F-001

終端記号のクラスタリングを用いた遺伝的プログラミングによるルール抽出

Rule Extraction by Genetic Programming with Clustered Terminal Symbols

原 章† 田中 晴子‡ 市村 匠† 高濱 徹行†

Akira Hara Haruko Tanaka Takumi Ichimura Tetsuyuki Takahama

1 はじめに

遺伝的プログラミング (Genetic Programming; GP)[1] は，木構造プログラムの進化的最適化手法である．木の葉となる終端記号と，節点となる非終端記号の組み合わせを遺伝操作により変化させ，より良い解を探索する．GP をデータベースからのルール抽出問題に適用する場合，各属性を終端記号とする設定が用いられることが多いが [2]，属性数が大きいほど探索領域が膨大になり探索の効率が悪化する．この問題を解決するため，属性間の類似度に基づく突然変異を導入した GP[3] が提案された．

本研究では別のアプローチとして，探索の初期ではクラスタリングにより類似した属性をまとめて属性数を減らすことで，粗く大域的な探索を行い，その後は同一クラスタとなった属性の情報を利用して局所的に探索を進める方法を提案する．

2 GP によるルール抽出

本研究では，GP によるデータベースからのルール抽出に焦点をあてる．表 1 に示すような M 個の属性と 2 クラスに分類された結果からなるデータベースに対して，GP を用いて IF_THEN ルールを抽出する場合を考える．この IF_THEN ルールの例としては，次のようなものが考えられる．

IF $(A_1 > 0.3)$ or $((A_3 < 0.4)$ and $(A_M < 0.5))$ THEN Class 1

このように抽出対象をどちらか一方のクラスに決めると，ルールの後件部が定まり固定となるため，GP では IF_THEN ルールの前件部のみを最適化すればよい．

前件部を木構造プログラムにより表現するとき，終端記号 T には，各属性およびそれがとりうる値を表す実数，すなわち $T = \{A_1, A_2, \ldots, A_M, \Re\}$ を用い，また，関数記号には $F = \{and, or, >, <\}$ を用いることが多い．そして，学習データの分類精度を適応度として，個体集団を進化させることにより，分類精度の高いルールを獲得できる．

しかし，データベースの属性数が非常に多いデータベースに対して，この方法を適用した場合，終端記号集合のサイズも同時に大きくなり，組み合わせ数が膨大となるため，探索性能が悪化する．

表 1: データベースの例

事例 ID \ 属性	A_1	A_2	A_3	...	A_M	Class
1	0.2	0.5	0.2		0.7	0
2	0.2	0.2	0.3		0.4	1
3	0.1	0.5	0.6		0.3	0
4	0.9	0.3	0.5		0.5	1
⋮	⋮	⋮	⋮		⋮	⋮
N	0.5	0.9	0.2		0.3	1

3 終端記号のクラスタリングを用いた GP

提案手法では，まずクラスタリングにより類似した属性からなるクラスタを形成する．探索初期は，各クラスタ中心を新たな 1 つの属性と見なし，このクラスタ中心属性のみを GP の終端記号に利用することで探索空間を狭めることを狙う．また，探索後半はルールの表現能力の向上のため，扱う属性をデータベースが本来持つ属性に戻すが，先のクラスタの情報を利用した局所探索を行うことにより，探索の効率化を図る．

まず，属性のクラスタリングの方法について説明する．クラスタリングを行うには対象同士の距離を定義する必要がある．表 1 のように，M 個の属性 $A_1, A_2, ..., A_M$ を持つ事例数 N のデータベースは N 行 M 列の行列と見なすことができる．この行列を各列について最大値 1，最小値 0 となるように正規化する．正規化した行列の各列を属性の特徴を表すベクトルととらえ，これをクラスタリングの対象とする．a_{ji} を属性 A_i における事例 j の値を正規化した値とし，任意の属性 A_i と A_k の距離 $d(A_i, A_k)$ を以下で定義する．

$$d(A_i, A_k) = \sqrt{\sum_{j=1}^{N} (a_{ji} - a_{jk})^2} \quad (1)$$

クラスタリングの方法は，階層的クラスタリングと K-means 法に代表される非階層的クラスタリングに大きく分けられる．本研究では，クラスタリングに K-means 法を用いて探索効率の改善を図った K-means-GP と，階層的クラスタリングを用いた Hierarchical Clustering-GP の 2 種類の検討を行った．以下で各手法について述べる．

†広島市立大学大学院情報科学研究科知能工学専攻
Graduate School of Information Sciences, Hiroshima City University
‡広島市立大学情報科学部知能情報システム工学科
Faculty of Information Sciences, Hiroshima City University

3.1 K-means-GP (K-GP)

探索初期は K-means 法で作られた K 個のクラスタ中心を属性として GP を実行する．途中のある世代で，各個体の持つルールのクラスタ中心属性の部分を，そのクラスタ中心に最も近い元の属性に置き換える．以後，元の属性のみを終端記号とし，属性の突然変異先を同じクラスタ内の属性に限定する．

なお，探索初期は K-GP と同一であるが，途中で属性の置換を行わず，属性の突然変異先をクラスタ中心を含む同クラスタ内の属性に限定する手法についても実験を行った．これを c(combined)K-GP と呼ぶこととする．

3.2 Hierarchical Clustering-GP (HC-GP)

階層的クラスタリングは，クラスタリングを逐次的に行うため，クラスタリングの過程を保存しておくことにより，クラスタ数の変化を扱えるという利点がある．凝集型の階層的クラスタリングは，最初，分類対象の各データを要素数 1 のクラスタとし，クラスタ間の距離が最も近いクラスタから逐次的に統合する．本研究では，クラスタ間の距離は，最短距離法により求めることとする．

階層的クラスタリングで作った各クラスタからクラスタ中心に近い 1 つの属性を選出し，それを終端記号とする．探索の過程で，一定世代の間適応度の上昇が見られない場合は，クラスタを分割し，終端記号に使用する属性を増やす操作を行う．

4 実験と考察

提案手法を 166 個の実数値属性をもつ 2 クラス分類問題である MUSK database[4] に適用した．従来手法 (sGP) と提案手法の適応度の比較を図 1 と表 2 に示す．

いずれの提案手法でも従来手法より探索効率が改善されている．また，K-GP では属性の置き換えを行った 300 世代目で適応度が下がることがわかる．この適応度の落ち込みの原因は，クラスタ中心とそれに最も近いデータベース本来の属性との差によるものである．cK-GP ではクラスタ中心で作られたルールをそのまま利用するので，適応度の落ち込みを回避できている．探索精度の面では cK-GP が最も性能が高い．しかし，cK-GP では獲得されたルールにクラスタ中心を表す属性がそのまま残り利用されることがある．一方，K-GP と HC-GP は，ルールに現れる属性は本来のデータベースの属性のみである．元の属性のみでルールが表現されルールの可読性が高いという観点では HC-GP が優れているといえる．

5 おわりに

本論文では，データベースからのルール抽出問題に GP を適用する際，探索初期ではデータベースの属性を

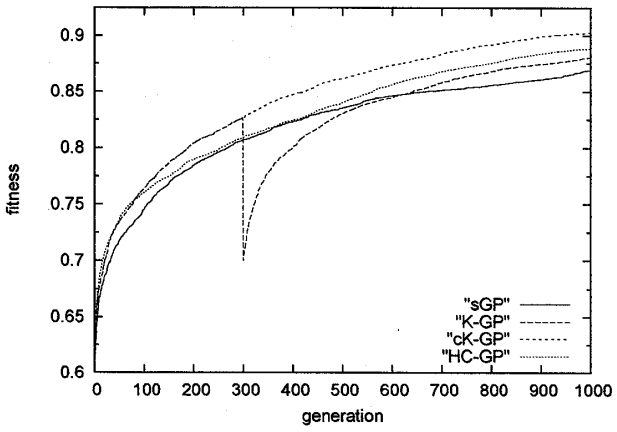

図 1: 最良個体の適応度の推移

表 2: 適応度と標準偏差

GP の種類	最良適応度	標準偏差
sGP	0.868954	0.028182
K-GP	0.880573	0.029356
cK-GP	0.902939	0.029139
HC-GP	0.888653	0.027898

クラスタリングして GP に用いる終端記号数を減らし，その後クラスタ情報をもとに局所的探索を行う手法を提案した．

実験の結果，いずれの提案手法でも標準的 GP より探索効率が改善された．また，分類精度を重視する場合は cK-GP，ルールの表現を重視する場合は HC-GP を用いることで，より効率的に解を探索できることが分かった．

今後は，より大規模な属性を持つ問題に提案手法を適用し，性能を検証する必要がある．また，K-GP と HC-GP では各々に一長一短があったが，これらをもとにルールの精度と可読性の高さを同時に満たす手法の実現の検討を行いたい．

参考文献

[1] John R. Koza, “Genetic Programming: On the Programming of Computers by Means of Natural Selection”, MIT Press (1992).

[2] Alex A. Freitas, “Data Mining and Knowledge Discovery with Evolutionary Algorithms”, Springer (2002).

[3] 蔵内 克哉, 原 章, 市村 匠, 高濱 徹行, “属性間の類似度に基づく突然変異を導入した遺伝的プログラミングによるルール抽出”, 電子情報通信学会総合大会, D-8-18 (2008)

[4] UCI Machine Learning Repository, “http://archive.ics.uci.edu/ml”.