Topic 11: Advanced Topics (III) – Ensemble learning（集成学习）

Min Zhang
z-m@tsinghua.edu.cn

Background

"Two heads are better than one."

"三个臭皮匠，顶一个诸葛亮"

- Integrate results of multiple learning approaches to improve the performance

Ensemble learning

1. Introduction to ensemble learning

Two concepts

- Strong learner: learning algorithm with high accuracy
- Weak learner: performance on any training set is slightly better than chance prediction

$$\text{error} = \tfrac{1}{2} - \gamma$$

Can we improve a weak learner to a strong learner?

Introduction to ensemble learning

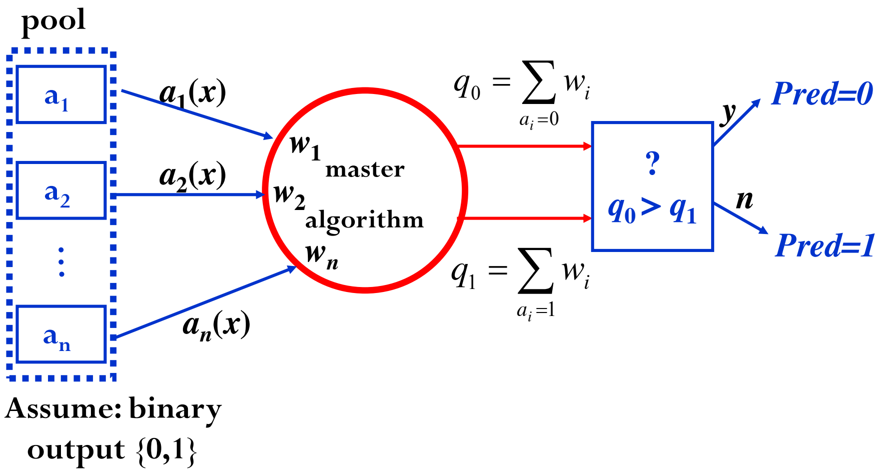
- INTUITION: *Combining Predictions of an ensemble is more accurate than a single classifier*
- Justification: (Several reasons)
  - Easy to find quite correct "rules of thumb" however hard to find single highly accurate prediction rule.
  - If the training examples are few and the hypothesis space is large then there are several equally accurate classifiers.
  - Hypothesis space does not contain the true function, but it has several good approximations.
  - Exhaustive global search in the hypothesis space is expensive so we can combine the predictions of several locally accurate classifiers.

Ensemble learning: basic idea

- Sometimes a single classifier (e.g. decision tree, neural network, …) won't perform well, but a weighted combination of them will.
- Each learner in the pool has its own weight
- When ask to predict the label for a new example
  - Each expert makes its own prediction
  - Then the master algorithm combine them using the weights for its own prediction (i.e. the "official" one)

2. Weighted Majority Algorithm (加权多数算法)

Weighted majority algorithm –Prediction



Weighted majority algorithm – Training

a_i is the i[th] pred. algorithm in pool A,; each alg. is arbitrary function from X to {0,1}

w_i is the weight associates with a_i

- $\forall i$, $w_i \leftarrow 1$
- For each training example (or trail) $<x,c(x)>$
  - Set $q_0 \leftarrow q_1 \leftarrow 0$
  - For each algorithm a_i
    - If $a_i(x)=0$, then $q_0 \leftarrow q_0+w_i$, else $q_1 \leftarrow q_1+w_i$
  - If $q_0 > q_1$, then predict $c(x)=0$, else predict $c(x)=1$

    (case for $q_0 = q_1$ is arbitrary)
  - For each $a_i \in$ A
    - If $a_i(x) \neq c(x)$, then $w_i \leftarrow \beta\, w_i$ ($\beta \in [0,1)$ is the penalty coefficient)

$\beta = 0$ yields Halving algorithm over A

Weighted majority (WM) algorithm: mistake bound

- Let W_t = sum of weights before trail t ($W_1 = n$, $\beta = 1/2$)
- On trail t such that WM makes a mistake, the total weight of algorithms with the mistake is:

$$W_t^{mis} = \sum_{a_i(x_t) \neq c(x_t)} w_i \geq W_t/2$$

- So $W_{t+1} = W_t - W_t^{mis}/2 \leq 3W_t/4$
- After seeing all samples (sample set S), M = total number of mistakes

$$W_{|S|+1} \leq W_1 (3/4)^M = n\,(3/4)^M$$

- Let $a_{opt} \in A$ be the alg. that makes fewest error on arbitrary sequence S of examples; k = number of mistakes; then the final weight of a_{opt} is $(1/2)^k$
- $(1/2)^k \leq n\,(3/4)^M$, yielding $\boxed{M \leq \frac{k + \log_2 n}{-\log_2(3/4)} \leq 2.4\,(k + \log_2 n)}$

Weighted majority (WM) algorithm: mistake bound (cont.)

- For any arbitrary sequence of samples:

$$M \leq 2.4\ (k + \log_2 n)$$

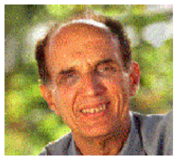
- Other results:
  - Bounds hold for general values of $0 \leq \beta < 1$ ***(Pls analyze by yourself.)***
  - Better bounds hold for many sophisticate algorithms, but only better by a constant value (worst case lower bound is $\Omega(k+\log n)$)
  - Get bounds for real-valued labels and predicts
  - Can track shifting concept (where best alg. can suddenly change in S)
  - Don't make any weight too low (compared to other weights) (i.e. don't over-commit)

3. Bagging

If we have only one weak learner,

how to improve the performance by ensemble?

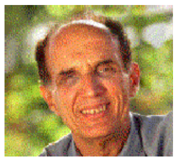
Bagging: background



- Bagging = Bootstrap aggregating
- Bootstrap: proposed by Bradley Efron in 1993
  - Professor of Statistics
  - Stanford University
  - Bootstrap, Biostatistics, Statistical methods in Astrophysics
- *"I like working on applied and theoretical problems at the same time and* ***one thing nice about statistics is that you can be useful in a wide variety of areas.*** *So my current applications include biostatistics and also astrophysical applications. The surprising thing is that the methods used are similar in both areas. I gave a talk called* ***Astrophysics and Biostatistics--the odd couple*** *at Penn State that made this point."*

25 introduction to machine learning: ensemble learning

Bagging: background



- Bagging = Bootstrap aggregating
- Bootstrap: proposed by Bradley Efron in 1993
  - Professor of Statistics
  - Stanford University
  - Bootstrap, Biostatistics, Statistical methods in Astrophysics
- Bootstrap sampling (拔靴法/自举法采样)
  - Given a set D containing m training examples
  - Create D_i by drawn m examples uniformly at random with replacement from D (drawn with replacement, 取出放回)
  - Expect D_i to omit some examples from D

26 introduction to machine learning: ensemble learning

Bagging: algorithm

Leo Breiman

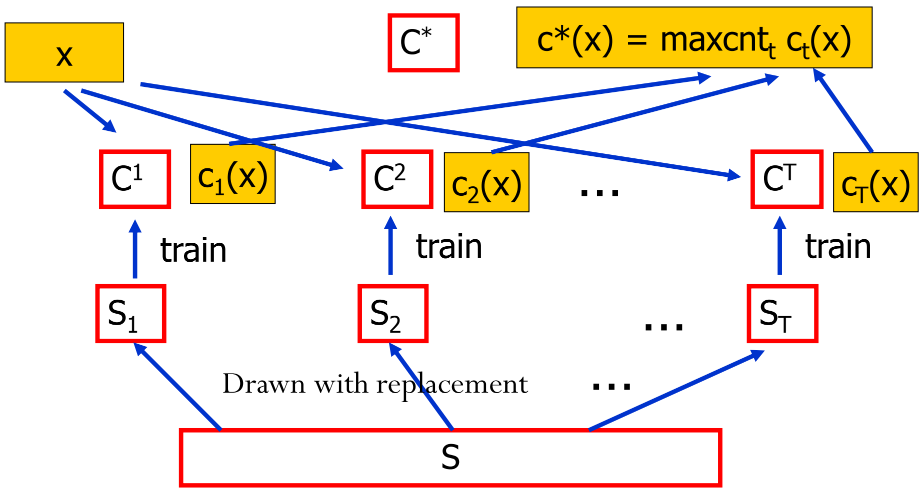
- Bagging: proposed by Breiman in 1994
  - Professor Emeritus of Statistics, Berkeley
  - Member of American Academy of Science
- Bagging algorithm

For $t = 1, 2, \ldots, T$ Do
  create boostrap sample D_t from S
  train a classifier H_t on D_t
Classify new instance $x \in X$ by majority vote of H_t (equal weights)

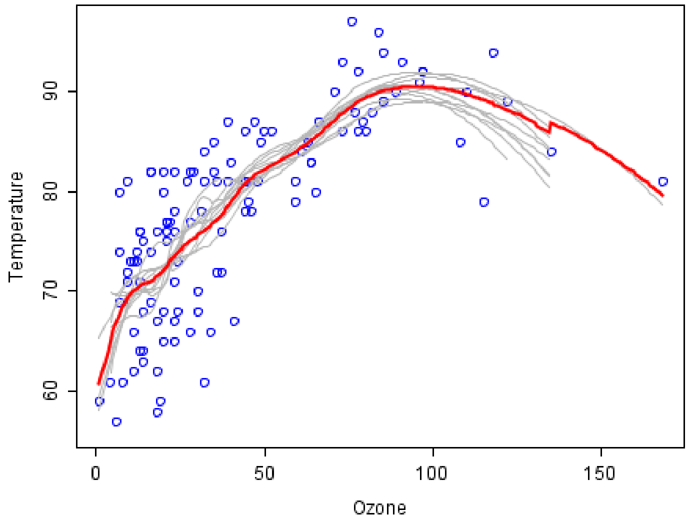
You can also use different combining strategy on your problem.

- Can predict continuous output

Bagging



Bagging application example



Data set: Rousseeuw and Leroy (1986), concerning ozone levels vs. temperature.

100 boostrap samples. Gray lines: first 10 predictor; red line: mean

30

How Many Bootstrap Samples?

Table 5.1
Bagged Missclassification Rates (%)

No. Bootstrap Replicates	Missclassification Rate
10	21.8
25	19.5
50	19.4
100	19.4

Breiman "Bagging Predictors" Berkeley Statistics Department TR#421, 1994

31

Bagging: Results (cont.)

Given sample S of labeled data, Breiman did the following 100 times and reported average:

Approach I:

1. **Divide S randomly into test set T(10%) and training set D(90%)**
2. **Learn decision tree from D, let e_s be its error rate on T**

Approach II:

Do 50 times: create bootstrap set D_i, learn decision tree, let e_B be the error of a majority vote of trees on T, so ensemble size = 50)

Table 1 Missclassification Rates (Percent)

Data Set	$\bar{e}_S$	$\bar{e}_B$	Decrease
waveform	29.0	19.4	33%
heart	10.0	5.3	47%
breast cancer	6.0	4.2	30%
ionosphere	11.2	8.6	23%
diabetes	23.4	18.8	20%
glass	32.0	24.9	22%
soybean	14.5	10.6	27%

Breiman "Bagging Predictors" Berkeley Statistics Department TR#421, 1994

Bagging: Results (cont.)

- Same experiment, but use a nearest neighbor classifier (Euclidean distance)
- Results

Data Set	$\bar{e}_S$	$\bar{e}_B$	Decrease
waveform	26.1	26.1	0%
heart	6.3	6.3	0%
breast cancer	4.9	4.9	0%
ionosphere	35.7	35.7	0%
diabetes	16.4	16.4	0%
glass	16.4	16.4	0%

- What happened ? Why ?

Bagging： special points

- Bagging helps when learner is "unstable"
  - " The vital element is the instability of the prediction method"
  - E.g. Decision tree, neural network
- Why？
  - Unstable: small change in training set cause large change in hypothesis produced
  - "If perturbing the learning set can cause significant changes in the predictor constructed, then bagging can improve accuracy." (Breiman 1996)

Bagging： special points (cont.)

- Each base classifier is trained on less data
  - Only about 63.2% of the data points are in any bootstrap sample
- However the final model has seen all the data
  - On average a point will be in > 50% of the bootstrap samples

Recall

- Weighted majority algorithm
  - Same data set, different learning algorithms
  - Generate multiple models, and weighted combination
- Bagging
  - One data set, one weak learner
  - Generate multiple training samples to train multi-models, and ensemble

Is there an ensemble algorithm that takes into account the differences of the data in learning?

Boosting

4. Boosting

Boosting background

- Comes from PAC-Learning Model

(PAC-learning will be introduced in the next week)

  - Valiant Leslie G. proposed PAC in 1984
    - Harvard University
    - Member of America Academy of Science
    - A world leader in theoretical computer science
    - 2010 Turing Award

Boosting： basic idea

- "Learn from failures"
- Basic idea：
  - Assign a weight to each example
  - T iterations, increase weights of misclassified examples after each iteration – focus more on "hard" ones

Boosting background

- [Kearns&Valiant'88]

  Open problem of finding a boosting algorithm
- [Schapire'89], [Freund'90]

  First polynomial-time boosting algorithms
- [Drucker, Schapire & Simard '92]

  First experiments using boosting
- [Freund & Schapire '95]
  - Introduced AdaBoost algorithm
  - Strong practical advantages over previous boosting algorithms
- Experiments using AdaBoost， continuing development of theory & algorithms (using not-so-weak learners, etc)

AdaBoost

- Initially assign an equal weight $1/N$ to each example；
- *For* $t = 1, 2, \ldots, T$ *Do*
  - Generate a hypothesis C_t；
  - Compute the error rate E_t :

    E_t = sum of the weights of all misclassified samples;
  - $\alpha_t = \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}$
  - Update the weight of each example:

    correctly classified: $W_{new} = W_{old} * e^{-\alpha_t}$

    misclassified: $W_{new} = W_{old} * e^{\alpha_t}$
  - Normalize weights (the sum of weights=1)；
- Combine all C_t with the voting weight of α_t

AdaBoost.M1

Vs. AdaBoost

- Initially assign an equal weight $1/N$ to each example;
- *For t = 1, 2, …, T Do*
  - Generate a hypothesis C_t;
  - Compute the error rate E_t :

    E_t= sum of the weights of all misclassified samples;
  - $\beta_t = E_t / (1 - E_t)$ **α_t=1/2 ln ((1- E_t)/ E_t)**
  - Update the weight of each example:

    correctly classified: $W_{new} = W_{old} * \beta_t$

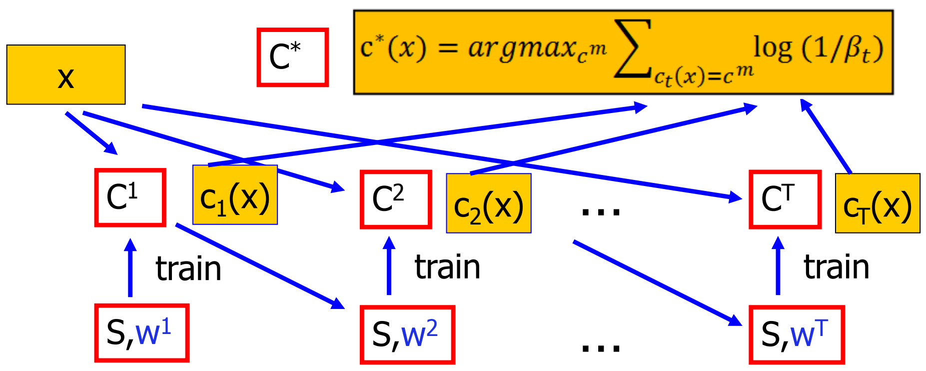
    misclassified: $W_{new} = W_{old}$

    $W_{new} = W_{old} * e^{-\alpha_t}$

    $W_{new} = W_{old} * e^{\alpha_t}$
  - Normalize weights (the sum of weights=1);
- Combine all C_t with the voting weight of $\log[1/\beta_t]$ α_t

43 introduction to machine learning: ensemble learning

Boosting

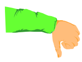
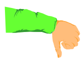
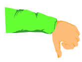
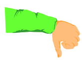
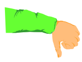
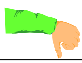


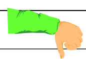
44 introduction to machine learning: ensemble learning

AdaBoost example (1)

T1	T2	T3	T4	Ob
1	0	1	1	1
1	0	1	1	1
1	1	1	1	1
1	1	1	0	0
1	0	1	0	0
1	1	0	1	0
1	0	0	1	0
1	1	0	1	0

AdaBoost example (1)

T1	T2	T3	T4	Ob	Weight
1	0	1	1	1	
1	0	1	1	1	
1	1	1	1	1	
1	1	1	0	0	
1	0	1	0	0	
1	1	0	1	0	
1	0	0	1	0	
1	1	0	1	0	

Size of  represents the degree of the weight.

hypothesis

T1	T2	T3	T4	Ob	Weight	if T1=1 then Ob=0 else Ob=1
1	0	1	1	1		0
1	0	1	1	1		0
1	1	1	1	1		0
1	1	1	0	0		0
1	0	1	0	0		0
1	1	0	1	0		0
1	0	0	1	0		0
1	1	0	1	0		0

Size of represents the degree of the weight.

hypothesis

T1	T2	T3	T4	Ob	Weight	if T1=1 then Ob=0 else Ob=1	New Weight
1	0	1	1	1		0	
1	0	1	1	1		0	
1	1	1	1	1		0	
1	1	1	0	0		0	
1	0	1	0	0		0	
1	1	0	1	0		0	
1	0	0	1	0		0	
1	1	0	1	0		0	

Size of represents the degree of the weight.

Another hypothesis

T1	T2	T3	T4	Ob	Weight
1	0	1	1	1	
1	0	1	1	1	
1	1	1	1	1	
1	1	1	0	0	
1	0	1	0	0	
1	1	0	1	0	
1	0	0	1	0	
1	1	0	1	0	

Another hypothesis

T1	T2	T3	T4	Ob	Weight	if T3=1 then Ob=1 else Ob=0
1	0	1	1	1		1
1	0	1	1	1		1
1	1	1	1	1		1
1	1	1	0	0		1
1	0	1	0	0		1
1	1	0	1	0		0
1	0	0	1	0		0
1	1	0	1	0		0

Another hypothesis

T1	T2	T3	T4	Ob	Weight	if T3=1 then Ob=1 else Ob=0	New Weight
1	0	1	1	1		1	
1	0	1	1	1		1	
1	1	1	1	1		1	
1	1	1	0	0		1	
1	0	1	0	0		1	
1	1	0	1	0		0	
1	0	0	1	0		0	
1	1	0	1	0		0	

Another hypothesis

T1	T2	T3	T4	Ob	Weight
1	0	1	1	1	
1	0	1	1	1	
1	1	1	1	1	
1	1	1	0	0	
1	0	1	0	0	
1	1	0	1	0	
1	0	0	1	0	
1	1	0	1	0	

Another hypothesis

T1	T2	T3	T4	Ob	Weight	if T4=1 then Ob=1 else Ob=0
1	0	1	1	1		1
1	0	1	1	1		1
1	1	1	1	1		1
1	1	1	0	0		0
1	0	1	0	0		0
1	1	0	1	0		1
1	0	0	1	0		1
1	1	0	1	0		1

Another hypothesis

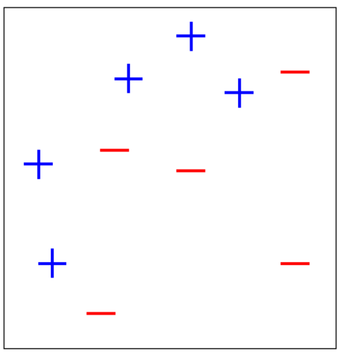
T1	T2	T3	T4	Ob	Weight	if T4=1 then Ob=1 else Ob=0	New Weight
1	0	1	1	1		1	
1	0	1	1	1		1	
1	1	1	1	1		1	
1	1	1	0	0		0	
1	0	1	0	0		0	
1	1	0	1	0		1	
1	0	0	1	0		1	
1	1	0	1	0		1	

AdaBoost example (1)

T1	T2	T3	T4	Ob	Hypotheses: if T1=1 then Ob=0 else Ob=1	if T3=1 then Ob=1 else Ob=0	if T4=1 then Ob=1 else Ob=0	Simple Majority Voting
1	0	1	1	1	0	1	1	1
1	0	1	1	1	0	1	1	1
1	1	1	1	1	0	1	1	1
1	1	1	0	0	0	1	0	0
1	0	1	0	0	0	1	0	0
1	1	0	1	0	0	0	1	0
1	0	0	1	0	0	0	1	0
1	1	0	1	0	0	0	1	0

AdaBoost example (2)

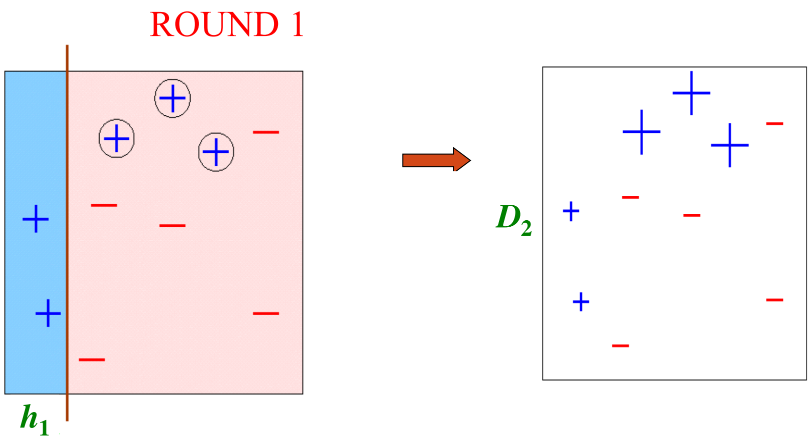
D_1



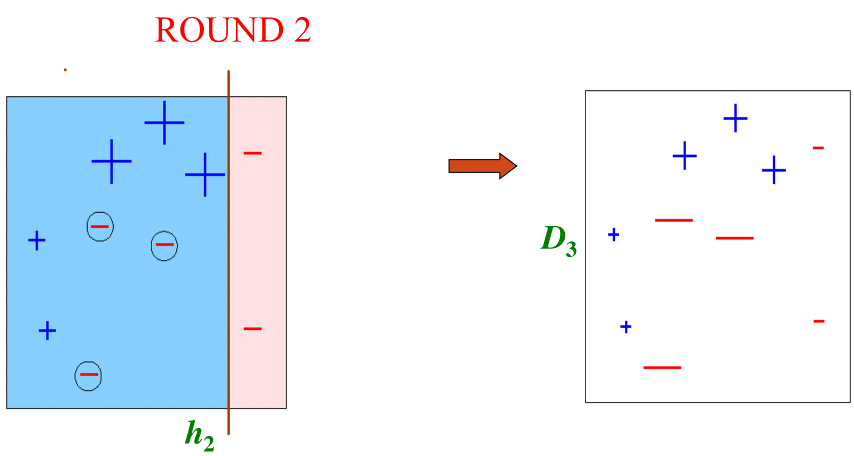
Original Training set : Equal Weights to all training samples

-- from "**A Tutorial on Boosting" by** Yoav Freund and Rob Schapire

AdaBoost example (2)

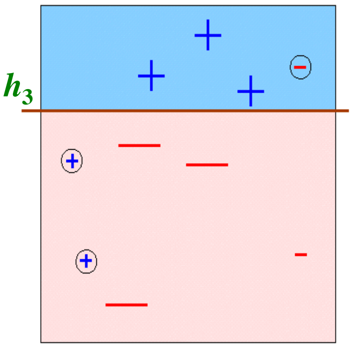


AdaBoost example (2)



AdaBoost example (2)

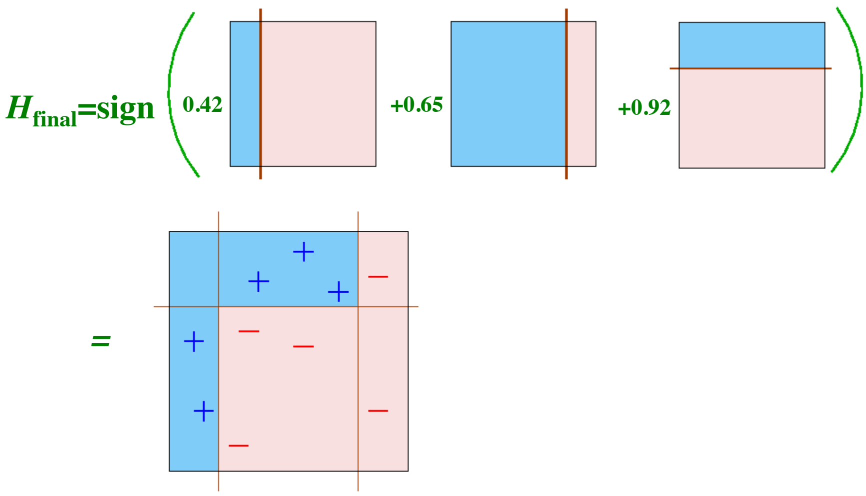
ROUND 3



ε_3=0.14

α_3=0.92

AdaBoost example (2): final hypothesis



Practical Advantages of AdaBoost

- (quite) Fast
- Simple + easy to program
- Only a single parameter to tune (T)
- No prior knowledge
- Flexible: can be combined with any classifier (neural net, C4.5, …)
- Provably effective (assuming weak learner)
  - Shift in mind set: goal now is merely to find hypotheses that are better than random guessing

AdaBoost caveats

- Performance depends on data & weak learner
- AdaBoost can fail if
  - Weak hypothesis too complex (overfitting)
  - Weak hypothesis too weak ($\alpha_t \to 0$ too quickly),
    - Underfitting
    - Low margins → overfitting
- Empirically, AdaBoost seems susceptible to noise

5. Discussions

Bagging vs. Boosting

- Training set
  - Bagging: Randomly selected samples, independent
  - Boosting: Decided by the previous one, dependent
- Prediction function
  - Bagging: no weights; easier to parallelize
  - Boosting: weights grow exponentially; sequential production

Bagging vs. Boosting (cont.)

- Performance
  - In practice, bagging almost always helps.
  - On average, boosting helps more than bagging, but it is also more common for boosting to hurt performance
  - Bagging doesn't work so well with stable models. Boosting might still help.
  - Boosting might hurt performance on noisy datasets. Bagging doesn't have this problem

Reweighting vs. Resampling

- Example weights might be harder to deal with
  - Some learning methods can't use weights on examples
  - Many common packages don't support weighs on the train
- We can resample instead:
  - Draw a bootstrap sample from the data with the probability of drawing each example is proportional to it's weight
- Reweighting usually works better but

  resampling is easier to implement

Bagging & boosting applications

- Content filtering in the Internet
- Image recognition
- Handwritten recognition
- Speech recognition
- Text categorization
- ……

A little bit more…

- Research topics
  - A uniformed theoretical framework for bagging and boosting?
  - Overfitting analyses on boosting
  - Other ensemble learning approaches?
- If you are interested in more details
  - Mistake bounds of boosting
  - Boosting and the largest margin

Overview

- Introduction to ensemble learning
- Approaches
  - Weighted majority algorithm
  - Bagging
    - Boostrap sampling
  - Boosting
- Further discussion
  - Bagging vs. boosting
  - Reweighting vs. resampling

References

- L. Breiman, "Bagging predictors", Machine Learning, 24(2):123-140, 1996.
- www.boosting.org
- T.Hastie, R.Tibshirani, J.Friedman. "The Elements of Statistical Learning - Data Mining,Inference, Prediction." Springer Verlag.
- R. Meir and G. Rätsch. An introduction to boosting and leveraging. In S. Mendelson and A. Smola, editors, *Advanced Lectures on Machine Learning*, LNCS, pages 119-184. Springer, 2003. In press. Copyright by Springer Verlag.
- R.E. Schapire. A brief introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1999.
- R. E. Schapire. The Boosting Approach to Machine Learning: An Overview. In *MSRI Workshop on Nonlinear Estimation and Classification*, Berkeley, CA, Mar. 2001.