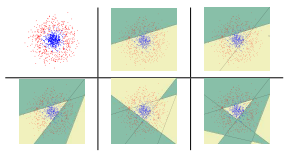
Adaptive Filters and Machine Learning

Boosting and Bagging

Poltayev Rassulzhan
rasulzhan@gmail.com



Can a set of weak classifiers be combined to derive a strong classifier?

Background

- Resampling
- Bootstrap
  We are using training set and different subsets in order to validate results

Combining models

- Can a set of weak classifiers be combined to derive a strong classifier?
- YES
- We are taking average results from different models
- Benefits:
  - classification performance will be better than single classifier
  - more resilience (elastic) to noise
- Minuses
  - models become difficult to explain
  - time consuming
- The main idea is "wisdom of the (simulated) crowd"

Bootstrap

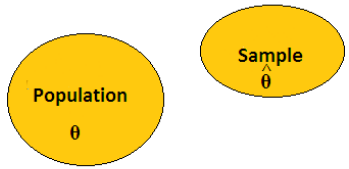
A "bootstrap" data set is one created by randomly selecting n points from the training set D, with replacement.

- D itself contains n points, there is nearly always duplication of individual points in a bootstrap data set.
- In boostrap estimation, selection process is independently repeated B times to yield B bootstrap data set, which are treated as independent set.
- Boostrap estimate of statistic θ, denoted $\hat{\theta}^{*(.)}$ is

$$\hat{\theta}^{*(.)} = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^{*(b)}, \tag{1}$$

where $\hat{\theta}^{*(b)}$ is estimate on boostrap sample b.

The usual statistical problem



Task: estimate the population parameter θ using the sample estimate $\hat{\theta}$

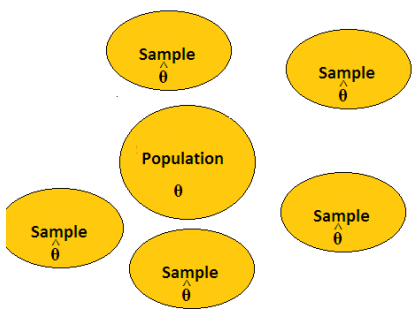
Statistical Question

How wrong it estimate?

Statistical Answer

- assess on variability of $\hat{\theta}$
- standard errors, confidence intervals, p-values for
- hypothesis tests about θ

Assess variability of the sample estimate $\hat{\theta}$ by taking additional samples, obtaining new estimates of θ each time.



Bias and Variance

- Bias

$$bias_{boot} = \frac{1}{B}\sum_{b=1}^{B} \hat{\theta}_b^{*(b)} - \hat{\theta} = \hat{\theta}^{*(.)} - \hat{\theta} \tag{2}$$

- Variance

$$Var_{boot}[\theta] = \frac{1}{B}\sum_{b=1}^{B} [\hat{\theta}^{*(b)} - \hat{\theta}^{*(.)}]^2 \tag{3}$$

Outline

History

- Introduced by Breiman (1996)
- "Bagging" stands for "bootstrap aggregating".
- It is an ensemble method: a method of combining multiple predictors.

Terms

- The arcing - adaptive reweighting and combining. It refers to reusing or selecting data in order to improve classification.
- Bagging - a name derived from "bootstrap aggregation" - uses multiple versions of a training set, each created by drawing n' $<$ n samples from D with replacement.
- A learning algorithm combination is informally called unstable if "small" changes in the training data lead to significantly different classifiers and relatively "large" changes in accuracy.

Bagging

Bootstrap Aggregation - Bagging Aggregation

- Imagine we have m sets of n independent observations $S^{(1)} = \{(X_1, Y_1), ..., (X_n, Y_n)\}^{(1)}, ..., S^{(m)} = \{(X_1, Y_1), ..., (X_n, Y_n)\}^{(m)}$ all taken iid from same underlying distribution P
- Traditional approach: generate some $\varphi(x, S)$ from all the data samples
- Aggregation: learn $\varphi(x, S)$ by averaging $\varphi(x, S^{(k)})$ over many k

Boostrapping

- unfortunately we usually have one single observations set S
- boostrap S to form the $S^{(k)}$ observation sets
- choose some samples, duplicate them until you fill a new $S^{(i)}$ of the same size of S
- the samples not used by each set are validation samples

Bagging

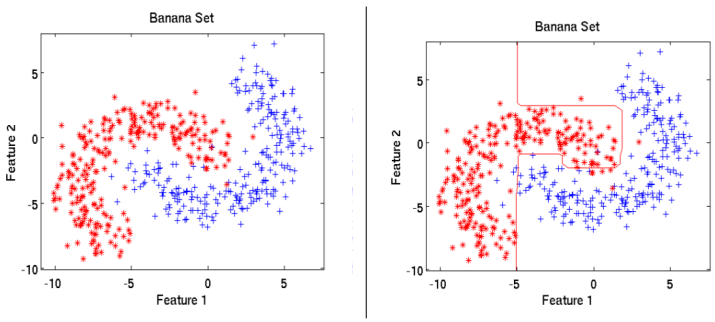
So bagging is

- bootrap model
- randomly generate D set of cardinality M from the original set Z with replacement
- corrects the optimistic bias of R-Method
- "bootstrap aggregation"
- create boostrap samples of a traininig set using sampling with replacement
- where each boostrap sample is used to train different component of base classifier
- where classification is done by plurarity voting

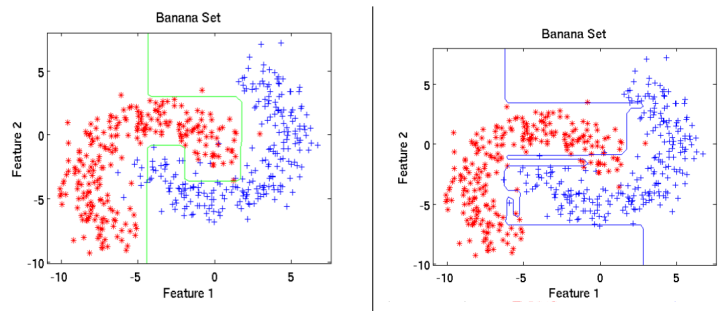
Details

1. Traininig phase
   Initialize the parameters
   - D = 0, the ensemble
   - L, the number of classifier to train
2. For k = 1,...,L
   - Take a boostrap sample S_k, from Z
   - Build a classifier D_k using S_k as the training set
   - Add the classifier to the current ensemble, $D = D \bigcup D_k$
3. Return D
   Classification phase
4. Run D_1,...,D_L on the input x
5. The class with the maximum number of votes is chosen as the label for x.

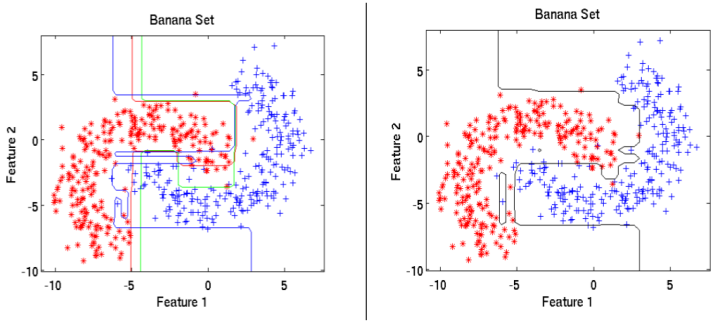
Example



Example (Cont.)



Example (Cont.)



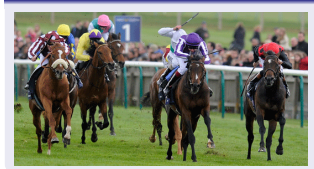
Conclusions from bagging

For error in learning is due to noise, bias and variance:

- noise is error by the target function
- bias is where the algorithm can not learn the target.
- variance comes from the sampling, and how it affects the learning algorithm
- does bagging minimizes these errors ?
- YES!!!
- averaging over bootstrap samples can reduce error from variance especially in case of unstable classifiers

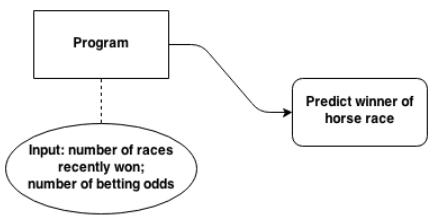
Problem

Horse-racing gambler



Goal

Maximize winnings



(4)

Problem: Betting strategy

Consider expert algorithm:

no initial data	with given information
	"rule of thumb"

Table : Betting startegy

"rule of thumb"

- Bet on the horse that recently won most races
- Bet on horse with most favorite odds
- etc...

Problem: Problem questions

In other words algorithm looks like that:

- choose small subset of data
- derive rough rule of thumb
- test second subset of data
- derive second rule of thumb
- repeat T-times

Problems

- How choose collections of races presented to expert for extract rules of thumb?
- How combine all rules into single to make accurate prediction?

Answers

- concentrate on "hardest" examples, that often misclassified by previous rules of thumb
- take weighted majority of rules of thumb

Introduction to Boosting

Boosting - general method for improving accuracy of any given learning algorithm

Details

- assume given "weak" learning algorithm that can consistently find classifiers ("rules of thumb") at least slightly better than 51% that is "weak learning assumption".
- given sufficient data, a boosting algorithm can probably construct single classifier with very high accuracy 98-99%

PAC learning model

Boosting has roots in theoretical machine (PAC) learning model
get random examples from unknown, arbtrary distribution
Strong and Weak learning algorithm

- Strong PAC learning algorithm for any distribution with high probability given polynomially many examples can find classifier with arbitrary small generalization error
- Weak PAC learning model same but generaization error only needs to be slightly better that random guessing $(\frac{1}{2} - \gamma)$

Kearns and Valiant says does weak learnability model make strong learnability?
We begin by describing the most popular boosting algorithm due to Freund and Schapire (1997) called "AdaBoost.M1."

What is AdaBoost?

AdaBoost (adaptive boosting) allows the designer to continue adding weak learners until some desired low training error has achieved.
AdaBoost "focused in" on the informative or "difficult" patterns.
AdaBoost is algorithm for constructing a "strong" classifier as linear combination

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{5}$$

of "simple" "weak" classifiers $h_t(x)$.
h_t - "weak" or basis classifier, hypothesis, "feature"
$H(x) = sign(f(x))$ - "strong" or final classifier/hypothesis

Do you remember description of boosting?

- we have training set $(x_1, y_1),...,(x_m, y_m)$
- $y_i \in \{-1, +1\}$ correct label of instance $x_i \in X$
- for t = 1,...,T:
  - construct distribution D_t on $1, .., m$
  - find weak classifier ("rule of thumb")

$$h_t : X \longrightarrow \{-1, +1\} \tag{6}$$

with small error ϵ_t on D_t:

$$\epsilon_t = Pr_{i \sim D_t}[h_t(x_i) \neq y_i] \tag{7}$$

- output final classifier H_{final}

AdaBoost

- construction D_t
  - Initialize weights $D_1(i) = \frac{1}{m}$
  - given D_t and h_t:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & if \quad y_i = h_t(x_i) \\ e^{\alpha_t} & if \quad y_i \neq h_t(x_i) \end{cases} \tag{8}$$

where Z_t = normalization factor

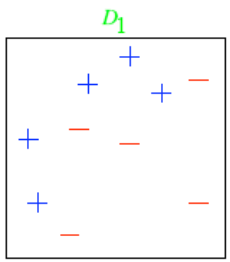
$$\alpha_t = \frac{1}{2} ln(\frac{1 - \epsilon_t}{\epsilon_t}) > 0 \tag{9}$$

- final classifier
  - $H_{final(x) = sign(\sum_t \alpha_t h_t(x))}$

Toy example of Robert Schapire

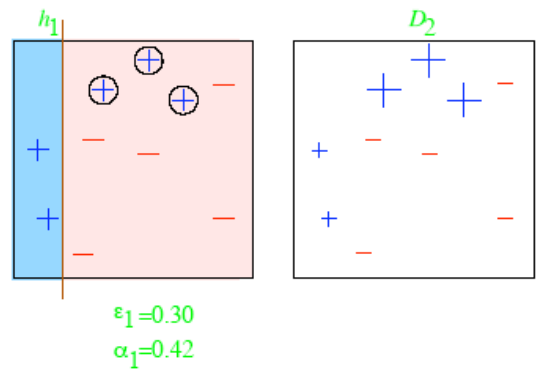
In that example, we have weak classifiers

- vertical half-plane
- horizontal half-plane



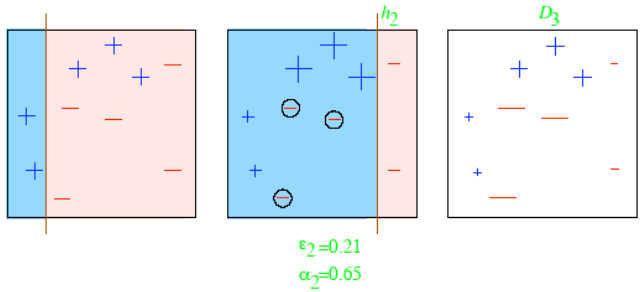
(10)

Round 1

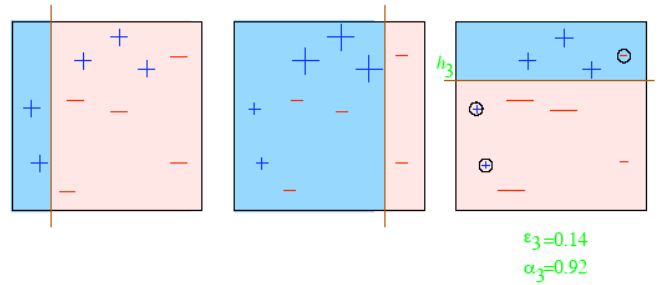


(11)

Round 2

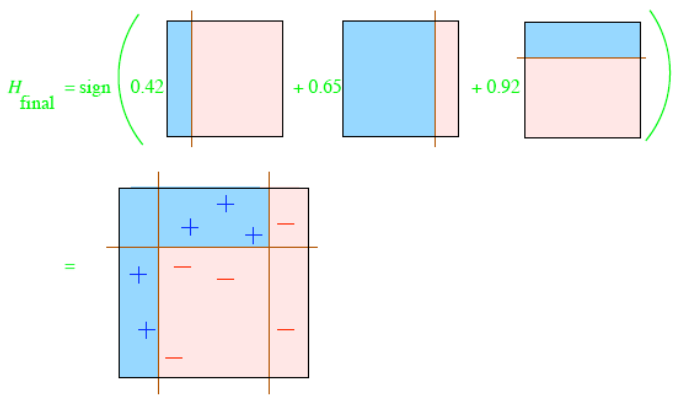


Round 3



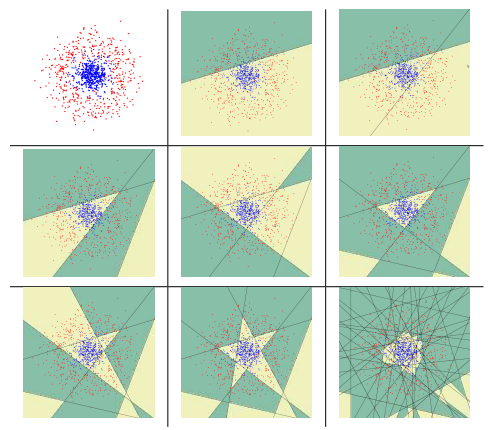
(12)

Final Classifier



(13)

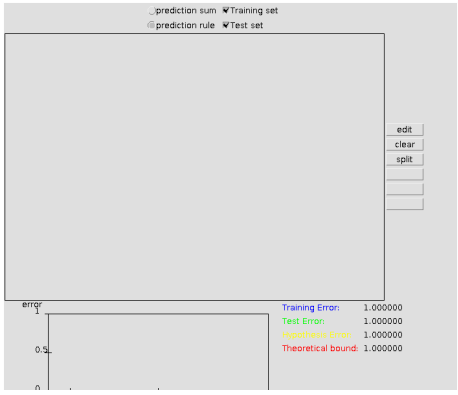
One more example



From Jiri Matas and Jan Sochman

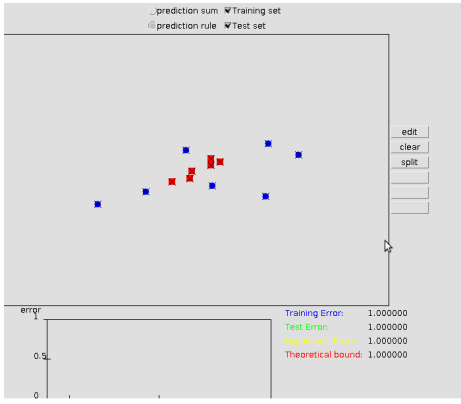
Practice

- Test http://cseweb.ucsd.edu/ yfreund/adaboost/



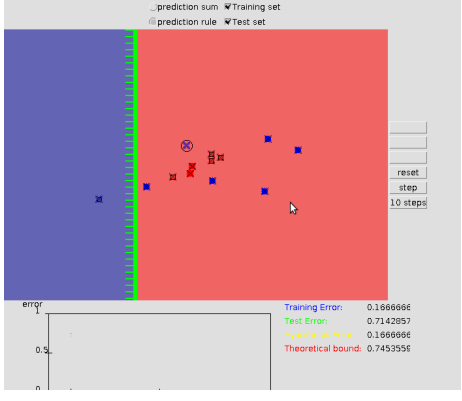
Testing

Practice



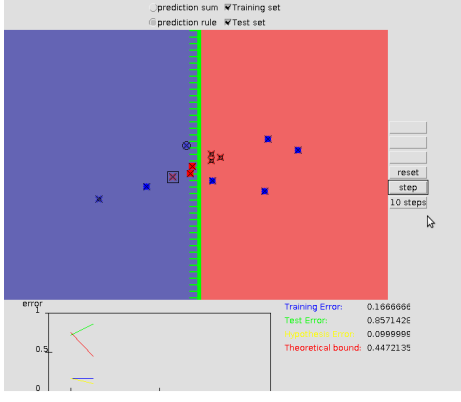
Testing

Practice



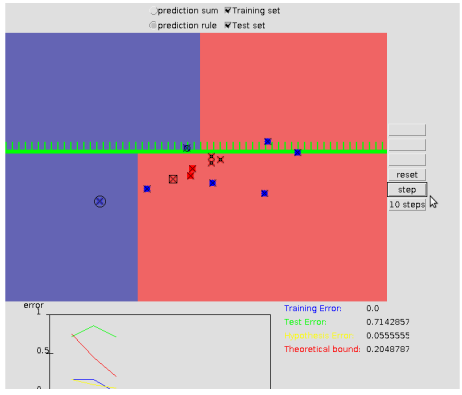
Testing

Practice



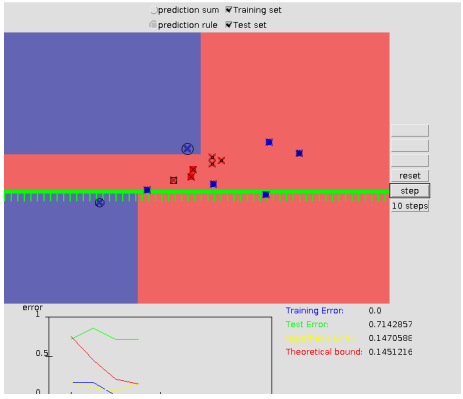
Testing

Practice



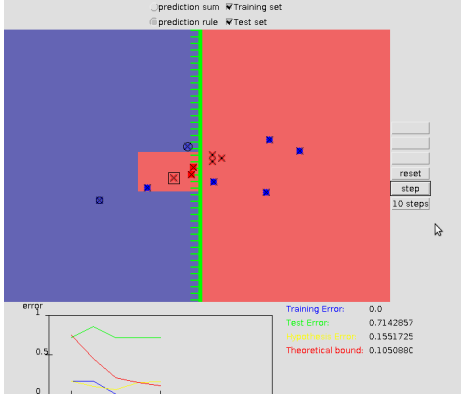
Testing

Practice



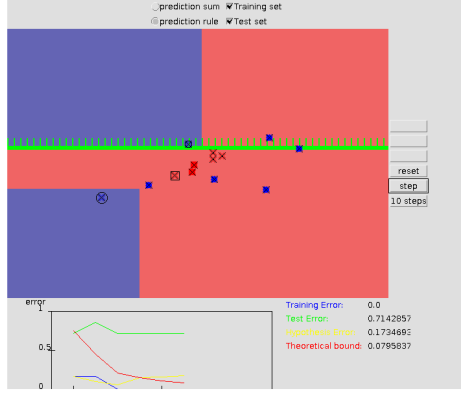
Testing

Practice



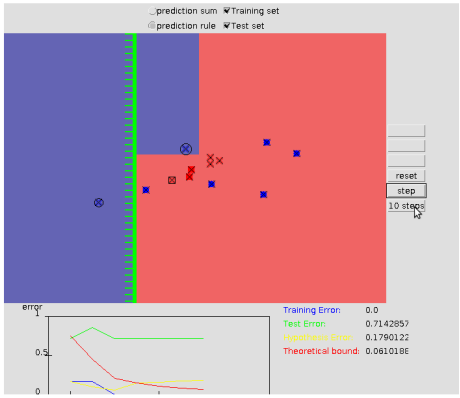
Testing

Practice



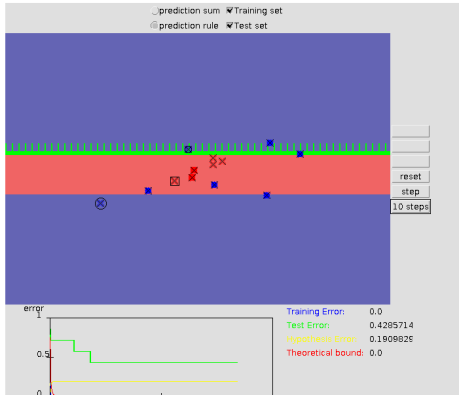
Testing

Practice



Testing

Practice



Testing

Training error:Theorem

Theorem

- write ϵ_t as $1/2 - \gamma_t \quad [\gamma_t =$"edge"$]$
- then
  training error$(H_{final}) \leq \prod_t [2\sqrt{\epsilon_t(1-\epsilon_t)}]$
  $= \prod_t \sqrt{1-4\gamma_t^2} \leq exp(-2\sum_t \gamma_t^2)$
- so: if $\forall_t$: $\gamma_t \geq \gamma > 0$
  then training error$(H_{final}) \leq e^{-2\gamma_t^2 T}$

We must understand that AdaBoost is adaptive:

- does not need to know γ or T a priori
- can exploit $\gamma_t \gg \gamma$

Training error: Proof

We can prof theorem in 3 steps.
Step #1

$$D_{T+1}(i) = \frac{1}{m} * \frac{exp(-y_i f(x_i))}{\prod_t Z_t} \tag{14}$$

where

$$f(x) = \sum_t \alpha_t h_t(x). \tag{15}$$

Proof:

$$D_{T+1}(i) = \frac{1}{m} * \frac{exp(-\alpha_t y_i h_t(x_i))}{\prod_t Z_t} \tag{16}$$

Unwrapping recurrence, we get that

$$D_{T+1}(i) = D_1(i) * \frac{exp(-\alpha_1 y_i h_1(x_i))}{Z_1} ....D_1(i) * \frac{exp(-\alpha_T y_i h_T(x_i))}{Z_T} \tag{17}$$

Training error: Proof

Step #2

The training error of final classifier H is at most

$$\prod_{t=1}^{T} Z_t \tag{18}$$

Prof:

$$\begin{aligned}
\text{training error(H)} &= \frac{1}{m}\sum_i \begin{cases} 1 \ if \ y_i \neq H(x_i) \\ 0 \ else \end{cases} && \text{by definition} \\
&= \frac{1}{m}\sum_i \begin{cases} 1 \quad if \ y_i f(x_i) \leq 0 \\ 0 \ else \end{cases} && H(x) = sign(f(x)) \\
& && y_i \in \{-1., +1\} \\
&\leq \frac{1}{m}\sum_i exp(-y_i f(x_i)) && \text{since } e^{-z} \geq 1 \text{ if } z \leq 0 \\
&= \sum_i D_{T+1}(i) \prod_t Z_t = \prod_t Z_t && \text{by Step \#1 above}
\end{aligned}$$

Training error: Proof

Step #3

The last step is to compute Z_t

We can compute this normalization constants as follows:

$$\begin{aligned}
Z_t &= \sum_i D_t(i) \times \begin{cases} e^{-\alpha_t} \ if \ h_t(x) = y_i \\ e^{\alpha_t} \ if \ h_t(x) \neq y_i \end{cases} \\
&= \sum_{i:h_t(x_i)=y_i} D_t(i) e^{-\alpha_t} + \sum_{i:h_t(x_i)\neq y_i} D_t(i) e^{\alpha_t} \\
&= e^{-\alpha_t} \sum_{i:h_t(x_i)=y_i} D_t(i) + e^{\alpha_t} \sum_{i:h_t(x_i)\neq y_i} D_t(i) \\
&= e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t && \text{by definition of } \epsilon_t \\
&= 2\sqrt{(1-\epsilon_t)\epsilon_t} && \text{by our choice of } \alpha_t \\
&= \sqrt{(1-4\gamma_t^2)} && \text{plugging in } \epsilon_t = \frac{1}{2} - \gamma_t \\
&\leq e^{-2\gamma_t^2} && \text{using } 1 + x \leq e^x \text{ for all real x}
\end{aligned}$$

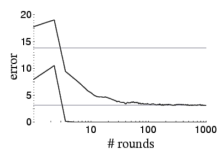
Combining with Step #2 gives the claimed upper bound on the training error of H. THEOREM PROVED

Training error: Result

Training error theorem

$$\frac{1}{m}\sum_i \begin{cases} 1 \ if \ y_i \neq H(x_i) \\ 0 \ else \end{cases} \leq \prod_{t=1}^{T} Z_t \leq exp(-2\sum_t \gamma^2)$$

AdaBoost will achieve zero training error (exponentially fast):



Digits recognition

Boosting

- robust for overfititng
- test error decreases even after training error is zero

Generalization error

$$error_{true}(H) \leq error_{train}(H) + O(\sqrt{\frac{dT}{m}}) \tag{19}$$

where

- T – number of boosting rounds
- d – VC dimension of weak learner, measures complexity of classifier
- m – number of training examples

Margin

We can define margin as follows

$$\gamma(x_i) = y_i * \frac{\alpha_1 h_1(x) + ... + \alpha_m h_m(x)}{\alpha_1 + ... + \alpha_m} \quad (20)$$

where $\gamma(x_i) \in [-1, +1]$, positive if $H(x_i) = y_i$
Iterations of AdaBoost increase the margin of training examples

Margin

Theory

- Testing error continues to decrease
- Margin for an object is related to certainty of its classification.
- Positive and large margin is correct classification
- Negative margin is incorrect classification
- Very small margin is uncertainty in classification

Application

Viola-Jones Haar-Like wavelets

2 rectangles of pixels

$\varphi_1(x)$ $\varphi_2(x)$

Millions of possible classifiers

etc...

Note

$I(x)$ is pixel of image I at position x

$$f(x) = \sum_{x \in A} I(x) - \sum_{x \in B} I(x) \quad (21)$$

$$\varphi(x) = \begin{cases} 1 & if\ f(x) > 0, \\ -1 & otherwise \end{cases} \quad (22)$$

Viola-Jones Result

Review

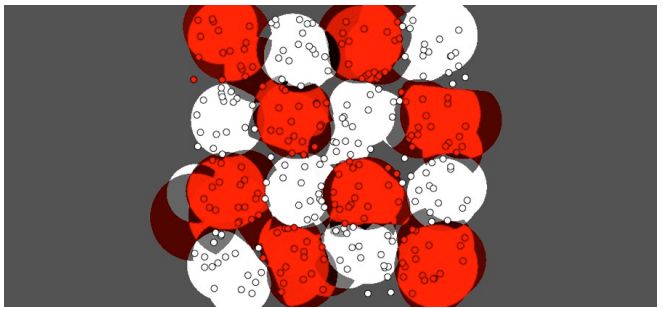
- SVM with kernel K

$$max \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i x_j) \tag{23}$$

  where $0 \leq \alpha_i \leq C$
- Classification of x: $\hat{y} = sign(\hat{w_0} + \sum_{\alpha_i > 0} \alpha_i y_i K(x_i, x))$
- A positive-definie kernel corresponds to dot product in feature space

SVM

Weak Learners don't need to be weak!



20 boosted SVMs with 5 SVs and the RBF kernel

Similarity

- So we understand that boosting has similar idea with combine classifiers:
  given hypothesis functions $h_1(x), ..., h_m(x)$

$$H(x) = \alpha_1 h_1(x) + ... + \alpha_m h_m(x), \tag{24}$$

- α_i is the vote assigned to classifier h_i.
- Prediction:

$$\hat{y}(x) = sign\ H(x) \tag{25}$$

- Classifier h_i can be simple (e.g. based on single feature).

Bagging and Boosting

- Bagging: linear combination of multiple learners
  - Very robust to noise
  - A lot of redundant effort
- Boosting: weighed combination of arbitrary
  - Very strong learner from very simple ones
  - Sensitive to noise

Bagging and Boosting (Cont.)

- Bagging: each model is trained independently
- Boosting: each model is built on top of the previous ones

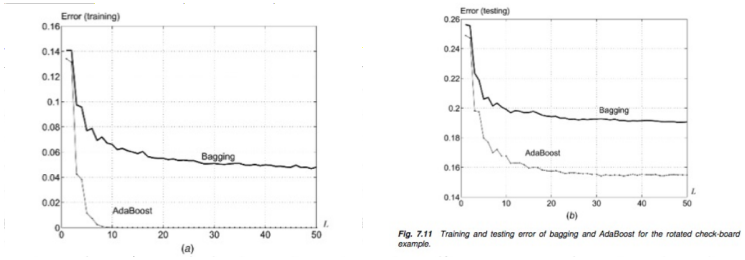


Fig. 7.11 Training and testing error of bagging and AdaBoost for the rotated check-board example.

References

- Richard O. Duda, Peter E. Hart and David G. Stork. *Pattern Classification*
- Jiri Matas and Jan Sochman. *AdaBoost*
- Bishop. *Boosting*
- Robert E. Schapire. *Boosting*
- Robert E. Schapire and Yoav Freund. *A Short Introduction to Boosting*