

構造推定を用いた遺伝的プログラミング

-隠れアノテーション確率文法 GP-

Genetic Programming with Estimation of Subroutines

長谷川 禎彦*¹
Yoshihiko Hasegawa

伊庭 齊志*²
Hitoshi Iba

東京大学新領域創成科学研究科基盤情報学専攻

Department of Frontier Informatics, Graduate School of Frontier Sciences, The University of Tokyo

Genetic Programming (GP) which mimics the natural evolution to optimize functions and programs, has been applied to many problems. Recently, probabilistic evolutionary algorithms have been proposed by many researchers to overcome the defect of the random operations. Although probabilistic context free grammar is often used for the probabilistic program evolution, it assumes the independence among the production rules. We have proposed a new program evolution algorithm based on probabilistic context free grammar using latent annotation. Computational experiments on two subjects (royal tree problem and DMAX problem) demonstrate that our new approach is highly effective compared to prior approaches.

1. はじめに

本論文では、隠れアノテーション確率文脈自由文法 (PCFG-LA: Probabilistic Context Free Grammar with Latent Annotations) [Matsuzaki 05] を用いた新しい進化計算アルゴリズム PAGE (Programming with Annotated Grammar Estimation) を提案する。

GP 及び GA の主要なオペレータは交叉と突然変異である。交叉と突然変異をランダムに適用することで、解の進化が行われる。このような考え方は生物の進化をモデル化したものであるが、近年、分布推定という観点から進化計算手法が見直されており、それに基づいた手法が多くの関心を集めている [Larrañaga 01]。従来は、GA ベースのアルゴリズムに対する手法が主流であったが、最近では確率モデルを GP に適用した手法も多く提案されている。これら GP に基づく手法は確率モデル遺伝的プログラミング (PMBGP: Probabilistic Model Building Genetic Programming) と呼ばれる。

PMBGP としては、現在までに多くの手法が提案されているが [長谷川 07, Hasegawa 06]、大きく分けて二つの手法に分類することができる: プロトタイプ木に基づく手法と確率文脈自由文法 (PCFG: Probabilistic Context Free Grammar) に基づく手法である。PCFG に基づく手法では、関数やプログラムを文脈自由文法 (CFG: Context Free Grammar) で表し、その適用確率を学習する。

PCFG においては文脈依存性を考慮しない。基本的な PCFG に基づく GP では、単純に各パラメータを推定し、推定したパラメータに基づき、新しい個体を生成する。このような手法はパラメータ推定が容易であるという利点があるものの、GP においてはノード間の依存関係が存在するため、文脈非依存という仮定は強すぎる場合が多い。

言語処理においては隠れアノテーション PCFG (PCFG-LA: PCFG with Latent Annotations) [Matsuzaki 05] と呼ばれる手法が提案されている。この手法では、アノテーションは隠れており観測されない。PCFG-LA ではアノテーション付きの生成規則確率は EM アルゴリズムによって推定される。このようなアノテーションは「深さ」などのヒューリスティクスに基づくアノテーションと比較して柔軟なモデルであり、GP

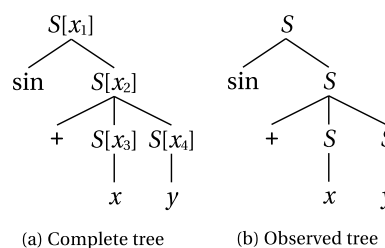


図 1: アノテーション付きの木 (a) と観測される木 (b)。

のように様々な問題に適用されることを前提としたアルゴリズムにおいては、非常に適していると考えられる。そこで本論文では、PCFG-LA を基本モデルとした PMBGP である PAGE (Programming with Annotated Grammar Estimation) を提案する。

2. PCFG-LA

基本的な確率文脈自由文法 [北 99] では、文脈非依存性を仮定している。しかし、この仮定は強すぎる場合が多く、アノテーションを用いることで非依存性の仮定を緩和させる手法が、言語処理の分野で提案されている。アノテーションは、非終端記号に付加情報をつけるものであるが、付加情報としては、親ノード、兄弟ノードなどが一般的である。このような、あらかじめ与えられたアノテーションではなく、アノテーション自体も学習データから推定する手法が提案されている [Matsuzaki 05]。この文献では、PCFG-LA とよばれる隠れアノテーションのモデルを提案しており、EM アルゴリズムによるパラメータ導出手法を示している。PCFG-LA においては、全ての非終端記号にはアノテーションがラベル付されている。非終端記号は、アノテーション付表現では $A[x]$ のように表現される。 A は非終端記号であり、 x はアノテーションを表す (アノテーションは観測されない)。図 1 はアノテーション付での木構造を表したもの (a) と、観測される木構造を表したものである (b)。

完全データ T , X の尤度は式 1 で表される。

$$P(T, \mathbf{X}; \Theta) = \pi(S[x_1]) \prod_{r \in D_{T[\mathbf{X}]}} \beta(r) \quad (1)$$

ここで, T は導出木, x_i は i 番目の非終端記号のアノテーション, \mathbf{X} は $\mathbf{X} = \{x_1, x_2, \dots\}$, $\pi(S[x])$ はルートでの $S[x]$ の確率, $\beta(r)$ は生成規則 r の確率, $D_{T[\mathbf{X}]}$ は T において用いられているアノテーション付生成規則の集合 (重複あり), Θ はパラメータ $\Theta = \{\pi, \beta\}$ である.

周辺化尤度は, 式 1 を \mathbf{X} についての和をとることで計算される (式 2).

$$P(T; \Theta) = \sum_{\mathbf{X}} P(T, \mathbf{X}; \Theta). \quad (2)$$

アノテーションは観測されないため, アノテーションを含む生成規則の確率は EM アルゴリズムによって推定される. 二つのパラメータ Θ' と Θ における, 木 T の対数尤度の差を考える. 対数尤度の差の下限は以下 (式 3) のように計算される.

$$\begin{aligned} & \log P(T; \Theta') - \log P(T; \Theta) \\ &= \sum_{\mathbf{X}} P(\mathbf{X}|T; \Theta) \log \frac{P(T, \mathbf{X}; \Theta')}{P(T, \mathbf{X}; \Theta)} \frac{P(\mathbf{X}|T; \Theta)}{P(\mathbf{X}|T; \Theta')} \\ &\geq \sum_{\mathbf{X}} P(\mathbf{X}|T; \Theta) \log \frac{P(T, \mathbf{X}; \Theta')}{P(T, \mathbf{X}; \Theta)} \end{aligned} \quad (3)$$

これから, EM アルゴリズムによって最大化すべき $Q(\Theta'|\Theta)$ は式 4 で表される. 式 4 を最大化することで, パラメータ更新規則が求められる (2.2 節).

$$Q(\Theta'|\Theta) = \sum_{T_i \in \mathbf{T}} \sum_{\mathbf{X}_i} P(\mathbf{X}_i|T_i; \Theta) \log P(T_i, \mathbf{X}_i; \Theta') \quad (4)$$

自然言語においては, チョムスキー標準型 (CNF: Chomsky Normal Form) を仮定する場合が多い. しかし, プログラム進化においては, チョムスキー標準型にするメリットがあまりなく, また用いないほうが関数を分かりやすく表現することができる. 木構造 GP によって表すことのできる, 全ての関数は以下の生成規則 (式 5) によって表現できる.

$$S \rightarrow gS \dots S. \quad (5)$$

$S \in \mathcal{N}$ であり, $g \in \mathcal{T}$ である (\mathcal{N}, \mathcal{T} は CFG における非終端, 終端ノードの集合を表す. なお, GP における関数, 終端ノードの集合は \mathfrak{F} 及び \mathfrak{T} と表記する.). g は $g \in \mathfrak{F} \cup \mathfrak{T}$ であり, GP においては関数ノード ($+$, $-$, \sin , \cos など) 又は終端ノード (x, y など) を表す. \mathcal{N} としては S 以外用いる必要はない. 式 5 の右辺にある S の個数は, g の引数の数に等しい. 例えば, $g = +$ の場合 $S \rightarrow +SS$ であり, $g = x$ の場合 $S \rightarrow x$ となる.

アノテーション付の生成規則は式 6 によって表すことができる.

$$S[x] \rightarrow gS[z_1] \dots S[z_a] \quad (6)$$

ここで $x, z_m \in H$ であり, a は g の引数の数を表す. 今 h をアノテーションの数 (H の要素数) とし, g の引数の数を a とする. この場合, 式 6 を表すのに必要なパラメータ数は h^{a+1} となる. アノテーションの数が多くなるにつれ, パラメータ数

が指数的に増加し, パラメータ推定の計算量の点や精度の面で大きな問題となってしまふ. そこで, 本提案手法では式 7 で表されるような生成規則に限定する.

$$S[x] \rightarrow gS[y]S[y] \dots S[y] \quad (7)$$

この生成規則では, 右辺のアノテーションは全て同じであると仮定している. このような仮定を置くことで, パラメータの数は h^2 となる. GP で扱う関数は可換である関数 ($+$, \times など) が多数存在する. そのため, 上で示すような仮定は自然であると考えられる.

2.1 前向き・後向き確率

式 4 を計算するためには, 前向き・後向き確率を用いる必要がある. 後向き確率 $b_T^i(x)$ は i 番目の非終端記号 $S[x]$ より下の部分が生成される確率であり, 前向き確率 $f_T^i(y)$ は i 番目の非終端記号 $S[y]$ より上の部分が生成される確率である. 前向き・後向き確率は再帰的に計算することが出来る (式 8, 9, 10).

$$b_T^i(x) = \sum_{y_i \in H} \beta(S[x] \rightarrow g_i S[y] \dots S[y]) \prod_{j \in ch(i, T)} b_T^j(y) \quad (8)$$

$$\begin{aligned} f_T^i(y) &= \sum_{x \in H} f_T^{pa(i, T)}(x) \beta(S[x] \rightarrow g_T^{pa(i, T)} S[y] \dots S[y]) \\ &\quad \times \prod_{j \in ch(pa(i, T), T), j \neq i} b_T^j(y) \quad (i \neq 1) \end{aligned} \quad (9)$$

$$f_T^i(y) = \pi(S[y]) \quad (i = 1) \quad (10)$$

ここで, $ch(i, T)$ は T における i 番目の非終端記号の子ノードインデックスの集合を返す関数であり, $pa(i, T)$ は i 番目の非終端記号の親ノードインデックスを返す関数である. g_T^i は i 番目の非終端記号に接続された終端記号を表す.

前向き・後向き確率を用いることで, $P(T; \Theta)$ は以下のように計算することが出来る.

$$P(T; \Theta) = \sum_{x \in H} \pi(S[x]) b_T^1(x) \quad (11)$$

$$\begin{aligned} P(T; \Theta) &= \sum_{x, y \in H} \beta(S[x] \rightarrow gS[y] \dots S[y]) f_T^i(x) \\ &\quad \times \prod_{j \in ch(i)} b_T^j(y) \quad (i \in cover(g, T)) \end{aligned} \quad (12)$$

$cover(g, T_i)$ は終端 g を生成する規則が, 木 T のどの非終端ノードから始まっているかを表す.

2.2 パラメータ更新規則

前向き・後向き確率を用い, $Q(\Theta'|\Theta)$ を最大化することでパラメータ更新規則が求まる. 本論文で用いる生成規則は CNF ではなく, 又右辺のアノテーションが同じであるという仮定を置いているため, 文献 [Matsuzaki 05] にある更新規則とは若干異なる (導出は省略する).

$$\pi'(S[x]) \propto \pi(S[x]) \sum_{T_i \in \mathbf{T}} \frac{b_{T_i}^1(x)}{P(T_i)} \quad (13)$$

Algorithm 1 PAGE

1. Generate initial individuals.
 $g \leftarrow 0$
 $\mathcal{D}_0 \leftarrow \text{generate_individual}(\Theta_0)$
2. Select promising individuals ($M \times P_s$).
 $\mathcal{D}_g^{\text{sel}} \leftarrow \text{select_individual}(\mathcal{D}_g)$
3. Estimate parameters ($\Theta = \langle \pi, \beta \rangle$) using EM algorithm.
 $\Theta_g \leftarrow \text{estimate_parameter}(\mathcal{D}_g^{\text{sel}})$
4. Generate individuals with parameters.
 $g \leftarrow g + 1$
 $\mathcal{D}_g \leftarrow \text{generate_individual}(\Theta_g)$

Steps from 2 to 5 are repeated until termination criteria are met.

変数	パラメータ	値
M	集団数	—
h	アノテーション数	—
P_s	選択率	0.1
P_e	エリート率	0.1

表 1: PAGE の主要パラメータ .

$$\beta'(S[x] \rightarrow gS[y] \dots S[y]) \propto \beta(S[x] \rightarrow gS[y] \dots S[y]) \times \sum_{T_i \in \mathbf{T}} \frac{1}{P(T_i)} \sum_{j \in \text{cover}(g, T_i)} f_{T_i}^j(x) \prod_{k \in \text{ch}(j, T_i)} b_{T_i}^k(y) \quad (14)$$

式 3 から分かるように, $Q(\Theta|\Theta')$ を最大化することで周辺化対数尤度 ($L(\Theta; \mathbf{T}) = \log P(\mathbf{T}; \Theta) = \sum_{T_i \in \mathbf{T}} \log P(T_i; \Theta)$) は単調に増加する . パラメータの初期値は, 文献 [Matsuzaki 05] と同様の値を用いる (式 15) .

$$\beta(S[x] \rightarrow gS[y] \dots S[y]) \propto e^{-\gamma} \beta(S \rightarrow gS \dots S) \quad (15)$$

式 15 において, γ は $[-\log 2, \log 2]$ の一様乱数であり, $\beta(S \rightarrow gS \dots S)$ はアノテーション無し生成規則の適用確率を表す . さらに, $\beta(S[x] \rightarrow gS[x] \dots S[x]) = 0$ とする . これは, GP の扱う構造は有限サイズであり, 再帰的な生成規則を必要としないことに由来する .

3. アルゴリズム

PCFG-LA を基本文法としたプログラム進化アルゴリズム PAGE (Programming with Annotated Grammar Estimation) を提案する . PAGE の Pseudo Code をアルゴリズム 1 に, PAGE の主要パラメータを表 1 に示す . ここで, アノテーション数 h はあらかじめ与えられる .

4. 計算機実験

提案手法 PAGE の有効性を示すために, 二つのベンチマークテストに適用し, 探索能力を調べた . ベンチマークテストとしては, Royal Tree 問題と騙し最大値 (DMAX) 問題を選択した . それぞれの問題の詳細については文献 [Punch 98, Hasegawa 06] を参照されたい . 比較に用いたアルゴリズムは PCFG-GP (アノテーションを用いない PCFG を用いた GP) と SGP (Simple GP) である .

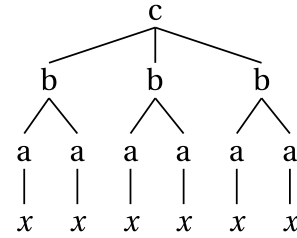


図 2: Royal Tree 問題での最適構造 (レベル c) .

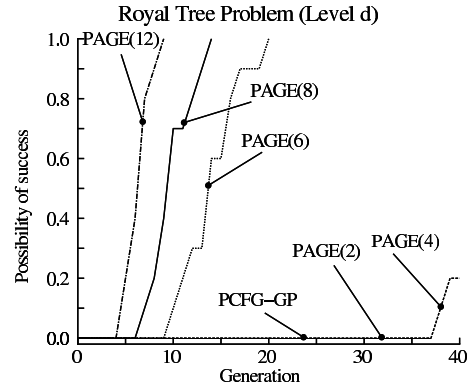


図 3: 世代毎の探索成功率 .

4.1 Royal Tree 問題

隠れアノテーションの有効性を調べるために, 提案手法である PAGE, 隠れアノテーションのない通常の PCFG-GP の二つのアルゴリズムを, Royal Tree 問題 [Punch 98] (レベル d) に適用した . Royal Tree 問題は GP の探索メカニズムを調べるために考案された問題であり, GA で有名な Royal Road 関数 [Mitchell 92] を木構造に拡張した問題である . Royal Tree 問題では, 小さな部分構造を次々に組み合わせることで, より大きな部分構造を獲得する (図 2) .

PAGE におけるアノテーション数は $h = 2, 4, 6, 8, 12$ の 5 つのパラメータで実験を行った . 個体数 M は両方のアルゴリズムで $M = 1000$ とした . PCFG-GP 及び PAGE で用いた生成規則は以下のような規則である . 小文字は CFG での終端記号を表し, 大文字は非終端記号を表す . また $\mathcal{S} = \{a, b, c, d\}$, $\mathcal{T} = \{x\}$ である .

- $S \rightarrow aS$
- $S \rightarrow bSS$
- $S \rightarrow cSSS$
- $S \rightarrow dSSSS$
- $S \rightarrow x$

表 2: PCFG-GP によって推定された Royal Tree 問題の生成規則 .

Production rule	Probability
$S \rightarrow x$	0.60
$S \rightarrow aS$	0.10
$S \rightarrow bSS$	0.10
$S \rightarrow cSSS$	0.10
$S \rightarrow dSSSS$	0.09

表 3: PAGE によって推定された Royal Tree 問題の生成規則 (アノテーション付) の確率 ($h = 6$) . なお, 確率が小さい規則は省略している .

Production rule	Probability
$S[0] \rightarrow x$	1.0
$S[1] \rightarrow a S[5]$	0.028
$S[1] \rightarrow b S[5] S[5]$	0.86
$S[1] \rightarrow c S[5] S[5] S[5]$	0.026
$S[1] \rightarrow d S[5] S[5] S[5] S[5]$	0.059
$S[1] \rightarrow x$	0.019
$S[2] \rightarrow d S[4] S[4] S[4] S[4]$	0.14
$S[2] \rightarrow x$	0.86
$S[3] \rightarrow x$	1.0
$S[4] \rightarrow c S[1] S[1] S[1]$	0.99
$S[5] \rightarrow a S[0]$	0.33
$S[5] \rightarrow a S[2]$	0.23
$S[5] \rightarrow a S[3]$	0.25
$S[5] \rightarrow b S[0] S[0]$	0.031
$S[5] \rightarrow c S[0] S[0] S[0]$	0.059
$S[5] \rightarrow c S[3] S[3] S[3]$	0.018
$S[5] \rightarrow d S[0] S[0] S[0] S[0]$	0.042
$S[5] \rightarrow x$	0.025

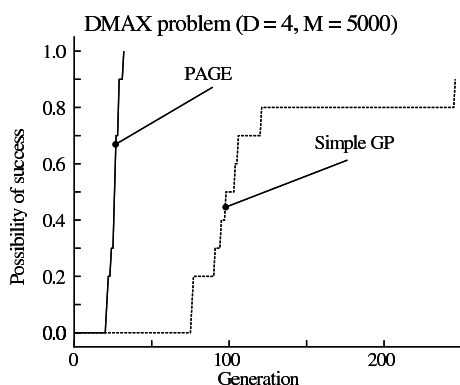


図 4: 世代毎の探索成功率 .

図 3 は世代毎の探索成功率を示したものである . この図から分かるように, PCFG-GP は最適解の獲得に成功していない . Royal Tree 問題の最適解の導出木に用いられる生成規則の多くは $S \rightarrow x$ である . そのため, 単純な PCFG を用いた場合, 場所にかかわらず $S \rightarrow x$ が適用されるため (表 2), 生成された木構造のサイズが小さくなる . 結果として最適解を上手く獲得できていない . 提案手法 PAGE においては, それぞれのノード ($\mathcal{F} \cup \mathcal{T} = \{a, b, c, d, x\}$) を生成する非終端 S に, 別のアノテーションが付く . 表 3 はアノテーション数 $h = 6$ の場合に, PAGE によって推定された生成規則である . この表から, 例えば関数 b を生成するには $S[1]$, 関数 c を生成するには $S[4]$ が用いられており, 部分構造を高い確率で生成することが出来る .

4.2 騙し最大値 (DMAX) 問題

提案手法の, SGP に対する有効性を, 騙し最大値 (DMAX: Deceptive MAX) 問題に適用することで示す . 比較対象アルゴリズムは PCFG-GP, SGP である . DMAX 問題のパラメータは $D = 4, m = 5, r = 3$ とし, 集団数 $M = 5000$ とした . アノテーション数は $h = 6$ とし, SGP のトーナメントサイズは $T_s = 2$ とした .

PCFG-GP 及び PAGE で用いられる生成規則は以下のような規則である .

$$S \rightarrow +_5 SSSSS$$

$$S \rightarrow \times_5 SSSSS$$

$$S \rightarrow 0.95$$

$$S \rightarrow \lambda$$

ここで λ は以下で表す定数である .

$$\lambda^r = 1, \lambda \in \mathbb{C}, r \in \mathbb{N} \quad (16)$$

図 4 に, 各アルゴリズムの探索成功率を示す . PCFG-GP は一度も最適解を獲得することが出来なかったため, 図からは省いている . 図から分かるように, PAGE は SGP, PCFG-GP と比較して遥かに少ない評価回数で, 最適解を獲得していることが分かる . 文献 [Hasegawa 06] にあるように, DMAX 問題は非常に強い騙し要素を含む . このような騙し要素を含む問題に対しても, PAGE は非常に有効であることが分かる .

5. おわりに

本論文では, 隠れアノテーションモデルを用いた, 新しい確率モデル GP を提案した . 提案手法 PAGE の有効性を, Royal Tree 問題及び DMAX 問題に適用することで示した . PAGE では EM アルゴリズムによる, パラメータ推定を行っているが, 変分ベイズ法などの, より進んだ推定手法の適用などが考えられる . また, Real World 問題への適用を行ってみたいと考えている . これらの点は, これからの課題である .

参考文献

[Hasegawa 06] Hasegawa, Y. and Iba, H.: Estimation of Bayesian Network for Program Generation, in *Proceedings of The Third Asian-Pacific Workshop on Genetic Programming*, pp. 35–46, Hanoi, Vietnam (2006)

[長谷川 07] 長谷川 禎彦, 伊庭 斉志: ペイジアンネットワーク推定による確率モデル遺伝的プログラミング, 人工知能学会論文誌, 22 巻, 1 号, pp. 37–47 (2007)

[北 99] 北 研二: 確率的言語モデル, 東京大学出版 (1999)

[Larrañaga 01] Larrañaga, P. and Lozano, J. A.: *Estimation of Distribution Algorithms*, Kluwer Academic Publishers (2001)

[Matsuzaki 05] Matsuzaki, T., Miyao, Y., and Tsujii, J.: Probabilistic CFG with latent annotations, in *In Proceedings of the 43rd Meeting of the Association for Computational Linguistics (ACL)*, pp. 75–82 (2005)

[Mitchell 92] Mitchell, M., Forrest, S., and Holland, J. H.: The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance, in Varela, F. J. and Bourgine, P. eds., *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life, 1991*, pp. 245–254, Paris (1992), A Bradford book, The MIT Press

[Punch 98] Punch, W. F.: How Effective are Multiple Populations in Genetic Programming, in Koza, J. R., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D. B., Garzon, M. H., Goldberg, D. E., Iba, H., and Riolo, R. eds., *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pp. 308–313, University of Wisconsin, Madison, Wisconsin, USA (1998), Morgan Kaufmann