2003年日本オペレーションズ・リサーチ学会
春季研究発表会

AdaBoostのロバスト化

02203190　筑波大学　*佐野夏樹　Natsuki Sano
筑波大学　鈴木秀男　Hideo Suzuki

1　はじめに

ブースティング (Boosting) は逐次的に学習データの分布を変化させ，その度に弱学習機（学習アルゴリズム）をあてはめ，これらの弱学習機を組み合わせて精度の高い学習機械を構成する手法である．代表的なものとしてAdaBoost, Bagging 等が挙げられデータマイニングにおいてニューラルネットワークや決定木等の性能を強化するために用いられる．次にそのアルゴリズムを示す．

2　AdaBoost

AdaBoost M1(Freund and Schapire[3])

1. Initialize observation weights $w_i = 1/N, i = 1, \ldots, N$
2. Repeat m =1,2, ... ,M
   (a) Fit a classifier $f_m(x) \in \{-1, 1\}$ using w_i
   (b) Compute $\mathrm{err}_m = E_w[I_{(y \neq f(x))}]$
   (c) Compute $c_m = \log((1 - \mathrm{err}_m)/\mathrm{err}_m)$
   (d) $w_i \leftarrow w_i \exp[c_m \cdot \mathrm{I}_{y_i \neq f_m(x_i)}]$, $i = 1, 2, \ldots N$ and normalize so that $\sum_i w_i = 1$
3. Output $F(x) = sign[\sum_{m=1}^{M} c_m f_m(x)]$

ここで学習データは (y_i, x_i) $i = 1, \cdots, N$ であり $y \in \{-1, 1\}$ の２値判別問題を考える．観測値に対する重みは１回目のラウンドにおいては一様に $1/N$ となっている．この観測値に対してなんらかの弱学習機があてはめられるのであるが，その学習において正解した観測値に対しては次のラウンドにおけるその観測値に対する重みを小さく，誤った観測値に対しては重みを重くするのがBoosting の特徴である．また毎ラウンドあてはめられた弱学習機に対しても信頼度 c_m が与えれる．最終的な判別は $\sum_{m=1}^{M} c_m f_m(x)$ に sign を取ることで行われる．

3　損失関数の提案

AdaBoost はシンプルで強力な手法であるがミスラベルのようなノイズに対して強い影響を受けることが知られている [2]．ミスラベルのある学習データに AdaBoost を適用すると弱学習機はラベルが間違った例題に対する重みを指数的に増加させるので AdaBoost M1 2(c) においてラベルの間違った例題に特化した弱学習機を構成するようになり，そのような学習機に対してもある程度の信頼度を与えてしまう．結果として，最終的に出力される学習機もラベルの間違った例題に大きく影響を受けテストデータに対する誤り率が増加してしまう．これを損失関数の観点から見ると，AdaBoost が指数関数を損失関数としている [5] ことに起因している．図１にそのグラフを示す．x 軸は正規化されてはいないが，マージンと呼ばれ分類の難しさを表す量である．マージンがマイナスということは多くの弱学習機が判別を誤ったことになり，判別の難しいデータということになる．しかしながら，このような観測値に対して極端に大きなペナルティーをかけて良いのだろうか?データ収集者がミスラベルを行った可能性もある．そこで本研究はより緩やかな関数としてＹ軸対称なシグモイド関数 $1/\exp(1 + \lambda y F(x))$ を損失関数として提案する．この関数は 0,1 損失関数の近似となっており，この期待損失を最小化することはベイズルールの近似となる．

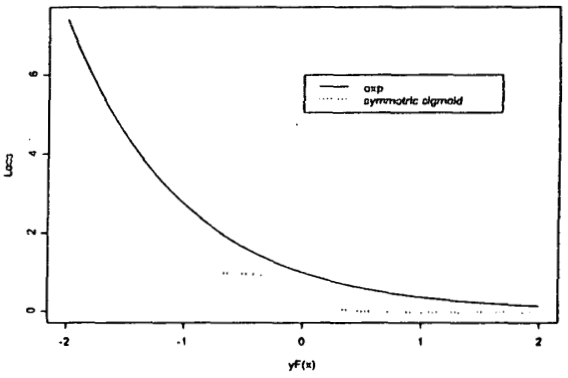

図 1: AdaBoost の損失関数と提案する損失関数

4 提案するBoosting

本研究では次のアルゴリズムを提案する.

1. $F_0(x) = \bar{y}$
2. Repeat m =1,2, ... ,M
   (a) $\tilde{y}_i = \frac{\lambda y_i e^{\lambda y_i F(x_i)}}{(1+e^{\lambda y_i F(x_i)})^2}$ for $i = 1, \cdots, N$
   $at\ F(x_i) = F_{m-1}(x_i)$
   (b) $\arg\min_\beta \sum_{i=1}^{N} [\tilde{y}_i - x_i\beta]^2$
   (c) $F_m(x_i) = F_{m-1}(x_i) + x_i\beta_m$
3. Output $F(x) = sign[F_m(x)]$

AdaBoostが最急降下法によって損失関数の最小化を行っているとみなす研究はいくつかあるが(例えば[4]), 本研究も同様に3節において提案した損失関数を最急降下法によって最小化することによりアルゴリズムの導出を行った. 2(a)において提案する損失関数のgradient $\tilde{y} = \lambda y e^{\lambda y F(x)}/(1 + e^{\lambda y F(x)})^2$ を求めている. このgradientは学習データ点においてのみ定義されているので2(b)(c)においてこれを最小2乗法によって補間している. その予測値を足し合わせることで最終的な判別を行う.

5 応用例

UCI Machine Learning Repository[1] の2値データGerman Credit Dataに対して提案アルゴリズムを適用した. AdaBoostの弱学習機械には深さ3の決定木を用いた. 学習データの数は300でテストデータの数は700とした. 100回のラウンドによる学習誤り率とテスト誤り率の変化を図2に示す. AdaBoostと提案アルゴリズムのテスト誤り率の比較を図3に示す. 本研究で提案するアルゴリズムは学習誤り率についてはAdaBoostよりも劣っていたが, テストデータにおいては安定した結果を示しており, 概ねAdaBoostよりも優れていることがわかる (図3).

6 おわりに

今後の方針としてシミュレーションによりノイズの量の変化に対するテスト誤り率の変化を見る必要がある. また今回のアルゴリズムは各弱学習機に対する信頼度を一様にし, Baggingの一種のような形になっているが, より良い重みの決定法について検討する必要がある.

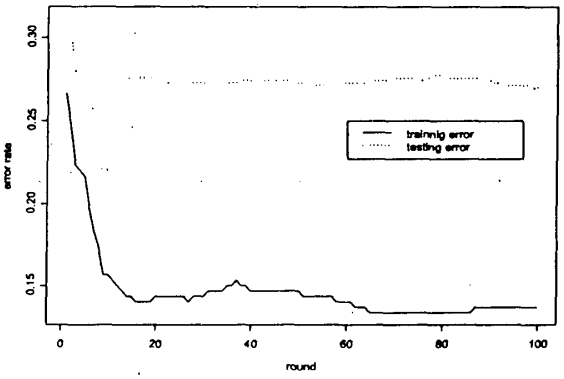

図 2: 提案するアルゴリズムの誤り率

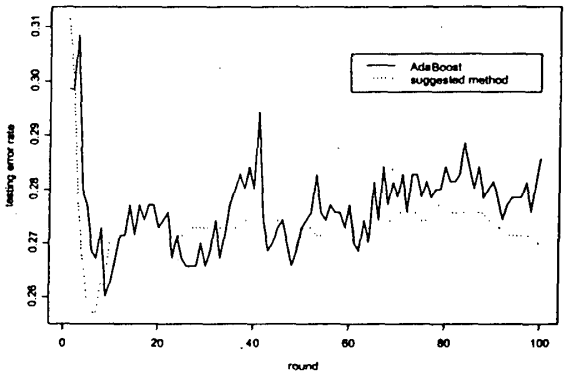

図 3: AdaBoostと提案するアルゴリズムのテスト誤り率の比較

参考文献

[1] http://www.ics.uci.edu/ mlearn/MLRepository.html.

[2] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms:baging, boosting, and variants. *Machine Learning*, 35:105–139, 1998.

[3] Y. Freund and R. E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

[4] J Friedman. Greedy function approximation: A gradient boosting machine. *Annals of statistics*, 29, 2001.

[5] T. Friedman, J. Hastie and R. Tibishirani. Additive logistic regression: A statistical view of boosting. *Annals of statistics*, 28:337-407, 2000.