4Q-3

文脈自由文法の漸次学習における規則集合の探索方式の検討と改良

今田 圭太 † 中村 克彦 ‡

†‡ 東京電機大学 理工学部

1 まえがき

文脈自由文法の推論 (学習) とは, 与えられた正例および負例の記号列から, これらを満足する文脈自由文法を合成することであり, 機械学習の基礎的な研究テーマである. われわれは, 文脈自由文法の漸次学習のためのSynapseシステムを作成し改良を続けてきた. Synapseでは正例の各記号列に対してブリッジ法と呼ばれる規則合成アルゴリズムによって追加が必要な (生成) 規則を合成し, 最小規則探索と直列探索と呼ばれる探索アルゴリズムにもとづいて, 最小または最小に近い規則集合を求める. 最小規則探索は, 正例の各部分集合を満足する最小の規則集合を探索する. 一方, 直列探索は, 各正例を満足するような最小の規則を漸次的に規則集合に追加する.

本報告では, この二つの探索方式について評価を行い改良法について検討する. 直列探索は最小規則探索に比べて一般に高速であるが, 規則集合は大きくなり, 言語によっては正しい規則集合が求められないこともある. この研究の目的は直列探索方式のこのような問題点を解決し, さらに多くの文法を高速にを合成できるように改良することである.

2 Synapseシステム

Synapseは 次の形の規則からなる文法を合成することができる. ここで A, B, C は非終端記号 , a は終端記号, β , γ は終端記号または非終端記号をあらわす.

チョムスキー標準形 $A \to BC$, $A \to a$

拡張チョムスキー標準形 $A \to \beta\gamma$, $A \to B$

ブリッジ法にもとづく規則合成 正例の記号列の上向き構文解析が失敗したとき, 構文解析の結果である不完全な導出木に対して, これを完成するような規則を生成する.

最小規則探索方式 最小規則探索では, 与えられたすべての正例を導出し, すべての負例を導出しない最小の規則集合を反復深化によって求める.

入力 S_P(正例の順序集合), S_N(負例の集合)
出力 P(生成規則の集合)

1. $R_{max} \leftarrow 0$, $P \leftarrow \phi$
2. S_P の各文字列 w に対して次の操作をほどこす. 失敗したら R_{max} に 1 を加えてこれを繰り返す.
   (a) w に対して構文解析を行なう. P が w を導出したら終了.
   (b) w を導出するような R_{max} 個以内の規則を P に加える.
   (c) S_N のすべての記号列対して構文解析を行なう. 導出したら失敗.

直列探索方式 直列探索方式では, 各正例を導出し, すべての負例を導出しない最小の規則を反復深化によって求め, 規則集合に漸次追加する.

入力 S_P(正例の順序集合), S_N(負例の集合)
出力 P(生成規則の集合)

1. $P \leftarrow \phi$
2. S_P の各文字列 w に対して次の操作をほどこす.
   (a) $R_{max} \leftarrow 0$
   (b) w に対して構文解析を行なう. P が w を導出したら終了.
   (c) 次の手続きを実行する. 成功したら終了. 失敗したら R_{max} を一つ増やして繰り返す.
      i. w を導出するような R_{max} 個以内の規則を P に加える.
      ii. S_N のすべての記号列に対して 構文解析を行なう. 導出したら失敗.

3 直列探索の評価

直列探索によって合成した規則集合と最小の規則集合とを規則集合の大きさについて, 次の方法によって

Evaluations and Improvements of Search Strategies for Rule Sets in Incremental Learning of Context Free Grammars

Keita IMADA†, Katsuhiko NAKAMURA‡
†‡Tokyo Denki University, School of Science and Engineering
350-0394 Ishizaka Hatoyama-cho Hiki-gun Saitama-ken Japan
†imada@naklab.k.dendai.ac.jp ‡nakamura@k.dendai.ac.jp

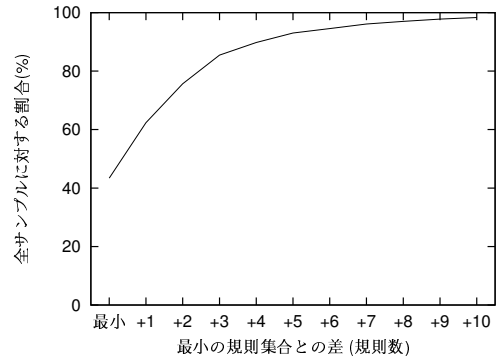

図 1: 直列探索における規則数の増え方

比較した.

1. ランダムに規則集合 P を合成する
2. 長さ n までの文字列をすべて発生し, P によって構文解析を行なう , 成功したものを正例の集合 S_P, 失敗したものを負例の集合 S_N とする.
3. S_P と S_N を Synapse システムに与え, 直列探索と最小規則探索のそれぞれの方式で規則合成を試みる.

規則数 5〜12 の文法から, 長さ 1〜10 までのすべてのの正負のサンプルセットを Synapse に与え, 直列探索の評価を行なった. 作成した約 1000 個の文法のうち, 直列探索によってこの 43.4% において, 最小規則探索と同じ大きさの規則集合を合成した. さらに 85.4% の文法が 最小規則数 + 3 の範囲内に収まり, 94.5% の文法において 最小規則数 + 6 以下の規則集合を合成した (図 1).

4 直列探索の改良

既に得られた規則集合が負例を導出せず, 開始記号を右側に含むような規則を含んでいなければ, この規則集合に何らかの規則集合を追加して, 任意の文脈自由文法と等価な文脈自由文法が得られることは明らかである. したがって, 直列探索によって規則集合が収束するか否かはブリッジ法によって, どのような規則が生成されるかに依存している.

Synapse は次の文法の言語に対して直列探索を適用したとき, 正しい文法が得られなかった (表 1).

$$S \to Pa,\ P \to ab \mid aQ \mid PP, \quad Q \to Pb$$

この例では負例を追加する毎に規則集合が大きくなり, 規則集合が収束せず, 十分な正例と負例を与えても直列探索において正しい文法を得ることができなかった. これは, ブリッジ法によって合成される規則が, 既に得られた規則集合とその規則集合によってサンプルを構

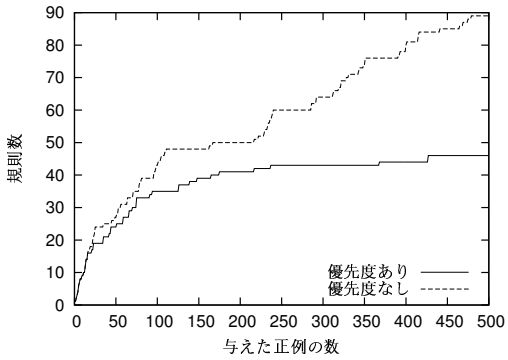

図 2: 優先度を用いた直列探索の改善

表 1: 直列探索が成功しない例

与えられた正例	追加した規則
aba	$S \to aP, P \to ba$
abab	$S \to SP$
aabba	$P \to aQ, Q \to bP$
aaabbba	$Q \to aR, R \to bQ$
aaaabbbba	$R \to aT, T \to bR$
...	...

文解析した結果に依存すること, また最小の規則集合を選択していることに起因する. このような, 直列探索が成功しない例について次の方式を試みた.

1. 候補となる規則集合の中から, 2 つの規則を選択し, これらを規則集合に追加する.
2. ブリッジ法によって生成された規則に対して, 規則の形に応じた優先度を付加する.

この言語については, 上の 1 の方式で正しい規則集合を合成した. また, ww の形をしていない文字列の集合においても正例の与え方によっては規則数が収束しない場合があるが, 2 の方式によって 図 2 で示されるように直列探索による規則数の収束性に改善がみられた.

5 むすび

直列探索では, 多くの言語において最小の規則集合よりも規則数が増えることを許容することで, 最小規則探索よりも高速に文法を合成することができるが, 言語によっては十分な例を与えても, 規則集合が収束しない例の存在が明らかになった. このような例に対して, いくつかの改良を適用し, その有効性について確認した.

参考文献

[1] K. Nakamura, Incremental Learning of Context Free Grammars by Bridging Rule Generation and Search for Semi-Optimum Rule Sets, *8th International Colloquium on Grammatical Inference(ICGI 2006), LNAI 4201 Springer pp72-84(2006)*