

ソフトコンピューティングの
概念に基づく機械学習手法としての
インクドロップスプレッド法の分割表現手法

尾崎 新斗

電気通信大学大学院 情報理工学研究科

博士(工学)の学位申請論文

2014年3月

ソフトコンピューティングの
概念に基づく機械学習手法としての
インクドロップスプレッド法の分割表現手法

博士論文審査委員会

主査	内海	彰	教授
委員	板倉	直明	教授
委員	由良	憲二	教授
委員	高玉	圭樹	教授
委員	本多	中二	名誉教授

著作権所有者

尾崎 新斗

2014 年

Partitioning Methods for the Ink Drop Spread method Based on the Idea of Soft Computing

Yoshito Ozaki

Abstract

This thesis proposes new partitioning methods for the Ink Drop Spread (IDS) method, which is the machine learning technique based on the idea of soft computing.

Soft computing is a robust, low-cost, and tractable computing method in comparison with classical mathematical approaches. The IDS method has been proposed on the basis of the idea of soft computing. This method extracts the input-output relation of the target system as the simple pattern information, and integrates those patterns to model the target system. The method divides a multi-input-single-output (MISO) target system into multiple single-input-single-output (SISO) systems, and models each SISO system by plotting input/output data. In order to generate multiple patterns for each SISO system, the IDS method projects the input-output data of the target system onto the two-dimensional planes. The IDS method must divide the input domains which affect the target system, and make multiple patterns for accurately modeling the target system. Therefore it is important for the IDS method to decide the partitions of each input domain. Then some partitioning methods have been proposed.

First, this thesis describes other soft computing methods and adjustment methods of parameters used by these methods in the second chapter.

The third chapter describes the IDS method and existing methods.

The fourth chapter describes proposed partitioning methods. Existing partitioning methods have the problem that they are likely to generate too many partitions and thus require a long processing time. Therefore we propose a new partitioning methods for the IDS method. First proposed method extracts the information useful for deciding appropriate partitions from the images generated on the two-dimensional plane,

decides partitions using that information, and adjusts the generated partitions. Second proposed method divides input domains by considering the relationship between inputs.

The fifth chapter describes comparing proposed method with the existing partition methods. Furthermore, through comparing proposed methods with other modeling methods such as Feedforward Neural Network and Support Vector Machine, we demonstrate that our IDS method is more effective in approximating complex functions and solving classification problems.

The sixth chapter describes the summary of the result of the fifth chapter and the consideration.

Finally, this thesis summarize usefulness of proposed partitioning methods for the IDS method and discusses future problems and developments.

ソフトコンピューティングの 概念に基づく機械学習手法としての インクドロップスプレッド法の分割表現手法

尾崎 新斗

概要

本論文ではソフトコンピューティングの概念に基づく機械学習手法であるインクドロップスプレッド (IDS) 法の新しい分割表現手法を提案し、その有効性を示す。

ソフトコンピューティングは従来の精度の高い数値的な情報処理手法に対し、あいまいさを許容して過度な精密性の追求を避けることによって、柔軟性、頑健性、扱いやすさを活かすようにした柔軟な情報処理手法である。IDS 法はこのソフトコンピューティングの概念に基づき、対象システムの特性を、単純であいまいさのある複数の 2 次元パターン情報に置き換え、それらのパターン情報をファジィ推論をもちいて統合し、対象システムのモデリングを行う手法である。IDS 法において複雑な対象システムをモデリングするためには、影響の強い入力を分割することで複数のパターン情報を作成する必要がある。このため対象システムの分割表現を決定する分割表現手法が提案されている。

本論文では、まず 2 章で他のソフトコンピューティング手法とその手法でもちいるパラメータの調整方法について述べる。ここでは IDS 法の分割表現のような構造の探索方法について、他手法ではどのように行われているかに着目し、それぞれの手法について述べる。

3 章では IDS 法と既存の分割表現手法について述べる。

4 章では提案する新しい分割表現手法であるヒューリスティックルールをもちいた分割表現手法と逐次分割表現手法について述べる。既存の分割表現手法では機械的に分割表現を作成、探索を行っていた。このため多くの計算時間が必要となり、過学習などの問題があった。そこでパターン情報から情報を抽出し、その情報を反映したヒューリスティックルールに基づく手法を提案する。また既存の分割表現手法では各入力毎に分割を設定していたため、分割どうしが影響してしまい不要な分割が発生していた。そこで逐次分割を行うことで、他の入力における分割を考慮して段階的に分割を決定する手法を提案する。

5 章では提案手法の有効性を検証するために、既存の分割表現手法や、IDS 法以外の手

法であるニューラルネットワークやサポートベクターマシンとの比較を行う．ここでは関数近似や分類問題に各手法を適用し，入力数や問題を変えながら提案手法の有効性について述べる．

6章では5章の結果のまとめと考察について述べる．

最後に7章では，本論文全体を通してのまとめと今後の課題および展望について述べる．

目次

第 1 章	序論	1
1.1	概要	1
1.2	論文の構成	2
第 2 章	ソフトコンピューティング	4
2.1	概要	4
2.2	ファジィ推論	5
2.3	ニューラルネットワーク	6
2.4	GA	7
2.5	融合型	8
第 3 章	インクドロップスプレッド法	11
3.1	概要	11
3.2	アルゴリズム	12
3.3	対象システムの分割表現手法	17
3.3.1	等分割表現手法	17
3.3.2	GA をもちいた分割表現手法	19
3.4	IDS 法の特徴	20
3.4.1	IDS 法の利点	20
3.4.2	IDS 法の問題点	21
第 4 章	提案する IDS 法の分割表現手法	23
4.1	概要	23
4.2	ヒューリスティックルールをもちいた分割表現手法	23
4.2.1	分割アルゴリズム	24
4.2.1.1	インクの濃淡情報の取得	26
4.2.1.2	区間情報の作成	27

4.2.1.3	ナロウパスの変化量	27
4.2.1.4	分割 1	28
4.2.1.5	分割 2	28
4.2.1.6	分割 3	28
4.2.2	調整アルゴリズム	29
4.2.2.1	調整 1	29
4.2.2.2	調整 2	30
4.3	逐次分割表現手法	30
4.4	提案手法におけるメンバーシップ関数	32
第 5 章	手法の比較	34
5.1	2 入力関数近似	34
5.1.1	各分割表現手法をもちいた IDS 法と FNN の比較	34
5.1.2	メンバーシップ関数の違いによる比較	42
5.2	3 入力関数近似	44
5.3	2 値分類問題	50
5.3.1	2 重らせん識別問題	50
5.3.2	Pima Indians Diabetes データセット	53
5.4	多値分類問題	56
5.4.1	出力算出方法の違いによる識別率の変化	58
5.4.2	IDS 法における分割表現手法の比較	59
5.4.3	IDS 法と SVM の比較	61
第 6 章	総合考察	67
第 7 章	結論	71
7.1	本論文のまとめ	71
7.2	課題と今後の展望	72
	謝辞	75
	参考文献	77
付録 A	各評価実験における結果の補足	79
A.1	2 入力関数近似	79
A.2	3 入力関数近似	84
A.3	2 値分類問題	87

A.4 多值分類問題	89
関連論文	99

目次

2.1	FNN	7
2.2	ニューロ・ファジィでのメンバーシップ関数	8
2.3	Genetic fuzzy system における染色体	9
3.1	インクドロップ	12
3.2	IDS 法の構造	12
3.3	2 入力 1 出力システムのデータ	14
3.4	$x_1 - y$ 平面に射影されたデータ	14
3.5	$x_1 - y$ 平面のパターン情報と抽出される特徴	15
3.6	作成されるパターン情報	16
3.7	等分割表現手法において探索する分割数の組み合わせ	18
3.8	等分割表現手法におけるメンバーシップ関数	19
3.9	メンバーシップ関数と染色体	20
4.1	インクの濃淡から分割	25
4.2	分割アルゴリズムの構造	26
4.3	既存の分割方法	31
4.4	$x_1 - y$ 平面を構成する入力空間の分割例	32
4.5	提案手法におけるメンバーシップ関数	33
5.1	Hwang の 5 関数	36
5.2	g_1 の近似精度	37
5.3	g_2 の近似精度	37
5.4	g_3 の近似精度	38
5.5	g_4 の近似精度	38
5.6	g_5 の近似精度	38
5.7	Hwang の 5 関数における学習時間	39

5.8	Hwang の 5 関数における IDS 法の最終的なパターン情報数	39
5.9	Hwang の 5 関数における IDS 法の最大パターン情報数	40
5.10	Hwang の 5 関数における逐次分割手法による IDS 法の出力例	40
5.11	Hwang の 5 関数における FNN の出力例	41
5.12	三角型メンバーシップ関数	43
5.13	台形型メンバーシップ関数	43
5.14	ガウシアン型メンバーシップ関数	43
5.15	関数 (5.2) の近似精度	46
5.16	関数 (5.3) の近似精度	46
5.17	関数 (5.2) における学習時間	46
5.18	関数 (5.3) における学習時間	47
5.19	関数 (5.2) における IDS 法の最終的なパターン情報数	47
5.20	関数 (5.3) における IDS 法の最終的なパターン情報数	47
5.21	関数 (5.2) における IDS 法の最大パターン情報数	48
5.22	関数 (5.3) における IDS 法の最大パターン情報数	48
5.23	等分割表現手法の分割例	48
5.24	GA をもちいた分割表現手法の分割例	49
5.25	ヒューリスティックルールをもちいた分割表現手法の分割例	49
5.26	逐次分割表現手法の分割例	49
5.27	2 重らせん識別問題の学習データ	51
5.28	2 重らせん識別問題のテストデータ	51
5.29	2 重らせん識別問題の識別率	52
5.30	2 重らせん識別問題における学習時間	52
5.31	2 重らせん識別問題における IDS 法の最終的なパターン情報数	52
5.32	2 重らせん識別問題における IDS 法の最大パターン情報数	53
5.33	Pima Indians Diabetes データセットの識別率	54
5.34	Pima Indians Diabetes データセットの学習時間	55
5.35	Pima Indians Diabetes データセットにおける IDS 法の最終的なパターン 情報数	55
5.36	Pima Indians Diabetes データセットにおける IDS 法の最大パターン情 報数	55
5.37	iris データセットの識別率	59
5.38	iris データセットにおける IDS 法の学習時間	60
5.39	iris データセットにおける IDS 法の最終的なパターン情報数	60
5.40	iris データセットにおける IDS 法の最大パターン情報数	60

5.41	各データセットにおける IDS 法と SVM の識別率	62
5.42	各データセットにおける IDS 法と SVM の学習時間	62
5.43	2 入力 3 値分類問題 (300 データ)	64
5.44	関数 (5.6) から作成した分類問題の例	64
A.1	Hwang の 5 関数における等分割表現手法による IDS 法の出力例	81
A.2	Hwang の 5 関数における GA をもちいた分割表現手法による IDS 法の 出力例	82
A.3	Hwang の 5 関数におけるヒューリスティックルールをもちいた分割表現 手法による IDS 法の出力例	83

表目次

5.1	Hwang の 5 関数	37
5.2	メンバーシップ関数毎の近似精度:FVU	44
5.3	Pima Indians Diabetes データセットの内容	54
5.4	データセットの概要	57
5.5	データセットにおけるデータのばらつきと近接距離	58
5.6	IDS 法の出力算出方法の違いによる識別率 (%) の比較	58
5.7	2 入力 3 値分類問題の比較	65
5.8	関数 (5.6) から作成した分類問題の識別率 (%)	65
5.9	関数 (5.6) から作成した分類問題の学習時間 (s)	65
5.10	クラス数の増加による学習時間の増加率	65
5.11	入力数の増加による学習時間の増加率	66
6.1	比較結果のまとめ	69
6.2	多値分類での IDS 法と SVM の比較	70
A.1	Hwang の 5 関数の近似精度:FVU(10^{-2})	79
A.2	Hwang の 5 関数の学習時間 (s)	80
A.3	Hwang の 5 関数における IDS 法の最終的なパターン情報数	80
A.4	Hwang の 5 関数における IDS 法の最大パターン情報数	80
A.5	関数 (5.2) の近似精度:FVU(10^{-2})	84
A.6	関数 (5.3) の近似精度:FVU(10^{-2})	84
A.7	関数 (5.2) の学習時間 (s)	84
A.8	関数 (5.3) の学習時間 (s)	85
A.9	関数 (5.2) における IDS 法の最終的なパターン情報数	85
A.10	関数 (5.3) における IDS 法の最終的なパターン情報数	85
A.11	関数 (5.2) における IDS 法の最大パターン情報数	85
A.12	関数 (5.3) における IDS 法の最大パターン情報数	86

A.13	2重らせん識別問題の識別率 (%)	87
A.14	2重らせん識別問題の学習時間 (s)	87
A.15	2重らせん識別問題における IDS 法の最終的なパターン情報数	87
A.16	2重らせん識別問題における IDS 法の最大パターン情報数	87
A.17	Pima Indians Diabetes データセットの識別率 (%)	87
A.18	Pima Indians Diabetes データセットの学習時間	88
A.19	Pima Indians Diabetes データセットにおける IDS 法の最終的なパターン情報数	88
A.20	Pima Indians Diabetes データセットにおける IDS 法の最大パターン情報数	88
A.21	Car Evaluation データセット	89
A.22	Statlog (Shuttle) データセット	89
A.23	Statlog (Vehicle) データセット	90
A.24	Wine データセット	90
A.25	Wine Quality (red) データセット	90
A.26	Ecoli データセット	91
A.27	Glass Identification データセット	91
A.28	Image Segmentation データセット	91
A.29	Iris データセット	92
A.30	Waveform データセット	92
A.31	Yeast データセット	92
A.32	Iris データセットにおける IDS 法の識別率 (%)	92
A.33	Iris データセットにおける IDS 法の学習時間 (s)	93
A.34	Iris データセットにおける IDS 法の最終的なパターン情報数	93
A.35	Iris データセットにおける IDS 法の最大パターン情報数	94
A.36	各データセットにおける IDS 法と SVM の識別率 (%)	94
A.37	各データセットにおける IDS 法と SVM の学習時間 (s)	94
A.38	2入力3値分類問題 (データ数 300)	95
A.39	2入力3値分類問題 (データ数 3000)	95
A.40	2入力3値分類問題 (データ数 30000)	95
A.41	関数 (5.6) から作成した分類問題 (4入力3クラス)	95
A.42	関数 (5.6) から作成した分類問題 (4入力5クラス)	96
A.43	関数 (5.6) から作成した分類問題 (4入力8クラス)	96
A.44	関数 (5.6) から作成した分類問題 (5入力3クラス)	96
A.45	関数 (5.6) から作成した分類問題 (5入力5クラス)	97

A.46 関数 (5.6) から作成した分類問題 (5 入力 8 クラス) 97

第 1 章

序論

1.1 概要

従来の精度の高い数値的な情報処理手法にたいし，柔軟性，頑健性に優れ，また使い勝手もよい処理手法としてソフトコンピューティングが注目を集めている [11]．ソフトコンピューティングの具体的手法として，ファジィ，ニューロ，遺伝的アルゴリズム (GA) 等が提案されているが，これらは人間の知的機能や生体内のメカニズムを真似てモデルリングおよび処理を行うものであり，それらの優れた特性を強調した手法といえる．

Ink Drop Spread (IDS) 法もこのソフトコンピューティングの概念に基づき，Shouraki and Honda[20] によって提案され，Murakami and Honda[15][16] も 1 つの手法として確立をめざし様々な検討を行っており，ここでは IDS 法がノイズに強く，非線形性の問題によく対応し，ロバスト性が高いことを示している．また櫻井と本多 [19] は複雑な空調吹き出し口風速特性のモデリングや，制御問題への適用を行っている．

IDS 法は対象システムの入出力関係を単純なパターン情報として抽出し，それを組み合わせて対象システムのモデリングを行う．これは例えば，人が掌の上で棒を立たせ，それを制御する場合，人は棒の傾きと手の動きの関係をぼんやりとしたパターンとして脳裡に浮かべ，そのパターンを基に感覚的に手を動かすであろう，といったことを想定してそれを手法としたものである．これにたいしファジィでは人間の動作を，この例の場合「棒が右に傾いたら，手を右に動かす」といった頭に浮かぶ言語表現をモデリングして手法としたものといえる．IDS 法はこのように，対象システムのモデリングに複雑な数式やアルゴリズムを一切使用せず，単純なパターン情報とそれらを巧妙に組み合わせるアナログ的発想によることが大きな特色である．また IDS 法では対象システムの入出力関係がパターン情報として抽出できればモデリングを行うことができ，非線形性の強いシステム，不連続なシステム，微分不可能な特性をもつシステムなどでも対応が可能である．

IDS 法では多入力 1 出力の対象システムを複数の 1 入力 1 出力システムに分け，この

分けた 1 入力 1 出力システム毎に，入出力データを 2 次元の平面に射影し，パターン情報を作成する．このときモデルの精度を上げるために，出力に対する影響の強い入力次元については，その次元をいくつか分割して複数のパターン情報で表現する必要がある．このため IDS 法ではこの入力次元の分割表現が重要となる（ファジィにおけるルール数やメンバーシップ関数の設定の仕方に相当）．

入力次元の分割表現手法には，各入力とも機械的に等分割し，それらの組み合わせから最も精度のよいものを探索し，分割表現を決定する Murakami and Honda の手法 [14] や，GA をもちいて分割位置を決定する Sagha らの手法 [2] が提案されている．しかし Murakami and Honda の手法では分割位置を調整をしないために余分な分割が発生してしまい，Sagha らの手法では大量の分割表現の評価が必要となり計算量が多くなるといった問題がある．またこれらの手法は分割数や分割位置を各入力毎に決定するため，他の入力同士が影響し無駄な空間が作成されてしまい，精度の低下や必要な計算資源の増加を招く．

そこで本論文では，パターン情報の性質に基づいて得たヒューリスティックなルールにより，入力次元の分割位置と分割数を同時かつ高速に決定できる手法と，対象システムの分割を段階的に行うことで他の入力の分割を考慮して新しい分割を行う手法を提案する．ヒューリスティックルールをもちいた手法では，「パターン情報がこのような状態であるならここで分割すべきだ」といったヒューリスティックな観点から分割表現を決定する．すなわちまず未分割のままのパターン情報から分割表現を決め，さらにその分割表現から作成されたパターン情報からより適切な分割表現となるように調整を行う．この手法は，分割位置と分割数を同時に決定でき，分割数を減らし，かつモデルの精度を保ちながら高速なモデリングが可能である．また逐次分割を行う手法では，他の入力における分割を考慮して段階的に分割を決定することで，既存手法において作成されていた無駄な分割を減少させ，精度の低下を防ぎ，必要な計算資源の削減が可能である．

1.2 論文の構成

ここで，本章以降の論文の構成を各章毎に示す．

まず 2 章ではソフトコンピューティングの他の手法について述べる．ここでは IDS 法における分割表現手法のような構造の探索方法について，他手法ではどのように行われているかに着目し，ファジィ推論，ニューラルネットワーク，GA とそれらの融合型について述べる．

3 章では分割数と分割位置の 2 つの観点に着目し，IDS 法について詳しく述べる．次に分割表現手法である Murakami and Honda の手法 [14] や Sagha らの手法 [2] の手法についても述べる．

4 章では提案手法である 2 種の分割表現手法，ヒューリスティックルールをもちいた分割表現手法と逐次分割表現手法について詳しく述べる．

5 章では 4 章で述べた提案手法の有効性を示すために既存の分割表現手法や，IDS 法以外の手法として Feedforward Neural Network(FNN) や Support Vector Machine(SVM) との比較を行う．ここでは関数近似問題や分類問題に各手法を適用し，問題を変えながら提案手法の有効性について述べる．

6 章では 5 章の結果のまとめと考察について述べる．

最後に 7 章では，本論文全体を通してのまとめと今後の課題および展望について述べる．

第 2 章

ソフトコンピューティング

2.1 概要

ソフトコンピューティングは L.A.Zadeh[11] が提唱する概念である。この概念は対象とする問題を精密かつ正確に解析することにより高度で緻密なシステム設計を目指し、正確な情報のもとでの高精度なコンピューティング (ハードコンピューティング) 手法を確立してきた従来の工学的手法とは異なり、人間らしくあいまさを許容し、むしろそれを活かして過度な精密性の追求を避けることによって、取扱いやすさ、頑強性、低コストを達成するための新しい情報処理技法である。これは不正確性や不確実性をどこまで許容できるかを探り、高度な精度性や確実性を要求せず、低精度なコンピューティングによる問題解決を目的とする。具体的にはファジィ理論、ニューラルネットワーク、進化論的計算手法、確立推論などさまざまな分野が含まれ、これらを融合的に利用する。

これらの手法の利用分野として機械学習がある。機械学習とは人間のもつ学習能力を機械で再現することを目的とする分野であり、主に教師あり学習と教師なし学習がある。教師あり学習では限られたデータから、未知のデータを正しく予測することが目的であり、回帰問題や分類問題がある。IDS 法はこの教師あり学習問題に適用される機械学習手法であり、ファジィ推論やニューラルネットワーク、進化論的計算手法も教師あり学習問題に適用される機械学習手法である。また教師なし学習の代表例としてクラスタリングがあり、ファジィクラスタリングのようにファジィ理論をもちいた機械学習手法もある。

本章では IDS 法と同様に、教師あり学習問題に適用されるファジィ推論やニューラルネットワーク、GA、それらの融合型について述べる。

2.2 ファジィ推論

ファジィ推論では複数のファジィルールを組み合わせることによって、1つの結論を表現する。ルールの数に応じて複雑な入出力関係を表すことができる。例えば入力を (x_1, x_2) 、出力を y として

$$y = f(x_1, x_2) \quad (2.1)$$

を表現する場合、入力の集合を $X_i (i = 1, \dots, n)$ 、出力の集合を Y_j とすると式 (2.1) は

$$Y_j = f(X_i) \quad (2.2)$$

と表され、ファジィルールは

$$R^i : \text{If } x_1 \text{ is } A_{k1} \text{ and } x_2 \text{ is } A_{l2} \text{ then } y = B_j \\ (i = 1, \dots, n; j = 1, \dots, m; k = 1, \dots, m_1; l = 1, \dots, m_2) \quad (2.3)$$

と表される。ここで R^i は i 番目のファジィルール、 A_{k1} 、 A_{l2} は集合 X_i の x_1 、 x_2 軸上のファジィ集合を規定するメンバーシップ関数のラベル、 B_j は Y_j を規定するメンバーシップ関数のラベル、 n はルール数、 m 、 m_1 、 m_2 はメンバーシップ関数の数である。メンバーシップ関数は x や y が A や B に属する度合を示す。

ファジィ推論でこれらの複数のファジィルールから推論値 \hat{y} を求めるには、

$$\mu_i = A_{k1}(x_1) \wedge A_{l2}(x_2) \quad (2.4)$$

$$y^* = \bigvee_{i=1}^n \{\mu_i \wedge B_j\} \quad (2.5)$$

と推論式が与えられ、 y^* の重心を求めることにより得られる。式 (2.4) は非ファジィ化と呼ばれ、ここでは min-max 重心法をもちいている。また後件部が定数 w_i となる場合を簡略ファジィ推論とよび、

$$\hat{y} = \frac{\sum_{i=1}^n \mu_i w_i}{\sum_{i=1}^n \mu_i} \quad (2.6)$$

として推論値を求める。

これらの推論値はファジィルールの数やメンバーシップ関数のパラメータ次第で変化するため、これらの調整が必要となる。

2.3 ニューラルネットワーク

情報処理手法としてのニューラルネットワークは生物の神経細胞の構造を模した手法であり、多数のニューロン（神経細胞）から信号を受け取り、他のニューロンへ信号を受け渡すことによって様々な情報処理を行う。このニューロン間の結合強度を調節することでネットワークの特性を変化させる。ここでは一般的なニューラルネットワークであるFNN[5]と、そのネットワークの訓練を行う誤差逆伝搬法について述べる。

FNNはネットワーク内の信号が、入力から出力へと一方向にしか進まない構造をしたニューラルネットワークのことであり、単純パーセプトロン、多層パーセプトロンなどがある。ここで入力 $x_j (j = 1, \dots, p)$ における2層パーセプトロンの出力 \hat{y} を式(2.7)に示す。2層パーセプトロンは図2.1に示すように入力層と中間層、出力層の3層構造であり、ニューロンは中間層にのみ存在する。

$$\hat{y}_i = \sum_{k=1}^m W_{ik} f_k \left(\sum_{j=1}^p w_{kj} x_j - w_{k0} \right) \quad (2.7)$$

ここで w_{k0} は k 番目のニューロンのバイアスであり、 w_{kj} は j 番目の入力と k 番目のニューロンの結合強度、 W_{ik} は i 番目の出力と k 番目のニューロンの結合強度、 f は活性化関数であり、シグモイド関数(式(2.8))が一般的である。

$$S(x) = \frac{1}{1 + e^{-x}} \quad (2.8)$$

次に結合強度 w_{kj} 、 W_{ik} の訓練方法の1つである誤差逆伝搬法について述べる。誤差逆伝搬法では訓練データ(教師データ) (x_j, y_i) をもちいて w_{kj} 、 W_{ik} の調整を行う。評価関数として式(2.9)をもちいる。

$$E = \frac{1}{2} (\hat{y} - y)^2 \quad (2.9)$$

この評価関数 E が最小となるように、最急降下法をもちいて結合強度 w_{kj} 、 W_{ik} の調整を行う。この際、誤差から出力層とニューロンの結合強度を調整し、その調整量から入力層とニューロンの結合強度を調整することから、誤差逆伝搬法と呼ばれる。調整前の結合強度を $w_{kj}(t)$ 、 $W_{ik}(t)$ 、調整後の結合強度を $w_{kj}(t+1)$ 、 $W_{ik}(t+1)$ とすると式(2.10)、(2.11)となる。

$$w_{kj}(t+1) = w_{kj}(t) - \eta \frac{\partial E}{\partial w_{kj}} \quad (2.10)$$

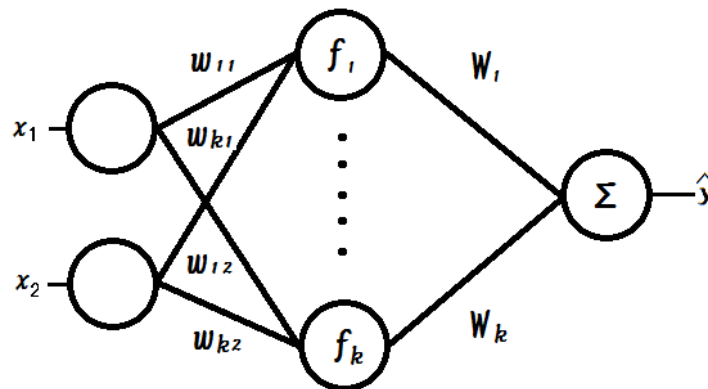


図 2.1 FNN

$$W_{ik}(t+1) = W_{ik}(t) - \eta \frac{\partial E}{\partial W_{kj}} \quad (2.11)$$

ここで η は学習係数である．こうして訓練データから繰り返し結合強度を調整することでネットワークの特性を変化させ学習を行う．

ニューラルネットワークの性能はニューロン間の結合強度に依存するため，適切なニューロン数や層数を探索する手法が提案されている．主に 2 種類の方法が提案されており，pruning algorithm[18] と constructive algorithm[10] がある．pruning algorithm では必要よりも大きいネットワークを作成し，使用されないニューロンを削除していく．constructive algorithm では pruning algorithm とは逆に小さなネットワークから徐々にネットワークを大きくする．

2.4 GA

GA[3] は生物の進化過程，自然淘汰と遺伝をモデル化したもので，最適化問題などにもちいられる．GA では染色体の形で対象問題の解を表現する．この染色体を複数作成し，評価関数をもちいて各染色体を評価する．評価の高い染色体ほど次世代に残る可能性が高くなる．次世代を生成するために遺伝現象をモデル化した遺伝的操作を行い，染色体を変化させる．こうして染色体の淘汰と生成を繰り返し，より対象問題に適した染色体を探索する．

GA の基本的な仕組みを記載する．

1. 初期染色体集団の生成．初期の解としてランダムに染色体を予め設定した母集団数分作成する．

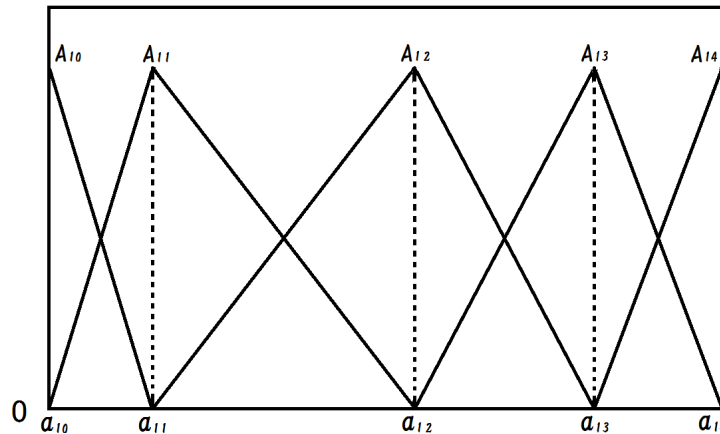


図 2.2 ニューロ・ファジィでのメンバーシップ関数

2. 各染色体の評価．評価関数をもちいて対象問題への適合度を求める．
3. 染色体の選択．適合度の高い染色体ほど次世代の集団に影響を与えるよう，染色体を選択する．選択方法としてはルーレット方式やトーナメント方式，エリート戦略などがある．
4. 次世代の生成．選択された染色体をもちいて次世代を生成する．この操作には交叉と突然変異がある．交叉は2種の染色体の一部を交換するもので，1点のみ交換する，n箇所交換する，乱数をもちいて交換などがある．また突然変異は1つの染色体の一部を変化させるものであり，ランダムに変化させるものや，順番を入れ替えるものがある．
5. 予め設定した世代数になるまで2～4を繰り返し，染色体を変化させる．

2.5 融合型

ニューロ・ファジィ [1][13] や Genetic fuzzy system[7] が提案されている．これらの手法はファジィ推論でもちいられる各種ファジィルールのメンバーシップ関数の形状を決定したり，ファジィルールの数を決定する．

ニューロ・ファジィでは，ファジィルールの後件部を実数値とする簡略ファジィ推論を採用している．ファジィルールは入力を $x_j (j = 1, \dots, m)$ とし，出力を y として

$$R^i : \text{If } x_1 \text{ is } A_{i1} \text{ } \dots \text{ and } x_m \text{ is } A_{im} \text{ then } y = w_i$$

ここで $A_{ij} (i = 1, \dots, n)$ は前件部のメンバーシップ関数， w_i は後件部の実数値を表す．前件部のメンバーシップ関数 A_{ij} はルール毎に設定されている．メンバーシップ関数の

	幅	中央値		幅	中央値
A_{jk}, B_l	0.11	0.23	...	0.22	0.98
	x_1	x_2		x_N	
R_i	2	0	...	3	
	R_1	R_2		R_M	
B_l	1	0	...	3	

図 2.3 Genetic fuzzy system における染色体

形状は，中心値を a_{ij^i} ，幅を $(a_{ij^{i+1}} - a_{ij^{i-1}})$ として表す三角型とする (図 2.2)．このとき各メンバーシップ関数 $A_{ij^i}(x_j)$ は式 (2.12) で表される．

$$A_{ij^i}(x_i) = \begin{cases} \frac{1}{a_{ij^i} - a_{ij^{i-1}}}(x_i - a_{ij^{i-1}}), & a_{ij^{i-1}} \leq x_i \leq a_{ij^i} \\ 1 - \frac{1}{a_{ij^{i+1}} - a_{ij^i}}(x_i - a_{ij^i}), & a_{ij^i} \leq x_i \leq a_{ij^{i+1}} \\ 0, & x_i \leq a_{ij^{i-1}}, a_{ij^{i+1}} \leq x_i \end{cases} \quad (2.12)$$

また簡略ファジィ推論による推論結果 \hat{y} は

$$\mu_i = \prod_{j=1}^m A_{ij^i}(x_i) \quad (2.13)$$

$$\hat{y} = \frac{\sum_{i=1}^n \mu_i w_i}{\sum_{i=1}^n \mu_i} \quad (2.14)$$

となる．ここでは式 (2.4) と違い，非ファジィ化に積和重心法をもちいている．パラメータの調整は 1 個の入出力データが与えられるごとに，デルタルールに基づき調整する．節 2.3 と同様式 (2.9) の最小化問題となる．

$$a_{ij}(t+1) = a_{ij}(t) - K \frac{\partial E}{\partial a_{ij}} \quad (2.15)$$

$$w_i(t+1) = w_i(t) - K \frac{\partial E}{\partial w_i} \quad (2.16)$$

ただし K は学習係数．調整が行われても誤差が変化しなくなった場合，ルールの生成領域の選択，前件部のメンバーシップ関数の生成，後件部の実数値の生成の 3 手順を行い，ファジィルールの自動生成を行う．生成領域には誤差が最も大きい領域が選択され，2 つのメンバーシップ関数の頂点間の midpoint を新たな頂点とし，メンバーシップ関数を新たに生成する．

Genetic fuzzy system では染色体にメンバーシップ関数の形状やファジィルールにもちいるメンバーシップ関数の番号をもちいることで、より問題に適したファジィルールを探索する手法である。ここで染色体としてもちいるメンバーシップ関数は、2.2 節のファジィルールにおける A_{k1} , A_{l2} や B_j のことであり、三角型メンバーシップ値の幅と中心位置を交互に並べることで表現される。またファジィルールの構造も染色体として表し、図 2.3 のようにファジィルール R^i において、入力 x_k の何番目のメンバーシップ関数を利用するか、 B_l は何番目のファジィルールに関係するかといった形で表現される。IDS 法の分割表現手法の 1 つである Sagha らの手法 [2] も、Genetic fuzzy system と同様にメンバーシップ関数の形状を染色体で表現する。ただし IDS 法でもちいるファジィルールでは、後件部がメンバーシップ関数 B_j ではなくパターン情報から抽出した特徴となるため、染色体として表現しない。また前件部のメンバーシップ関数の形状も三角形に限定せず、台形も考慮した表現を使用している。

第3章

インクドロップスプレッド法

3.1 概要

IDS法は対象システムの入出力関係を単純なパターン情報として抽出し、それを組み合わせて対象システムのモデリングを行う。これは例えば、人が掌の上で棒を立たせ、それを制御する場合、人は棒の傾きと手の動きの関係をぼんやりとしたパターンとして脳裡に浮かべ、そのパターンを基に感覚的に手を動かすであろう、といったことを想定してそれを手法としたものである。

IDS法によるモデリングでは、入力数を N とすると、パターン情報を作成するために、各入力 x_i ($i = 1, 2, \dots, N$) と出力 y の関係を $x_i - y$ 平面にデータを射影することから始める。このとき、射影されるデータは入力 x_i 以外の入力 x_j ($j \neq i$) の次元を全て圧縮している。次に射影した各データそれぞれについてデータの存在する位置を中心として、直径 r の円形となるように幅を持ったデータに変換する。この処理はインクドロップと呼ばれており、インクが水面に滲むように中心を濃く、中心から離れるほど薄くプロットする(図 3.1)。

すなわちデータをぼかすことで、データ間の補間やノイズの除去を行い、データ数が少なくてもおおよその傾向をつかめるようにするのである。個々のインクの模様が互いに重なるにつれて、重なり合う部分が濃くなり平面上に濃淡のあるパターンが形成され、一筋の経路や広がりという特性が浮かび上がる。IDS法ではこれらのパターン情報を各入力毎に作成し、それらをファジィ推論をもちいて統合しモデルを構成する。

IDS法では対象システムの入出力関係がパターン情報として描ければモデリングを行うことができ、非線形性の強いシステム、不連続なシステム、微分不可能な特性をもつシステムなどでも対応が可能である。ただしこのようなシステムにおいて多入力だった場合、入力空間を分割し、複数のパターン情報を作成する必要がある。分割位置や分割数といった分割表現がIDS法の性能に大きく影響する。このため分割位置や分割数を決定する分

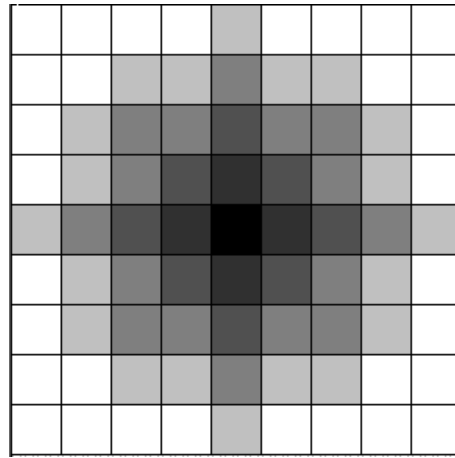


図 3.1 インクドロップ

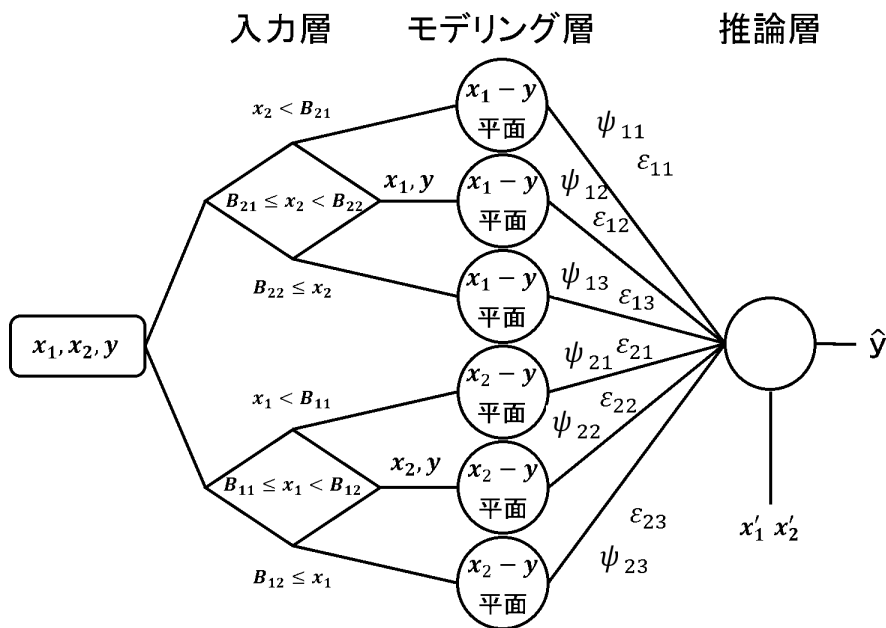


図 3.2 IDS 法の構造

割表現手法が重要となる。

3.2 アルゴリズム

IDS 法では分割表現を決定し、その分割表現をもとに対象システムのモデルを作成する。分割表現は予め与える場合と、作成されたモデルから分割表現手法をもちいて自動的に調節する方法 (3.3 節) がある。

IDS法が対象システムのモデルを作成する場合，入力層，モデリング層，推論層の3層で処理を表すことができる．図3.2は2入力1出力 (x_1, x_2, y) の対象システムにおいて，各入力を (B_{11}, B_{12}) ， (B_{21}, B_{22}) で分割した場合のIDS法の構造である．ここで分割位置 B_{ij} は入力 x_i の j 番目の分割位置である．

入力層では予め与えられた各入力の分割位置 B_{ij} に従い学習データを分類し， $x - y$ 平面に射影する．射影される $x_i - y$ 平面は

$$B_{1k} \leq x_1 < B_{1k+1}, \dots, B_{jl} \leq x_j < B_{jl+1}, \dots, B_{Nm} \leq x_N < B_{Nm+1} \quad \text{となる } x_i - y \text{ 平面 } (i \neq j) \quad (3.1)$$

のように，条件付きで作成される．一般に，作成される平面の総数 L は，各入力の分割数が m_1, \dots, m_N である場合，

$$L = \sum_{i=1}^N l_i \quad (3.2)$$

$$l_i = \prod_{j=1}^N m_j, \quad j \neq i \quad (3.3)$$

となる．ここで l_i は入力 x_i の平面の数である．多入力1出力の学習データ (x_i, y) ($i = 1, 2, \dots, N$)が与えられた場合， N 個の $x_i - y$ 平面に射影されたデータが作成される．ここで図3.2の，2入力1出力 (x_1, x_2, y) 対象システムの各入力を3分割した場合を例として入力層の説明を行う．図3.3に示すデータが対象システムから得られたとする．入力は (x_1, x_2) ，出力は y であり，ここでは $0.00 \leq x_1, x_2 \leq 1.00$ と規格化している． $0.00 \leq x_2 < B_{21}$ となる $x_1 - y$ 平面， $B_{21} \leq x_2 < B_{22}$ となる $x_1 - y$ 平面， $B_{22} \leq x_2 \leq 1.00$ となる $x_1 - y$ 平面が作成され， $x_2 - y$ 平面についても同様のため6種の平面が作成され，学習データが図3.4のように射影される．

モデリング層では各平面に射影された学習データを受け取り，各平面毎にパターン情報を作成する．パターン情報を作成する際には $x - y$ 平面は離散化して表現しており，平面上の各位置毎にインク濃度 D_{jk} ($1 \leq j, k \leq P$)を保持する．ここで (j, k) は平面上の位置， P は平面の解像度であり大きくとればデータをより細かく表現できるようになる．図3.1のインクドロップの具体的な表現方法は，インクドロップのインク濃度を O_{jk} ，射影されたデータの位置を (x, y) とすると，

$$O_{jk} = \begin{cases} [o_{jk}] & o_{jk} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.4)$$

$$o_{jk} = \frac{r+1}{2} - \sqrt{(x-j)^2 + (y-k)^2}$$

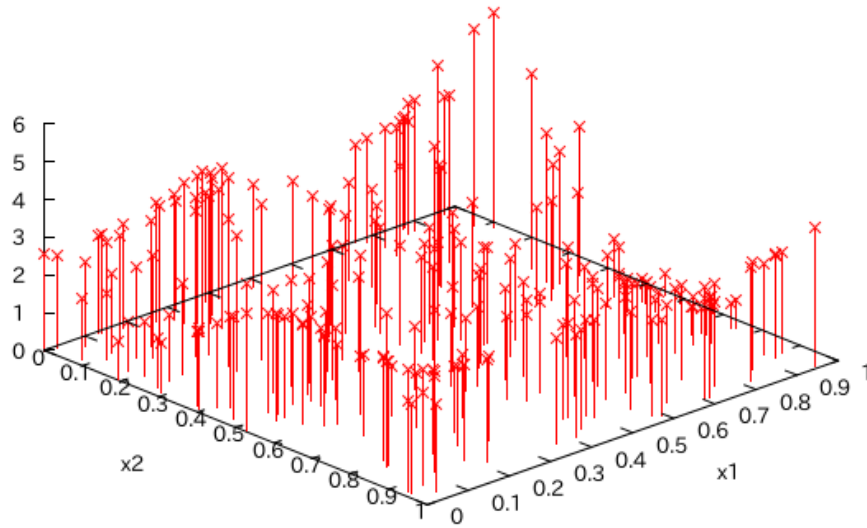
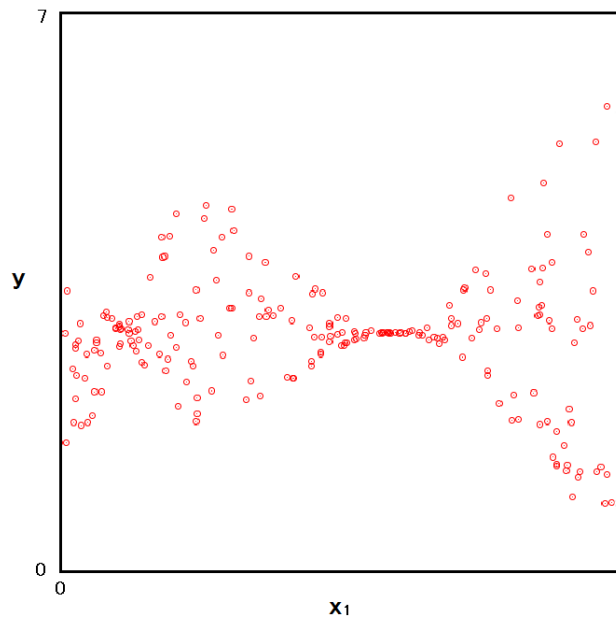


図 3.3 2 入力 1 出力システムのデータ

図 3.4 $x_1 - y$ 平面に射影されたデータ

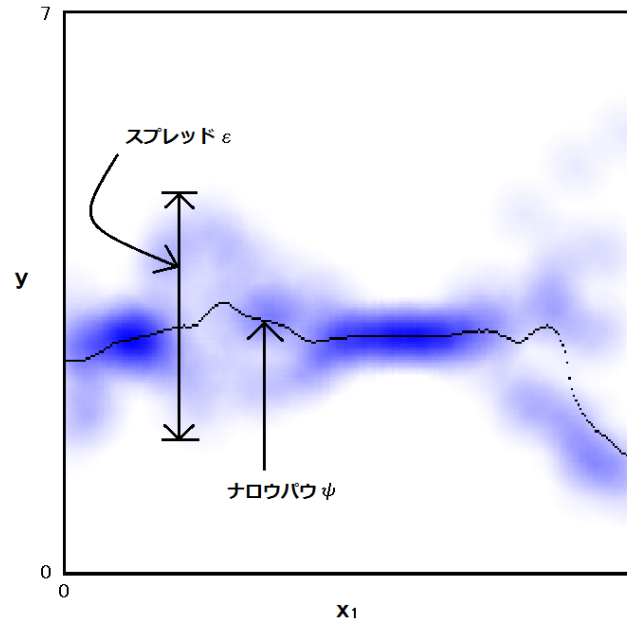


図 3.5 $x_1 - y$ 平面のパターン情報と抽出される特徴

となる．これを射影されたデータ全てについて行い，平面のインク濃度 D_{jk} に足し合わせることで図 3.5 のようなパターン情報を作成する．さらに作成したパターン情報から出力にたいするインクの重心位置であるナロウパス $\psi(x)$ (式 (3.5)) と，出力にたいするインクの広がりであるスプレッド $\epsilon(x)$ (式 (3.6)) を抽出する．

$$\psi(x) = \min\{a \mid \sum_{k=1}^a D_{xk} > \frac{1}{2} \sum_{k=1}^P D_{xk}\} \quad (3.5)$$

$$\epsilon(x) = \max\{k \mid D_{xk} > 0\} - \min\{k \mid D_{xk} > 0\} + 1 \quad (3.6)$$

各平面毎に $\psi(x)$ と $\epsilon(x)$ が抽出されるため，図 3.2 の，2 入力 1 出力 (x_1, x_2, y) 対象システムの各入力を 3 分割した場合を例とすると，図 3.6 に示す 6 種のパターン情報が作成され，

- $x_2 < B_{21}$ の場合の $\psi_{11}(x_1)$ と $\epsilon_{11}(x_1)$
- $B_{21} \leq x_2 < B_{22}$ の場合の $\psi_{12}(x_1)$ と $\epsilon_{12}(x_1)$
- $B_{22} \leq x_2$ の場合の $\psi_{13}(x_1)$ と $\epsilon_{13}(x_1)$
- $x_1 < B_{11}$ の場合の $\psi_{21}(x_2)$ と $\epsilon_{21}(x_2)$
- $B_{11} \leq x_1 < B_{12}$ の場合の $\psi_{22}(x_2)$ と $\epsilon_{22}(x_2)$
- $B_{12} \leq x_1$ の場合の $\psi_{23}(x_2)$ と $\epsilon_{23}(x_2)$

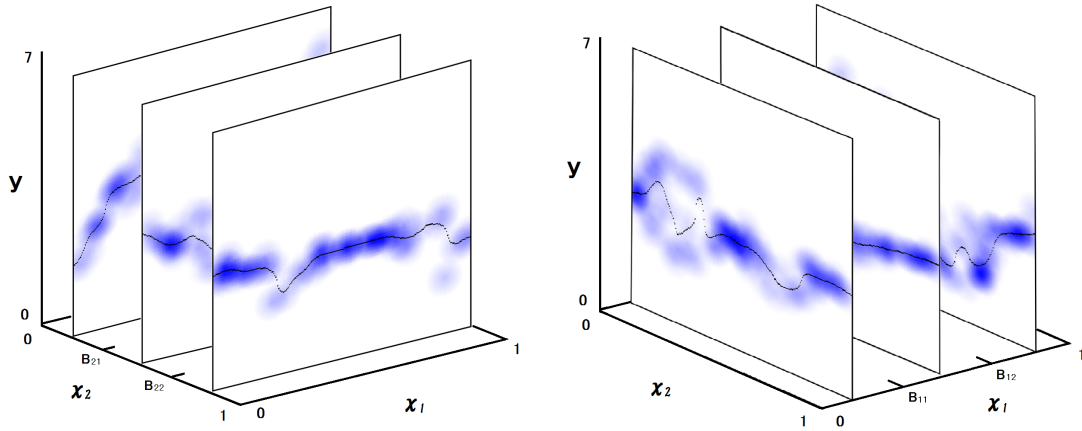


図 3.6 作成されるパターン情報

の6種のパターン情報 $\psi(x)$ とスプレッド $\epsilon(x)$ が抽出される。

推論層ではモデリング層から $\psi(x)$ と $\epsilon(x)$ を受け取り、ファジィ推論をもちいてこれらを統合し、出力をもとめる。ファジィルールは

$$R^g : \text{If } x_1 \text{ is } A_{k1} \text{ and } \dots \text{ } x_j \text{ is } A_{lj} \text{ and } \dots \text{ and } x_N \text{ is } A_{mN} \text{ then } y = \psi_{ih}(x_i) \quad (3.7)$$

$$i \neq j$$

と表される。ここで R^g は g 番目のファジィルール、 A_{k1} 、 A_{lj} 、 A_{mN} は入力 x_j ($j = 1, 2, \dots, N, \neq i$) 軸上のメンバーシップである。図 3.2 の、2 入力 1 出力 (x_1, x_2, y) 対象システムの各入力を 3 分割した場合を例とすると、

$$\begin{aligned} R_{11} &: \text{If } x_2 \text{ is } A_{12} \text{ then } y = \psi_{11}(x_1) \\ R_{12} &: \text{If } x_2 \text{ is } A_{22} \text{ then } y = \psi_{12}(x_1) \\ R_{13} &: \text{If } x_2 \text{ is } A_{32} \text{ then } y = \psi_{13}(x_1) \\ R_{21} &: \text{If } x_1 \text{ is } A_{11} \text{ then } y = \psi_{21}(x_2) \\ R_{22} &: \text{If } x_1 \text{ is } A_{21} \text{ then } y = \psi_{22}(x_2) \\ R_{23} &: \text{If } x_1 \text{ is } A_{31} \text{ then } y = \psi_{23}(x_2) \end{aligned} \quad (3.8)$$

の6種のファジィルールとなる。これらをファジィ推論をもちいて統合し、出力を計算する。テストデータ x'_i ($i = 1, 2, \dots, N$) におけるモデルの出力 \hat{y} を、式 (3.9) に示すナロウパスの重み付き平均からもとめる。

$$\hat{y} = \frac{\sum_{i=1}^N \sum_{j=1}^{l_i} \omega_{ij}(\epsilon(x'_i)) \Gamma_{ij} \psi_{ij}(x'_i)}{\sum_{i=1}^N \sum_{j=1}^{l_i} \omega_{ij}(\epsilon(x'_i)) \Gamma_{ij}} \quad (3.9)$$

ここで重み Γ_{ij} は式 (3.10) よりもとめられる .

$$\Gamma_{ij} = A_{j1}(x_1) \wedge \cdots \wedge A_{jk}(x_k) \wedge \cdots \wedge A_{jN}(x_N) \quad (3.10)$$

$k \neq i$

式 (3.10) は min-max 重心法をもちいた非ファジィ化であり, Γ_{ij} は入力値のルールへの適合度である . 次に式 (3.9) の ω_{ij} は ϵ_{ij} から求められる重みで ,

$$\omega_{ij}(\epsilon_{ij}(x'_i)) = \exp \left(- \left(\frac{(\epsilon_{ij}(x'_i) - 1.1r)/P}{\sigma} \right)^2 \right) \quad (3.11)$$

からもとめる . 式 (3.11) において σ は 0.2 としている . ω_{ij} はスプレッドがインクの直径に近ければ大きく , スプレッドがインクの直径より大きい , または小さければ小さい値となる . これはスプレッドの値がインクの直径より大きい場合 , そのスプレッドに対応するナロウパスでは表現できないデータが存在することを意味し , スプレッドの値がインクの直径より小さい場合 , データが不足していることが予想され , そのスプレッドに対応するナロウパスが正しいのか判別がつかないため , それらのナロウパスの重みを小さくするためにもちいる . また学習データで与えられた入力の最小値よりも小さい値がテストデータとして与えられた場合 , または最大値よりも大きい値が与えられた場合 , 学習データの最小値 , または最大値がテストデータとして与えられたものとして出力を計算する . 以上が IDS 法が学習データより特徴を抽出し , モデリングを行う流れとなる .

分割表現手法はこのモデリング結果を利用し , 新しい分割表現を作成し , 再度モデリングを行うといった処理を繰り返し行い , 適切な分割表現の探索を行う .

3.3 対象システムの分割表現手法

3.3.1 等分割表現手法

これは文献 [14] で提案されている手法であり , 各入力毎に分割数の最大値と最小値を設定し , 入力次元を分割数分に等分割し , モデルの精度が最もよくなる分割数の組み合わせを探索する手法である . 全ての組み合わせを探索したのでは計算量が膨大になるため , 基準をもちいて探索する組み合わせを削減する . この基準には , 各入力の分割数が同一 ($m_1 = m_2 = \cdots = m_N$) の場合に作成されるパターン情報から , 各入力毎にスプレッドから求められる重み ω_{ij} の平均 t_i

$$t_i = \frac{1}{l_i} \sum_{j=1}^{l_i} \frac{1}{P} \sum_{k=1}^P \omega_{ij}(\epsilon(k)) \quad (3.12)$$

総パターン情報数	分割数			
	m_1	m_2	m_3	
3	1	1	1	← $t_1 =$ $t_2 =$ $t_3 =$
●		●		
●		●		
●		●		
12	2	2	2	← $t_1 = 0.39$ $t_2 = 0.41$ $t_3 = 0.37$
○ 15	1	3	3	
15	3	1	3	
○ 15	3	3	1	
16	2	2	3	
○ 16	2	3	2	
○ 16	3	2	2	
○ 21	2	3	3	
21	3	2	3	
○ 21	3	3	2	
27	3	3	3	

図 3.7 等分割表現手法において探索する分割数の組み合わせ

を算出し、最大値と最小値をもちいる。ここで入力 a における重み平均 t_a が最大値であり、入力 $b(a \neq b)$ における t_b が最小値とすると、入力 a, b の分割数 m_a, m_b が $m_a \geq m_b$ となる分割数の組み合わせのみを探索することで、探索する組み合わせを削減している。これは t_a が大きいということは、 $x_a - y$ 平面で作成されるパターン情報のスプレッドが適切であることを意味し、 t_b が小さいということは、 $x_b - y$ 平面で作成されるパターン情報のスプレッドが広いことを意味する。このため $x_a - y$ 平面よりも $x_b - y$ 平面が多く作成されるように分割数を設定することで、 $x_b - y$ 平面で作成されるパターン情報のスプレッドを小さくすることが目的である。

3 入力 1 出力 (x_1, x_2, x_3, y) のシステムにおいて、各入力の最大分割数を 3、最小分割数を 1 とし、 $m_1 = m_2 = m_3 = 2$ の t_i を求め、重み平均 t_i の最大値が入力 2、最小値が入力 3 だったとする。この時 $m_1 = m_2 = m_3 = 2$ の場合のパターン情報の総数 12 よりパターン情報の総数が多く、 $m_1 = m_2 = m_3 = 3$ の場合のパターン情報数の総数 27 よりも少なくなる分割数の組み合わせにおいて、 $m_2 \geq m_3$ となる分割数の組み合わせのみを探索する。つまり図 3.7 において ○ のついた分割数の組み合わせのみ探索する。またパ

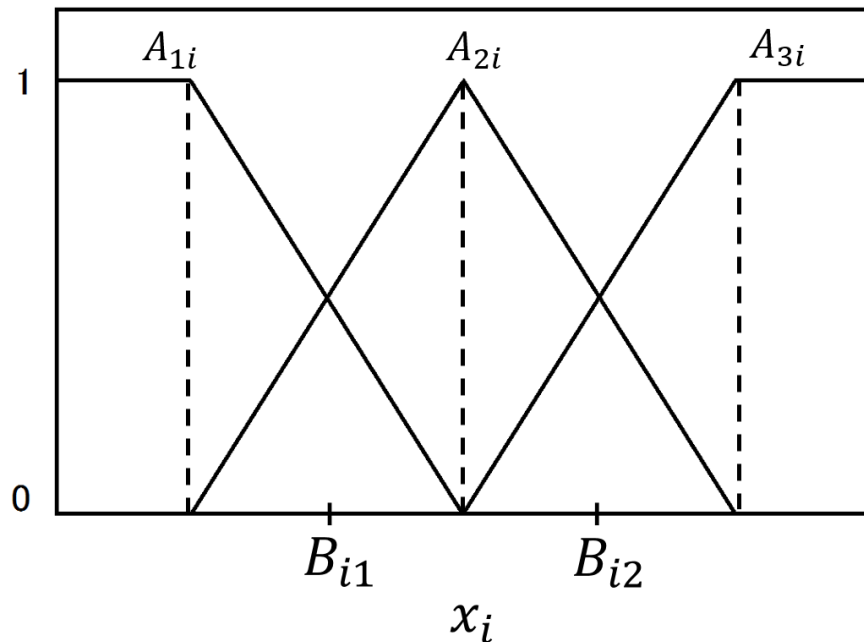


図 3.8 等分割表現手法におけるメンバーシップ関数

ターン情報の総数が 12 未満の場合は, $m_1 = m_2 = m_3 = 1$ の t_i をもちい, 総数が 27 以上の場合は, $m_1 = m_2 = m_3 = 3$ の t_i と使用する t_i が変化する.

モデルの評価には精度をもちい, 与えられた学習データの 7 割をもちいて 3.2 節に示すアルゴリズムからモデルを作成し, 残りの 3 割の学習データをもちいてモデルを評価する. この評価が最も高くなる分割数の組み合わせを探索し, その分割数の組み合わせと全ての学習データをもちいて最終的なモデルを作成する.

等分割表現手法におけるメンバーシップ関数は図 3.8 のように表し, 例えば A_{2i} のメンバーシップ関数は分割位置 B_{i1} と B_{i2} を 2 等分する位置を中心とし, B_{i1} と B_{i2} の差の 2 倍を底辺とする 2 等辺三角形型となる. 等分割表現手法では分割は全て等分割となっているため, 他のメンバーシップ関数も同様に作成される. ただし入力 of 最大値や最小値に関わるメンバーシップ関数 (A_{1i} や A_{3i}) では, 他のメンバーシップ関数と同様の底辺を取れないため, 台形型となる.

3.3.2 GA をもちいた分割表現手法

これは文献 [2] で提案されている手法であり, 図 3.9 に示すように, 各入力のメンバーシップ関数が一定となる部分の長さ ($X(1), X(3), X(5)$) と交差する部分の長さ ($X(2), X(4)$) を染色体とし, GA をもちいてモデルの精度がよくなる染色体を探索する手法である. 図 3.9 では入力 x_i のメンバーシップ関数が 3 つある場合を想定し, A_{1i} は

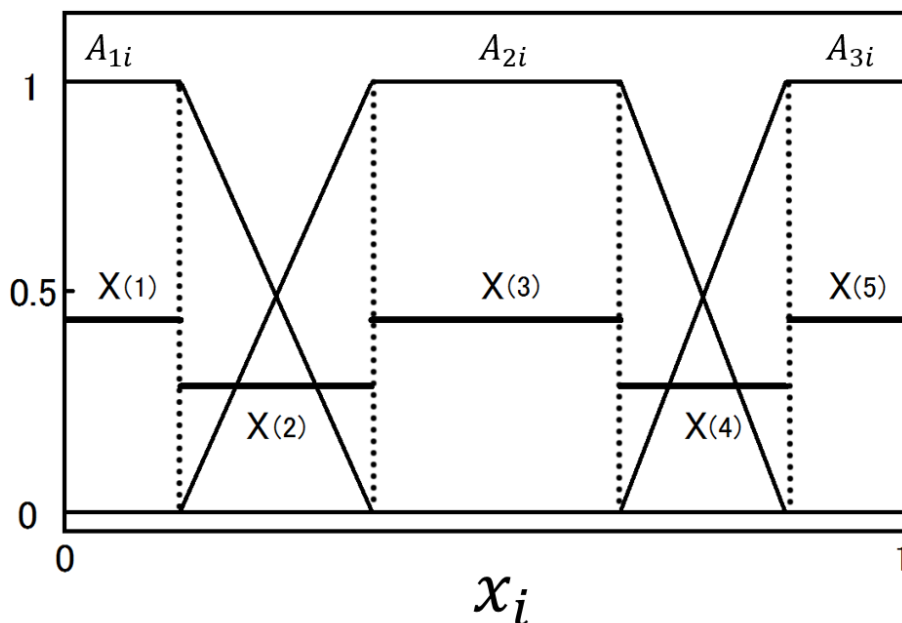


図 3.9 メンバースhip関数と染色体

$X(1)$ の間は 1, $X(2)$ の間で 1 から 0 へと単調減少し, それ以降は 0 となる. また A_{2i} は $X(1)$ の間は 0, $X(2)$ の間で 0 から 1 へと単調増加し, $X(3)$ の間は 1, $X(4)$ の間で 1 から 0 へと単調減少し, それ以降は 0 となる. A_{3i} は $X(3)$ までの間は 0, $X(4)$ の間で 0 から 1 へと単調増加し, それ以降は 1 となる.

染色体の評価には, その染色体で作成されたモデルの精度をもちいる. 3.3.1 節と同様に与えられた学習データの 7 割をもちいて 3.2 節に示すアルゴリズムからモデルを作成し, 残りの 3 割の学習データをもちいてモデルを評価する. 最終的に最も評価の高くなる染色体と全ての学習データをもちいてモデルを作成する.

分割数は染色体の長さで決定されるが, 文献 [2] では染色体長を固定していた. 本論文では分割数の決定も考慮するため, 染色体長を可変とする. また分割位置は交差する部分の長さを等分割する位置としている. 染色体の操作には一点交叉と突然変異, エリート戦略をもちいる. 本論文では母集団中の 1 割をエリートとし, 1 割を突然変異, 残りの 8 割を交叉で作成する.

3.4 IDS 法の特徴

3.4.1 IDS 法の利点

IDS 法の利点として

- 非線形性への適用に優れる
- ノイズに強い
- フォールトトレランスが高い
- 学習結果が視覚的に表現される (わかりやすい)
- 学習時間が短い

が挙げられる。非線形性への適用に優れることや、ノイズへの耐性、フォールトトレランスについては Murakami and Honda[15][16] によって検証されている。

IDS 法では非線形な対象システムを非線形な複数の 1 入力 1 出力パターン情報で表現できる。このため IDS 法は、FNN において結合強度を調整しにくい変化の複雑な問題や、ファジィ推論では極めて多くのファジィルールを必要とする非線形な問題への適用に優れる。

IDS 法ではデータを 1 入力 1 出力の平面上に拡散させてインクをプロットし、出力にたいするインクの重心を特徴として抽出する。このためデータが十分にあることが前提となるが、ノイズデータの影響を小さくすることが可能である。また複数のパターン情報から抽出された特徴から出力を計算することも、ノイズの影響を抑える働きをしている。

IDS 法は複数の特徴から出力を計算するため、一部の特徴が障害により欠落した場合でも、その影響を小さくすることが可能である。

ファジィ推論の学習結果はファジィルールの形で言語的に表現され、FNN の学習結果は結合強度として表現される。FNN は学習結果がブラックボックス化しており、学習結果の解析が困難である。またファジィ推論では言語的に表現されるため学習結果の解析が容易であるが、ファジィ推論では非線形性が強くなると極めて多くのファジィルールを必要とするため、解析が難しくなることがある。IDS 法は学習結果を 2 次元パターン情報で表現するため、結果の解析を容易に行える。

IDS 法では 1 つの学習データにたいして、入力数回のインクドロップを行うことで学習が終了する。FNN では 1 つの学習データにたいして、複数の結合強度を複数回調整する必要があり、インクドロップや調整にかかる計算量コストは違うが、学習時間は IDS 法の方が早くなる。ファジィ推論ではファジィルールを調整する必要があり、2.4 節で述べた手法をもちいる場合、やはり IDS 法の方が計算時間が早くなる。

3.4.2 IDS 法の問題点

IDS 法の問題点として

- 必要な計算資源が多い
- 各種パラメータの設定方法が人手である

が挙げられるある。

IDS 法ではパターン情報を作成し、そのパターン情報から特徴を抽出することで学習を行う。パターン情報を作成するために多くの計算資源が必要となり、他手法よりも必要な計算資源が多くなる。また IDS 法の各入力の分割数が同一 ($m_i = M$) の場合、式 (3.2) のパターン情報の総数 L は

$$L = NM^{N-1} \quad (3.13)$$

となる。ここで N は入力数であり、入力数が増えることでパターン情報の総数が極端に増加することがわかる。このため入力数が増加した場合、必要な計算資源が他手法に比べて圧倒的に多くなる。

IDS 法のパラメータとしてインクの直径や平面解像度、分割表現、メンバーシップ関数が挙げられる。分割表現やメンバーシップ関数については 3.3 節に示すようにある程度自動的に設定する手法があるが、インクの直径や平面解像度に関しては経験的に決定されている。

第 4 章

提案する IDS 法の分割表現手法

4.1 概要

3.4.2 節に挙げた IDS 法の問題点として、入力数が増加した場合に必要な計算資源が極端に増加する問題がある。この問題を解決するため、式 (3.13) における M をできるだけ小さくする必要がある。既存の分割表現手法では、精度のよくなる分割表現を機械的に探索していた。IDS 法が作成するパターン情報には対象システムの特徴が含まれていると考えられるが、既存手法では分割表現を決定する際にこの情報を利用しておらず、多くの計算時間が必要となり、過学習の問題もあった。そこでパターン情報からナロウパスやスプレッド等の特徴を抽出し、それらを反映したヒューリスティックルールに基づく分割表現手法を提案する。この手法をもちいることで、既存手法よりも高速に適切な分割位置を決定することが可能となり、分割数の削減が可能となる。

また別の解決方法として、パターン情報の総数が式 (3.13) とならない方法を提案する。既存の分割表現手法では分割を各入力毎に設定していたため、入力数が増加した場合に分割どうしが影響しあい不要な分割が発生していた。不要な分割が発生することで精度の低下や、多くの計算資源が必要となるといった問題があった。そこで逐次分割を行うことで分割どうしが影響しないように、他の入力における分割を考慮して段階的に分割を増やす手法を提案する。この手法をもちいることで、既存手法において作成されていた無駄な分割を減少させ、精度の低下を防ぎ、作成されるパターン情報の最大数を小さくすることが可能となり、必要な計算資源の削減が可能となる。

4.2 ヒューリスティックルールをもちいた分割表現手法

ここではパターン情報のナロウパス、スプレッド等の特徴に基づくヒューリスティックルールをもちいて、直接分割表現を決定する手法を示す。パターン情報は対象システムの

データから作成されるものであり，そこには対象システムの特徴が含まれていると考えられる．またパターン情報は $x_i - y$ の1入力1出力で表されており，入力 x_i が対象システムにどのような影響を与えているのかをパターン情報から得られると考えられる．この考えに基づき，IDS法の使用経験から得たルールをもちいて分割表現を決定する．まず，基本となるルールを以下に示す．

- インクが薄い部分では，その薄い部分の x_i にたいしての広さから分割表現を決める．
- インクが濃い部分において，ナロウパスの変化が大きいならば，ナロウパスの変化に応じて分割表現を決める．
- インクが濃い部分において，ナロウパスの変化が小さいならば，分割を行わない，もしくはその濃い部分の x_i にたいしての広さから分割表現を決める．

以上のルールをもちいた分割方法を分割アルゴリズムとし，4.2.1節で説明する．

次に決められた分割によりパターン情報を作成し，そのパターン情報からさらに以下のルールをもちいて分割を評価，調整する．

- x_i にたいしてインクが存在しない部分が広いならば，そのパターン情報を構成する分割を取り消す．
- スプレッドが広いならば，スプレッドが狭くなるように分割を行う．

以上のルールをもちいた分割を調整する方法を調整アルゴリズムとして，4.2.2節で説明する．

ここでモデリングにおいて好ましいパターン情報とは以下の2条件を満たすものである．

- スプレッドが広くない
- 任意の x_i にたいしてインクが存在する (すなわち空白ではない)

ナロウパスはインクの重心であるため，スプレッドが広い場合は誤差を含んでしまう．またインクが存在しない場合，ナロウパスを設定できない．上記の2つのルールはこの好ましいパターン情報となるように，分割表現を決定することを目的とする．

4.2.1 分割アルゴリズム

分割アルゴリズムではまず，全てのデータをプロットした $x_i - y$ 平面のパターン情報から， x_i の分割数，分割位置を決定する．これは全データをプロットしたパターン情報から対象システムの特徴を複雑な計算をもちいることなく予測できるためである．例えば $x_i - y$ 平面のパターン情報においてインクが濃い部分とは，データが集中しやすい部分で

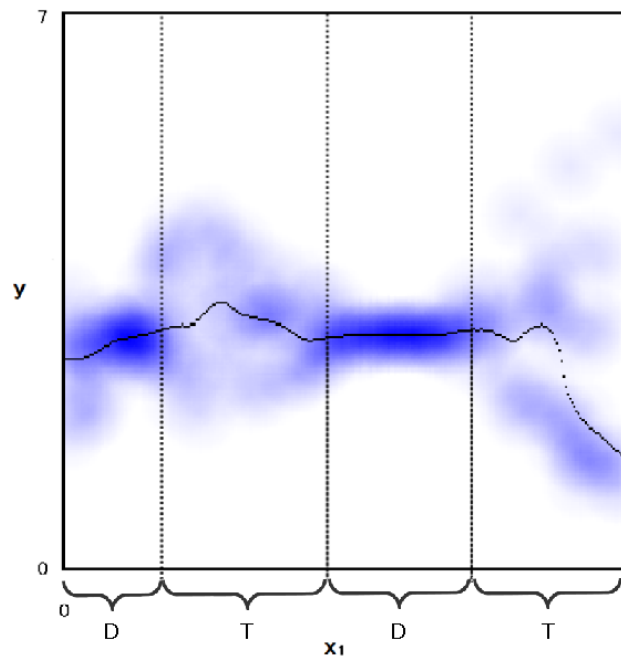


図 4.1 インクの濃淡から分割

あり，出力にたいして入力 x_i の影響が大きく，他の入力の影響が小さいことを表している．逆にパターン情報においてインクが薄い部分では，データのばらつきが大きく，出力にたいして入力 x_i の影響が小さく，他の入力の影響が大きいことを表している．このため図 4.1 のようにパターン情報を濃い，または薄い部分に分けるように x_i を分割し，分割された x_i の各区間 (図 4.1 の D や T) を濃い区間，薄い区間とし，さらに細かな分割を行っていく．

分割アルゴリズムの概要を図 4.2 に示す．分割アルゴリズムでは $x_j - y (i \neq j)$ 平面において作成されるパターン情報が，前述した好ましいパターン情報となるように，入力 x_i の分割表現を決定する．分割アルゴリズムではまず $x_i - y$ 平面のパターン情報を受け取り，インクの濃い部分や薄い部分を同定するために，任意の x_i に対するインクの濃さの最大値を取得する．この情報から閾値をもちいて濃い，または薄いといった 2 値の濃淡情報に置き換える．次に置き換えた濃淡情報からパターン情報の濃い部分，薄い部分を導出し，濃いまたは薄い部分が切り替わる位置で x_i を分割する．続いて分割されて生成された区間毎に，薄い区間であれば分割 1，濃い区間においてナロウパスの変化が大きければ分割 2，それ以外は分割 3 としてヒューリスティックルールに対応した別々の処理を行う．最後に分割情報を集計し， x_i に関する分割表現を決定する．各処理の説明を 4.2.1.1 節から 4.2.1.6 節で行う．

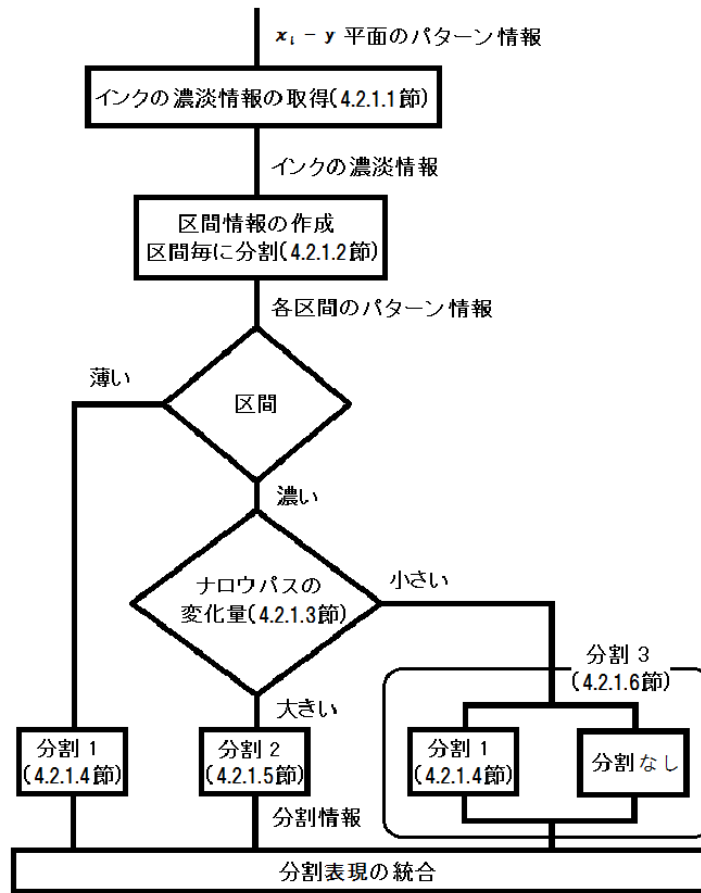


図 4.2 分割アルゴリズムの構造

4.2.1.1 インクの濃淡情報の取得

$x_i - y$ 平面のパターン情報からインクの濃い部分，または薄い部分を同定するために， x_i 方向にたいしてパターン情報が濃い，または薄いといった濃淡情報 $s_j (1 \leq j \leq P)$ を作成する． $x_i - y$ 平面上の位置 (j, k) のインクの濃さ D_{jk} をもちいると，インクの濃淡の判定は，以下の規則に従って行う．

$\max\{S_{jk} \mid 1 \leq k \leq P\} > C$ ならば， s_j は濃い

$\max\{S_{jk} \mid 1 \leq k \leq P\} \leq C$ ならば， s_j は薄い

C は以下の式 (4.1) で求められる閾値である．

$$C = \max_{j,k} \left\{ \frac{1}{2} D_{jk}, K \frac{Rn}{P} \right\} \quad (1 \leq j, k \leq P) \quad (4.1)$$

ここで K は定数であり，ここでは 0.4 としている．また R はインクドロップの中心を通る水平線上の濃さの合計， n は学習データの総数である．式 (4.1) では $x_i - y$ 平面全体で最も濃いインクの濃さの 2 分の 1 と，与えられたデータにおいて出力 y が変化しなかった場合に作成される $x_i - y$ 平面の濃さを K 倍したものを比較し，より濃い方を採用している．このような比較を行う理由は，平面全体にインクが均一に広がっている場合，平面上の濃さの最大値が小さくなり，平面全てが薄いにも関わらず濃いと判断される問題を避けるためである．

4.2.1.2 区間情報の作成

$x_i - y$ 平面のパターン情報において x_i を変化させた際に，閾値 z 以上連続して「濃い」と判定される区間におけるパターン情報を濃い部分，連続して「薄い」と判定される区間におけるパターン情報を薄い部分とする．

この定義より x_i 上の各区間の長さの最小値は z となる．ここで区間の長さが短いほど，その区間において作成される $x_j - y$ 平面 ($j \neq i$) に与えられるデータ量が減少する．データ量が少なくなりすぎると $x_j - y$ 平面のパターン情報において， x_j にたいして空白の部分が作成されてしまう．このため区間の長さの最小値は $x_j - y$ 平面に十分なデータ量を与えられる長さであることが望ましい．そこで，出力 y が一定の場合に，任意の x_i にたいしてインクが存在するであろう最低データ数 d を求め， d を越えるように，区間の長さの最小値となる z を設定する．また各入力にたいして個別に z を定義することは難しいため，各入力における z は同一であるとした．

$$\min\{d \mid d \geq \frac{P}{r}\} \quad (4.2)$$

$$\min\{z \mid z \geq P \left(\frac{d}{n}\right)^{\frac{1}{N-1}}\} \quad (4.3)$$

ここで r はインクの直径である．こうして求めた z を基準として， z より長く連続して濃いと判定されるならば濃い区間， z より長く連続して薄いと判定されるならば薄い区間とし， z よりも短い場合には，その短い部分を挟んでいる区間に取り込むことにしている．

4.2.1.3 ナロウパスの変化量

任意の区間におけるナロウパスにおいて， y 方向にたいしての最大値と最小値の差をナロウパスの変化量とする．この変化量が閾値 $V = 2r$ を越えた場合，ナロウパスの変化が大きいとする．

4.2.1.4 分割1

インクの薄い区間では、データが他の入力に影響して変化することを表しており、この区間でのナロウパスはノイズが入りやすく、分割表現を決定する指標に適さない。そのためこの区間では $leng/v$ 個に等分割する。ここで v は予め設定した閾値 ($z < v$) であり、 $leng$ は対象となっている区間の区間幅である。ここでは $v = r$ としている。

また薄い区間では、濃い区間よりも分割の間隔を広くする必要がある。これは4.2.1節で述べたように薄い区間では他の入力によって出力 y が変化しており、この区間の分割を細かく行くと、 $x_j - y$ 平面で作成されるパターン情報に与えられるデータ量が減少してしまい、出力 y の変化に十分対応できない可能性があるからである。

4.2.1.5 分割2

$x_i - y$ 平面においてインクの濃い区間であり、ナロウパスの変化が大きい場合、 $x_j - y$ 平面で作成されるスプレッドは広くなることが予想される。このため分割2ではナロウパスが一定以上変化した場合に分割を行うことで、 $x_j - y$ 平面で作成されるスプレッドを狭くすることを目的とする。

分割2のアルゴリズムは

```
last = start
```

```
FOR j = start to end DO
```

```
  IF  $\Delta v_{last,j} > V$  AND  $j - last, end - j > z$  THEN
```

```
    位置  $j$  を分割位置とする
```

```
    last = j
```

```
  ENDIF
```

```
ENDFOR
```

となる。ここで $start$ と end は対象となる区間の x_i 軸上での開始位置と終了位置であり、 $last$ は直前の分割位置である。また $\Delta v_{last,j}$ は $last$ から j までの区間におけるナロウパスの変化量である。つまりこの分割2における分割位置は、直前の分割位置から現在着目している位置 j までの区間におけるナロウパスの変化量が V を超え、その位置で分割を行った場合に作成される前後の区間の長さが閾値 z よりも長くなる位置である。また分割数は条件を満たした分割位置の数となる。

4.2.1.6 分割3

分割2と異なり、ナロウパスの変化が小さいということは、その区間の全ての入力にたいして出力が一定であることを意味する。この場合分割を行っても精度の向上につながらない可能性がある。ただし入力 x_i の影響が小さいだけであり、分割を行った方が精度が向

上する場合もある。このため分割を行った場合と行わなかった場合の分割表現を設定，モデリングを行い，精度のよい分割表現を採用する。分割を行う場合，分割 1 と同様の手法で分割を行う。

4.2.2 調整アルゴリズム

分割アルゴリズムによって決定された分割表現では，与えられたデータが各入力にたいして均一でない場合，パターン情報に与えられるデータ量に偏りが発生する。与えられるデータ量が少なければインクの無い区間が発生し，多ければスプレッドが広がってしまう可能性がある。そのため分割アルゴリズムをもちいて作成されたパターン情報に，インクが存在しない部分やスプレッドが広い部分が存在するか調べ，分割の調整を行う。この調整には 4.2.2.1 節や 4.2.2.2 節に示すヒューリスティックルールに基づいたアルゴリズムをもちいる。分割が変化するたびにパターン情報を再構成し，これらのアルゴリズムで調整を行い，調整するべきパターン情報がなくなったら終了とする。

4.2.2.1 調整 1

$x_j - y$ 平面のパターン情報において，インクが存在しない部分がある場合，そのパターン情報を形成する分割位置を調査し，その分割位置での分割を取り消す。これは分割を取り消すことでパターン情報に与えられるデータ量を増やし， x_i にたいして空白のあるパターン情報を減らすためである。ここでインクが存在しない部分とは，パターン情報において出力 y 方向にインクが存在しない部分が， x_i 方向にたいして r より長く連続している部分のことである。

例えば図 3.6 の (B_{21}, B_{22}) 間のデータをもちいて作成された $x_1 - y$ 平面のパターン情報にインクが存在しない部分がある場合，分割位置 B_{21}, B_{22} を調査し，その分割位置での分割を取り消すべきか否かを決定する。 x_i の分割位置の調査では，その分割位置によって作成される $x_j - y$ 平面 ($i \neq j$) におけるパターン情報のナロウパス同士の比較を行う。分割位置 B_{21} を調査する場合， $(0, B_{21})$ 間のデータをもちいて作成された $x_1 - y$ 平面のパターン情報と， (B_{21}, B_{22}) 間のデータをもちいて作成された $x_1 - y$ 平面のパターン情報におけるナロウパス $(\psi_{11}(x_1), \psi_{12}(x_1))$ の比較を行う。比較には平均二乗誤差をもちい，式 (4.4) から求められる誤差 δ が 0.075 よりも小さい場合，その分割位置は取り消される候補となる。

$$\delta = \frac{\sum_{x=1}^P (\psi_{ij}(x) - \psi_{ik}(x))^2}{P} \quad j \neq k \quad (4.4)$$

同様の調査を分割位置 B_{22} についても行い、よりナロウパス間の誤差が小さい候補での分割を取り消す。

4.2.2.2 調整2

パターン情報においてスプレッドが広い場合、そのパターン情報を形成する入力次元をさらに分割する。これは分割を増やすことで、調整1とは逆にパターン情報に与えられるデータ量を減らし、スプレッドを狭くするためである。ここでパターン情報のスプレッドが広いとは、そのパターン情報の平均スプレッドが $2r$ より大きい場合である。

入力次元をさらに分割する場合、最も広い入力次元を等分割する位置を分割位置とし、分割を行う。ただしその入力次元が調整1で分割を取り消されて作成されたものである場合、分割を行わない。また最も広い入力次元が閾値 z の2倍よりも小さい場合も分割を行わない。

4.3 逐次分割表現手法

これまでに紹介した分割表現手法では、入力数が3以上のシステムのモデリングを行うさいに、無駄な分割を行っていた。これは紹介した分割表現手法全てにおいて、分割位置を各入力毎に決定するためであり、分割同士が影響してしまい不要な分割が行われていた。具体的には3入力1出力の対象システム (x_1, x_2, x_3, y) において、 $x_1 - y$ 平面のパターン情報を構成する $(x_2 < B_2, x_3 < B_3)$ 空間を作成する場合、図4.3に示すように、 $(x_2 \geq B_2, x_3 < B_3)$ 、 $(x_2 < B_2, x_3 \geq B_3)$ 、 $(x_2 \geq B_2, x_3 \geq B_3)$ の3種の空間も同時に作成されてしまう。また $x_1 - y$ 平面を構成する入力空間を作成するために利用した分割を、他の平面を構成する入力空間を作成する際にも利用しており、1つの分割が複数の平面の作成に影響を与えていた。これは例えば各入力の分割を (B_1, B_2, B_3) で行った場合、 B_1 は $x_2 - y$ 平面と $x_3 - y$ 平面、 B_2 は $x_1 - y$ 平面と $x_3 - y$ 平面、 B_3 は $x_1 - y$ 平面と $x_2 - y$ 平面の分割の際に利用されることとなり、入力 x_i の分割が他の全ての入力において作成されるパターン情報に影響を与えることとなる。このため分割表現の調整が難しくなっていた。

これらの問題を解決するために、1つの分割によって1つの領域のみが分割される必要がある。さらに $x_i - y$ 平面を構成する入力空間の分割と $x_j - y$ 平面を構成する入力空間の分割を別の分割とするために、各平面毎に分割表現を決定する必要がある。また分割を行う前にその分割を評価することが難しいため、不要な分割が作成される場合もある。そこで以前の分割表現をもとに更に分割、統合を行うことで、段階的に分割表現を調整し $(x_2 < B_2, x_3 < B_3)$ のみを分割する逐次分割表現手法を提案する。この手法は分割アルゴリズムと統合アルゴリズムの2種のアルゴリズムから構成されている。以下に分割アル

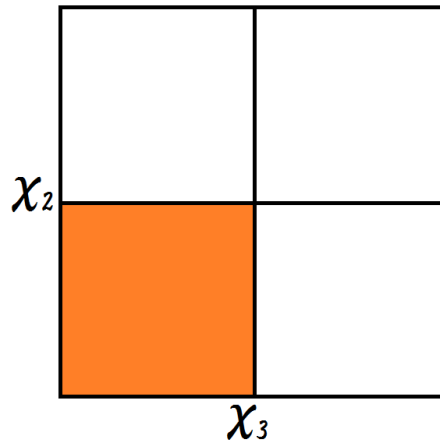


図 4.3 既存の分割方法

ゴリズムを記載する．

- 1) パターン情報を入力空間を分割せずに作成する．
- 2) パターン情報上のスプレッドの最大値がインクドロップの直径 r の 2 倍以上，パターン情報を構成する入力空間の中で最も広い入力範囲が 2Υ 以上ならばその入力を 2 等分する．
- 3) 分割により作成された空間毎にパターン情報を作成する．
- 4) 2 の条件を満たすパターン情報がなくなるまで，2 と 3 を繰り返す．

閾値 Υ は分割しすぎることを防ぐためにもちいる．続いて統合アルゴリズムを記載する．

- 1) x 方向にたいして，その入力範囲の 1 割以上インクが存在しないパターン情報の中で，最もインクが存在しないパターン情報のナロウパスと，そのパターン情報を構成する入力空間と隣接する入力空間で作成されたパターン情報のナロウパスとの平均 2 乗誤差を計算する (式 (4.4)) ．
- 2) 誤差が出力範囲の 1 割以下である隣接入力空間の中で，最も誤差の小さい入力空間を統合し，パターン情報を再度作成する．
- 3) 1, 2 の条件を満たすパターン情報がなくなるまで，1, 2 を繰り返す．

これらのアルゴリズムを各入力毎に作成されるパターン情報に適應することで，作成される分割は各平面毎に別の分割として作成される．

この手法が $(x_2 < B_2, x_3 < B_3)$ のみを分割する手順を説明する．まず分割アルゴリズムが適用され，入力空間 (x_2, x_3) を分割せずに $x_1 - y$ 平面を作成する．次に作成した $x_1 - y$ 平面の最大スプレッドが $2r$ 以上か確認し， x_2 か x_3 のどちらか，範囲の広い方を

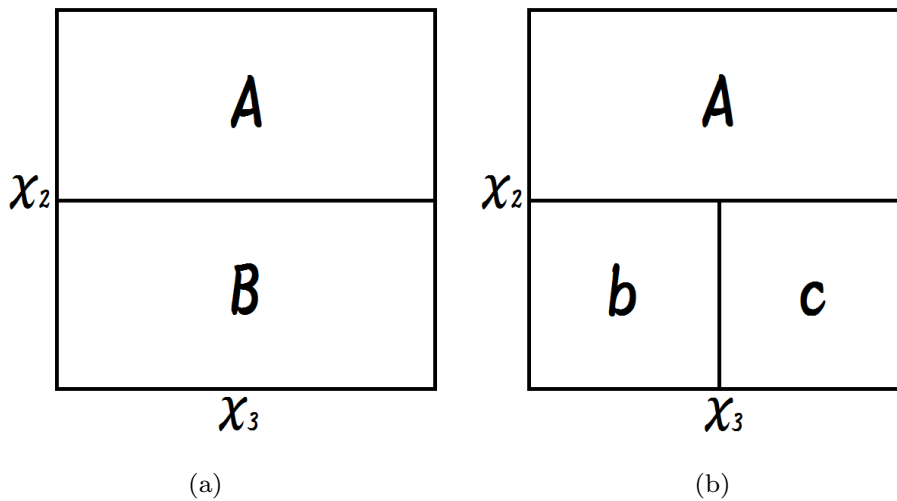


図 4.4 $x_1 - y$ 平面を構成する入力空間の分割例

等分割する．ここでは x_2 が分割されたとし，入力空間 A と B が作成される (図 4.4(a))．作成された入力空間 A, B それぞれにおいてパターン情報を作成する．この時入力空間 A において作成されるパターン情報のスプレッドが小さく， B において作成されるパターン情報のスプレッドが $2r$ 以上であるとする．すると入力空間 B を分割することとなり，範囲の広い入力 x_3 が分割されることとなる． B を x_3 にたいして等分割すると入力空間 b と c が作成される (図 4.4(b))． b と c にたいしても同様にパターン情報を作成し，スプレッドの最大値が大きいを確認する．ここではこれ以上の分割は行われなかったとし，分割アルゴリズムが停止する．

次に統合アルゴリズムが適用され，入力空間 A, b, c において作成されるパターン情報を調査し，最もインクの不足しているパターン情報を選択する．ここで c において作成されるパターン情報においてインクの不足が最も多く，統合アルゴリズム 1) の条件を満たしているとする．すると統合アルゴリズムでは， A と c, b と c において作成されるパターン情報上のナロウパス同士の比較を行う． A と c が統合アルゴリズム 2) の条件を満たしているとする， A と c が統合される．また (B_2, B_3) が入力 (x_2, x_3) をそれぞれ等分割する位置だとすれば，入力空間 (x_2, x_3) から入力空間 $b(x_2 < B_2, x_3 < B_3)$ のみを分割したこととなる．

4.4 提案手法におけるメンバーシップ関数

提案手法では 3.3.1 節の等分割表現手法と違い分割位置が偏る場合がある．このため分割位置を 2 等分する位置を中心とし，その分割位置の差の 2 倍を底辺とした三角型メンバーシップ関数を作成すると，分割間の差が大きい場合のメンバーシップ関数 (図 4.5 に

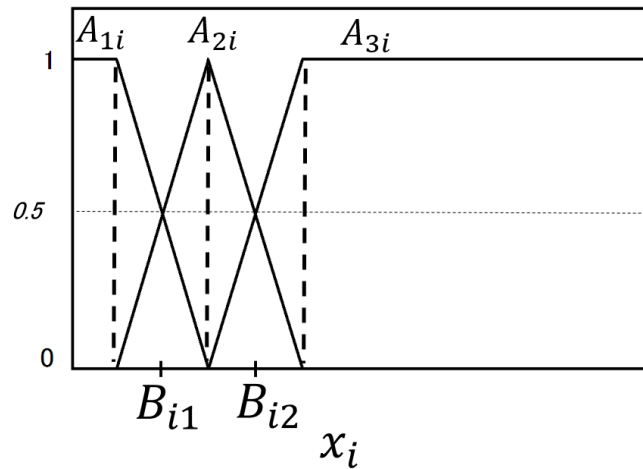


図 4.5 提案手法におけるメンバーシップ関数

における A_{3i}) の影響範囲が大きくなり, 分割間の差が小さい場合のメンバーシップ関数 (図 4.5 における A_{1i} や A_{2i}) の影響が小さくなる. このため先に分割位置の差が小さい場合のメンバーシップ関数を, 等分割表現手法と同様の方法で設定し, 隣接するメンバーシップ関数の傾きが, 分割位置の差が小さい場合のメンバーシップ関数の傾きを反転したものとなるように設定する.

例えば A_{1i} のメンバーシップ関数を作成する場合, $\min\{x_i\}$ 側に他のメンバーシップが無い場合, $\min\{x_i\}$ から B_{i1} と $\min\{x_i\}$ の中点までは 1 となる. B_{i1} 側には隣接しているメンバーシップが存在するため, A_{1i} の分割間 ($\min\{x_i\}$ と B_{i1}) の差と, A_{2i} の分割間 (B_{i1} と B_{i2}) の差を比較する. A_{1i} の分割間の差の方が小さい場合, A_{1i} の分割間の中点から B_{i1} で 0.5 となるように傾きを設定し, 0 まで単調減少させ, 以降は全て 0 とする. A_{1i} の分割間の差の方が大きい場合, $\min\{x_i\}$ から A_{2i} の分割間の中点までの A_{2i} のメンバーシップ関数を先に設定し, その中点までの A_{1i} と A_{2i} のメンバーシップ関数を足し合わせた際に 1 となるように A_{1i} のメンバーシップ関数を設定する. A_{2i} の分割間の中点以降は A_{1i} のメンバーシップ関数は 0 となる. A_{2i} のメンバーシップ関数を作成する場合, A_{1i} と A_{3i} に隣接しているため, A_{2i} の分割間の中点より小さい場合は A_{1i} との関係から, A_{2i} の分割間の中点より大きい場合は A_{3i} との関係からメンバーシップ関数が設定される.

第 5 章

手法の比較

5.1 2 入力関数近似

5.1.1 各分割表現手法をもちいた IDS 法と FNN の比較

近似対象として図 5.1 と表 5.1 に示す関数をもちいる．これらの関数は文献 [5] でもちいられた関数近似のベンチマークである．これらの関数は g_1 から g_5 へと出力の変化が複雑になっていることがわかる．これらの関数の近似を各分割表現手法をもちいた IDS 法で行わせ，近似精度，学習時間，最終的なパターン情報数，分割表現の探索中に作成された最大パターン情報数の比較を行う．また IDS 法以外の他手法として FNN との比較を行う．

近似精度には文献 [5] にて使用されているばらつきを考慮した誤差の和 (fraction of variance unexplained ; FVU) をもちいる．

$$FVU = \frac{\sum_{l=1}^L (\hat{y}(x_l) - y_l)^2}{\sum_{l=1}^L (y_l - \bar{y})^2} \quad (5.1)$$

$$\bar{y} = (1/L) \sum_{l=1}^L y_l$$

ここで $x_l (l = 1, 2, \dots, L)$ は l 番目の入力ベクトルで， y_l は対象システムの正解出力であり， $\hat{y}(x_l)$ は生成されたモデルの出力である．FVU の分母は対象システムの正解出力の偏差平方和，分子は対象システムの正解出力と生成されたモデルの出力の 2 乗誤差である．つまり値が小さいほど精度が高いことを意味する．学習時間として学習データを取得し，学習が終了するまでの時間を比較する．これは IDS 法では分割表現の探索を開始してから，最終的な分割表現を決定しモデルを作成するまでの時間であり，FNN では結合強度の調整を開始してから，収束するまでの時間とする．最終的なパターン情報数とは分割表現手法によって決定されたパターン情報の数であり，IDS 法が作成する平面の枚数であ

る．最終的なパターン情報数が少ないということは，少ない平面枚数でよい精度を示すことができる分割表現手法ということであり，より適切に分割数や分割位置を決定できていることを意味する．また分割表現手法毎に作成する最大パターン情報数には差があり，この値が小さいほど IDS 法が作成する最大平面枚数が小さいことを意味し，必要な計算資源が小さいことを意味する．

学習データ数は 225 とし，各関数からランダムに 100 セット作成する．テストデータには各入力の間隔が均等になるように 10000 データを用意する．IDS 法の解像度は 256 とし，インクの直径は解像度の 15% となる 39 とする．等分割表現手法でもちいる分割の最大値と最小値は 10 と 1 とする．GA をもちいた分割表現手法で使用する染色体の母集団数は 30，世代数は 20 とする．また染色体長の最大値を，等分割表現手法の最大分割数と同じ分割数となるように，19 とする．逐次分割表現手法においてもちいる閾値 γ は，等分割表現手法の最大分割数と近い分割数となるように，各入力の 12.5% とした．FNN のニューロン数は 10 とし，学習前の結合強度と学習後の結合強度の差が，学習前の結合強度の 0.5% 未満となるまで学習を繰り返した．

図 5.2~5.6 に 100 試行の平均近似精度を，図 5.7 に平均学習時間を，図 5.8 に最終的なパターン情報数の平均を，図 5.9 に最大パターン情報数の平均を示す．またこれらの値を付録 A.1 節に掲載する．更に図 5.10 に逐次分割手法による IDS 法の出力例を，図 5.10 に FNN の出力例を示す．等分割手法，GA をもちいた分割表現手法，ヒューリスティックルールをもちいた分割表現手法による IDS 法の出力例については付録 A.1 節に掲載する．関数 g_1, g_2 について IDS 法は FNN に近似精度で劣るものの，関数 $g_3 \sim g_5$ に関しては勝っている．これは IDS 法が，FNN のように結合強度の最適化を行うことで対象システムの特徴を表現せず，パターン情報としてデータから対象システムの特徴を抽出することができるためであり，出力の変化が複雑な関数ほど FNN よりもよい精度を示すことができることがわかる．これは図 5.10, 5.11 からわかり，関数 g_1, g_2 では IDS 法の出力はなめらかな曲線を描けておらず，FNN の方がより対象システムを再現できている．これは IDS 法が射影やインクの重心をナロウパスとしてもちいること，出力の計算に利用するメンバーシップ関数がナロウパス間の出力を線形補間するように設定されていることが原因である．しかし関数 $g_3 \sim g_5$ では FNN が出力の変化に対応しきれず，再現できていない出力の変化が存在し，IDS 法の方が出力の変化をよく表せていることがわかる．分割表現手法の違いによる近似精度の差は，FNN との差ほど大きくないが，等分割表現手法と提案手法 2 種の精度がよいことがわかる．次に図 5.7 より，提案手法 2 種は FNN や既存手法と比較して学習時間が短く，特に逐次分割表現手法の学習時間が短いことがわかる．また図 5.8 から提案手法 2 種は等分割表現手法よりもパターン情報数が少なくなった．GA をもちいた分割表現手法が最もパターン情報数を少なくしているが，図 5.2~5.6 より GA をもちいた分割表現手法が IDS 法の中で最も精度が悪く，分割表現を正しく調

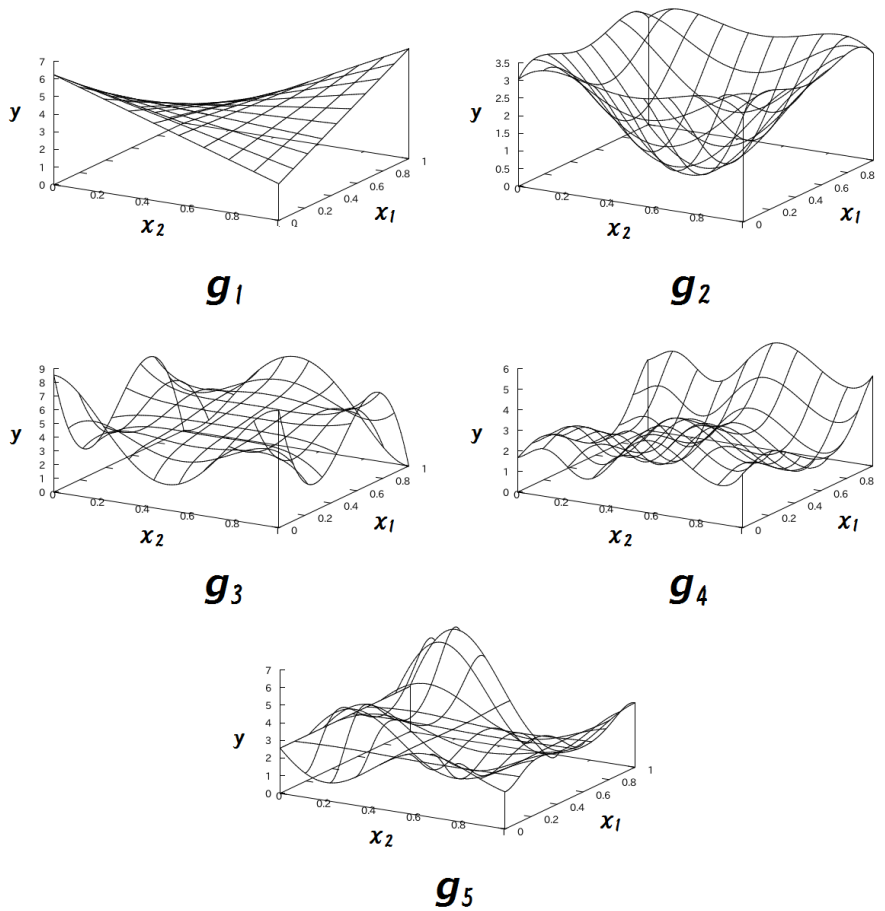


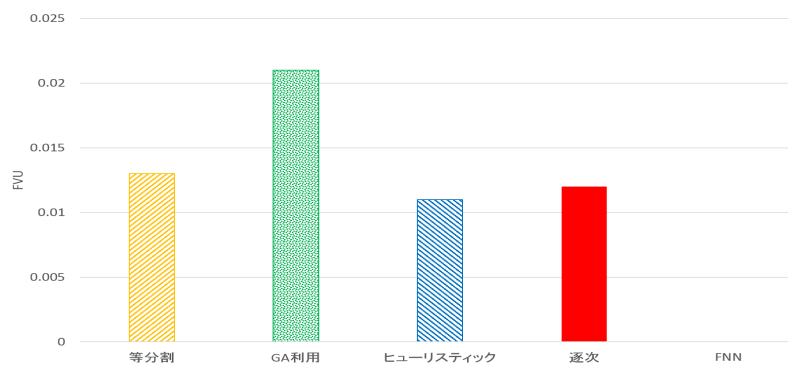
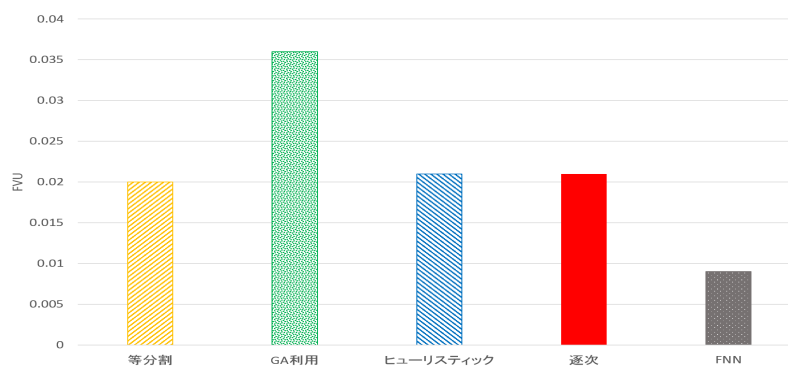
図 5.1 Hwang の 5 関数

整できていないことがわかる。これらのことから、提案手法 2 種はパターン情報数を減らしよい精度となる分割位置の探索に成功していると言え、特にヒューリスティックルールをもちいた分割表現手法は分割表現をよく調整できていると言える。次に図 5.9 から、提案手法 2 種は既存手法と比較して作成されるパターン情報の最大数が小さく、特にヒューリスティックルールをもちいた分割表現手法の最大パターン情報数が小さいことがわかる。以上のことから、提案手法は以前に作成したパターン情報から適切に情報を抽出し、構造の決定に役立てていることがわかる。

表 5.1 Hwang の 5 関数

$g_1(x_1, x_2)$	$= 10.3910((x_1 - 0.4000)(x_2 - 0.6000) + 0.3600)$
$g_2(x_1, x_2)$	$= 24.2340(r^2(0.7500 - r^2))$ $r^2 = (x_1 - 0.5000)^2 + (x_2 - 0.5000)^2$
$g_3(x_1, x_2)$	$= 42.6590(0.1000 + \tilde{x}_1(0.0500 + \tilde{x}_1^4 - 10.0000\tilde{x}_1^2\tilde{x}_2^2 + 5.0000\tilde{x}_2^4))$ $\tilde{x}_1 = x_1 - 0.5000 \quad \tilde{x}_2 = x_2 - 0.5000$
$g_4(x_1, x_2)$	$= 1.3356(1.5000(1.0000 - x_1) + e^{2.0000x_1 - 1.0000} \sin(3.0000\pi(x_1 - 0.6000)^2)$ $+ e^{3.0000(x_2 - 0.5000)} \sin(4.0000\pi(x_2 - 0.9000)^2))$
$g_5(x_1, x_2)$	$= 1.9000(1.3500 + e^{x_1} \sin(13.0000(x_1 - 0.6000)^2)e^{-x_2} \sin(7.0000x_2))$

$0 \leq x_1, x_2 \leq 1$

図 5.2 g_1 の近似精度図 5.3 g_2 の近似精度

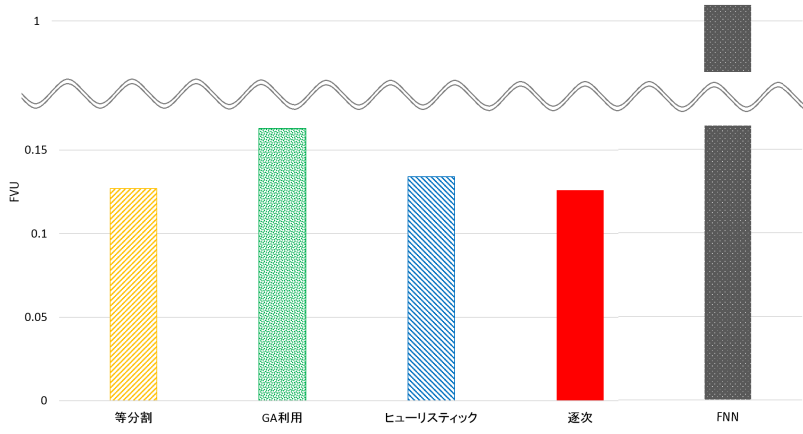


図 5.4 g_3 の近似精度

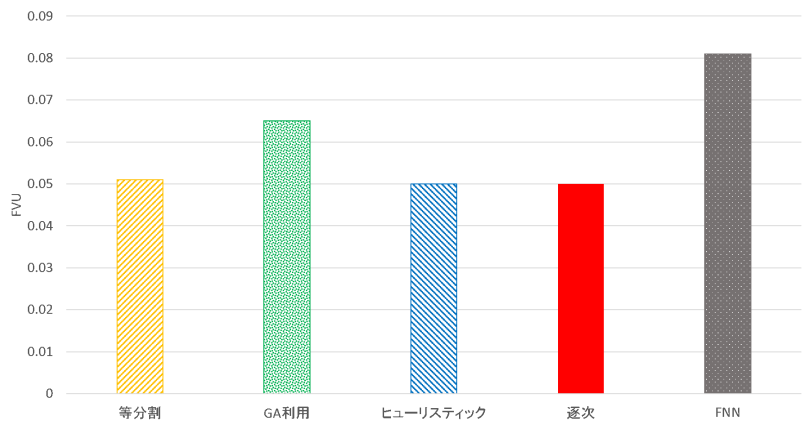


図 5.5 g_4 の近似精度

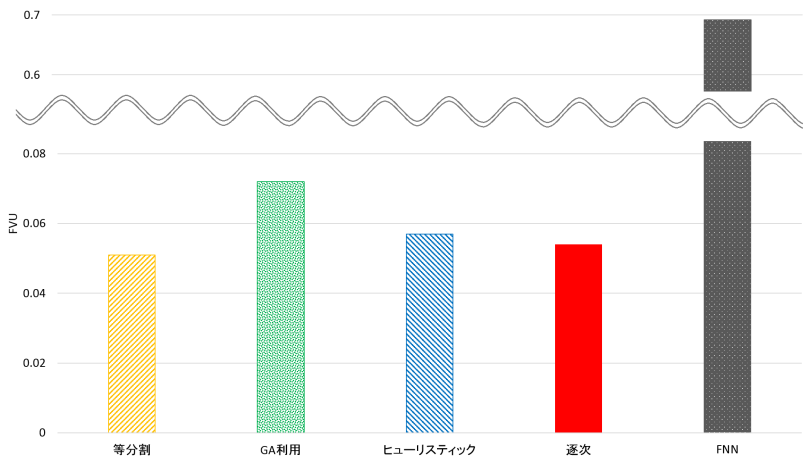


図 5.6 g_5 の近似精度

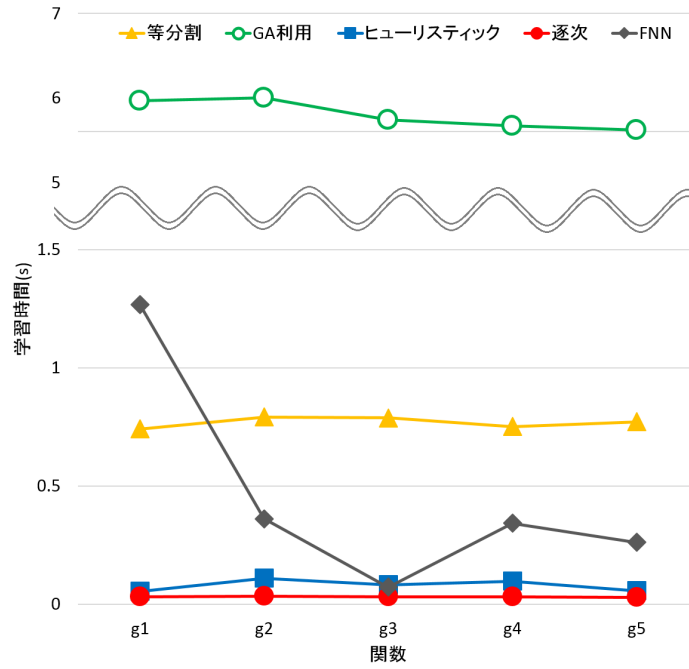


図 5.7 Hwang の 5 関数における学習時間

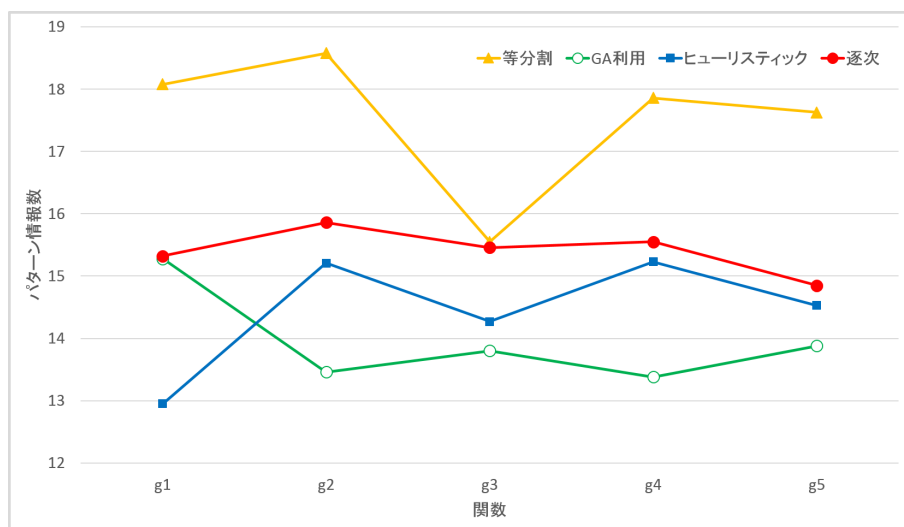


図 5.8 Hwang の 5 関数における IDS 法の最終的なパターン情報数

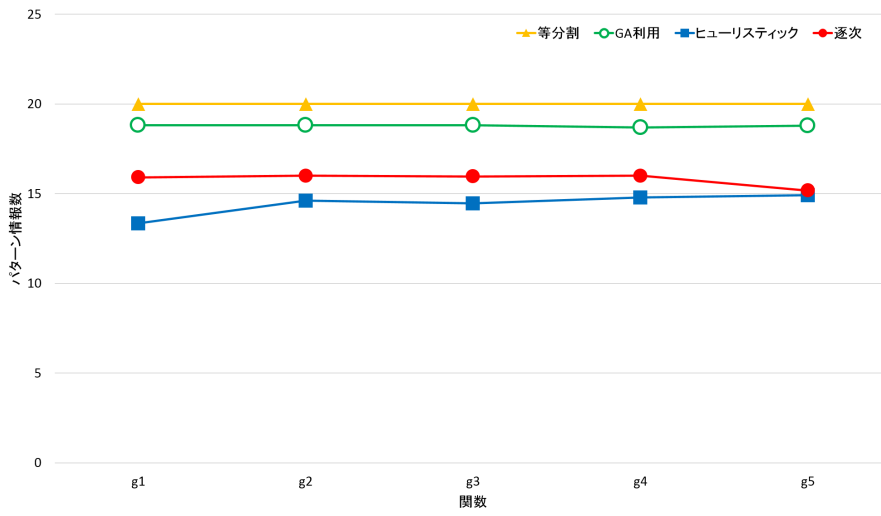


図 5.9 Hwang の 5 関数における IDS 法の最大パターン情報数

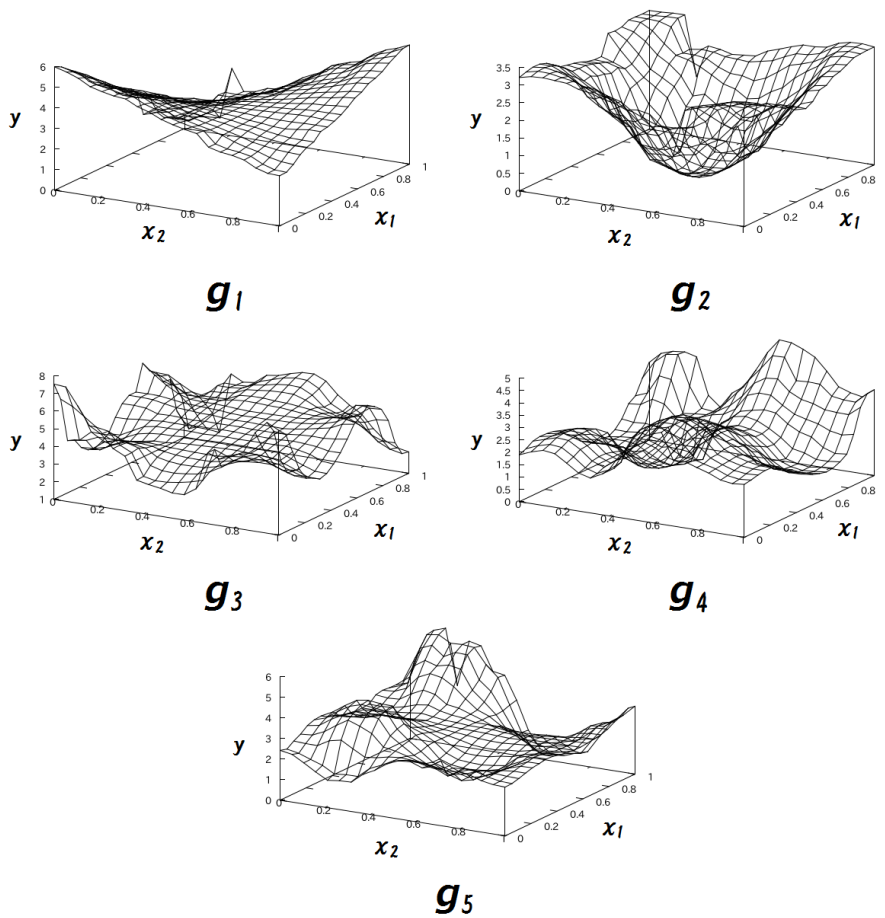


図 5.10 Hwang の 5 関数における逐次分割手法による IDS 法の出力例

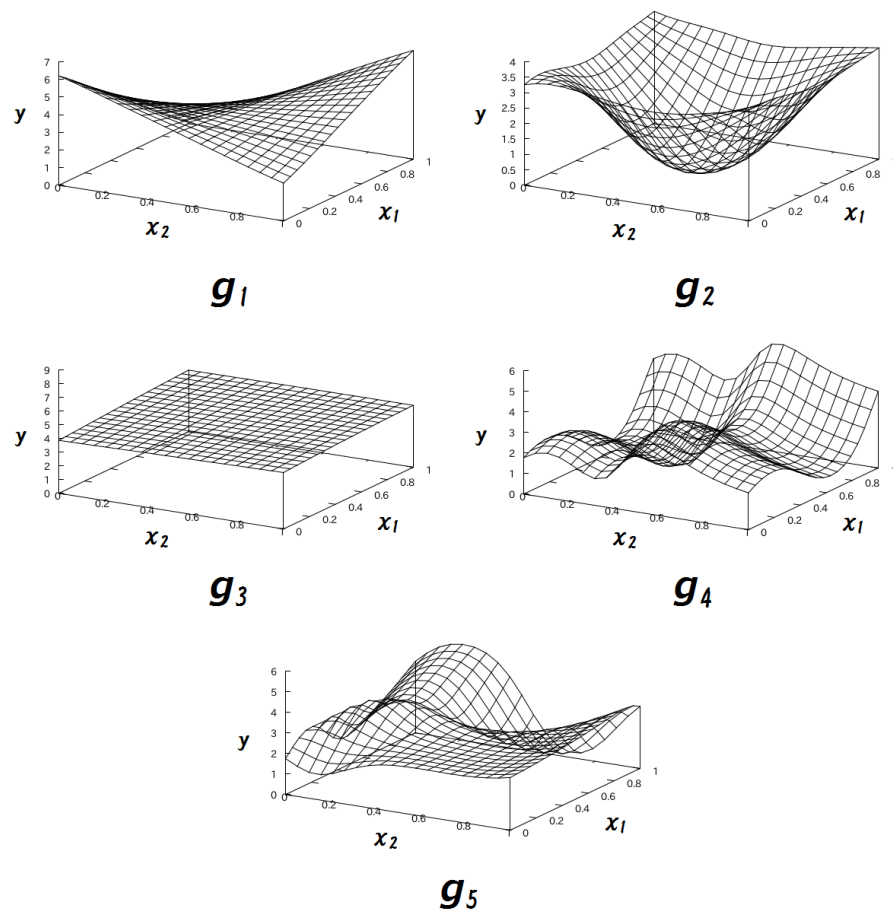


図 5.11 Hwang の 5 関数における FNN の出力例

5.1.2 メンバーシップ関数の違いによる比較

IDS 法の出力は使用するメンバーシップ関数によって変化する．このため使用するメンバーシップ関数が変われば，IDS 法の精度も変化する．ここでは 5.1.1 節でもちいた Hwang の 5 関数をもちいて，メンバーシップ関数が IDS 法の近似精度にどのような変化を与えるか検証する．また分割表現手法の違いによる精度の差は小さいことが予想されるため，ここでは比較しない．メンバーシップ関数には三角型，台形型，ガウシアン型の 3 種のメンバーシップ関数 (図 5.12~5.14) をもちいる．分割表現には各入力を 5 等分したものをもちいる．

表 5.2 に各メンバーシップ関数をもちいた IDS 法の 100 回分の平均近似精度を示す．関数 g_1, g_2 では三角型のメンバーシップ関数をもちいた場合に最も精度がよく，関数 $g_3 \sim g_5$ では台形型のメンバーシップ関数をもちいた場合に最も精度がよくなった．以上のことから，関数 g_1, g_2 のような変化の緩やかな関数は 1 つのナロウパスから出力を計算するよりも，複数のナロウパスから出力の変化を計算することでより精度のよいモデルとなることがわかる．一方，関数 $g_3 \sim g_5$ のような変化の複雑な問題では，ナロウパスどうしの違いが大きくなってしまいうため，複数のナロウパスから出力の変化を予測するよりも，各入力毎に 1 つのナロウパスから出力を計算する部分が多い方が精度のよいモデルとなることがわかる．またガウシアン型メンバーシップ関数をもちいた IDS 法では，入力毎にルールの適合度 Γ の総和が変わってしまい，特定の入力におけるナロウパスのみから出力が計算される場合がある．このためガウシアン型メンバーシップ関数をもちいた IDS 法では精度がよくならなかった．

GA をもちいた分割表現手法ではメンバーシップ関数の構造を染色体として利用しており，三角型や台形型のメンバーシップ関数を元々取ることができる．また提案手法では 4.4 節で記載したように，分割位置が偏る場合があり，等分割表現手法と同様の三角型のメンバーシップ関数をもちいると，分割位置の差が小さい部分の影響が小さくなり精度が低下する．このため提案手法では 1 つの分割位置の差のみではなく，両隣の分割位置との関係を考慮して三角型と台形型のメンバーシップ関数を複合して利用しており，台形型メンバーシップ関数のみとしても影響が小さいことが予想される．またガウシアン型のメンバーシップ関数は分割表現手法に関わらず精度が低下する可能性がある．等分割表現手法においても分割数が上昇すると 3 種のメンバーシップ関数の差が小さくなり，影響が小さくなることが予想される．

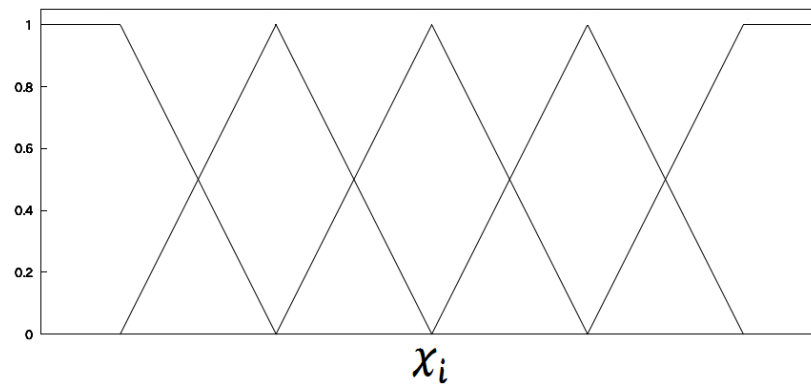


図 5.12 三角型メンバーシップ関数

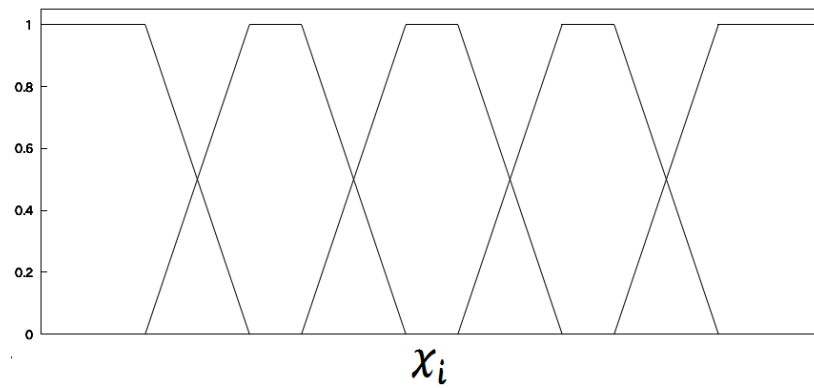


図 5.13 台形型メンバーシップ関数

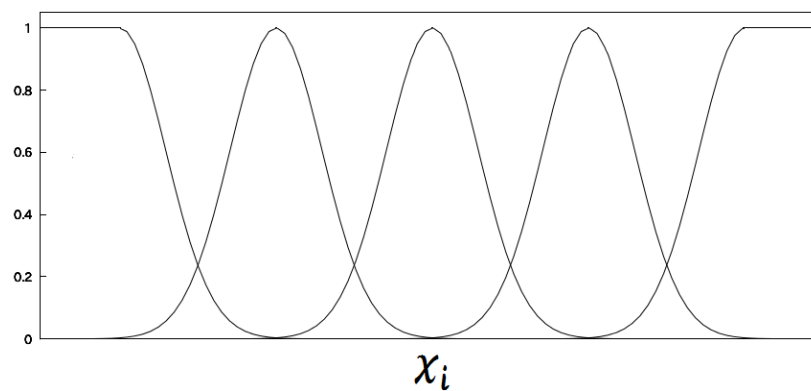


図 5.14 ガウシアン型メンバーシップ関数

表 5.2 メンバシップ関数毎の近似精度:FVU

	g_1	g_2	g_3	g_4	g_5
三角型	0.013	0.026	0.159	0.068	0.089
台形型	0.015	0.029	0.151	0.063	0.085
ガウシアン型	0.018	0.033	0.154	0.067	0.089

5.2 3 入力関数近似

次に関数 (5.2), (5.3) に示す 3 入力関数の近似問題をもちいる。関数 (5.2) は文献 [12] で用いられた 3 入力関数であり, 関数 (5.3) は 5.1 節でもちいた関数 g_5 を改変して作成した。

$$h_1(x_1, x_2, x_3) = \frac{1}{1 + e^{-e^{x_1 + (x_2 - 0.5)^2 + 3 \sin(\pi x_3)}}} + 2x_1 \sin(2\pi x_2) \quad (5.2)$$

$$h_2(x_1, x_2, x_3) = \prod_{i=1}^3 e^{x_i} \sin(13(x_i - 0.6)^2) \quad (5.3)$$

比較には 5.1.1 節と同様に近似精度, 学習時間, 最終的なパターン情報数, 最大パターン情報数をもちいる。

関数 (5.2), (5.3) からランダムに学習データを作成する。学習データ数を 500, 1000, 1500 の 3 種とし, それぞれのデータ数毎に 100 セット用意し, 学習を 100 回行う。テストデータには各入力を等間隔に取った 1,000,000 データをもちいる。IDS 法の解像度は 256 とし, インクの直径は解像度の 15% となる 39 とした。等分割表現手法でもちいる分割の最大値と最小値は 10 と 1 とした。GA をもちいた分割表現手法で使用する染色体の母集団数は 40, 世代数は 30 とした。また染色体長の最大値を, 等分割表現手法の最大分割数と同じ分割数となるように, 19 とした。逐次分割表現手法においてもちいる閾値 Υ は, 等分割表現手法の最大分割数と近い分割数となるように, 各入力の 12.5% とした。FNN のニューロン数は 20 とし, 学習前の結合強度と学習後の結合強度の差が, 学習前の結合強度の 0.5% 未満となるまで学習を繰り返した。

図 5.15 に関数 (5.2) の 100 試行の平均近似精度を, 図 5.16 に関数 (5.3) の 100 試行の平均近似精度を, 図 5.17 に関数 (5.2) の平均学習時間を, 図 5.18 に関数 (5.3) の平均学習時間を, 図 5.19 に関数 (5.2) の最終的なパターン情報数の平均を, 図 5.20 に関数 (5.3) の最終的なパターン情報数の平均を, 図 5.21 に関数 (5.2) の最大パターン情報数の平均を, 図 5.22 に関数 (5.3) の最大パターン情報数の平均を示す。またこれらの値を付録 A.2 節に掲載する。関数 (5.3) における各手法による入力空間の分割例を, 図 5.23~5.26 に

示す．

図 5.15 より，関数 (5.2) において IDS 法は FNN と比較して精度が劣っている．これは関数 (5.2) が変化の緩やかな関数であり，5.1 節でもちいた関数 g_1, g_2 と同様の理由である．また図 5.16 より，関数 (5.3) において IDS 法は FNN と比較して精度がよい．これは関数 (5.3) が複雑な関数であり，5.1 節でもちいた関数 $g_3 \sim g_5$ と同様の理由で精度が勝っている．IDS 法の各分割表現手法の中では，関数 (5.2)，(5.3) のどちらとも提案手法の 1 つである逐次分割表現手法が最もよい精度を示した．ただし関数 (5.2) においてデータ数が増加した場合には，既存の分割表現手法の近似精度と同程度となった．5.1 節と違いヒューリスティックルールをもちいた分割表現手法は関数 (5.2)，(5.3) の両方で精度が悪くなった．これは入力数が増えたことにより，4.2.2 節の調整アルゴリズムによる分割，削除が困難になったためだと考えられる．図 5.19，5.20 に示す最終的なパターン情報数と，図 5.21，5.22 に示す最大パターン情報数の差があまりないことや，図 5.25 の分割例からも，ヒューリスティックルールをもちいた分割表現手法が分割を調整できていないことがわかる．精度と最終的なパターン情報数も合わせて考慮すると，逐次分割表現手法が最も精度よく，かつパターン情報を少なくしてモデリングできていることがわかる．これは図 5.26 の分割例のように，逐次分割表現手法が他手法よりも複雑な分割表現を作成できるためである．ただし関数 (5.2) において，学習データ数が多い場合 GA をもちいた分割表現手法が最もよい分割表現を決定している．これは関数 (5.2) が変化の緩やかな関数であり，GA をもちいた分割表現手法ではメンバーシップ関数を調整することが可能であるため，少ないパターン情報数でよい精度を示せたと考えられる．次に図 5.17，5.18 より，提案手法 2 種はどちらも既存手法よりも学習時間が短く，関数 (5.2) では FNN より短く，関数 (5.3) では FNN と同程度となっている．また等分割表現手法は 2 入力関数近似の場合と比較して，学習時間がデータの上昇量よりも多く増えている．これは入力数が増えたことにより探索すべき分割数の組み合わせが増大してしまったためである．図 5.21，5.22 より，提案手法 2 種はどちらも最大パターン情報数が小さくなっており，使用する計算資源が抑えられていることがわかる．また逐次分割表現手法が計算資源の使用量を抑えて，最もよい精度となる分割表現を決定していることがわかる．ヒューリスティックルールではパターン情報から一定の分割数が作成され，その後調整されるため，関数 (5.3) では最大パターン情報数が小さくならず，関数 (5.2) では調整アルゴリズムが機能せず最大パターン情報数が増加しなかった．一方，逐次分割表現手法では少ないパターン情報数から徐々に増加させるため，計算資源の使用量を抑えることができた．以上のことから，入力数が増加した場合でも，逐次分割表現手法は適切な分割表現を少ない計算資源で高速に決定することができることがわかる．

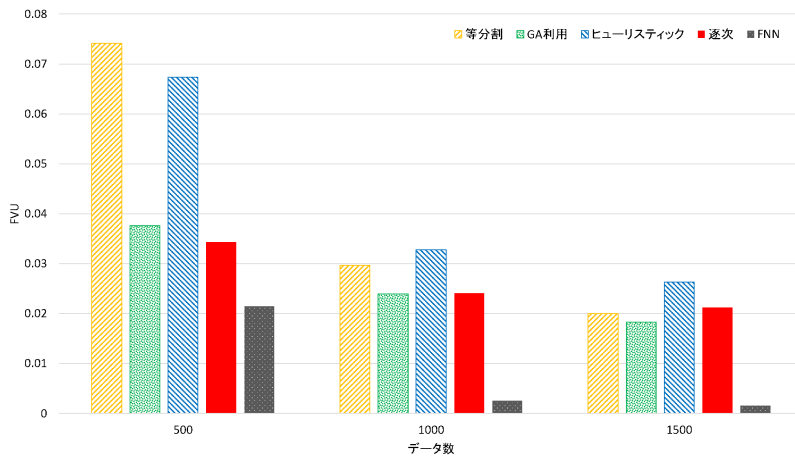


図 5.15 関数 (5.2) の近似精度

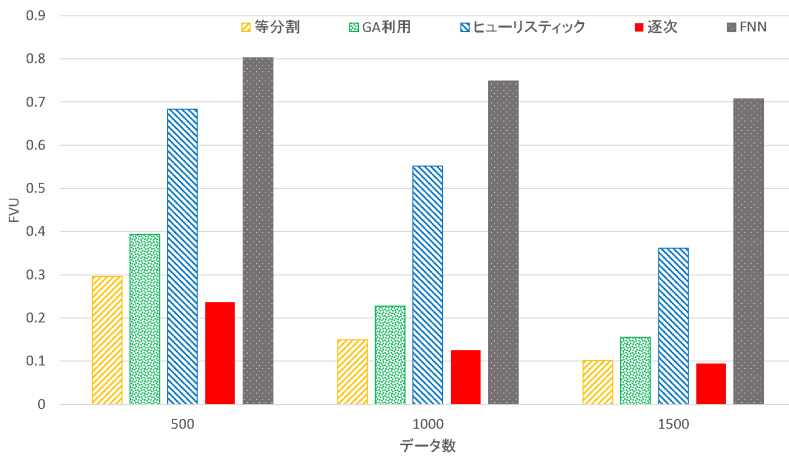


図 5.16 関数 (5.3) の近似精度

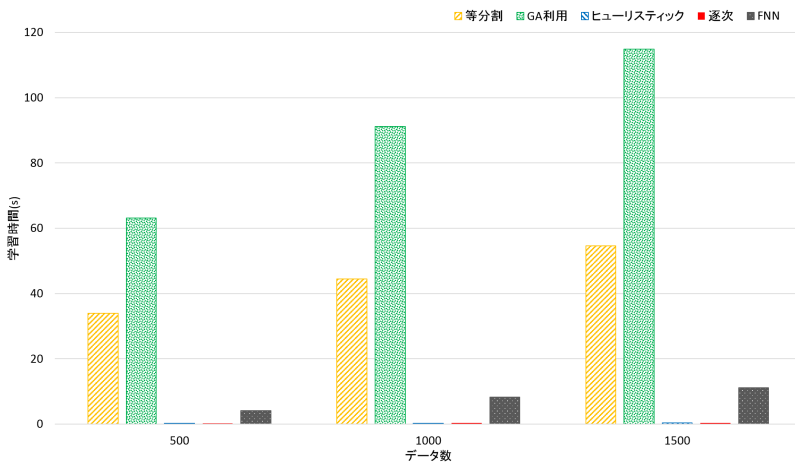


図 5.17 関数 (5.2) における学習時間

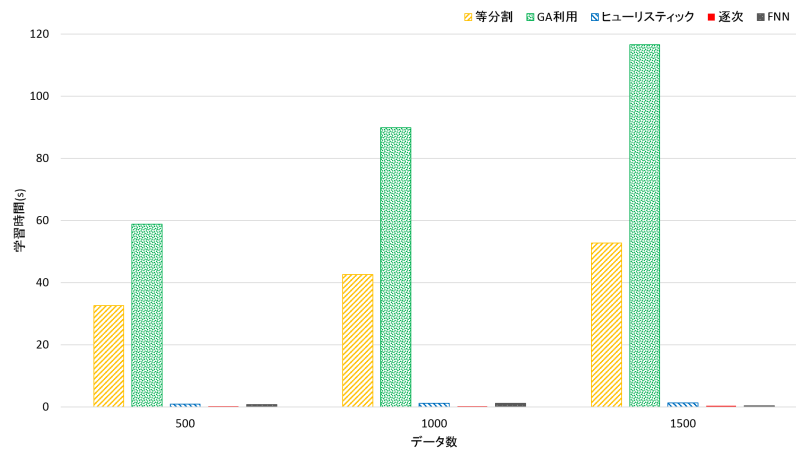


図 5.18 関数 (5.3) における学習時間

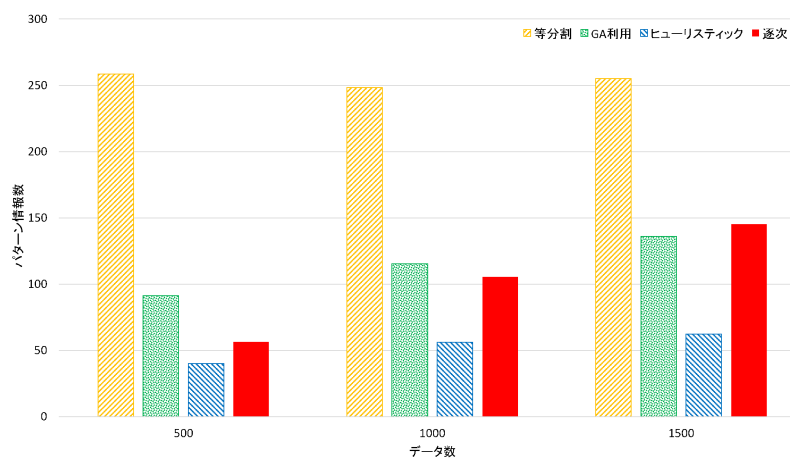


図 5.19 関数 (5.2) における IDS 法の最終的なパターン情報数

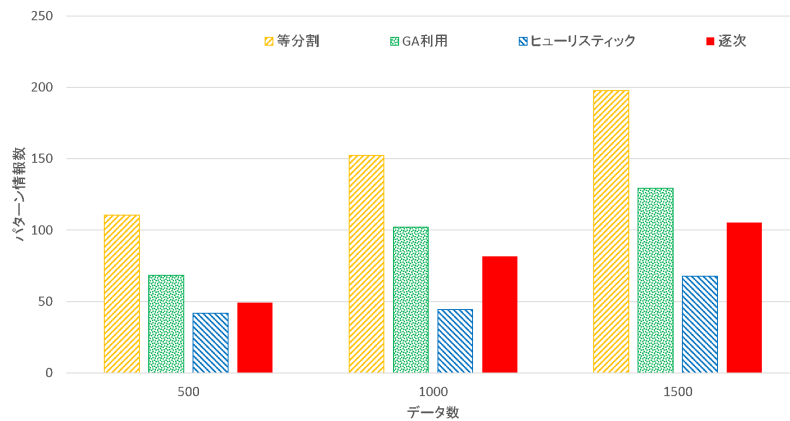


図 5.20 関数 (5.3) における IDS 法の最終的なパターン情報数

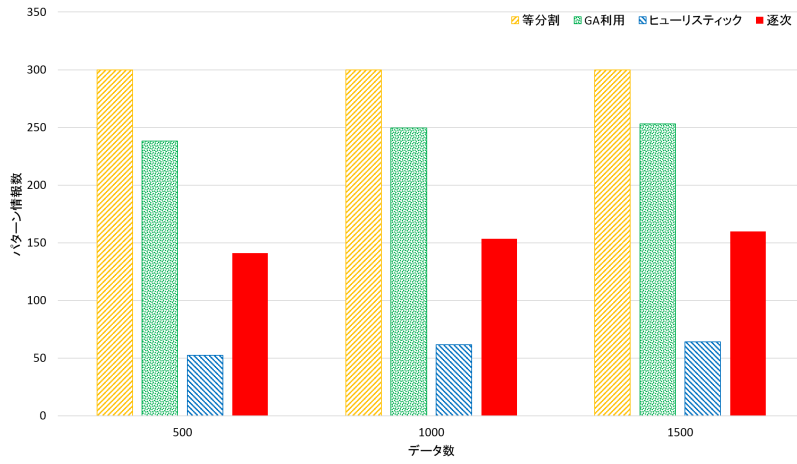


図 5.21 関数 (5.2) における IDS 法の最大パターン情報数

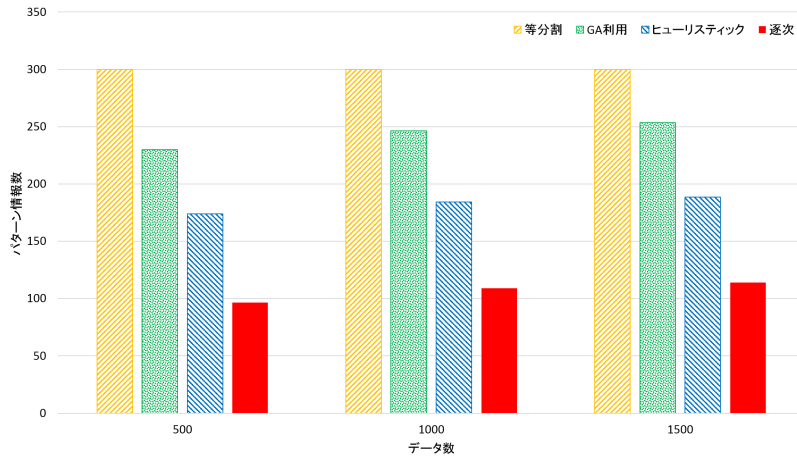


図 5.22 関数 (5.3) における IDS 法の最大パターン情報数

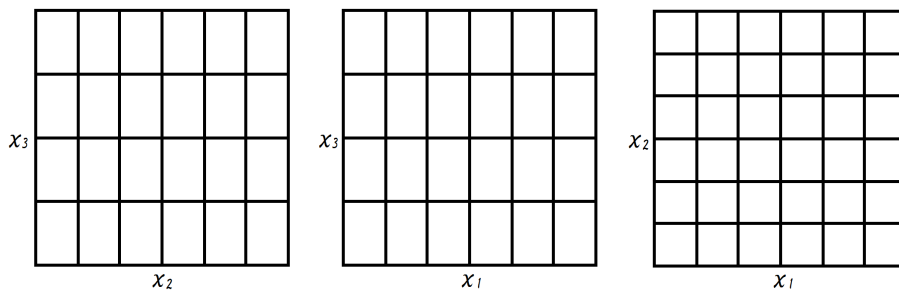


図 5.23 等分割表現手法の分割例

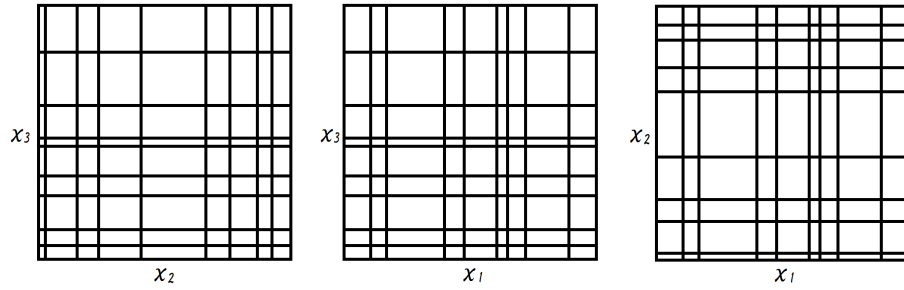


図 5.24 GA をもちいた分割表現手法の分割例

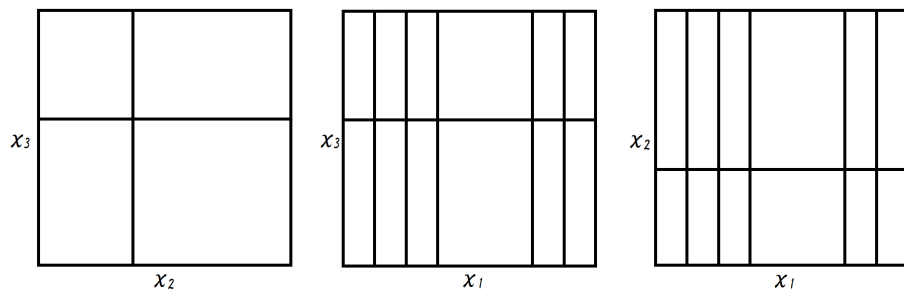


図 5.25 ヒューリスティックルールをもちいた分割表現手法の分割例

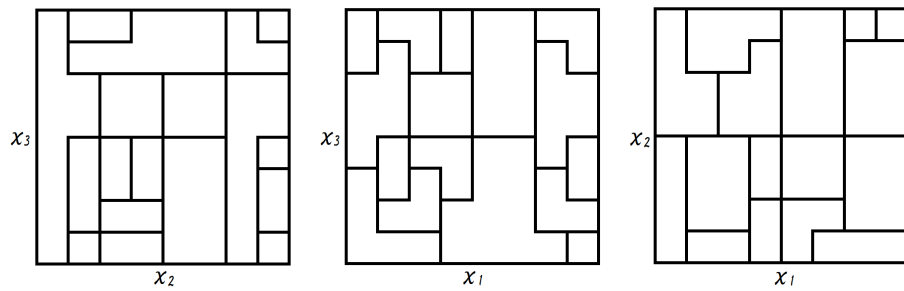


図 5.26 逐次分割表現手法の分割例

5.3 2値分類問題

2値分類問題にIDS法を適用し、分割表現手法の違いによる性能の変化を検証する。2値分類問題には2重らせん識別問題と、Pima Indians Diabetes データセットをもちいる。2重らせん識別問題は文献 [9] にてもちいられたベンチマークであり、識別能力の検証のために様々な手法 [4][6] において採用されている。Pima Indians Diabetes データセットは文献 [8] にてもちいられたベンチマークであり、UCI machine learning repository^{*1}より取得しもちいる。これらの問題の出力は0か1の2値であるため、IDS法では関数近似問題と同様の方法で学習を行い、モデルの出力が0.5以上ならば1を、0.5未満ならば0を出力させることで対応する。比較には識別率とパターン情報数をもちいる。またIDS法以外の他手法として、FNNとSVMとの比較を行う。SVMにはパッケージとして利用されているTinySVM^{*2}をもちいる。TinySVMにはカーネル関数として線形、多項式、Neural、Radial Basis Function(RBF)、ANOVAの5種類が用意されているため、最も識別率のよくなるカーネル関数をもちいて比較を行う。またその他のパラメータはデフォルトのものをもちいる。

5.3.1 2重らせん識別問題

2重らせん識別問題は図5.27に示す2種のらせんを識別する問題である。学習データ数は1つのらせんにつき97データ、総計194データ(図5.27)とし、テストデータにはその倍の388データ(図5.28)をもちいた。IDS法の解像度は64、インクの直径は解像度の10%となる7とした。等分割表現手法でもちいる分割の最大値と最小値は10と1とした。GAをもちいた分割表現手法で使用する染色体の母集団数は30、世代数は20とした。また染色体長の最大値を、等分割表現手法の最大分割数と同じ分割数となるように、19とした。逐次分割表現手法においてもちいる閾値 Υ は、等分割表現手法の最大分割数と近い分割数となるように、各入力の12.5%とした。FNNのニューロン数は10とし、学習前の結合強度と学習後の結合強度の差が、学習前の結合強度の0.5%未満となるまで学習を繰り返した。

図5.29に識別率を、図5.30に学習時間を、図5.31に最終的なパターン情報数を、図5.32に最大パターン情報数を示す。またこれらの値を付録A.3節に掲載する。図5.29より、IDS法はFNNより精度がよく、SVMと同程度の精度となった。IDS法内ではヒューリスティックルールをもちいた分割表現手法が最も精度がよくなり、SVMよりもよい精

*1 <http://archive.ics.uci.edu/ml/>

*2 <http://chasen.org/taku/software/TinySVM>

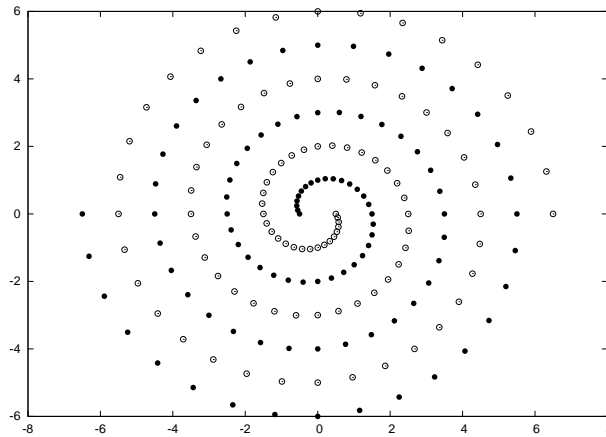


図 5.27 2重らせん識別問題の学習データ

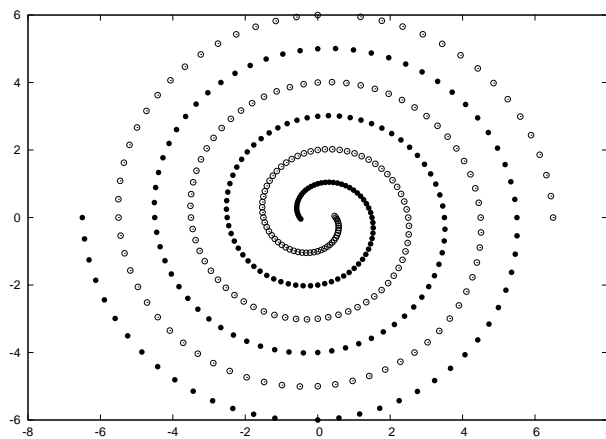


図 5.28 2重らせん識別問題のテストデータ

度となった。図 5.29 より、学習時間は提案手法 2 種と SVM が最も短くなった。また図 5.31 よりヒューリスティックルールをもちいた分割表現手法が最も多いパターン情報数となった。図 5.31, 5.32 から GA をもちいた分割表現手法以外、最終的なパターン情報数が最大パターン情報数とほぼ一致した。このことから、IDS 法が精度よくこの 2 重らせん識別問題に適應するためには分割数を増やし多くのパターン情報を作成したほうがよいことがわかる。ヒューリスティックルールをもちいた分割表現手法以外では最大分割数を事前に設定する必要があるが、最大分割数を事前に大きく設定してしまうと、パターン情報数が少なくてもよい問題においても、最大パターン情報数が大きくなってしまい計算資源を多く使用してしまうといった問題がある。ヒューリスティックルールをもちいた分割表現手法は問題に応じて自動的にパラメータを設定できるため、この問題を回避できていることがわかる。

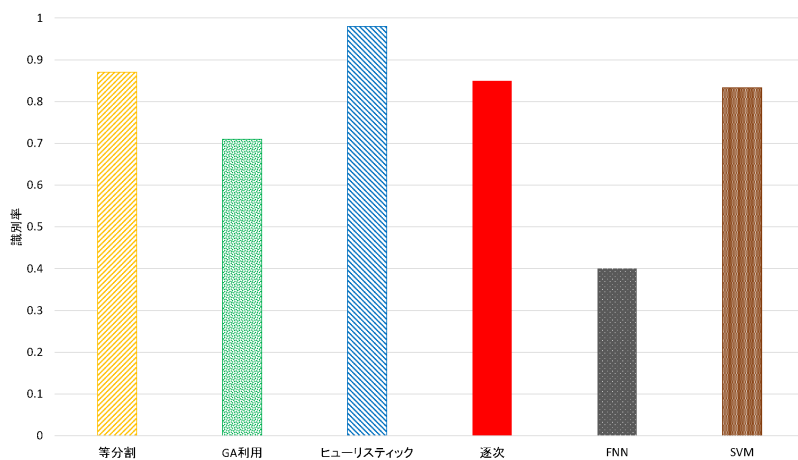


図 5.29 2重らせん識別問題の識別率

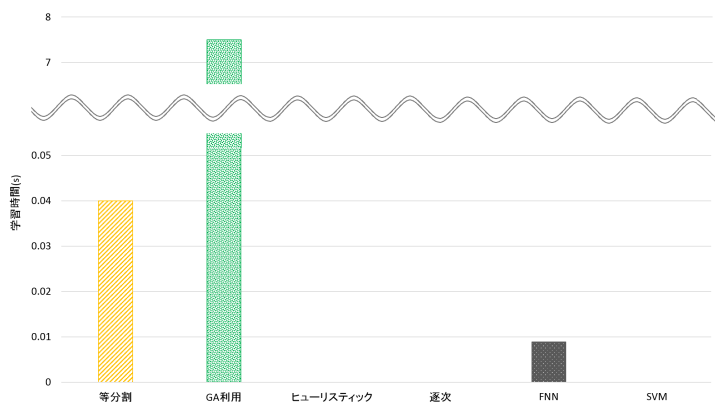


図 5.30 2重らせん識別問題における学習時間

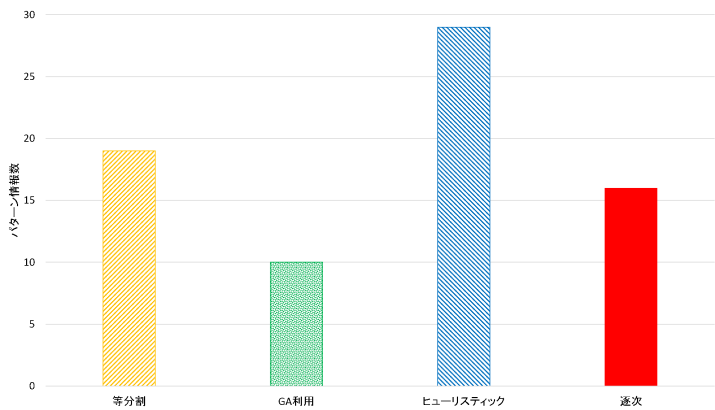


図 5.31 2重らせん識別問題におけるIDS法の最終的なパターン情報数

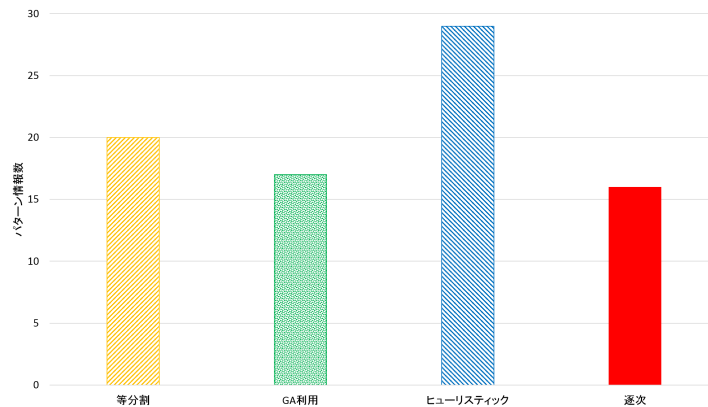


図 5.32 2 重らせん識別問題における IDS 法の最大パターン情報数

5.3.2 Pima Indians Diabetes データセット

Pima Indians Diabetes データセットの内容を表 5.3 に示す。このデータセットは 768 データからなり、8 種の入力から糖尿病か否かの 2 値を回答する問題となる。学習データはランダムに 576 データを選出し、テストデータには残りの 192 データをもちい、これを 5 セット用意した。IDS 法の解像度は 256、インクの直径は解像度の 15% となる 39 とした。等分割表現手法でもちいる分割の最大値と最小値は 3 と 1 とした。GA をもちいた分割表現手法で使用する染色体の母集団数は 40、世代数は 30 とした。また染色体長の最大値を、等分割表現手法の最大分割数と同じ分割数となるように、5 とした。逐次分割表現手法においてもちいる閾値 γ は、等分割表現手法の最大分割数と近い分割数となるように、各入力の 25% とした。各手法の最大分割数を小さくした理由は、5.1 節で示したとおり、入力数が増加すればするほど等分割表現手法や GA をもちいた分割表現手法の計算時間が増大してしまう。また作成されるパターン情報数も増大してしまい、多くの計算資源が必要となる。このため既存の分割表現手法の最大分割数を小さくし、それにあわせて逐次分割表現手法の閾値 γ も大きくした。FNN のニューロン数は 50 とし、学習前の結合強度と学習後の結合強度の差が、学習前の結合強度の 0.5% 未満となるまで学習を繰り返した。

図 5.33 に 5 試行の平均識別率を、図 5.34 に 5 試行の平均学習時間を、図 5.35 に 5 試行の平均最終的なパターン情報数を、図 5.36 に 5 試行の平均最大パターン情報数を示す。またこれらの値を付録 A.3 節に掲載する。ただし等分割表現手法において学習時間が 1 時間を越えてしまったため、学習時間の取得を 1 時間で打ち切り、識別率や最終、最大パターン情報数は別の計算機をもちいて取得した。図 5.33 より IDS 法は FNN よりも精度がよく、SVM と同程度の精度となった。また IDS 法内では精度の差はあまりない。図

表 5.3 Pima Indians Diabetes データセットの内容

データ数	入力	出力
768	妊娠回数 血漿グルコース濃度 血圧 上腕三頭筋の皮の厚さ 血清インスリン ボディマス指数 血統 年齢	糖尿病か否か

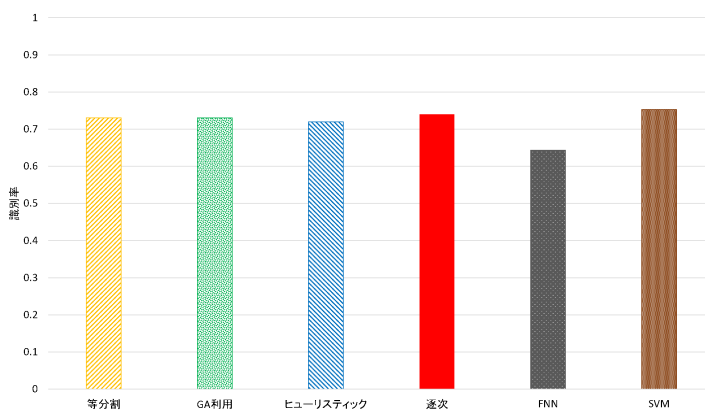


図 5.33 Pima Indians Diabetes データセットの識別率

5.34 より提案手法は既存手法よりも学習時間が短くなった。ただし FNN や SVM よりも学習時間は長くなった。これは 3.4.2 節でも記載したように、IDS 法が学習を行う際に入力数の影響を強く受けているためであり、提案手法をもちいることで抑えられるものの、他手法よりも学習時間が長くなった。図 5.35 より逐次分割表現手法のパターン情報数が最も多くなった。逐次分割表現手法において作成されるパターン情報は平均 1450 個であり、プロットされるデータの総計は 576×8 となることから、1 つのパターン情報は平均 3 データから作成されており、インクの直径を考慮すると x_i 方向にたいして 50% 以下のインクしかプロットされていないことがわかる。つまり逐次分割表現手法は入力空間を分割しすぎていえる。この理由としては、2 値分類問題であるためナロウパスの変化が激しくなり、類似度が高くならず、入力空間を統合できなかつたと考えられる。このため逐次分割表現手法は既存手法に比べて多くのパターン情報を作成したと考えられる。

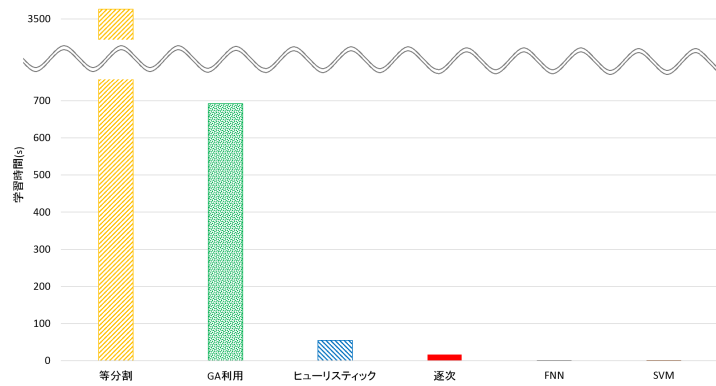


図 5.34 Pima Indians Diabetes データセットの学習時間

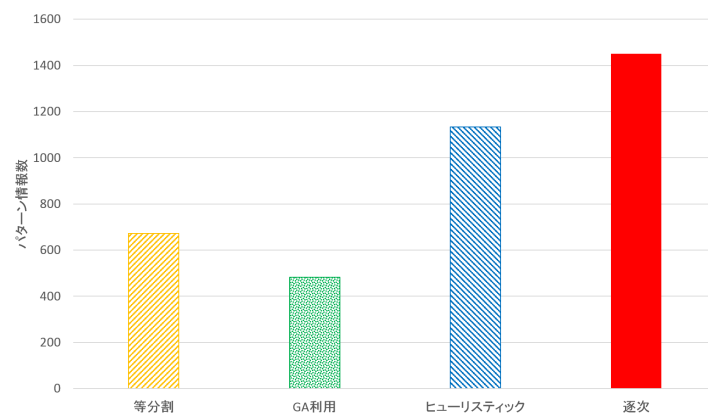


図 5.35 Pima Indians Diabetes データセットにおける IDS 法の最終的なパターン情報数

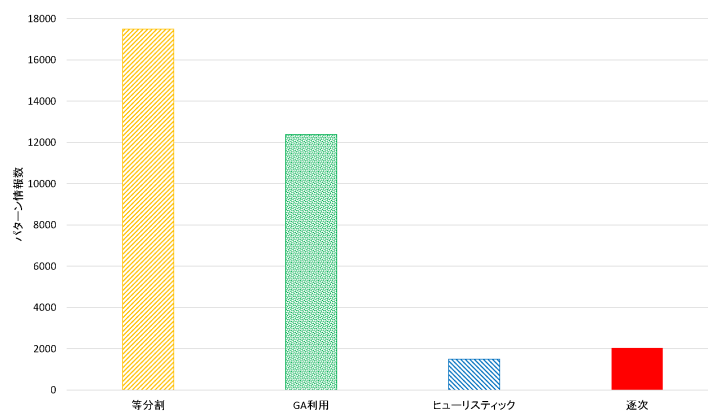


図 5.36 Pima Indians Diabetes データセットにおける IDS 法の最大パターン情報数

5.4 多値分類問題

2 値分類問題にたいし IDS 法がよく適応できることがわかったため、さらに多値分類問題へ IDS 法を適用し、分割表現手法の違いによる性能の変化を検証する。5.3.2 節の Pima Indians Diabetes データセットと同様に、UCI machine learning repository より Car Evaluation, Statlog (Shuttle), Statlog (Vehicle), Wine, Wine Quality (red), Ecoli, Glass Identification, Image Segmentation, Iris, Waveform, Yeast のデータセットを取得しもちいる。各データセットの概要を表 5.4 に示し、各データセットの詳細を付録 A.4 節に掲載する。また各データセットのクラス毎のデータのばらつきの平均と、クラス毎の最近接データまでの距離平均から各データセットを分類し、その結果を表 5.5 に示す。ただし Car Evaluation データセットは入力値が離散値であるため、この分類は行わず表 5.5 には示さない。データのばらつきには、各データセットの入力を $[0:1]$ にスケールリングし、クラス毎に重心からのユークリッド距離の平均: CD をもとめ、データ数で重み付き平均を算出したものもちいる。またデータ間の近接距離には、ばらつき同様データセットをスケールリングし、クラス毎に最近接データまでのユークリッド距離の平均: ND をもとめ、データ数で重み付き平均を算出したものもちいる。 CD は式 (5.4) から、 ND は式 (5.5) からとめた。

$$CD_c = \frac{1}{D_c} \sum_{j=1}^{D_c} \frac{\sqrt{\sum_{i=1}^N (x_{ij} - \bar{x}_i)^2}}{\sqrt{N}} \quad (5.4)$$

$$ND_c = \frac{1}{D_c} \sum_{j=1}^{D_c} \frac{\sqrt{\sum_{i=1}^N (x_{ij} - \acute{x}_{ik})^2}}{\sqrt{N}} \quad (j \neq k) \quad (5.5)$$

ここで D_c はクラス c に属するデータの数、 N は入力数、 x_{ij} はクラス c に属する j 番目のデータの i 番目の入力、 \bar{x} はクラス c に属するデータの重心、 \acute{x}_k はデータ x_j の最近接データである。表 5.5 ではばらつきが 0.10 未満となるデータセットをばらつきが小さいとし、0.10 以上のデータセットをばらつきが大きいとした。また近接距離が 0.05 未満となるデータセットを近接距離が小さいとし、0.05 以上となるデータセットを近接距離が大きいとした。ばらつきが小さい場合、データが密集していることを意味し、ばらつきが小さい全てのデータセットにおいて近接距離も小さいものとなった。ばらつきが大きくかつ近接距離が小さい場合、5.3.1 節でもちいた 2 重らせん問題はここに分類されることから、細長い、線状にデータが分布していることが予想される。ばらつきが大きくかつ近接距離が大きい場合、データが入力空間において広く拡散していることを意味する。

表 5.4 データセットの概要

	Car Evaluation	Statlog (Shuttle)	Statlog (Vehicle)	Wine	Wine Quality	Ecoli	Glass Identification	Image Segmentation	Iris	Waveform	Yeast
データ数	1728	58000	946	178	1599	336	214	2310	150	5000	1484
入力数	6	7	18	13	11	8	10	19	4	21	8

表 5.5 データセットにおけるデータのばらつきと近接距離

		ばらつき	
		小	大
近接 距離	小	Statlog (Shuttle) Ecoli, Yeast Image Segmentation	Glass Identification Iris Wine Quality
	大		Statlog (Vehicle) Wine Waveform

表 5.6 IDS 法の出力算出方法の違いによる識別率 (%) の比較

重み付き平均	重み合計値の比較
35.33	95.22

5.4.1 出力算出方法の違いによる識別率の変化

多値分類問題では、5.3 節のように閾値をもちいて IDS 法の出力を分類することはできない。これは出力が離散であり、かつ出力間に関係が無いためであり、式 (3.9) のようにナロウパスの加重平均を取ったのでは間違った値が出力されてしまう。例えば 2 つのナロウパスがあり片方はクラス 1 を、もう一方はクラス 3 を示していた場合、加重平均を取ることどちらのナロウパスも示していないクラス 2 が出力されてしまう可能性がある。そこで多値分類問題では IDS 法の出力は各クラス毎にそのクラスを示すナロウパスの重み Γ (式 (3.10)) と ω (式 (3.11)) を掛け合わせたものの合計値を比較し、最もその合計値が大きいクラスを出力とする。またナロウパスはインクの出力にたいする重心から最大濃度となる位置とし、スプレッドも出力にたいするインクの広がりではなくインクの存在するクラス数へと変更する。これも出力が離散であり、かつ出力間に関係が無いため、重心やインクの広がりをもちいると誤った値を出力する原因となる。

iris データセットを 10 分割し、1 つをテストデータとしてもちい、残りを学習データとしてもちいる 10 分割交差検定を行う。また入力空間の分割には、各入力を 3 等分に分割したのもちいている。表 5.6 に IDS 法における出力の計算方法の違いによる、iris データセットの識別率を示す。表 5.6 より、出力の計算方法の違いにより、識別率が大きく変わることがわかる。このため多値分類における IDS 法の出力には、重み合計値が最大となるクラスをもちいる。

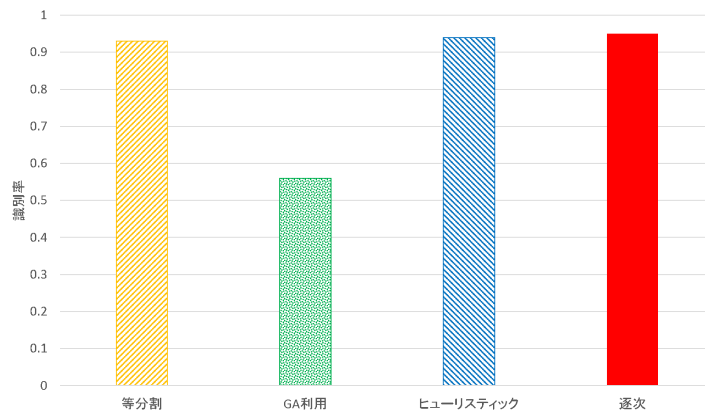


図 5.37 iris データセットの識別率

5.4.2 IDS 法における分割表現手法の比較

入力数が多いため、IDS 法の分割表現手法の違いによる性能の比較には iris データセットのみをもちいる。これは既存手法をもちいた IDS 法では入力数が多くなり、計算時間や計算資源が多くなりすぎるためである。出力の計算方法の比較と同様に、iris データセットを 10 分割し、1 つをテストデータとしてもちい、残りを学習データとしてもちいる 10 分割交差検定を行う。IDS 法の解像度は 256 とし、インクの直径は解像度の 15% となる 39 とする。等分割表現手法でもちいる分割の最大値と最小値は 10 と 1 とする。GA をもちいた分割表現手法で使用する染色体の母集団数は 40、世代数は 30 とする。また染色体長の最大値を、等分割表現手法の最大分割数と同じ分割数となるように、19 とする。逐次分割表現手法においてもちいる閾値 γ は、等分割表現手法の最大分割数に近い分割数となるように、各入力の 12.5% とする。また提案手法でもちいるナロウパスの類似度は二乗誤差ではなく一致度をもちいる。

図 5.37 に 10 試行の平均識別率を、図 5.38 に平均学習時間を、図 5.39 に最終的なパターン情報数の平均を、図 5.40 に最大パターン情報数の平均を示す。またこれらの値を付録 A.4 節に掲載する。図 5.37~5.40 より逐次分割表現手法が最もよい精度、少ないパターン情報数となる分割表現を決定できており、少ない計算資源で高速に学習できていることがわかる。このことから多値分類問題においても逐次分割表現手法が分割をよく調整し、適したパターン情報を作成できることがわかる。

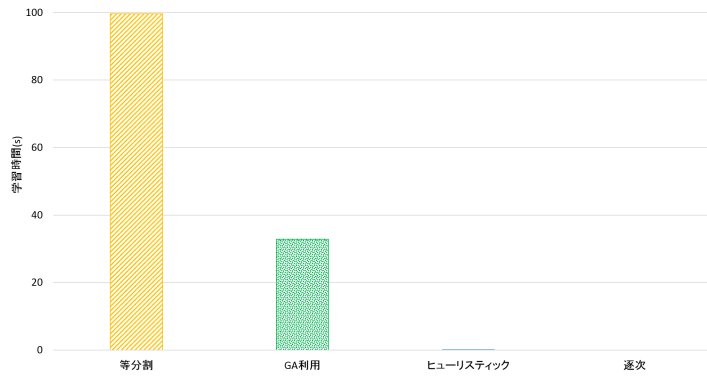


図 5.38 iris データセットにおける IDS 法の学習時間

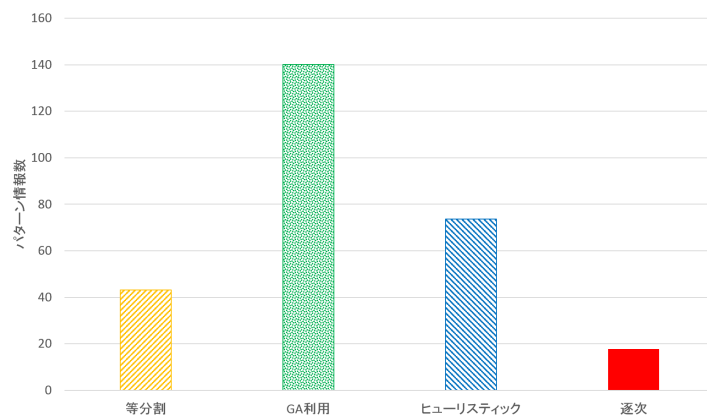


図 5.39 iris データセットにおける IDS 法の最終的なパターン情報数

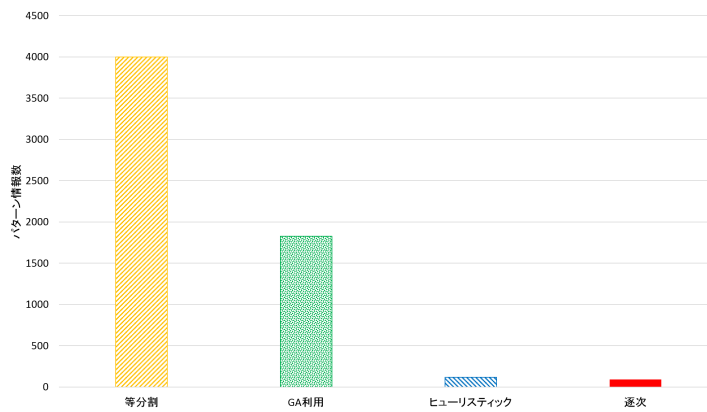


図 5.40 iris データセットにおける IDS 法の最大パターン情報数

5.4.3 IDS 法と SVM の比較

IDS 法以外の他手法として SVM との比較を行う。SVM には多値分類に対応したパッケージとして利用されている LibSVM^{*3}[17] をもちいる。LibSVM ではカーネル関数として線形，多項式，RBF，シグモイドの 4 種が用意されているため，各データセットにたいして最も識別率のよくなるカーネル関数をもちいた。また IDS 法の分割表現手法は逐次分割表現手法をもちいる。これは 5.4.2 節同様，既存手法では計算時間や計算資源が多くかかるためであり，ヒューリスティックルールをもちいた分割表現手法では入力数が増加するとよい分割表現が決定できないためである。表 5.4 に示す 10 種の多値分類問題全てにたいして IDS 法と SVM を適用する。各データセットを 10 分割し，1 つをテストデータとしてもちい，残りを学習データとしてもちいる 10 分割交差検定を行う。

図 5.41 に各手法の識別率を，図 5.42 に学習時間を示す。またこれらの値を付録 A.4 節に掲載する。図 5.41 より，識別率の差が 3% 以上となったデータセットは Statlog (Vehicle)，Wine，Wine Quality，Waveform と Glass Identification となった。これらのデータセットの中で IDS 法が勝っているのは Wine Quality と Glass Identification であり，残り 3 種のデータセットでは劣っている。この 3 種のデータセットは表 5.5 より，全てばらつきが大きくかつ近接距離の大きいデータセットである。また IDS 法が識別率で勝っている Wine Quality，Glass Identification はどちらも近接距離が小さく，ばらつきの大きいデータセットである。また大きな差のないデータセットでは，Car Evaluation を除き近接距離の小さいデータセットとなっている。このことから，IDS 法を多値分類問題に適用する場合，各クラスのばらつきではなく，各クラスにおけるデータ間の近接距離が小さい問題に適用することで，SVM と同程度かより高い識別率を示せると考えられる。これは IDS 法がインクドロップによりパターン情報を作成し，そのパターン情報からモデルを作成することに起因しており，IDS 法では与えられたデータの近傍は同様の出力を示すことを予想し，インクドロップを行うため，学習データに近いデータにたいする出力はより正確なものとなり，学習データから離れたデータにたいしては誤差が大きくなるためである。Car Evaluation では入力が離散的であり，インクドロップでデータの補完を行うことができないが，このデータセットには入力のとりえる値の全てのパターンが含まれるため，IDS 法でもよい識別率を示せたと考えられる。

また図 5.42 より，多くの問題について SVM の方が学習時間が短いことがわかる。ただしデータ数の多い Statlog (Shuttle) においては，IDS 法の方が学習時間が短くなった。これは SVM ではソフトマージンの計算等，データ数に応じて学習時間が長くなるためだと考えられる。

^{*3} <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

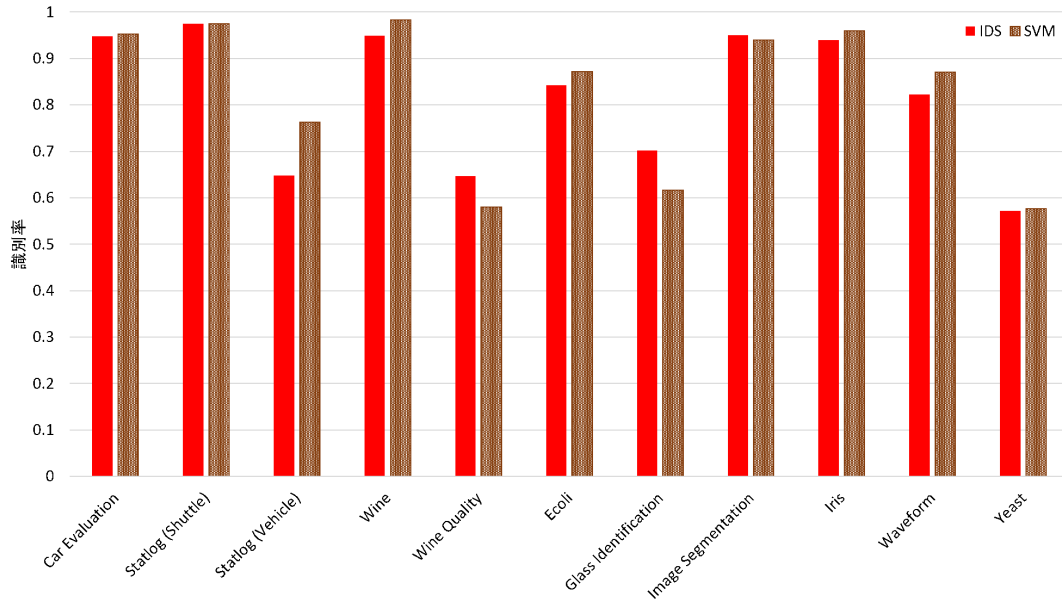


図 5.41 各データセットにおける IDS 法と SVM の識別率

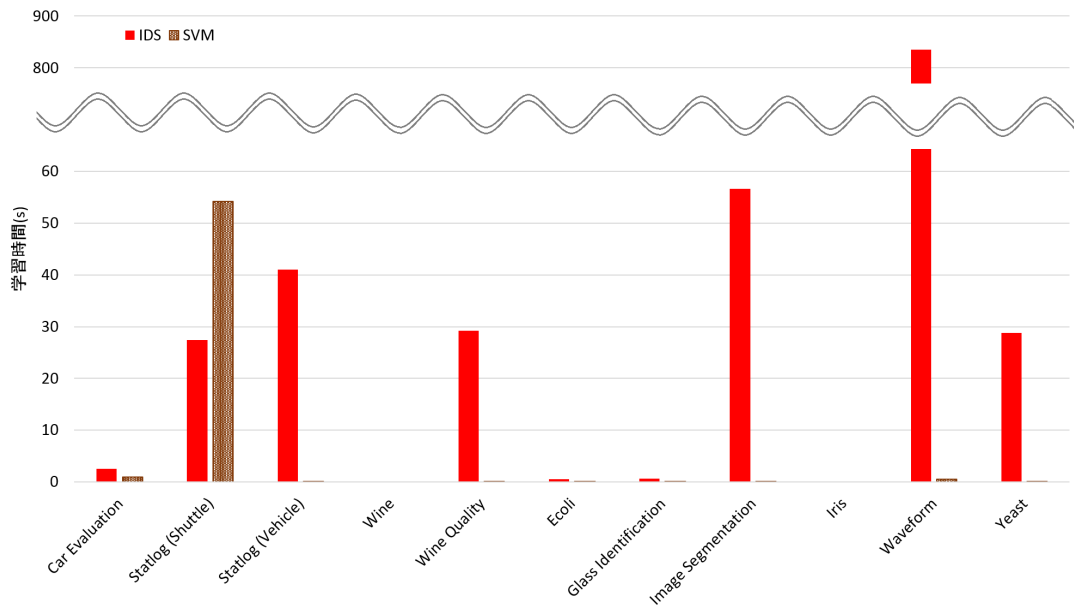


図 5.42 各データセットにおける IDS 法と SVM の学習時間

図 5.41, 5.42 より, 近接距離が小さく, ばらつきの大きいデータセットでは SVM よりも IDS 法の方が識別率が高くなり, データ数の多い問題では IDS 法の方が学習時間が短いことが分かった. 更に詳しく検証を行うために, 近接距離が小さくばらつきの大きいデータセットを作成し, 比較を行う.

2 入力 3 値分類問題として, 図 5.43 に示す問題を作成した. この問題は近接距離が小さく, ばらつきの大きいデータであり, データ数を 300, 3000, 30000 と変化させて IDS 法と SVM に適用する. 各データセットの詳細を付録 A.4 節に掲載する.

また別の問題として, 関数 (5.3) を 3 入力から n 入力とした関数 (5.6) から多値分類問題を作成する.

$$y = \prod_{i=1}^n e^{x_i} \sin(13(x_i - 0.6)^2) \quad (5.6)$$

入力 x_i を関数 (5.3) と同様に乱数を用いて作成し, 関数 (5.6) の出力 y が小さいデータ集合をクラス 1, 次に小さいデータ集合をクラス 2 と設定していき, 近接距離が小さくばらつきの大きいデータセットを作成する. こうして作成した入力数が 2, クラス数が 3 となるデータセットの例を図 5.44 に示す. この多値分類問題では入力数を 4, 5, クラス数を 3, 5, 8 と変化させてデータセットを作成し, IDS 法と SVM に適用する. 各データセットの詳細を付録 A.4 節に掲載する. 各データセットを 10 分割し, 1 つをテストデータとしてもちい, 残りを学習データとしてもちいる 10 分割交差検定を行う.

表 5.7 に 2 入力 3 値分類問題の識別率と学習時間を, 表 5.8 に関数 (5.6) から作成した分類問題の識別率を, 表 5.9 に関数 (5.6) から作成した分類問題の学習時間を示す. 各条件において最も良い値を太字で示す. また関数 (5.6) から作成した分類問題において, クラス数が増加した場合の学習時間の増加率を表 5.10 に, 入力数が増加した場合の学習時間の増加率を表 5.11 示す. 表 5.10 はクラス数が 3 だった場合の学習時間を 1 とし, クラス数が 5 や 8 の場合の学習時間の比をまとめたものであり, 表 5.11 は入力数が 4 だった場合の学習時間を 1 とし, 入力数が 5 の場合の学習時間の比をまとめたものである. 表 5.7 より, IDS 法は SVM よりも識別率が高いか同程度となった. SVM は学習データが増加すると識別率が向上した. また学習時間ではどのデータ数においても IDS 法が早い結果となった. このことから学習時間に関して, IDS 法は SVM よりもデータ数の影響を受けにくいと言える. 次に表 5.8 より, IDS 法は全ての条件において SVM よりも識別率が高い. 表 5.7 の結果も合わせて考慮すると, 近接距離が小さくばらつきの大きいデータセットにおいて, データ数が十分に存在しない場合に, IDS 法の識別率が SVM よりも高くなると考えられる. 次に 5.9 より, どの条件においても IDS 法は学習時間において SVM に劣っている. これは関数 5.6 から作成した分類問題における各クラスの境界面が複雑なためであり, 分割表現の探索を行うために多くの時間が必要となったためであ

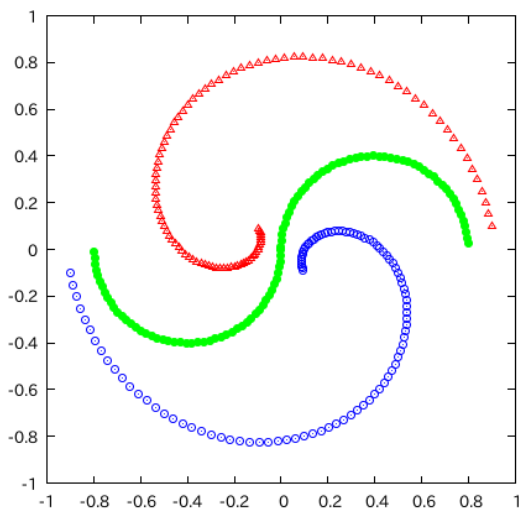


図 5.43 2 入力 3 値分類問題 (300 データ)

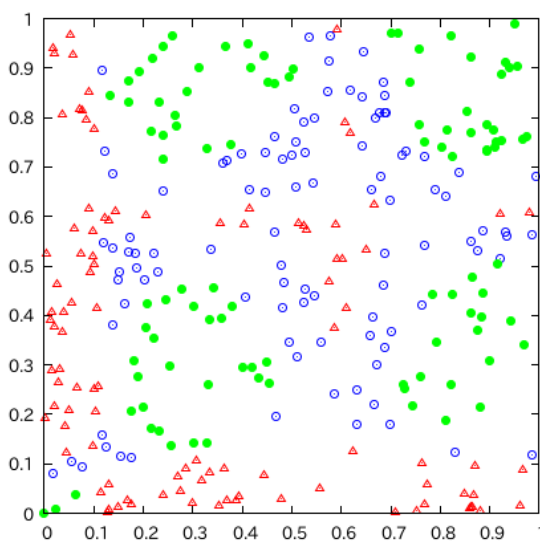


図 5.44 関数 (5.6) から作成した分類問題の例

る．また表 5.10 より，クラス数が増加した場合の学習時間の増加率は IDS 法と SVM の間に大きな差はないが，表 5.11 より，入力数が増加した場合の学習時間の増加率は IDS 法の方が大きい．このことから学習時間に関して，IDS 法は SVM よりも入力数の影響を受けやすいと言える．

表 5.7 2 入力 3 値分類問題の比較

		300	3000	30000
識別率 (%)	IDS	99.67	100.00	100.00
	SVM	73.67	95.90	100.00
学習 時間 (s)	IDS	0.00	0.01	0.08
	SVM	0.01	0.40	17.99

表 5.8 関数 (5.6) から作成した分類問題の識別率 (%)

			クラス数		
			3	5	8
入力数	4	IDS	80.17	60.97	47.05
		SVM	66.53	44.89	33.05
	5	IDS	76.25	60.89	47.42
		SVM	65.59	46.16	34.14

表 5.9 関数 (5.6) から作成した分類問題の学習時間 (s)

			クラス数		
			3	5	8
入力数	4	IDS	1.71	3.14	4.99
		SVM	0.67	1.21	2.07
	5	IDS	217.41	572.22	1064.01
		SVM	17.32	40.56	68.44

表 5.10 クラス数の増加による学習時間の増加率

			クラス数		
			3	5	8
入力数	4	IDS	1.00	1.84	2.92
		SVM	1.00	1.81	3.09
	5	IDS	1.00	2.63	4.89
		SVM	1.00	2.34	3.95

表 5.11 入力数の増加による学習時間の増加率

		IDS		SVM	
		入力数		入力数	
		4	5	4	5
クラス数	3	1.00	127.14	1.00	25.85
	5	1.00	182.24	1.00	33.52
	8	1.00	213.23	1.00	33.06

第 6 章

総合考察

前章では IDS 法における既存の分割表現手法と、提案する分割表現手法の比較を行った。また IDS 法以外の他手法である FNN と SVM とも比較を行った。その結果を表 6.1 と表 6.2 にまとめて記載する。表 6.1 では差が平均の $\pm 5\%$ 未満ならば同率とし、最良が複数ある場合は次点を記載しない。表 6.2 では識別率の差が $\pm 3\%$ 未満ならば同程度とした。

提案手法の 1 つであるヒューリスティックルールをもちいた分割表現手法は、分割位置を調整でき、精度を維持してパターン情報数を減らし、学習時間や計算資源の使用量を減らすことに成功した。これは 5.1.1 節の 2 入力関数近似問題において、近似精度が等分割手法と同程度によく、かつ最終的なパターン情報を等分割手法よりも減少させ、学習時間を既存手法よりも短くし、最大パターン情報数を最も少なくしたことからわかる。またデータ数や平面解像度から手法内で使用するパラメータを自動的に設定できるため、他の手法では最大パターン情報数の問題から、最大分割数を常に大きくはできないといった問題を回避し、適切な分割数を自動的に設定できる。これは 5.3.1 節の 2 重らせん識別問題の結果からも言える。ただし入力数が増えると 1 つの分割が複数のパターン情報に影響を与えてしまうため、分割の調整能力が低下する。これは 5.2 節の 3 入力関数近似において、IDS 法内で最も近似精度が悪くなっていることや、5.3.2 節や 5.4.2 節において、識別率は他の分割表現手法と同程度となっているが、最終的なパターン情報数が既存手法よりも多くなっていることからわかる。

もう 1 つの提案手法である逐次分割表現手法は、徐々に分割を増やし、新しく作成されたパターン情報のみ作成することで高速に、かつ計算資源の使用量を抑えて分割表現を決定できる。これは 5.1.1、5.2、5.4.2 節の各問題において、他の分割表現手法と同程度かよりよい精度を示し、最大パターン情報数や学習時間を既存手法よりも少なくしていることからわかる。また複雑な入力空間を分割、生成することができ、精度を維持してパターン情報数を減らすことができる。これは 5.2 節の 3 入力関数近似や 5.4.2 節の Iris

データセットの分類において、既存手法よりも少ないパターン情報数で最も精度のよくなる分割表現を決定できていることからわかる。ただし、分割位置を入力空間を 2 等分する位置でしか設定できないため、2 入力の対象システムではヒューリスティックルールをもちいた分割表現手法に劣る。これは 5.1.1 節の 2 入力関数近似において、ヒューリスティックルールをもちいた分割表現手法の方が最終的なパターン情報数が小さいことからわかる。更に出力の変化が激しい問題の場合、ナロウパスの類似度が低下し、入力空間の統合が行えず、パターン情報数を減らせない場合がある。これは 5.3.2 節の Pima Indians Diabetes データセットにおいて、逐次分割表現手法の最終的なパターン情報数が最も多くなっていることからわかる。

また 5.1.2 節にて、IDS 法でもちいるメンバーシップ関数の違いによる精度の変化を検証した。その結果出力の変化が緩やかな問題については、ナロウパスどうしの影響を考慮できるようにメンバーシップ関数を設定することで精度がよくなり、出力の変化が複雑な問題では、ナロウパスどうしの影響が狭くなるようにメンバーシップ関数を設定することで精度がよくなることを示した。これは変化が緩やかな問題では複数のナロウパスから出力を求めることでより正確に出力を決定できるためであり、変化が複雑な問題ではナロウパスどうしの差が大きく、複数のナロウパスから出力を求めるよりも特定のナロウパスから出力を求めることで、正確に出力を決定できる。

次に 5.4.1 節にて、IDS 法を多値分類問題に適用するために新しい出力算出方法を提案した。従来の重み付き平均をもちいた出力算出方法では、ナロウパスが示していないクラスを出力してしまい識別率が低下していたが、重み合計値の最も大きいクラスを出力とすることで、よい識別率となることを示した。

IDS 法と、他手法である FNN や SVM との比較では、5.1.1 節や 5.2 節において、変化の複雑な関数近似問題では FNN よりも精度がよく、提案手法をもちいた IDS 法は学習時間が FNN よりも短くなることを示した。次に 5.3 節において 2 値分類問題でも IDS 法が FNN より識別率がよくなることを示し、SVM と同程度かよりよい精度となることを示した。ただし IDS 法は入力数が増加することで作成されるパターン情報数が増えてしまうため、5.3.2 節の Pima Indians Diabetes データセットでは、IDS 法は FNN や SVM と比較して学習時間が長くなった。5.4.3 節にて IDS 法と SVM を多値分類問題に適用し比較を行い、IDS 法が同じクラスとなるデータ間の近接距離が小さい問題において、SVM と同程度かよりよい精度となることを示した。また学習時間に関して、IDS 法は SVM よりもデータ数の影響を受けにくく、入力数の影響を強く受けることを示した。

表 6.1 比較結果のまとめ

○:最良 △:次点 ×劣る -:比較なし

			IDS 法				他手法	
			既存手法		提案手法			
			等分割	GA 利用	ヒューリ スティック	逐次 分割	FNN	SVM
5.1.1 節 2 入力 関数	精度	関数 g_1, g_2	△	×	△	△	○	-
		関数 $g_3 \sim g_5$	○	×	○	○	×	-
	学習時間		×	×	△	○	×	-
	最終パターン情報数		×	○	△	×	-	-
	最大パターン情報数		×	×	○	△	-	-
5.2 節 3 入力 関数	精度	関数 (5.2)	×	△	×	△	○	-
		関数 (5.3)	△	×	×	○	×	-
	学習時間		×	×	△	○	×	-
	最終パターン情報数		×	×	○	△	-	-
	最大パターン情報数		×	×	○	○	-	-
5.3.1 節 らせん 識別問題	識別率		△	×	○	△	×	△
	学習時間		×	×	○	○	×	○
	最終パターン情報数		×	○	×	△	-	-
	最大パターン情報数		×	○	×	○	-	-
5.3.2 節 Pima Indian Diabetes	識別率		○	○	○	○	×	○
	学習時間		×	×	×	×	△	○
	最終パターン情報数		△	○	×	×	-	-
	最大パターン情報数		×	×	○	△	-	-
5.4.2 節 多値 分類 IDS 法	識別率		○	×	○	○	-	-
	学習時間		×	×	△	○	-	-
	最終パターン情報数		△	×	×	○	-	-
	最大パターン情報数		×	×	△	○	-	-

表 6.2 多値分類での IDS 法と SVM の比較

○:優 △:同程度 × 劣

			IDS	SVM
識別率	ばらつき:大	近接距離:大	×	○
	ばらつき:大	近接距離:小	○	×
	ばらつき:小	近接距離:小	△	△
学習時間	入力数:大		×	○
	データ数:大		○	×

第 7 章

結論

7.1 本論文のまとめ

本論文では、IDS 法の分割表現手法としてヒューリスティックルールをもちいた分割表現手法と、逐次分割表現手法を提案し、その有効性を示した。また多値分類問題に逐次分割表現手法をもちいた IDS 法を適用し、多値分類問題でも IDS 法が有効であることを示した。

5 章にて提案手法と既存手法を用いた IDS 法を関数近似問題や分類問題に適用し、性能の比較を行った。ヒューリスティックルールをもちいた分割表現手法は、分割位置を調整でき、精度を維持して最終的なパターン情報数を減らすことに成功し、既存手法よりも適切な分割表現を決定できる。更に学習時間や必要な計算資源を既存手法よりも減らすことに成功した。またデータ数や平面解像度から手法内で使用するパラメータを自動的に設定できるため、適切な分割数を自動的に設定できる。ただし入力数が増えると 1 つの分割が複数のパターン情報に影響を与えてしまうため、分割の調整能力が低下し、精度の悪化やパターン情報数の増加を招く。このため 2 入力の対象システムへの利用が適している。もう 1 つの提案手法である逐次分割表現手法は、徐々に分割を増やし、新しく作成されたパターン情報のみ作成することで既存手法よりも使用する計算資源を減らし、高速に分割表現を決定できる。また入力数が多い場合、複雑な入力空間を分割、生成することができるため、精度を維持してパターン情報数を減らすことができる。しかし、分割位置を入力空間を 2 等分する位置でしか設定できないため、2 入力の対象システムではヒューリスティックルールをもちいた分割表現手法に劣る。このため 3 入力以上の対象システムへの利用に適している。また、出力の変化が激しい問題の場合、ナロウパスの類似度が低下し、入力空間の統合が行えずパターン情報数を減らせない場合がある。このため問題に応じて類似度の計算方法を変更する必要がある。

5.1.2 節にて、IDS 法でもちいるメンバーシップ関数が増えると、精度が変化するこ

とを示した。出力の変化が緩やかな問題については、複数のナロウパスどうしの影響を考慮できるようにメンバーシップ関数を設定することで精度がよくなり、出力の変化が複雑な問題では、逆にナロウパスどうしの影響範囲が狭くなるようにメンバーシップ関数を設定することで精度がよくなることを示した。

5.4.1 節にて、IDS 法を多値分類問題に適用するために、新しい出力算出方法を提案した。従来の重み付き平均をもちいた出力算出方法ではナロウパスが示していないクラスを出力してしまい正答率が低下していたが、重み合計値が最も大きくなるクラスを出力とすることで、よい正答率となることを示した。

また5章ではIDS法と他手法との比較も行い、FNNとの比較では2入力や3入力の関数近似問題に適用し、IDS法が出力の変化が複雑な近似対象においてFNNよりもよい精度となることを示し、既存手法をもちいたIDS法では学習時間で劣っていたが、提案手法をもちいたIDS法では学習時間でも勝ることを示した。また2値分類問題でもIDS法とFNN、SVMの比較を行い、IDS法がFNNよりも正答率が高くなり、SVMと同程度かよりよい正答率となることを示した。ただし入力数が増加すると、学習時間では他手法に劣る。次に逐次分割手法をもちいたIDS法とSVMを多値分類問題において比較し、IDS法が同じクラスとなる最近接データ間の距離が小さい問題において、SVMと同程度かよりよい精度となることを示し、データ数が多く、入力数の少ない問題ではIDS法の学習時間がSVMの学習時間よりも短くなることを示した。

7.2 課題と今後の展望

4章にて、新しいIDS法の分割表現手法を提案したが、まだ改善すべき課題も残されている。5章にて、ヒューリスティックルールを用いた分割表現手法では、入力数が増加した場合に分割を適切に設定できないことが分かっている。逐次分割表現手法では分割位置を、対象となる空間を2等分にする位置でしか設定しておらず、より適切な分割位置を設定できる可能性がある。

IDS法の精度向上のために、分割領域の非線形化が挙げられる。提案手法では分割領域が矩形となっており、非線形化を行うことでより適切な分割表現を設定できる可能性がある。また現在利用しているメンバーシップ関数では出力を計算する際に、各ナロウパスから線形補間が行われてしまうため、FNNのようななめらかな曲線を再現できない。また学習データを離散化された平面に射影するため、平面の解像度を十分大きくしなければなめらかな曲線を描けない。しかし解像度を大きくすればそれだけ必要な計算資源が必要になってしまう。このためメンバーシップ関数の調整手法や、平面解像度をパターン情報毎に調整する手法の開発が必要である。

次にIDS法をより大きな入力数の問題に対応させるために、出力計算の高速化や、入力

次元の削減が挙げられる．IDS 法の出力計算には 1 つのルール毎に入力数 -1 のメンバーシップ関数が必要となり，多くの計算資源が必要となる．また入力が多い対象システムにおいては全ての入力毎にパターン情報を作成する必要性が低くなることが予想される．このため事前に学習データの入力次元を削減したり，一部の入力にたいしてのみパターン情報を作成する手法が必要である．

このような課題を改善し，IDS 法をより精度よく，より扱いやすい手法として確立することで，様々な問題に適用させていきたい．特に本論文において IDS 法が多値分類問題にたいしてもよい精度を示すことが分かったため，より多くの問題に適用させていきたい．

謝辞

本研究を博士論文としてまとめる機会をいただくとともに、終始適切なご指導を賜りました、内海 彰 教授に心より深謝申し上げます。

本研究を進めるにあたって多くのご助言をいただきました本多 中二 名誉教授、西野 順二 助教に深く感謝いたします。

本論文の審査過程において、数々のご助言とご指導を賜りました、板倉 直明 教授、由良 憲二 教授、高玉 圭樹 教授、本多 中二 名誉教授に深謝申し上げます。

最後に、本研究を進めるにあたって多くのご助言を下さいました内海研究室の皆様感謝いたします。

参考文献

- [1] 林勲, 古橋式. ファジィ・ニューラルネットワーク. 朝倉書店, 1996.
- [2] H.Sagha, S.B.Shouraki, H.Beigy, H.Khasteh, and E.Enayati. Genetic ink drop spread. *Proceedings of Second International Symposium, on Intelligent Information Tecnology Application, IEEE computer society*, pp. 603–607, 2008.
- [3] 伊庭斉志. 知の科学 進化論的計算手法. オーム社, 2005.
- [4] J.-N.Hwang, S.-R.Lay, and I-C.Jou. The cascade-correlation learning: a projection pursuit learning perspective. *IEEE Transsactions on Neural Networks, Vol.7, No. 2*, pp. 278–289, 1996.
- [5] J.-N.Hwang, S.-R.Lay, M.Maechler, R.D.Martin, and J.Schimert. Regression modeling in back-propagation and projection pursuit learning. *IEEE Transsactions on Neural Networks, Vol.5, No. 3*, pp. 342–353, 1994.
- [6] J.A.K.Suykens and J.Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters, Vol.9, Issue 3*, pp. 293–300, 1999.
- [7] J.Liska and S.S.Melsheimer. Complete design of fuzzy logic systems using genetic algorithms. *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference, Vol.2*, pp. 1377–1382, 1994.
- [8] J.W.Smith, J.E.Everhart, W.C.Dickson, W.C.Knowler, and R.S.Johannes. Using the adap learning algorithm to forecast the onset of diabetes mellitus. *Proceedings of the Symposium on Computer Applications and Medical Care, IEEE Computer Society Press*, pp. 261–265, 1988.
- [9] K.J.Lang and M.J.Witbrock. Learning to tell two spirals apart. *Proceedings of the 1988 Connectionist Models Summer School*, pp. 52–59, 1988.
- [10] Tin-Yau Kwok and Dit-Yan Yeung. Objective functions for training new hidden units in constructive neural networks. *IEEE Transsactions on Neural Networks, Vol.8*, pp. 1131–1148, 1997.
- [11] L.A.Zadeh. Soft computing and fuzzy logic. *IEEE Software, Vol.11, No. 6*, pp.

- 48–56, 1994.
- [12] L.Ma and K.Khorasani. Constructive feedforward neural networks using hermite polynomial activation functions. *IEEE Transactions on Neural Networks*, Vol.16, pp. 821–833, 2005.
- [13] 増田達也, 夜久正司. ユニット生成機能をもつニューラルネットワークによるファジィ推論ルールの獲得手法. 日本ファジィ学会誌, Vol.5, No. 2, pp. 174–183, 1993.
- [14] M.Murakami and N.Honda. A fast structural optimization technique for ids modeling. *2007 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS 2007)*, pp. 204–209, 2007.
- [15] M.Murakami and N.Honda. A study on the modeling ability of the ids method : A soft computing technique using pattern-based information processing. *International Journal of Approximate Reasoning*, Vol.45, No. 3, pp. 470–487, 2007.
- [16] M.Murakami and N.Honda. Performance of the ids method as a soft computing tool. *IEEE Transactions on Fuzzy Systems*, Vol.16, No. 6, pp. 1582–1596, 2008.
- [17] R.Fan, P.Chen, and C.Lin. Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, vol.6, pp. 1889–1918, 2005.
- [18] R.Reed. Pruning algorithms - a survey. *IEEE Transactions on Neural Networks*, vol.4, pp. 740–747, 1993.
- [19] 櫻井義尚, 本多中二. パターン情報に基づく能動的学習法による非線形システムのモデリングと制御への適用. 日本知能情報ファジィ学会誌, Vol.17, No. 3, pp. 335–366, 2005.
- [20] S.B.Shouraki and N.Honda. Recursive fuzzy modeling based on fuzzy interpolation. *Journal of Advanced Computational Intelligence*, Vol.3, No. 2, pp. 114–125, 1999.

付録 A

各評価実験における結果の補足

5章における精度，や学習時間の測定を行うにあたり，下記の性能の計算機を使用した．

CPU : *Intel(R)Core(TM)2DuoCPU3.0GHz*
メモリー : 4GB

A.1 2 入力関数近似

5.1 節に記載した図 5.2~5.6 の値を表 A.1 に，5.7~5.9 の値を表 A.2~A.4 に記載する．また 5.1 節に記載しなかった等分割表現手法，GA をもちいた分割表現手法，ヒューリスティックルールをもちいた分割表現手法による IDS 法の出力例を図 A.1~A.3 に記載する．

表 A.1 Hwang の 5 関数の近似精度:FVU(10^{-2})

	g_1	g_2	g_3	g_4	g_5
等分割	1.35	2.02	12.51	5.01	5.08
GA 利用	2.12	3.55	16.33	6.54	7.23
ヒューリスティック	1.07	2.14	13.44	5.03	5.68
逐次	1.19	2.06	12.63	5.02	5.43
FNN	0.02	0.81	104.26	8.12	69.21

表 A.2 Hwang の 5 関数の学習時間 (s)

	g_1	g_2	g_3	g_4	g_5
等分割	0.74	0.79	0.79	0.75	0.77
GA 利用	5.97	6.00	5.74	5.67	5.62
ヒューリスティック	0.05	0.11	0.08	0.10	0.06
逐次	0.03	0.03	0.03	0.03	0.03
FNN	1.27	0.36	0.07	0.34	0.26

表 A.3 Hwang の 5 関数における IDS 法の最終的なパターン情報数

	g_1	g_2	g_3	g_4	g_5
等分割	18.08	18.58	15.55	17.86	17.63
GA 利用	15.28	13.46	13.80	13.38	13.38
ヒューリスティック	12.95	15.21	14.27	15.23	14.53
逐次	15.32	15.86	15.46	15.55	14.85

表 A.4 Hwang の 5 関数における IDS 法の最大パターン情報数

	g_1	g_2	g_3	g_4	g_5
等分割	20.00	20.00	20.00	20.00	20.00
GA 利用	18.81	18.82	18.81	18.70	18.79
ヒューリスティック	13.34	14.61	14.46	14.79	14.92
逐次	15.92	16.00	15.96	16.00	15.18

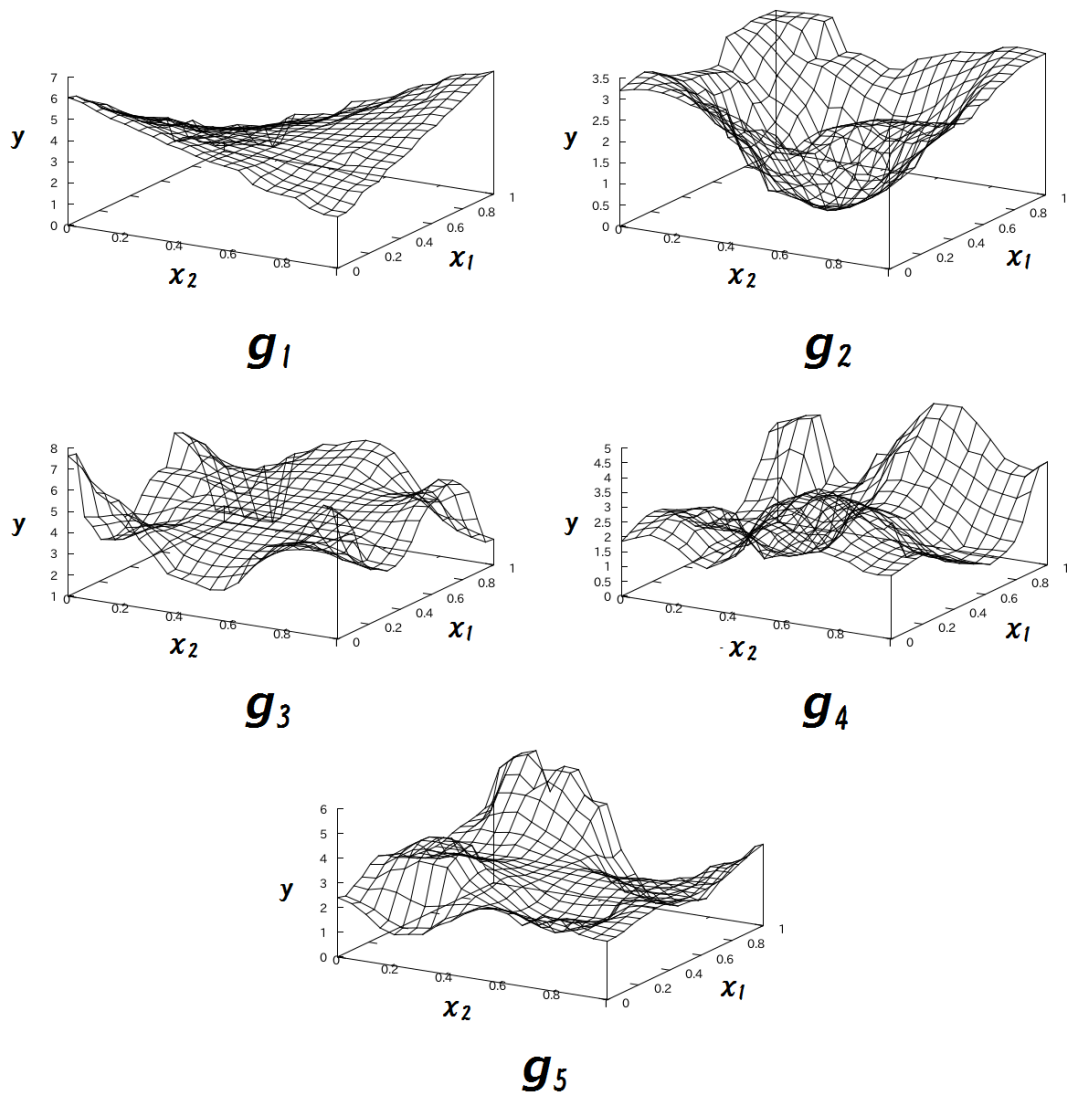


図 A.1 Hwang の 5 関数における等分割表現手法による IDS 法の出力例

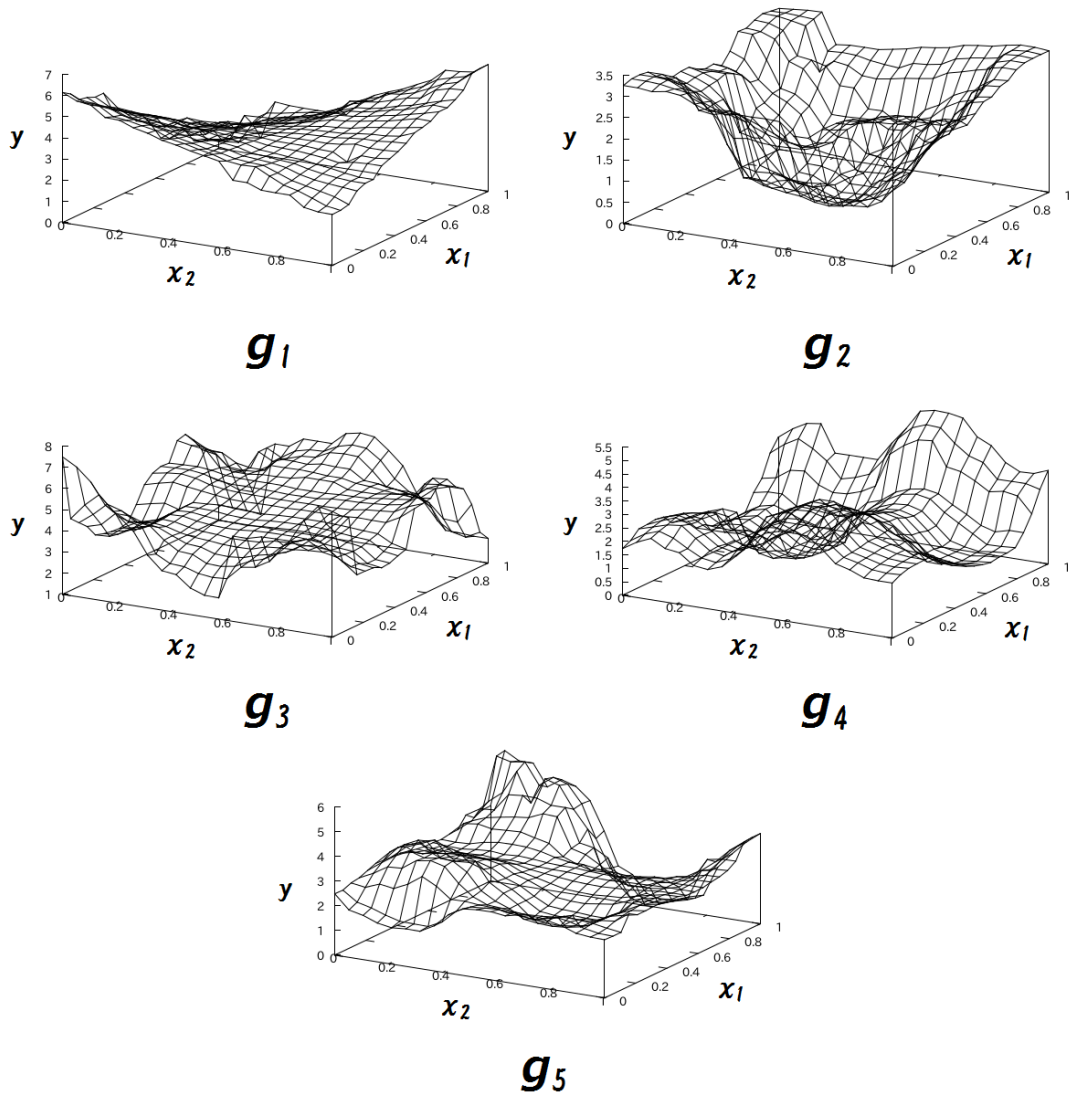


図 A.2 Hwang の 5 関数における GA をもちいた分割表現手法による IDS 法の出力例

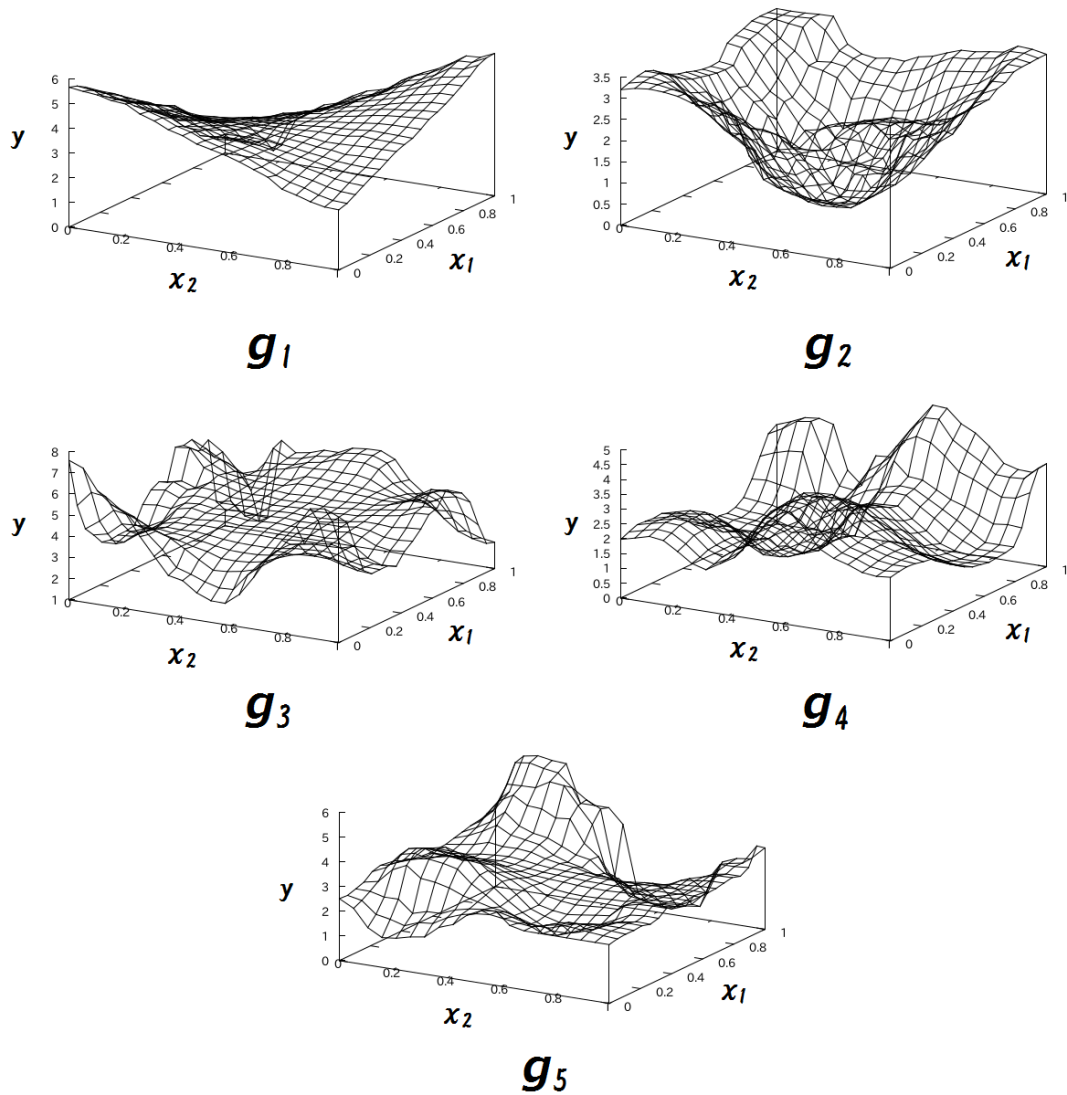


図 A.3 Hwang の 5 関数におけるヒューリスティックルールをもちいた分割表現手法による IDS 法の出力例

A.2 3 入力関数近似

5.2 節に記載した図 5.15~5.22 の値を表 A.5~A.12 に記載する。

表 A.5 関数 (5.2) の近似精度:FVU(10^{-2})

	500	1000	1500
等分割	7.42	2.97	2.00
GA 利用	3.76	2.40	1.83
ヒューリスティック	6.74	3.28	2.63
逐次	3.44	2.41	2.12
FNN	2.15	0.25	0.16

表 A.6 関数 (5.3) の近似精度:FVU(10^{-2})

	500	1000	1500
等分割	29.58	15.03	10.20
GA 利用	39.44	22.72	15.51
ヒューリスティック	68.30	55.23	36.13
逐次	23.72	12.53	9.44
FNN	80.21	74.90	70.72

表 A.7 関数 (5.2) の学習時間 (s)

	500	1000	1500
等分割	33.96	44.48	56.56
GA 利用	63.09	91.14	114.90
ヒューリスティック	0.28	0.32	0.38
逐次	0.28	0.37	0.45
FNN	4.19	8.22	11.14

表 A.8 関数 (5.3) の学習時間 (s)

	500	1000	1500
等分割	32.57	42.59	52.73
GA 利用	58.80	89.88	116.62
ヒューリスティック	0.88	1.18	1.29
逐次	0.20	0.28	0.35
FNN	0.75	1.14	0.40

表 A.9 関数 (5.2) における IDS 法の最終的なパターン情報数

	500	1000	1500
等分割	258.39	248.46	255.17
GA 利用	91.21	115.41	136.17
ヒューリスティック	40.11	56.31	62.37
逐次	56.53	105.54	145.38

表 A.10 関数 (5.3) における IDS 法の最終的なパターン情報数

	500	1000	1500
等分割	110.40	152.45	197.60
GA 利用	68.29	102.10	129.26
ヒューリスティック	41.68	44.40	67.82
逐次	49.23	81.80	105.34

表 A.11 関数 (5.2) における IDS 法の最大パターン情報数

	500	1000	1500
等分割	300.00	300.00	300.00
GA 利用	237.99	249.40	252.88
ヒューリスティック	52.29	61.59	63.96
逐次	140.96	153.59	159.97

表 A.12 関数 (5.3) における IDS 法の最大パターン情報数

	500	1000	1500
等分割	300.00	300.00	300.00
GA 利用	229.98	246.27	253.66
ヒューリスティック	173.91	184.10	188.49
逐次	96.63	109.01	114.11

A.3 2値分類問題

5.3.1 節に記載した図 5.29~5.32 の値を表 A.13~A.16 に記載する．また 5.3.2 節に記載した図 5.33~5.36 の値を表 A.17~A.20 に記載する．

表 A.13 2重らせん識別問題の識別率 (%)

等分割	GA 利用	ヒューリスティック	逐次	FNN	SVM
86.86	70.88	97.94	85.05	39.94	83.33

表 A.14 2重らせん識別問題の学習時間 (s)

等分割	GA 利用	ヒューリスティック	逐次	FNN	SVM
0.04	7.50	0.00	0.00	0.01	0.00

表 A.15 2重らせん識別問題における IDS 法の最終的なパターン情報数

等分割	GA 利用	ヒューリスティック	逐次
19	10	29	16

表 A.16 2重らせん識別問題における IDS 法の最大パターン情報数

等分割	GA 利用	ヒューリスティック	逐次
20	17	29	16

表 A.17 Pima Indians Diabetes データセットの識別率 (%)

等分割	GA 利用	ヒューリスティック	逐次	FNN	SVM
72.71	72.68	72.16	73.71	64.27	75.31

表 A.18 Pima Indians Diabetes データセットの学習時間

等分割	GA 利用	ヒューリスティック	逐次	FNN	SVM
3600.00	692.20	54.778	16.826	0.23	0.02

表 A.19 Pima Indians Diabetes データセットにおける IDS 法の最終的なパターン情報数

等分割	GA 利用	ヒューリスティック	逐次
672.80	483.20	1134.00	1450.80

表 A.20 Pima Indians Diabetes データセットにおける IDS 法の最大パターン情報数

等分割	GA 利用	ヒューリスティック	逐次
17496.00	12393.00	1490.40	2043.60

A.4 多値分類問題

5.4 節でもちいる 10 種のデータセットの詳細を表 A.21~A.31 に記載する。また 5.4.2 節に記載した図 5.37~5.40 の値を表 A.32~A.33 に記載し、5.4.3 節に記載した図 5.41、5.42 の値を表 A.36、A.37 に記載する。また 2 入力 3 値分類問題の各データセットの詳細を表 A.38~A.40 に、関数 (5.6) から作成した分類問題の各データセットの詳細を A.41~A.46 に記載する。

表 A.21 Car Evaluation データセット

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
1728	6	カテゴリー	4	1210	38.67	13.61
				384	32.59	13.61
				69	27.47	13.61
				65	23.12	13.61

表 A.22 Statlog (Shuttle) データセット

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
58000	9	整数	7	45586	3.44	0.01
				50	4.97	0.31
				171	7.53	0.22
				8903	3.70	0.01
				3267	4.97	0.04
				10	6.24	3.30
				13	7.13	1.31

表 A.23 Statlog (Vehicle) データセット

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
946	18	整数	4	240	18.74	6.18
				240	18.59	6.24
				240	16.83	4.78
				226	13.71	5.32

表 A.24 Wine データセット

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
178	13	整数, 実数	3	59	11.79	8.94
				71	16.07	11.89
				48	13.39	9.57

表 A.25 Wine Quality (red) データセット

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
1599	11	実数	6	10	13.27	10.73
				53	12.22	8.25
				681	11.60	3.14
				638	12.25	3.34
				199	12.07	3.56
				18	12.65	8.35

表 A.26 Ecoli データセット

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
336	8	実数	8	143	8.62	3.56
				77	11.83	5.85
				2	10.73	21.45
				2	23.01	46.02
				35	8.92	6.04
				20	9.79	7.46
				5	6.32	6.38
				52	8.79	4.57

表 A.27 Glass Identification データセット

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
214	10	実数	6	70	8.13	2.69
				76	13.06	4.75
				17	8.10	4.59
				13	20.06	9.80
				9	12.62	9.45
				29	12.80	7.32

表 A.28 Image Segmentation データセット

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
2310	19	実数	7	330	8.45	1.05
				330	9.21	1.38
				330	11.37	2.19
				330	10.88	2.61
				330	11.00	1.71
				330	8.92	1.17
				330	8.45	2.16

表 A.29 Iris データセット

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
150	4	実数	3	50	8.25	2.98
				50	10.15	3.79
				50	11.86	4.51

表 A.30 Waveform データセット

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
5000	21	実数	3	1657	13.89	8.99
				1647	12.47	8.81
				1696	12.47	8.86

表 A.31 Yeast データセット

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
1484	8	実数	10	463	8.14	2.72
				429	9.05	3.67
				244	8.88	3.64
				51	9.92	5.35
				163	7.51	3.61
				44	6.96	3.83
				37	6.92	3.70
				30	9.25	5.14
				20	18.53	6.37
				5	6.54	7.67

表 A.32 Iris データセットにおける IDS 法の識別率 (%)

等分割	GA 利用	ヒューリスティック	逐次
93.00	56.00	94.00	95.00

表 A.33 Iris データセットにおける IDS 法の学習時間 (s)

等分割	GA 利用	ヒューリスティック	逐次
99.62	32.79	0.02	0.01

表 A.34 Iris データセットにおける IDS 法の最終的なパターン情報数

等分割	GA 利用	ヒューリスティック	逐次
43.10	140.20	73.70	17.70

表 A.35 Iris データセットにおける IDS 法の最大パターン情報数

等分割	GA 利用	ヒューリスティック	逐次
4000.00	1829.80	118.80	94.60

表 A.36 各データセットにおける IDS 法と SVM の識別率 (%)

	Car Evaluation	Statlog (Shuttle)	Statlog (Vehicle)	Wine	Wine Quality	Ecoli	Glass Identification	Image Segmentation	Iris	Waveform	Yeast
IDS	94.73	97.48	64.78	94.93	64.73	84.24	70.13	95.02	94.00	82.20	57.14
SVM	95.20	97.49	76.24	98.31	58.04	87.20	61.68	93.94	96.00	87.04	57.68

表 A.37 各データセットにおける IDS 法と SVM の学習時間 (s)

	Car Evaluation	Statlog (Shuttle)	Statlog (Vehicle)	Wine	Wine Quality	Ecoli	Glass Identification	Image Segmentation	Iris	Waveform	Yeast
IDS	2.52	27.42	40.98	0.15	29.18	0.49	0.65	56.66	0.01	834.96	28.75
SVM	0.94	54.19	0.05	0.00	0.14	0.01	0.01	0.09	0.00	0.53	0.09

表 A.38 2入力3値分類問題 (データ数 300)

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
300	2	実数	3	100	20.85	1.19
				100	20.85	1.19
				100	20.27	1.01

表 A.39 2入力3値分類問題 (データ数 3000)

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
3000	2	実数	3	1000	20.68	0.12
				1000	20.68	0.12
				1000	20.59	0.10

表 A.40 2入力3値分類問題 (データ数 30000)

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
30000	2	実数	3	10000	20.67	0.01
				10000	20.67	0.01
				10000	20.62	0.01

表 A.41 関数 (5.6) から作成した分類問題 (4入力3クラス)

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
3000	4	実数	3	1000	30.46	4.73
				1000	23.35	4.65
				1000	29.14	4.74

表 A.42 関数 (5.6) から作成した分類問題 (4 入力 5 クラス)

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
3500	4	実数	5	700	32.17	4.73
				700	25.86	4.99
				700	23.58	5.02
				700	26.64	5.27
				700	30.08	4.50

表 A.43 関数 (5.6) から作成した分類問題 (4 入力 8 クラス)

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
4400	4	実数	8	550	33.67	4.60
				550	29.30	4.94
				550	24.92	5.03
				550	22.06	5.16
				550	24.63	5.35
				550	26.17	5.48
				550	28.35	5.07
				550	30.50	4.35

表 A.44 関数 (5.6) から作成した分類問題 (5 入力 3 クラス)

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
15000	5	実数	3	5000	30.61	4.96
				5000	24.23	4.87
				5000	29.01	4.90

表 A.45 関数 (5.6) から作成した分類問題 (5 入力 5 クラス)

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
20000	5	実数	5	4000	32.26	4.85
				4000	26.57	4.94
				4000	24.20	4.95
				4000	26.73	5.10
				4000	30.14	4.70

表 A.46 関数 (5.6) から作成した分類問題 (5 入力 8 クラス)

データ数	入力数	入力の種類	クラス数	クラス毎 のデータ数	CD(10^{-2})	ND(10^{-2})
50000	5	実数	8	6250	33.31	3.98
				6250	29.60	4.28
				6250	22.84	4.18
				6250	25.32	4.22
				6250	24.63	4.46
				6250	26.17	4.53
				6250	28.35	4.29
				6250	30.50	3.86

関連論文

関連論文の印刷公表の方法及び時期

1. 全著者名： 尾崎 新斗，本多 中二，内海 彰
論文題目： ヒューリスティックルールを用いたインクドロップスプレッド法の研究
印刷公表の方法及び時期： 日本知能ファジィ学会誌，Vol.25，No.5，pp.865-879，
2013年10月
(第4章，第5章に関連)
2. 全著者名： Yoshito Ozaki and Akira Utsumi
論文題目： A new partitioning method for the IDS method
印刷公表の方法及び時期： Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, pp.3927-3932, 2013年10月
(第4章，第5章に関連)