Comp 135
Introduction to Machine Learning and Data Mining

Fall 2016

Professor: Roni Khardon

Computer Science
Tufts University

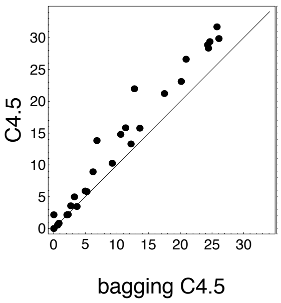
Weak and Strong Learning

- Suppose we have a learning algorithm that always (for any distribution over train/test data) gives reasonable but not necessarily great performance (e.g, accuracy >= 0.6).
- Can we somehow use this algorithm to do better? How?

Some General and Specialized Alg

- Bagging: use bootstrap sample
- Bagging of Decision Trees
- Random Forests
  - Bagging
  - Random subset of features at each node
- Random Trees
  - Select randomly among top 20 features at each node

Improving over Decision Trees



[SFBL98]

Improving over Decision Trees

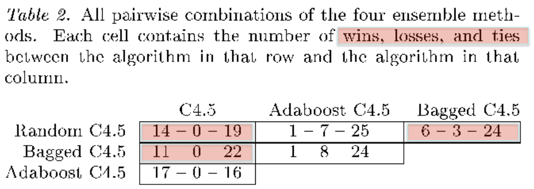


Table 2. All pairwise combinations of the four ensemble methods. Each cell contains the number of wins, losses, and ties between the algorithm in that row and the algorithm in that column.

	C4.5	Adaboost C4.5	Bagged C4.5
Random C4.5	14 – 0 – 19	1 – 7 – 25	6 – 3 – 24
Bagged C4.5	11 0 22	1 8 24	
Adaboost C4.5	17 – 0 – 16		

[D00]

- An aside: algorithmic connection to Query by Committee:
- In QbC we need to sample hypotheses from the "posterior".
- We then decided to ask for the label of an example if the sampled hypotheses disagreed on its label.
- Sampling as here can be effective.

Stability of Base Classifiers

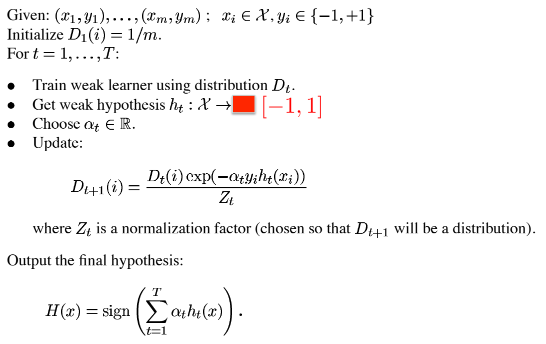
- Which of these classifiers are stable/ sensitive?
  - kNN
  - Decision Trees
  - Linear Threshold Elements (SVM)
  - Naive Bayes
  - "ZeroR"
  - "OneR"

Forcing Classifier Diversity

- Can we force the hypotheses produced by different runs to be different (even when base classifiers is not sensitive)?
- How?

- Schapire's first algorithm
- Filtering vs fixed sample / weighting
- How to predict: adaptive algorithms and weights/votes

Confidence Rated Adaboost [SS99]



Given: $(x_1, y_1), \ldots, (x_m, y_m)$; $x_i \in \mathcal{X}, y_i \in \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : \mathcal{X} \to$ 🟥 $[-1, 1]$
- Choose $\alpha_t \in \mathbb{R}$.
- Update:

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

Confidence Rated Adaboost

- In Adaboost code use

$$r_t = \sum_i D_t(i) y_i h_t(i) = E_{i \sim D_t}[y_i h_t(i)]$$

$$\alpha_t = \frac{1}{2} \ln \frac{1 + r_t}{1 - r_t}$$

- When predictions of h_t are in {-1,1} Update is such that:

  error of h_t on D_{t+1} is 0.5

Comparisons and Explanations

- Training set → generalization analysis
- Fact: $\text{Train Error} \leq e^{-\frac{1}{2}\sum r_t^2} \leq e^{-\frac{1}{2}Tr^2}$

  When $r_t \geq r$ for all t

- When T is "not too large" and "not too small": learning theory analysis guarantees that true error (on unseen data) is low

Comparisons and Explanations

- Boosting vs Bagging vs Random Forests
- Revisit accuracy slides above
- Note sensitivity to noise

Noise = 0%	C4.5	Adaboost C4.5	Bagged C4.5
Random C4.5	5 − 0 − 4	1 − 6 − 2	3 − 3 − 3
Bagged C4.5	4 − 0 − 5	0 − 5 − 4	
Adaboost C4.5	6 0 3		

Noise = 5%	C4.5	Adaboost C4.5	Bagged C4.5
Random C4.5	5 2 2	3 2 4	1 5 3
Bagged C4.5	6 − 0 − 3	5 − 1 − 3	
Adaboost C4.5	3 − 3 − 3		

Noise = 10%	C4.5	Adaboost C4.5	Bagged C4.5
Random C4.5	4 − 1 − 4	5 − 1 − 3	1 − 6 − 2
Bagged C4.5	5 − 0 − 4	6 − 1 − 2	
Adaboost C4.5	2 3 4		

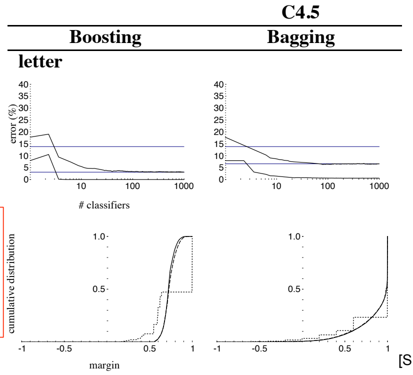
Noise = 20%	C4.5	Adaboost C4.5	Bagged C4.5
Random C4.5	5 − 2 − 2	5 − 0 − 4	0 − 2 − 7
Bagged C4.5	7 0 2	6 0 3	
Adaboost C4.5	3 − 6 − 0		

[D00]

Comparisons and Explanations

- Boosting vs Bagging vs Random Forests
- Revisit accuracy slides above
- Note sensitivity to noise
- Bias-Variance effect of bagging and boosting
- Margin explanation

Train/Test Error and Margin



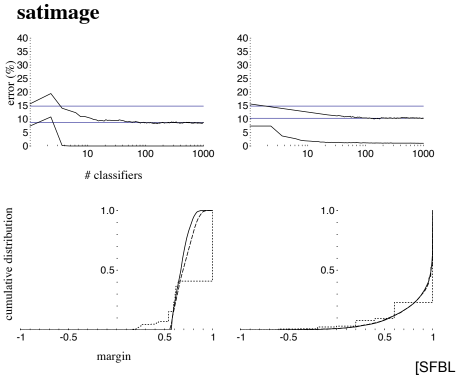
% samples

with margin<x

after 5,100,1000 iterations

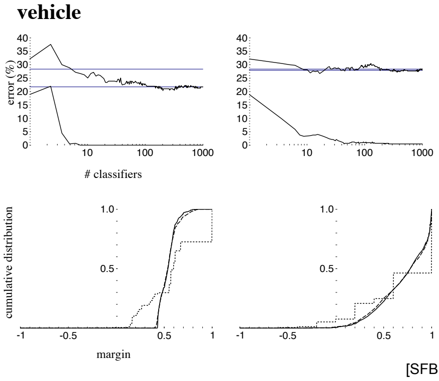
[SFBL98]

Train/Test Error and Margin



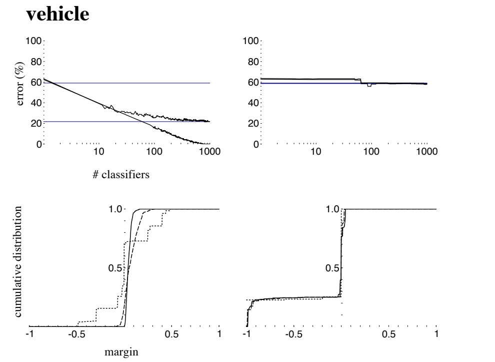
[SFBL98]

Train/Test Error and Margin



[SFBL98]

Same plot type: base learner OneR

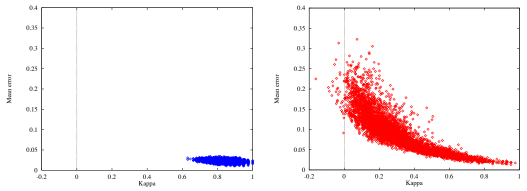


[SFBL98]

Margin Explanation

- Adaboost optimizes cumulative margin
- Learning theory says that this implies good performance
- But attempts at algorithms to optimize cumulative margin directly not as successful

Visualizing Diversity



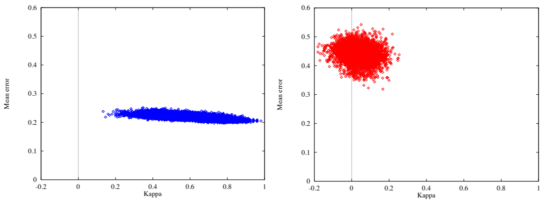
Bagging Boosting

Sick dataset; base learner C4.5; no noise

Kappa={1→ same hyp; 0 → indepndent; -1 → reverse labels}

[D00]

Visualizing Diversity



Bagging Boosting

Sick dataset; base learner C4.5; 20% noise

Kappa={1→ same hyp; 0 → indepndent; -1 → reverse labels}

[D00]

Adaboost vs SVM

- Similar final hyp when h_t is one feature
- But different optimization setting
- And different criterion:
  - Max min margin
  - Exponentially weighted cumulative margin (exponential loss)

Ensemble Methods

- Main idea: voting among diverse set of hypotheses can help reduce errors
- Different schemes to take advantage of and/or force diversity
- Bagging, Random Forests, Ada-Boosting
- Many variants exist
- Other ways of combining classifiers are also possible