# 非線形判別分析におけるブースティング手法の改良

理学専攻・情報科学コース 2040640 川戸翔子

## 1　はじめに

判別分析とは, いくつかのグループに分かれているデータを基に, それらが「どういう基準で分けられているのか」という関係を解析することで, 分類されていないサンプルがどのようにグループに属するかを予測する手法である.

本研究では, 判別分析手法の一つである AdaBoost.M1 を改良し, ノイズに耐えるロバストな判別境界を求める手法を考察する.

## 2　AdaBoost.M1

ブースティングとは, 大量の弱学習器を順番に学習し, 組み合わせることで学習を強化する手法で, 前の学習器が誤分類したデータを優先的に正しく分類できるように学習していく.

AdaBoost.M1 とは, ブースティングの最も代表的な手法である.

AdaBoost.M1 のアルゴリズムは以下の通りである.

1. データの重みを初期化. $w_i = 1/N, i = 1, 2, \cdots, N$.

2. $m = 1$ から $m = M$ まで以下を繰り返す.

(a) 重み $w_i$ を使い, 弱学習器 $G_m(x)$ でトレーニングデータを学習する.

(b) エラー率 $\mathrm{err}_m$ を計算する.

$$\mathrm{err}_m = \frac{\sum_{i=1}^{N} w_i \, I\big(y \neq G_m(x_i)\big)}{\sum_{i=1}^{N} w_i}$$

(c) 弱学習器の重ね合わせの重み $\alpha_m$ を計算する.

$$\alpha_m = \log\left(\frac{1 - \mathrm{err}_m}{\mathrm{err}_m}\right)$$

(d)$w_i$ を更新する.

$$w_i \leftarrow w_i \cdot \exp\Big(\alpha_m \, I\big(y_i \neq G_m(x_i)\big)\Big), \; i = 1, 2, \ldots, N.$$

3. 最終的な学習器 $G(x)$

$$G(x) = \mathrm{sign}\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right)$$

を出力.

ここで, $N$ は与えられたデータの数, $M$ は弱学習器の数, $y$ はクラスラベル 1 または -1, $G_m$ は予測子変数を与えられると 1 または -1 をとる予測を生成する弱学習器, $w_i$ はデータへの重み, 関数 $I$ は

$$I = \begin{cases} 1 & \text{if } y \neq G_m(x), \\ 0 & \text{if } y = G_m(x). \end{cases}$$

この AdaBoost.M1 には, ノイズに弱いという弱点がある.

## 3　シグモイド関数を用いた改良

AdaBoost.M1 がノイズに弱い理由は, 間違った観測の重みを, 次のラウンドで集中して学習させるように重くするため. そこで, エラー率を求める時と, 重みを更新する時に使われている 2 値関数 $I$ を改良することを提案する.

関数 $I$ は弱学習器の出力とクラスラベルが合っているかどうかしか考慮していない. この関数 $I$ を合っている確率を表す関数, シグモイド関数 $f(x) = \frac{1}{1+e^{-\beta x}}$ に変更して実験を行った.

### 3.1　実験概要

図 1 のような各クラス 200 個ずつ 400 個のノイズのあるデータと, 弱学習器を 800 個用意し, 図 1 のような判別境界を描けるか, 2 値関数を使用した場合とシグモイド関数を使用した場合とで比較する.

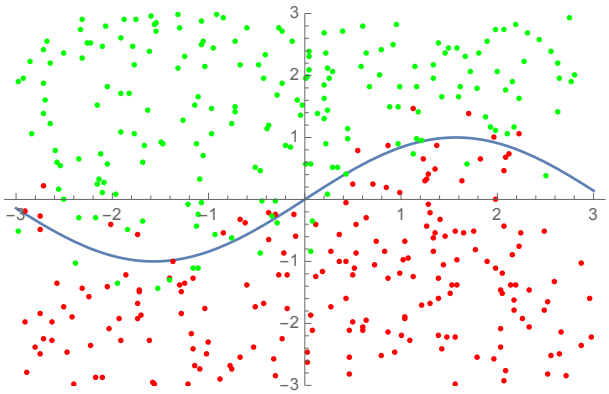



図 1

### 3.2　実験結果

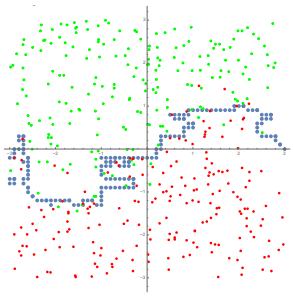


図 2：2 値関数

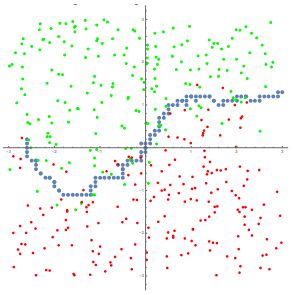


図 3：シグモイド関数

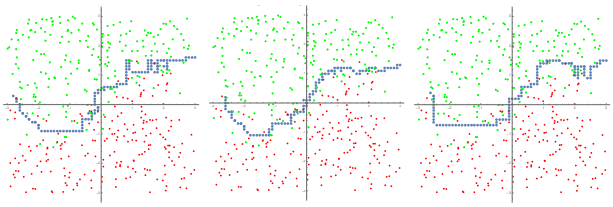
図 2 と図 3 を比較すると, シグモイド関数を用いたときの方が滑らかな境界を描けることが分かる.

## 4　重み, パラメータの調整

弱学習器の重み $\alpha$ とシグモイド関数のパラメータ $\beta$ の値が, 滑らかな境界を引くことにどう影響するか調

べるために数値実験を行った.

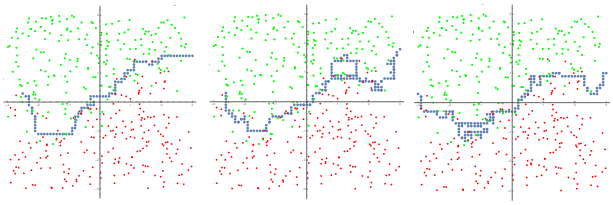
$\alpha$ は, ある閾値よりも絶対値が小さければ 0 にするトランケーションを行う. そのトランケーション処理を行う時の閾値を, 0.01, 0.025, 0.05 と変化させた結果が下図である.



0.01　　0.025　　0.05

上図より, $\alpha$ はの絶対値が 0.025 より小さいものを 0 とした時が最も滑らかな境界を描けることが分かる. しかし, $\alpha$ のトランケーションは, 閾値の最適値が, 与えるサンプルデータによって毎回変化する可能性があるため, 毎回比較する必要がある.

さらに, シグモイド関数のパラメータ $\beta$ を, 1, 5, 10 と変化させて比較した. 結果は下図の通りである.



$\beta = 1$　　$\beta = 5$　　$\beta = 10$

上図より, $\beta$ の値が 1 の時に最も滑らかな境界を描けることが分かる. $\beta$ の値を大きくしていくと, シグモイド関数はステップ関数に近づいていくため, ノイズの影響を受けやすくなる.

# 5　多クラスへの拡張

2 クラスの判別分析手法である AdaBoost.M1 を, 多クラスの判別分析にも対応させる改良を考察した.

## 5.1　改良手法の概要

まず一つのクラスについて, これまでの 2 クラスの判別分析と同様に, データがそのクラスに属するか属さないかを判別し, その処理をそれぞれのクラスについて行うことで, 多クラスの判別分析を達成する.

## 5.2　改良手法のシミュレーション

サンプルデータとして, 図 4 のような, 各クラス 100 個ずつ 4 クラスに別れている合計 400 個のデータを用意した. 弱学習器の数は 800 個とした.

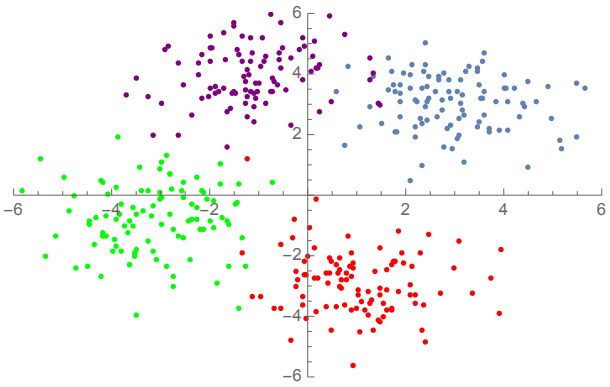



図 4

シミュレーションの結果は図 5 で, 4 クラスのデータについて判別境界を引くことができた.

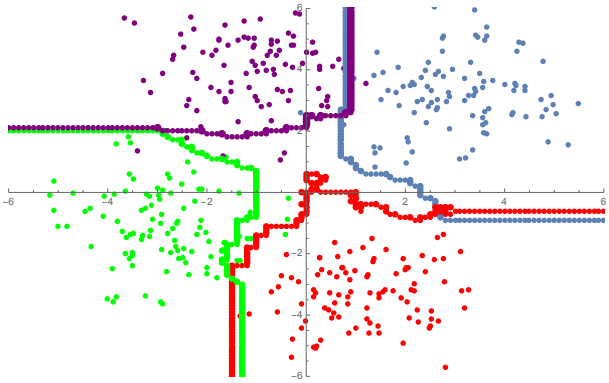


図 5

# 6　まとめ

AdaBoost.M1 のノイズに弱いという弱点は, 2 値関数の代わりにシグモイド関数を用いることで改善できることがわかった. 弱学習器の重みやシグモイド関数のパラメータを調整することも, 滑らかな境界を描くことに影響することがわかった. 最適なパラメータの調整方法に関しては今後の課題となる.

また, 1 つのクラスについて, データがそのクラスに属するか属さないかを判別し, その処理をそれぞれのクラスについて行うことで, AdaBoost.M1 を多クラスの判別分析に拡張することができた.

### 参考文献


[1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, New York, NY, (2017)

[2] 佐野夏樹, 鈴木秀男, AdaBoost のロバスト化, 日本オペレーションズ・リサーチ学会, 春季研究発表会,1-F-8 (2003)

[3] 「判別分析とは」https://www.intage.co.jp/glossary/057/