことが挙げられる．一般的でありながらstop wordには指定されていない“make”のような語を取り除くには，他の文書によく出現するかどうかを調べればよい．そこで，一般語をふるいおとす目的で，

[4]コーパスはJAIR(Journal of Artificial Intelligence Research)の93年(Vol.1)から2001年(Vol.14)までの論文全文166篇とした．また，語 v に対するidfの重みづけは $\log(N/df(v))+1$ とした．ただし N は全文書数，$df(v)$ は語 v が出現する文書数である．

[5]本手法と同様に構造的な特徴からキーワードを抽出するため，コーパスは不要である．

表 10.8: 論文 [Matsuo 01] の $CB_v \times idf$ 上位語

Term	CB_v	Frequency
small world	2.58	37
shortest path	1.76	34
short cuts	0.94	7
contractor	0.90	6
rare	0.80	29
co-occurrence	0.73	6
sentence	0.63	15
path length	0.50	39
important term	0.48	8
document	0.15	8

語 v の重みを

$$CB_v \times idf(v)$$

とする工夫を行った．この結果を本手法・idf として表中に示している．tfidf と比べても，よい性能が得られていることが分かる．また，frequency index も，本手法単独のときよりもさらに下がっており，一般語を取り除くという目的が果たせていることが見て取れる．実際に，論文 [Matsuo 01] に対して本手法・idf で得たキーワードを表 10.8 に示す．表中のほぼ全ての語が，論文のキーワードとして適切な語となっている．

結論として，本手法は頻度が低いにも関わらず重要な語を抽出することができる．しかしながら，キーワード抽出アルゴリズムとしての精度をあげるために idf と組み合わせることにより，非常によいパフォーマンスが得られる．

10.5.2 本手法の計算量

本手法は，最短路の探索を繰り返し行うので計算コストは比較的高い．ある語 v について式 (10.3) に示される CB_v を求めるには，全てのノードペアの間の最短路の計算が必要である．最短路の計算にはダイクストラ法やワーシャル-フロイド法（例えば [伊里 76]）などのアルゴリズムを用いることができるが，全ノードペアの最短路の計算量はノード数 n に対して $O(n^3)$ の計算量となる．本手法ではこの部分の計算量が最もオーダが高く，これが

本手法の計算量となる．

本章での実験ではnは200〜300程度であり，CPU Pentium II 333MHzの計算機に実装したLinux上のC言語のプログラムで30秒以内でキーワードが得られる．しかし，計算量をいかに減らしながら同様のキーワードを得るかも今後の課題のひとつであろう．

10.6 関連研究との比較・考察

本手法では，文書から共起グラフを生成し，その構造的な重要性に着目してキーワードを抽出する．しかし，文書からどのように共起グラフを生成するか，また，どのような構造的な重要性に着目するかにはさまざまなバリエーションが考えられる．

まず，文書から共起グラフを生成する際，ノードは出現回数が3以上の単語を全て用いている．この値を大きくすれば，頻度による足切りを行うことになり，より頻度を重視した結果となる．また，この値をこれ以上小さくすると，ノード数が極端に多くなり，その文書に偶然出現したような必然性のない語まで出てきてしまう．このような理由から，ここでは出現回数3回以上の語をノードとしている．

次に，ノード間のリンクを張る際，本手法ではJaccard係数により単語間の共起関係を計っている．Jaccard係数は語同士がどのくらい類似しているかを計る指標であるが，他にも共起頻度を用いる方法や相互情報量を用いる方法などがある[徳永 99]．これらの指標についても比較を行ったが，共起頻度を用いた場合には出現頻度の高い語からリンクが多く張られ，結果的に出現頻度の高い語が抽出されやすくなる．一方，相互情報量は，2つの語が独立に生起する場合の確率と共起する確率を比較するので，出現回数が小さく，しかも同時に出現する語のペアに対して大きな値が出やすくなる．その結果，出現頻度の小さい語に偏ってリンクが張られ，キーワードとして選ばれやすくなる．以上の知見を踏まえて，本手法ではJaccard係数を用いた．

また，本手法ではグラフの構造における重要性を，Small World構造に着目してcontributionという指標で計っているが，グラフの中心性を計る指標としてはFreemanのcentralityの概念[Freeman 79]が有名である．これは，人間関係などのネットワークにおける情報伝

達やコントロールにおいて，あるノードがどのくらい中心的かを計る指標である．Freeman は以下の３つの指標を提示している．

- *degree*: ひとつのノードがいくつのノードとリンクしているかである．つまりノードの次数 (degree) によって中心性を計る．
- *betweenness*: あるノードが他のノードのペアの最短経路にどのくらいの割合で含まれているかを計る．つまり，情報を伝える際に，そのノードを通らなければいけない度合を表す．
- *closeness*: あるノードから他のノードへの最短パスの合計である．グラフ中のどのノードにも近いノードが中心的であるとするものである．

これらの指標を用いた語の重み付けの検討も行ったが，得られる語はグラフの中心部に偏ってしまう．例えば，論文 [Matsuo 01] に対して適用すると，“path length”, “characteristic path”, “characteristic path length”, “path length averaged” など，お互いに近い位置にあるフレーズが大量に出てきてしまう．文書中では，中心部の語だけではなく端の方の語も重要であるし，中心部と端の語の関係を表す語も重要であるが，これらの指標はもともと中心性の指標であるため，キーワード抽出のための特徴量としては適当でない．

本手法の欠点として，文書からグラフ構造を正確に取り出すために，文書の長さがある程度必要である点が挙げられる．論文の抄録程度の長さでは，適切なグラフを抽出することは難しいが，WordNet[Miller 95] などの語義のグラフ構造や，文書集合全体の語のグラフ構造を利用することにより解決できる可能性もあるだろう．

本章では，文書から抽出した語の共起グラフが Small World 構造であることを示し，構造的に重要な役割を担う語をキーワードとして提示する手法を提案した．頻度による語の重み付けの背景にある「何度も繰り返し言及される概念は重要な概念である」という仮定は，簡単でありながら非常に強力である．しかし，本手法では「概念のネットワーク上で重要な位置を占める概念は重要な概念である」というもう一つの仮定を提案している．すなわち，文書における語の網羅性を，ネットワークにおける影響力の強さと捉え，特定性の指標としての idf と組み合わせている．

第11章　仮説推論を利用した複数文書要約

前章では，文書内の語の共起グラフから，そのSmall World構造を利用して，語の重要度を判断する手法を提案した．では，この共起グラフを文の重要度の決定，つまり要約に用いることはできないだろうか? 本章では，多くの語と語の関係をカバーするような文は重要だと考え，複数文書から要約文を生成することを試みる．なお，その計算過程において，仮説推論の高速解法を利用している．

11.1　文書の要約技術

テキスト自動要約研究の歴史は古いが，その多くのものは，テキスト中の文（あるいは，形式段落）を1つの単位とし，それらに重要度を付与して重要なものを選択し，寄せ集めて要約を作成する [奥村 99]．一般に，重要文抽出による要約と呼ばれる．重要度評価の際に用いられているテキスト中の情報は，次のような7つに分けられる [Paice 90]．

- テキスト中のキーワードの出現頻度を利用する，
- テキスト中あるいは段落中での位置情報を利用する，
- テキストのタイトル等の情報を利用する，
- テキスト中の文間の関係を解析したテキスト構造を利用する，
- テキスト中の手がかり表現を利用する，
- テキスト中の文あるいは単語間のつながりの情報を利用する，
- テキスト中の文間の類似性の情報を利用する．

本章では，テキスト中の単語間のつながりの情報を利用するが，このような手法には，例えば以下のようなものがある [奥村 99].

[Skorokhod'ko 72] では，文をノード，文間の関係をリンクとするグラフでテキストを表現し，多くの文と関係がある文が重要であるという考えに基づき，重要文を抽出している．文中の単語が同一概念を参照しているような文間にリンクがあるとしている．[Hoey 91] では，語彙的結束性[1]の情報を利用し，文間で単語によるつながりが多いほど，文間のつながりが強いと考え，他の文とのつながりの強さに基づき，要約を作成する手法を示している．Mani[Mani 97] らは，テキスト中の単語などがノードであり，その間の隣接性，構文的関係，共参照関係，語彙的類似性などの関係をアークで表現したグラフでテキストを表現し，このグラフ中での活性値の伝播により，高い活性値を得た単語，句，文を重要と見なす重要文抽出手法を示している．

語の共起グラフは，Mani らのものに近い．ただし，構文的関係や語彙的な類似性などは考慮していない．

11.2 語の共起グラフと文書の要約

複数文書の要約では，内容が重複するのを避けることが重要になる．冗長な箇所を削除しても，複数テキストの要約文書としてはまだ十分であるとは言えない．複数のテキストを比較し要点をまとめることが重要であり，テキスト間の共通点とともに，相違点も明らかにすることが必要である [奥村 99][奥村 00].

著者は，本研究室の岡崎らと共に，次のようなシステムを構築した．まず，複数文書に対して，前章と同じように語の共起グラフを構築する．図 11.1、11.2 は，要約の対象となる毎日新聞の記事 9 件（“ハイブリッド，カー，発売， 開発” に対しての検索結果）の一部である．この複数の記事から生成した語の共起関係を示したものが図 11.3 である．各ノードは語を表し，リンクは語の共起の頻度が 2 回以上であることを表している．また，文書ごとのリンクを異なる濃さで表示している．

[1]語彙的結束性 (lexical cohesion) とは，同一の語彙項目を 2 回選択すること，または密接な関係にある 2 つの語彙項目を選択することで，文間の意味的なつながりが表されることである [Haliday 76].

さて，このグラフ上でどのような特徴を持つ文が重要であろうか．各文は，その文で用いられている語と語の関係を明らかにする．つまり，このグラフ上ではリンクをカバーすることに相当する．したがって，多くの語と語の関係を明らかにするような文，つまり，多くのリンクをカバーするような文を重要文として選べばよさそうである．さらに，ある文を選んだときに，その文と同じようなリンクをカバーする文を選んでも，冗長な情報になってしまう．選ぶべき文は組み合わせ的に決まるものであり，次のような最適化問題に帰着することができる．

$$Min. \quad f = \sum_{i \in K} cost_i x_i$$

ただし，K はリンクの集合，$cost_i$ は，リンク i が要約に含まれないときのペナルティコスト，x_i はリンク i が要約に含まれなければ 1，そうでなければ 0 である 0-1 変数である．さらに，次のような制約に従う．s_j を，文 j を選択するときには 1，選択しないときには 0 となる 0-1 変数とする．$s_j = 1$ のときには，文 j に含まれるリンク i に関して $x_i = 0$ に，そうでなければ $x_i = 1$ になる．また，選択する文の語数に関しては，次のような制限がある．

$$\sum s_j l_j \leq L \tag{11.1}$$

l_j は，文 j の語数，L は要約文の語数の上限を表す．

このように，複数文書要約問題は上述の仮定のもとで，文を要約文に含めるか含めないかという組み合わせ最適化問題で表すことができる．これは，式 (11.1) の制約以外は次のように仮説推論問題で表すことができる．

リンクの総数を k，文の総数を m とする．まず，満たすべきゴールは，「リンク 1 からリンク k まで，すべてのリンクが考慮されている」ことをあらわす G である．

$$G \leftarrow x_1, x_2, \ldots, x_k.$$

文 j を選択するという仮説は h_{s_j} で表し，コストは 0 とする．例えば，文 1 でリンク 13 番，リンク 220 番，リンク 223 番がカバーされるとすると，

$$x_{13} \leftarrow h_{s_1}, \quad x_{220} \leftarrow h_{s_1}, \quad x_{223} \leftarrow h_{s_1}$$

と記述することができる．一方，選択されないリンク i に対しては，

$$x_i \leftarrow h_{emp_i} \quad (i = 1 \ldots k)$$

という仮説 h_{emp_i} を便宜的に用意しておく．そして，このコストを1（もしくは適当な値）とする．

以上で，「カバーされないリンクの数が最も小さくなるように文を選択する」という問題を記述できたことになる．しかし，このままでは，全ての文を選択することでコストが最小値0となってしまう．要約では，文字数の制限が本質的であるので，この制限を入れなければならない．

文字数の制限を入れるには，文を選択する仮説 h_{s_j} にコストを付与すればよい．しかし，「リンクがカバーされないこと」と「文字数」とは異なる性質のコストであり，単純に足し合わせることは適当でない．むしろ，決められた文字数の中で，最もリンクをカバーするような文を選ぶというように，文字数は制約として考えた方が適当である．6章で示した2種の置き換え法の協調による高速仮説推論法では，変数間の制約をある程度自由に記述することができる．したがって，式11.1に相当する制約，例えば，

$$39h_{s_1} + 77h_{s_2} + 34h_{s_3} + \ldots \leq 500$$

（文1が39文字，文2が77文字，文3が34文字,...，全体の文字数が500文字以内の場合）を別に記述しておく．

このように，各複数文書要約課題に対して，知識ベースを生成し，G を証明するような仮説の組を求めることで，要約となる文の集合を求めることができる．図11.4は，前述の例に対する要約結果を示している．この例は，比較的うまくいっている例であるが，説明的な文がうまく抜き出されており，文間の重複もあまりない．なお，新聞記事では，リード文を取るという単純な手法が非常によい性能を示すが，本手法でもリード文が抽出されるケースは多い．したがって，本手法の有効性がある程度示されていると考えている．

しかしながら，複数文書の要約には，より細かい（もしくはアドホックな）手法を多く組み込まなければならない．例えば，「今日」という語があれば，その新聞の日付で置き換

えたり，指示語を含む文はなるべく取らないようにするなどである．また，重要な「文」を抽出するのではなく，重要な句や語の単位で文を抽出しつなぎ合わせるなどの処理も，読みやすい要約生成には必要になる．このように要約としての精度を上げるための工夫は今後の課題である．

なお，この研究は，自動要約のコンペティションである第3回NTCIRワークショップ[2]のテキスト自動要約タスク (2001年9月～2002年10月) に，本研究室の岡崎 直観君らと共にチームで参加しているものであり，その評価は2002年10月の最終報告において公表される予定となっている．

[2]国立情報学研究所により開催．http://research.nii.ac.jp/ntcir/workshop/work-ja.html

トヨタ自動車、米ゼネラル・モーターズと低公害車で提携

トヨタ自動車と米ゼネラル・モーターズ（GM）は１９日、次世代低公害車の本命として期待されている燃料電池電気自動車（ＦＣＥＶ）など、環境対応型の先進技術車を共同開発することで合意したと日米で同時発表した。環境対応は２１世紀に向けた自動車産業最大の課題で、各国の有力メーカーが膨大な資金を投じて技術開発にしのぎを削っている。販売台数で世界１位のGMと、３位のトヨタが手を組んだことで、競争は一段と激化し、国際的な合従連衡も加速することになりそうだ。　共同開発するのは、燃料の水素と空気中の酸素を化学反応させて発電し、モーターで走るＦＣＥＶのほか、ガソリンエンジンと電気モーターを併用するハイブリッド自動車（ＨＶ）、蓄電池でモーターを動かす電気自動車（ＥＶ）などをめぐる幅広い技術。

.....(中略).....

トヨタは、他社に先駆けて１９９７年にＨＶ「プリウス」を発売。ＦＣＥＶでも既に二酸化炭素や窒素酸化物を一切出さない試作車を発表し世界のトップレベルの技術を持っている。

燃費効率良い乗用車　「最高」求め競争激化

世界の自動車メーカーが、燃料消費量を極めて低い水準に抑えた乗用車の開発にしのぎを削っている。本田技研工業が７月６日、ガソリン１リットルで３５キロメートル走る量産車では世界最高の燃費効率のハイブリッド車（ガソリンエンジンと電気モーターの併用車）を今秋発売すると発表したのに続き、独フォルクスワーゲン（VW）は同２３日、軽油１リットルで３３・４５キロメートル走るディーゼルエンジン車を売り出した。各国の環境規制が強化される中、環境にやさしい乗用車の主導権を狙う争いは激しさを増している。　燃費競争で先行したのはトヨタ自動車。１９９７年１２月に世界初の量産ハイブリッド車「プリウス」を発売した。ガソリン１リットルで２８キロメートルという従来のガソリン車の倍の燃費効率と画期的な性能で注目を集めたが、これをしのぐ「３リッターカー」が相次いで誕生することになる。

.....(中略).....

ただ、トヨタは、来年からプリウスの欧米向け輸出を始めるほか、人気のＲＶ（レジャー用多目的車）タイプも追加し、国内販売で月５０００台を目指す。現在は技術開発競争の側面が強いが、各社が競って市場投入すれば、需要のすそ野が広がり、量産化による生産コストが下がり、価格引き下げの可能性が広がる。

図 11.1: “ハイブリッド カー 発売 開始” に関する複数の文書 (計９文書)

低燃費は世界最高水準　インサイトに試乗

ホンダのハイブリッドカー「インサイト」に乗った。トヨタのプリウスですっかりイメージが定着したハイブリッドカーだが、ホンダのそれは少し趣が違う。どんなクルマなのか、プリウスとどこが違うのか検証してみよう。併せてホンダの環境問題への取り組みについてもリポートする。　ハイブリッドカーとは、ガソリンエンジンと電気（モーター）の併用で走るクルマで、一昨年末、トヨタが世界に先駆けて量産市販車「プリウス」を発売した。低燃費と排ガスのクリーンさが売り物で、日本国内だけの発売だったが世界的な注目を集めた。　「インサイト」はプリウスに続く世界で２番目の量産ハイブリッドカーである。１１月１日の発売だが、日本国内だけではなく北米を中心に海外でも順次発売する。プリウスも近々にアメリカで発売する予定だし、来春にはニッサンからも新型車がお目見えする。世界のハイブリッドカー市場は日本車がリードしたかっこうである。　同じハイブリッドカーでもインサイトとプリウスでは発想がだいぶ違う。例えばプリウスの動力装置はモーターを主にして、ガソリンエンジンを補助にしているのに対してインサイトはガソリンエンジンが主でモーターが補助である。つまり通常走行ではプリウスが電気、インサイトはガソリンを使って走るのだ。　「どちらを主にするかが問題なのではない。要はどちらを主にするほうが環境に優しいか、燃費の節約になるかである。その尺度で見るとうちのインサイトの燃費はどこにも負けない」とホンダのインサイト開発者は説明する。　配布された資料によるとインサイトの燃費は１０・１５モードで３５キロ／リットル（５速ＭＴ車）であり、量産ガソリン車では現在のところ世界最高の低燃費だという。確かにこの数字は驚異的である。今や世界の低燃費競争は３リッターカー（３リットルのガソリンで１００キロ走る）になっているが、インサイトは見事にそれをクリアしていることになる。となればガソリンを主にするか、電気を主にするかは問題にはならない。

.....(中略).....

この環境問題への取り組みをさらに明確にし、「２００５年」という年を強調した。つまり０５年までに平均燃費を１９９５年比２５％向上＝２０１０年の新燃費基準を５年前倒し達成、０５年までにＨＣ、ＮＯｘの総排出量を１９９５年比７５％削減などの具体的な目標である。　走りで世界を制覇したホンダが環境でも世界をリードできるか、じっくり見極めたいところだ。

図 11.2: “ハイブリッド カー 発売 開始” に関する複数の文書 (計 9 文書)

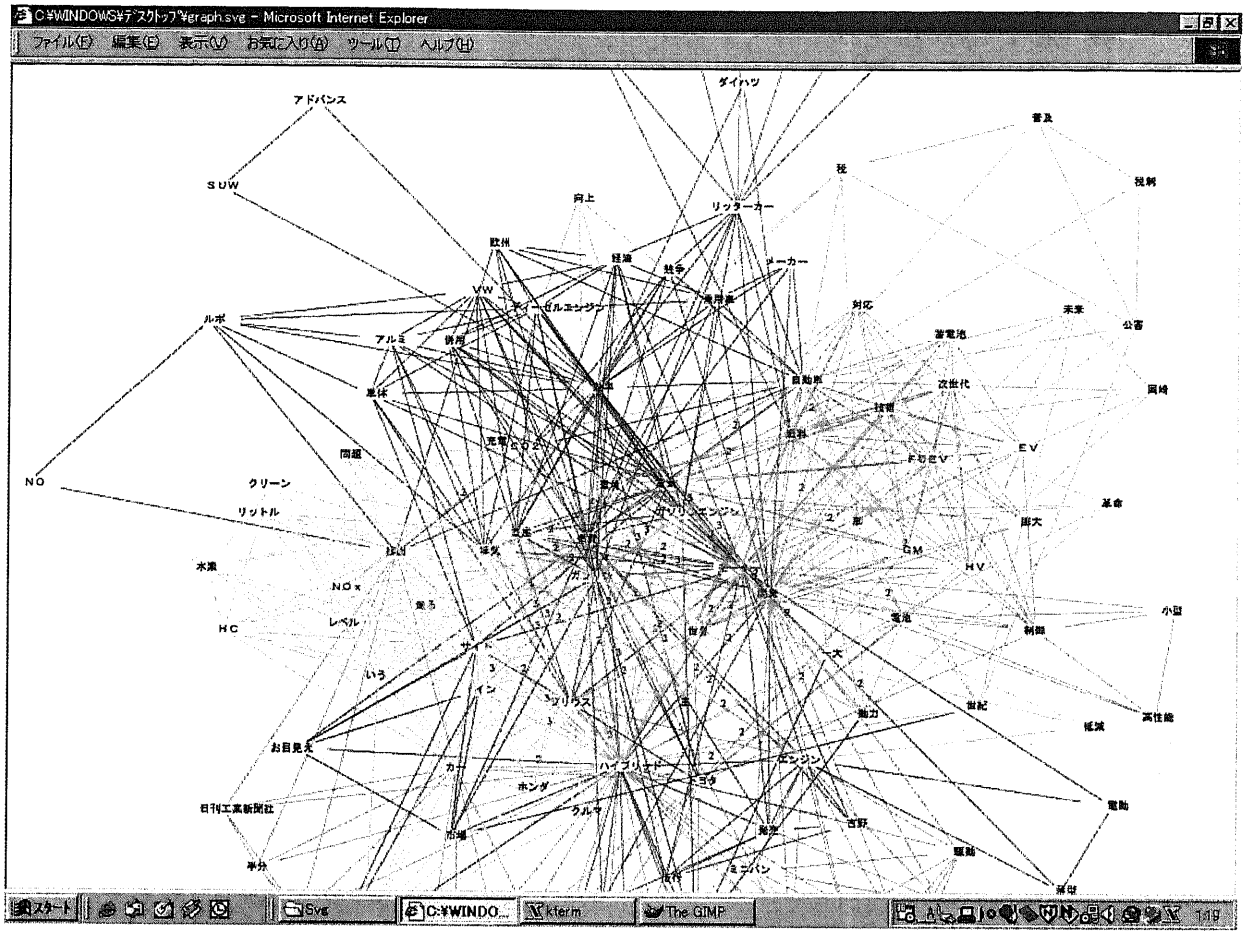


図 11.3: 複数文書の語の共起グラフ

トヨタ自動車と米ゼネラル・モーターズ（ＧＭ）は１９日、次世代低公害車の本命として期待されている燃料電池電気自動車（ＦＣＥＶ）など、環境対応型の先進技術車を共同開発することで合意したと日米で同時発表した。ハイブリッドカーとは、ガソリンエンジンと電気（モーター）の併用で走るクルマで、一昨年末、トヨタが世界に先駆けて量産市販車「プリウス」を発売した。…「インサイト」はプリウスに続く世界で２番目の量産ハイブリッドカーである。…同じハイブリッドカーでもインサイトとプリウスでは発想がだいぶ違う。…また排ガスのクリーンさも環境カーには欠かせない条件だが、インサイトは炭化水素（ＨＣ）、一酸化炭素（ＣＯ）、窒素酸化物（ＮＯ二）とも２０００年規制値を５０％以上下回るレベルだという。本田技研工業が７月６日、ガソリン１リットルで３５キロメートル走る量産車では世界最高の燃費効率のハイブリッド車（ガソリンエンジンと電気モーターの併用車）を今秋発売すると発表したのに続き、独フォルクスワーゲン（ＶＷ）は同２３日、軽油１リットルで３３・４５キロメートル走るディーゼルエンジン車を売り出した。

図 11.4: 500 文字の制限内での要約例 (487 文字)

第III部

結論

第12章　おわりに

本研究では、知識処理に関わる2つの問題点の解決を試みている.

- 大量の知識をどのように獲得するか.
- 大量の知識を用いてどのように高速に推論を行うか.

これに対するそれぞれのアプローチが、文書からの主題抽出と仮説推論の高速化である.

しかしながら、このような枠組を考える際には「知識」とは何かをよく考えなければならないだろう. 何をキーワードと思うかが人の事前知識や文脈、目的によって異なってくるように、知識もそれを用いる文脈や目的に依存する.

仮説推論は，人間の知能のある一面を捉えているだろう. しかし，人間の知能との間には，依然として大きなへだたりがある. AIの有名な書物「ゲーデル，エッシャー，バッハ」[Hofstadter 79]には，次のようなくだりがある.

> ... ここで起った出来事について，きわめて重要な事柄がある. それは人間と機械のひとつの違いを示している. MIUシステム[1]の定理を次から次へと生成するコンピュータプログラムを書くことは十分に可能な—実は，ごくやさしい—ことであろう. また，Uが生成されたときにだけ停止するように，そのプログラムを細工することもできる. そしてわれわれは，そのようなプログラムはけっして停止しないことを知っている. それは驚くべきことではない. しかも，もし友人にUを生成してみてくれと頼んだら，どうなるだろうか? その友人がしばらくして戻ってきて，最初の文字Mは消せないから，Uを作ることなど野性のガチョウを追いかけるようなものだとこぼしても，別に驚きはしないだろう.

[1]本文中で導入される公理と生成規則から成るある形式システムである.

> 非常に賢い人でなくても，自分がしていることを何かしら観察しないわけにはいかないし，その観察から仕事に対するよい洞察が生れる．これこそ，すでに述べたように，コンピュータのプログラムには欠けているものである．(p.53)

また，同著ではこのようにも述べられている．

> ... ところで，内的表現を選ぶということ自体がひとつの型の問題—そして最も扱いにくいもの—なのであるから，問題還元の技法を逆にしてその上に重ねてみようと考えるかも知れない! そうするためには，抽象的空間のおびただしい変種を表現する方法がなければならないだろうが，これはずばぬけて複雑な企てである．(中略) いずれにしても，人工知能に痛切に欠けているのは，「一歩退いて」何が進行しているのかを見わたすことができ，この展望に立って目前の仕事に対するとり組み方を修正できるようなプログラムなのである．(p.602)

これが書かれたのは20年以上前であるが，これは今でも，人工知能研究の目指すべき大きな空白領域であろう．

この問題に対するアプローチのひとつとして，人の気づきを支援すること，データのどこに着目すればいいかという手がかりをつかむことが考えられる．それは，人間にとっても重要であろうし，コンピュータにより知的な処理をさせるという点からも非常に重要なことであろう．まだ，研究は始まったばかりだが，なぜ人間が森羅万象のデータの洪水の中からあることに「気づく」ことができるのか，興味を持って研究を続けたいと考えている．

謝辞

本研究に進めるにあたり，多くの方から御指導・御鞭撻を賜わりました．

大変御多忙の身でありながら熱心に御指導して下さいました，石塚 満教授に深く感謝致します．石塚先生には，卒業研究，博士過程と合わせて4年間もの長きにわたりお世話になりました．研究者としてまだまだ未熟ものですが，今後とも引続き御指導の程，宜しくお願い致します．

また，修士過程では人工知能に興味を持っていた私に電力系統における人工知能の利用という大変面白いテーマを与えて頂いた横山 明彦教授に深く感謝致します．ありがとうございました．

筑波大学 大澤 幸生助教授には，さまざまな面から多くの知的な刺激を頂きました．また，JST 協調と制御領域「自然現象・社会動向の予兆発見と利用」のリサーチスタッフとして研究の機会を与えて頂きました．ありがとうございました．また，JST で研究を共にした松村 真宏氏，吉川 史氏の両氏には，日頃からさまざまな面で大変お世話になりました．ありがとうございました．

I deeply appreciate kind support and advice from Dr. Helmut Prendinger for all these three years. Especially, my English papers are greatly improved by his efforts.

また，学生生活を楽しく過ごさせて頂いた冨倉 由樹央氏はじめ東大クラブの多くの友人と，研究活動を一緒に頑張ってきた友部 博教君，岡崎 直観君，福田 隼人君はじめ石塚研松尾ぐみの皆様に感謝いたします．

最後に，長い学生生活を辛抱づよく支えて頂いた家族に深く感謝いたします．

参考文献

[阿部 92] 阿部, 石塚 : 推論パスネットワーク上での類推による高速仮説推論システム, 人工知能学会誌, Vol. 7, No. 1, pp. 77–86 (1992).

[阿部 98] 阿部明典 : 欠如節を生成する推論法, 電子情報通信学会論文誌, Vol. J81-D-II, No. 6, pp. 1285–1292 (1998).

[Adamic 99] Adamic, L. A.: The Small World Web, in *Proc. ECDL'99*, pp. 443–452 (1999).

[相澤 00] 相澤彰子 : 語と文書の共起に基づく特徴度の数量的表現について, 情報処理学会論文誌, Vol. 41, No. 12, pp. 3332–3343 (2000).

[Albert 00] Albert, R., Jeong, H., and Barabási, A.: Error and attack tolerance of complex networks, *Nature*, Vol. 406, pp. 378–382 (2000).

[A.M.Abdelbar 98] A.M.Abdelbar, : An algorithm for finding MAPs for belief networks through cost-based abduction, *Artificial Intelligence*, Vol. 104, pp. 331–338 (1998).

[Amaral 00] Amaral, L. A. N., Scala, A., Barthélémy, M., and Stanley, H. E.: Classes of small-world networks, *Proceedings of the National Academy of Sciences*, Vol. 97, No. 21 (2000).

[Balas 80] Balas, E. and Martin, C.: Pivot and Complement — A Heuristic for 0-1 Programming, *Management Science*, Vol. 20, pp. 86–96 (1980).

[Bertsekas 89] Bertsekas, D. P. and Tsitsiklis, J. N.: *Parallel and Distributed Computation*, Prentice-Hall (1989).

[Brin 98] Brin, S. and Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine, in *Proc. 7th WWW Conf.* (1998).

[Broder 00] Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J.: Graph structure in the web, in *Proc. 9th WWW Conf.* (2000).

[Charniak 90] Charniak, E. and Shimony, S. E.: Probabilistic Semantics for Cost Based Abduction, in *Proceedings 7th National Conference on Artificial Intelligence (AAAI-90)*, pp. 106–119 (1990).

[Charniak 94] Charniak, E. and Shimony, S. E.: Cost-based abduction and MAP explanation, *Artificial Intelligence*, Vol. 66, pp. 345–374 (1994).

[Church 90] Church, K. W. and Hanks, P.: Word association norms, mutual information, and lexicography, *Computational Linguistics*, Vol. 16, No. 1, pp. 22–29 (1990).

[Collins 98] Collins, J. J. and Chow, C. C.: It's a small world, *Nature*, Vol. 393, pp. 409–410 (1998).

[Craven 98] Craven, M., Dipasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., and Slattery, S.: Learning to Extract Symbolic Knowledge from the World Wide Web, in *Proc. AAAI-98* (1998).

[Dagan 93] Dagan, I., Marcus, S., and Markovitch, S.: Contextual word similarity and estimation from sparse data, in *Proc. 31st Annual Meeting of the Assosiation for Computational Linguistics*, pp. 31–37 (1993).

[Dagan 99] Dagan, I., Lee, L., and Pereira, F.: Similarity-Based Models of Word Cooccurrence Probabilities, *Machine Learning*, Vol. 34, No. 1–3, pp. 43–69 (1999).

[Davis 60] Davis, M. and Putnam, H.: A Computing Procedure for Quantification Theory, *Journal of the Association for Computing Machinary*, Vol. 7, No. 3, pp. 201–215 (1960).

[堂前 94] 堂前, 石塚：仮説推論高速化のための知識ベースリフォメーション, 人工知能学会誌, Vol. 9, No. 4, pp. 595–603 (1994).

[D.Poole 93] D.Poole, : Probabilistic Horn abduction and Bayesian networks, *Artificial Intelligence*, Vol. 64, pp. 81–129 (1993).

[D.Poole 98] D.Poole, , A.Mackworth, , and R.Goebel, : *Computational Intelligence – a logical approach –*, Oxford University Press (1998).

[Dunning 93] Dunning, T.: Accurate Methods for the Statistics of Surprise and Coincidence, *Computational Linguistics*, Vol. 19, No. 1, pp. 61–74 (1993).

[Edmundson 69] Edmundson, H.: New Methods in Automatic Abstracting, *Journal of ACM*, Vol. 16, No. 2, pp. 264–285 (1969).

[Eiter 95] Eiter, T. and Gottlob, G.: The complexity of logic-based abduction, *Journal of the Association for Computing Machinary*, Vol. 42, No. 1–2, pp. 3–42 (1995).

[Etzioni 96] Etzioni, O.: The world wide web: Quagmire or gold mine, *Communications of the ACM*, Vol. 39, No. 11, pp. 65–68 (1996).

[Even-Zohar 99] Even-Zohar, Y., Roth, D., and Zelenko, D.: Word Prediction and Clustering, in *Bar-Ilan Symposium on the foundations of artificial intelligence* (1999).

[Finger 87] Finger, J.: Exploiting constraints in design synthesis, Technical Report TR STAN-CS-88-1024, Stanford University (1987).

[Flake 00] Flake, G. W., Lawrence, S., and Giles, C. L.: Efficient Identification of Web Communities, in *Proc. ACM SIGKDD-2000*, pp. 150–160 (2000).

[Franco 83] Franco, J. and Paull, M.: Probabilistic Analysis of the Davis Putnam Procedure for Solving the Satisfiability Problem, *Discrete Aplied Mathematics*, Vol. 5, pp. 77–87 (1983).

[Freeman 79] Freeman, L. C.: Centrality in social networks: Conceptual clarification, *Social Networks*, Vol. 1, pp. 215–239 (1979).

[福田 02a] 福田, 松尾, 石塚：ユーザの閲覧履歴を考慮したキーワード抽出によるブラウジング支援, 電子情報通信学会 全国大会 (2002).

[福田 02b] 福田, 松尾, 石塚：ユーザ個人の閲覧履歴からのキーワード抽出によるブラウジング支援, 電子情報通信学会 言語理解とコミュニケーション研究会 (2002).

[Fürnkranz 98] Fürnkranz, J.: A Study Using N-grams Features for Text Categorization, Technical report, Austrian Research Institute for Artificial Intelligence (1998), OEFAI-TR-98-30.

[Gomes 97] Gomes, C. and Selman, B.: Problem Structure in the Presence of Perturbations, in *Proc. AAAI-97*, pp. 221–227 (1997).

[goo] Google, http://www.google.com.

[Gu 93] Gu, J.: Local Search for Satisfiability (SAT) Problem, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 4, pp. 1108–1129 (1993).

[Gu 94] Gu, J.: Global Optimization for Satisfiability Problem, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 6, No. 3, pp. 361–381 (1994).

[Guare 90] Guare, J.: *Six Degrees of Separation: A Play*, Vintage Books, New York (1990).

[グエン 01] グエン, 石川, 阿部：知識の類似性を利用した概略推論法, 電子情報通信学会論文誌 (D-I), Vol. J84-D-I, No. 4, pp. 389–400 (2001).

[Haliday 76] Haliday, M. A. K. and Hasan, R.: *Cohesion in English*, Addison Wesley Longman (1976), (テクストはどのように構成されるか, 安藤 貞雄 他訳, ひつじ書房 1997).

[Harnad 90] Harnad, S.: The Symbol Grounding Problem, *Physica D*, No. 42, pp. 335–346 (1990).

[Hashimoto 00] Hashimoto, K., Matsumoto, K., and Shiratori, N.: Probabilistic Modeling of Alarm Observation Delay in Network Diagnosis, in *Proc. of PRICAI 2000*, pp. 734–744 (2000).

[Hisamitsu 00] Hisamitsu, T., Niwa, Y., and Tsujii, J.: A Method of Measuring Term Representativeness — Baseline Method Using Co-occuences Distribution —, in *Proc. Coling 2000*, pp. 320–326 (2000).

[Hobbs 93] Hobbs, J., Stickel, M., Appelt, D., and Martin, P.: Interpretation as abductions, *Artificial Intelligence*, Vol. 63, No. 1–2, pp. 69–142 (1993).

[Hoey 91] Hoey, M.: *Patterns of lexis in text*, Oxford University Press (1991).

[Hofmann 98] Hofmann, T. and Puzicha, J.: Statistical Models for Co-occurrence Data, Technical Report AIM-1625, Massachusetts institute of technology (1998).

[Hofstadter 79] Hofstadter, D. R.: *Goedel, Escher Bach: an Eternal Golden Braid*, NY: Basic Books (1979), (ゲーデル，エッシャー，バッハ あるいは不思議の環，野崎昭弘 他 訳，白揚社，1985).

[堀 99] 堀浩一：思考の可視化，可視化情報学会誌，Vol. 19, No. 72, pp. 2–6 (1999).

[井上 92] 井上克巳：アブダクションの原理，人工知能学会論文誌，Vol. 7, No. 1, pp. 48–59 (1992).

[伊里 76] 伊里，古林：ネットワーク理論，日科技連出版社 (1976).

[石塚 96] 石塚満：知識の表現と高速推論，第6章，丸善 (1996).

[石塚 97] 石塚，原：数理計画法とAIの推論，人工知能学会誌，Vol. 12, No. 2, pp. 179–187 (1997).

[伊藤 91] 伊藤，石塚：推論パスネットワークによる高速仮説推論システム，人工知能学会誌，Vol. 6, No. 4, pp. 501–509 (1991).

[Kageura 96] Kageura, K. and Umino, B.: Methods of Automatic Term Recognition, *Terminology*, Vol. 3, No. 2, pp. 259–289 (1996).

[Kamath 94] Kamath, A., Motwani, R., Palem, K., and Spirakis, P.: Tail Bounds for Occupancy and the Satisfiability Threshold Conjecture, *FOCS'94*, pp. 502–511 (1994).

[Kautz 97] Kautz, H., Selman, B., and Shah, M.: The Hidden Web, *AI magagine*, Vol. 18, No. 2, pp. 27–35 (1997).

[河原 98] 河原，山本，佐々木，久保川，朝原：仮説推論を用いた停電作業スケジューリングに関する一考察，電気学会論文誌B, Vol. 118-B, No. 4 (1998).

[木本 91] 木本晴夫：日本語新聞記事からのキーワード自動抽出と重要度評価, 電子情報通信学会誌, Vol. 74-D-I, No. 8, pp. 556–266 (1991).

[Kleinberg 99] Kleinberg, J.: The Small-World Phenomenon: An Algorithmic Perspective, Technical Report TR 99-1776, Cornell University (1999).

[近藤 94] 近藤, 石塚：述語論理知識を扱う仮説推論における最適解の高速推論法, 人工知能学会誌, Vol. 9, No. 2, pp. 110–118 (1994).

[Kosala 00] Kosala, R. and Blockeel, H.: Web mining research: A survey, *ACM SIGKDD Explorations*, Vol. 1, No. 2, pp. 1–15 (2000).

[越野 01] 越野, 林, 木村, 広瀬：述語論理知識を扱う全解探索仮説推論の高速化, 人工知能学会誌, Vol. 16, No. 2, pp. 202–211 (2001).

[Kumar 99] Kumar, S. R., Raghavan, P., Rajagopalan, S., and Tokins, A.: Trawling the web for emerging cyber communities, in *Proc. 8th WWW Conf.* (1999).

[Lempel 00] Lempel, R. and Moran, S.: The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect, in *Proc. 9th WWW Conf.* (2000).

[Levy 97] Levy, A. Y., Fikes, R. E., and Sagiv, Y.: Speeding up Inferences using Relevance Reasoning: a Formalism and Algorithms, *Artificial Intelligence*, Vol. 97, pp. 83–136 (1997).

[Luhn 57] Luhn, H. P.: A statistical approach to mechanized encoding and searching of literary information, *IBM Journal of Research and Development*, Vol. 1, No. 4, pp. 390–317 (1957).

[牧野 90] 牧野, 石塚：制約評価機構つき仮説推論システムとその回路ブロック設計への応用, 人工知能学会誌誌, Vol. 5, No. 5, pp. 640–648 (1990).

[Mani 97] Mani, I. and Bloedorn, E.: Multi-document Summarization by Graph Search and Matching, in *Proc. of the 14th National Conference on Artificial Intelligence*, pp. 622–628 (1997).

[Marchiori 00] Marchiori, M. and Latora, V.: Harmony in the small-world, *Physica A*, Vol. 285, pp. 539–546 (2000).

[Mathias 01] Mathias, N. and Gopal, V.: Small worlds: How and why, *Physical Review E*, Vol. 63, No. 2 (2001).

[松尾 98] 松尾, 二田, 石塚：SL 法：線形・非線形計画法の併用によるコストに基づく仮説推論の準最適解計算, 人工知能学会誌, Vol. 13, No. 6, pp. 953–961 (1998).

[松尾 00] 松尾, 石塚：並行プロセスによる高速仮説推論法, 第 120 回知能と複雑系研究会、情処研報, Vol. 2000, No. 55, pp. 1–8 (2000).

[Matsuo 01] Matsuo, Y., Ohsawa, Y., and Ishizuka, M.: A Document as a Small World, in *Proceedings the 5th World Multi-Conference on Systemics, Cybenetics and Infomatics (SCI2001)*, Vol. 8, pp. 410–414 (2001).

[松尾 02a] 松尾, 大澤, 石塚：ユーザの心理的距離に則した Web ページ間の新しい距離の定義, 人工知能学会 (2002), 投稿中.

[松尾 02b] 松尾, 石塚：語の共起の統計情報に基づく文書からのキーワード抽出アルゴリズム, 人工知能学会 (2002), 投稿中.

[松尾 02c] 松尾, 大澤, 石塚：電子掲示板における会話からのハイライト部分の抽出, 第 46 回人工知能基礎論研究会 (2002).

[松村 99] 松村, 大澤, 谷内田：AAS: 文書の組合わせによってユーザの興味を満足する検索システム, 人工知能学会誌, Vol. 14, No. 6, pp. 245–253 (1999).

[Milgram 67] Milgram, S.: The small-world problem, *Psychology Today*, Vol. 2, pp. 60–67 (1967).

[Miller 95] Miller, G. A.: WordNet: A lexical database for English, *Communications of the ACM*, Vol. 38, No. 11, pp. 39–41 (1995).

[Minton 92] Minton, S., Johnston, M. D., Philips, A. B., and Laird, P.: Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems, *Artificial Intelligence*, Vol. 58, pp. 161–205 (1992).

[Mitchell 96] Mitchell, D. and Levesque, H.: Some pitfalls for experimenters with random SAT, *Artificial Intelligence*, Vol. 81, pp. 111–125 (1996).

[Morris 93] Morris, P.: The Breakout Method for Escaping from Local Minima, in *Proc. AAAI-93*, pp. 40–45 (1993).

[村田 01] 村田剛志：参照の共起性に基づくWebコミュニティの発見, 人工知能学会誌, Vol. 16, No. 3, pp. 316–323 (2001).

[長尾 76] 長尾, 水谷, 池田：日本語文献における重要語の自動抽出, 情報処理, Vol. 17, No. 2, pp. pp.110–117 (1976).

[中島 01] 中島, 橋田, 森, 伊東, 本村, 車谷, 山本, 和泉, 野田：情報インフラに基づくグラウンディングとその応用 – サイバーアシストプロジェクトの概要 –, コンピュータソフトウェア, Vol. 18, No. 4, pp. 48–56 (2001).

[那州 99] 那州川, 諸橋, 長野：テキストマイニング–膨大な文書データの自動分類による知識発見–, 情報処理, Vol. 2, No. 40, pp. 358–364 (1999).

[Ng 97] Ng, H. T., Goh, W. B., and Low, K. L.: Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization, in *Proc. ACM SIGIR'97* (1997).

[Noreault 77] Noreault, T., McGill, M., and Koll, M. B.: *A Performance Evaluation of Similarity Measure, Document Term Weighting Schemes and Representations in a Boolean Environment*, Butterworths, London (1977).

[大平 99] 大平, 帆足, 松本, 橋本, 白井：AIC を用いた重要語抽出手法と重要語を用いたターム重みづけ手法の提案・評価, 知識発見のための自然言語処理シンポジウム (1999).

[大澤 94] 大澤, 石塚：仮説推論における準最適解を多項式時間で計算するネットワーク化バブル伝播法, 電子情報通信学会論文誌, Vol. J 77-D-II, No. 9, pp. 1817–1829 (1994).

[大澤 95a] 大澤, 石塚：改良型ネットワーク化バブル伝播法による低次多項式時間仮説推論法, 人工知能学会誌, Vol. 10, No. 1, pp. 123–130 (1995).

[大澤 95b] 大澤, 石塚：多項式時間仮説推論を達成するネットワーク化バブル伝播法の述語論理への拡張, 人工知能学会誌, Vol. 10, No. 5, pp. 731–740 (1995).

[Ohsawa 97] Ohsawa, Y. and Ishizuka, M.: Networked Bubble Propagation: A Polynomial-time Hypothetical Reasoning Method for Computing Near-optimal Solutions, *Artificial Intelligence*, Vol. 91, pp. 131–154 (1997).

[大澤 98a] 大澤, 石塚：コストに基づく仮説推論を多項式時間で達成する新しい十分条件, 人工知能学会誌, Vol. 13, No. 3, pp. 415–423 (1998).

[大澤 98b] 大澤, 須川, 谷内田：ユーザの変化する興味を理解し表現する文献検索支援システム Ind ex Navigator, 人工知能学会誌, Vol. 13, No. 3 (1998).

[大澤 01] 大澤, 相馬, 臼井, 松村, 松尾：会話展開キーグラフによる Web コミュニティーの特性表現, SIG-FAI, SIG-KBS 合同研究会 (2001).

[岡本 93] 岡本, 石塚 : 整数計画法の近似解法を適用した準最適解計算の高速仮説推論法, 人工知能学会誌, Vol. 8, No. 2, pp. 222–229 (1993).

[奥村 99] 奥村, 難波 : テキスト自動要約に関する研究動向, 自然言語処理, Vol. 6, No. 6 (1999).

[奥村 00] 奥村, 難波 : テキスト自動要約に関する最近の話題, Technical Report IS-TM-2000-001, 北陸先端科学技術大学院大学 (2000).

[Paice 90] Paice, C.: Constructing Literature Abstracts by Computer: Techniques and Prospects, *Information Processing and Management*, Vol. 26, No. 1, pp. 171–186 (1990).

[Pereira 93] Pereira, F., Tishby, N., and Lee, L.: Distributional Clustering of English Words, in *Proc. 31th Meeting of the Association for Computational Linguistics*, pp. 183–190 (1993).

[Poole 87] Poole, D., Goebel, R., and Aleliunas, R.: *Theorist: a logical reasoning system for defaults and diagnosis*, Springer Varlag, New York (1987).

[Poole 90] Poole, D.: A methodology for using a default and abductive reasoning system, *International Journal of Intelligent Systems*, Vol. 5, No. 5, pp. 521–548 (1990).

[Porter 80] Porter, M. F.: An algorithm for suffix stripping, *Program*, Vol. 14, No. 3, p. 130 (1980).

[Prendinger 00] Prendinger, H., Ishizuka, M., and Yamamoto, T.: The Hyper System: Knowledge Reformation for Efficient First-order Hypothetical Reasoning, in *Proc. PRICAI 2000*, pp. 93–103 (2000).

[Rajman 98] Rajman, M. and Besancon, R.: Text Mining – knowledge extraction from unstructured textual data, in *Proceedings of the 6th Conference of International Federation of Clas sification Societies* (1998).

[Reiter 87] Reiter, R.: A theory of diagnosis from first principles, *Artificial Intelligence*, Vol. 13, No. 1-2, pp. 57–95 (1987).

[Salton 88] Salton, G.: *Automatic Text Processing*, Addison-Wesley (1988).

[Santos, Jr. 94] Santos, Jr., E.: A Linear Constraint Satisfaction Approach to Cost-Based Abduction, *Artificial Intelligence*, Vol. 65, No. 1, pp. 1–27 (1994).

[Schutze 95] Schutze, H., Hull, D. A., and Pedersen, J. O.: A comparison of classifiers and document representations for the routing problem, in *Proc. ACM SIGIR'95* (1995).

[Selman 93] Selman, B. and Kautz, H.: Domain-Independent Extensions to GSAT:Solving Large Structured Satisfiability Problems, in *Proc. IJCAI-93*, pp. 290–295 (1993).

[Selman 96] Selman, B., Mitchell, D., and Levesque, H.: Generating Hard Satisfiability Problems, *Artificial Intelligence*, Vol. 81, pp. 17–29 (1996).

[Selman 97] Selman, B., Kautz, H., and McAllester, D.: Ten Challenges in Propositional Reasoning and Search, in *Proceedings 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, pp. 50–54 (1997).

[Shneiderman 98] Shneiderman, B.: *Designing the User Interface: Strategies for Effective Human-Computer Interaction, 3rd Ed.*, Addison-Wesley (1998).

[Skorokhod'ko 72] Skorokhod'ko, E.: Adaptive method of automatic abstracting and indexing, *Information Processing*, Vol. 71, pp. 1179–1182 (1972).

[Sparck-Jones 72] Sparck-Jones, K.: A Statistical Interpretation of Term Specificity and Its Application in Retrieval, *Journal of Documentation*, Vol. 28, No. 5, pp. 111–121 (1972).

[砂山 99] 砂山, 大澤, 谷内田：事象ごとの生起確率から未知事象発見を支援する手法とそのアンケート調査への適用, 人工知能学会論文誌, Vol. 14, No. 2, pp. 349–358 (1999).

[高間 95] 高間, 石塚：知識の実行時リフォメーションに基づく仮説推論の高速化手法, 人工知能学会誌, Vol. 10, No. 6, pp. 913–920 (1995).

[棚橋 99] 棚橋, 福田茂紀, 石塚：命題レベルの高速解法の利用を図るコストに基づく述語論理版仮説推論法, 人工知能学会誌, Vol. 14, No. 6, pp. 1100–1107 (1999).

[Tanaka-Ishii 96] Tanaka-Ishii, K. and Iwasaki, H.: Clustering co-occurrence graph using transitivity, in *Proc. 16th International Conference on Computational Linguistics*, pp. 680–585 (1996).

[Tanaka 96] Tanaka, K. and Iwasaki, H.: Extraction of lexical translations from non-aligned corpora, in *Proc. 16th International Conference on Computational Linguistics*, pp. 580–585 (1996).

[寺本 99] 寺本, 宮原, 松本：類似文書検索のためのタームの共起語分布分析による計算, 情報処理学会第 59 回全国大会論文誌, IP-06 (1999).

[Tkach 98] Tkach, D.: Text Mining Technology, White paper, IBM Software Solutions (1998).

[徳永 99] 徳永健伸：情報検索と言語処理, 東京大学出版会 (1999).

[辻野 97] 辻野, エドガー, 桝谷：画像認識のためのマルチエージェントによる仮説推論, 人工知能学会誌, Vol. 12, No. 3, pp. 440–447 (1997).

[Turing 50] Turing, A. M.: Computing machinery and intelligence, *Mind*, Vol. 59, p. 433 (1950).

[Voudouris 95] Voudouris, C. and Tang, E. P. K.: Guided Local Search, Technical report csm-247, University of Essex, UK (1995).

[Wah 97] Wah, B. W. and Shang, Y.: Discrete Lagrangian-Based Search for Solving MAX-SAT Problems, in *Proc. IJCAI-97*, pp. 378–383 (1997).

[Walsh 99] Walsh, T.: Search in a Small World, in *Proc. IJCAI-99*, pp. 1172–1177 (1999).

[Watts 98] Watts, D. and Strogatz, S.: Collective dynamics of small-world networks, *Nature*, Vol. 393, pp. 440–442 (1998).

[Watts 99] Watts, D.: *Small worlds: the dynamics of networks between order and randomness*, Princeton (1999).

[Witten 91] Witten, I. H. and Bell, T. C.: The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression, *IEEE Trans. Information Theory*, Vol. 37, pp. 1085–1094 (1991).

[Wu 00] Wu, Z. and Wah, B. W.: An Efficient Global-Search Strategy in Discrete Lagrangian Methods for Solving Hard Satisfiability Problems, in *Proceedings 17th National Conference on Artificial Intelligence (AAAI-2000)*, pp. 310–315 (2000).

[Y.Matsuo 01] Y.Matsuo, , Y.Ohsawa, , and M.Ishizuka, : Average-click: A new definition of distance on the World Wide Web, in *WI-2001* (2001), to appear.

[吉川] 吉川：人工物工学の提唱：http://www.race.u-tokyo.ac.jp/overview/overview01-j.html.

[Zanette ar] Zanette, D. H.: Dynamics of rumor propagation on small-world networks (to appear).

発表文献

学会誌論文

- 松尾豊, 二田丈之, 石塚満. SL 法：線形・非線形計画法の併用によるコストに基づく仮説推論の準最適解計算. 人工知能学会誌, Vol. 13, No. 6, pp. 953–961, 1998.
- 松尾豊, 横山明彦. 送電網の過負荷解消のための FACTS 機器設置の最適化手法. 電気学会 B 部門誌, Vol. 120-B, No. 8/9, pp. 1061–1070, 2000.
- 松尾豊, 石塚満. コストに基づく仮説推論の 2 種の連続値最適化問題への置換法とその協調による推論法. 人工知能学会誌, Vol. 16, No. 4, pp. 400–407, 2001.
- 松尾 豊, 大澤 幸生. 語の共起の統計情報に基づく文書からのキーワード抽出アルゴリズム. 人工知能学会誌, 条件付採録
- 松尾豊, 大澤幸生, 石塚満. Small world 構造に基づく文書からのキーワード抽出. 情報処理学会論文誌, 条件付採録
- 松尾　豊, 大澤　幸生, 石塚　満. ユーザの心理的距離に即した Web ページ間の新しい距離の定義, 人工知能学会誌, 投稿中
- Mitsuru Ishizuka and Yutaka Matsuo. SL method for computing a near-optimal solution using linear and non-linear programming in cost-based hypothetical reasoning. *Knowledge-based System*, Submitting.
- Naohiro Matsumura, Yutaka Matsuo, Yukio Ohsawa and Mitsuru Ishizuka. Discoverign Emerging Topics from WWW. *Journal of Contingencies and Crisis Manage-*

ment, Submitting.

- Yutaka Matsuo, Yukio Ohsawa, and Mitsuru Ishizuka. Average-clicks: A New Measure of Distance on the World Wide Web. *Journal of Intelligent Information Systems*, Submitting

特許出願

- 「データからの特徴アイテム抽出方法」, 松尾 豊・石塚 満, 特願 2001-254905 平成 13 年 8 月 24 日

国際会議論文

- Mitsuru Ishizuka and Yutaka Matsuo. SL method for computing a near-optimal solution using linear and non-linear programming in cost-based hypothetical reasoning. In *Proceedings 5th Pacific Rim Conference on Artificial Intelligence (PRICAI'98)*, LNAI 1531, pp. 611–625, 1998.

- Yutaka Matsuo and Akihiko Yokoyama. Optimization of installation of FACTS device in power system planning by both tabu search and nonlinear programming methods. In *Proceedings Intelligent System Application to Power Systems (ISAP-99)*, pp. 250–254, 1999.

- Yutaka Matsuo and Mitsuru Ishizuka. Fast hypothetical reasoning by parallel processing. In *Proceedings AAAI-2000 Workshop*, 2000.

- Yutaka Matsuo and Mitsuru Ishizuka. Fast hypothetical reasoning by parallel processing. In *Proceedings 6th Pacific Rim Conference on Artificial Intelligence (PRICAI'00)*, LNAI 1886, p. 790, 2000.

- Mitsuru Ishizuka, Yutaka Matsuo and Helmut Prendinger. Polynomial-time Cost-

based Hypothetical Reasoning: Propositional and Predicate Logic Cases, Dagstuhl seminar on Deduction, 2001.

- Yutaka Matsuo, Yukio Ohsawa, and Mitsuru Ishizuka. A document as a small world. In *Proceedings the 5th World Multi-Conference on Systemics, Cybenetics and Infomatics (SCI2001)*, Vol. 8, pp. 410–414, 2001.

- Yutaka Matsuo, Yukio Ohsawa, and Mitsuru Ishizuka. Average-clicks: A new measure of distance on the World Wide Web. In *Proceedings First Asia-Pacific Converence on Web Intelligence (WI-2001)*, 2001.

- Yutaka Matsuo, Yukio Ohsawa, and Mitsuru Ishizuka. Discovering hidden relation behind a link. In *Proceedings the Fifth International Conference on Knowledge-based Intelligent Information Engineering Systems & Allied Technologies (KES-2001)*, 2001.

- Yutaka Matsuo, Yukio Ohsawa, and Mitsuru Ishizuka. KeyWorld: Extracting keywords from a document as a small world. In *Proceedings the Fourth International Conference on Discovery Science (DS-2001)*, 2001.

- Yutaka Matsuo, Yukio Ohsawa and Mitsuru Ishizuka. A Psychological Metric of Distance on the WWW. submitting to WWW2002.

- Yutaka Matsuo and Mitsuru Ishizuka. Two Basic Transformation of Clauses into Constraints for Cost-based Hypothetical Reasoning. submitting to ECAI-2002.

- Yutaka Matsuo and Mitsuru Ishizuka. Keyword Extraction from a Single Document using Co-occurrence Statistical Information. submitting to AAAI-2002.

国内発表

- 松尾　豊, 二田　丈之, 石塚　満. 線形・非線形計画法の併用による高速仮説推論の改善. 情報処理学会第 54 回（平成 9 年前期）全国大会, 2-151, 1997.
- 松尾　豊, 二田　丈之, 石塚　満. 仮説推論における準最適解を多項式時間で計算するＳＬ法. 1997 年度人工知能学会全国大会 (第 11 回), S1-03, 1997.
- 松尾　豊, 横山　明彦. 電力系統設備計画における移相器設置の最適化手法. 平成 10 年電気学会電力・エネルギー部門大会，188, 1998.
- 松尾　豊, 横山　明彦. 電力系統設備計画における FACTS 機器設置の最適化手法. 平成 10 年電力技術・電力系統技術合同研究会，PS-98-72, PSE-98-62, 1998.
- 松尾　豊, 横山　明彦. 電力系統設備計画における緊急制御を考慮した FACTS 機器設置に関する一考察. 平成 11 年電気学会全国大会, 1400, 1999.
- 松尾　豊、横山　明彦.　電力系統設備計画における送電網の過負荷解消のための FACTS 機器設置の最適化手法 (Optimization of Installation of FACTS Device to Avoid Thermal Constraints in Power System Planning). 平成 11 年電気学会電力・エネルギー部門大会, 28, 1999.
- 松尾　豊、石塚 満. 並行プロセスによる高速仮説推論法. 平成 12 年情報処理学会全国大会, 2000.
- 松尾 豊, 石塚 満. 仮説推論問題における問題例の考察. 第 1 回 MYCOM（人工知能学会）, 2000.
- 松尾 豊, 石塚 満. 並行プロセスによる高速仮説推論法. FJK-2000（情報処理学会研究会）, 2000.
- 松尾 豊, 石塚 満. 仮説推論における問題例の分析と検証. 第 41 回人工知能基礎論研究会 (SIG-FAI), 2000.

- 松尾 豊, 石塚 満. 仮説推論の柔軟な拡張についての考察. 平成12年情報処理学会全国大会, 2000.
- 松尾 豊, 大澤 幸生, 石塚 満. チャンスディスカバリーの最適化問題としての定式化. 人工知能学会 基礎論研究会, 2000.
- 松尾 豊, 石塚 満. 仮説推論の難易度による単体法の有効性の検討. 人工知能学会全国大会, 2001.
- 松尾 豊, 大澤 幸生, 石塚 満. Small world と Average-clicks. 第2回 MYCOM, 2001.
- 加藤　優, 松尾 豊, 石塚 満. 文書の構造によるクラスタリング. 第2回 MYCOM, 2001.
- Hiroshi Taira, Yutaka Matsuo, Yukio Ohsawa, Mitsuru Ishizuka. AreaView2001: A WWW Organization System with KeyGraph Technology. JSAI Workshop 2001.
- Yutaka Matsuo, Yukio Ohsawa, Mitsuru Ishizuka. Document as a Small World, JSAI Workshop. 2001.
- 大澤, 相馬, 臼井, 松村, 松尾. 会話展開キーグラフによる Web コミュニティーの特性表現. SIG-FAI, SIG-KBS 合同研究会、2001.
- 松尾 豊、大澤 幸生、石塚 満. 電子掲示板における会話のハイライト部分の抽出. SIG-FAI 研究会、2002.
- 松尾 豊、大澤 幸生、石塚 満. Small World のさまざまな拡張についての考察. SIG-FAI 研究会、2002.
- 福田 隼人、松尾 豊、石塚 満. ユーザ個人の閲覧履歴からのキーワード抽出によるブラウジング支援. 電子情報通信学会 言語理解とコミュニケーション研究会 (NLC)、2002 (発表予定).

- 福田 隼人、松尾 豊、石塚 満. ユーザの閲覧履歴を考慮したキーワード抽出によるブラウジング支援. 電子情報通信学会全国大会、2002 (発表予定).

記事等

- 「電子掲示版から面白い着眼点抽出」, 日本工業新聞, 平成 14 年 1 月 23 日.