

AdaBoost Toolbox

A MATLAB Toolbox for Adaptive Boosting

Alister Cordiner

Advanced Multimedia Research Laboratory
School of Computer Science and Software Engineering
University of Wollongong

23rd January 2009



Outline

- 1 Overview of AdaBoost
- 2 Using the toolbox
- 3 Using with other toolboxes

Overview of AdaBoost

- 1 Meta-learning algorithm for increasing accuracy of weak learners by weighting and combining them
- 2 Training samples reweighted based on previous performance to focus the training
- 3 Numerous extensions to the original AdaBoost algorithm

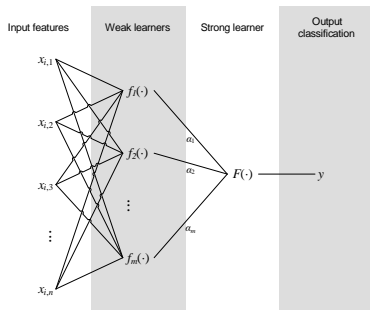


Figure: AdaBoost viewed as a neural network

Pseudocode for Gentle AdaBoost

Gentle AdaBoost

- 1 Start with weights $w_i = 1/N$, $i = 1, \dots, N$, $F(x) = 0$
- 2 Repeat for $m = 1, 2, \dots, M$
 - 1 Estimate $f_m(x)$ by weighted least squares fitting of y to x :

$$f_m = \arg \min_f \left(\epsilon_m = \sum_{i=1}^N w_i (y_i - f(x_i))^2 \right)$$

- 2 Set $w_i \leftarrow w_i \exp[-y_i \cdot f_m(x_i)]$ for $i = 1, 2, \dots, N$
 - 3 Renormalise such that $\sum_{i=1}^N w_i = 1$
- 3 Output the final classification:

$$F(x) = \begin{cases} \sum_{m=1}^M f_m(x) \geq 0 & +1 \\ \sum_{m=1}^M f_m(x) < 0 & -1 \end{cases}$$

AdaBoost Toolbox

- Implements Discrete AdaBoost, Gentle AdaBoost and Real AdaBoost
- Can be customised to implement different AdaBoost variants
- Designed to mimic the usage of the Neural Network Toolbox
- Designed for flexibility (rather than speed)

Steps for training and testing

Training:

- 1 Initialise a new AdaBoost object
- 2 Customise any object properties
- 3 Call the `train` method on the training data

Testing:

- 1 Call the `sim` method on the testing data
- 2 Use performance functions to evaluate the result

Customisable properties:

- Weak learners
- Input mapping function
- Feature extraction function
- Weak learner weighting rule
- Output combination rule
- Weight distribution initialisation, normalisation and update rule
- Resampling

Example 1

GentleBoost classification

Listing 1: Example of using the toolbox for classification

```
1 % load some toy training data
2 load databclass2d x y
3
4 % instantiate a GentleBoost object
5 bst = newgab;
6
7 % train the classifier for 50 rounds
8 bst = train(bst,x,y,50);
9
10 % simulate the trained classifier
11 ysim = sim(bst,x);
12
13 % find the training error rate of the classifier
14 er = perfer(ysim,y);
15 fprintf('Error rate = %f\n', er);
```

Example 2

GentleBoost regression

Listing 2: Example of using the toolbox for regression

```
1 % load some toy training data
2 load datareg3d x y
3
4 % instantiate a GentleBoost object
5 bst = newgabr;
6
7 % train the classifier for 50 rounds
8 % (use zero-mean shifted data)
9 bst.b = mean(y);
10 bst = train(bst,x,y-bst.b,50);
11
12 % simulate the trained classifier
13 ysim = sim(bst,x);
14
15 % find the training error rate of the classifier
16 er = perfmse(ysim,y);
17 fprintf('Error rate = %f\n', er);
```

Example 3

Changing the weak learner

Listing 3: Example of customising the weak learner

```
1 % instantiate a GentleBoost object
2 bst = newgab;
3
4 % use a decision stump weak learner
5 bst.weakSimFcn = @stumpfunc;
6 bst.weakTrainFcn = @stumplearn;
7 bst.singleFeatLearner = true;
8
9 % train the classifier for 50 rounds
10 bst = train(bst,x,y,50);
```

Example 4

Changing the feature extraction function

Listing 4: Example of customising the feature extraction function

```

1 % cell array containing list of image filenames
2 imgs = dir('*.jpg');
3
4 % instantiate a GentleBoost object
5 bst = newgab;
6
7 % function to read in and vectorise images (the features
8 % will be the pixel values)
9 bst.featExtractFcn = @(x) reshape(im2double(imread(x)),1,[]);
10
11 % train the classifier for 50 rounds
12 bst = train(bst, {imgs.name}, y, 50);
    
```

Assessing performance

The toolbox can provide numerous statistics to evaluate the accuracy of trained AdaBoost objects:

- TPR (true positive rate)
- FPR (false positive rate)
- ROC (receiver operating characteristic)
- DR (detection rate)
- MSE (mean squared error)
- AUC (area under curve)
- EER (equal error rate)

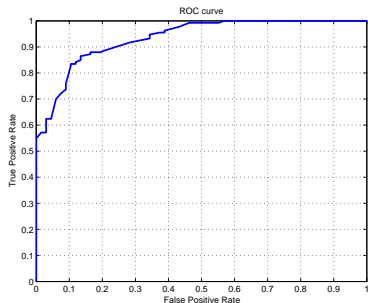


Figure: Example ROC curve generated by the toolbox

Using other toolboxes

Regression analysis

The technical report includes examples of using functionality provided by other MATLAB toolboxes:

- Classifier and regressor performance assessment
- Cross-validation training
- Weak learners

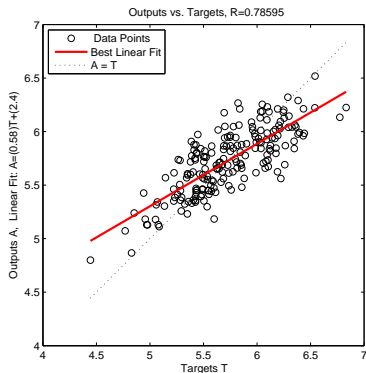







Figure: Example of regression analysis using functions from the Neural Network Toolbox

Bibliography

-  Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.
-  R. E. Schapire, Y. Freund, P. Barlett, and W. S. Lee, “Boosting the margin: A new explanation for the effectiveness of voting methods,” *The Annals of Statistics*, vol. 26, pp. 1651–1686, October 1998.
-  Y. Freund and R. E. Schapire, “A short introduction to boosting,” *Journal of Japanese Society for Artificial Intelligence*, vol. 14, pp. 771–780, September 1999.
-  J. Friedman, T. Hastie, and R. Tibshirani, “Additive logistic regression: a statistical view of boosting,” *Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.
-  R. Polikar, “Ensemble based systems in decision making,” *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.