決定木構築における進化の早期停止

川連 太陽[†] 趙 強福[‡]
会津大学[†‡]

1. はじめに

機械学習において、大まかにわけると記号的アプローチと、非記号的アプローチの２つに分けることができる。一般に、記号的アプローチは理解しやすい結果を得ることができる。何かをさせたいというだけなら普通は非記号的アプローチで十分である。しかし、何故そうなるかという理由を得たい場合が多い。そのような場合、記号的アプローチのほうが有利である。

決定木は全ての結論に対し理由を示しているので、通常理解しやすいと考えられている。しかし、データ集合が大きいとき決定木はとても大きくなり[7]、理解しにくくなる。わかりやすさを考慮するなら、決定木を小さくする必要がある[7][8]。一般的に、最小の決定木を見つけることは NP 完全問題である[3]。最適解に近い解を見つけるために、我々は訓練データを進化させ、その結果から小さくて理解しやすい決定木を生成する方法を提案した[1]。この方法の問題点として、計算量は多いことである。

本論文では計算量を減らすために進化の早期停止について考察する。進化過程を客観的視点から観測するためにバリデーション集合を導入した。これにより過学習を感知し、汎化能力を落とさず進化を早期停止する。我々はこの方法で計算量を約７割減らすことに成功した。また過学習を回避することにより汎化能力を高めることができた例もあった。

2. 訓練データの進化に基づく理解しやすい決定木の設計

以前我々が提案した、訓練データを進化させる方法を簡単に説明する。小さい木を得るために、訓練データの一部を用いた設計方法を提案した。決定木の性能が落ちないように、データをどのように選択するかが問題である。この問題を解決するために、GA を使用した。ここで個体となるのは、選択したいデータの部分集合である。個体の遺伝子型は、選択されたデータの位置の列である。この列から、データを訓練集合 D から取り出し、*C 4.5[2]*により決定木を生成する。生成された決定木の D 全体に対する認識率を求め、それを個体の適合度(**fitness**)として使う。その適合度を基に次世代の個体群を生成する。この繰り返しで重要な訓練データを進化によって選択する。

3. バリデーション集合を利用した進化の早期停止

前述のアルゴリズムは、重要な訓練データを選択し、それから小さくて理解しやすい決定木を設計することができる。問題点としては計算量が大きいことである。計算量を減らすために、進化を早く止める方法を考える。そのために、バリデーション集合**V**を導入する。

考え方としては、個体の適合度を D で評価すると共に、この個体から設計した決定木の汎化能力を V で評価する。汎化能力は進化に直接関係ないが、それを用いて進化の度合いを判断できる。進化の初期段階では、汎化能力は適合度と共に上昇するが、過学習となる時点で逆に下がっていく。この現象を利用して、過学習を感知し、進化を早めに止めることができる。図１はこの方法のブロック図である。

4．過学習の感知方法

V の適合度を使用して過学習の感知方法として二つの方法を提案する。また、二つの方法で共通に、ある世代数だけ進化が停止すると、進化が停滞しているとみなし、進化を停止する。

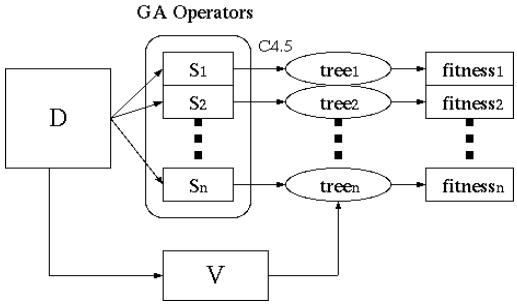

図 1 Illustration of the proposed algorithm

Method 1: Sliding Window を用いる方法

ある一定の範囲を見渡す方法を Sliding Window と呼ぶ。これは、現在の世代から、ある一定の範囲 *s* 分だけ前の世代を見渡せる。また、この範囲に対して *s/2* 点の移動平均をかける。現在の世代を *G* とすると、この時点での移動平均は *G-s/2* 世代における移動平均となる。このように移動平均をかけることで V に対する認識率の微小変動（ノイズ）を除去できる。また *s* を大きくとると、ノイズをより除去できるが、範囲を大きくとりすぎると重要な情報まで失われる恐れがある。

ノイズを除去した後、そのグラフに対して極小値を感知する。極小値が今までの最小値を更新しつづけるなら進化が活発に行われているとみなし進化を続ける。最小値を更新しなくなると、進化が停滞しているか過学習とみなし進化を停止する。

Method 2: 変化点を用いる方法

変化点を用いる方法としては、V に対する認識率が変化した時点を保持しておき、それらを結ぶことにより、その傾向を感知する。これは変化点が現れるたびに更新され、逐次検査される。このグラフの変化量が減少したとき、進化の停滞もしくは過学習とみなし、進化を停止する。

また、グラフの生成方法は変化した時点の前後でより認識率の低い値の時点と、値の同じ範囲の中心の時点を用いて、３次の多項式補間によりグラフを生成した。

5. 実験

5.1 実験データ

実験には以下のデータベースを使用した。

- The credit approval database (crx for short): 2 classes, 15 attributes, 690 examples, 10 fold
- The car evaluation database (car): 4 classes, 6 attributes, 1728 examples, 20 fold
- The tic-tac-toe endgame database (tic-tac-toe): 2 classes, 9 attributes, 958 examples, 10 fold

- The dermatology database (dermatology): 6 classes, 34 attributes, 366 examples, 10 fold

5.2 実験パラメータ

- Maximum number of generation: 1000
- Population size: 100
- Genotype size: 50
- Crossover type: 2-point crossover
- Crossover rate: 70 %
- Mutation rate: 5 %
- Selection type: Truncation selection
- Truncation rate: 30 %
- (Method 1) Sliding Window Size: 10
- (Method 2) Number of Change Point: 20

Method 1 には *Sliding Window Size* として、 Sliding Window が見渡す幅を、Method 2 には *Number of Change Point* として保持しておく変化点の数をそれぞれ設定した。

5.3 実験結果

実験結果を以下の表に示す。数値は１０回か２０回の平均である。表１は訓練集合に対するエラー率、表２はテスト集合に対するエラー率、表３は終了した世代数、表４、表５にそれぞれ訓練集合、テスト集合に対する t-test の結果をしめした。また t-test は 95%で検定を行った。

表 1 Error rate for training set

Database	V not used	Method 1	Method 2
crx	0.105636	0.118131	0.121739
car	0.165997	0.183609	0.190428
Tic-Tac-Toe	0.204710	0.233296	0.229252
dermatology	0.025500	0.038240	0.052957

表 2 Error rate for test set

Database	V not used	Method 1	Method 2
crx	0.153623	0.131884	0.147827
car	0.184650	0.183487	0.198670
Tic-Tac-Toe	0.246360	0.257818	0.261941
dermatology	0.054655	0.049249	0.068544

表 3 Number of generations used

Database	V not used	Method 1	Method 2
crx	1000.0	181.3	313.8
car	1000.0	199.5	244.2
Tic-Tac-Toe	1000.0	197.7	240.6
dermatology	1000.0	148.7	191.4

6. まとめ

本論文において、進化の早期停止を行う方法について考察し、バリデーション集合を用いた方法を提案した。実験結果から両 Method 共に、確実に早い段階で停止しているのがわかる。計算量はおよそ七割以上減らすことができた。特に Method 1 は Method 2 に比べ早く停止している。また、訓練集合に対するエラーは、Method 1, Method 2 共に早期停止しない場合に比べ悪くなっているが、テスト集合に対するパフォーマンスの低下は見られなかった。特に Method 1 については、パフォーマンスが良くなる例も見られた。この場合、進化を早期停止することにより過学習が回避できたのだと思われる。ただ、Method 2 は Method 1 に比べ、計算量、汎化能力共に低いことがわかる。

表 4 Results of t-test for training set

Database	V not used vs Method 1	V not used vs Method 2
crx	0.0122989± 0.00631071	0.0161032± 0.010256
car	0.0176124± 0.00618014	0.0244309± 0.00709009
Tic-Tac-Toe	0.0285862± 0.00998896	0.0245391± 0.0136521
dermatology	0.01274± 0.00565787	0.0274575± 0.0143078

表 5 Results of t-test for test set

Database	V not used vs Method 1	V not used vs Method 2
crx	-0.021739± 0.021138	-0.0057969± 0.0205536
car	-0.00116275± 0.0161574	0.0139199± 0.0183394
Tic-Tac-Toe	0.011458± 0.053224	0.015581± 0.023861
dermatology	-0.0054053± 0.0286882	0.0138889± 0.0307512

文　献

[1] T. Endou and Q. F. Zhao, “Generation of comprehensible decision trees through evolution of training data,” Proc. IEEE Congress on Evolutionary Computation (CEC’2002), pp. 1221-1225, 2002

[2] J.Ross.Quinlan, “C4.5: programs for machine learning,” Morgan Kaufmann Publishers, 1993.

[3] L.Hyafli, ``Construction optimal binary decision trees is NP-complete,"Information Processing Letters 5(1): 15-17, 1976.

[4] M.Shirasaka, Q.F.Zhao, O.Hamami, K.Kuroda and K.Saito, “Automatic Design of Binary Decision Trees Based on Genetic Programming,” Proc. The Second Asia-Pacific Conference on Simulated Evolution and Learning, 1998.

[5] Q.F.Zhao and M.Shirasaka, “A Study on Evolutionary Design of Binary Decision Trees,” Proc. Congress on Evolutionary Computation: 1988-1993, 1999.

[6] T. Tanigawa and Q. F. Zhao, “A study on efficient generation of decision trees using genetic programming,” Proc. Genetic and Evolutionary Computation Conference (GECCO'2000), Las Vegas, 1047-1052, 2000.

[7] T. Oates and D. Jensen, “The Effects of Training Set Size on Decision Tree Complexity,” Proc. 14th International Conference on Machine Learning, Morgan Kaufmann, 254-262, 1997.

[8] F. J. Provost, D. Jensen and T. Oates, “Efficient Progressive Sampling,” *Knowledge Discovery and Data Mining*, 23-32, 1999.

「Early Stoppage for Evolutionary Design of Decision Trees」

†「Takaharu Kawatsure・University of Aizu」

‡「Qiangfu Zhao・University of Aizu」