人　工　知　能 95－6
（1994．7．21）

確率的制約プログラミング

橋田 浩一
〒305 つくば市梅園 1-1-4
電子技術総合研究所
自然言語研究室

宮田 高志
〒113 文京区本郷 7-3-1
東京大学
理学部情報科学科

長尾 確
〒141 品川区東五反田 3-14-13
高輪ミューズビル
ソニーコンピュータサイエンス研究所

制約(情報の流れを捨象した記述のレベル)に基づく情報処理システムの設計における主要な研究課題は、全体として妥当な計算が行なわれることを保証する方法である。ここでは、隠れマルコフモデルや確率的文脈自由文法などの一般化としてHorn節プログラムに確率的意味を与えた上で、最尤推定を行なう計算法を提案する。その計算は、確率的従属事象を効率的に扱うものであり、また一般的な意味でのA* 探索と見なすことができる。

Probabilistic Constraint Programming

HASIDA Kôiti
Electrotechnical Laboratory
1-1-4 Umezono, Tukuba
Ibaraki 305, Japan
hasida@etl.go.jp

MIYATA Takashi
University of Tokyo
7-3-1 Hongo, Bunkyo-ku,
Tokyo 113, Japan
mya-u@is.s.u-tokyo.ac.jp

NAGAO Katashi
Sony Computer Science Lab. Inc.
3-14-13 Higashi-gotanda,
Shinagawa-ku, Tokyo 141, Japan
nagao@csl.sony.co.jp

A major research issue in designing information-processing systems based on constraint (level of description abstracting away from information flow) is how to guarantee global adequacy of computation. A probabilistic semantics of Horn clause programs is introduced, which is a generalization of Hidden Markov Models, Stochastic Context-Free Grammars, etc., and a computational method for maximum-likelihood estimation is proposed. This computation deals efficiently with probabilistically dependent events, and is regarded as a sort of A* search in a general sense.

1 はじめに

情報の流れをこと細かく記述する手続き型の設計法に従う限り、複雑な情報の流れを実現しようとするとシステムが複雑化して管理不能になり、設計が破綻する。人間なみに柔軟な情報処理が機械にできていないのは、こういうわけでさまざまな種類の情報の間の複雑な相互作用を実現できないからである。そこで、仕事の手順を具体的にこと細かく設計するのをやめて、情報の流れをいっさい明示しない設計、つまり**制約** (constraint) を用いることを考える。制約に基づく方法では、設計の段階で破綻することはないにしてもそれによって実現される計算は極めて複雑なものになるから、ここでの主要な研究課題は、いかに情報の流れを制御して全体として妥当な計算が行なわれるようにするか、という問題である。

本稿では、制約に基づく設計において全体として妥当な計算が行なわれることを保証するための一般的な方法について述べる。それは、1 階の Horn 節プログラムによって表現された制約の意味を解の候補の確率分布としてとらえ、統計的最尤解を求める、一種の A* 探索法である。この枠組は、隠れマルコフモデル (HMM) (Xuang, Ariki, & Jack, 1990) や確率的文脈自由文法 (SCFG) や確率的木接合文法 (STAG) (Shabes, 1992) など、広く用いられている既存の確率的方法の自然な拡張になっている。それら既存の方法では確率的に独立な事象しか扱っていないが、以下で論ずる主要な問題は、説明 (解の候補) の間の構造共有から生ずる確率的依存関係の扱いである。

以下ではまず 2 節で基本的な概念を導入し、3 節において構造共有を用いた効率的な記号計算法について述べ、次に 4 節で包摂化を用いて最尤解を求める方法を与えた上で、5 節で計算の例を示す。

2 確率的 Horn 節プログラム

以下ではたとえば

(a) −p(A) −q(A).

(b) p(X) −X=f(a).

(c) p(Y) −Y=f(b).

(d) q(Z) −Z=f(U) −r(U).

(e) q(W) −W=g(V) −r(V).

(f) r(a).

(g) r(g).

のような 1 階の Horn 節プログラムを考える。大文字で始まる名前は変数を表わし、それ以外の名前は述語名と関数名である。要素式または束縛に符号 − の付いたものは負リテラルであり、符号なしの要素式は正リテラルである。以後本文中では、誤解の恐れないがない限り、リテラルと要素式または束縛とを特に区別せずに表記する。議論の便宜のため、以下ではしばしばプログラムを図で表わす。たとえば上記のプログラムは図 1 のように図示される。

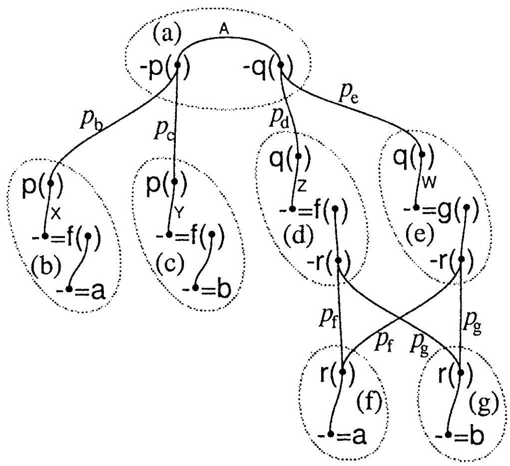

図 1: プログラムの図的表示

各閉曲線に囲まれた部分が節を表わす。節の中にある結線は、X などの項を表わす。節の外を通る結線を**単一化結線** (unification link) と呼ぶ。単一化結線は項同士またはリテラル同士の単一化可能性を表わす。2 つのリテラルが単一化結線で結ばれているときは、それらの対応する項が単一化結線で結ばれており、逆も真であるとする。ただし、束縛の間の単一化結線は束縛項 (束縛の第 1 引数) の間の単一化結線を含まない。また、2 つの束縛の間の単一化結線は初期

状態においては存在せず、後述のような計算の進行に伴って作られると考える。

要素式および束縛の連言を**仮説** (hypothesis) と呼ぶ。各節の前件 (正リテラルを除いたものの否定) はひとつの仮説である。正リテラルを持たない節を**先頭節** (top clause) と言う。先頭節の前件が、証明すべき最上位の仮説である。上記のプログラムの場合、(a) の否定 $\exists A\,\{p(A) \wedge q(A)\}$ が最上位の仮説である。(先頭節が複数個あるときは、いずれかひとつが表わす仮説を証明すればよいとする。) ここでは通例に従って最小 Herbrand モデルを仮定する (Lloyd, 1984)。したがって、p(•) の解集合は {f(a), f(b)} であり、q(•) の解集合は {f(a), f(b), g(a), g(b)} であるから、このプログラム全体の解集合は {f(a), f(b)} となる。

仮説の**説明** (explanation) とは、融合 (resolution) によってその仮説を束縛の連言に変換する仕方のことであり、その際に用いる節を下降的かつ左から右へという融合の適用順序[1]で並べたものによって表わす。たとえば、上記の仮説の説明のうち、f(a) という解に対応するものは、⟨b, d, f⟩ である。一方、⟨b, d, g⟩ という説明は矛盾を含む。単に「説明」と言った場合は、(いずれかの) 先頭節の前件の説明を意味する。プログラムは説明の集合を表わすと考えられる。

図1に示した通り、各節 Φ の正リテラルと他のリテラルとの間の単一化結線に正定数 p_Φ が割り当てられており、これを Φ の**基本確率** (basic probability) と言う。ひとつの説明の確率をそれが含む節 (の具現例) の基本確率の積とする。たとえば、説明 ⟨b, d, f⟩ の確率は $p_b p_d p_f$ である。各述語ごとにその定義節の基本確率の和を 1 以下に正規化しておけば、各仮説の解の確率の和が有限な値に収束するから、これに正規化を施すことにより、プログラムの意味を解の確率分布として解釈することができる[2]。

以下では、このようなプログラムに基づく最尤推定の方法、すなわち、最大の確率を持つ解 (最尤解) を求める計算法について論ずる。この計算においては、Prolog のようにトップダウンとか左から右へとかいう処理の方向性があらかじめ固定されておらず、確率的な文脈に依存してさまざまな方向への情報の流れが生ずる。この意味において、プログラムは制約 (constraint) として振舞う。

最尤推定を行なうには、目下のところ最大確率を持つ説明について、それが矛盾を含むかどうかを調べ、矛盾した説明にしか含まれないプログラムの部分を消去してゆけばよい。最初に見付かった無矛盾な説明が最尤解のひとつである。この計算は A* 探索法の一種と考えられる。その際、計算の効率を上げるため、異なる説明の間でかなり徹底的に構造共有を行なうことにより、なるべく多くの説明を一挙に調べられるような方法 (Hasida, 1994) を用いる。本論文で述べる主要な技術的成果は、そのような構造共有の方式の下で確率最大の説明を効率的に同定する方法である。

3 記号計算

プログラム中の 2 つの項の間に**依存関係** (dependency) があるとは、それらの項 (の具現例) がある説明において単一化されることだとしよう。そうした単一化は、図1のようなプログラムのネットワーク中の項の経路によって媒介される。そのような経路をその依存関係の**依存経路** (dependency path) と言う。たとえば図1においては、項 X と W の間の依存関係が太線で示した依存経路 X·A·W によって媒介されている。

記号計算の目的は、矛盾を含む説明を除去することと考える。説明における矛盾とは、2 つの矛盾する束縛項 (たとえば •=f(•) と •=g(•) の第 1 引数) がその説明の中で依存経路によって結ばれていることである。このような依存経路を矛盾した依存経路と呼ぼう。図1の X·A·W は矛盾した依存経路である。

すると、矛盾した依存経路を除去する計算を行

[1] この議論は各節の前件のリテラルの間に左右の順序を仮定しているが、この順序は便宜上のものであり、実際の計算には無関係である。また、以下で述べる計算は融合ではない。

[2] このような確率的解釈が必要なのは基本確率に関するパラメタ学習を行なう場合などであり、以下で述べる計算においては確率的解釈は必須ではない。

なえばよいことになる。それには、そのような依存経路の通り道にある節や結線を消去する必要があるが、それによって解が失われないようにするためには、矛盾を含む説明のみに含まれる部分だけを消去しなければならない。たとえば、図1の節 (e) はいかなる解にも含まれないので消去してもよいが、(b) は解 ⟨b, d, f⟩ に含まれるので消去してはならない。

解と矛盾した説明が重なり合っている可能性があるとき、矛盾した説明だけを除去するため、**包摂化** (subsumption) という演算を用いて一種のプログラム変換を行なう。包摂化とは、単一化を一般化したものである。プログラムの各部がその基礎具現例 (ground instances) の集合を表わすと考えて、項 ξ と項 η について $\xi \supseteq \eta$ であることを ξ が η を**包摂** (subsume) すると言い、さらに ξ が束縛項であるとき、ξ を η の**原点** (origin) と言う。項 (の集合) δ による包摂化は、δ を原点とする項を生成してゆく演算である。δ を包摂化の原点と言う。

この包摂化は、δ から出発して、δ を原点とする項からなる依存経路を作ってゆく。その依存経路上の項 τ が生成されたとき、τ が δ と矛盾しない束縛項ならば、τ と δ に対する2つの束縛を単一化結線で結ぶ。一方、τ が δ と矛盾する束縛項ならば τ を消去する (または初めから生成しない)。こうして矛盾した依存経路をすべて除去することができ、かつ解が失われることはないように、以下で包摂化の仕方を定める。ただし、紙面の都合により正確な詳細と形式的な正当性の証明は省略する。

項 (の集合) δ による項 σ から項 τ への包摂化の様子を図2に示す。δ は σ の原点であって τ の原点ではない。また、σ を引数とするリテラル π と、τ を引数とするリテラル ρ とは、単一化結線 λ によって結ばれているとする。(そのような π と ρ と λ の組が複数個あってもよい。)

節 Φ' は τ を含む節 Φ と同形であり、τ と対応する Φ' 中の項 τ' の原点の集合は τ の原点の集合に δ を加えたもので、Φ と Φ' の他の対応する項の原点の集合は等しいとする。Φ' のリテラルで ρ に対応するものを ρ' とすると、この包摂化により、λ は ρ' と π とを結ぶ単一化結線に置き換わる。その結果、$\tau' \supseteq \tau \cap \sigma$ となり、また、包摂化の後の τ を τ'' とすると、$\tau'' = \tau - \sigma$ である。Φ' がプログラム中に存在していない場合には包摂化によって Φ' を新たに作ることになる。そのような場合の包摂化を**展開** (unfolding) と言い、そうでない場合の包摂化を**畳込み** (folding) と言う[3]。展開においては $\tau' = \tau \cap \sigma$ であり、畳込みにおいては τ' に $\tau \cap \sigma$ が加わる。また展開においては、Φ の中の ρ 以外のリテラルに繋がった各単一化結線は複製されて、Φ' の中の対応するリテラルに繋がる。これにより包摂化は説明の集合を保存する。

束縛項 δ が包摂化によって複数個の項に分割されることがあるが、ある条件が成立すれば、それらの項の各々を包摂化の原点とする必要はなく、それらの和集合をまとめてひとつの原点 δ として扱うことにより、包摂化の原点の個数の増大を抑制し、δ を原点とする項を含むリテラルや節を構造共有して計算を効率化できる。その条件とは、δ のどの要素 (δ が分かれてできた項) を取っても、それと単一化結線で結ばれている項の集合が等しいというものである[4]。4節で述べる尤度の計算においてこの条件を用いる。束縛項同士の間の単一化結線は存在しないので、束縛の間の単一化結線が消去されてもこの条件には影響しない。また、もし $i \geq 2$ に対して束縛の第 i 引数同士の間の単一化結線を通る包摂化が行なわれても、この条件は依然として成立していることがある。

以上の計算法によって、たとえば、文脈自由文法などに基づく統語解析を多項式オーダーの計算量で行なうことができる。これは、文法規則に相当する節の具現化が、入力語列に含まれる項からの包摂化のみによって行なわれ、したがって、入力語例中の語数を n、ひとつの節に含まれる項の個数の最大値を M とすれば、包摂化によってできる節の個数が $O(n^M)$ となるためである。さらに、節を部分的に複写する手法を用いるこ

[3] 本稿で述べるプログラム変換はたとえば Tamaki and Sato (1983) のそれとは変換の単位が異なるので、「展開」や「畳込み」の意味もそれに応じて異なる。

[4] この条件は少し弱めることができるかも知れない。

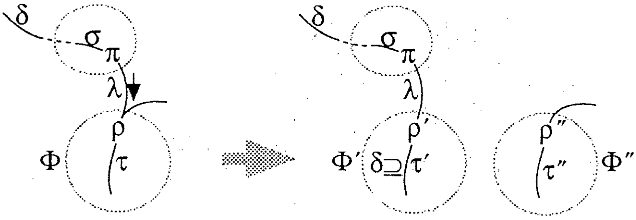

図 2: δ による σ から τ への包摂化

とにより、文脈自由文法の場合、空間計算量は $O(n^2)$ に、時間計算量は $O(n^3)$ になる (Hasida, 1994)。これは、よく用いられる効率的な統語解析アルゴリズムの計算量に等しい。

4 尤度

ある節を含む最大確率の説明の確率をその節の**尤度** (likelihood) と呼ぶ。最尤解を求めるには、尤度が最大の節を処理するような包摂化を行なっていけばよい。そのような包摂化が実行不能になったとき、確率最大の節から構成される説明は束縛の間の矛盾を含まないから解であり、しかもそれより大きな確率を持つ説明はないから、それが最尤解である。以下では、包摂化によって維持される構造共有の下で各節の尤度を効率よく計算する方法を示す。包摂化に伴って節や単一化結線が消去されたとき、この計算法によって各節の尤度を効率よく再計算することができる。

各リテラル α は**内部尤度** (internal likelihood) $\mathrm{IL}(\alpha)$ と**外部尤度** (external likelihood) $\mathrm{EL}(\alpha)$ を持つと考え、節の尤度がその中のリテラルの外部尤度の積として計算されるように、これらを定義する。図3に示すように、リテラルの内部尤度とは同じ節の中の他のリテラルの外部尤度の積であり、これは節の尤度をそのリテラルの外部尤度で割ったものに等しい。リテラルの外部尤度は、そのリテラルと単一化結線で結ばれた他のリテラルからの**接続尤度** (connection likelihood)[5] の最大値である。ただし、束縛 α の第 1 引数が、

[5] 一般には、束縛されていない項からその原点への接続尤度も考える必要があるが、5 節の例題では不要なので、詳細は割愛する。さらに、以下では簡単のため、どの項の原点もたかだか 1 つの場合を考える。

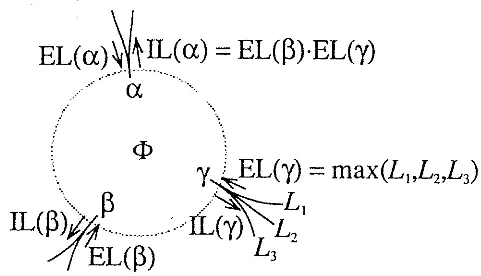

図 3: 尤度の間の関係

他の束縛項の原点 (の要素) でなく、かつ他の束縛項を原点とする項でもなければ、$\mathrm{EL}(\alpha)=1$ である。リテラル α から β への接続尤度 $\mathrm{CL}(\alpha,\beta)$ は、α と β が異符号ならば、両者を結ぶ単一化結線の基本確率と α の内部尤度との積である。α と β が同符号の場合、すなわちそれらが束縛に負号を付けた形のリテラルである場合はやや複雑なので、後で例を用いて説明する。

5 例

図1のプログラムにおける各リテラルの内部尤度と外部尤度を図4 に示す。太線は最大確率 $p_\mathrm{c}p_\mathrm{d}p_\mathrm{f}$ の説明を表わすが、この説明は矛盾を含んでいる。図に示したように、$p_\mathrm{b} < p_\mathrm{c}$ とすれば、

$$\begin{aligned}\mathrm{EL}(\mathrm{p_a}) &= \max(p_\mathrm{b}\mathrm{IL}(\mathrm{p_b}),\, p_\mathrm{c}\mathrm{IL}(\mathrm{p_c})) \\ &= \max(p_\mathrm{b},\, p_\mathrm{c}) = p_\mathrm{c}\end{aligned}$$

であり、$\mathrm{EL}(\mathrm{q_a})$、$\mathrm{EL}(\mathrm{r_d})$、$\mathrm{EL}(\mathrm{r_e})$ に関しても同様である。ここで、リテラルはその述語名 (束縛の場合には関数名) に節の添字を付けて表わし、リテラルの符号は省略する。この説明に含ま

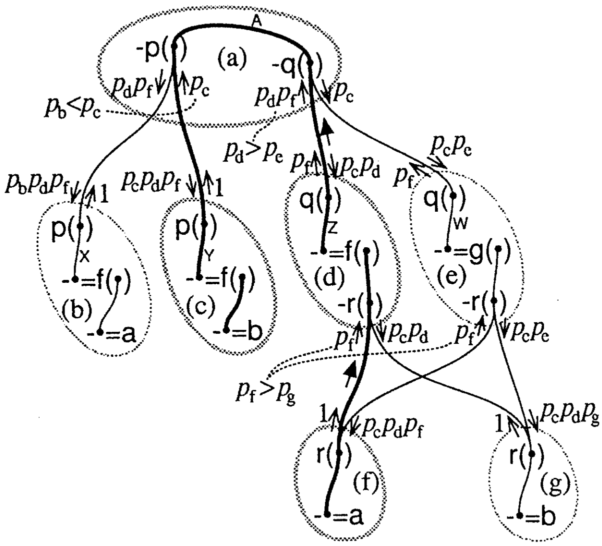

図 4: Z と a による包摂化の起動

れる節に対する包摂化としては、(c) から (a)、(d) から (a)、および (f) から (d) に向かうものが可能であるが、ここでは後の 2 つが実行されると仮定する。先端が黒い 3 角形の矢印が包摂化を表わす。

節 (e) はいかなる解にも含まれないが、このことを検出するには、W による包摂化を進めて行くか、単に W から項の単一化結線を辿って行けばよい。ここでは、Z による包摂化に伴って自動的に W による包摂化も行なわれるので、それによって W と X および W と Y の間の矛盾が検出され、こうして、W を原点とする A の具現例およびそれを含む (a) の具現例が消去されると考えてもよい。いずれにせよ、今後 (e) はどの説明にも含まれないので無視する。

図4に示した 2 つの包摂化を行ない、さらに Z による包摂化を (b) と (c) まで進めると、図5のようになる。(f) から (d) への包摂化によって (d) が (d1) と (d2) に分割され、Z が Z1 と Z2 に分割されているが、Z1 と単一化結線によって結ばれた項も Z2 と単一化結線によって結ばれた項も A1 だけだから、依然として Z=Z1∪Z2 をまとめて原点として扱うことができ、Z1 による包摂化と Z2 による包摂化を別個に行なう必要はない。

また、(f) から (d) への包摂化に伴い、(g) から (d) への包摂化も同時に行なわれたと考える。

本稿では簡単のため、ある項を原点とする包摂化を起動したときは、その瞬間に存在していた単一化結線を通ってその包摂化を最大限まで一挙に進めるものと仮定する[6]。この場合は Z による包摂化が (c) だけでなく (b) にも及ぶと考える。これにより、f_{d1}、f_{d2} と f_{b1}、f_{c1} の間の単一化結線ができる。

ここで接続尤度の計算法について述べておこう。説明 E の確率を pE と書く。

$$\begin{aligned}\mathrm{CL}(f_{b1},f_{d1})\mathrm{IL}(f_{d1}) &= \mathrm{p}\langle \mathrm{b1, d1, f}\rangle \\ \mathrm{CL}(f_{b1},f_{d2})\mathrm{IL}(f_{d2}) &= \mathrm{p}\langle \mathrm{b1, d2, g}\rangle\end{aligned}$$

となるように $\mathrm{CL}(f_{b1},f_{d1})$ および $\mathrm{CL}(f_{b1},f_{d2})$ を定義したい。最大確率の説明 (この場合には ⟨c1, d1, f⟩) においては束縛の外部尤度が 1 であると仮定すると、ここでは下記が成り立つ。

$$\begin{aligned}\mathrm{EL}(f_{c1}) &= \mathrm{EL}(f_{d1}) = 1 \\ \mathrm{IL}(f_{c1}) &= \mathrm{IL}(f_{d1}) = \mathrm{p}\langle \mathrm{c1, d1, f}\rangle\end{aligned}$$

[6] この仮定は緩和することができるが、その場合は注 5で触れたように尤度の計算法を一般化しなければならない。

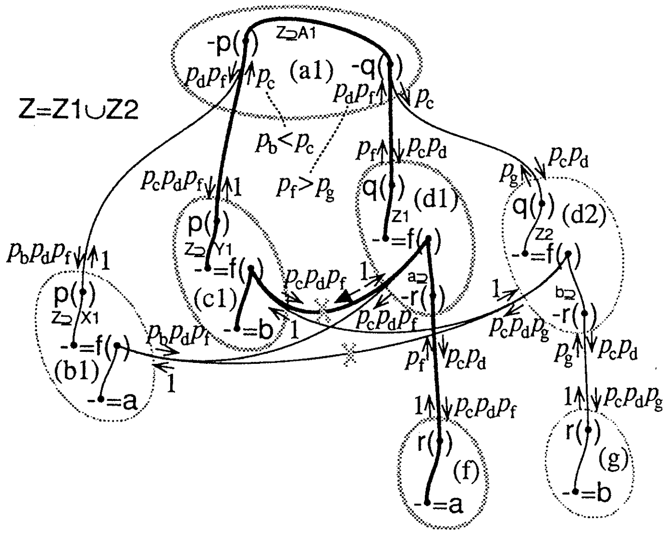

図 5: Z と a による包摂化の結果

また、

$$
\begin{aligned}
\mathrm{IL}(\mathrm{f_{b1}}) &= \mathrm{p}\langle \mathrm{b1, d1, f} \rangle \\
\mathrm{IL}(\mathrm{f_{d2}}) &= \mathrm{p}\langle \mathrm{c1, d2, g} \rangle
\end{aligned}
$$

が成り立つ。これは、(b1) を含む最大確率の説明は (d1) と (f) を含み、(d1) を含む最大確率の説明は (c1) を含むからであるが、一般の場合に同様のことを導くには、原点の要素 (ここでは Z1 と Z2) のいずれについてもそれと単一化結線で結ばれている項の集合 (ここでは {A1}) が等しいという、3 節で述べた条件を用いる。以上より下記を得る。

$$
\begin{aligned}
\mathrm{CL}(\mathrm{f_{b1}},\mathrm{f_{d1}}) &= \mathrm{p}\langle \mathrm{b1, d1, f} \rangle / \mathrm{IL}(\mathrm{f_{d1}}) \\
&= \mathrm{IL}(\mathrm{f_{b1}})/\mathrm{IL}(\mathrm{f_{c1}}) \\
\mathrm{CL}(\mathrm{f_{b1}},\mathrm{f_{d2}}) &= \mathrm{p}\langle \mathrm{b1, d2, g} \rangle / \mathrm{IL}(\mathrm{f_{d2}}) \\
&= \mathrm{p}\langle \mathrm{b1, d2, g} \rangle / \mathrm{p}\langle \mathrm{c1, d2, g} \rangle \\
&= \mathrm{p}\langle \mathrm{b1, d1, f} \rangle / \mathrm{p}\langle \mathrm{c1, d1, f} \rangle \\
&= \mathrm{IL}(\mathrm{f_{b1}})/\mathrm{IL}(\mathrm{f_{c1}}) \\
\mathrm{CL}(\mathrm{f_{b1}},\mathrm{f_{d1}}) &= \mathrm{CL}(\mathrm{f_{b1}},\mathrm{f_{d2}}) = p_b/p_c
\end{aligned}
$$

同様にして

$$
\begin{aligned}
\mathrm{CL}(\mathrm{f_{c1}},\mathrm{f_{d1}}) &= \mathrm{CL}(\mathrm{f_{c1}},\mathrm{f_{d2}}) \\
&= \mathrm{IL}(\mathrm{f_{c1}})/\mathrm{IL}(\mathrm{f_{c1}}) = 1 \\
\mathrm{CL}(\mathrm{f_{d1}},\mathrm{f_{b1}}) &= \mathrm{CL}(\mathrm{f_{d1}},\mathrm{f_{c1}}) \\
&= \mathrm{IL}(\mathrm{f_{d1}})/\mathrm{IL}(\mathrm{f_{d1}}) = 1 \\
\mathrm{CL}(\mathrm{f_{d2}},\mathrm{f_{b1}}) &= \mathrm{CL}(\mathrm{f_{d2}},\mathrm{f_{c1}}) \\
&= \mathrm{IL}(\mathrm{f_{d2}})/\mathrm{IL}(\mathrm{f_{d1}}) = p_g/p_f
\end{aligned}
$$

となる。さらにこれを用いて

$$
\begin{aligned}
\mathrm{EL}(\mathrm{f_{b1}}) &= \mathrm{EL}(\mathrm{f_{c1}}) \\
&= \max(\mathrm{CL}(\mathrm{f_{d1}},\mathrm{f_{b1}}), \mathrm{CL}(\mathrm{f_{d2}},\mathrm{f_{b1}})) \\
&= \max(1, p_g/p_f) = 1 \\
\mathrm{EL}(\mathrm{f_{d1}}) &= \mathrm{EL}(\mathrm{f_{d2}}) \\
&= \max(\mathrm{CL}(\mathrm{f_{b1}},\mathrm{f_{d1}}), \mathrm{CL}(\mathrm{f_{c1}},\mathrm{f_{d1}})) \\
&= \max(p_b/p_c, 1) = 1
\end{aligned}
$$

を得るから、先の仮定が正しいことがわかる。

一般に、単一化結線で結ばれた束縛 α と β に対し、α から β への接続尤度 $\mathrm{CL}(\alpha,\beta)$ は $\mathrm{IL}(\alpha)/\mathrm{IL}(\gamma)$ である。α の第 1 引数が β の第 1 引数の原点 δ の要素であれば γ は ξ、β の第 1 引数が α の第 1 引数の原点 δ の要素であれば γ は η である。ここで ξ と η は単一化結線で結ばれた束縛であり、ξ の第 1 引数が δ の要素、η の第 1 引数が δ を原点とする項であるもののう

ち、$\mathrm{IL}(\xi)\mathrm{IL}(\eta)$ が最大となる組とする。ξ と η の選択は γ の選択を前提しないので、尤度の計算は収束する。ξ と η を含む説明が δ の要素を含む説明のうちで確率が最大のものであることに注意せよ。α が ξ または η のときは $\mathrm{CL}(\alpha,\beta)=1$ であり、β と単一化結線で繋がったいかなる束縛 α' に対しても $\mathrm{CL}(\alpha,\beta)\geq\mathrm{CL}(\alpha',\beta)$ だから、$\mathrm{EL}(\beta)=1$ となる。したがって、この最大確率の説明の中のすべての束縛の外部尤度は 1 となる。ちなみに、$\mathrm{IL}(\xi)=\mathrm{IL}(\eta)$ となることもわかる。

図5の場合のように、δ の要素に対する束縛と他の束縛の間の単一化結線が消去されていなければ、δ の要素と δ を原点とする各項を結ぶ各依存経路を含む説明が存在するから、上記のように、束縛の外部尤度はすべて 1 となる。

δ の要素に対する束縛と他の束縛との間の単一化結線が消去されると[7]、上記のようにして選択された γ が変更されることがある。そのような変更が図5の直後に生ずる。図5の f_{c1} と f_{d1} の間の単一化結線と f_{b1} と f_{d2} の間の単一化結線は a と b による束縛の間の矛盾を含むので消去され、外部尤度の計算は以下のようになる。

$$
\begin{aligned}
\mathrm{EL}(\mathsf{f}_{b1}) &= \mathrm{CL}(\mathsf{f}_{d1},\mathsf{f}_{b1}) \\
&= \mathrm{IL}(\mathsf{f}_{d1})/\mathrm{IL}(\mathsf{f}_{d1}) = 1 \\
\mathrm{EL}(\mathsf{f}_{c1}) &= \mathrm{CL}(\mathsf{f}_{d2},\mathsf{f}_{c1}) \\
&= \mathrm{IL}(\mathsf{f}_{d2})/\mathrm{IL}(\mathsf{f}_{d1}) = p_g/p_f \\
\mathrm{EL}(\mathsf{f}_{d1}) &= \mathrm{CL}(\mathsf{f}_{b1},\mathsf{f}_{d1}) \\
&= \mathrm{IL}(\mathsf{f}_{b1})/\mathrm{IL}(\mathsf{f}_{c1}) = p_b/p_c \\
\mathrm{EL}(\mathsf{f}_{d2}) &= \mathrm{CL}(\mathsf{f}_{c1},\mathsf{f}_{d2}) \\
&= \mathrm{IL}(\mathsf{f}_{c1})/\mathrm{IL}(\mathsf{f}_{c1}) = 1
\end{aligned}
$$

ここで $p_b p_f > p_c p_g$ とすれば、以下を得る。

$$
\begin{aligned}
\mathrm{EL}(\mathsf{p}_{a1}) &= \max(p_b\mathrm{IL}(\mathsf{p}_{b1}), p_c\mathrm{IL}(\mathsf{p}_{c1})) \\
&= \max(p_b\mathrm{EL}(\mathsf{f}_{b1}), p_c\mathrm{EL}(\mathsf{f}_{c1})) \\
&= \max(p_b, p_c p_g/p_f) = p_b
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{EL}(\mathsf{q}_{a1}) &= \max(p_d\mathrm{IL}(\mathsf{p}_{d1}), p_d\mathrm{IL}(\mathsf{p}_{d2})) \\
&= p_d\max(\mathrm{EL}(\mathsf{f}_{d1})\mathrm{EL}(\mathsf{r}_{d1}), \\
&\qquad \mathrm{EL}(\mathsf{f}_{d2})\mathrm{EL}(\mathsf{r}_{d2})) \\
&= p_d\max((p_b/p_c)p_f, p_g) \\
&= p_b p_d p_f/p_c \\
\mathrm{IL}(\mathsf{f}_{b1}) &= p_b\mathrm{IL}(\mathsf{p}_{a1}) = p_b\mathrm{EL}(\mathsf{q}_{a1}) \\
&= p_b^2 p_d p_f/p_c \\
\mathrm{IL}(\mathsf{f}_{c1}) &= p_c\mathrm{IL}(\mathsf{p}_{a1}) = p_c\mathrm{EL}(\mathsf{q}_{a1}) \\
&= p_b p_d p_f \\
\mathrm{IL}(\mathsf{f}_{d1}) &= p_d\mathrm{IL}(\mathsf{q}_{a1})p_f = p_d\mathrm{EL}(\mathsf{p}_{a1})p_f \\
&= p_b p_d p_f \\
\mathrm{IL}(\mathsf{f}_{d2}) &= p_d\mathrm{IL}(\mathsf{q}_{a1})p_g = p_d\mathrm{EL}(\mathsf{p}_{a1})p_g \\
&= p_b p_d p_g
\end{aligned}
$$

したがって、

$$
\begin{aligned}
\mathrm{IL}(\mathsf{f}_{b1})\mathrm{IL}(\mathsf{f}_{d1}) &= (p_b p_d p_f)^2 p_b/p_c \\
\mathrm{IL}(\mathsf{f}_{c1})\mathrm{IL}(\mathsf{f}_{d2}) &= (p_b p_d p_f)^2 p_g/p_f
\end{aligned}
$$

となる。これより

$$
\mathrm{IL}(\mathsf{f}_{b1})\mathrm{IL}(\mathsf{f}_{d1}) > \mathrm{IL}(\mathsf{f}_{c1})\mathrm{IL}(\mathsf{f}_{d2})
$$

がわかるので、$\mathrm{CL}(\mathsf{f}_{b1},\mathsf{f}_{d1})$ と $\mathrm{CL}(\mathsf{f}_{b1},\mathsf{f}_{d1})$ の計算において分母が $\mathrm{IL}(\mathsf{f}_{c1})$ から $\mathrm{IL}(\mathsf{f}_{b1})$ に修正される。こうしてプログラムの状態は図6のようになり、ここでの尤度の計算は以下のようになる[8]。

$$
\begin{aligned}
\mathrm{EL}(\mathsf{f}_{b1}) &= \mathrm{CL}(\mathsf{f}_{d1},\mathsf{f}_{b1}) \\
&= \mathrm{IL}(\mathsf{f}_{d1})/\mathrm{IL}(\mathsf{f}_{d1}) = 1 \\
\mathrm{EL}(\mathsf{f}_{c1}) &= \mathrm{CL}(\mathsf{f}_{d2},\mathsf{f}_{c1}) \\
&= \mathrm{IL}(\mathsf{f}_{d2})/\mathrm{IL}(\mathsf{f}_{d1}) \\
&= (p_b p_d p_g)/(p_b p_d p_f) = p_g/p_f \\
\mathrm{EL}(\mathsf{f}_{d1}) &= \mathrm{CL}(\mathsf{f}_{b1},\mathsf{f}_{d1}) \\
&= \mathrm{IL}(\mathsf{f}_{b1})/\mathrm{IL}(\mathsf{f}_{b1}) = 1 \\
\mathrm{EL}(\mathsf{f}_{d2}) &= \mathrm{CL}(\mathsf{f}_{c1},\mathsf{f}_{d2}) \\
&= \mathrm{IL}(\mathsf{f}_{c1})/\mathrm{IL}(\mathsf{f}_{b1})
\end{aligned}
$$

[7] そのような単一化結線の消去に伴って他の単一化結線を消去すべきことがあるが、それは δ が 2 引数以上の要素式の引数であった場合だけであり、目下の例題には関係ないので、本稿では考えない。

[8] 一般には、接続尤度の計算における情報の流れが巡回しないように注意する必要があるが、この例題には無関係なので詳細は省略する。

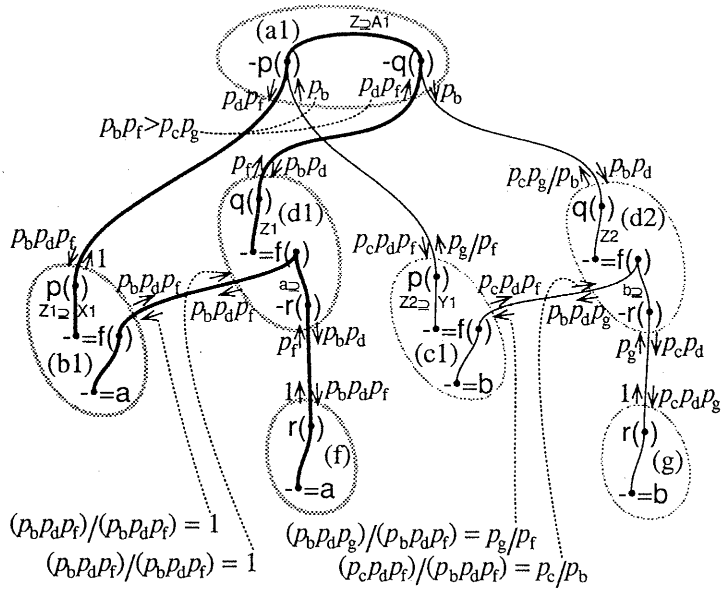

図 6: f_{b1} と f_{d2} および f_{c1} と f_{d1} の間の単一化結線を消去した結果

$$
\begin{aligned}
&= (p_c p_d p_f)/(p_b p_d p_f) = p_c/p_b \\
\mathrm{EL}(\mathsf{p}_{a1}) &= \max(p_b \mathrm{IL}(\mathsf{p}_{b1}), p_c \mathrm{IL}(\mathsf{p}_{c1})) \\
&= \max(p_b \mathrm{EL}(\mathsf{f}_{b1}), p_c \mathrm{EL}(\mathsf{f}_{c1})) \\
&= \max(p_b, p_c p_g / p_f) = p_b \\
\mathrm{EL}(\mathsf{q}_{a1}) &= \max(p_d \mathrm{IL}(\mathsf{p}_{d1}), p_d \mathrm{IL}(\mathsf{p}_{d2})) \\
&= p_d \max(\mathrm{EL}(\mathsf{f}_{d1})\mathrm{EL}(\mathsf{r}_{d1}), \\
&\qquad \mathrm{EL}(\mathsf{f}_{d2})\mathrm{EL}(\mathsf{r}_{d2})) \\
&= p_d \max(p_f, (p_c/p_b) p_g) = p_d p_f
\end{aligned}
$$

リテラルの説明とは、その要素式のみからなる仮説の説明のことだとしよう。図5では、p_{a1} の説明と q_{a1} の説明との任意の組み合せを含むような全体の説明が存在する、という意味において、p_{a1} と q_{a1} とは確率的に独立である。しかし図6では、⟨b1, d1, f⟩ と ⟨c1, d2, g⟩ という 2 つの説明しかないので、p_{a1} と q_{a1} の説明の間には確率的依存関係がある。直接間接に呼び出す関係にない 2 つのリテラルの間にこうした依存関係がある場合でも、われわれの方法では、これら 2 つのリテラルの間の構造が多くの説明の間で共有されうる。共有される構造はこの例では (a1)[9] というひとつの節であるが、一般にはさらに大きな構造が共有されることがある。HMM や SCFG はこうした依存関係が生じない場合を扱っているが、Horn 節プログラムに議論を一般化した場合、効率的な計算のためには、このような依存関係の下で構造共有を行なう必要がある。

6 おわりに

HMM や SCFG の一般化として確率的 Horn 節プログラムを考え、これに関して効率的に最尤推定を行なう方法を論じた。この方法は、たとえば文脈自由文法の統語解析を既存のアルゴリズムと同等の効率で行なう一般的な制約処理手続きと、それによって推持される構造共有、さらにそこから生ずる確率的依存関係の下で、プログラムの各部分の尤度 (その部分を含む最大確率の説明の確率) を計算することにより、一般的な意味での A* 探索を行なうものである。本稿では 5 節の例の理解に不要な議論を省略したが、ここで省略

[9] (a1) は説明には明示的に含まれないが、実質的には共有されていると考えられる。

した詳細については近いうちに報告したい。たとえば、再帰的なプログラムを再帰的なプログラムに変換することも畳込みによって可能である。

可能な包摂化を尽くせば、すべての説明が解になっており、しかもどの解も失われていないので、プログラムは正しい解の確率分布を表現している。もちろん計算が終了しないこともあるが、説明の確率の総和は計算に伴って単調に減少し、しかも0より小さくはならないから、必ず収束する。また、未処理の説明のうちで尤度が最大のものの処理を優先することにより、正しい解の確率分布への収束が保証される。そのような計算は、基本確率に関するパラメタ学習に用いることができる。パラメタ学習を行なうには、各パラメタ p に関する解の確率の総和 P の偏微分 $\frac{\partial P}{\partial p}$ を用いた最急降下法を用いればよいと考えられるが、$\frac{\partial P}{\partial p}$ を求めるための一般化された逆伝搬 (Pineda, 1988) も上記の尤度の計算を大きく越えない手間で実行可能である。

本稿で述べた方法は、 HMM や SCFG や STAG など、確率が基本確率の積和の形で表わされるいろいろな枠組のうちで最も一般的なものである。これら既存の方法を用いた従来の処理方法は、各作業に特有の具体的な処理手順を用いていたために、たとえば音声認識と言語処理を有機的に統合できなかったが、われわれの方法では作業に固有の処理手順を一切用いず、一様な処理メカニズムから既存のアルゴリズムの挙動が創発するから、多様な情報処理過程を統合し、さまざまな種類の情報の間のより緊密な相互作用を実現できる可能性がある。

また、この方法は、 Horn 節プログラムという一般的な記述力を持つ記号体系を用いているので、 HMM や SCFG に比べてはるかに広い射程を持つ。たとえば、 Bayesian ネットワーク (Pearl, 1988) やコスト付アブダクション (Hobbs et al., 1993) などの推論メカニズムをこの枠組の中で実現することができるだろう。また、上述のような方法によるパラメタ学習だけではなく、例に基づく学習や帰納学習による記号的な知識の獲得と組合わせることもできるので、記号的な知識の学習と統計的な情報処理との相互作用を探究する手段となろう。さらに、確率に加えて効用 (utility) の概念を導入し、期待効用 U を用いて、 $\frac{\partial U}{\partial p}$ が大きい基本確率 p を持つ節に含まれる行為を優先的に起動するという行為の理論を考えることができる。効用をどのような形で定義するか、あるいはその場合に情報処理の制御と行為の制御がどのような関係を持つかなどに関して、考察を進めているところである。

文　献

Hasida, K. (1994). Emergent Parsing and Generation with Generalized Chart. *Proceedings of the Fifteenth International Conference on Computational Linguistics.*

Hobbs, J. R., Stickel, M. E., Appelt, D. E., & Martin, P. (1993). Interpretation as Abduction. *Artificial Intelligence*, **63** (1-2), 69–142.

Lloyd, J. W. (1984). *Foundations of Logic Programming.* Springer.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Los Altos, CA: Morgan Kaufmann.

Pineda, F. J. (1988). Generalization of Backpropagation to Recurrent and Higher Order Neural Networks. In D. Z. Anderson (Ed.), *Neural Information Processing Systems*, pp. 602–611.

Shabes, Y. (1992). Stochastic Lexicalized Tree-Adjoining Grammars. In C. Boitet (Ed.), *Proceedings of the Fourteenth International Conference on Computational Linguistics*, pp. 426–432. Nantes:

Tamaki, H. & Sato, T. (1983). Unfold/Fold Transformation of Logic Programs. *The 2nd International Conference on Logic Programming*, pp. 127–138.

Xuang, X., Ariki, Y., & Jack, M. A. (1990). *Hidden Markov Models for Speech Recognition.* Edinburgh University Press.