

進化的計算手法を用いた特徴選択アルゴリズムの提案

Reduction of feature dimensions for feature selection using Evolutionary Computation and penalty function

川村 篤志^{*1}

Atsushi Kawamura

チャクラボルティ バサビ^{*1}

Basabi Chakraborty

^{*1} 岩手県立大学 ソフトウェア情報学部

Iwate Prefecture University Faculty of Software and Information Science

Feature selection is an important preprocessing for datamining and pattern recognition problem. This process selects meaningful features, removes unnecessary features, and facilitates classification and extraction for data. However, as the number of feature increases, the combination of features exponentially increases. Therefore, making it difficult to select a combination of good features. Evolutionary computational(EC) algorithms are well known for solving the combinatorial optimization problem, and high accuracy can be calculated.

In this work, we propose a new fitness function which is composed of accuracy with another penalty term which penalizes for large number of features. The proposed new fitness function is used with EC algorithm for feature selection. Simulation experiments are done using benchmark data set and the efficiency of the proposed fitness function for feature reduction is verified.

1. はじめに

近年、インターネットの普及により、個人からのデータ送出手が容易になっている。SNS やソーシャルメディアをはじめとして、IoT やクラウドなども増加しており、それらを介すことにより様々な種類の多量なデータの獲得が可能になってきている。加えて、解析技術や計算機性能の向上によりビッグデータと呼ばれる大規模なデータの解析、収集が容易になっている。このことにより、ビッグデータを解析することで知見を獲得しようという動きが活発化している。また、高度な知能情報処理を実現する技術として、機械学習の研究領域が近年ますます重要性を帯びている。現代において我々はデータ解析を行う際の認識、分類精度や計算処理速度を向上させることで、効率的にデータを処理すること、データ解析の性能を高めていくことが重要となってきている。

回収したデータを用いて解析を行う場合に、データをそのまま使用して解析を行ってしまうと処理コストがとて高く、得られる精度も悪くなってしまう。そこで、解析を行う前に、特徴選択という前処理が必要となる。特徴選択を行うことにより解析時の計算コストの低下、精度の向上が見込める。しかし、前処理で得られた特徴部分集合によって解析時の精度が変化する。また、多くの特徴を持つデータから最適な特徴の組み合わせを選択することは非常に困難である。

以上の理由から効率よく特徴の組み合わせを選択できる手法を考える必要がある。

本研究では、進化的計算と独自に提案する評価関数を用いて特徴選択を行い、全ての特徴を使用して算出される精度を上回ること、機械学習での学習を高速化すること、従来の特徴選択で獲得できる精度を維持したまま、特徴部分集合内で解析に用いられる特徴の数(次元数)を削減することを目的とする。

この目的を達成することにより、適切な特徴部分集合の獲得ができ、解析時に使用される次元数が減少し処理コストの低下が見込め、次元の呪いが緩和できる。

2. 特徴選択

特徴選択とは、生データから意味のある特徴部分集合を選択する手法である。特徴選択を行う理由としては、精度や汎化

連絡先: 川村 篤志, 岩手県立大学, ソフトウェア情報学部,
パターン認識と機械学習研究室

性能の向上、次元の呪いの緩和、処理の高速化、2,3 次元まで次元数を削減できた際の生データでの可視化、データの可読性の向上などが挙げられる。

一般的な解析を行う過程において、生データから特徴を選択する際に 01 配列集合を用いてマスキング処理を行い、0 の際はその特徴を使用せず 1 の部分特徴のみを選択、使用し評価を行う。ここで用いられる 01 配列からなる特徴の組み合わせを、本稿では特徴部分集合と呼称する。

特徴選択手法は主に 2 つの手順によって行われる。

1 つ目は特徴部分集合をどのように選択するかという探索手法の設定である。手法例としては、総当たり法や Sequential Forward Selection(SFS)、進化的計算などが挙げられる。解析するデータの特徴数が少ない場合には、特徴の組み合わせを全て探索する総当たり法などは有用な手法ではあるが、特徴数が増加すると特徴の組み合わせは指数的に増加するために適切な特徴部分集合の選択が困難になってしまう。

2 つ目は、選択された特徴部分集合をどのように評価するかという評価指標の設定である。評価指標については、主にラッパ法、フィルター法の 2 種に大別される。ラッパ法は、選択された特徴部分集合ごとに分類器など実際の評価に使用する手法をそのまま評価値とする方法である。フィルター法は、選択された特徴部分集合内の特徴データの性質を用いて、分類器など実際の評価手法を使用しない単純な評価指標で評価値を得る方法である。主に、情報ゲインや相互情報量などが評価指標として用いられる。本研究ではラッパ法を使用する。ラッパ法は、実際に分類器を用いた評価を用いるのでどのデータでも宛がうことができるため汎用性が高く、獲得できる精度も安定して高い。しかし、特徴部分集合ごとに逐次分類器での評価を行うため計算コストが高く、過剰適合を起こす可能性がある。

本研究で用いる特徴選択手法は、1 つ目の探索手法においては進化的計算を用いる。2 つ目の評価指標においては LinearSVM(Linear Support Vector Machine)で分類を行った精度を評価値として用いる。

3. 進化的計算

進化的計算は、組み合わせ最適化問題などの解決に用いられるメタヒューリスティックなアルゴリズムである。Genetic Algorithm(GA) [Goldberg, 89]が属する進化的アルゴリズムと Particle Swarm Optimization(PSO)が属する群知能の 2 種に大別される。本研究では特徴選択の探索手法として進化的計算

手法を二値化した以下 BGA, BPSO[Kennedy 97], Binary Cuckoo Search(BCS) [Rodrigues 13], Binary Firefly Algorithm(BFFA) [Broderick 14], Binary Bat Algorithm(BBA) [Nakamura 14], Binary Gravitational Algorithm(BGSA) [Esmat 10], Binary Dragonfly Algorithm(BDFA) [Seyedali 15]の7つを用いる。

4. 提案手法

従来では、分類精度での評価をそのまま評価指標として用いることが多い。しかし、その場合では獲得された特徴部分集合がそれぞれ同等な評価値であった場合でも次元数が大きく変動することが多い。

本研究で提案する評価指標は、進化計算時の特徴部分集合の評価値に次元数の割合でペナルティを与え、同等な評価値であっても少ない次元数を獲得するものである。また、選択された特徴部分集合が良い精度であっても、獲得された次元数が多い場合には処理コストの大幅な低下が見込めないの、高い次元数の獲得を抑止する目的もある。

従来の評価指標として、テストデータの正答数 A をテストデータ数の合計 Z で割ったものを正答率(1)式とした。本研究では、(1)式の評価に付随するペナルティ変数として α ($0 < \alpha$)を定義した。そして、総次元数 N を選択した次元数 D で除算を行った使用次元率にペナルティ α を乗算したものをペナルティ関数として提案する。そして、(1)式からペナルティ関数を減算したものを(2)式とする。

$$S1 = \frac{A}{Z} \quad (1)$$

$$S2 = S1 - \alpha \times \frac{D}{N} \quad (2)$$

5. 実験

本研究で提案した手法について、データセットを用いた従来手法との比較評価実験とペナルティレートの考察を行った

5.1 実験評価

実験での評価を行うベンチマークデータセットとして、UCI Data Set Repository¹から7つのデータセットを使用する。表1には、左からデータ名、次元数、クラス数、データ数を表記する。

評価指標内の精度評価が LinearSVM であることで教師データとテストデータに分割されていないデータセットについては教師データとテストデータを4:6の比率で分割を行った。

ペナルティレートの評価については、従来手法のペナルティ変数 0 から 0.05 ずつ 0.5 まで 11 個のペナルティレートから検証を行う。データセットは、Wine,Cancer,Sonar で検証を行う。

表 1. データセット

	次元数	クラス数	データ数
Wine	13	3	178
Cancer	30	2	569
Sonar	60	2	208
Hill	100	2	train:606 test:606
Gas	128	6	3600
Madelon	500	2	train:2000 test:600
Isolet	617	26	train:6238 test:1559

5.2 実験手法

実験で用いた7つの探索手法のパラメータを下記表2,3に表記する。表2はWine, Cancer, Sonar, Hillでのパラメータであり、群数と最大反復数から探索数が6000となるように設定した。表3はGas, Madelon, Isoletのパラメータ設定であり、表2の最大反復数を300から500、15から25に変更し、探索数を10000となるように設定した。各手法のユニークなパラメータについては、各論文や実験から調整を行った。評価実験での提案手法内のペナルティ変数は0.15とした。

表 2. Wine, Cancer, Sonar, Hillでのパラメータ

	群数	最大反復数	Param
BGA	20	300	二点交叉,ランク法 mu=0.05
BPSO	20	300	w=0.5,c=1.0
BCS	20	300	p=0.25,α=0.1,β=1.5
BGSA	20	300	Ec=1,Rp=1
BFFA	20	15	α=0.25,β=0.2,γ=1
BBA	20	300	loud=0.25,r=0.1
BDFA	20	300	

表 3. Gas, Madelon, Isoletでのパラメータ

	群数	最大反復数	Param
BGA	20	500	二点交叉,ランク法 mu=0.05
BPSO	20	500	w=0.5,c=1.0
BCS	20	500	p=0.25,α=0.1,β=1.5
BGSA	20	500	Ec=1,Rp=1
BFFA	20	25	α=0.25,β=0.2,γ=1
BBA	20	500	loud=0.25,r=0.1
BDFA	20	500	

5.3 実験結果

評価では、各手法の分類精度と次元数の2つを用い、各手法、各データセットにおいて(1)式を従来手法、提案(2)式を提案手法としてそれぞれWine, Cancer, Sonar, Hillのデータセットでは100回、Gas, Madelon, Isoletのデータセットでは50回の評価を行い、その平均値を最終評価とした。結果は以下の表5~11に表記する。

また、比較として全ての特徴を使用した結果を表4に表記する。こちらも、100回の評価を行った。

ペナルティレートの実験は、各ペナルティ変数をそれぞれの探索手法50回ずつの平均値を用いる。図1~6は、Wine,Cancer,Sonarのデータセットから獲得された各ペナルティ変数の精度と次元数のグラフである。

表 4. 全特徴での評価

	Mean	max	Min
Wine	0.78664	0.91589	0.43925
cancer	0.88149	0.92983	0.56140
Sonar	0.75200	0.75200	0.75200
Hill	0.93058	0.94720	0.86469
Gas	0.93787	0.96713	0.78241
Madelon	0.53073	0.58500	0.49830
Isolet	0.94547	0.94548	0.94484

表 5. Wine

従来手法			提案手法	
精度	次元数	探索手法	精度	次元数
0.95075	8.20	BGA	0.93215	3.93
0.96178	8.70	BPSO	0.93794	3.64
0.95963	8.95	BCS	0.93682	4.59
0.94271	5.82	BFFA	0.93271	2.98
0.95841	8.46	BBA	0.93117	4.07

¹ <https://archive.ics.uci.edu/ml/datasets.html>

0.95336	8.27	BGSA	0.92813	4.38
0.95860	8.61	BDFA	0.93421	3.49

表 6. Cancer

従来手法		提案手法		
精度	次元数	探索手法	精度	次元数
0.94556	16.41	BGA	0.94708	7.07
0.96605	13.49	BPSO	0.96058	4.13
0.95708	16.12	BCS	0.93965	9.68
0.95930	8.54	BFFA	0.94988	3.58
0.96222	14.60	BBA	0.94896	6.67
0.95035	13.85	BGSA	0.94263	8.96
0.96330	13.33	BDFA	0.95865	5.35

表 7. Sonar

従来手法		提案手法		
精度	次元数	探索手法	精度	次元数
0.84656	32.48	BGA	0.84072	21.22
0.88000	31.35	BPSO	0.88296	18.40
0.84400	37.83	BCS	0.83456	30.83
0.83512	18.73	BFFA	0.82480	9.53
0.88224	30.24	BBA	0.84472	18.16
0.84136	31.41	BGSA	0.82712	22.68
0.86624	30.77	BDFA	0.86704	20.08

表 8. Hill

従来手法		提案手法		
精度	次元数	探索手法	精度	次元数
0.95921	71.54	BGA	0.95218	46.78
0.97340	59.40	BPSO	0.96848	38.44
0.95858	67.56	BCS	0.95250	57.50
0.94383	81.24	BFFA	0.92240	33.68
0.97644	64.68	BBA	0.96890	39.50
0.96617	68.08	BGSA	0.94571	42.60
0.96894	58.86	BDFA	0.96664	42.30

表 9. Gas

従来手法		提案手法		
精度	次元数	探索手法	精度	次元数
0.97598	88.38	BGA	0.96736	50.44
0.98361	72.36	BPSO	0.98220	34.84
0.97963	85.34	BCS	0.96696	59.34
0.98971	19.98	BFFA	0.98686	12.04
0.98459	80.26	BBA	0.98287	29.98
0.97966	92.62	BGSA	0.96846	47.28
0.98419	64.54	BDFA	0.98311	41.18

表 10. Madelon

従来手法		提案手法		
精度	次元数	探索手法	精度	次元数
0.595100	351.70	BGA	0.579146	106.27
0.628033	260.32	BPSO	0.620203	211.54
0.609796	322.53	BCS	0.604825	296.29
0.592347	161.84	BFFA	0.591579	79.00
0.620590	316.40	BBA	0.612078	123.92
0.608403	330.98	BGSA	0.598063	198.51
0.616215	258.29	BDFA	0.618906	221.75

表 11. Isolet

従来手法		提案手法		
精度	次元数	探索手法	精度	次元数
0.951289	501.96	BGA	0.933456	186.71
0.957382	374.65	BPSO	0.951003	235.42
0.949779	409.85	BCS	0.914130	146.71
0.946440	560.35	BFFA	0.927240	121.93
0.960592	468.69	BBA	0.948983	145.96

0.957004	494.34	BGSA	0.933611	221.70
0.959148	424.78	BDFA	0.952534	205.43

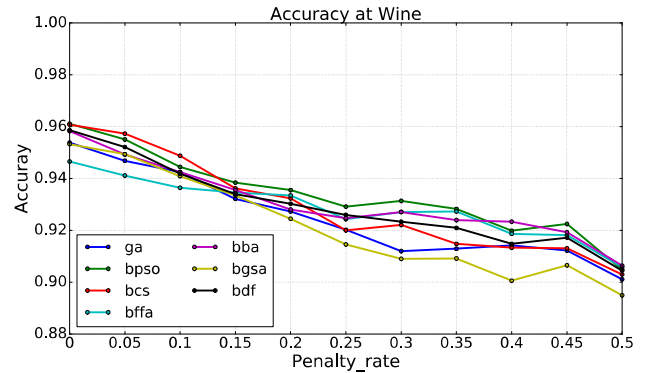


図 1. Wine 評価

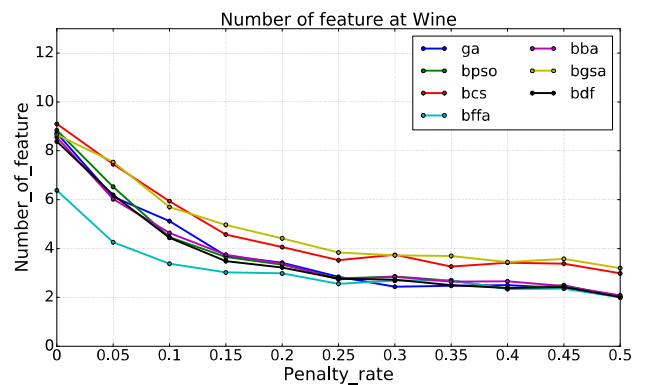


図 2. Wine 次元数

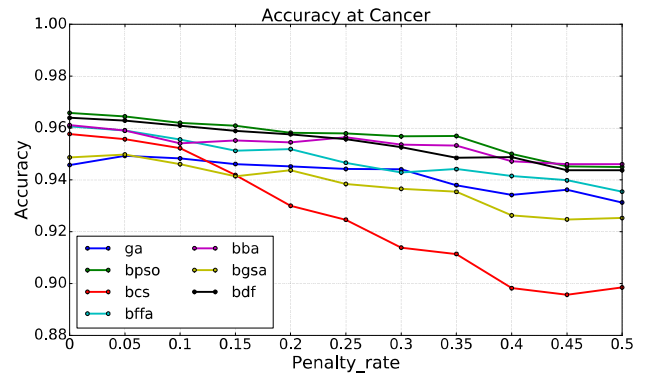


図 3. Cancer 評価

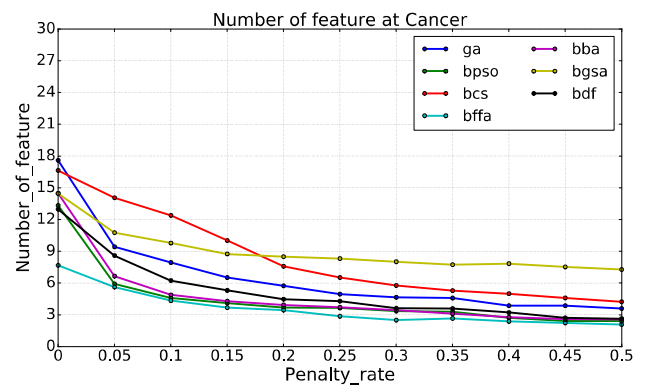


図 4. Cancer 次元数

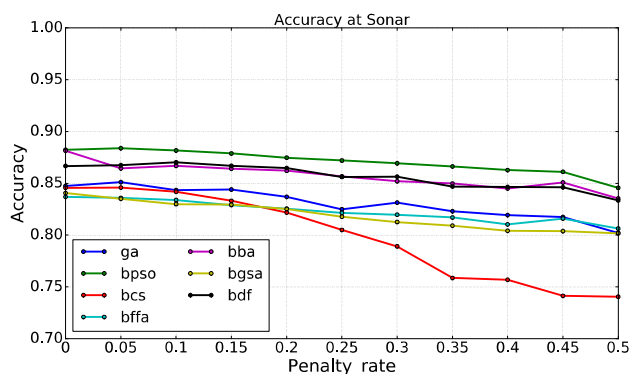


図 5. Sonar 評価

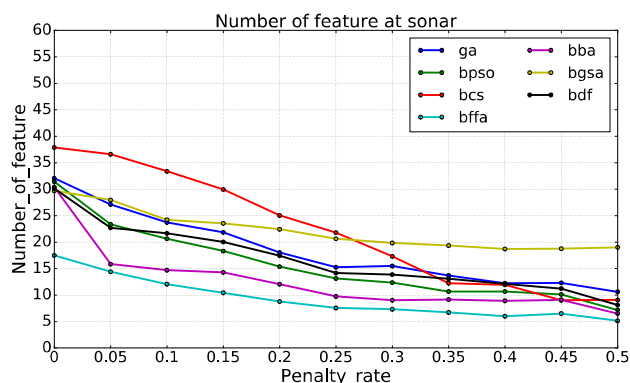


図 6. Sonar 次元数

5.4 考察

実験の結果、どの手法でも精度をあまり減少させることなく次元数が削減できることが確認された。表 5, 6, 9, 10, 11 の結果では、ペナルティ関数を付与することで、殆どの探索手法で元データの 3 分の 1 を下回る次元数を獲得できた。

また、提案手法の評価は Hill や Isolet の一部の精度を除き、全特徴の評価の最大数値よりも上回っていることが判明した。これは、全特徴を用いた精度よりも高精度且つ、次元数を削減できるということを証明し、精度を向上させるとともに削減された次元数だけコンピュータ処理時間も削減を行えるということでもある。

全体の評価を通して BFFA で顕著な結果が獲得できた。精度は一部を除き他の手法に比べて劣っていたが、次元数を大幅に削減できることがどのデータセットの結果からでも見て取れる。

精度においては BBA, BPSO が優秀であった。Cancer や Hill, Madelon の結果からわかる通り、従来手法の BGA での評価よりも提案手法を用いた BPSO や BBA が精度、次元数ともに優れた結果を獲得できることが多く判明した。

総次元数が多い Madelon や Isolet では提案手法の精度が従来手法から 0.03 程度の減少を示したが、次元数が大幅に減少されなかった。このことから次元数が多くなればなるほどペナルティ関数の効果は薄くなっていくことが窺える。これは、ペナルティ関数のペナルティパラメータが 0 から α の範囲で総次元数での分割を行っているため、次元数が増えるほどペナルティの比重が軽くなっていくことが原因と考えられる。また、Wine でも精度が 0.02 ほど減少したが、こちらは総次元数が少ないため分割されたペナルティ間の幅が広いことが原因だと思われる。

図 1~6 のペナルティレートの検証結果から、ペナルティを与え

られた際の評価は 0.01~0.03 の幅で減少しているのに対して、次元数は 0.05 の時点から大きく減少を見せている。しかし、ある程度までペナルティ変数が増加し、次元数が減少するとあまり減少が見られなくなった。次元数の多くは 0.25 以降から大きく減少することではなく、そのデータセットで減少される大まかな限界かと考えられる。ペナルティ変数 0.5 時における Wine, Sonar では、精度がほかに比べ大きく減少を見せている。その際の次元数を見ると、こちらも同じくほかに比べ減少を見せている。このことから、精度と次元数が比例していることが確認できるとともに、一定のペナルティ変数を超えると精度が大幅に減少し、次元数も減少することが確認された。

また、図 1~6 から 1 次元ごとの精度の減少率は 0.0001~0.003 程度であることが算出された。

6. おわりに

6.1 まとめ

本研究では、進化計算の評価時に獲得された特徴部分集合の次元数を用いた独自の評価指標を提案した。結果として、わずかな精度の減少に対し大幅な次元数の削減に成功した。

6.2 今後の展望

今後は、多目的最適化手法や多目的群知能、ハイブリッド手法などの探索手法を追加して評価比較を充実させるとともに、他の群知能手法、データセットを追加して実験を行うことで、さらなる精度と次元数の削減の向上を図っていきたい。

また、次元数以外の相関やクラス距離など、データの性質を考慮した新たなペナルティ関数を思索していきたい。

参考文献

- [Goldberg. 89] Goldberg, David.: Genetic Algorithms in Search, Optimization and Machine Learning. Reading, MA: Addison-Wesley Professional, (1989)
- [Kennedy 97] Kennedy, J. and Eberhart, R. C.: A discrete binary version of the particle swarm algorithm, Conference on Systems, Man, and Cybernetics, Piscataway, NJ: IEEE Service Center, pp. 4104-4109 (1997)]
- [Rodrigues 13] D. Rodrigues, L. A. M. Pereira, N. Souza, C. C. O. Ramos and Xin-She Yang.: BCS: A Binary Cuckoo Search algorithm for feature selection. 2013 IEEE International Symposium on Circuits and Systems. Pp.465-468 (2013)
- [Broderick 14] Broderick Crawford, Ricardo Soto, Miguel Olivares- Suarez and Enrique Norero.: A Binary Coded Firefly Algorithm that Solves the Set Covering Problem. Romanian Journal of Information Science and Technology 17(3): pp 252-264 (2014)
- [Nakamura 14] R. Y. M. Nakamura, L. A. M. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa and X.-S. Yang.: Binary bat algorithm. Neural Computing and Applications, September 2014. Volume 25. Issue 3. pp 663-681 (2014)
- [Esmat 10] Esmat Rashedi - Hossein, Nezamabadi-pour and Saeid Saryazdi.: BGSA: binary gravitational search algorithm. Natural Computing. vol. 9. pp 727- 745 (2010)
- [Seyedali 15] Seyedali Mirjalili.: Dragonfly Algorithm: A New Meta-heuristic Optimization Technique for Solving Single-objective, Discrete, and Multi-objective Problems. Neural Computing and Applications. May 2016. Volume 27. Issue 4. pp 1053-1073 (2015)