

Support Vector Machine を用いた AdaBoost の重み最適化

松川 徹[†] 栗田 多喜夫[‡]

[†] 筑波大学大学院システム情報工学研究科 〒305-8573 つくば市天王台 1-1-1

[‡] 産業技術総合研究所 〒305-8568 つくば市梅園 1-1-1

E-mail: [†] t.matsukawa@aist.go.jp, [‡] takio-kurita@aist.go.jp

あらまし AdaBoost のアルゴリズムでは、弱識別器を各ラウンドに1つずつ順に学習する。弱識別器の重みもラウンド順に決まるが、一度決定された重みは変化しない。これらの重みは全ての弱識別器が学習された後に全弱識別器の結果を用いて最適化出来る。本稿では、ソフトマージン線形 SVM を用いて AdaBoost の重みを最適化する2つの手法を提案する。1つは全弱識別器が構築された後に1回最適化を行う手法であり、もう1つは、AdaBoost の全学習ラウンドで最適化を行う手法である。7つのベンチマークデータセットに対して評価実験を行い、提案手法が AdaBoost と同等以上の識別性能を持つことを示した。

キーワード AdaBoost, サポートベクターマシン, 重み最適化

Weight Optimization of AdaBoost using Support Vector Machine

Tetsu MATSUKAWA[†] Takio KURITA[‡]

[†] Graduate School of Systems and Information Engineering, University of Tsukuba

1-1-1 Tennodai, Tsukuba-shi, Ibaraki, 305-8573 Japan

[‡] National Institute of Advanced Industrial Science and Technology(AIST)

1-1-1 AIST Central-2, Umezono, Tsukuba-shi, Ibaraki, 305-8568 Japan

E-mail: [†] t.matsukawa@aist.go.jp, [‡] takio-kurita@aist.go.jp

Abstract The AdaBoost algorithm trains weak classifiers sequentially and the weights of each classifier are also determined sequentially. But once the weight of weak classifier is calculated, it is not changed. These weights can be further optimized using classification results of all weak classifiers after all the weak classifiers were constructed. This paper proposes the weight optimization method of the AdaBoost algorithm using soft-margin linear SVM. There are two possibilities to apply the weight optimization. One is to apply the optimization once after all weak classifiers are constructed. The other is to apply the optimization at every step of boosting. Experiments on 7 commonly used benchmark datasets show that the proposed optimization methods can give better or comparable error rates compared to the original AdaBoost.

Keyword AdaBoost, Support Vector Machine, weight optimization

1. まえがき

ブースティングは、多数の精度の低い識別器を組み合わせて非常に精度の高い識別器を作るアンサンブル学習方式である。AdaBoost のようなブースティングアルゴリズムは、現実の識別問題に適用され、高い識別性能が示されている[4]~[8]。特に Freund と Schapire により開発された AdaBoost(Adaptive Boosting)は、顔や歩行者の検出やテキストの分類等に用いられ、画像・文書識別分野で多くの研究者によって着目されている。

AdaBoost のアルゴリズムでは、弱識別器を各学習ラウンドに1つずつ順に学習する。ここで、弱識別器の学

習には、前回のラウンドで学習された弱識別器の結果に応じて重みづけされたデータを用いている。特に、前回の学習ラウンドで誤認識したデータに対して大きな重みが与えられる。全ての弱識別器を学習したら、強識別器を全弱識別器の出力の重み付き線形結合で構築する。ここで、各々の識別器の重みは、各弱識別器の性能に応じて決定している。

しかし、AdaBoost のアルゴリズムでは一度決定された弱識別器の重みは変化しない。Additive に最適化された重みより、全体の識別結果を用いて最適化された重みの方が有利であると考えられるため、各弱識別器の重みは全ての弱識別器が構築された後に全弱識別器

の出力を用いて最適化する余地がある。本稿では、ソフトマージン線形 SVM を用いて AdaBoost の重みを最適化する 2 つの手法を提案する。1 つは全弱識別器が構築された後に 1 回最適化を行う手法 (AdaBoost+SVM.final) であり、もう 1 つは、AdaBoost の全学習ラウンドで最適化を行う手法 (AdaBoost+SVM.seq) である。

7 つのベンチマークデータセットに対して評価実験を行い、提案手法が AdaBoost に比べ同等以上の識別性能を持つことを示す。特に AdaBoost+SVM.seq は AdaBoost+SVM.final よりも高い識別性能を示した。

2. AdaBoost

ブースティングは 1986 年に Schapire により初めて導入され[1]、その後 Freund による改良[2]を経て、1995 年に Freund と Schapire によって AdaBoost とよばれるアルゴリズムが提案された[3]。AdaBoost のアルゴリズムの疑似コードを図 1 に示す。

ここで、2 クラスの分類問題を考える。学習データは特徴ベクトル $\mathbf{x}_1, \dots, \mathbf{x}_N$ とクラスラベル $\{t_1, \dots, t_N\}$ からなる。ここで、クラスラベルは $t_n \in \{-1, 1\}$ とする。それぞれのデータは $1/N$ に初期化されている重み w_n を持っている。ここで重み付きデータに対する弱識別関数 $y(\mathbf{x}) \in \{-1, 1\}$ を持っていることを仮定する。オリジナルの AdaBoost では重み付きデータの学習を行うために resample が用いられていたが、4. の実験では reweighting を用いた。これは reweighting の方が resample より良い結果が得られることが実験的に示されているからである[6]。

各ラウンドで、AdaBoost は各ラウンドまでの誤認識率から決定された重み付きデータを用いて新しい弱識別器を学習する。最終的に必要な数の弱識別器が学習されたら、各弱識別器の線形結合として強識別器が構築される。

AdaBoost のアルゴリズムは指数損失関数を用いた Forward stagewise additive modeling と解釈できる[10]。従って、AdaBoost アルゴリズムでは一度重み α_m が決定されれば、それらの重みは変化しない。よって、これらの重みは全ての弱識別器が構築された後に最適化する余地がある。本稿ではソフトマージン線形 SVM を用いた AdaBoost の重み最適化手法を提案する。

3. サポートベクターマシンを用いた重み最適化

本章では、提案する重み最適化手法を述べる。

AdaBoost の最終モデルはパーセプトロンの形をしている。SVM(サポートベクターマシン)はマージン最大化の基準により高い汎化性能を持つ識別器である。

Input: training set $\mathbf{x}_1, \dots, \mathbf{x}_N$ $\{t_1, \dots, t_N\}$
 where $t_n \in \{-1, 1\}$

Initialize: the data weights $\{w_n\}$ by setting
 $w_n^1 = 1/N$ for $n = 1, \dots, N$

Do for $m = 1, \dots, M$:

(a) Train a weak classifier $y_m(\mathbf{x}) \in \{-1, 1\}$ to the training data using weights $\{w_n^m\}$

(b) Evaluate the quantities

$$\varepsilon_m = \left\{ \sum_{n=1}^N w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n) \right\} / \left\{ \sum_{n=1}^N w_n^{(m)} \right\}$$

and then use these to evaluate

$$\alpha_m = \ln\left\{ (1 - \varepsilon_m) / \varepsilon_m \right\}$$

(c) Update the data weights

$$w_n^{(m+1)} = w_n^{(m)} \exp\left\{ \alpha_m I(y_m(\mathbf{x}_n) \neq t_n) \right\}$$

and normalize so that $\sum_{n=1}^N w_n^{(m)} = 1$

Output: the final model

$$Y_M(\mathbf{x}) = \text{sign}\left\{ \sum_{m=1}^M \alpha_m y_m(\mathbf{x}) \right\}$$

図 1. The boosting algorithm AdaBoost

SVM の内、線形 SVM は SVM の基本形でありパーセプトロンの学習法である[11]。線形 SVM の識別関数は次式により定義される。

$$f(\mathbf{x}) = \sum_{m=1}^M \alpha_m y_m(\mathbf{x}) + b$$

データが線形分離不可能な場合に対応するために、ソフトマージンとよばれる手法が提案されている。誤識別の許容度を許すためのスラッグ変数 ξ_i を導入して、重み $\{\alpha_m\}$ は次の最適化問題を解くことにより求められる。

$$\begin{aligned} \min_{\alpha, b, \xi} \quad & \frac{1}{2} \alpha^T \alpha + C \sum_{i=1}^M \xi_i \\ \text{subject to} \quad & t_i (\alpha^T y_i(\mathbf{x}) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad i = 1, \dots, M \end{aligned}$$

C は誤識別項のペナルティパラメータである。パラメータ C は Cross Validation により決定する。

本稿ではソフトマージン線形 SVM を用いた 2 つの最適化手法を提案する。1 つの最適化手法は AdaBoost のアルゴリズムにより全ての弱識別器が生成された後に、ソフトマージン線形 SVM を用いて重み $\{\alpha_m\}$ を最適化する手法である。これを AdaBoost+SVM.final と呼

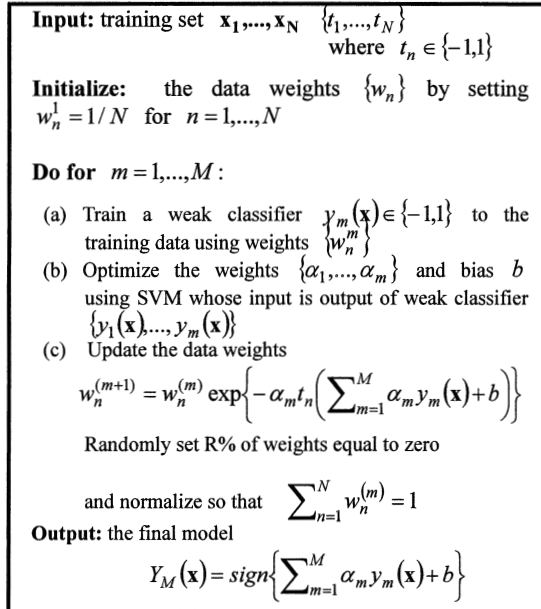


図 2. The sequential weight optimized AdaBoost algorithm AdaBoost+SVM.seq

ぶ。もう一つの最適化手法は各学習ラウンドで SVM を用いて重み $\{\alpha_m\}$ を逐次最適化する手法であり、これを AdaBoost+SVM.seq と呼ぶ。AdaBoost+SVM.seq の疑似コードを図 2 に示す。

図 2 において学習データに対する重み更新則はオリジナルの AdaBoost の指数損失関数より導出されている。アンサンブル学習において弱識別器の多様性が重要であることが知られている。また、SVM で学習された新しい識別器の重みが 0 に近くなり、その他の識別器の重みが全ラウンドと同じになると、次のラウンドにおいても同じ識別器が生成される場合がある。そのように同じ識別器が構築されると、学習が停止してしまう。このような状況を回避するためにデータのリサンプリングを行う。すなわち、各ラウンドにおいて全学習データの内 (100-R) % のデータを用いて弱識別器を学習する。一方で、弱識別器の重みを最適化するには全てのデータを用いる。

バイアス項 b を除いて提案手法の最終モデルは Ada-

Boost と同じ形となっている。従って、SVM の学習を繰り返すことにより AdaBoost と比べ学習時の計算コストは増大するが、識別時の計算コストは AdaBoost と変わらない。

4. 実験

4.1 実験の設定

本章では、7 つのベンチマークデータセットを用いて行った実験について述べる。データセットの特性を表 1 に示す。データセットは全て 2 値分類問題である。6 つのデータセット (breast-cancer, ionosphere, german, pima, sonar, liver) は UCI Machine Learning Repository[14]から取得した。また、fourclass[13]の内、欠損値のあるデータは除外した。全てのデータセットはランダムに学習データとテストデータに分類した。

弱識別器にはブースティングの実験[4]-[8]でよく用いられる Quinlan らの決定木 C4.5[12]の内、深さ 1 のもの (decision stump) を用いた。C4.5 のアルゴリズムは入力ベクトルから情報量ゲイン比を基準として分割に最適な属性 (次元) を選択する。

実装に際して、オリジナルの C4.5 は重み付きデータをサポートしていないので、情報量ゲイン比を計算する際に重み付き確率を用いた。また、オリジナルの C4.5 は連続値データをサポートしていないので、重み付き識別率が最も高くなる場所に閾値を設定して連続変数を離散化した。ブースティングのラウンド数を $M=100$ と設定した。SVM の実装として libsvm[13]を用い、5 fold cross validation を用いて SVM の C パラメータを決定した。

表 1. Characteristics of the datasets

Dataset	Dataset size	Training size	Number of Attribute
fourclass	855	428	2
breast-cancer	683	342	10
ionosphere	351	176	34
german	1000	500	24
pima	768	384	8
sonar	208	104	60
liver	345	173	6

表 2. Training and Test error rates of AdaBoost(AB) and AdaBoost+SVM.final(ABS.fin) in 100 round (%).

Dataset	AB(train)	AB(test)	ABS.f(train)	ABS.f(test)
fourclass	8.41	12.88	5.60	9.13
breast-cancer	0.29	4.98	0.29	4.69
ionosphere	0	9.14	0	8.57
german	22.00	25.8	22.00	26.40
pima	17.70	22.39	13.02	25.00
sonar	0	25.96	0	28.84
liver	14.40	27.32	0.89	32.55

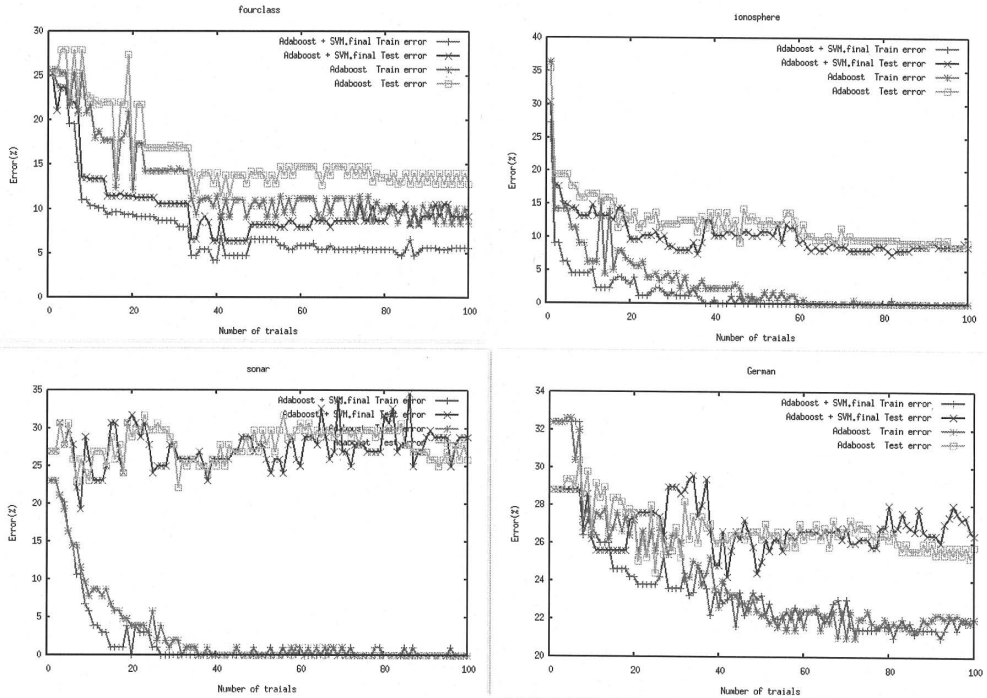


図 3. Error Rates of AdaBoost and AdaBoost.final

4. 2 実験結果 (AdaBoost+SVM.final)

はじめに、SVM による最適化の効果を調べるため、AdaBoost で構成した弱識別器を SVM で統合する最適化手法 (AdaBoost+SVM.final) の性能評価を行った。

AdaBoost と AdaBoost+SVM.final の 100 ラウンド目の学習データとテストデータに対する誤識別率を表 2 に示す。学習データにおいては全てのデータセットに対して AdaBoost+SVM.final の誤識別率は AdaBoost の誤識別率を下回っている。この結果は提案手法の学習データにおける最適化の有効性を示している。

テストデータにおいては、3 つのデータセット (fourclass, breast-cancer, ionosphere) において AdaBoost+SVM.final の誤識別率が AdaBoost の誤識別率を下回っている。一方で他の 4 つのデータセット (german, pima, sonar, liver) においては AdaBoost の誤識別率のほうが AdaBoost+SVM.final の誤識別率より少し低いかまたは互角となっている。4 つのデータセット (fourclass, ionosphere, sonar, german) に対する各ラウンドでの誤識別率のグラフを図 3 に示す。すべてのデータセットに対し、ほぼ全ての学習ラウンドにおいて AdaBoost+SVM.final のエラー率が AdaBoost のエラー率を下回っていることがわかる。データセット

fourclass においては、汎化性能 (テストデータに対する識別性能) も全ての学習ラウンドにおいて改善されていることが解る。データセット ionosphere においても全てのラウンドで若干の汎化性能の改善がみられる。german と sonar においては AdaBoost+SVM.final の 100 ラウンドでの汎化性能は AdaBoost に比べ若干低下しているが、他のラウンドでは AdaBoost よりも性能が高い場合と低い場合がある。よってこれらのデータセットに対しては、AdaBoost+SVM.final の汎化性能は AdaBoost と互角と考えられる。

4. 3 実験結果 (AdaBoost+SVM.seq)

次に AdaBoost の重みを各ラウンドで SVM を用いて最適化する手法 (AdaBoost+SVM.seq) の評価を行った。サンプリングパラメータ R に対応する学習データとテストデータの 100 ラウンドでの誤識別率を表 3, 表 4 にそれぞれ示す。4 つのデータセット (fourclass, ionosphere, sonar, german) において最適なパラメータ R における各ラウンドの誤識別率のグラフを図 4 に示す。データセット fourclass において、テストデータの識別性能は AdaBoost+SVM.final に比べて、誤識別率が 9.13% から 7.25% へ改善されている。

表 3. Training error rate of AdaBoost+SVM.seq in 100 round (%)

Dataset	Sampling parameter R								
	10	20	30	40	50	60	70	80	90
fourclass	19.62	13.08	18.92	18.45	8.64	2.33	4.67	9.36	11.70
breast-cancer	1.16	1.73	2.04	2.33	1.75	1.75	1.75	2.04	2.04
ionosphere	1.70	0	2.27	0	0	3.40	0.56	0	1.70
german	32.40	32.40	27.00	29.00	23.80	29.6	22.80	22.6	21.0
pima	25.0	23.95	22.05	21.09	15.34	13.28	15.10	17.70	17.70
sonar	0	0	0	0	0	1.92	0	0	6.73
liver	32.36	19.65	24.85	20.80	24.27	17.90	23.69	22.54	23.69

表 4. Test error rate of AdaBoost+SVM.seq in 100 round (%)

Dataset	Sampling parameter R								
	10	20	30	40	50	60	70	80	90
fourclass	22.04	14.28	22.01	20.60	11.24	7.25	9.83	9.36	11.70
breast-cancer	4.98	3.81	4.39	4.98	3.81	4.39	2.93	2.93	3.51
ionosphere	10.28	9.71	8.00	13.14	8.57	9.14	8.00	10.28	11.42
german	28.80	28.80	32.20	26.60	31.0	26.0	30.0	23.60	26.6
pima	25.00	25.00	25.00	23.95	23.17	27.08	25.26	25.00	25.0
sonar	27.88	23.07	23.07	16.34	24.03	22.11	19.23	22.11	21.15
liver	45.93	35.46	29.06	27.32	28.48	29.06	32.55	29.65	29.65

表 5. Test error rates of AdaBoost(AB), AdaBoost+SVM.final(ABS.fin), AdaBoost+SVM.seq(ABS.seq), linear-SVM(SVM(linear)) and RBF kernel SVM (SVM(RBF)) (%).

Dataset	AB	ABS.fin	ABS.seq	SVM(linear)	SVM(RBF)
fourclass	12.88	9.13	7.25	23.88	0
breast-cancer	4.98	4.69	2.93	6.45	0
ionosphere	9.14	8.57	8.00	15.28	6.85
german	25.8	26.40	23.6	22.60	23.40
pima	22.39	25.00	23.17	22.91	23.43
sonar	25.96	28.84	16.34	12.50	12.50
liver	27.32	32.55	27.32	30.23	31.39

データセット ionosphere においては fourclass ほどの大幅な改善は見られなかったが、AdaBoost の誤識別率よりは改善されている。データセット sonar においては 1 ステップでの SVM 最適化により汎化性能が改善されていないが、毎ステップ最適化することにより AdaBoost に比べ 25.96% から 16.34% へと大幅な汎化性能の改善が見られた。breast-cancer と german においても毎ステップ SVM により最適化を行うことでそれぞれ AdaBoost+SVM.final に比べ 26.4% から 23.6%、4.69% から 2.93% へと汎化性能の改善が見られた。liver と pima における汎化性能は AdaBoost と互角であった。

パラメータ R によっては、AdaBoost+SVM.seq は AdaBoost の識別性能よりも悪くなっている。このような状況は、ランダム性が不十分であったため多数の同じ弱識別器が構築されていたためである。従って、パラメータ R の選択が必要と考えられる。

表 5 に各データベースに対する 5 つの手法 (AdaBoost, AdaBoost+SVM.final, AdaBoost+SVM.seq, linear-SVM, RBF-SVM) の識別率を示す。AdaBoost が 1 ステップの SVM 最適化により汎化性能が改善されるかはデータに依存している。1 ステップの SVM 最適化により改善されたデータは毎ステップでの SVM 最適化によりさらに改善されていることがわかる。

5. まとめ

本稿では、AdaBoost の重みを線形 SVM により最適化する手法を提案した。

7 つのベンチマークデータセットを用いた実験により、提案手法は AdaBoost に比べ高い汎化性能を持つかまたは同等の性能であることが示された。特に AdaBoost+SVM.seq は AdaBoost+SVM.final より高い識別性能を示していた。

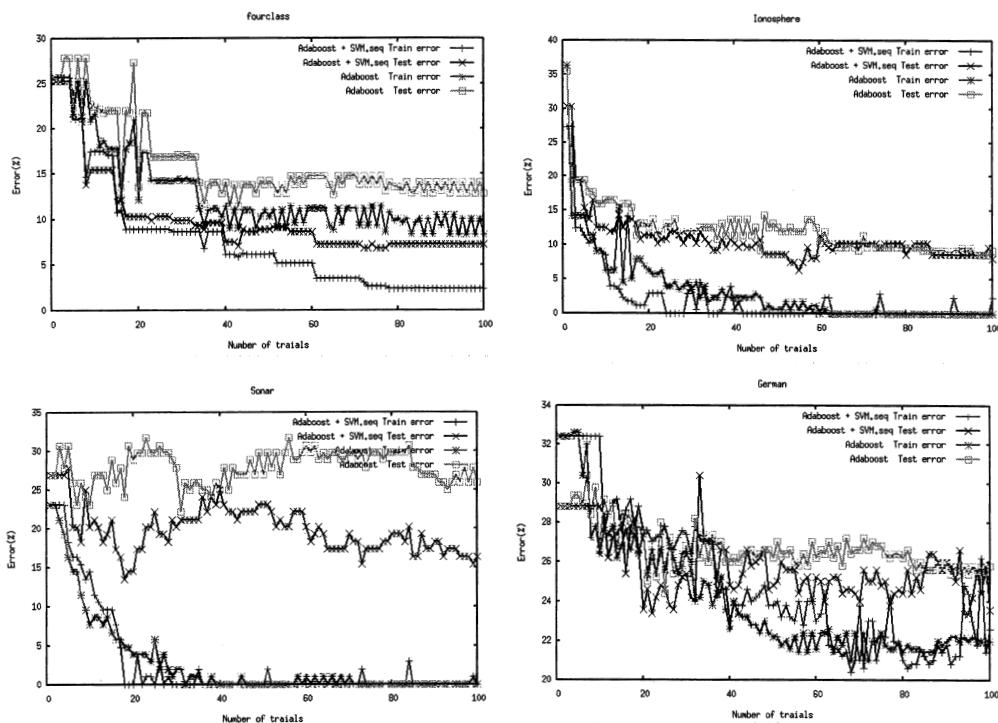


图 4. Error Rates of AdaBoost and AdaBoost.seq

文 献

- [1] R.Schapire: The strength of weak learnability: *Machine Learning*, 5(2): pp.197-pp.227, (1990)
- [2] Y.Freund: Boosting a weak learning algorithm by majority: *Information and Computation*, 121(2): pp.256-pp.285 (1995)
- [3] Y.Freund and R.Schapire: A decision-theoretic generalization of on-line learning and an application to boosting: *Journal of Computer and System Sciences*, pp.325-332 (1997)
- [4] E.Bauer, and R.Kohavi: An empirical comparison of voting classification algorithm: Bagging, boosting and variants, *Machine Learning* 36, pp.105-139: Kluwer Academic Publishers (1999)
- [5] Y.Freund and R.Schapire: Experiments with a new boosting algorithm: In Proc. of the 13th Intl. Conf. on Machine Learning, pp.148-156 (1996)
- [6] J.R.Quinlan: Bagging, boosting, and C4.5: Proc. of the Thirteenth National Conf. of Artificial Intelligence, AAAI Press and the MIT Press, pp.725-730 (1996)
- [7] R.Maclin and D.Optiz: An empirical evaluation of bagging and boosting: In Proc. of the Fourteenth National Conf. on Artificial Intelligence: pp.546-551(1997)
- [8] T.G.Diteterich: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization: *Machine Learning* 40: pp.139-157, Kluwer Academic Publishers (2000)
- [9] R.E.Shapire: The boosting approach to machine learning: An over view: In Workshop on Nonlinear Estimation and Classification. MSRI (2002)
- [10] J.Friedman, T.Hastie, and R.Tibshirani: Additive logistic regression: A statistical view of boosting: *The Annals of Statistics*, 38(2): pp.337-374 (2000)
- [11] V.N.Vapnik: *Statistical Learning Theory*: John Wiley&Sons(1998)
- [12] J.R.Quinlan: *C4.5: Programs for Machine Learning*. Morgan Kaufmann (1993)
- [13] C-C.Chang and C-J.Lin. : LIBSVM: a library for support vector machines (2001) software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- [14] A.Asuncion and D.j.Newman: UCI Machine Learning Repository: Irvine, CA: University of California, School of Information and Computer Science (2007) <http://www.ics.uci.edu/~mlern/ML>