学位論文

形式文法の確率的一般性と応用に関する研究

東京大学大学院 工学系研究科 電子工学専攻 近山研究室

柴田 剛志

目 次

1　概要

2　準備

2.1　マルコフ決定過程と強化学習

この章では強化学習の基礎となるマルコフ決定過程と、強化学習の代表として Q 学習について述べる。本論文ではマルコフ決定過程から Q 学習にいたるのと同様の道筋で決定過程化した単純文法からその下での Q 学習を定義する。

2.2　マルコフ決定過程

確率変数の列を $X_1, X_2, \cdots$ とし、それらのとりうる値の有限集合を $S = \{s_1, \cdots, s_{|S|}\}$ とする。X_t の実現値 $x_t \in S$ を時刻 t における状態と呼ぶ。

有限状態のマルコフ連鎖とは、X_{t+1} の確率分布が X_t のみに依存する確率過程で、時刻 t での状態を i_t として、

$$P(X_{t+1} = x_{t+1} | X_1 = x_1, \cdots, X_t = x_t) = P(X_{t+1} = x_{t+1} | X_t = x_t)$$

が満たされる。斉時マルコフ連鎖とは、時間に対して一様なマルコフ連鎖で、任意の t で $P(X_{t+1} = s_j | X_t = s_i) = P(X_2 = s_j | X_1 = s_i)$ が成り立つもので、$P(X_2 = s_j | X_1 = s_i)$ を遷移確率とよび、P_{ij} であらわす。

有限状態かつ有限行動のマルコフ決定過程 (Markov Decision Process : 以下 MDP) とは、有限状態のマルコフ連鎖に「行動」と「報酬」の概念を付け加えたもので、次の 4 つ組 $\langle S, U, C, P \rangle$ を持つ確率過程である。

- S : 状態の有限集合
- U : 行動の有限集合
- C : 報酬 $S \times S \times U \to \mathbb{R}$
- P : 遷移確率 $S \times S \times U \to [0, 1]$、任意の $s_i \in S$ および $u \in U$ に対して、
$$\sum_{s_j \in S} P(s_i, s_j, u) = 1$$

S の元をとりうる値としてもつ確率変数の列を $X_1, X_2, \cdots$ とし、U の元をとりうる値としてもつ確率変数の列を $Y_1, Y_2, \cdots$ とする。Y_t の実現値 $u_t \in U$ を時刻 t における行動と呼ぶ。

$$\begin{aligned} &P(X_{t+1} = x_{t+1} | X_1 = x_1, Y_1 = u_1, \cdots, X_t = x_t, Y_t = u_t) \\ &\qquad = P(X_2 = x_{t+1} | X_1 = x_t, Y_1 = u_t) = P(x_t, x_{t+1}, u_t) \end{aligned}$$

をみたす。

以下 $s_i, s_j \in S$ 、$u \in U$ に対して、$C(s_i, s_j, u)$ を $C_{ij}(u)$、$P(s_i, s_j, u)$ を $P_{ij}(u)$ と書くことにする。

各状態と時刻に対し、そこでの行動決定すること、すなわち、$\mathcal{M} = \{S \to U\}$ として、列 $(\mu_1, \mu_2, \cdots)$,$\mu_t \in \mathcal{M}$ を方策という。方策に対するマルコフ連鎖を $Y_t = \mu_t(X_t)$ とすることで決めることができる。

$$P(X_{t+1} = s_j | X_1 = x_1, \cdots, X_t = s_i) = P(X_{t+1} = s_j | X_t = s_i) = P_{ij}(\mu_t(s_i))$$

である。成分が $P_{ij}(\mu_t(s_i))$ の $|S| \times |S|$ 行列を μ_t に関する遷移確率行列といい、$P(\mu_t)$ であらわす。また、任意の t で $\mu_t = \mu_1$ のとき、静的方策という。このとき、$X_1, X_2, \cdots$ は斉時マルコフ連鎖になる。状態 s_i の方策 μ における残り d ステップの報酬期待値とは、方策によってきまるマルコフ連鎖に対する次の期待値

$$V_i(\mu, d) = E_\mu\Big[\sum_{t=1}^{d} \gamma^{t-1} C(x_t, x_{t+1}, \mu_t(x_t))\Big| x_1 = s_i\Big]$$

であり、$V_i(\mu) = \liminf_{d\to\infty} V_i(\mu, d)$ を価値という。$\gamma \in [0, 1]$ は定数で、割引率とよばれる。$\boldsymbol{V}(\mu, d)$ を各成分が $V_i(\mu, d)$ のベクトル、$\boldsymbol{c}(\mu_t)$ を 各成分が $\sum_{j=1}^{|S|} P_{ij}(\mu_t) C_{ij}(\mu_t)$ のベクトルとすると、

$$\boldsymbol{V}(\mu, d) = \sum_{t=1}^{d} \gamma^{t-1} P(\mu_1) \cdots P(\mu_{t-1}) \boldsymbol{c}(\mu_t)$$

と書ける。また、$T_{\mu_t} : \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}$ を $T_{\mu_t}(\boldsymbol{x}) = \boldsymbol{c}(\mu_t) + \gamma P(\mu_t)\boldsymbol{x}$ で定義すると、

$$\boldsymbol{V}(\mu, d) = T_{\mu_1} \cdots T_{\mu_d}(\boldsymbol{0})$$

がなりたつ。

2.2.1 最適方策

状態 s_i における最適価値は、$V_i^* = \sup_{\mu \in \{\mathbb{N} \to \mathcal{M}\}} V_i(\mu)$ で定義される。また、$T : \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}$ を $T(\boldsymbol{x}) = \max_{\mu \in \mathcal{M}} T_\mu(\boldsymbol{x})$ とおく。全ての状態における最適価値を価値としてもつ方策を最適方策というが、最適方策が存在するかどうかについては、次の命題が成り立つ [**?**] [**?**]。命題 割引率 γ が $\gamma < 1$ を満たすとする。このとき、

1. $\boldsymbol{V}^*$ は $\mathbb{R}^{|S|}$ における T の唯一の不動点である。
2. 任意の $\boldsymbol{x} \in \mathbb{R}^{|S|}$ および $\mu \in \mathcal{M}$ について、$\lim_{t\to\infty} T^t(\boldsymbol{x}) = \boldsymbol{V}^*$ および $\lim_{t\to\infty} T_\mu^t(\boldsymbol{x}) = \boldsymbol{V}(\mu)$ が成り立つ。
3. 静的方策 $(\mu^*, \mu^*, \cdots)$ が最適方策であることの必要十分条件は $T_{\mu^*}(V^*) = V^*$ である。

$T(\boldsymbol{x}) = \boldsymbol{x}$ をベルマン最適方程式という。

2.2.2 エピソード型

MDP がエピソード型であるとは、ある $s_1 \in S$ に対して、$P(s_1, s_1, u) = 1$、$C(s_1, s_1, u) = 0$ が任意の u で成立することである。このような s_1 のことを終状態という。また、ある $\mu \in \mathcal{M}$ または静的方策 $\mu_s = (\mu, \mu, \cdots)$ が適正であるとは、任意の $s_i \in S$ について $\lim_{t\to\infty}[P(\mu)^t]_{i1} = 1$ である。すなわち、どんな状態からスタートしても終状態に収束するということである。

最適方策の存在について、次の命題が成り立つ [?] [?]。命題 エピソード型の MDP を考える。適正な静的方策が存在し、適正でない全ての静的方策には価値が $-\infty$ となる状態が存在するとする。このとき、

1. $\boldsymbol{V}^*$ は $\{\boldsymbol{x} \in \mathbb{R}^{|S|} | x_1 = 0\}$ における T の唯一の不動点である。
2. 任意の $\boldsymbol{x} \in \mathbb{R}^{|S|}$ について、$\lim_{t\to\infty} T^t(\boldsymbol{x}) = \boldsymbol{V}^*$ が成り立つ。
3. 静的方策 $(\mu^*, \mu^*, \cdots)$ が最適方策であることの必要十分条件は $T^*_{\mu}(\boldsymbol{V}^*) = \boldsymbol{V}^*$ である。

2.3 Q 学習

強化学習の目的は、形式的には、マルコフ決定過程において、時刻 t までの状態と報酬の履歴を入力とし、時刻 t での行動を出力として、オンラインで最適方策を求めることである。

最適行動価値 $Q^* : |S| \times U \to \mathbb{R}$ は、$u = \mu(s_i)$ として、$\boldsymbol{Q}^*(i, u) = [T_\mu(\boldsymbol{V}^*)]_i$ で定義される。ベルマン最適方程式はつぎと同値である。

$$Q^*(i, u) = c_i(u) + \sum_{j=1}^{|S|} P_{ij}(u) \min_{v \in U} Q^*(j, v)$$

Q 学習は、$\boldsymbol{Q}(\mu^*)$ を次のような繰り返しで求めるオンライン学習である。$\boldsymbol{Q}(\mu^*)$ が求まったら、最適方策 $\mu^*(i)$ は $\arg\max_{u \in U(i)} Q_{i,u}(\mu^*)$ で与えられる。

X_t, Y_t の実現値を s_{i_t},u_t として、

$$Q(i_t, u_t) := (1 - \alpha_t(i_t, u_t))Q(i_t, u_t) + \alpha_t(i_t, u_t)\left(C_{i_t i_{t+1}}(u_t) + \gamma \max_{v \in U} Q(i_{t+1}, v)\right) \tag{1}$$

ただし、α_t は学習のステップサイズパラメータをあらわす確率変数であり、$P(\alpha_t(i, u) = 0 | (X_t, Y_t) \neq (s_i, u)) = 1$ および、$\sum_{t=0}^{\infty} \alpha_t(i_t, u_t)^2 < \infty$ および、$\sum_{t=0}^{\infty} \alpha_t(i_t, u_t) = \infty$ を満たすものとする。

これを満たす例としては、たとえば、$\forall s_i \in S, u \in U$ で、$\sum_{t=0}^{\infty} P(u_t = u, i_t = i) = \infty$ が満たされるときに、時刻 t までに s_i で u を選んだ回数を $n_t(i, u)$ として、$\alpha_t(i, u) = \frac{1}{n_t(i,u)}$ ととったときなどである。

次の定理が成り立つ [4]。定理 Q 学習の更新式 1 を繰り返せば次の条件うちどれか一つを満たせば最適行動価値に概収束する。

1. $\gamma < 1$ のとき。
2. エピソード型で全ての方策が適正であるとき。
3. 前節の仮定および $Q_t(i,u)$ が有界である確率が 1 のとき。

2.4 正例からの学習

2.5 記号の定義等

空集合を $\varnothing$、空列を ε、自然数全体、整数全体、実数全体をそれぞれ $\mathbb{N}, \mathbb{Z}, \mathbb{R}$ とする。

文脈自由文法は 4 つの組み $\langle N, \Sigma, R, S\rangle$ で定義される。

- N：非終端記号の有限集合
- Σ：終端記号の有限集合
- R：生成規則の有限集合
- S：開始記号,$S \in N$

また、文脈自由文法 G による言語を $L(G)$ であらわす。

簡潔に表現するため、特に定義なく出てきた文字の場合、次のものをあらわしていることとする。

- a, b, c, d：終端記号
- w, x, y, z：終端記号の有限列
- A, B, C, D：非終端記号
- $\alpha, \beta, \gamma, \delta$：非終端記号の有限列
- Γ, Δ：非終端記号または終端記号の有限列

生成規則を $A \to a\alpha$ のようにあらわし、1 回の最左導出を $\alpha \Rightarrow a\beta$ のようにあらわし、0 回を含む任意回の最左導出を $\alpha \overset{*}{\Rightarrow} x\alpha$ のようにあらわすとする。

定義 文脈自由文法 $G = \langle \Sigma, R, N, S\rangle$ と $G' = \langle \Sigma', R', N', S'\rangle$ が同型であるとは、以下が成り立つことを言う。

1. $\Sigma = \Sigma'$
2. 全単射 $h : N \to N'$ が存在する。
3. $R' = \{h(A) \to h(\Gamma) \mid A \to \Gamma \in R\}$
4. $S' = h(S)$

G と G' が同型であることを $G \simeq G'$ であらわす。

定義 文脈自由文法 $G = \langle \Sigma, R, N, S\rangle$ が既約であるとは、全ての非終端記号 $A \in N$ に対して、$S \overset{*}{\Rightarrow} xAz \overset{*}{\Rightarrow} xyz$ なる導出が存在することである。

定義 文脈自由文法 $G = \langle N, \Sigma, R, S\rangle$ がグライバッハ標準形であるとは、全ての生成規則が $A \to a\alpha$ の形をしているものをいう。

また、グライバッハ標準形の文脈自由文法 $G = \langle N, \Sigma, R, S \rangle$ に対して、$\# : R \rightarrow \mathbb{Z}$ を

$$\#(r) = (r \in R \text{ に出てくる非終端記号の個数}) - 1$$

で定義する。

2.6 単純文法

定義 CFG $G = \langle N, \Sigma, R, S \rangle$ が、単純文法 (Simple Grammer：以下 SG) であるとは、以下の条件を満たすときをいう。

1. G はグライバッハ標準形である
2. 任意のルール $A \rightarrow a\alpha, B \rightarrow b\beta \in R$ について、$A = B$ かつ $a = b$ ならば $\alpha = \beta$

定義から、SG は、文字列のプリフィックスだけから、後続の非終端記号の列が一意に定まる性質を持つ文法であるといえる。

本論文では、SG のサブクラスのうち、次の 3 つをあつかう。

- 右固有単純文法 (Right-unique Simple Grammar：以下 RSG) のクラス
- 無重複単純文法 (Nonmultiplicative Simple Grammar：以下 NMSG) のクラス
- 無重複右固有単純文法 (Nonmultiplicative Right-unique Simple Grammar：以下 NMRSG) のクラス

定義 RSG は、上の SG の条件 (1) および次の条件を満たす文法である。

2. 任意のルール $A \rightarrow a\alpha, B \rightarrow b\beta \in R$ について、$a = b$ ならば $\alpha = \beta$

この条件は SG の条件 (2) を強くしたものであり、RSG のクラスは SG のクラスに含まれる。

NMSG および NMRSG については後述する。

2.7 正の例からの学習

本論文では、正の例からの学習についてのみ扱う。ここでいう学習とは、直感的に言うと、ある一連のデータを使って、そのデータの元となった文法と等価な文法を言い当てることである (図 1 参照)。正の例とは、ある文法を元にして、その文法によって導出される文の列で、かつその文法によって導出される文が全てが出てくる列である。

定義 文法 G に対する正の例 d とは、全射

$$d : \mathbb{N} \to L(G)$$

である。

G における正の例全体の集合を $D_G = \{d : \mathbb{N} \to L(G) | d \text{ は全射 } \}$ とおく。

定義 $\mathcal{G}$ をある文法クラスとする。アルゴリズム $A : \{(w_1, \cdots, w_n) | w_i \in \Sigma^*, n \in$
$mathbbN\} \to \mathcal{G}$ が正の例 d を入力として文法 G' に収束するとは、ある $n_0 \in \mathbb{N}$ があって、任意の $n \geq n_0$ にたいして、$A(d(1), \cdots, d(n)) = G'$ が成立することである。

定義 ある文法クラス $\mathcal{G}$ が学習可能とは、$\mathcal{G}$ に含まれる任意の文法 G に対して、G の任意の正の例を入力としたとき、$L(G') = L(G)$ となるような G' に収束するアルゴリズムが存在することである。

次のことが分かっている。定理(吉仲) RSG のクラスは学習可能である。さらに、ステップあたりの計算量は $O(|\Sigma|^6 \mathcal{R}^{|\Sigma|+7})$ 以下である。 ここで、$\mathcal{R}$ は、サンプル文字列の長さの総和である。定理(吉仲) NMSG のクラスは学習可能である。

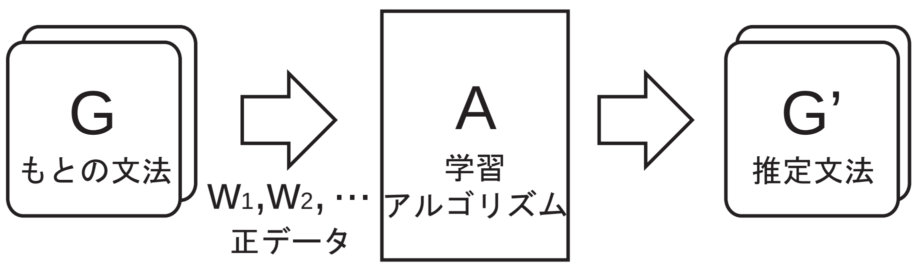

図 1: 正の例からの学習

2.8 RSG 固有の性質

定義 $f : \Sigma^* \to \mathbb{Z}$ が準同型とは次が成立つことである。

1. 任意の $x, y \in \Sigma^*$ に対して、$f(xy) = f(x) + f(y)$
2. $f(\varepsilon) = 0$

定義 既約な RSG $G = \langle N, \Sigma, R, S \rangle$ に対して、準同型写像 $\#_G : \Sigma^* \to \mathbb{Z}$ が、$A \to a\alpha \in R$ に対して、$\#_G(a) = |\alpha| - 1$ とすることで定義できる。$\#_G$ を G の形状(shape) とよぶ。

$\#_G(x)$ は文字列 x を導出することによる非終端記号の数の収支をあらわしている。すなわち、$\alpha \overset{*}{\Rightarrow} [G] x\beta$ ならば $|\alpha| + \#_G(x) = |\beta|$ が成り立つ。

定義 既約な RSG $G=\langle N,\Sigma,R,S\rangle$ に対して、写像 $\$_G:\Sigma^*\to\mathbb{Z}$ が次のように定義される。

1. $\$_G(\varepsilon)=0$
2. 任意の $x\in\Sigma^+$ に対して、$\$_G(x)=\max\{1-\#_G(x')|x'$は x の真プリフィックス $\}$

導出の途中で非終端記号の数が 0 個になったらそれ以上は導出できないので、x を導出するには最低 $1+(-\#_G(x'))$ 個の非終端記号が必要である。つまり、$\$_G(x)$ は文字列 x を導出するために最低限必要な非終端記号の数をあらわしている。すなわち、$\alpha\overset{*}{\Rightarrow}[G]x\beta$ ならば $|\alpha|\geq\$_G(x)$ が成り立つ。

以下、RSG が一つしかないときやどの文法についてか明らかなときは、下付き文字を省略して単に $\#$ や $\$$ と書くことにする。

定義 RSG $G=\langle N,\Sigma,R,S\rangle$ が標準形であるとは、G が既約かつ次が成立することである。任意の $A\to aA_0\cdots A_{\#(a)}\in R$ に対して、$i\neq j$ ならば $A_i\neq A_j$、かつ任意の $A\to aA_0\cdots A_{\#(a)},B\to bB_0\cdots B_{\#(b)}\in R$ に対して、$a\neq b$ ならば $\{A_0,\cdots,A_{\#(a)}\}\cap\{B_0,\cdots,B_{\#(b)}\}=\varnothing$。

全ての既約な RSG は標準形で書くができる。すなわち、任意の RSG G に対して、$L(G)=L(G')$ かつ $\#_G=\#_{G'}$ となる標準形である G' が存在する。なお、あとで確率化した文法を考えるが、その場合標準形に直すことで元の確率文法と潜在的な表現能力が異なってくる。

補題 $L\subset\Sigma^*$ とする。また、準同型写像 $\widetilde{\#}:\Sigma^*\to\mathbb{Z}$ を任意の $a\in\Sigma$ に対して $\widetilde{\#}(a)\geq-1$ であるものとする。このとき、任意の $x\in L$ について、

1. $\widetilde{\#}(x)=-1$
2. $\widetilde{\$}(x)=1$; ただし $\widetilde{\$}$ の定義は $\widetilde{\$}(\varepsilon)=1$、$\widetilde{\$}(x)=\max\{1-\widetilde{\#}(x')|x'$は x の真プリフィックス $\}$

が成り立つことと、$\#_G=\widetilde{\#}$ かつ $L\subset L(G)$ となる RSG G が存在することは同値である。

上記のような $\widetilde{\#}$ は L に対する "可能な形状 (possible shape)" と呼ばれている。つまり、L の全ての要素を導出できる RSG が存在することの必要十分条件は、可能な形状が存在することである。

2.9 RSG の学習

基本的に、学習アルゴリズムには順次新しい文が入ってきて、そのたびに G を更新してゆくという手順をとる。t 個目までの入力文を $L_t=\{w_1,w_2,\cdots,w_t\}$ とする。L_t を含むようなものを出力しつづけて、何かある文法 G_* に収束するアルゴリズムがあったとする。このとき、$\lim_{t\to\infty}L_t$ は正の例だから、学習対象元の文法を G_0 とすると、$L(G_0)=\lim_{t\to\infty}L_t\subset L(G_*)$ となる。したがって、L_t を含むような RSG のうちで極小の RSG を出力することができれ

ばアルゴリズムは学習対象の文法と等価な文法を学習することができる。なぜなら、$L(G_0) \subsetneq L(G_*)$ とすると、$L_t \subset L(G_0) \subsetneq L(G_*)$ より、G_* は極小でなくなるからである。

2.9.1 可能な形状の列挙

t 個目までの文 $\{w_1, w_2, \cdots, w_t\}$ が入力されたときを考える。このときの可能な形状を次のようにして計算する。

$\Sigma_t = \{a \in \Sigma \mid a$ は $w_1, w_2, \cdots, w_t$の中に出てくる $\} = \{a_1, a_2, \cdots, a_{|\Sigma_t|}\}$ とする。$\boldsymbol{v} : \Sigma^* \to \mathbb{Z}^{|\Sigma_t|}$ を、各成分を

$$v_i(w) = w \text{ の中に出てくる } a_i \text{の個数}$$

として定義する。

このとき、$L = \{w_1, w_2, \cdots, w_t\}$、$\boldsymbol{s} = (\widetilde{\#}(a_1), \widetilde{\#}(a_2), \cdots, \widetilde{\#}(a_{|\Sigma_t|}))$ とおくと、補題 2.8 の条件 2 は、

$$\begin{pmatrix} \boldsymbol{v}(w_1) \\ \boldsymbol{v}(w_2) \\ \vdots \\ \boldsymbol{v}(w_t) \end{pmatrix} \boldsymbol{s}^T = \begin{pmatrix} -1 \\ -1 \\ \vdots \\ -1 \end{pmatrix}$$

と書ける。したがって、まず上記方程式用いて補題 2.8 の条件 1 を満たす形状を全て列挙した後、補題 2.8 の条件 2 を満たさないものを削除すればよい。$(|\Sigma_t|, t)$ 行列 M_t を $M_t = \left(\boldsymbol{v}^T(w_1), \boldsymbol{v}^T(w_2), \cdots, \boldsymbol{v}^T(w_t)\right)^T$ とおくと、自由な変数の個数は $\dim \ker(M_t)$ となる。一方で、任意の $a \in \Sigma_t$ について、$-1 \leq \widetilde{\#}(a) < \max\{|w_1|, |w_2|, \cdots, |w_t|\}$ となるので、可能な形状の数は $(\max\{|w_1|, |w_2|, \cdots, |w_t|\} + 1)^{\dim \ker(M_t)}$ 以下になる。

したがって、$\dim \ker(M_*) = \lim_{t \to \infty} \dim \ker(M_t)$ として、$\dim \ker(M_*) < k$ となるような k がわかっていれば、$\dim \ker(M_t)$ をチェックして k 未満になるときのみ形状の候補を列挙することにすれば、大幅に計算量を削減できる。

2.9.2 各形状ごとの最小の RSG の決定と極小の RSG の出力

補題 $L \subset \Sigma^*$ とする。また、$\widetilde{\#}$ を L の可能な形状とする。このとき、次を満たす標準形のRSG G_0 が存在する。特に L が有限集合のとき、図3 のアルゴリズムで計算できる。

1. $\widetilde{\#} = \#_{G_0}$ かつ $L \subset L(G_0)$
2. $\widetilde{\#} = \#_G$ を満たす任意の RSG G に対して、$L \subset L(G)$ ならば $L(G_0) \subset L(G)$

Input $\{w_1, w_2, \cdots, w_t\}, w_i \in \Sigma^*$, $k \in \mathbb{Z}$
  if($\dim\ker(M_t) \geq k$){
    output 次元が多すぎる;
  }
  $\mathcal{S} := \varnothing$;
  $\mathcal{S}_F := \{(s_1, \cdots, s_k) | s_i$は自由$, -1 \leq s_i < \max\{|w_1|, |w_2|, \cdots, |w_t|\}\}$;
  for(each $(s_1, \cdots, s_k) \in \mathcal{S}_F$){
    if($M_t\boldsymbol{s} = -\mathbf{1}$ の解の全ての成分が -1 以上の整数である
        かつ、全ての w_i で $\widetilde{\$}(w_i) = 1$ である){
      $\mathcal{S} := \mathcal{S} \cup \{\boldsymbol{s}\}$;
    }
  }
  output S and halt

図 2: 可能な形状を列挙するアルゴリズム

Input $\{w_1, w_2, \cdots, w_t\}, w_i \in \Sigma^*$ および可能な形状 $\widetilde{\#}$
  $N := \{S\} \cup \{A_{a,i} | a \in \Sigma, 0 \leq i \leq \widetilde{\#}(a)\}$;
  $R = \varnothing$;
  for(each w_i){
    for($n := 0$; $n < |w|$; $n := n + 1$){
      $S \stackrel{*}{\Rightarrow} [R]xA\alpha$、$|x| = n$、$w_i = xay$ として、
        $R := R \cup \{A \to aA_{a,0}A_{a,1} \cdots A_{a,\widetilde{\#}(a)}\}$;
    }
  }
  output $\langle N, \Sigma_t, R, S \rangle$ and halt

図 3: ある形状に対して最小の RSG を求めるアルゴリズム

今，$\{w_1, w_2, \cdots, w_t\}$ を含むような極小の RSG を計算したいわけだから、可能な形状の各候補について、補題 2.9.2 の G_0 を計算して、さらにそれらの中で極小のものを求めればよい。$\{w_1, w_2, \cdots, w_t\}$ に対する可能な形状の最小の RSG を $\{G_1, \cdots, G_m\}$ とすると、RSG の包含関係判定の次の定理がなりたつ。定理(吉仲) G, G' をRSG とする。$L(G) \subset L(G')$ かどうかは、$O(|\Sigma|^6\mathcal{R}^8)$ 以下の計算量で判定できる。 実際のアルゴリズムについては省略するが、経験的には上記のオーダーはあくまで上限であってかなりはやく包含関係を判定することができる。包含関係を判定できるので、学習アルゴリズムは $G_1, \cdots, G_m$ の中から極小のものをどれか一つ選んで出力すればよい(図 4 参照)。

Input 文法の候補 $\mathcal{G} = \{G_1, G_2, \cdots, G_m\}$
  $G_0 := G_1$;
  for(each $G \in \mathcal{G}$){
    if($G \subset G_0$){
      $G_0 := G$;
    }
  }
  output G_0 and halt

図 4: 候補の中から極小の RSG を出力するアルゴリズム

2.9.3 RSG 学習の例

学習対象の RSG として、$G = \langle N, \Sigma, R, S \rangle$

$$R = \{\ S \to aA_0A_1A_2,\ S \to bB_0B_1,\ \{A_0, B_0, E_0\} \to cC_0,\ \{A_2, C_0\} \to d,\ C_0 \to eE_0E_1E_2,\ E_0 \to f,\\ A_1 \to g,\ B_1 \to hH_0,\ \{A_2, H_0\} \to i,\ E_2 \to j\ \}$$

とする。実験のために導出をランダムに行い正の例を生成しそれを $\dim\ker(M_t)$ の上限を 5 として学習させてみた。生成した正の例はたとえば次のようになる。

$$W = \{bcefijhi, bcecdijhi, bcdhi, bcdhi, acefijgi, bcdhi, acececefijijijgi, acdgd, acdgi, bcdhi,\\ acecefijijgi, acecefijijgi, bcdhi, bcefijhi, acecdijgd, bcefijhi, bcdhi, bcefijhi, bcecefijijhi,\\ acefijgi, bcefijhi, bcdhi, acdgi, acdgd, bcdhi, acdgd, acecececdijijijgd, \cdots\}$$

この正の例の t 番目まで入力を横軸として、$\dim\ker(M_t)$、$|\Sigma_t|$、および $\{w_1, w_2, \cdots, w_t\}$ を含む文法の候補の数の遷移を図 5 にしめす。

100 個の正の例を入力した結果、学習していた文法 G_* は、

$$R_* = \{\ S \to aA_0A_1,\ S \to bB_0B_1,\ \{A_0, B_0, E_0\} \to cC_0,\ \{C_0, G_0\} \to d,\ C_0 \to eE_0E_1E_2,\ E_0 \to f,\\ A_1 \to gG_0,\ B_1 \to hH_0,\ \{A_2, H_0\} \to i,\ E_2 \to j\ \}$$

であった。$L(G) = L(G_*)$ が成立している。

実際にもっとも時間のかかるのは可能な形状の列挙であり、定理 2.7 において計算量のオーダーが $|\Sigma|$ に関して指数時間であるのは、可能な形状の個数が原因である。$\dim\ker(M_t)$ の上限を決めて、それ以上のときは計算しないようにすることで、$|\Sigma|$ に関して多項式時間で計算できる。

3 決定過程化した文法と強化学習

3.1 確率文脈自由文法

$G = \langle N, \Sigma, R, S \rangle$ をグライバッハ標準形の CFG とする。これに、生成規則の選択確率 $P : R \to [0, 1]$ を加えたものを確率文脈自由文法といい、G_P

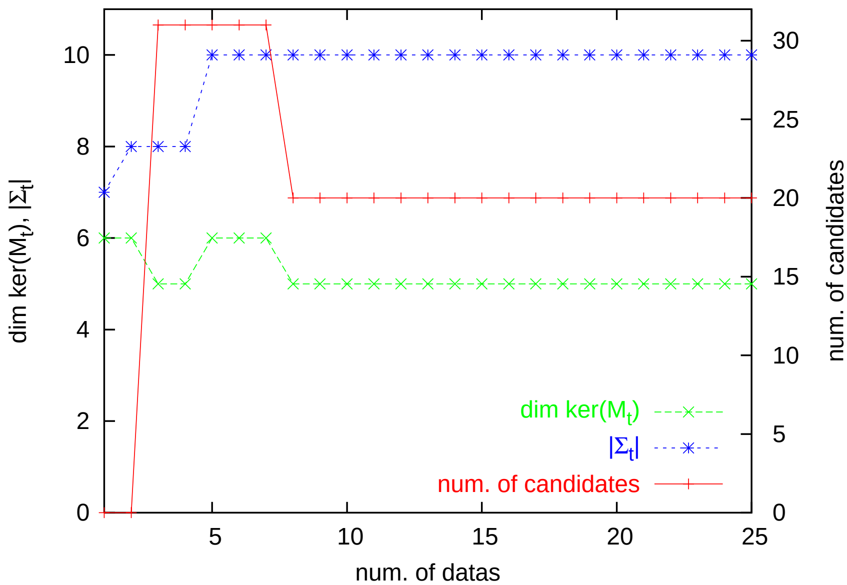

図 5: 正の例による RSG 学習過程の例

であらわす。すなわち、$G_P = \langle N, \Sigma, R, S, P\rangle$。ただし、任意の $A \in N$ で、$\sum_{r\in R_A} P(r) = 1$ とする。

離散確率変数列 $X_1, X_2, \cdots$ を、終端記号または非終端記号の有限列すなわち $(N \cup \Sigma)^*$ の元を値としてもつ斉時マルコフ連鎖とする。$(N \cup \Sigma)^*$ が状態となるので状態数は可算無限個である。また、各 X_t を最左導出で t 回生成規則を適用したものをあらわすものとする。すなわち、$X_1, \cdots, X_t$ の実現値を $\Gamma_1, \cdots, \Gamma_t$ とすると、

$$P(X_t = \Gamma_t | X_1 = \Gamma_1, \cdots, X_{t-1} = \Gamma_{t-1}) = P(X_2 = \Gamma_t | X_1 = \Gamma_{t-1})$$

をみたすとする。ここで、定義より $\Gamma_{t-1} \Rightarrow \Gamma_t$ が成り立って最左導出でなければならないから、

$$\delta(\Gamma_{t-1}, \Gamma_t) = \begin{cases} 1 & : \Gamma_{t-1} \Rightarrow \Gamma_t \text{が真で最左導出とき、または} \Gamma_t = \Gamma_{t-1} \in \Sigma^* \text{のとき} \\ 0 & : \text{上記以外} \end{cases}$$

として、最左導出に適用された生成規則を r とすると、

$$P(X_2 = \Gamma_t | X_1 = \Gamma_{t-1}) = \delta(\Gamma_{t-1}, \Gamma_t) P(r)$$

である。ただし、$\Gamma_{t-1} \in \Sigma^*$ のときは、生成規則が適用されないから、$P(r) = 1$ とする。$X_1 = S$ としたものを G_P から定まるマルコフ連鎖という。また、$w \in \Sigma^*$ に対して、$P(w) = P(\lim_{t\to\infty} X_t = w)$ を文 w の生成確率とよ

ぶ。定義 確率文脈文法 G_P が整合的であるとは、G_P で定まるマルコフ連鎖 $X_1, X_2, \cdots$ が Σ^* の中に概収束することである。 系 確率文脈文法 G_P が整合的であることの必要十分条件は $\sum_{w \in L(G)} P(w) = 1$ である。

3.1.1 整合的でない確率文脈自由文法

導出の順序についてであるが、整合的であるときは前節のように最左導出で議論しても問題がない。整合的でない場合、最左導出では、永遠に到達しない非終端記号が存在しうる。これは、直感的に言うと、最左導出が深さ優先で導出しているからである。

このことを正確に議論するために、まず、導出が無限に続く場合も考慮して、導出木を以下のように定義する。定義 G によって生成される導出木とは、各ノードに $N \cup \Sigma$ の元がラベルされている順序ありの有向木であって、次を満たすもとする。

1. ルートノードのラベルは S
2. 任意のノードについてラベルが N の元のとき、終端ノードであるか、子ノードがある場合は数は有限であり、ラベルを $A \in N$、子のラベルを順に $X_1, \cdots, X_m$ とすると $A \to X_1 \cdots X_m \in R$ である。
3. 任意のノードについてラベルが Σ の元のとき、終端ノードである。

とくに、ラベルが N の元である終端ノードを未解決ノードと呼ぶことにする。未解決ノードが存在する場合、不完全な導出木とよび、そうでない場合は完全な導出木と呼ぶことにする。 ノードの数が有限の場合は普通の導出木である。それを有限な導出木と呼ぶことにする。また、T に用いた生成規則の数、すなわちラベルが N の元である非終端ノードの数を $|T|$ であらわす。また、T が T' のルートを含む部分木であることを、$T \sqsubset T'$ であらわす。また、$T \sqsubset T'$ のとき、T のノード n とそれに対応する T' のノード n' を同一視して、$n = n'$ とする。いいかえると $n = n'$ とはルートからのパスが等しいということである。

定義 G の有限で不完全な導出木 T に対する次の写像

$$f(T) = (T \text{ の未解決ノード})$$

を G における導出順とよぶ。 f は可算集合から可算集合への写像である。たとえば、

$f_0(T) = (T$ の未解決ノードの中で最左のノード$)$

$f_1(T) = (T$ の未解決ノードの中で深さが最も浅いものの中で最左のノード$)$

はともに導出順である。f_0 を最左導出順、f_1 を最浅導出順と呼ぶことにする。

また、確率文脈自由文法 G_P に対し、文法 G の導出順を簡単に G_P の導出順と呼ぶことにする。G_P に導出順 f が与えられると、それにしたがって、

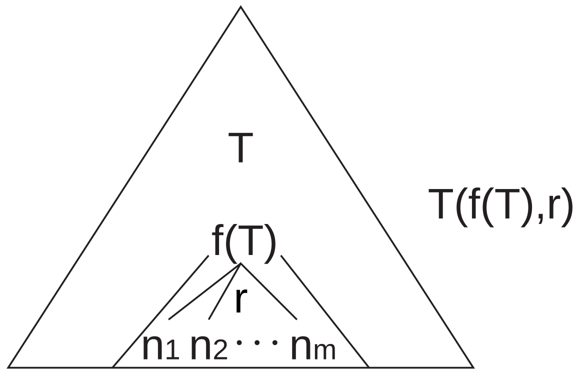

図 6: 導出順

前と同様にマルコフ連鎖を定めることができる。離散確率変数列 $X_1, X_2, \cdots$ を、有限な導出木を値としてもつものとして、その実現値を $T_1, \cdots, T_t$ とする。G_P, f に対する斉時マルコフ連鎖

$$P(X_t = T_t | X_1 = T_1, \cdots, X_{t-1} = T_{t-1}) = P(T_{t-1}, T_t)$$

を、次のように定める。不完全な導出木 T の未解決ノードの一つを n、そのラベルを $L(n)$ として、T に生成規則 $r = L(n) \to \Gamma \in R$ によって n の下に Γ を付け加えた導出木を $T(n, r)$ と書くことにして、

$$P(T_{t-1}, T_t) = \begin{cases} P(r) & : T_{t-1}\text{が不完全な導出木でかつ } r \in R_{L(f(T_{t-1}))}\text{かつ } T_t = T_{t-1}(f(T_{t-1}), r) \text{ のとき} \\ 1 & : T_{t-1}\text{が完全な導出木でかつ } T_t = T_{t-1}\text{のとき} \\ 0 & : \text{上記以外} \end{cases}$$

とくに、最左導出順を考え、各 X_t の終端ノードの列を Y_t とすると、$Y_1, Y_2, \cdots$ は前に定義した G_P で定まるマルコフ連鎖である。

定義 G_P の導出順 f が完全であるとは、G の任意の有限で不完全な導出木 T およびその任意の未解決ノード n に対して、

$$P(\{\forall i \in \mathbb{N}, f(X_i) \neq n\} \big| X_1 = T) = 0$$

が成り立つことである。ただし、$X_1, X_2, \cdots$ は G_P および f で定まるマルコフ連鎖で、$X_1 = T$ とする。 系 G_P の導出順 f が完全ならば、任意の T,n、および自然数 j に対して、$\sum_{i \in \mathbb{N}} P(f(X_i) = n \big| T \sqsubset X_j) = 1$。

Proof. $f(X_i) = n$ であれば それ以降は n は未解決ノードでなくなるから、すべての $i < j$ で $f(X_j) \neq n$ よって $f(X_i) = n$ は任意の i で背反である。 また、$T \sqsubset X_j$ であれば X_k が n を未解決ノードにもつような $k \leq j$ が存在するから、与式が成り立つ。 □

系 最浅導出順は任意の G_P で完全である。また、G_P が整合的でなければ、最左導出順は完全ではない。 補題 確率文脈自由文法を G_P、その完全な導出順を f とし、$X_1, X_2, \cdots$ を G_P, f で定まるマルコフ連鎖とする。このとき、任意の有限な導出木 T に対して、$P(T) = P(\{\exists i, T \sqsubset X_i\})$ とおくと、

$$P(T) = P(r_1) \cdots P(r_{|T|})$$

が成り立つ。ただし $r_1 \cdots r_{|T|}$ は T に使われた生成規則とする。

Proof. $|T|$ についての帰納法で示す。

$|T| = 0$ のとき、$T = S$ だから、$\forall T'$ で $S \sqsubset T'$ より、$P(\{\exists i, T \sqsubset X_i\}) = 1$。

$T' = T(n, r)$ とする。このとき、$\{T(n, r) \sqsubset X_i\} \subset \{T \sqsubset X_i\}$ より、$P(T(n, r), T) = P(T(n, r))$。したがって $P(T(n, r)|T) = P(r)$ を示せばよい。

$$\begin{aligned}
P(T(n,r)\big|T) &= \sum_{i=1}^{\infty} P(T(n,r) \sqsubset X_i \big| \{\exists j, T \sqsubset X_j\}) \\
&= \sum_{i=1}^{\infty} P(f(X_i) = n, X_{i+1} = X_i(r,n) \big| \{\exists j, T \sqsubset X_j\}) \\
&= \sum_{i=1}^{\infty} P(X_{i+1} = X_i(r,n) \big| f(X_i) = n, \{\exists j, T \sqsubset X_j\}) P(f(X_i) = n \big| \{\exists j, T \sqsubset X_j\}) \\
&= \sum_{i=1}^{\infty} P(r) P(f(X_i) = n \big| \{\exists j, T \sqsubset X_j\}) = P(r)
\end{aligned}$$

□

3.2 決定過程化した文法

$G = \langle \Sigma, N, R, S \rangle$ をグライバッハ標準形の CFG とする。これに、行動の有限集合 U、各生成規則の確率

$$P : R \times U \to [0, 1] \qquad \forall u \in U, \forall A \in N, \sum_{r \in R_A} P(r, u) = 1$$

および報酬

$$C : R \times U \to \mathbb{R}$$

を加えた $G_{U,P,C} = \langle \Sigma, N, R, S, U, P, C \rangle$ を決定過程化した文法と呼ぶことにする。以下、ある文法クラスを決定過程化したものを、SG-DP のように "-DP" をつけてあらわすことにする。

これより、2.1 章と同様に、$G_{U,P,C}$ と G の導出順 f からマルコフ決定過程を定めることができる。すなわち、有限の導出木を値としてもつ離散確率変数の列を $X_1, X_2, \cdots$ とし、U の元を値としてもつ確率変数の列を $Y_1, Y_2, \cdots$ として、

$$P(X_t = T_t | X_1 = T_1, Y_1 = u_1, \cdots, X_{t-1} = T_{t-1}, Y_{t-1} = u_{t-1}) = P(T_{t-1}, T_t, u_{t-1})$$

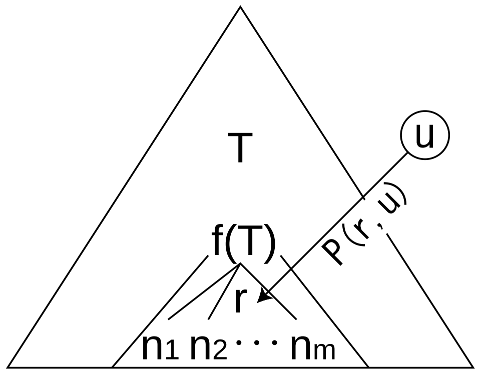

図 7: 決定過程化した文法

をみたすものを考える。ただし、

$$P(T_{t-1}, T_t, u_t) = \begin{cases} P(r, u_t) & : T_{t-1}\text{が不完全な導出木でかつ } r \in R_{L(f(T_{t-1}))}\text{かつ } T_t = T_{t-1}(f(T_{t-1}), r) \text{ のとき} \\ 1 & : T_{t-1}\text{が完全な導出木でかつ } T_t = T_{t-1}\text{のとき} \\ 0 & : \text{上記以外} \end{cases}$$

また、報酬はT_{t-1}, T_tで使われた生成規則を$r(T_{t-1}, T_t)$として、$C(r(T_{t-1}, T_t), u_{t-1})$である。これを$C(T_{t-1}, T_t, u_{t-1})$とかくことにする。有限な導出木の集合を$\mathcal{T}$として、$\mathcal{M} = \{\mathcal{T} \to U\}$として、列$(\mu_1, \mu_2, \cdots)$、$\mu_t \in \mathcal{M}$で方策を定義することができる。しかし、導出木は、導出順とあわせて考えると、もともと今までの導出の履歴の全てを含んでいると考えることができる。$|X_i| = i$より任意の$i, j \in \mathbb{N}, i \neq j$に対して$X_j$と$X_i$の値の集合は共通部分を持たないから、方策$\mu_1, \mu_2, \cdots$に対して、$|T| = i$のとき、$\mu_i(T) = \mu(T)$とすることで表現することができる。したがって、以下では、方策は$\mathcal{M}$の要素であるとする。

また、この場合、方策μを決めれば、$Y_t = X_t(\mu)$とすることで斉時マルコフ連鎖

$$P(X_t = T_t | X_1 = T_1, \cdots, X_{t-1} = T_{t-1}) = P(T_{t-1}, T_t, \mu(T_{t-1}))$$

が定まる。

3.3 決定過程化された文法の性質

3.3.1 深さ d の報酬期待値

決定過程化された文法を$G_{U,P,C}$、その導出順をf、方策をμとする。ある$T \in \mathcal{T}$に対する深さdの報酬期待値を、$f(T)$の下位ノードで、$f(T)$からの

深さが d までのノードによる報酬の和の期待値とする。すなわち、$T \sqsubset T'$ となる有限の導出木 T, T' に対して、

$$\delta_d(T,T') = \begin{cases} 1 & : T' \text{が不完全でかつ } f(T') \text{ が } f(T) \text{ の深さ } d \text{ 以下の下位ノード} \\ 0 & : \text{上記以外} \end{cases}$$

および、$d(T,T') = f(T)$ から $f(T')$ までの深さ として、として、深さ d の報酬期待値 $V_T(\mu, d)$ を、

$$V_T(\mu,d) = E_\mu\big[\sum_{i=1}^{\infty} \delta_d(T,X_i)\gamma^d(T,X_i)C(X_i,X_{i+1},\mu(X_i))\big|X_1 = T\big]$$

とする。式中の $\sum_{i=1}^{\infty} \delta_d(T,X_i)\gamma^{d(T,T')}C(X_i,X_{i+1},\mu(X_i))$ が有界な離散確率変数となることは、深 $f(T)$ から深さ d までの生成規則の取り方と行動のとり方が有限個しかないことから明らかである。また、深さ d の価値を $V_T^*(d) = \sup_{\mu \in M} V_T(\mu, d)$ とする。

補題 T の深さ d の報酬期待値を最大にする方策が存在して、T によらずに各ノードのラベルと深さに対して一意に定まる。すなわち、$V_T(\mu^*, d) = V_T^*(d)$ となる μ^* が存在し、$\delta(T,T') = 1$ を満たす任意の T' に対して $\mu^*(T') = h_d(L(f(T')), d(T,T'))$ となる写像 $h_d : N \times \{0, \cdots, d\} \to U$ が存在する。また、$f(T) = f(T') = A$ ならば $V_T^*(d) = V_T^*(d)$ であり、それを $V_A^*(d)$ であらわす。

Proof. 深さが 0 のとき、$V_T(\mu,0) = E_\mu[C(X_1,X_2,\mu(T))] = \sum_{r \in R_{L(f(T))}} P(r,\mu(T))C(r,\mu(T))$ より

$$h_0(A,0) = \arg\max_{u \in U} \sum_{r \in R_A} P(r,u)C(r,u)$$
$$V_A^*(0) = \sum_{r \in R_A} P(r,h_0(A,0))C(r,h_0(A,0))$$

とおくと、任意の T で $V_T^*(0) = V_{L(f(T))}^*(0)$ が成り立つ。

深さが d のとき、f は完全な導出順であることと、$\delta_d(T,T_i) \neq 0$ となる i

は有限個しかないことより、

$$
\begin{aligned}
&E_\mu\big[\sum_{i=1}^{\infty}\delta_d(T,X_i)\gamma^d(T,X_i)C(X_i,X_{i+1},\mu(X_i))\big|X_2=T(f(T),r)\big]\\
&=C(r,\mu(T))+E_\mu\big[\sum_{i=1}^{\infty}\gamma\sum_{n\in O_{f(T),r}}\delta_{d-1}(T(n,r),X_i)\gamma^{d(T(n,r),X_i)}C(X_i,X_{i+1},\mu(X_i))\big|X_2=T(f(T),r)\big]\\
&=C(r,\mu(T))+\gamma\sum_{n\in O_{f(T),r}}E_\mu\big[\sum_{i=1}^{\infty}\delta_{d-1}(T(n,r),X_i)\gamma^{d(T(n,r),X_i)}C(X_i,X_{i+1},\mu(X_i))\big|X_2=T(f(T),r)\big]\\
&=C(r,\mu(T))+\gamma\sum_{n\in O_{f(T),r}}E_\mu\big[\sum_{i=1}^{\infty}\delta_{d-1}(X_1,X_i)\gamma^{d(X_1,X_i)}C(X_i,X_{i+1},\mu(X_i))\big|T(n,r)\sqsubset X_1,f(X_1)=n\big]\\
&\le C(r,\mu(T))+\gamma\sum_{n\in O_{f(T),r}}V^*_{L(n)}(d-1)
\end{aligned}
$$

である。ただし $O_{n,r}$ をノード n の下に付け加えられたノードの集合とする。$N(r,i)$ を $r\in R$ の右辺に出てくる $i+1$ 番目の非終端記号の集合とすると、

$$
\begin{aligned}
h_d(A,0)&=\arg\max_{u\in U}\sum_{r\in R_A}P(r,u)(C(r,u)+\gamma\sum_{i=0}^{\#(r)}V^*_{N(r,i)}(d-1))\\
h_d(A,k)&=h_{d-1}(A,k-1)\\
V^*_A(d)&=\sum_{r\in R_A}P(r,h_d(A,0))(C(r,h_d(A,0))+\gamma\sum_{i=0}^{\#(r)}V^*_{N(r,i)}(d-1))
\end{aligned}
$$

とすれば任意の T で $V^*_T(d)=V^*_{L(f(T))}(d)$ が成り立つ。 □

3.3.2 最適価値

補題 3.3.1 より、深さと非終端記号に対して行動が一意に定まるとしても問題が無い。したがって、深さ d における非終端記号 n_i の個数を x_{id} として、$\boldsymbol{x}_d=(x_{1d},\cdots,x_{|N|d})$ とおき、また、$\boldsymbol{u}_d=(u_{1d},\cdots,u_{|N|d})$、$u_{id}\in U$ を深さ d での n_i に対する行動とすると、

$$P(\boldsymbol{x}_d|\boldsymbol{x}_1,\boldsymbol{u}_1,\cdots,\boldsymbol{x}_{d-1},\boldsymbol{u}_{d-1})=P(\boldsymbol{r}_d|\boldsymbol{x}_{d-1},\boldsymbol{u}_{d-1})$$

が成立することがわかる。したがって、$\boldsymbol{x}_d$ を状態、$\boldsymbol{u}_d$ を行動と考えれば MDP となっている。

非終端記号の有限集合を $N=\{n_1,n_2,\cdots,n_{|N|}\}$、生成規則の集合を $R=\{r_1,r_2,\cdots,r_{|R|}\}$、静的方策の集合を $M=\{N\to U\}$ とし、方策を $\boldsymbol{\pi}=(\mu_1,\mu_2,\cdots)$ であわらすとする。定義 [非終端記号数の期待値行列] $(|N|,|R|)$ 行列 $Q(\mu)$ を

$$Q_{ij}(\mu)=\begin{cases}p_{r_j}(\mu) & \text{if } r_j\in R_{n_i}\\ 0 & \text{otherwise}\end{cases}$$

とし、$(|R|,|N|)$ 行列 C を

$$C_{ij} = r_i \text{ の右辺にでてくる } n_j \text{ の個数}$$

とする。非終端記号数の期待値行列

$$A(\mu) \doteq Q(\mu)C$$

で定義する。 は各成分 $A_{ij}(\mu)$ が n_i から一回の導出で出てくる n_j の個数の期待値をあらわす行列であり、次が成り立つ [11]。

$$E[\boldsymbol{x}_d^T] = \boldsymbol{x}_1^T A(u_1)\cdots A(u_{d-1})$$

以降は、MDP での遷移確率行列を非終端記号数の期待値行列に置き換えるだけで議論が成り立つ。定義 [価値、最適価値] 方策 π に対する価値 $\boldsymbol{V}(\pi)$ を、$c_i(\mu) = \sum_{r\in R_{n_i}} P(r,\mu(N_i))C(r,\mu(N_i))$ とおいて、

$$[\boldsymbol{V}(\pi)]_i = \lim_{t\to\infty}\inf V_{n_i}(t) = \lim_{t\to\infty}\inf\big[\sum_{k=0}^{t}\gamma^k A(\mu_1)\cdots A(\mu_k)\boldsymbol{c}(\mu_{k+1})\big]_i$$

とし、最適価値 $\boldsymbol{V}^*$ を、

$$[\boldsymbol{V}^*]_i \doteq \sup_{\pi\in\Pi}[\boldsymbol{V}(\pi)]_i$$

とする。また、π が最適とは、

$$\boldsymbol{V}(\pi) = \boldsymbol{V}^*$$

が成立することとする。

定義 [写像 T_μ,T] $\mu \in M$ に対して、写像 $T_\mu : R^{|N|} \to R^{|N|}$ を

$$T_\mu(\boldsymbol{x}) = \boldsymbol{c}(\mu) + \gamma A(\mu)\boldsymbol{x}$$

で定める。また、写像 $T : R^{|N|} \to R^{|N|}$ を

$$[T(\boldsymbol{x})]_i = \max_{\mu\in M}[T_\mu(\boldsymbol{x})]_i$$

で定める。 系 $\pi = (\mu_1,\mu_2,\cdots)$ とすると、次式が成立する。

$$\boldsymbol{V}(\pi) = \lim_{t\to\infty}\inf T_{\mu_1}\circ\cdots\circ T_{\mu_t}(\boldsymbol{0})$$

定義 $\boldsymbol{x},\boldsymbol{y} \in R^{|N|}$ に対し、$\boldsymbol{x} \le \boldsymbol{y}$ を 全ての i で $x_i \le y_i$ が成立することとする。 系 写像 T_μ、T の単調性
$\boldsymbol{x} \le \boldsymbol{x}'$ ならば、任意の $\mu \in M$ で $T_\mu(\boldsymbol{x}) \le T_\mu(\boldsymbol{x}')$、および $T(\boldsymbol{x}) \le T(\boldsymbol{x}')$ が成立する。

Proof. A の要素が全て非負より明らか。 □

定義 [整合性]

$\mu \in M$ が整合的であるとは、μ によって定まる確率文脈自由文法が整合的であることとする。

定義 [適正]

$\mu \in M$ が適正であるとは、$\rho(A(\mu)) < 1$ が成立することとする。ただし $\rho(A)$ は A のスペクトル半径とする。 スペクトル半径の性質より以下が成り立つ。系 μ が適正であることは次の二つのどちらとも同値である。全ての i, j に対して

<u>(1)</u> $\lim_{t\to\infty}[A(\mu)^t]_{ij} = 0$

<u>(2)</u> $\lim_{t\to\infty}[\sum_{s=1}^{t} A(\mu)^s]_{ij} < \infty$ 任意の $r \in R$ で $\#(r) \leq 0$ のとき、A 終状態をのぞいた遷移確率行列となる。したがって系 3.3.2 より、μ が適正であることは、$A(\mu)$ を遷移確率とみなして MDP と考えたとき μ が適正であることと同値である。

また、μ は適正ならば整合的である [11]。ただし逆は真ではない。例えば $\rho(A(\mu)) = 1$ のときは整合的であるときと整合的ないときとがある。前者の例としては $S \to a,\ S \to SS$ がともに 1/2 の確率で起こるケースを考えればよく、後者の例としては $S \to S$ のみが起こるケースを考えればよい。系 任意の $\mu \in M$ に対して、$\rho(A(\mu)) \leq \max_{r\in R} \#(r) + 1$ が成り立つ。

Proof. $\rho(A(\mu)) \leq \max_i \sum_j A(\mu)_{ij} \leq \max_i \sum_j C(\mu)_{ij} = \max_{r\in R} \#(r) + 1$ □

命題次の二つの条件

1. $\gamma\rho(A(\mu)) < 1$ となる $\mu \in M$ が存在する。
2. 全ての $\gamma\rho(A(\mu)) \geq 1$ となる μ について、$\boldsymbol{x}(\mu) = -\infty$

が満たされれば、

1. $\boldsymbol{V}^*$ は $R^{|N|}$ における T の唯一の不動点である。
2. 任意の $\boldsymbol{x} \in R^{|N|}$ について、$\lim_{t\to\infty} T^t(\boldsymbol{x}) = \boldsymbol{V}^*$ が成り立つ。
3. 静的方策 $(\mu^*, \mu^*, \cdots)$ が最適方策であることの必要十分条件は $T^*_{\mu}(\boldsymbol{V}^*) = \boldsymbol{V}^*$ である。

が成り立つ。

Proof. MDP における遷移確率行列を期待値行列に置き換えると、系 3.3.2 が成り立つことより、文献 [?] の命題 3.3 の証明がそのまま成り立つ。 □

同様に最適行動価値 $Q^*: N \times U \to \mathbb{R}$ が、$u = \mu(A_i)$ が成り立つとして、$\boldsymbol{Q}^*(A_i, u) = [T_\mu(\boldsymbol{V}^*)]_i$ で定義される。ベルマン最適方程式 $T(\boldsymbol{x}) = \boldsymbol{x}$ は次の方程式と同等である。すなわち最適行動価値が次の方程式の唯一の解となる。

$$Q_{A,u} = \sum_{r \in R_A} P(r,u)(C(r,u) + \gamma \sum_{i=0}^{\#(r)} \max_{v \in U} Q_{N(r,i),v})$$

3.4 グライバッハ標準形の CFG-DP と POMDP

グライバッハ標準形の CFG-DP を $G_{U,P,C} = \langle \Sigma, N, R, S, U, P, C \rangle$ とする。N を内部状態、Σ を観測状態とみなし、さらに全ての生成規則の右辺の非終端記号の数を１つ以下にすれば、$G_{U,P,C}$ は POMDP になる。したがって、$G_{U,P,C}$ に対して以下のような学習を考えれば、グライバッハ標準形の CFG-DP は POMDP を包含している問題であるといえる。

- 入力：文と単語ごとの報酬値 の時系列
- 出力：行動
- 学習の目的：最適な行動の一群

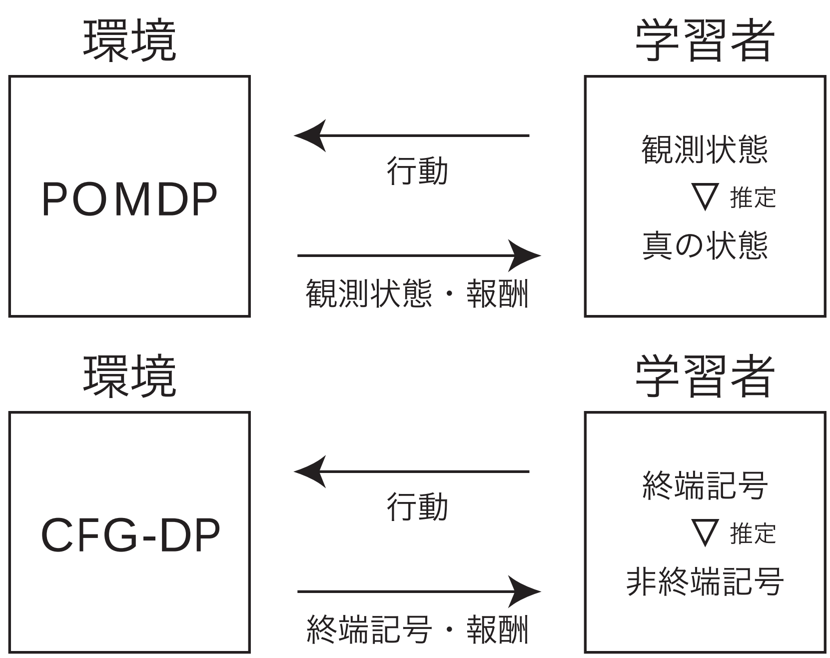

図 8: POMDP と CFG-DP

そこで、SG-DP を考えると、導出の順序が定まっていれば、時刻 t までの入力 $a_1 \cdots a_t \in \Sigma^*$ からそのときの非終端記号を一意に決定することができ

る。すなわち、SG-DP は、N を内部状態、Σ を観測状態と呼ぶことにすれば、内部状態の列が観測状態の履歴から 1 つに特定することができるということである。

さらに、全ての生成規則の右辺の非終端記号の数を 1 つ以下にすれば、SG-DP は有限状態の MDP と等しくなるから、SG-DP は有限状態の MDP を含んでいることがわかる。そのため、CFG-DP では POMDP を含む問題となるため、難しい問題となるが [7]、SG-DP では Q 学習を拡張することで簡単に最適方策を求める学習アルゴリズムを作ることができる。

3.4.1 SG-DP 下における Q 学習

SG では、t 番目に適用された生成規則を $r_t \in R$ とすると、それまでに得られた終端記号の列 $a_1 \cdots a_t \in \Sigma^*$ から r_t を一意に特定することができる。そのため、 Q 学習の更新式 (以下 SG-QL) は以下のようにすればよい。

$$Q_{A_t,u_t} := (1-\alpha_t(A_t,u_t))Q_{A_t,u_t} + \alpha_t(A_t,u_t)(C(r_t,u_t) + \gamma \sum_{i=0}^{\#(r)} \max_{v \in U} Q_{N(r,i),v}) \tag{2}$$

ただし、α_t は学習のステップサイズパラメータをあらわす確率変数であり、$P(\alpha_t(A,u) = 0 | (A_t,u_t) \neq (A,u)) = 1$ および、$\sum_{t=0}^{\infty} \alpha_t(A_t,u_t)^2 < \infty$ および、$\sum_{t=0}^{\infty} \alpha_t(A_t,u_t) = \infty$ を満たすものとする。

定理 更新式 5 を繰り返せば次の条件を満たせば最適行動価値に概収束する。

- 任意の $\mu \in M$ に対して $\gamma\rho(A(\mu)) < 1$

Proof. SG-QL の収束の証明の方法は QL の場合 [4] とほぼ同じで、問題は、写像 $H : \mathbb{R}^{|N \times U|} \to \mathbb{R}^{|N \times U|}$ を

$$H(\boldsymbol{Q})_{A,u} = \sum_{r \in R_A} P(r,u)(C(r,u) + \gamma \sum_{i=0}^{\#(r)} \max_{v \in U} Q_{N(r,i),v})$$

ととって、これが縮小写像になるかどうかという点だけであり、実際にそうなることを以下で示す。

任意の $\mu \in M$ に対して、$\gamma\rho(A(\mu)) < 1$ だから、$\boldsymbol{\xi} \in R^{|N|}$ に関する方程式

$$\xi_{A,u} = 1 + \gamma \sum_{r \in R_A} P(r,u) \sum_{i=0}^{\#(r)} \max_{v \in U} \xi_{N(r,i),v}$$

は、報酬を全て 1 としたときのベルマン最適方程式だから、任意の $\xi_{A,u}$ で 1 以上の解を持つ。よって、

$$\delta = 1 - \frac{1}{\max_{A \in N, u \in U} \xi_{A,u}}$$

とおくと、$0 \geq \delta < 1$ であって、任意の $\xi_{A,u}$ で

$$\delta\xi_{A,u} \geq (1 - \frac{1}{\xi_{A,u}})\xi_{A,u} = \gamma \sum_{r \in R_A} P(r,u) \sum_{i=0}^{\#(r)} \max_{v \in U} \xi_{N(r,i),v}$$

が成り立つ。また、任意の i で $q_i > 0, \xi_i > 0$ ならば、

$$\frac{\sum_{i=0}^{n} q_i}{\sum_{i=0}^{n} \xi_i} \leq \max_i \frac{q_i}{\xi_i}$$

が成り立つ。これらを使うと、任意の $\boldsymbol{Q}, \overline{\boldsymbol{Q}}$ に対して、

$$\begin{aligned}
&\frac{|H(\boldsymbol{Q})_{A,u} - H(\overline{\boldsymbol{Q}})_{A,u}|}{\xi_{A,u}} \\
&\leq \delta \frac{\gamma \left| \sum_{r \in R_A} \sum_{i=0}^{\#(r)} P(r,u)(\max_{v \in U} Q_{N(r,i),v} - \max_{v \in U} \overline{Q}_{N(r,i),v}) \right|}{\gamma \sum_{r \in R_A} \sum_{i=0}^{\#(r)} P(r,u) \max_{v \in U} \xi_{N(r,i),v}} \\
&\leq \delta \frac{\sum_{r \in R_A} \sum_{i=0}^{\#(r)} P(r,u) \max_{v \in U} \left| Q_{N(r,i),v} - \overline{Q}_{N(r,i),v} \right|}{\sum_{r \in R_A} \sum_{i=0}^{\#(r)} P(r,u) \max_{v \in U} \xi_{N(r,i),v}} \\
&\leq \delta \max_{r \in R, P(r,u) \neq 0} \max_{B \in N} \frac{P(r,u) \max_{v \in U} \left| Q_{B,v} - \overline{Q}_{B,v} \right|}{P(r,u) \max_{v \in U} \xi_{B,v}} \\
&\leq \delta \max_{B \in N} \max_{v \in U} \frac{\left| Q_{B,v} - \overline{Q}_{B,v} \right|}{\xi_{B,v}}
\end{aligned}$$

と評価できるから、重みつき最大値距離 $|\boldsymbol{x}|_{\boldsymbol{\xi}} = \max_i \frac{|x_i|}{\xi_i}$ に関して縮小写像となる。 □

MDP における Q 学習では、$\gamma < 1$ であれば必ず収束したが、この場合はそうではない。これは直感的にいうと、SG-DP の場合、後続の非終端記号が複数あるため、各階層あたりの適用される生成規則の数が指数的に増えることが許され、単に $\gamma < 1$ となる γ をかけていっただけでは総和が有限に抑えられる保証はないからである。

$\gamma < 1/\max_{\mu \in M} \rho(A(\mu))$ となるように γ をとればよいが、実際には生成確率は未知であるため、$\rho(A(\mu))$ の値を計算することはできない。しかし、例えば、適用された生成規則ごとに γ をかえて、より厳しい条件だが、任意の生成規則 r で $\gamma < 1/(\max_{r \in R} \#(r) + 1)$ とすれば、系 3.3.2 より、どのような SG-DP に対してもその割り引きつき期待報酬値が有限になり、収束する。通常強化学習では $\gamma = 0.9$ 程度であるが、この方法では、右辺に非終端記号が n 個あったならば、$\gamma < 1/n$ としなければならず、将来の分の加味がかなり減ってしまうことになる。

一方、もし、任意の方策 μ が整合的ならば、$\rho(A(\mu)) \leq 1$ であるので、$\gamma < 1$ ならば上記のアルゴリズムは収束する。したがって、方策が全て整合的であると判断できるなら γ は普通の強化学習と同じように考えてとってよい。

3.5 SG-QL の例：迷路問題

MDP とはならないが、PRSG-DP にはなるような迷路問題を考え、前節で定義した SG-QL を用いて、報酬を最大化するような方策を学習させた。

一方、同じ問題で、QL を用いても学習を行ったが、環境が MDP となっていないため、短いが報酬が最大化されないルートが学習される結果となった。

迷路問題は、エージェントが図 9 左中の S からスタートして G に到着した時点で報酬がもらえるというものである。エージェントは上下左右の 4 方向に動くことができ、動くごとに報酬として -0.5、壁にぶつかったら報酬として -1 受ける。それに加えて、ゴールの後に、地点 A を通った場合は状態 ‘good’ が発生して、報酬として +10、地点 B を通った場合は状態 ‘bad’ が発生して、-10 を得る。これは最後まで分からないとする。そして、A、B、および G の一歩手前の地点は、一度しか通れないとする。

このとき、まず、観測できる事柄はどの地点にいるかおよびどちらを通ったかであるから、

$$\Sigma = \{(i,j) | (i,j) \text{ は図の壁でない地点 }\} \cup \{\text{‘good’}, \text{‘bad’}\}$$

である。次に生成規則を考える。この場合、地点 A,B を除いては観測状態と同じであると見なして良い。しかし、地点 A,B が観測された際には、状態 ‘good’ や ‘bad’ を生成するので、

$$\begin{aligned} R = &\{\{A_{(i\pm1,j)}, A_{(i,j\pm1)}\} \to (i,j)A_{(i,j)} | (i,j) \in \Sigma - \{(2,4),(6,4)\}\} \\ &\cup \{A_{(2,3)} \to (2,4)A_{(2,4)}B,\ A_{(6,3)} \to (6,4)A_{(6,4)}C,\ B \to \text{‘good’},\ C \to \text{‘bad’}\} \end{aligned}$$

となる。

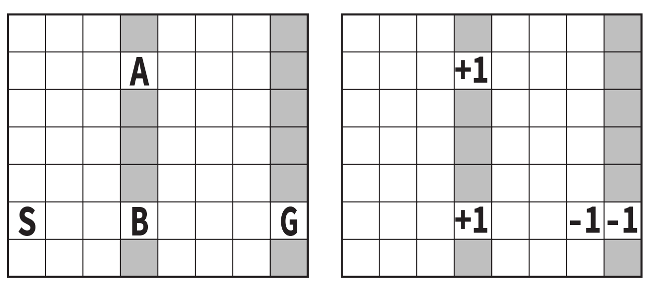

図 9: 比較に用いた迷路問題と形状

このときの形状、つまり SG-QL の各地点における # の割り当ては、図 9 右のようにとる。書いていないところはすべて 0 である。

また、普通の QL を適用する場合は、状態として各マスを割りあてることにする。

したがって、QL の状態数は壁も入れて $7 \times 8 = 56$ で、SG-QL の内部状態数 $|S|$ も $7 \times 8 + 2 - 2 = 56$ であり、どちらも同じ数の状態で表現されている。

学習はともに、学習係数 $\alpha = 0.2$ とし、戦略を $\varepsilon = 0.2$ の ε-グリーディ戦略とした。割引率は SG-QL にあわせて QL でも $\gamma = 1$ とした。

ε-グリーディ戦略とは、ε の確率でランダムな行動をし、$1 - \varepsilon$ の確率で、グリーディな行動をするということである。

ここで、エピソードとは、エージェントがスタート S からゴール G に到着するまでのことである。すなわち、エピソードとは文のことである。

図 10 は、SG-QL と QL のエピソードあたりの総報酬値と、エピソード回数 (学習回数) との関係を表している。学習回数を重ねるにつれ、総報酬値は、SG-QL、QL ともにある値に近づいてゆくが、その最終値は SG-QL のほうが QL よりも高い。

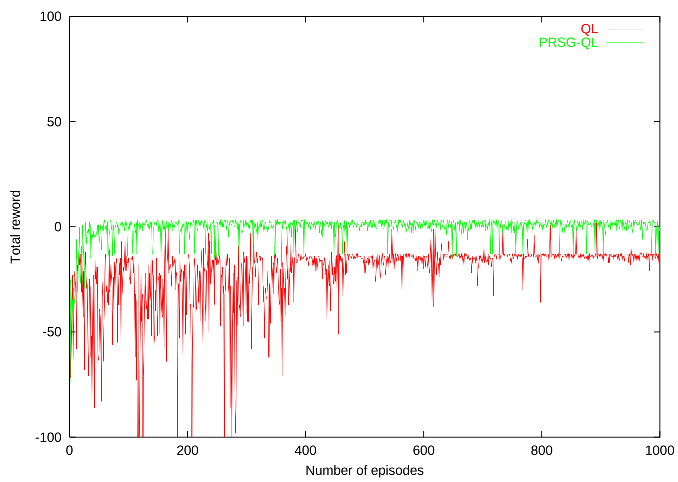

図 10: SG-QL と QL のエピソードあたりの報酬の比較

また、図 11 は、SG-QL と QL のエピソードあたりの移動回数と、エピソード回数 (学習回数) との関係を表している。同様に、学習回数を重ねるにつれ、SG-QL、QL ともにある値に近づいてゆくが、その最終値は QL のほうが短い。

これは、QL が距離は短いが報酬の低い B 地点を通るような方策を学習したのに対して、SG-QL が、この問題で最大報酬をあたえる、A 地点を通るような方策を学習した結果といえる。

実際に、エピソードを 5000 回繰り返したあと、グリーディな方策をとらせた場合の SG-QL と QL の結果は表 1 のようになり、QL では B 地点を通り、SG-QL では A 地点を通るような方策が学習が行われていることがわかる。

これは、各セルを状態ととったときには、A 地点、B 地点のどちらを通ったか、という過去の履歴によってゴールでの報酬が変化するため、即ち、過程のマルコフ性が崩れるため、当然の結果であるといえる。従って、例えば、A、B のどちらを通ったかで同じ位置にいても状態を 2 つに分けておけば、状

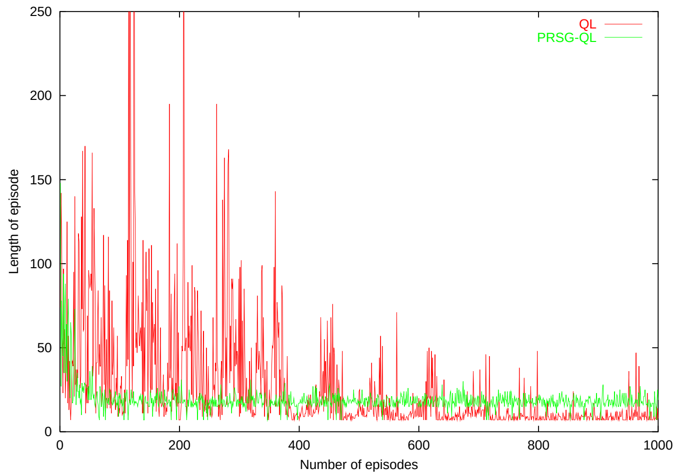

図 11: SG-QL と QL のエピソードあたりの移動数の比較

表 1: QL と SG-QL の学習結果

	QL	SG-QL
履歴	$(6,1)(6,2)(6,3)$ $(6,4)(6,5)(6,6)$ $(6,7)(6,8)$	$(6,1)(5,1)(4,1)(3,1)$ $(3,2)(2,2)(2,3)(2,4)$ $(2,5)(2,6)(2,7)(3,7)$ $(4,7)(5,7)(6,7)(6,8)$
移動回数	7	15
総報酬	-13.5	2.5

態数は倍となるが環境は MDP となり、QL でも B 地点を通る道が学習されると考えられる。

4 一般性の枠組みが崩れない場合

4.1 この章の目的

3 章では、環境となる確率単純文法について、確率を除いた文法自身は既知であると仮定したうえでそれにそって強化学習をする方法について提案し、考察した。しかし、文法の枠組みを決めるためには、MDP のように状態数を決めるだけではだめで、どのような生成規則が必要になるか決定しなければならない。

そこで、この章では、環境となる確率単純文法がそのもととなる文法を含めて全て未知であるとしたとき、文法の枠組み自体の学習も可能な方法について考える。

まず、文法の枠組みがわかればそれが SG のサブクラスであれば SG-QL で学習することができるので、確率単純文法から確率を除いた、もとの文法を推定したい。入力のデータは、ある確率単純文法から独立に出力された、文の列 $\{w_1, w_2, \cdots, w_n\}$ である。このとき、まず考えられるのは次の方法である。

方法 1

もととなる文法の候補 $\{G_1, G_2, \cdots\}$ を列挙する。

そして、各 G_{iP} を P について最尤推定を行っていったものを、$\{G_{1\widehat{P_1}}, G_{2\widehat{P_2}}, \cdots\}$ とする。

さらにそのなかでもっとも確率の高くなるものをえらんで出力する。 方法 1 では、文の数 n を無限大に持ってゆけば、最終的には、出力は、学習対象の確率単純文法と等価になるような確率単純文法の集合のなかに確率収束するだろう。

もちろん、方法 1 以外にも様々な優れた統計的手法がある [?]。しかし、そのような統計的手法では、生成規則の確率を求めてしまうことになり、強化学習の、遷移確率には陽に触れない、という利点を捨ててしまうことになる。一般に、MDP の最適化問題は、遷移確率を陽に求めることができるなら、ダイナミックプログラミングを用いて計算すればよい、という結論になるからである。

また、方法 1 では、決して一つの文法に確率収束してくれるわけではない。等価な確率単純文法では、優劣をつけることができないから、出力は、いつまでたってもそれらの間をふらふらと移動することになる。

等価な確率単純文法であっても、もとの文法の形が異なれば、最適方策自身はかわらなくても、途中経過がかわってくるので、強化学習がうまくいかなくなる。

そこで、あえて統計的な処理は一切せずに推定を行いたい。 方法 2
頻度を考えず正の例だと考えて学習する。そのかわり、

文法 G に対して、その同値類$\overline{G} = \{G' | L(G) = L(G')\}$ を考える。
任意の G の任意の P に対して $G_P = G^*_Q$となるような $G^* \in \overline{G}$および Q が存在する。

が成り立つようなクラスを考え、G^* を出力する。 方法 2 は逆説的かもしれないが、生成規則の確率については全く触れてないので、そのようなクラスをある程度広く取ることができるなら、よい方法と考えられる。

4.2 確率単純文法の性質

あるCFG のクラスのサブクラスの文法に生成規則の選択確率を加えて確率化したものを頭に 'P' をつけて、例えば 'PCFG' という風に書くことにする。$G_P = \langle N, \Sigma, R, S, P \rangle$ を PSG とする。また、確率文脈自由文法 $G_P = \langle N, \Sigma, R, S, P \rangle$ が与えられたとき、それから P を取り除いた文脈自由文法 $G = \langle N, \Sigma, R, S \rangle$ を G_P の元の文法と呼ぶことにする。

ここで、確率としての制約から、P のとりうる範囲は、

$$Z_G = \{P : R \to [0,1] | \forall A \in N, \sum_{r \in R_A} P(r) = 1\}$$

であった。このとき、確率言語 $K(G_P)$ を、

$$K(G_P) : \Sigma^* \cup \{\phi\} \to [0,1]$$

であり、

$$K(G_P)(w) = \begin{cases} \prod_{i=1}^{|w|} P(r(w)_i) & w \in L(G) \\ 0 & w \notin L(G) \\ 1 - \sum_{x \in L(G)} \prod_{i=1}^{|x|} P(r(G,x)_i) & w = \phi \end{cases}$$

と定義する。ただし、$w \in L(G)$ のとき w を導出するときにその適用した生成規則は一意に決まって、それを $(r(G,w)_1, \cdots, r(G,w)_{|w|})$ とする。

すなわち、$w \in \Sigma^*$ のときは、$K(G_P)(w)$ は G_P から w が導出される確率を意味し、$K(G_P)(\phi)$ は、無限に導出を続ける確率である。

また、G_P が整合的であることの必要十分条件は $K(G_P)(\phi) = 0$ である。

また、G をもととする PCFG から生成される確率言語全体を

$$\mathbb{K}^+(G) = \{K(G_P) | P \in Z_G\}$$

と定義する。

G が既約であり、かつ任意の $r \in R$ で $P(r) > 0$ ならば、G_P を既約と呼ぶことにする。そうすると、G_P が既約ならば、明らかに、

$$L(G) = \{w \in \Sigma^* | K(G_P)(w) \neq 0\}$$

となる。また、G をもととする既約な PCFG から生成される確率言語全体を

$$\mathbb{K}(G) = \{K(G_P) \in \mathbb{K}^+(G) | G_P : \text{既約}\}$$

であらわす。

系 $\mathbb{K}(G) \cap \mathbb{K}(G') \neq \emptyset$ ならば $L(G) = L(G')$。

Proof. 定義より、$K(G_P) = K(G'_Q)$ となる既約な G_P, G'_Q が存在するから、

$$L(G) = \{w \in \Sigma^* | K(G_P)(w) \neq 0\} = \{w \in \Sigma^* | K(G'_Q)(w) \neq 0\} = L(G')$$

□

最左導出 $S \overset{*}{\Rightarrow} x\alpha$, $(r_1, r_2, \cdots, r_{|x|})$ に対して,

$$Pre(G_P)(x) = (\prod_{i=1}^{|x|} P(r_i))$$

とおく。G_P が整合的なとき、

$$Pre(G_P)(x) = \sum_{w \in \{w \in L(G) | \exists y, w = xy\}} P(w) \tag{3}$$

がなりたつ。

系 G_P,G_Q が既約で整合的ならば、$P = Q$ の必要十分条件は $K(G_P) = K(G_Q)$ である。

Proof. $P = Q$ ならば $K(G_P) = K(G_Q)$ は明らか。

G が既約だから、任意の $r = A \to a\alpha$ について、最左導出 $S \overset{*}{\Rightarrow} xA\beta$, $(r_1, r_2, \cdots, r_{|x|})$ となるものが存在する。また、G_P 既約より、$\forall x \in L(G), Pre(G_P)(x) \neq 0$ だから、

$$\frac{Pre(G_P)(xa)}{Pre(G_P)(x)} = P(r)$$

がなりたつ。従って、任意の $w \in L(G)$ に対して、$K(G_P)(w) = Pre(G_P)(w)$ であるから、$Pre(G_P)(w) = Pre(G_Q)(w)$ を示せばよいが、G_P,G_Q はともに整合的なので、式 3 および $K(G_P) = K(G_Q)$ より成り立つ。 □

補題 G, H を既約なSGとする。$\mathbb{K}(G) \subset \mathbb{K}(H)$ ならば次が成り立つ。$S \overset{*}{\Rightarrow} [G]xA\alpha$,$S \overset{*}{\Rightarrow} [H]xB\beta$ ならば、$S \overset{*}{\Rightarrow} [H]yB\beta'$ となる任意の y について $S \overset{*}{\Rightarrow} [G]yA\alpha'$。

Proof. $G = \langle N, \Sigma, R, S\rangle, H = \langle N', \Sigma, R', S\rangle$ とする。

$S \overset{*}{\Rightarrow} [G]xA\alpha$,$S \overset{*}{\Rightarrow} [H]xB\beta$ かつ $S \overset{*}{\Rightarrow} [G]yC\alpha'$,$S \overset{*}{\Rightarrow} [H]yB\beta'$ となる x, y が存在すると仮定して矛盾を導く。

系 4.2 より $L(G) = L(H)$ だから、$r = A \to a\gamma \in R'$ に対して、$s = B \to a\gamma' \in R$, $t = C \to a\gamma'' \in R$ が存在する。また、G 既約より、G_P が既約で整合的でかつ $P(s) \neq P(t)$ となるような P が存在する。$\mathbb{K}(G) = \mathbb{K}(H)$ より $K(G_P) = K(H_Q)$ となるような H_Q が存在し、G_P は整合的なので式 3 より $Pre(G_P) = Pre(H_Q)$ が成り立つ。

しかし、$P(s) = Pre(G_P)(xa)/Pre(G_P)(x) = Pre(H_Q)(xa)/Pre(H_Q)(x) = Q(r)$、$P(t) = Pre(G_P)(ya)/Pre(G_P)(y) = Pre(H_Q)(ya)/Pre(H_Q)(y) = Q(r)$ より $P(s) = P(t)$ となって矛盾。 □

G_P を既約で整合的であるとする。$\Omega = \{d : \mathbb{N} \to \Sigma^*\}$ とする。また、全単射 $f : \mathbb{N} \to \Sigma^*$ が存在する。$(W_n : \Omega \to \mathbb{R} | n \in \mathbb{N})$,$W_n(d) = f^{-1}(d(n))$ を独立な離散確率変数列で分布

$$F((-\infty, x]) = \sum_{n=1}^{[x]} K(G_P)(f(n))$$

をもつものとする。このとき G_P が既約ならば W が正のデータである確率は 1 である。すなわち次が成り立つ。補題 G_P が既約ならば $P(D_G) = 1$

Proof. 任意の $d \in \Omega/D_G$ に対して、$\exists w \in L(G)$, $\forall n \in \mathbb{N}$,$d(n) \neq w$ または $\exists n$, $d(n) \in \Sigma^*/L(G)$ が成り立つ。したがって、$\Omega/D_G = \bigcup_{w \in L(G)} \bigcap_{n \in \mathbb{N}} \{d \in \Omega | d(n) \neq w\} \cup \bigcup_{n \in \mathbb{N}} \{d \in \Omega | d(n) \in \Sigma^*/L(G)\}$ が成り立つ。よって、

$$\begin{aligned}
P(\Omega/D_G) &\leq \sum_{w \in L(G)} P(\bigcap_{n \in \mathbb{N}} \{d \in \Omega | d(n) \neq w\}) + \sum_{n \in \mathbb{N}} P(\{d \in \Omega | d(n) \in \Sigma^*/L(G)\}) \\
&= \sum_{w \in L(G)} \prod_{n \in \mathbb{N}} P(W_n \neq f^{-1}(w)) + \sum_{n \in \mathbb{N}} P(W_n \notin f^{-1}(L(G))) \\
&= 0
\end{aligned}$$

□

4.3 PCFG のもとの文法の学習

G_P の確率 P はおいておいて、そのもとの文法 G だけを学習することを考えよう。これは、強化学習が遷移確率 P に陽にふれることなく学習する、というメリットをそのまま生かしたいからである。

このようなことが可能な PCFG のクラスは、どのような条件を満たす必要があるか考えたい。そのためにまず、学習可能性を次のように定義をする。

定義 ある<u>PCFG</u> のサブクラス $\mathcal{G}$ のもとの文法が統計なしに学習可能とは、$\mathcal{G}$ に含まれる任意の既約な G_P のもとの文法 G の任意の正のデータに対して、$K(G_P) \in \mathbb{K}(G')$ となるような G' に収束するアルゴリズムが存在することとする。また、そのアルゴリズムを $\mathcal{G}$ のもとの文法が統計なしに学習できるアルゴリズムと呼ぶ。 また、決定過程化した文法のクラスについても、同様に、もとの文法と、もとの文法が統計なしに学習可能ということが定義できる。

これは直感的にいって、確率 P の値には触れていないから、統計的な処理をせずに既約な G_P のもとの文法 G が学習できるということと考えられる。

また、これは文法の学習可能性 (定義 2.7) よりも厳しい条件である。アルゴリズムは、元の文法を G_P とすると、$L(G) = L(G')$ となるような G' を求めるだけではだめで、$K(G_P) = K(G'_Q)$、つまり言語の確率分布の等しくなるような G' を求めなければならない。

系 $\mathcal{G}$ を<u>PCFG</u> のサブクラスとし、$\mathcal{G}^- = \{G | G_P \in \mathcal{G}\}$ とおく。$\mathcal{G}$ のもとの文法が統計なしに学習可能ならば、$\mathcal{G}^-$ は学習可能である。

Proof. $\mathcal{G}$ のもとの文法が統計なしに学習できるアルゴリズムを A とする。任意の $G \in \mathcal{G}^-$ に対して、$G^* = \lim_{i\to\infty} A(d_i)$ とおくと、P によらずに d は決まるから、$\mathbb{K}(G) \subset \mathbb{K}(G^*)$。よって、系 4.2 より、$L(G) = L(G^*)$。ゆえに、$\mathcal{G}^-$ は A で学習可能である。 □

系 <u>PSG</u> のクラスは統計なしに学習不可能である。系 $\overline{G} = \{G' \in \mathcal{G} | L(G') = L(G)\}$ とする。$\mathcal{G}$ のもとの文法が統計なしに学習可能ならば、次が成り立つ。任意の $G \in \mathcal{G}^-$ に対して、$\bigcup_{G' \in \overline{G}} \mathbb{K}(G') \subset \mathbb{K}(G_0)$ となるような $G_0 \in \overline{G}$ が存在する。

Proof. $\mathcal{G}$ のもとの文法が統計なしに学習できるアルゴリズムを A とする。$G^* = \lim_{i\to\infty} A(d_i)$ とおく。G の任意の正のデータ d について、$G' \in \overline{G}$ ならば、$L(G) = L(G')$ より、$d \in D'_G$ である。このとき、定義より、任意の既約な G'_P について $K(G'_P) \in \mathbb{K}(G^*)$。よって、$\mathbb{K}(G') \subset \mathbb{K}(G^*)$ となるから、$\bigcup_{G' \in \overline{G}} \mathbb{K}(G') \subset \mathbb{K}(G^*)$。また、系 4.2 より、$L(G) = L(G^*)$ だから、$G^* \in \overline{G}$。 □

補題 $G = \langle N, \Sigma, R, S\rangle$ を既約な<u>RSG</u>、$H = \langle N', \Sigma, R', S'\rangle$ を G と等価な標準形の<u>RSG</u> とすると、$\mathbb{K}(G) \subset \mathbb{K}(H)$ がなりたつ。ただし、ここでいう等価とは、$L(G) = L(H)$ かつ $\#_G = \#_H$ であることとする。

Proof. 任意の G_P となる P に対して、H_Q となる Q が存在して、$K(G_P) = K(H_Q)$ が成り立てばよい。

G を標準形に直すときに $A \in N$ から書き換えた N' の元の集合を $N'(A)$ とする。任意の $B \in N'(A)$ に対して、$B \to a\alpha' \in R'$ ならば $A \to a\alpha \in R$ である。また逆に、$A \to a\alpha \in R$ ならばある $B \in N'(A)$ があって $B \to a\alpha' \in R'$ である。

$a_1 \cdots a_m \in L(G)$ に対して、G での最左導出での適用された生成規則を順に $r_1, \cdots, r_m$ とし、H での生成規則を順に $s_1, \cdots, s_m$ とする。$r_i = A \to a_i\alpha$, $s_i = B \to a_i\alpha'$ とすると、 $B \in N'(A)$ が成り立つ。従って、任意の $r \in R'$ に対して、$r = B \to a\alpha'$、$B \in N'(A)$ とすると、$Q(r) = P(A \to a\alpha)$ ととれば、任意の $w \in L(G)$ に対して $K(G_P)(w) = K(G_Q)(w)$ が成り立つ。 □

命題 PRSG のクラスはもとの文法が統計なしで学習不可能である。

Proof. 実際に統計なしで学習不可能な言語の例を示す。

$$L = \{ace, acf, ade, adf, bce, bcf, bde, bdf\}$$

言語が L である RSG は標準形では次の二つだけである。

$\Sigma = \{a, b, c, d, e\}$ とする。

$$G_P = \langle N, \Sigma, R, S, P\rangle \qquad N = \{A_1, B_1, A_2, B_2, S\}$$

$$\begin{aligned} R_P = \{S &\to aA_1A_2 : p, \\ S &\to bB_1B_2 : 1-p, \\ A_1 &\to c : q_1, \\ A_1 &\to d : 1-q_1, \\ B_1 &\to c : r_1, \\ B_1 &\to d : 1-r_1, \\ A_2 &\to e : q_2, \\ A_2 &\to f : 1-q_2, \\ B_2 &\to e : r_2, \\ B_2 &\to f : 1-r_2\ \} \end{aligned}$$

$$G'_Q = \langle N', \Sigma, R', S, Q\rangle \qquad N' = \{A, B, C, D, S\}$$

$$\begin{aligned} R'_Q = \{S &\to aA : p', \\ S &\to bB : 1-p', \\ A &\to cC : q'_1, \\ A &\to dD : 1-q'_1, \\ B &\to cC : r'_1, \\ B &\to dD : 1-r'_1, \\ C &\to e : q'_2, \\ C &\to f : 1-q'_2, \\ D &\to e : r'_2, \\ D &\to f : 1-r'_2\ \} \end{aligned}$$

	$P(ace)$	$P(acf)$	$P(ade)$	$P(adf)$
G_P	pq_1q_2	$pq_1(1-q_2)$	$p(1-q_1)q_2$	$p(1-q_1)(1-q_2)$
$G'_{P'}$	$p'q'_1q'_2$	$p'q'_1(1-q'_2)$	$p'(1-q'_1)r'_2$	$p'(1-q'_1)(1-r'_2)$

$P(bce)$	$P(bcf)$	$P(bde)$	$P(bdf)$
$(1-p)r_1r_2$	$(1-p)r_1(1-r_2)$	$(1-p)(1-r_1)r_2$	$(1-p)(1-r_1)(1-r_2)$
$(1-p')r'_1q'_2$	$(1-p')r'_1(1-q'_2)$	$(1-p')(1-r'_1)r'_2$	$(1-p')(1-r'_1)(1-r'_2)$

たとえば、

$$\frac{P(acf)}{P(ace)} = \frac{1-q_2}{q_2} \qquad \frac{P(adf)}{P(ade)} = \frac{1-q_2}{q_2}$$
$$\frac{P'(acf)}{P'(ace)} = \frac{1-q'_2}{q'_2} \qquad \frac{P'(adf)}{P'(ade)} = \frac{1-r'_2}{r'_2}$$

したがって、任意の P に対して $\frac{P(acf)}{P(ace)} = \frac{P(adf)}{P(ade)}$ であるので、$q'_2 \neq r'_2$ となるようにとれば、そのような $G'_{P'}$ に対して $K(G'_{P'}) = K(G_P)$ となるような P は存在しない。逆に、

$$\frac{P(bce)}{P(ace)} = \frac{1-p}{p}\frac{r_1}{q_1}\frac{r_2}{q_2} \qquad \frac{P(bcf)}{P(acf)} = \frac{1-p}{p}\frac{r_1}{q_1}\frac{1-r_2}{1-q_2}$$
$$\frac{P'(bce)}{P'(ace)} = \frac{1-p'}{p'}\frac{r'_1}{q'_1} \qquad \frac{P'(bcf)}{P'(acf)} = \frac{1-p'}{p'}\frac{r'_1}{q'_1}$$

より、任意の P' に対して $\frac{P'(bce)}{P'(ace)} = \frac{P'(bcf)}{P'(acf)}$ であるので、$q_2 \neq r_2$ となるようにとれば、そのような G_P に対して $K(G'_{P'}) = K(G_P)$ となるような P' は存在しない。

以上より、$\mathbb{K}(G)$ と $\mathbb{K}(G')$ は包含関係にない。また、H を $L(H) = L$ を満たす標準形でない RSG とすると、補題 4.3 より $H \subset G$ または $H \subset G'$ であるから、当然 H は G と G' の両方を包含することはできない。したがって、$\bigcup_{G' \in \overline{G}} \mathbb{K}(G') \subset \mathbb{K}(G_0)$ となるような $G_0 \in \overline{G}$ は存在しないから、系 4.3 より PRSG は統計なしに学習できない。 □

4.4 PNMSG

この節では、PNMSG のクラスがもとの文法が統計なしで学習可能な文法のクラスであることを示す。

NMSG とは、吉仲によって提案された単純文法のサブクラスで、以下の性質を満たす文法である。単純文法 $G = \langle N, \Sigma, R, S\rangle$ が NMSG であるとは、既約であって、$A \in N$ が直接導出する文字全部の集合を

$$U(A) = \{a \in \Sigma \mid {}^{\exists}\alpha \in N^*, (A \to a\alpha) \in R\}$$

として、

$$U(A) = U(B) \text{ ならば } A = B$$

が成り立つときをいう。

NMSG は、Deepest Form という特殊な形の文法に一意に変換できる。

$G = \langle N, \Sigma, R, S\rangle$ を NMSG とする。$A \in N$ が $B \in N$ に付かれているとは、任意の導出について、

$$S \xrightarrow{*} wA\alpha \quad \text{ならば} \quad \alpha = B\gamma$$

が満たされることを言う。ただし、$w \in \Sigma^*$ $\alpha, \beta, \gamma \in N^*$

すなわち、最左導出においては、A の右隣りには必ず B がある、ということである。

G が Deepest Form であるとは、任意の $A \in N$ について、A がなににも付かれていないことをいう。

Deepest Form への変換アルゴリズムを図 12 に示す。ここで、

$$R_B = \{A | A \text{ は } B \text{ に付かれている }\}$$

である。

```
Input G
   while( R_B ≠ ∅ となる B が存在する ){
      for( each C → aα ∈ R ){
         if(C ∈ R_B ){
            α を αB に置き換える;
         }
      for( each A ∈ R_B ){
         α(または αB) 中の AB を A に置き換える;
      }
   }
   output G and halt
```

図 12: Deepest Form に変換するアルゴリズム

次のことがなりたつことが示されている。命題(吉仲) 図12 のアルゴリズムを DF とする。また、G,G' を任意のNMSG とする。

1. DF(G) はNMSG でありDeepest Form である。
2. $L(G) = L(\mathrm{DF}(G))$。
3. $\mathrm{DF}(G) \simeq \mathrm{DF}(G')$ は $L(G) = L(G')$ の必要十分条件である。

補題 G と H をNMSG とすると、$L(G) = L(H)$ ならば $\mathbb{K}(G) = \mathbb{K}(H)$ が成り立つ。

Proof. 命題 4.4 より、$L(G) = L(H)$ から $\mathrm{DF}(G) \simeq \mathrm{DF}(H)$ が成り立つ。したがって、$\mathbb{K}(G) = \mathbb{K}(\mathrm{DF}(G))$ を示せばよい。

$G = \langle N, \Sigma, R, S\rangle$ とおく。Deepest Form への変換アルゴリズムの過程を考えると、変換によって生成規則の左辺の非終端記号は変化しない。すなわち、任意の $r = A \to a\alpha \in R$ に対して、$\mathrm{DF}(r) = A \to a\beta$ と書くことができる。したがって、$P'(\mathrm{DF}(r)) = P(r)$ として、$\mathrm{DF}(G_P) = \mathrm{DF}(G)_{P'}$ と定義することができる。したがって、任意の P について $K(\mathrm{DF}(G_P)) = K(G_P)$ であることを示せばよい。

まず、任意の $S \to a\alpha \in R$ について、P' の定義より、

$$Pre(G_P)(a) = P(S \to a\gamma) = P'(S \to a\gamma') = Pre(\mathrm{DF}(G_P))(a)$$

が成立する。

アルゴリズムの過程で B のつかれているを解除することによって、G' が G'' に変換されたとする。文献 [?] の補題 6.13 によると、

$$f_B(A) = \begin{cases} AB & \text{if } A \in R_B \\ A & \text{otherwise} \end{cases}$$

とおくと、$S \overset{*}{\Rightarrow} [G']x\alpha$ が成り立つことと、 $S \overset{*}{\Rightarrow} [G'']xf_B(\alpha)$ が成り立つこととは同値であるから、$S \overset{*}{\Rightarrow} [G']xA\alpha$ ならば、$S \overset{*}{\Rightarrow} [G'']xA\beta$ となる $\beta \in N^*$ が存在する。これが全ての変換過程において成立するから、結局、$S \overset{*}{\Rightarrow} [G]xA\alpha$ となるような G での任意の導出について、適当な $\beta \in N^*$ が存在して、$S \overset{*}{\Rightarrow} [DF(G)]xA\beta$ が成り立つ。よって、$Pre(G_P)(x) = Pre(\mathrm{DF}(G_P))(x)$ と仮定すると、$\mathrm{DF}(G_P)$ の定義より、

$$Pre(G_P)(xa) = Pre(G_P)(x)P(A \to a\gamma) = Pre(\mathrm{DF}(G_P))(x)P'(A \to a\gamma') = Pre(\mathrm{DF}(G_P))(xa)$$

が成立する。従って任意の $w \in L(G)$ について $K(G_P)(w) = Pre(G_P)(w)$ が成り立つから、$K(G_P)(w) = K(\mathrm{DF}(G_P))(w)$ が成立する。

□

命題 PNMSG のクラスはもとの文法が統計なしに学習可能である。

Proof. NMSG は学習可能だから、NMSG を学習できるアルゴリズムを A、任意の文法 G とその正のデータ d に対して、$G^* = \lim_{i\to\infty} A(d_i)$ とおくと、定義より $L(G) = L(G^*)$ だから 補題 4.4 より $\mathbb{K}(G) \subset \mathbb{K}(G^*)$ である。よって PNMSG は統計なしに学習可能。 □

同様に、PNMRSG のクラスであってももとの文法が統計なしに学習可能であることは明らかである。NMRSG とは、NMSG かつ RSG であることをいう。本論文で PNMSG でなく PNMRSG で議論する理由は、NMSG は理論的には学習可能であるが、計算量の観点から非常に効率が悪いからである。

ここで、命題 4.3 の反例を NMRSG で考えてみよう。すると、G では、$U_{A_1} = U_{B_1} = \{c, d\}$、$U_{A_2} = U_{B_2} = \{e, f\}$ より、$A_1 = B_1$、$A_2 = B_2$ でなければならないから、

$$G_P = \langle N, \Sigma, R, S, P \rangle \qquad N = \{A_1, A_2, S\}$$

$$\begin{aligned} R_P = \{ & S \rightarrow aA_1A_2 : p, \\ & S \rightarrow bA_1A_2 : 1-p, \\ & A_1 \rightarrow c : q_1, \\ & A_1 \rightarrow d : 1-q_1, \\ & A_2 \rightarrow e : q_2, \\ & A_2 \rightarrow f : 1-q_2 \ \} \end{aligned}$$

一方で、G' では、$U_A = U_B = \{c, d\}$、$U_C = U_D = \{e, f\}$ より、$A = B$、$C = D$ でなければならないから、

$$G'_Q = \langle N', \Sigma, R', S, Q \rangle \qquad N' = \{A, B, S\}$$

$$\begin{aligned} R'_Q = \{ & S \rightarrow aA : p', \\ & S \rightarrow bA : 1-p', \\ & A \rightarrow cC : q'_1, \\ & A \rightarrow dC : 1-q'_1, \\ & C \rightarrow e : q'_2, \\ & C \rightarrow f : 1-q'_2 \ \} \end{aligned}$$

である。これらより $L(G)$ に含まれる文の導出確率を計算して比較すると、

	$P(ace)$	$P(acf)$	$P(ade)$	$P(adf)$
G_P	pq_1q_2	$pq_1(1-q_2)$	$p(1-q_1)q_2$	$p(1-q_1)(1-q_2)$
$G'_{P'}$	$p'q'_1q'_2$	$p'q'_1(1-q'_2)$	$p'(1-q'_1)q'_2$	$p'(1-q'_1)(1-q'_2)$

$P(bce)$	$P(bcf)$	$P(bde)$	$P(bdf)$
$(1-p)q_1q_2$	$(1-p)q_1(1-q_2)$	$(1-p)(1-q_1)q_2$	$(1-p)(1-q_1)(1-q_2)$
$(1-p')q'_1q'_2$	$(1-p')q'_1(1-q'_2)$	$(1-p')(1-q'_1)q'_2$	$(1-p')(1-q'_1)(1-q'_2)$

となって、$p = p'$, $q_1 = q'_1$, $q_2 = q'_2$ とすれば明らかに $K(G_P) = K(G'_Q)$ なので $\mathbb{K}(G) = \mathbb{K}(G')$ となる。

4.5 SG-DP のサブクラスに対する Q 学習

$\mathcal{G}$ を SG-DP のサブクラスでもとの文法が統計なしで学習可能な文法のクラスとする。$G = \langle N, \Sigma, R, S \rangle \in \mathcal{G}^-$ に対して、$\boldsymbol{Q}$(G,U) を $N \times U$ から $\mathbb{R}$ へ

```
Input G,Q, U, w_1, ··· ,w_m
  if( G ≠ G̃(w_1, ··· , w_m) ){
    G := G̃(w_1, ··· , w_m);
    Q := Q(G,U);
  }
  w := ε:
  while( (a := Env(u)) ≠ ε ){
    w := wa
    if(S ⇒* [G]wα){
      Q := SG-QL の更新式;
    }
    u をランダムに選ぶ;
  }
  output G,Q, U, w_1, ··· , w_m, w and halt
```

図 13: SG-DP に対するもとの文法を未知としたときの強化学習

の射とする。図 13 のような繰り返しのアルゴリズムを考える。ここで、Env は $G_{U,P,C} \in \mathcal{G}$ と最左導出とで定められる MDP で表現される環境で、行動を入力として、終端記号または ε を出力するものとする。ε は文の終わりをあらわすものとする。また、$\widetilde{G}$ は $\mathcal{G}^-$ を学習可能なアルゴリズムであるとする。

補題 G,H を既約な SG とする。$\mathbb{K}(G) \subset \mathbb{K}(H)$ ならば任意の $G_{U,P,C}$ に対して、それと等価な $H_{U',P',C'}$ が存在する。ここでいう等価とは、

1. $L(G) = L(H)$
2. $U = U'$
3. 任意の $xay \in L(G)$ および任意の $u \in U$ に対して、$P(r(xa), u) = P'(r'(xa), u)$ および $C(r(xa), u) = C'(r'(xa), u)$。
   ここで、$r(xa)$,$r'(xa)$ をそれぞれ、$S \overset{*}{\Rightarrow} [G]yA\alpha \overset{*}{\Rightarrow} [G]ya\alpha'$、$S \overset{*}{\Rightarrow} [H]yB\beta \overset{*}{\Rightarrow} [H]ya\beta'$ として、$r(xa) = A \to a\gamma \in R$、$r'(xa) = B \to a\gamma' \in R'$ とおく。

Proof. 系 4.2 より $\mathbb{K}(G) = \mathbb{K}(H)$、また、$U' = U$ ととる。

H 既約より、任意の $s \in R'$ について、$s = r'(ya)$、$yaz \in L(G)$ となる ya が存在する。補題 4.2 より、$r'(xa) = r'(ya)$、$xaw \in L(G)$ となる任意の xa について、$r(xa) = r(ya)$ である。したがって、P' と C' を全ての u について $P'(r'(ya), u) = P(r(ya), u)$、$C'(r'(ya), u) = C(r(ya), u)$ と定義できる。 □

$G_{U,P,C}$ を SG-DP とし、$G_{U,P,C}$ と最左導出順によって決まる MDP を考える。SG の性質より $T_1 = S$ としたとき、任意の $P(X_i = T_i) \neq 0$ となる T_i に対して、$xy \in L(G)$ となる x が一対一に対応する。したがって、方策を

$\mu : \mathcal{T} \to U$ の変わりに $\widetilde{\mu} : \{x \in \Sigma^* | \exists y, xy \in L(G)\} \to U$ であらわすことができる。 このとき、任意の静的方策について既約かつ適正ならば、$H_{U',P',C'}$ を $G_{U,P,C}$ と等価な SG-DP とすると、$\widetilde{\mu}^*$ が $G_{U,P,C}$ の最適方策であることと $H_{U',P',C'}$ の最適方策であることとは同値である。

定理 $\mathcal{G}$ をSG-DP のサブクラスでもとの文法が統計なしで学習可能な文法のクラスとする。図13 のアルゴリズムを A として、$(G_t, \boldsymbol{Q}_t, w_1, w_2, \cdots, w_t) = A^t(G_0, \boldsymbol{Q}(G_0), \varepsilon)$ とする。Env をあらわす $G_{U,P,C} \in \mathcal{G}$ が任意の $\mu \in M$ で既約でかつ適正ならば、$\boldsymbol{Q}_t$ は任意の G_0,$\boldsymbol{Q(G_0)}$ について、$t \to \infty$ で $G_{U,P,C}$ と等価な $G'_{U',P',C'}$ の最適行動価値に収束する。

Proof. まず、任意の $\mu \in M$ が既約かつ整合的だから、補題 4.2 より $(\boldsymbol{Q}_t, w_1, w_2, \cdots)$ は正の例である。従って、$\mathcal{G}$ は学習可能だから、G_t は $L(G) = L(G^*)$ となる G^* に収束する。

さらに、$\mathcal{G}$ がもとの文法が統計なしで学習可能なことより $\mathbb{K}(G) \subset \mathbb{K}(G^*)$ だから、補題 4.5 より $G_{U,P,C}$ と等価な $G^*_{U',P',C'}$ が存在する。従って、G^* に収束した後は定理 3.4.1 より $G^*_{U',P',C'}$ の最適行動価値に収束する。 □

したがって、NMSG-DP と NMRSG-DP には、等価な決定過程化した文法の最適行動価値に収束するアルゴリズムが存在する。

実際に、NMRSG-DP の場合には、効率を考えて、アルゴリズムに以下のような工夫をして計算を行う。まず、文法学習のアルゴリズムは 2.4 章で述べた RSG の学習アルゴリズムを用いる。RSG の学習アルゴリズムは常に標準形を出力するので、補題 4.3 より、図 13 のアルゴリズムは学習対象の NMRSG-DP と等価な RSG-DP の最適行動価値に収束する。

文法候補であるが、$\dim \ker(M_t)$ をあらかじめ k 未満になるまで列挙しないと決めておく。実際の文法で $\dim \ker(M_*)$ が k 以上ならばこのアルゴリズムはうまくいかないことになるが、その場合は k を徐々に増やすなどすれば $\dim \ker(M_*)$ は $|\Sigma|$ 以下であるので収束させることができる。

次に、文法候補が決定していない、または破綻してしまった場合は、ランダムに行動せずに、MDP だと思って破綻してないときと同じ戦略を用いて行動することとする。

以上より、実際に用いたアルゴリズムの手順はおおよそ次のようになる。

1. 文が終るまで MDP だと仮定して行動。
2. $\dim \ker(M_t) \geq k$ ならば 1 へ。
3. RSG の学習アルゴリズムを用いて極小の文法 G_t を計算。
4. G_t の形状にあわせて $\boldsymbol{Q}$ のテーブルの数を調整。
5. G_t および SG-QL の更新式を用いて $\boldsymbol{Q}$ を更新。
6. 文法候補が破綻すれば、MDP だと仮定して行動。
7. 文が終われば 2 へ。

4.6 もとの文法を未知とした場合の強化学習の例

図 14 のような迷路を考える。3.5 節の迷路とほぼ同じで、異なる点は、ゴールに達すると +100 の報酬を得て、ゴールの後に、地点 B を通った回数だけ状態 ‘bad’ が発生してその回数だけ −50 の報酬を得る。

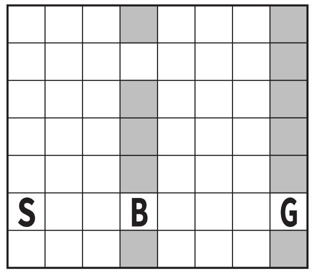

図 14: 例:迷路問題 2

このときの文法 $G = \langle N, \Sigma, R, S\rangle$ は、まず、観測できる事柄はどの地点にいるかおよびどちらを通ったかであるから、

$$\Sigma = \{(i,j)|(i,j)\text{ は図の壁でない地点 }\} \cup \{\text{‘bad’}\}$$

である。次に生成規則を考える。この場合、地点 B を除いては観測状態と同じであると見なして良い。しかし、地点 B が観測された際には、状態 ‘bad’ を生成するので、

$$\begin{aligned} R = \{&\{A_{(i\pm1,j)}, A_{(i,j\pm1)}\} \to (i,j)A_{(i,j)}|(i,j) \in \Sigma - \{(6,4)\}\} \\ &\cup \{\{A_{(6,3)}, A_{(6,4)}, A_{(6,5)}\} \to (6,4)A_{(6,4)}B,\ B \to \text{‘bad’}\} \end{aligned}$$

となる。これより、任意の $A, B \in N$,$A \neq B$ に対して $U(A) \neq U(B)$ がわかるから、NMRSG-DP のクラスに属することがわかる。文法は図 15 左のような形状をしている。3.5 節の迷路問題の例では形状含めた文法自身を学習者があらかじめ知っているということを前提としての強化学習であったが、この節では学習者が、形状を含めて文法を全く未知とし、単に NMRSG-DP のクラスに属することのみを前提条件として学習を行う。

実験の結果について述べる。学習パラメータとしては、割引率を $\gamma = 1$、学習係数を $\alpha = 0.2$、方策には ε-グリーディ方策を用いて $\varepsilon = 0.2$、$\dim\ker(M_t) < 3$ とした。

まず、もとの文法の推定だが、$t = 50$ で初めて $\dim\ker(M_t) < 3$ となり文法の候補を計算したが、候補は結局 1 つのみである。さらに、最終的には $\dim\ker(M_t) = 0$ となっており、自由度の観点からも異なる形状をもつ等価な文法が存在しないことがわかる (図 17)。また、最終的に学習した文法 (図 16) は学習対象の文法と等価である。実際、B を通らない場合を考えるとま

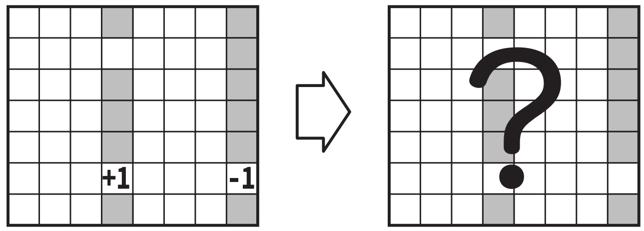

図 15: 迷路問題 2 における文法の形状

ず $\#(G)=-1$ であり、その他の場所は回数に関係なく $G1$ つで終わることから 0 であり、$\#(B)=1$ となるから、もと文法の形状以外はとり得ない。

次に、強化学習の結果であるが、エピソードの長さや総報酬は、文法が推定されるまで (図 18 上の矢印まで) は不安定だが、推定されてからは速やかに最大の総報酬およびそれを与えるパスの長さに収束していることがわかる。最終的に通った地点の履歴、エピソードの長さ、総報酬の値は表 2 のようになった。

また、20 から 30 エピソードに注目すると、エピソードの長さが短いにもかかわらず総報酬の値は最大の総報酬よりも 50 程度低く、B を通っていることが推測される。これは、文法の形状がわかるまでは MDP と考えて行動しているためであり、3.5 節の結果と同様である。

表 2: 最終的な学習結果

履歴	$(6,1)(5,1)(4,1)(4,2)(3,2)(3,3)(2,3)(2,4)$ $(2,5)(3,5)(4,5)(5,5)(6,5)(6,6)(6,7)(6,8)$
エピソードの長さ	16
総報酬	93

単純文法によって拡張した強化学習について論じた。

決定過程のもととなる文法を既知とした場合、SG のクラスで QL の拡張した SG-QL のアルゴリズムで強化学習が可能である。

もととなる文法を未知とした場合、文法も同時に学習しなければならず、SG や RSG のクラスでは統計なしで学習はできないことがわかったが、一方で、統計なしで学習できる単純文法のサブクラスとして、NMSG、NMRSG のクラスがあることを示した。

実際に効率的に問題が解けるアルゴリズムとして、NMRSG-DP 環境下における SG-QL を考えた。そのアルゴリズムでは RSG の学習アルゴリズムで枠組みとなっている文法を推定しながら SG-QL の更新式で学習を行うわけだ

$white(1,1)_0 S \rightarrow \{(6,1)\}$
$(1,1)_0 \rightarrow \{(1,1),(1,2),(2,1)\}$
$(1,2)_0 \rightarrow \{(1,1),(1,2),(1,3),(2,2)\}$
$(1,3)_0 \rightarrow \{(1,2),(1,3),(2,3)\}$
$(1,5)_0 \rightarrow \{(1,5),(1,6),(2,5)\}$
$(1,6)_0 \rightarrow \{(1,5),(1,6),(1,7),(2,6)\}$
$(1,7)_0 \rightarrow \{(1,6),(1,7),(2,7)\}$
$(2,1)_0 \rightarrow \{(1,1),(2,1),(2,2),(3,1)\}$
$(2,2)_0 \rightarrow \{(1,2),(2,1),(2,3),(3,2)\}$
$(2,3)_0 \rightarrow \{(1,3),(2,2),(2,4),(3,3)\}$
$(2,4)_0 \rightarrow \{(2,3),(2,4),(2,5)\}$
$(2,5)_0 \rightarrow \{(1,5),(2,4),(2,6),(3,5)\}$
$(2,6)_0 \rightarrow \{(1,6),(2,5),(2,7),(3,6)\}$
$(2,7)_0 \rightarrow \{(1,7),(2,6),(2,7),(3,7)\}$
$(3,1)_0 \rightarrow \{(2,1),(3,1),(3,2),(4,1)\}$
$(3,2)_0 \rightarrow \{(2,2),(3,1),(3,3),(4,2)\}$
$(3,3)_0 \rightarrow \{(2,3),(3,2),(3,3),(4,3)\}$
$(3,5)_0 \rightarrow \{(2,5),(3,5),(3,6),(4,5)\}$
$(3,6)_0 \rightarrow \{(2,6),(3,5),(3,7),(4,6)\}$
$(3,7)_0 \rightarrow \{(2,7),(3,6),(3,7),(4,7)\}$
$(4,1)_0 \rightarrow \{(3,1),(4,1),(4,2),(5,1)\}$
$(4,2)_0 \rightarrow \{(3,2),(4,1),(4,3),(5,2)\}$
$(4,3)_0 \rightarrow \{(3,3),(4,2),(4,3),(5,3)\}$
$(4,5)_0 \rightarrow \{(3,5),(4,5),(4,6),(5,5)\}$
$(4,6)_0 \rightarrow \{(3,6),(4,5),(4,7),(5,6)\}$
$(4,7)_0 \rightarrow \{(3,7),(4,6),(4,7),(5,7)\}$
$(5,1)_0 \rightarrow \{(4,1),(5,1),(5,2),(6,1)\}$
$(5,2)_0 \rightarrow \{(4,2),(5,1),(5,3),(6,2)\}$
$(5,3)_0 \rightarrow \{(4,3),(5,2),(5,3),(6,3)\}$
$(5,5)_0 \rightarrow \{(4,5),(5,5),(5,6),(6,5)\}$
$(5,6)_0 \rightarrow \{(4,6),(5,5),(5,7),(6,6)\}$
$(5,7)_0 \rightarrow \{(4,7),(5,6),(5,7),(6,7)\}$
$(6,1)_0 \rightarrow \{(5,1),(6,1),(6,2),(7,1)\}$
$(6,2)_0 \rightarrow \{(5,2),(6,1),(6,3),(7,2)\}$
$(6,3)_0 \rightarrow \{(5,3),(6,2),(6,4),(7,3)\}$
$(6,4)_0 \rightarrow \{(6,3),(6,4),(6,5)\}$
$(6,4)_1 \rightarrow \{(6,8)\}$
$(6,5)_0 \rightarrow \{(5,5),(6,4),(6,6),(7,5)\}$
$(6,6)_0 \rightarrow \{(5,6),(6,5),(6,7),(7,6)\}$
$(6,7)_0 \rightarrow \{(5,7),(6,6),(6,8),(7,7)\}$
$(7,1)_0 \rightarrow \{(6,1),(7,1),(7,2)\}$
$(7,2)_0 \rightarrow \{(6,2),(7,1),(7,2),(7,3)\}$
$(7,3)_0 \rightarrow \{(6,3),(7,2),(7,3)\}$
$(7,5)_0 \rightarrow \{(6,5),(7,5),(7,6)\}$
$(7,6)_0 \rightarrow \{(6,6),(7,5),(7,6),(7,7)\}$
$(7,7)_0 \rightarrow \{(6,7),(7,6),(7,7)\}$

図 16: 学習した文法

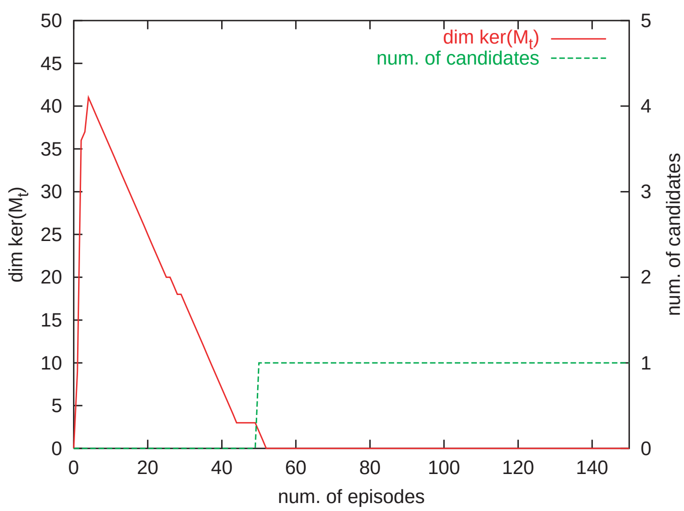

図 17: $\dim \ker(M_t)$ と文法候補の数の変化の様子

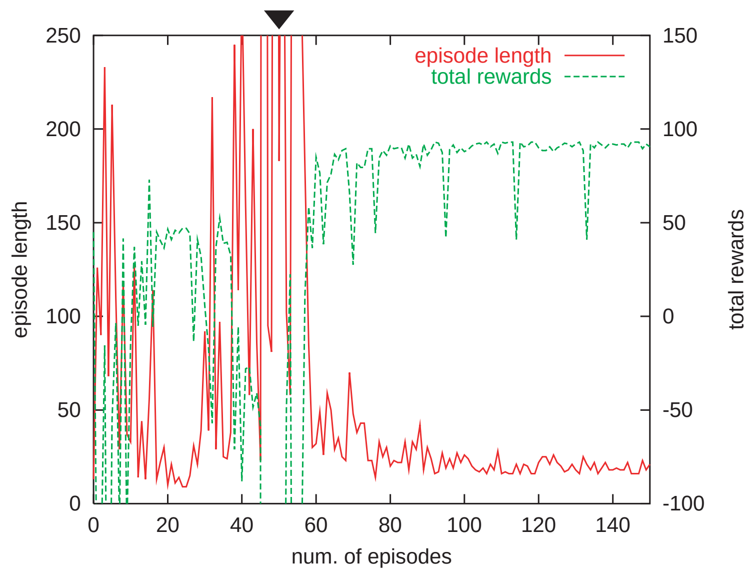

図 18: エピソードの長さと総報酬の変化の様子

が、もとの文法の持つ固有の自由度 $\dim\ker(M_*)$ があって、RSG 学習の計算量は $\dim\ker(M_*)$ で大きく左右され、ある k をとって $\dim\ker(M_t) < k$ となるように制限するともとの文法のサイズとエピソードの長さについて多項式時間で計算可能となることから、現実的な時間で計算することが可能である。

NMRSG-DP のクラスを仮定したときの問題点としては、SG-DP、RSG-DP のクラスが、文法の右辺の非終端記号の数を 1 つ以下に制限することで、有限状態の既約な MDP を包含していることに対して、NMRSG-DP のクラスはそれを包含していない。また、$\dim\ker(M_*) < k$ となる k は一般には未知であり、また、終端記号の個数に対して線形になるような NMRSG も存在するため、結局、文法のサイズに対して指数時間になる可能性もある。しかし、NMRSG-DP のクラスでも、例に示した迷路問題のような、文法の構造が比較的簡単なものでは、少ない $\dim\ker(M_*)$ を持つものが多数あると考えられ、実際には k は計算量の削減に大きく貢献している。

応用面では、強化学習の問題でタスクが複数に分岐して各タスクが独立な問題があって、その問題が NMRSG-DP に属していれば、どこでそのタスクが分岐するかを学習しながら、最終的には最適な行動を効率的に学習することができ、ロボットのより知能的な行動学習などでの応用が期待できる。

また、将来の展望としては、本論文では枠組みとなる文法を学習する際にあえて統計的な情報を無視して学習できるクラスを考えたが、終端記号の出現頻度などを考慮することで、より広いクラスに適用することができると考えている。

5 単純文法の確率的一般性と統合可能性

In grammatical inference in the limit from positive data, there is a trade-off between the richness of the language class and the efficiency of the algorithm. Although some general conditions on learning grammars from only positive data have been found [1, 2] and are well-known, these conditions only establishes the existence of a learning algorithm, and does not say anything about its efficiency. Preceding research has proposed several efficient algorithms that identify some subclasses of context-free languages [2, 6]. In particular, recent studies [10, 12, 13] have found some nonregular context-free languages that are efficiently learnable from positive data. Yoshinaka has proposed a polynomial-time algorithm that learns a subclass of context-free grammars, called right-unique simple grammars (RSGs) [13], which is a superclass of very simple grammars (VSGs) found as a efficiently learnable class by Yokomori [12].

Both the classes of RSGs and VSGs are subclasses of simple grammars (SGs). In this paper, we consider the properties and the unification methods of the subclasses of probabilistic simple grammars. In learning these subclasses from positive examples, since if the grammar is not probabilistic, the problem becomes the classical problem of grammatical inference from positive data, it may seem that there is no problem: first infer the target grammar from positive data, and then determine the probabilities of production rules by using a statistical method. However, this solution is not sufficient because although the inferred grammar generates the correct language, there is not necessarily some probability assignment of production rules on the inferred grammar such that it generates the correct probabilistic language. For example, let us consider CFGs G and G' whose rules are $\{S \to aS|b\}$ and $\{S' \to aA'|b, A' \to aA'|b\}$ respectively. Then it is obviously impossible for G to generate the same probabilistic language as G' if $\Pr(S' \to aA') \neq \Pr(A' \to aA')$, although $L(G) = L(G')$.

In Section 3, we introduce the notion of the probabilistic generality of simple grammars (SGs), where the class of SGs is a superclass of RSGs and VSGs. Probabilistic generality of a grammar is defined as the set of the probabilistic languages generated by probabilistic grammars that are obtained by assigning probabilities to the production rules of the grammar. We show that, for the class of SGs and the class of RSGs, there exist two grammars whose languages are equivalent, and for which the probabilistic generality of any grammar in the same class is not larger than both of them.

In Section 4, a new subclass of SGs called unifiable simple grammars (USGs) is introduced. The class of USGs is a superclass of RSGs. We show that for any two USGs that generate the same language, there is a USG whose probabilistic generality is larger than the two. This implies that all RSGs whose languages are equivalent can be unified to one USG, since the number of those RSGs is finite.

In Section 5, we give an application for which the results of this paper are required. We introduce context-free decision processes, which are an extension of finite Markov decision processes (MDPs), and introduce a modified Q-learning algorithm for their optimisation. A simple context-free decision process intuitively may be thought of a finite MDP with stacks. The class of RSGs is sufficiently large so that context-free decision processes based on RSGs include all episodic finite MDPs. We use Yoshinaka's learning method to output all the minimal grammars that can generate the histories, then construct a USG by unifying the output RSGs, and use the extended Q-learning for learning optimal decisions.

5.1 準備

First, we outline some standard notation and definitions.

A context-free grammar is denoted by $\langle V, \Sigma, R, S\rangle$, where V is a finite set of nonterminal symbols, Σ is a finite set of terminal symbols, $R \subset V \times (V \cup \Sigma)^*$ is a finite set of production rules and $S \in V$ is the start symbol. Let $G = \langle V, \Sigma, R, S\rangle$ be a CFG. We write $XAZ \Rightarrow_G XYZ$ iff $A \to Y \in R$ and $X, Z \in (V \cup \Sigma)^*$, and $\Rightarrow_G^*$ denotes the reflective and transitive closure of $\Rightarrow_G$. When G is clearly identified, we write simply $\Rightarrow$ instead of $\Rightarrow_G$. G is said to be reduced iff for all $A \in V$, there are some $x, y, z \in \Sigma^*$ such that $S \overset{*}{\Rightarrow} xAz \overset{*}{\Rightarrow} xyz$. The language of G, $L(G)$,is defined as $\{x \in \Sigma^* \mid S \overset{*}{\Rightarrow} x\}$. Let $L(G, X) = \{x \in \Sigma^* \mid X \overset{*}{\Rightarrow} x\}$, where $X \in (V \cup \Sigma)^*$. When G is clearly identified, we write simply $L(X)$ instead of $L(G, X)$. For $A \in V$, let R_A indicate $\{A \to X \in R\}$.

Let ε denote the empty sequence and $|x|$ denote the length of a sequence x. For a set V, let $|V|$ denote the number of the elements in V. For a CFG $G = \langle V, \Sigma, R, S\rangle$, let $|G|$ denote $\sum_{A \to X \in R} |AX|$.

Hereafter, let terminal symbols and nonterminal symbols be denoted by $a, b, c, \cdots$ and $A, B, C, \cdots$ respectively, and finite sequences of terminals symbols and of nonterminal symbols be denoted by $\cdots, x, y, z$ and $\alpha, \beta, \gamma, \cdots$ respectively.

Subclasses of CFGs we discuss in this paper are defined below.

定義 1. *A CFG $G = \langle V, \Sigma, R, S\rangle$ is called a simple grammar (SG) iff G is*

Greibach normal form, and

$$A \to a\alpha \in R \text{ and } A \to a\beta \in R \text{ imply } \alpha = \beta.$$

An SG G is called a right-unique simple grammar (RSG) iff

$$A \to a\alpha \in R \text{ and } B \to a\beta \in R \text{ imply } \alpha = \beta.$$

An SG G is called a very simple grammar (VSG) iff

$$A \to a\alpha \in R \text{ and } B \to a\beta \in R \text{ imply } \alpha = \beta \text{ and } A = B.$$

An RSG G is normal form iff it is reduced, and for all $C \in V$, $A \to a\alpha C\beta, B \to a'\alpha' C\beta' \in R$ implies $a = a'$, $\alpha = \alpha'$, $\beta = \beta'$ and $C \neq S$.

CFGs in GNF $G = \langle V, \Sigma, R, S\rangle$ and $H = \langle V', \Sigma, R', S'\rangle$ are equivalent modulo renaming nonterminals iff there is a bijection $\phi : V \to V'$ such that $\phi(S) = S'$, $A \to a\alpha \in R$ iff $\phi(A) \to a\widehat{\phi}(\alpha) \in R'$ where $\widehat{\phi}$ is the unique homomorphic extension of ϕ.

While the class of SGs is not learnable in the limit from positive data, for both the class of VSGs and the class of RSGs, there are the efficient learning algorithms, which satisfy conservativeness and consistency and output grammars in polynomial time in the size of the input positive examples.

Those algorithms for VSGs and RSGs are based on the following strategy. Let $\mathcal{C}$ be either the class of VSGs or the class of RSGs. Let positive presentation of the target grammar in $\mathcal{C}$ be $s_1, s_2, \cdots$, and output grammars be $G_1, G_2, \cdots$. For each i-th input of positive data, if s_i is in $L(G_{i-1})$ then $G_i := G_{i-1}$, otherwise $G_i := G$, where $G \in \mathcal{C}$ such that $\{s_1, \cdots, s_i\} \subset L(G)$ and $L(G)$ is minimal, namely, $\forall G' \in \mathcal{C}[L(G') \subsetneq L(G)$ implies $\{s_1, \cdots, s_i\} \not\subset L(G')]$. C has finite thickness, namely, for any finite language $D = \{s_1, \ldots, s_i\}$, at most finitely many (modulo renaming nonterminals) grammars G in C generate a language including D.

A function $\#_G : \Sigma^* \to \{-1, 0, \cdots\}$ for $G = \langle V, \Sigma, R, S\rangle \in \mathcal{C}$ is defined as $\#_G(\varepsilon) = 0$, $\#_G(a) = |\alpha| - 1$, where $A \to a\alpha \in R$ for some $A \in V$, and $\#(ax) = \#(a) + \#(x)$. Note that $\#(a)$ is well-defined due to the definition of the class C. Since $D \subset L(G)$ implies that $\#_G(s) = -1$ for all $s \in D$ and $\#_G(t) \geq 0$ for each proper prefix t of s, the number of possible #s for D is finite. When a possible # is given, it is easy to determine the minimal grammar in $\{G \in \mathcal{C} \mid \#_G = \#\}$. The algorithm outputs a minimal grammar among those minimal grammars.

Although Yoshinaka's algorithm can decide the inclusion of every two RSGs G and H in polynomial time in $|G|+|H|$, since the number of possible

#s can be exponential in $|\Sigma|$, those algorithm is also in exponential time in $|\Sigma|$.

Let $G = \langle V, \Sigma, R, S\rangle$ be an SG. A probability assignment P on G is a map from R to $[0,1]$ such that $\sum_{r\in R_A} P(r) = 1$ for all $A \in V$, where $R_A = \{A \to a\alpha \in R\}$. A probabilistic simple grammar (PSG) is a pair $\langle G, P\rangle$, where P is probability assignment on an SG G. $\langle G, P\rangle$ is reduced iff G is reduced and $P(r) \neq 0$ for all $r \in R$.

When G is an SG, every $x \in L(G)$ has a unique sequence of production rules that are used in the left-most derivation of $S \overset{*}{\Rightarrow}_G x$. Let us denote that sequence by $r(G, x, 1), \cdots, r(G, x, |x|)$. Then, the probabilistic language of a PSG $\langle G, P\rangle$, $\Pr(\,\cdot\,|\langle G, P\rangle) : \Sigma^* \to [0,1]$, is defined as

$$\Pr(x|\langle G, P\rangle) = \begin{cases} \prod_{i=1}^{|x|} P(r(G, x, i)) & \text{if } x \in L(G), \\ 0 & \text{otherwise.} \end{cases}$$

We define similarly that $\Pr(x|\langle G, P\rangle, A) = \prod_{i=1}^{|x|} P(r(A, x, i))$ if $x \in L(G, A)$, otherwise 0, where $r(G, A, x, 1) \cdots r(G, A, x, |x|)$ are the sequence of rules used in the derivation $A \overset{*}{\Rightarrow}_G x$.

5.2 確率的一般性について

定義 2. *The generality of an SG G is defined as*

$$\mathbb{K}(G) = \{\Pr(\cdot|\langle G, P\rangle) | P \textit{ is a probability assignment on } G\}.$$

G is more general than an SG H iff $\mathbb{K}(G) \subset \mathbb{K}(H)$.

The following lemma establishes requirements for $\mathbb{K}(G) \subset \mathbb{K}(H)$.

補題 1. *Let $G = \langle V, \Sigma, R, S\rangle$ and $H = \langle V', \Sigma, R', S'\rangle$ be reduced SGs. $\mathbb{K}(G) \subset \mathbb{K}(H)$ iff $L(G) = L(H)$ and there is some map $\psi : V'_{\geq 2} \to V$ such that $\forall A \in V'_{\geq 2} \forall x \in \Sigma^*$ [$S' \overset{*}{\Rightarrow}_H xA\alpha$ implies $S \overset{*}{\Rightarrow}_G x\psi(A)\beta$], where $V'_{\geq 2} = \{A \in V' \mid |R'_A| \geq 2\}$.*

Proof. We first show the "if" part. Let G, H, and ψ be as above. It is easy to see that for each $A \to a\alpha \in R'$ with $A \in V'_{\geq 2}$, G has exactly one rule of the form $\psi(A) \to a\beta$ for some $\beta \in V^*$. For a given probability assinment P on G, define a probability assinment Q on H by

$$Q(A \to a\alpha) = \begin{cases} P(\psi(A) \to a\beta) & \text{if } A \in V'_{\geq 2} \\ 1 & \text{otherwise.} \end{cases}$$

Clearly $\Pr(\cdot|\langle G, P\rangle) = \Pr(\cdot|\langle H, Q\rangle)$.

Second we show the "only if" part. Suppose that $\mathbb{K}(G) \subset \mathbb{K}(H)$ holds for two reduced SGs G and H. If $\langle G, P \rangle$ is a consistent PSG, there is a consistent probability assignment Q on H such that $\Pr(\cdot|\langle G, P\rangle) = \Pr(\cdot|\langle H, Q\rangle)$. We have $L(G) = L(H) = \{ w \in \Sigma^* \mid \Pr(w|\langle G, P\rangle) = \Pr(w|\langle H, Q\rangle) \neq 0 \}$.

To derive a contradiction, suppose that $L(G) = L(H)$ and there are some $x, y \in \Sigma^*$ such that $S' \overset{*}{\Rightarrow}_H xA\alpha_x$, $S' \overset{*}{\Rightarrow}_H yA\alpha_y$, $S \overset{*}{\Rightarrow}_G xB\beta_x$, $S \overset{*}{\Rightarrow}_G yC\beta_y$, $|R_A| \geq 2$ and $B \neq C$. Since $L(G) = L(H)$, $\forall a \in \Sigma$ [$\exists \gamma_A [A \to a\gamma_A \in R]$ iff $\exists \gamma_B [B \to a\gamma_B \in R']$] holds and thus $|R_B| = |R_C| = |R'_A| \geq 2$. Let $A \to a\gamma_A \in R'$, $B \to a\gamma_B \in R$, $C \to a\gamma_C \in R$. For $|R'_A| \geq 2$, a consistent probability assignment P can be found such that $P(B \to a\gamma_B) \neq P(C \to a\gamma_C)$. Let Q be a consistent probability assignment on H such that $\Pr(\cdot|\langle G, P\rangle) = \Pr(\cdot|\langle H, Q\rangle)$. Then,

$$P(B \to a\gamma_B) = \frac{\Pr(xa\Sigma^*|\langle G, P\rangle)}{\Pr(x\Sigma^*|\langle G, P\rangle)} = \frac{\Pr(xa\Sigma^*|\langle H, Q\rangle)}{\Pr(x\Sigma^*|\langle H, Q\rangle)}$$
$$= Q(A \to a\gamma_A) = \frac{\Pr(ya\Sigma^*|\langle H, Q\rangle)}{\Pr(y\Sigma^*|\langle H, Q\rangle)} = \frac{\Pr(ya\Sigma^*|\langle G, P\rangle)}{\Pr(y\Sigma^*|\langle G, P\rangle)} = P(C \to a\gamma_C).$$

This is a contradiction. □ □

定義 3. *Let $\mathcal{C}$ and $\mathcal{D}$ be subclasses of SGs. $\mathcal{C}$ is unifiable within $\mathcal{D}$ iff for all $G_1, G_2 \in \mathcal{C}$ such that $L(G_1) = L(G_2)$, there is $H \in \mathcal{D}$ such that $\mathbb{K}(G_1) \cup \mathbb{K}(G_2) \subset \mathbb{K}(H)$.*

The main purpose of this paper is to construct an SG G_* that is more general than a finite number of given RSGs whose languages are equivalent. However, neither the class of SGs nor the class of RSGs is unifiable within itself, as we demonstrate in what follows. In the following, we say that $\mathcal{C}$ is unifiable when $\mathcal{C}$ is unifiable within $\mathcal{C}$.

命題 1. *The class of SGs is not unifiable.*

Proof. Let $G = \langle V, \Sigma, R, S\rangle$ and $G' = \langle V', \Sigma, R', S'\rangle$ be SGs, whose rules are, respectively,

$\{S \to aAB,\ A \to aB|b|c,\ B \to aAB|bC_1|cC_2,\ C_1 \to aB|b|c,\ C_2 \to aB|b|c\}$ and
$\{S' \to aB'A',\ A' \to aB'|b|c,\ B' \to aA'B'|bC'_1|cC'_2,\ C'_1 \to aB'|b|c,\ C'_2 \to aB'|b|c\}$.

First we show that $L(G) = L(G')$. G and G' are isomorphic if we disregard the rules $S \to aAB$ and $S \to aB'A'$. Clearly

$$L(A) = L(C_1) = L(C_2) = L(A') = L(C'_1) = L(C'_2),\ \text{and}\ L(B) = L(B').$$

Moreover, it is not hard to see that for every $x \in \Sigma^*$ and $\gamma \in \{A\}^*$, the following are equivalent:

- $AA \stackrel{*}{\Rightarrow}_G x\alpha$ with $\phi(\alpha) = \gamma$ for some $\alpha \in V^*$,
- $B \stackrel{*}{\Rightarrow}_G x\beta$ with $\phi(\beta) = \gamma$ for some $\beta \in V^*$,

where $\phi : V^* \to \{A\}^*$ is the homomorphism such that $\phi(A) = \phi(C_1) = \phi(C_2) = A$ and $\phi(B) = AA$. Therefore, $L(AA) = L(B) = L(B') = L(A'A')$, and thus $L(S) = L(S')$.

Second, we show that no SG H is more general than both G and G'. Let $H = \langle V_H, \Sigma, R_H, S_H \rangle$ be an SG such that $L(H) = L(G) = L(G')$. Since $a^{2n}b^{2n+2} \in L(G)$, there are $D_n \in V_H$ and $\alpha_n \in V_H^*$ such that

$$S_H \stackrel{*}{\Rightarrow}_H a^{2n} D_n \alpha_n \stackrel{*}{\Rightarrow} a^{2n} b^{2n+2}$$

for each $n \in \mathbb{N}$. Since V_H is finite, we can find $m, n \in \mathbb{N}$ such that $m < n$ and $D_m = D_n$. Let k and E be such that $D_m = D_n \stackrel{*}{\Rightarrow}_H b^{k-1}E \Rightarrow b^k$, $\alpha_m \stackrel{*}{\Rightarrow}_H b^{2m+2-k}$ and $\alpha_n \stackrel{*}{\Rightarrow}_H b^{2n+2-k}$. Note that $k \leq 2m+2 < 2n+2$. Since $a^{2n}b^{k-1}cb^{2n+2-k} \in L(G) = L(H)$, we have $E \to c\gamma \in R$ and

$$S_H \stackrel{*}{\Rightarrow}_H a^{2n} D_n \alpha_n \stackrel{*}{\Rightarrow} a^{2n} b^{k-1} E \alpha_n \Rightarrow a^{2n} b^{k-1} c\gamma\alpha_n \stackrel{*}{\Rightarrow} a^{2n} b^{k-1} c b^{2n+2-k}.$$

Since H is an SG, $b^{2n+2-k} \in L(\gamma\alpha_n) \cap L(\alpha_n)$ implies $\gamma = \varepsilon$. Therefore, we have

$$S_H \stackrel{*}{\Rightarrow}_H a^{2n} D_n \alpha_n \stackrel{*}{\Rightarrow} a^{2n} b^k \alpha_n, \qquad S_H \stackrel{*}{\Rightarrow}_H a^{2n} D_n \alpha_n \stackrel{*}{\Rightarrow} a^{2n} b^{k-1} c\alpha_n.$$

Since $k < 2n+2$, $\alpha_n \neq \varepsilon$. If $k = 2j+1 < 2n+2$, then

$$S \stackrel{*}{\Rightarrow}_G a^{2n} b^k C_1 B^{n-j}, \qquad S \stackrel{*}{\Rightarrow}_G a^{2n} b^{k-1} c C_2 B^{n-j}.$$

By Lemma 1, H is not more general than G. Similarly, if $k = 2j+2 < 2n+2$, then H is not more general than G'. $\square$

The class of RSGs is also not unifiable. Let us consider the finite language $L = (a|b)(c|d)(e|f) = \{ace, acf, ade, adf, bce, bcf, bde, bdf\}$. In normal form, any RSG that generates L is equivalent, modulo renaming nonterminals, to either $G = \langle V, \Sigma, R, S \rangle$ or $H = \langle V', \Sigma, R', S \rangle$, whose rules are, respectively,

$$\{S \to aA|bB,\ A \to cC|dD,\ B \to cC|dD,\ C \to e|f,\ D \to e|f\} \text{ or}$$
$$\{S \to aA_0A_1|bB_0B_1,\ A_0 \to c|d,\ B_0 \to c|d,\ A_1 \to e|f,\ B_1 \to e|f\}.$$

$|R_A| = |R'_{A'}| = 2$ for all $A \in V$ and $A' \in V'$. $S \stackrel{*}{\Rightarrow}_G acC$ and $S \stackrel{*}{\Rightarrow}_G adD$, while $S \stackrel{*}{\Rightarrow}_H acA_1$ and $S \stackrel{*}{\Rightarrow}_H adA_1$. Thus $\mathbb{K}(G) \not\subset \mathbb{K}(H)$ from Lemma 1. On the other hand, $S \stackrel{*}{\Rightarrow}_G acC$ and $S \stackrel{*}{\Rightarrow}_G bcC$, while $S \stackrel{*}{\Rightarrow}_H acA_1$ and $S \stackrel{*}{\Rightarrow}_H bcB_1$. Thus $\mathbb{K}(H) \not\subset \mathbb{K}(G)$. It follows that there is no RSG I such that $L(I) = L$, $\mathbb{K}(G) \subset \mathbb{K}(I)$ and $\mathbb{K}(H) \subset \mathbb{K}(I)$ from Lemma 9.

5.3 単純文法のサブクラスにおける統合可能性

In this section, we introduce unifiable simple grammars (USGs). The class of USGs is unifiable and is a superclass of the class of RSGs. This implies that the class of RSGs is unifiable within the class of USGs. This is the main result of this paper.

Let $G = \langle V, \Sigma, R, S \rangle$ be an SG. Let $\sigma_G(A) = \{a \in \Sigma \mid A \to a\alpha \in R\}$ for $A \in V$. We write AB iff $\sigma_G(A) = \sigma_G(B)$. is an equivalence relation, thus let $\overline{A}$ denote the equivalence class containing A, i.e., $\overline{A} = \{A' \in V \mid A'A\}$. We also introduce the notation $\overline{U} = \{A' \in V \mid \exists A \in U, A' \in \overline{A}\}$ and $\overline{A_1 \cdots A_m} = \overline{A_1} \cdots \overline{A_m}$, where $U \subset V$.

定義 4. *An SG G is a Unifiable Simple Grammar (USG) iff*

$$\overline{A} = \overline{B},\ A \to a\alpha \in R \textit{ and } B \to a\beta \in R \textit{ imply } \overline{\alpha} = \overline{\beta}.$$

For a USG $G = \langle V, \Sigma, R, S \rangle$, we define a USG $G/ = \langle V/, \Sigma, R/, \overline{S} \rangle$ as

$$\begin{aligned} V/ &= \{\, \overline{A} \mid A \in V \,\} \\ R/ &= \{\, \overline{A} \to a\overline{B_1} \ldots \overline{B_n} \mid A \to aB_1 \ldots B_n \in R \,\} \end{aligned}$$

USGs $G = \langle V, \Sigma, R, S \rangle$ and $H = \langle V', \Sigma, R', S' \rangle$ are σ-isomorphic iff G/σ and H/σ are equivalent modulo renaming nonterminals. From the definition of USGs, G/σ is also a USG and $L(G/\sigma) = L(G)$.

To show the USGs are unifiable, we define neighbourhood pairs for a USG, and eliminate them keeping its generality. The intuitive meaning of neighbourhood pairs can be seen in Lemma 3. For all USGs G and H that have no neighbourhood pair, $L(G) = L(H)$ implies that G is σ-isomorphic to H (Lemma 7). If G and H are σ-isomorphic, it is easy to unify them (Lemma 8).

$$\begin{tikzcd} G \arrow[rr, "\substack{\text{eliminating}\\ \text{neighbourhood}\\ \text{pairs(Alg.1)}}"'] \arrow[d, phantom, "{\substack{L(G)=L(H),\\ \text{not } \sigma\text{-isomorphic}}}"'] & & G_o \arrow[d, phantom, "\}\ \sigma\text{-isomorphic}"] \\ H \arrow[rr] & & H_o \end{tikzcd}$$

Definition 5 and 6 are required for the definition of the neighbourhood pairs.

定義 5. *The upstream of* $A \in V$ *is defined as* $\mathsf{up}_G(A) = \{B \in V \mid B \overset{*}{\Rightarrow} xA\}$, *and* $\mathsf{up}_G(U) = \bigcup_{A \in U} \mathsf{up}_G(A)$ *where* $U \subset V$.

定義 6. *Let* $U_1, U_2 \subset V$ *and* $U_1 \cap U_2 = \varnothing$. *Let us define* $W(U_1, U_2)$ *as the minimal subset of* V^* *such that,*

- $\varepsilon \in W(U_1, U_2)$.

- $\alpha \in W(U_1, U_2)$ *implies* $\alpha A \in W(U_1, U_2)$ *for all* $A \notin U_1$.
- $\alpha \in W(U_1, U_2)$ *implies* $\alpha AB \in W(U_1, U_2)$ *for all* $A \in U_1$ *and* $B \in U_2$.

If $\alpha A\beta \in W(U_1, U_2)$ for some $A \in U_1$, $\beta = B\beta'$ for some $B \in U_2$. Thus we have the following lemma.

補題 2. $\alpha\beta \in W(U_1, U_2)$ iff

$$\begin{cases} \alpha', \beta' \in W(U_1, U_2) & \textit{if } \alpha = \alpha' A,\ \beta = B\beta' \textit{ and } (A, B) \in (U_1, U_2) \\ \alpha, \beta \in W(U_1, U_2) & \textit{otherwise} \end{cases}$$

定義 7. *A pair* $\langle U_1, U_2\rangle \in \mathcal{P}(V) \times \mathcal{P}(V)$ *is called a* <u>*neighbourhood pair*</u> *iff the following conditions hold.*

1. $U_1 \cap U_2 = \varnothing$.
2. $\exists A \in V\ (\ U_1 = \mathsf{up}(\overline{A})\ )$.
3. $\exists A \in V\ (\ U_2 = \overline{A}\ )$.
4. $S \notin U_1$.
5. *For all* $A \to a\alpha \in R$,
   - $A \in U_1$ *implies* $\alpha B \in W(U_1, U_2)$ *for some* $B \in U_2$.
   - $A \notin U_1$ *implies* $\alpha \in W(U_1, U_2)$.

補題 3. $\langle U_1, U_2\rangle$ *is an neighbourhood pair iff conditions 1, 2 and 3 in Definition 7, as well as the following condition, hold.*

- $S \stackrel{*}{\Rightarrow} x\alpha$ *implies* $\alpha \in W(U_1, U_2)$ *for all* x.

Proof. First, we show the "only if" part. We prove this by induction on $|x|$. For the base case, $S \in W(U_1, U_2)$ since $S \notin U_1$. For the inductive step, assume that $S \stackrel{*}{\Rightarrow} xA\beta$ and $A\beta \in W(U_1, U_2)$. We show that, for all $A \to a\alpha \in R$, $S \stackrel{*}{\Rightarrow} xA\beta \Rightarrow xa\alpha\beta$ implies $\alpha\beta \in W(U_1, U_2)$.

In the case of $A \in U_1$, since $A\beta \in W(U_1, U_2)$, we can express β in the form $\beta = B\beta'$, where $B \in U_2$ and $\beta' \in W(U_1, U_2)$. $\alpha B \in W(U_1, U_2)$ from the condition 5. Thus we have $\alpha\beta = \alpha B\beta' \in W(U_1, U_2)$. In the case of $A \notin U_1$, $\beta \in W(U_1, U_2)$, since $A\beta \in W(U_1, U_2)$. $\alpha \in W(U_1, U_2)$ from the condition 5. Thus we have $\alpha\beta \in W(U_1, U_2)$.

Second, we show the "if" part. For the condition 4, $S \notin U_1$ since $S \in W(U_1, U_2)$. For the condition 5, since G is reduced, for all $A \to a\alpha \in R$, there exists some β such that $S \stackrel{*}{\Rightarrow} xA\beta \Rightarrow xa\alpha\beta$ and $A\beta, \alpha\beta \in W(U_1, U_2)$.

In the case of $A \in U_1$, $A\beta \in W(U_1, U_2)$ implies $\beta = B\beta'$ for some $B \in U_2$. Thus we have $\alpha B \in W(U_1, U_2)$ since $\alpha\beta \in W(U_1, U_2)$. In the case of $A \notin U_1$, if $\alpha = \varepsilon$ then $\alpha \in W(U_1, U_2)$. If $\alpha = \alpha' A'$, i.e. $A \to a\alpha' A' \in R$, then $A \in \mathsf{up}(A')$. By the condition 2 and $A \notin U_1$, we have $A' \notin U_1$. Thus we have $\alpha \in W(U_1, U_2)$, since $\alpha\beta \in W(U_1, U_2)$. □ □

定義 8. *Let $\langle U_1, U_2 \rangle$ be an neighbourhood pair of a USG $G = \langle V, \Sigma, R, S \rangle$. We define a map $\phi_{U_1,U_2} : W(U_1, U_2) \to V'^*$, where $V' = (V - U_1) \cup (U_1 \times U_2)$, by*

- $\phi_{U_1,U_2}(\varepsilon) = \varepsilon$.
- $\phi_{U_1,U_2}(A\beta) = \begin{cases} A_B \phi_{U_1,U_2}(\beta') & \textit{if } A \in U_1 \textit{ and } \beta = B\beta', \\ A\phi_{U_1,U_2}(\beta) & \textit{otherwise.} \end{cases}$

$\Phi(G, \langle U_1, U_2 \rangle)$ denotes the USG obtained by eliminating useless nonterminals and rules from the USG $\langle V', \Sigma, R', S \rangle$, where

$$\begin{aligned} R' = &\{ A \to a\phi_{U_1,U_2}(\alpha) \mid A \to a\alpha \in R \textit{ and } A \in V - U_1 \} \\ &\cup \{ A_B \to a\phi_{U_1,U_2}(\alpha B) \mid A \to a\alpha \in R \textit{ and } A_B \in U_1 \times U_2 \} \end{aligned}$$

Note that ϕ_{U_1,U_2} is a bijection.

補題 4. *Let $G' = \Phi(G, \langle U_1, U_2 \rangle)$. For all x, α and β such that $\beta = \phi_{U_1,U_2}(\alpha)$,*

$$S \overset{*}{\Rightarrow}_G x\alpha \text{ iff } S \overset{*}{\Rightarrow}_{G'} x\beta.$$

Proof. We prove this by induction on $|x|$. We write ϕ for ϕ_{U_1,U_2}. For the base case, $S = \phi(S)$. For the inductive step, first we show the "only if" part. Let $S \overset{*}{\Rightarrow}_G xA\beta \Rightarrow xa\alpha\beta$. By the induction hypothesis, we have $S \overset{*}{\Rightarrow}_{G'} x\phi(A\beta)$.

Case 1. $A \notin U_1$. G' has the rule $A \to a\phi(\alpha) \in R'$.

$$S \overset{*}{\Rightarrow}_{G'} x\phi(A\beta) = xA\phi(\beta) \Rightarrow xa\phi(\alpha)\phi(\beta) = xa\phi(\alpha\beta).$$

Case 2. $A \in U_1$. Due to Lemma 3, $\beta = B\beta'$ for some $B \in U_2$. G' has the rule $A_B \to a\phi(\alpha B) \in R'$. Thus,

$$S \overset{*}{\Rightarrow}_{G'} x\phi(AB\beta') = xA_B\phi(\beta') \Rightarrow xa\phi(\alpha B)\phi(\beta') = xa\phi(\alpha B\beta') = xa\phi(\alpha\beta).$$

Second, we show the "if" part. Let $S \overset{*}{\Rightarrow}_{G'} xA'\beta' \Rightarrow xa\alpha'\beta'$. By the induction hypothesis, we have $S \overset{*}{\Rightarrow}_{G'} xA\beta$ with $\phi(A\beta) = A'\beta'$.

Case 1. $A' \notin U_1 \times U_2$. $A = A'$ and G has the rule $A \to a\alpha \in R$, where $\phi(\alpha) = \alpha'$ due to the definition of Φ. Thus $S \overset{*}{\Rightarrow}_G xA\beta \Rightarrow xa\alpha\beta$ and $\phi(\alpha\beta) = \alpha'\beta'$.

Case 2. $A' = A_B \in U_1 \times U_2$. G has the rule $A \to a\alpha \in R$ where $\phi(\alpha B) = \alpha'$ due to the definition of Φ. Thus$S \overset{*}{\Rightarrow}_G xA\beta \Rightarrow xa\alpha\beta$. By $\phi(A\beta) = A_B\beta'$, $\beta = B\gamma$ for some γ with $\phi(\gamma) = \beta'$. Thus $\phi(\alpha\beta) = \phi(\alpha B\gamma) = \phi(\alpha B)\phi(\gamma) = \alpha'\beta'$. □ □

補題 5. *Let $\langle U_1, U_2\rangle$ be an neighbourhood pair of a USG G. $\Phi(G, \langle U_1, U_2\rangle)$ is more general than G.*

Proof. Let $\psi : V' \to V$ where $V' = (V - U_1) \cup (U_1 \times U_2)$ be defined as $\psi(A) = A$ for $A \in V - U_1$ and $\psi(A_B) = A$ for $A_B \in U_1 \times U_2$. By Lemma 4, ψ satisfies the condition in Lemma 1. □ □

Algorithm 1 Transformation of USGs

Require: G is a USG.

  while There exists an neighbourhood pair $\langle U_1, U_2\rangle$ in G. **do**

    $G := \Phi(G, \langle U_1, U_2\rangle)$.

  end while

  return $G_o := G$.

補題 6. *Algorithm 1 terminates for all $G \in USGs$.*

Proof. Let $\langle U_1, U_2\rangle$ be an neighbourhood pair of G, $H = \langle V_H, \Sigma, R_H, S_H\rangle$ denote $\Phi(G, \langle U_1, U_2\rangle)$, and $G' = \langle V_{G'}, \Sigma, R_{G'}, S_{G'}\rangle$ denote G/σ. The following claims are easy to prove but useful for what follows:

- $\langle U_1/\sigma, U_2/\sigma\rangle$ is an neighbourhood pair of G', and $\Phi(G', \langle U_1/\sigma, U_2/\sigma\rangle)$ is equivalent to H/σ modulo renaming nonterminals.

$$\begin{tikzcd} G \arrow[r, "\Phi(\cdot,\langle U_1,U_2\rangle)"'] \arrow[d, "/\sigma"'] & H \arrow[d, "/\sigma"', "\text{modulo renaming nonterminals}"] \\ G' \arrow[r, "\Phi(\cdot,\langle U_1/\sigma,U_2/\sigma\rangle)"] & H' \end{tikzcd}$$

  Let $H' = \langle V_{H'}, \Sigma, R_{H'}, S_{H'}\rangle$ denote $\Phi(G', \langle U_1/\sigma, U_2/\sigma\rangle)$.

- G has no neighbourhood pair if G' has no neighbourhood pair.

- There is a trivial bijection π from $V_{G'}$ to $V_{H'}$ such that $\sigma_{G'}(A) = \sigma_{H'}(\pi(A))$.

We define $p(G', A) \in V_{G'}^*$ for $A \in V_{G'}$ as the longest sequence in $\{\gamma \in V_{G'}^* \mid \forall x\, [S_{G'} \overset{*}{\Rightarrow} xA\alpha$ implies $A\alpha = \gamma\alpha']\}$. From Lemma 3, if $|p(G', A)| = 1$ for all $A \in V_{G'}$, there is no neighbourhood pair in G'. Thus, it is enough to prove that

$$\sum_{A \in V_{G'}} |p(H', \pi(A))| < \sum_{A \in V_{G'}} |p(G', A)|, \tag{4}$$

from the second claim noted above. In the following, let us denote $\phi_{U_1/\sigma, U_2/\sigma}$ as ϕ. From Lemma 4, we have $p(H', \pi(A)) = \phi(p(G', A))$. It is obvious that $\phi(p(G', A)) \le p(G', A)$ for all $A \in V_{G'}$ from the definition of $p(G', A)$.

When $A \in U_1/$, since $p(G', A)$ is written as $AB\beta$, where $\{B\} = U_2/$, $\phi(p(G', A)) = A_B\phi(\beta)$. Thus $|\phi(p(G', A))| = 1 + |\phi(\beta)| \le 1 + |\beta| = -1 + |AB\beta|$. Consequently, we obtain Eq. 4. □ □

Since the above proof shows that the number of loop is less than $|G/\sigma|^2$, it is easy to prove that $|G_o|$ is $O(|G|^{|G/\sigma|^2})$, while $|G_o/\sigma|$ is $O(|G/\sigma|^3)$, where G_o is the output USG of Alg.1. This implies that the time complexity of finding neighbourhood pairs are $O(|G/\sigma|^6)$ in all. Thus the time complexity of Alg.1 is also $O(|G|^{|G|^2})$ when concerning only $|G|$. Let the ambiguity $\mathsf{amb}(G)$ of a USG G be defined as $|\{H/\sigma$ modulo renaming of nonterminals $\mid H \in$ USGs, $L(H) = L(G)\}|$. Since $|G_o|$ is limited to $O(|G|^{\mathsf{amb}(G)})$, the time complexity of Alg.1 is limited to $O(|G|^{\max\{\mathsf{amb}(G), 6\}})$.

補題 7. *Let two USGs G_o and H_o have no neighbourhood pair. If $L(G_o) = L(H_o)$, then G_o and H_o are σ-isomorphic.*

Proof. Let $G = \langle V, \Sigma, R, S\rangle$ and $H = \langle V', \Sigma, R', S'\rangle$ denote $G_o/$ and $H_o/$, respectively. It is sufficient to show that $L(G) = L(H)$ implies that G and H are equivalent modulo renaming nonterminals. Note that $\overline{A} = \{A\}$ for all $A \in V$, and thus $\sigma_G(A) = \sigma_G(A')$ implies $A = A'$.

First, we prove that $\sigma_G(A) = \sigma_H(B)$ implies $L(G, A) = L(H, B)$ for all $A \in V$ and $B \in V$. When $\sigma_G(A) = \sigma_H(B)$,

$$\forall x\, [S \overset{*}{\Rightarrow}_G xA\alpha \text{ iff } S \overset{*}{\Rightarrow}_H xB\beta],$$

since $L(G) = L(H)$. Thus $y \in L(G, A)$ implies that, for some $z \in L(H, B)$, z is a prefix of y or y is a prefix of z (if not so, $L(G) \ne L(H)$). We may assume that y is a prefix of z. Suppose that y is a <u>proper</u> prefix of z, i.e., $A \overset{*}{\Rightarrow}_G y$ and $B \overset{*}{\Rightarrow}_H yC\gamma$, then we have

$$\exists yC\gamma\, \forall x\, [S \overset{*}{\Rightarrow}_G xy\alpha \text{ iff } S \overset{*}{\Rightarrow}_H xyC\gamma\beta].$$

It follows that $\alpha = D\alpha$ for all x, where $D \in V$ and $\sigma_G(D) = \sigma_H(C)$. Thus there exists some D such that, for all x, $S \overset{*}{\Rightarrow}_G xA\alpha$ implies $\alpha = D\alpha$. For $A' \in \mathsf{up}_G(A)$, we also have, for all x, $S \overset{*}{\Rightarrow}_G xA'\alpha$ implies $\alpha = D\alpha$, because $S \overset{*}{\Rightarrow}_G xA'\alpha \overset{*}{\Rightarrow} xzA\alpha$ for some z. Thus, by Lemma 3, $\langle \mathsf{up}_G(A), \{D\}\rangle$ is an neighbourhood pair of G. Clearly, G_o has some neighbourhood pair iff G has some neighbourhood pair. This is a contradiction. Thus $y \in L(G,A)$ implies $y \in L(G,B)$ and vice versa, so $L(G,A) = L(G,B)$.

Second, we show that G and H are equivalent modulo renaming nonterminals. Let $\sigma_G(A) = \sigma_H(B)$, $A \to a\alpha \in R$ and $B \to a\beta \in R'$.

If $\alpha = \varepsilon$, $\beta = \varepsilon$ since $L(G,A) = L(H,B)$. If $\alpha = A_1 \cdots A_m$ and $m \geq 1$, we may assume that $\beta = B_1 \cdots B_n$ and $n \geq m$. Let us prove that $\sigma_G(A_i) = \sigma_H(B_i)$ by induction on i. For the base, $\sigma_G(A_1) = \sigma_H(B_1)$ since $L(G,A) = L(H,B)$. If $\sigma_G(A_1) = \sigma_H(B_1)$, $\cdots$ and $\sigma_G(A_i) = \sigma_H(B_i)$, then $L(G, A_1 \cdots A_i) = L(H, B_1 \cdots B_i)$. It follows that $\sigma_G(A_{i+1}) = \sigma_H(B_{i+1})$, since $L(G,A) = L(H,B)$. We have also $n = m$, since $L(G, A_1 \cdots A_m) = L(H, B_1 \cdots B_m)$ and $L(G,A) = L(H,B)$. □ □

Let $G_1 = \langle V_1, \Sigma, R_1, S_1 \rangle$ and $G_2 = \langle V_2, \Sigma, R_2, S_2 \rangle$ be USGs for which $L(G_1) = L(G_2)$, neither having any neighbourhood pairs . Let

$$
\begin{aligned}
&V' = \{\ (A_1, A_2) \in V_1 \times V_2 \mid s_{G_1}(A_1) = s_{G_2}(A_2)\},\\
&R' = \{\ (A_1, A_2) \to a(B_{1,1}, B_{2,1}) \cdots (B_{1,m}, B_{2,m}) \mid (A_1, A_2) \in V',\\
&\qquad A_1 \to aB_{1,1} \cdots B_{1,m} \in R_1 \text{ and } A_n \to aB_{2,1} \cdots B_{2,m} \in R_2\ \},\\
&S_* = (S_1, S_2).
\end{aligned}
$$

The USG G_*, obtained by <u>parallelizing</u> G_1 and G_2, is defined as $\langle V_*, \Sigma, R_*, S_* \rangle$, where V_* and R_* are arrived at by eliminating the useless nonterminals and rules from V' and R', respectively.

補題 8. *G_* is more general than G_1 and G_2.*

Proof. Let $\pi_i : V_* \to V_i$ be a map such that $\pi_i(A_1, A_2) = A_i$. From Lemma 7, $A_1 \to a\alpha_1 \in R_1$ iff $A_* \to a\alpha_* \in R_*$, $\pi_1(A_*) = A_1$ and $\pi_1(\alpha_*) = \alpha_1$. It follows that $S_* \stackrel{*}{\Rightarrow}_{G_*} x\alpha_*$ implies $S_i \stackrel{*}{\Rightarrow}_{G_i} x\pi_i(\alpha_*)$ for all x. □ □

定理 1. *The class of USGs is unifiable.*

Proof. Let USGs G_0 and H_0 be output by Algorithm 1 for the input USGs G and H, with $L(G) = L(H)$. G_0 and H_0 are more general than G and H, respectively, by Lemma 5. Therefore G_* obtained by parallelizing G_0 and H_0 is more general than G and H. □ □

For every RSL, there is a finite number of RSGs, modulo renaming nonterminals, that exactly generate the RSL. Moreover, it is easy to prove the following lemma.

補題 9. *For every RSG H, there is an RSG G in normal form, where G is σ-isomorphic to H and more general than H.*

It is easy to modify Yoshinaka's learning algorithm so that, for a given RSG G, it enumerates all RSGs in normal form that generate the same language as G. In that learning algorithm. From the above theorem, we have the following:

定理 2. *For every RSG G, we can construct a USG G_* such that for any RSG H with $L(H) = L(G)$, it holds that $\mathbb{K}(H) \subset \mathbb{K}(G_*)$. $|G_*|$ is $O(m(G)^{2\mathsf{amb}(G)})$, where $m(G) = \max\{|H| \mid H$ is an RSG and $L(G) = L(H)\}$.*

6 文法統合アルゴリズムの強化学習への応用

At first, let us introduce simple context-free decision processes, which are a natural extension of finite-state Markov decision processes.

定義 9. *Let $G = \langle V, \Sigma, R, S\rangle$ be an SG. $G_{U,P,C} = \langle V, \Sigma, R, S, U, P, C\rangle$ is a <u>simple context-free decision process</u> iff U, P, C are the following set and functions.*

- *U is a finite set of <u>actions</u>.*
- *P is a map from $R \times U$ to $[0, 1]$, called <u>a probability assignment</u>, where $\forall u \in U, \forall A \in V[\sum_{r \in R_A} P(r, u) = 1]$ holds.*
- *C is a map from Σ to $(-\infty, \infty)$, called <u>reward</u>.*

Hereafter, if G is an SG or an RSG, simple context-free decision processes $G_{U,P,C}$ are called an SG-DP or an RSG-DP, respectively.

Corresponding to a given SG-DP $G_{U,P,C}$, the sequence of discrete random variables is given as $X_1, Y_1, X_2, Y_2, \cdots$, where the domains of X_i and Y_i are $\Sigma^* V^*$ and U respectively, and $X_1 = S$. The following properties hold.

$$
\begin{aligned}
&\Pr(X_t = x_t\alpha_t | X_1 = S, Y_1 = u_1, \cdots, X_{t-1} = x_{t-1}\alpha_{t-1}, Y_{t-1} = u_{t-1}) \\
&= \Pr(X_t = x_t\alpha_t | X_{t-1} = x_{t-1}\alpha_{t-1}, Y_{t-1} = u_{t-1}) \\
&= \begin{cases} P(r, u_{t-1}) & \text{if } x_{t-1}\alpha_{t-1} \Rightarrow_G x_t\alpha_t \text{ with the rule } r \\ 1 & \text{if } x_{t-1} = x_t \text{ and } \alpha_{t-1} = \alpha_t = \varepsilon \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

An SG-DP $G_{U,P,C}$ is called an episodic finite Markov decision process iff $G = \langle V, \Sigma, R, S \rangle$ is reduced and can be expressed in the form:

for some $n \geq 1$, $k \geq 0$ and $V_1, \cdots, V_{n+k} \subset V$,
$V = \{A_1(= S), \cdots, A_n\}$, $\Sigma = \{a_1, \cdots, a_{n+k}\}$,
$R = \{A \to a_j A_j | A \in V_j, j = 1, \cdots, n\} \cup \{A \to a_{n+j} | A \in V_{n+j}, j = 1, \cdots, k\}$

The above definition is obviously equivalent to the usual definition of episodic finite MDPs. Note that G is an RSG whenever $G_{U,P,C}$ is an episodic finite MDP.

Let $G_{U,P,C} = \langle V, \Sigma, R, S, U, P, C \rangle$ be an SG-DP. A map $\mu : V \to U$ is called a policy. One of the main purpose of reinforcement learning is to determine the policy μ so as to maximise the expectation of the total reward from S. The value function $J : V \to (-\infty, \infty)$ under μ is defined as

$$J_\mu(A) = \sum_{x \in L(G,A)} \Pr(x | \langle G, P_\mu \rangle, A) \sum_{i=1}^{|x|} C(a_i),$$

where $x = a_1 \cdots a_{|x|}$, and P_μ is the probability assignment of G under μ, namely, for $B \to b\beta \in R$, $P_\mu(B \to b\beta) = P(B \to b\beta, \mu(A))$.

Let $M(G_{U,P,C}, \mu)$ be a $(|V|, |V|)$ matrix whose element $M(G_{U,P,C}, \mu)_{ij}$ represents the expectation of the number of A_j derivable in one step from A_i under μ, where $V = \{A_1, A_2, \cdots, A_{|V|}\}$. It is known that P_μ is consistent if $\rho(M(G_{U,P,C}, \mu)) < 1$, where $\rho(M)$ is the spectral radius of M [11].

When $\rho(M(G_{U,P,C}, \mu)) < 1$ for any $\mu \in \pi$, where π is the set of all policies, the optimal value function $J_* : V \to (-\infty, \infty)$ can be defined as $J_*(A) = \max_{\mu \in \pi} J_\mu(A)$. There exists some policy μ_* such that $J_{\mu_*}(A) = J_*(A)$ for all $A \in V$, called an optimal policy. The optimal action-value function $Q_* : V \times U \to \mathbb{R}$ are also defined as $Q_*(A, u) = \sum_{A \to aB_1 \cdots B_k \in R_A} P(A \to aB_1 \cdots B_k, u)(C(a) + \sum_{i=1}^{k} J_*(B_i))$. Note that the above definitions are a natural extension of the usual definitions on reinforcement learning whose discounting factor equals 1.

Let $G_{U,P,C} = \langle V, \Sigma, R, S, U, P, C \rangle$ be an SG-DP, and $H = \langle V', \Sigma, R', S' \rangle$ be an SG such that $\mathbb{K}(G) \subset \mathbb{K}(H)$. Let $\psi : V'_{\geq 2} \to V$ be some map that satisfies the property in Lemma 1. We can construct P', a probabilistic assignment of H, such that, for $A \to a\alpha \in R'$ and $u \in U$, $P'(A \to a\alpha, u) = P(\psi(A) \to a\beta, u)$ if $A \in V'_{\geq 2}$, otherwise $P'(A \to a\alpha, u) = 1$. We have the following:

定理 3. *Assume that $\rho(M(G_{U,P,C}, \mu)) < 1$ for all $\mu \in \pi(V, U)$, Q_t defined by the following iteration (Q-Learning) converges to the optimal action-value* *function of $H_{U,P',C}$ as $t \to \infty$ w.p. 1.*

$$Q_{t+1}(A_t, u_t) := (1 - \mathsf{k}_t) Q_t(A_t, u_t) + \mathsf{k}_t (C(a) + \sum_{i=1}^{k} \max_{v \in U} Q_t(B_i, v)), \quad (5)$$

*where the rule $A_t \to aB_1 \cdots B_k \in R'$ is randomly chosen with probability $P(\psi(A_t)$
$\to a\beta, u_t)$ if $A_t \in V'_{\geq 2}$, otherwise 1. $\mathsf{k}_t \in [0,1]$ is a random variable depending on A_t and u_t, called a* <u>*step-size parameter*</u>*. We assume that A_t, u_t and k_t satisfy the following conditions for all $(A, u) \in V' \times U$; $\sum_{\{t \in \mathbb{N} | (A_t, u_t) = (A, u)\}} \mathsf{k}_t^2 < \infty$, $\sum_{\{t \in \mathbb{N} | (A_t, u_t) = (A, u)\}} \mathsf{k}_t = \infty$.*

Proof. By the definition of P', We may assume that $G = H$ and ψ is the identity map. If $G = H$, the convergence of the Q-Learning method in the above theorem is proved by modifying the contraction mapping and the weighted norm in [4]. □ □

Now, we explain the relationship between learning an SG from positive data and Q-Learning on an SG-DP, and the necessity of probabilistic unification. We identify elements of Σ with observations and nonterminal symbols with unobservable states. The division of a process into observable and unobservable states follows the same scheme as appears in partially observable Markov decision processes (POMDPs) [7]. The difference from POMDPs is that nonterminal symbols are unobservable in SG-DPs but are determined if its grammar is known. In order to use the extended Q-learning method (Eq. 5), we must identify the sequence of nonterminals that corresponds to observations. We can regard histories of observations as positive data. Thus we can use the extended Q-learning method (Eq. 5) after identifying the grammar from histories of observations.

We assume that the class of environments belongs to the class of RSG-DPs, instead of to the class of SG-DPs, because the class of RSGs is of the most suitable size among subclasses of SGs. The class of RSGs is large enough to include all episodic finite MDPs, while also small enough to be learnable from positive data efficiently. Moreover, the class of RSGs is a probabilistic unifiable class within the class of USGs. Recall other subclasses of SGs we mentioned in this paper; the class of VSGs are efficiently learnable but VSG-DPs do not include all episodic finite MDPs, USGs are learnable from positive data but no efficient learning algorithm for them is known, and SGs are not even learnable from positive data.

Alg. 2 is a learning method for one episode in order to optimize the policy for RSG-DPs when the grammars are unknown. Let $G_{U,P,C}$ =

$\langle V, \Sigma, R, S, U, P, C\rangle$ be an RSG-DP (unknown). Let Env be a oracle function from {prefixes of $L(G)$} $\times\, U$ to $\Sigma \cup \{\varepsilon\}$. $\mathsf{Env}(x, u) = \varepsilon$ if $x \in L(G)$, otherwise, $\mathsf{Env}(x, u) = a$ such that a is randomly chosen with probability $P(A \;\to a\alpha, u)$, where $S \stackrel{*}{\Rightarrow} xA$. As the initial parameters, let the USG H, Q_H and Hist be as follows. $H = \langle V', \Sigma, R', S\rangle$, where $V' = \{[a] | a \in \Sigma\} \cup \{S\}$ and $R' = \{[a] \to b[b], S \to b[b] \mid a, b \in \Sigma\}$. $Q_H(A, u) = 0$ for all $A \in V$ and $u \in U$, where $Q_H : V \times U \to (-\infty, \infty)$. $\mathsf{Hist} := \varnothing$. Let $\mathsf{Str}(H, Q_H)$: {prefixes of $L(H)$} $\to U$ be some strategy e.g., ε-greedy strategy [9].

By the definition of the learnability from positive data, it holds that, for some $n \in \mathbb{N}$, for all $m > n$, $H_m = H_n$ and $\mathbb{K}(G) \subset \mathbb{K}(H_n)$, where H_n is the inferred grammar at the n-th episode. Thus, by Theorem 3, Q_{H_n} converges to the optimal action-value function.

Algorithm 2 A reinforcement learning for one episode on RSG-DP

Require: $H = \langle V, \Sigma, R, S\rangle$ is an USG and $Q_H : V \times U \to (-\infty, \infty)$.

  $x := \varepsilon$ and $u := \mathsf{Str}(H, Q_H)(x)$.
  while $(a := \mathsf{Env}(x, u)) \neq \varepsilon$ **do**
    if $S \stackrel{*}{\Rightarrow}_H xA\alpha \Rightarrow xa\beta\alpha$ **then**
      Update $Q_H(A, u)$ according to Eq. 5, $u := \mathsf{Str}(H, Q_H)(xa)$, and $x := xa$.
    else
      $x := xa$. u is randomly chosen under the uniform distribution on U.
    end if
  end while
  $\mathsf{Hist} := \mathsf{Hist} \cup \{x\}$.
  if $x \notin L(H)$ **then**
    $\mathcal{G}$ = all the RSGs in normal form generated by the algorithm for learning RSGs from Hist.
    H :=[the USG obtained by unifying all the RSGs in $\mathcal{G}$], with Q_H initialized to 0.
  end if

Finally, as an example of an RSG-DP and an application of the unification algorithm, we consider the problem of maximizing total reward under some conditions. An agent starts from the position $s = (1, 2)$ on the map (Fig. 19), and can move left, right, up or down, unless there is a wall in that direction. It costs 1 (-1 as a reward) per single step, and the agent is allowed to occupy a location either $f_+ = (5, 6)$ or $f_- = (5, 2)$ at most one time. When reaching to the goal $g = (9, 2)$, if the agent has passed through f_+, it observes h_+ w.p. 0.9 or h_- w.p. 0.1, whereas if the agent has passed through f_-, it observes h_+ w.p. 0.1 or h_- w.p. 0.9. The observation of h_+ implies that the agent gets 100 as a reward, and of h_- implies that it gets 50. In this case, the RSG $G = \langle V, \Sigma, R, S\rangle$ is written as follows. $\Sigma = \mathsf{Map} \cup \{h_+, h_-\}$, $V = \{[a,0] \mid a \in \mathsf{Map} \text{ and } a \neq g\} \cup \{[f_\pm, 1], S\}$, and

$$\begin{aligned} R =&\{[a,0] \to b[b,0] \mid b \in \mathsf{mov}(a) \text{ and } b \notin \{f_+, f_-, g\}\} \\ &\cup\{[a,0] \to b[b,0][b,1] \mid b \in \mathsf{mov}(a) \text{ and } b \in \{f_+, f_-\}\} \\ &\cup\{[a,0] \to g \mid g \in \mathsf{mov}(a)\} \cup \{[f_+,1] \to h_\pm,\ [f_-,1] \to h_\pm,\ S \to s[s,0]\}, \end{aligned}$$

where $\mathsf{Map} = \{(i,j) | (i,j) \text{ is a reachable position on the map.}\}$, and $\mathsf{mov}(i,j)$ denotes a set of positions where the agent can move from (i,j) in one step. For example, $\mathsf{mov}(S) = \{(1,1),(2,3),(2,2)\}$, $\mathsf{mov}(f_+) = \{(6,6)\}$ and $\mathsf{mov}(g) = \varnothing$.

There is another RSG $H = \langle V'\Sigma, R', S\rangle$ such that $L(G) = L(H)$, where $V' = \{[a,0] \mid a \in \mathsf{Map}\} \cup \{S\}$ and $R' = \{[a,0] \to b[b,0] \mid b \in \mathsf{mov}(a)\} \cup \{S \to s[s,0],\ [g,0] \to h_\pm\}$. Note that the RSG-DP based on H is an episodic finite MDP.

G and H are all the RSGs in normal form whose language is equivalent to $L(G)$, thus both G and H, and only G and H are output by the learning algorithm of RSGs from positive data. The USG $G_* = \langle V_*, \Sigma, R_*, S_*\rangle$ transformed from G by Alg. 1 is as follows. $V_* = \{[a,0] \mid a \in \mathsf{West}\} \cup \{[a,0]_\pm \mid a \in \mathsf{East}\} \cup \{[f_\pm, 0]_\pm, S_*\}$, and

$$\begin{aligned} R_* =&\{[a,0] \to b[b,0] \mid b \in \mathsf{mov}(a) \text{ and } b \in \mathsf{West}\} \\ &\cup\{[a,0]_\pm \to b[b,0]_\pm \mid b \in \mathsf{mov}(a) \text{ and } b \in \mathsf{East}\} \\ &\cup\{[a,0]_\pm \to g[f_\pm,1] \mid g \in \mathsf{mov}(a)\} \\ &\cup\{[(4,4\pm 2),0] \to f_\pm[f_\pm,0]_\pm,\ [f_+,1] \to h_\pm,\ [f_-,1] \to h_\pm,\ S \to s[s,0]\}, \end{aligned}$$

where $\mathsf{East} = \{(i,j) \in \mathsf{Map} \mid j \geq 6, (i.j) \neq g\}$, $\mathsf{West} = \{(i,j) \in \mathsf{Map} \mid j \leq 4\}$, and $[a,0]_\pm$ denote $[a,0]_{[f_\pm,1]}$. Table 3 shows the neighbourhood pair and changed rules for each loop in Alg. 1 for G.

G_* is σ-isomorphic to H and clearly $\mathbb{K}(H) \subset \mathbb{K}(G_*)$. Thus $\mathbb{K}(G_*)$ is more general than both G and H. Note that G_* is not an RSG but a USG.

The optimal length of episode of this problem is 16 when the agent is through f_+, and thus the maximum total reward is 79. Fig. 20 is an experiment of Alg. 2 on the above problem. It demonstrates that the agent approaches the optimal path and obtains maximum total reward after the grammatical inference and the unification are complete at approximately the 200th episode. In Fig. 21 our method is comparing to the naive Q-Learning method, in which the environment is assumed to be an episodic finite MDP

表 3: Neighbourhood pairs in Alg. 1 for G.

Loop	U_1	U_2	New rules obtained by $\Phi(\cdot, \langle U_1, U_2\rangle)$
1	$\{[f_+, 0]\}$	$\{[f_\pm, 1]\}$	$\{[(4,6),0] \to f_+[f_+,0]_+,\ [f_+,0]_+ \to (6,6)[(6,6),0][f_+,1]\}$
2	$\{[f_-, 0]\}$	$\{[f_\pm, 1]\}$	$\{[(4,2),0] \to f_-[f_-,0]_-,\ [f_-,0]_- \to (6,2)[(6,2),0][f_-,1]\}$
3	$\{[a,0] \mid a \in \mathsf{East}\}$	$\{[f_\pm, 1]\}$	$\{[f_\pm,0]_\pm \to [(6, 4\pm 2), 0]\} \cup$ $\{[a,0]_\pm \to b[b,0]_\pm \mid b \in \mathsf{mov}(a)$ and $b \in \mathsf{East}\} \cup$ $\{[a,0]_\pm \to g[f_\pm,1] \mid g \in \mathsf{mov}(a)\}$

(same as H). The total reward obtained by the naive Q-Learning method is approximately 40, indicating that the agent passed through f_-.

図 19: Example problem of RSG-DP.

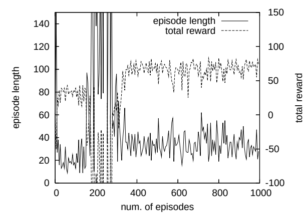

図 20: Total reward and episode length.

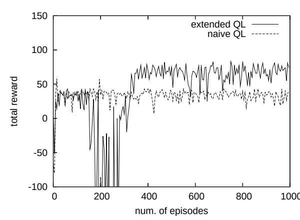

図 21: Comparison of QL and SG-QL methods.

7 文法型分布推定アルゴリズムにおける選択的アンサンブル効果

7.1 文法型分布推定アルゴリズム

7.2 組み合わせ最適化アルゴリズムにおけるブースティング的手法

機械学習における”ブースティング”という概念とその理論的な研究は、Kearns ら [] の次のような問いから始まったと言われている:

”Can a weak learner which is a bit better than random guessing be boosted into an arbitrarily accurate strong learner ? ”

実際 Kearns らは、精度が 1/2 よりもわずかに良いだけでよい PAC 学習である弱 PAC 学習を定義して、効率的に弱 PAC 学習可能ならば効率的に PAC 学習可能であることを示している []。

これは、精度の低い学習器であっても、それがランダムよりも少しでもましならば、その学習を繰り返すことによって、精度の十分よい学習器と同じになるという意味である。

普通はブースティングというと機械学習の一手法をさし、前節のように、既存の学習アルゴリズムをたくさん寄せ集めることによってよりよい精度の学習を行おうという、学習手法のことである。ブースティングでは、どのような学習アルゴリズムを束ねるかということについては自由であって、どのようなものに対しても組み合わせて用いることができる。

本節では、このようなブースティングの概念を、組み合わせ最適化問題をとくための最適化アルゴリズムに当てはめてみる。

すなわち、一般的にいって、さまざまな最適化問題があり、それに対するさまざまな効率的なアルゴリズムが考えられているが、最適化アルゴリズムにおけるブースティングとは、そのようなアルゴリズムを束ねて用いることであるとする。とくに、文法型分布推定アルゴリズムにおいて、適度な長さで止め、独立に何度か繰り返すことの効果について検証する。これは、「複数の独立したアルゴリズムを協調させる」という点において、いわゆる boosting 的な方法であるということができる。

7.3 組み合わせ最適化問題と最適化アルゴリズムの定義

組み合わせ最適化問題と、その最適化アルゴリズムについての一般的な定義を述べる。また、そのような最適化アルゴリズムにはどのような制約があるかについて、一般的なことを述べる。

まず、ここでは、最適化問題を簡単に次のように定義することにしよう (最適化問題のより正確な定義については cite など)。

変数の値をあらわす集合を$\mathcal{I}$、評価値をあらわす順序集合を$\mathbb{R}$とする。それに対して、評価関数の集合$\mathcal{F} \subset \mathcal{I} \times \mathbb{R}$があるとする。このとき、三つ組み$\langle \mathcal{I}, \mathbb{R}, \mathcal{F} \rangle$を最適化問題とよぶ。

アルゴリズムAが、最適化問題Pに対する最適化アルゴリズムであるとは、$k \in \mathbb{R}$および、$f \in \mathcal{F}$をあらわす記述またはオラクル（xを入力すれば$f(x)$を返してくれるブラックボックス）を入力として、$f(x) \geq k$となる$x \in \mathcal{I}$が存在する場合はそれを出力することである。

ここで、最適化アルゴリズムの入力として、関数$f \in \mathcal{F}$の記述を既知とするか未知とするかの二通り考えることができる。正確に言うと、一つは、Fの元が多項式時間で計算可能な記述を持ち、$f \in \mathcal{F}$の記述と$k \in \mathbb{R}$を入力とする場合であり、もう一つは、$f \in F$を表現するオラクルO_fと$k \in \mathbb{R}$を入力とする場合である。

前者と後者の違いがよくわかるように、例として、「藁の山のなかの針」の問題を挙げておこう。これは最適化問題としては、$\langle [0, N], \{0, 1\}, F \rangle$、ただし$F = \{f_i | i \in [0, N], \forall j \ [\ f(j) = 1 \text{ iff } i = j\}$と書ける。

$k = 1$の場合を考える。前者の場合、fを記述するためには、$f(x) = 1$となるxを記述すればよく、記述長は$\log N$程度ですみ、入力xをそのまま出力すれば$f(x) = 1 \geq k$となるので計算量は$\mathcal{O}(\log N)$で済む。後者の場合、fがブラックボックスとなっているので、どのようなアルゴリズムを用いても、計算量はFからfが均等にとられたとすると、平均でO_fに$N/2$回は問い合わせなければならない。

後者の場合、上記の定義のかわりに、次のようなアルゴリズムの定義を考える。

オラクル型最適化アルゴリズムとは、次の条件を満たす列挙アルゴリズムであるとする。

1. $f \in \mathcal{F}$をあらわすオラクルO_fを入力としてもつ。
2. O_fに問い合わせる毎に今までに問い合わせたなかで最大のものを出力する。
3. Fの任意の元について一度しかO_fに問い合わせない。

オラクル型最適化アルゴリズムからは、前の定義を満たす最適化アルゴリズムを簡単に作ることができるし、逆に前の定義を満たす最適化アルゴリズムからは上条件の(2),(3)を満たすように少し変更することでオラクル型最適化アルゴリズムを作ることができる。

7.3.1 No Free Lunch Theorem

最適化アルゴリズムは一般的にどの程度まで速くすることができるだろうか。

関数 f の記述を入力とする場合は、例えば、NP 困難な問題を帰着することができる問題を作ることができるので、そのような問題の場合、$P \neq NP$ を信じると、当然どのような最適化アルゴリズムを用いても f の記述長に対して多項式時間で停止することができないが、その制約の範囲内でより早く解けるように工夫することはできる。

オラクル型の場合は、オラクルへの問い合わせ回数が重要になることが多く、しばしば計算量の変わりにオラクルへの問い合わせ回数を用いてアルゴリズムの速さを評価することが多い。オラクルの問い合わせと問い合わせの結果の評価値に関して、次のような否定的な定理が成り立つことが知られている。

定理 4 (No Free Lunch, NFL [**?**, **?**]). $P = \langle \mathcal{I}, \mathbb{R}, \mathcal{F} \rangle$ を最適化問題とし、$|\mathcal{I}| < \infty$ とする。P に対するオラクル型最適化アルゴリズム全てを $\mathcal{A}$ とする。I 上の置換 $\sigma : \mathcal{I} \to \mathcal{I}$ の集合を S とする。このとき次は同値である。

- $\mathcal{F}$ が置換に対して閉じている、すなわち $F \circ \sigma = \{f \circ \sigma | f \in \mathcal{F}\}$ として、$\forall \sigma \in S\ [\mathcal{F} \circ \sigma = \mathcal{F}]$。
- $x_i(A, f)$ を $A(O_f)$ が i 番目に問い合わせた $\mathcal{I}$ の元として、$\forall A, B \in \mathcal{A}, \forall f \in \mathcal{F}, \exists g \in \mathcal{F} \forall i \in \{1, \cdots, |\mathcal{I}|\}\ [\ f(x_i(A, f)) = g(x_i(B, g))\ ]$。

この定理は直感的には明らかであるが、進化計算における効率性の議論において、否定的な結果として近年注目をあびた。

NFL が意味するところは単に、何をしても無駄な問題がある、つまり全てのオラクル型最適化アルゴリズムにおいて平均すると同じ結果しか得ることができない問題があるということだけでない。というのは、任意の二つのオラクル型最適化アルゴリズム A, B に対して、ある問題 $\langle \mathcal{I}, \mathbb{R}, \mathcal{F} \rangle$ に対して、問い合わせ履歴の評価値の平均だけをもとにして A よりも B のほうが良いと判断した場合、同じ理由で問題 $\langle \mathcal{I}, \mathbb{R}, (\mathcal{F} \circ \sigma) - \mathcal{F} \rangle$ では B よりも A のほうが良いと判断されるからである。

以降では特に後者のオラクル型の最適化アルゴリズムについて考えることにする。

7.3.2 オラクル型最適化アルゴリズムにおけるブースティング

焼きなまし法や進化的手法など、ランダムな要素を含む最適化アルゴリズムの場合、ある時間内 t に停止することができる (つまり $f(x) > k$ となる x を出力することができる) 確率を考えることができる。

一つのアルゴリズムを走らせ続けるのと、L 回オラクルに問い合わせた後、途中でやめてリセットしてはじめからやり直すのを繰り返す (Alg. 7.3.2) のと、どちらが停止するまでの問い合わせ回数の期待値を少なくすることができるだろうか。当然、オラクル型の場合を考えているので、NFL より、どの

ようなアルゴリズムを用いても同じ結果にしかならない場合がありうるが、直感的には、多くの場合において、ある最適な L があって、それを繰り返すのが一番よくなると期待される。というのは、あまり短いと十分良い解にたどり着く前にリセットされてしまうし、逆にあまり長くても、ローカルマキシマムに陥ってしまうと期待されるためである。

Algorithm 3 オラクル型最適化アルゴリズム A に対する繰り返しアルゴリズム

入力: オラクル O,$k \in \mathbb{R}$
while 1 **do**
  $A(O)$ が O に L 回問い合わせるまで実行し、その最後の出力を x とする。
  if $O(x) \geq k$ **then**
    return x
  end if
end while

実際、進化的手法をアルゴリズムとして選んだ場合、進化的手法が有用な ($\subset$ NFL の (1) が成り立たないような) ほとんどの最適化問題において、期待問い合わせ回数を最低にするような、最適な停止問い合わせ回数（以降最適問い合わせ回数、または最適評価回数と呼ぶ）があることがよく知られている。Fig. 7.3.2 に世代数と期待評価回数の関係の典型的なグラフを示す。これは wall-following 問題における集団数 1000 の 古典的な GP(交叉のみ),GP(突然変異のみ),GP(突然変異と交叉) と、柳井 [?, ?] による XEDP と呼ばれる木構造に対する分布推定型の進化的手法を比べたものである。この場合いずれの手法においてもおおよそ 20 世代程度で止めるのが最適で、期待評価回数が最小になっている。図 7.3.2 中の $I(M,t,p)$ は、集団数 M、世代数 t(評価

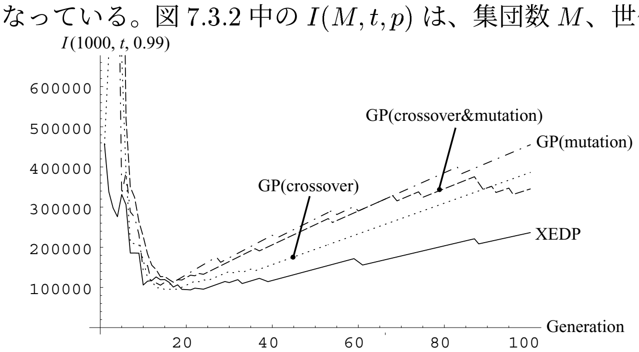

図 22: 世代数と期待評価回数の関係の例 ([?, ?] より抜粋)

回数としては Mt 回) の進化計算の繰り返しで p の確率で $f(x) > k$ となる x が見つかるために必要な最低評価回数を意味していて、繰り返し回数を K とすると MtK であらわされる。集団数 M、世代数 t で $f(x) > k$ となる x が

見つかる確率を $P(M,t)$ とすると、$(1-P(M,t))^K = 1-z$ より、

$$I(M,t,z) = MtK = \frac{\log(1-z)}{\log(1-P(M,t))}$$

と計算できる。進化計算では、しばしば、$\min_t I(M,t,0.99)$ がアルゴリズムの性能の評価の指標として用いられている [?, ?]。

最適評価回数が既知の場合はそこで停止して繰り返せばよいが、普通は問題ごとに異なる値をとるものであり、未知である。そこで、停止評価回数を少ないものから順に少しずつ増やしてゆくというアルゴリズムを考えてみよう。Alg.7.3.2 では、停止する評価回数を 1 回から α 倍づつしている。ただし $\alpha > 1$ とする。

Algorithm 4 オラクル型最適化アルゴリズム A に対する繰り返しアルゴリズム 2

入力: オラクル O,$k \in \mathbb{R}$
$L := 1$
while 1 **do**
  $A(O)$ が O に L 回問い合わせるまで実行し、その最後の出力を x とする。
  if $O(x) \geq k$ **then**
    return x
  end if
  $L := \alpha L$
end while

$p(l)$ を、$A(O)$ が O に l 回問い合わせるまで実行したときの最後の出力 x に対し、$O(x) \geq k$ をみたす確率とする。この場合の期待評価回数 $E(l)$ が最適期待評価回数 l_* に対して最悪どの程度大きくなってしまうのかを計算してみよう。l_* における探索成功率を p_* とし、N を $\alpha^N \leq l_* \leq \alpha^{N+1}$ を満たす自然数として、

$$\begin{aligned}
E(l) &= \sum_{n=0}^{\infty} \alpha^n p(\alpha^n) \prod_{m=0}^{n-1} (1-p(\alpha^m)) \\
&= \sum_{n=0}^{N} \alpha^n p(\alpha^n) \prod_{m=0}^{n-1} (1-p(\alpha^m)) \\
&\quad + \sum_{n=N+1}^{\infty} \alpha^n p(\alpha^n) \prod_{m=0}^{n-1} (1-p(\alpha^m)) \\
&\leq \sum_{n=0}^{N} \alpha^n p_* + \sum_{n=N+1}^{\infty} \alpha^n (1-p_*)^{n-1-N}
\end{aligned}$$

上式の不等式は $p(l)$ が l について単調増加であることから、$n \leq N$ となる n に対して $p(\alpha^n) \leq p_*$ および $1-p(\alpha^n) \leq 1$、$n \geq N+1$ となる n に対して $p(\alpha^n) \leq 1$ および $1-p(\alpha^n) \leq 1-p_*$ が成り立つため。$\alpha(1-p_*) < 1$ のと

き、上式から、

$$
\begin{aligned}
E(l) &\leq \frac{(\alpha^{N+1}-1)p_*}{\alpha-1} + \frac{(\alpha)^{N+1}}{1-\alpha(1-p_*)} \\
&\leq \left(\frac{\alpha p_*}{\alpha-1} + \frac{\alpha}{1-\alpha(1-p^*)}\right) l_* - \frac{p_*}{\alpha-1}
\end{aligned}
$$

となって l_* の定数倍に収まることがわかる。

次に、もう少し賢い次のようなアルゴリズム (Alg.7.3.2) を考える。ここで、abt[] は配列を表し、|abt| は abt[] の長さ、abt[n] は、$n <$ |abt| のとき abt[] の $n+1$ 番目の要素、そうでないとき ε（空列）をあらわすものとする。

Algorithm 5 オラクル型最適化アルゴリズム A に対する繰り返しアルゴリズム 3

```
入力: オラクル O,k ∈ ℝ
L := 1, abt[] := ε, wst[] := ε
while 1 do
  a を新しく実行する A(O) のインスタンスとする。
  for n := 0 ; n < |abt| ; n := n + 1 do
    a が O に α^n 回問い合わせるまで実行し、その最後の出力を x とする。
    if wst[n] > x または wst[n] = ε then
      wst[n] := x、 a と abt[n] をスワップ。
    end if
    if O(x) ≥ k then
      return x
    end if
  end for
end while
```

Alg.7.3.2 では、各 α^i の区切りごとに過去のプログラムの最良値と比較して、良いほうのみを伸ばす、ということを繰り返している。もう少し丁寧に書くと、Alg.2 のように α 倍ずつ伸ばしてリセットするだけではなくて、各 α^i の区切りごとに過去の中断したプログラムを保存しておいて、今回と保存されているものとで最良値の大きいほうのプログラムを選んで、のこり $\alpha^{i+1}-\alpha^i$ だけ実行するということを繰り返している。Alg.3 中の FOR ループを RETURN せずに抜けた時は、プログラムを i 用意して実行し、α^i の区切りごとに一番悪いものから先に消去して行くのと同じ結果になっている。したがって、Alg.1 で L に最適評価回数を代入した場合よりも良い結果が得られる可能性が大きいことが期待できる。

繰り返し述べるが、NFL から、全てのオラクル型最適化アルゴリズムに対して Alg.3 を適用するとより良い結果が得られる (期待評価回数を下げることができる) ということはいえない。また、どのような一般的な条件を仮定すると、その条件を満たすオラクル型最適化アルゴリズムと最適化問題に対し

て良い結果を得ることができるかということも、現実に即した条件で決めることは難しい。

したがって、ここでは、現実には満たされない場合がほとんどだが、理想的な条件として、出力 x の評価値 $(f(x))$ が内部状態などに依存せずに増加量がランダムに決まると仮定したとき、Alg.3 が実際に有効かどうかを考察してみる。すなわち、出力の評価値がランダムウォークであると仮定した場合、Alg.3 がどの程度有効かを考えてみよう。

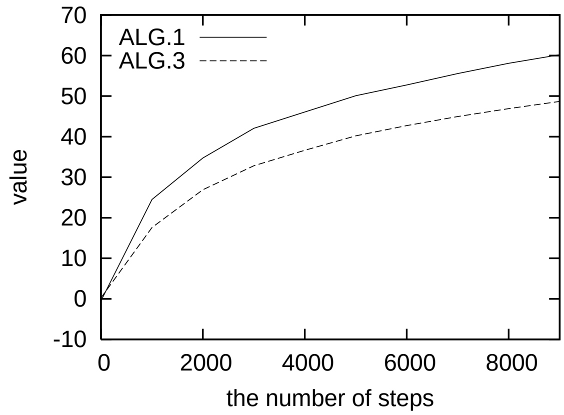

図 23: 各戦略に対するランダムウォークでの変位の平均

Fig.7.3.2 の ALG.1 で示されるグラフは、2 回の n ステップのランダムウォークの結果のうち良いほうを選ぶことによって得られる値の平均を示しており、Alg.1 で $L = n$ で 2 回繰り返したときの最大評価値の平均に相当している。一方、ALG.3 のグラフの Alg.3 の戦略で $\alpha = 2$ の場合の最大評価値である。横軸 n に対するステップ数は ALG.3 の場合もほぼ $2n$ である。なお、ALG.1 のケースは n ステップ後の分布が平均 0 分散 $\sqrt{n}$ の正規分布となることから計算することができて、$\sqrt{n/\pi}$ である。結果を見ると、この場合 Alg.3 の戦略が幾分有効であることがわかる。

次に、具体的な例をとって Alg.3 を検討して見てみよう。サイン関数への四則演算を用いたリグレッション問題を GP で解くことを考える (図 7.3.2)。

ノードは $\{+, *, 1/(\cdot), -(\cdot), (\cdot)^2\}$ および変数として x、定数として $\{\pm c^{\pm 1} | c = 0, 1, 2, 3, 4\}$、集団数は 2000、選択:トーナメントサイズ 2 のトーナメント戦略、生成オペレータは mutation と cross over を使用している。また、木が表現する g に対する評価関数は、

$$f(g) = -([-2\pi, 2\pi] \text{ までの 32 点での標準誤差の平均})$$

とし、GP に用いる評価関数としては、木のサイズについてペナルティを加えた

$$f_{gp}(g) = f(g)(1 + max(64, g \text{ の木のサイズ})/2560)$$

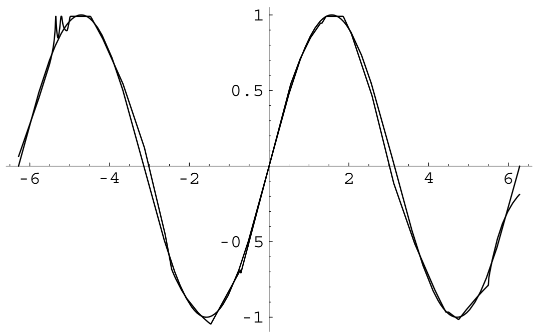

図 24: 例題:sin 関数に対するリグレッション

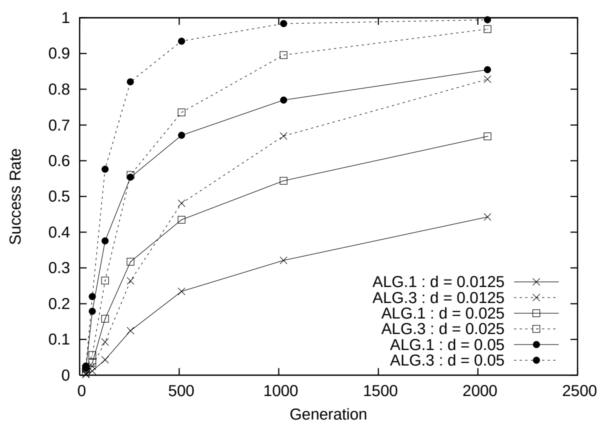

図 25: 成功確率の比較

とした。

$I(M, t, 0.99)$ は、t 世代でリセットして繰り返したとき、与えられた評価値 d に対して、確率 0.99 で $f(g) > d$ となるような g を出力するための最低の評価回数のことであった。以降 $I(M, t, 0.99)$ のことを I99 評価回数とよぼう。いずれの誤差の値に対しても、最適停止世代数は 256 付近で、(ALG.3 の I99 評価回数)/(ALG.1 の I99 評価回数) は 0.4-0.5 程度であった。また、図 7.3.2 を見てもわかるように、最適停止世代数以降で、ALG.3 での I99 評価回数は ALG.3 とくらべ上昇の程度が穏やかであるこれは、停止世代数に対して敏感でないということである。

この実験では、ランダムウォークの時と比べ、ALG.3 の優位性が目立つが、これは初期において当たり外れのようなものが GP に存在すること、つまり、少ない世代数での評価値の値と、将来の評価値の値とにランダムウォークの場合よりも強い相関関係があることが理由となっていると考えている。

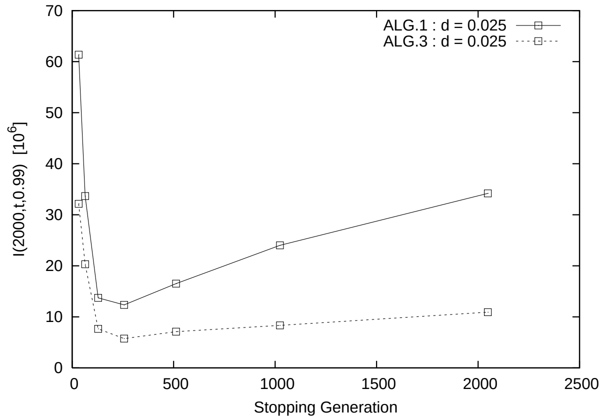

図 26: I99 評価回数の比較

8 結論

参考文献

[1] Angluin, D.: Inductive inference of formal languages from positive data. Information and Control **45** (1980) 117–135

[2] Angluin, D.: Inference of reversible languages. Journal of the Association for Computing Machinery **29** (1982) 741–765

[3] Barto, A. G. and Mahadevan, S.: Recent advances in hierarchical reinforcement learning. Discrete Event Dynamic Systems: Theory and Applications **13** (2003) 41–77

[4] Bertsekas, D. P. and Tsitsiklis, J. N.: Neuro-dynamic Programming. Athena Scientific (1996) Sec. 5.6

[5] Hirshfeld, Y and Jerrum, M. and Moller, F.: A polynomial algorithm for deciding bisimilarity of normed context-free processes. Theoretical Computer Science **158** (1996) 143–159

[6] Kobayashi, S. : Iterated transductions and efficient learning from positive data: A unifying view. In Proceedings of the 5th International Colloquium on Grammatical Inference, **1891** in Lecture Notes in Computer Science (2000) 157–170

[7] Kaelbling, L. P. , Littman, M. L. and Cassandra, A. R.: Planning and acting in partially observable stochastic domains. Artificial Intelligence **101** (1998) 99–134

[8] Sakakibara, Y.: Recent advances of grammatical inference. Theoretical Computer Science **185** (1997) 15–45

[9] Sutton, R. S. and Barto, A. G.: Reinforcement Learning: An Introduction. MIT Press (1998)

[10] Wakatsuki, M. Teraguchi, K. and Tomita, E.: Polynomial time identification of strict deterministic restricted one-counter automata in some class from positive data. Proceedings of the 7th International Colloquium on Grammatical Inference **3264** in Lecture Notes in Computer Science (2004) 260–272

[11] Wetherell, C.S.: Probabilistic languages: A review and some open questions. Computing Surveys, **12** No. 4 (1980) 361–379

[12] Yokomori, T.: Polynomial-time identification of very simple grammars from positive data. Theoretical Computer Science **298** (2003) 179–206

[13] Yoshinaka, R.: Polynomial-Time Identification of an Extension of Very Simple Grammars from Positive Data. Proceedings of the 8th International Colloquium on Grammatical Inference **4201** in Lecture Notes in Computer Science (2006) 45–58

発表文献

- 投稿中 "Probabilistic Generalization of Simple Grammars and Its Application to Reinforcement Learning", Theoretical Computer Science (the Special Issue ALT 2006)
- 2006 年 "Probabilistic Generalization of Simple Grammars and Its Application to Reinforcement Learning", International Conference on Algorithmic Learning Theory 2006, Lecture Notes in Computer Science 4264 348-362
- 2006 年 「確率単純文法における一般化と強化学習への応用」 数理解析研究所講究録 No.1489, pp.71-77, 2006 (冬の LA シンポジウム)
- 2003 年 「単純文法を用いた強化学習の拡張」, 日本神経回路学会第 13 回全国大会講演論文集, 2003