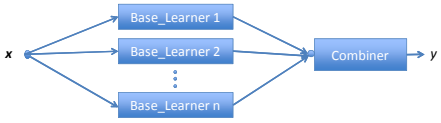
Ensemble Methods

Ishwar K Sethi

Ensemble Methods

- These methods train multiple learners for the same task
- These methods are also known as *committee-based methods,* or *multiple classifier systems*



Homogeneous vs. Heterogeneous Ensembles

- Homogeneous ensembles have all base learners of same type, e.g. decision trees
- Heterogeneous ensembles use a variety of base learners
- Base learners are also known as *weak learners* because these learners exhibit performance only slightly better than random guess. However, together they perform very well

Why a Collection of Weak Learners Performs Well?

- Suppose we have 5 completely independent learners each with an accuracy of 70%. We combine their decision using the majority rule. Then
  - $(.7^5)+5(.7^4)(.3)+10\ (.7^3)(.3^2)$
  - **83.7% majority vote accuracy**
  - 101 such classifiers
  - **99.9% majority vote accuracy**

Wisdom of crowds

Note: Binomial Distribution: The probability of observing *x* heads in a sample of *n* independent coin tosses,
where in each toss the probability of heads is *p*, is

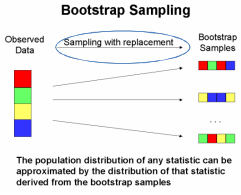
$$P(X = x|p, n) = \frac{n!}{r!(n-x)!}p^x(1-p)^{n-x}$$

How Do We Train Weak Learners?

- These learners must not be strongly correlated. Rather they should compliment each other. Why?
- Possible training approaches:
  - Different algorithms for different learners (Heterogeneous ensemble)
  - Same algorithm but different parameters
  - Different features for different learners (Random subspace)
  - Different data subsets for different learners (Bagging and boosting)

Bagging

- Bagging is an abbreviation of *Bootstrap Aggregating* (Breiman 1996)
- It uses bootstrap sampling to generate different data subsets for training different learners
- The output of different learners is combined using voting for classification and averaging for regression



Bagging

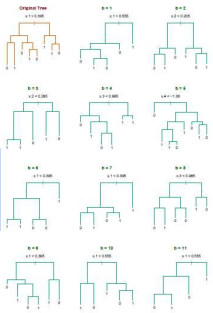
- Given m training examples, the probability that the i^{th} example is selected k times is approximated by Poisson distribution with $\lambda = 1$

  $p(k,\lambda) = e^{-\lambda}\,\lambda^k / k!$ for $k = 0, 1, 2, \ldots$

- Thus, the probability that the i^{th} example will occur at least once is $1 - (1/e) \approx 0.632$. This means that each base learner uses only about 63.2% of the original training examples. Thus, the remaining 36.8% of the examples, called **out-of-bag** examples, not used by a base learner can be used as test examples for evaluating its performance. Such an estimate is called **out-of-bag estimate**.

Bagging

- Some learners are more susceptible to addition or deletion of training examples. These kinds of learners, for example decision trees, are known as *unstable learners*.
- Learners such as *k-NN* classifier are hardly affected by some additions/deletions of training examples. Such learners are called *stable learners*.
- Bagging works well with unstable learners.



Tree pruning is not desirable because un-pruned trees are more unstable and consequently better Suited for bagging

Simulated data: sample of size N = 30, with two classes and 5 features, each having a standard Gaussian distribution with pairwise correlation 0.95. The response Y was generated according to $\Pr(Y = 1|x_i \le 0.5) = 0.2$, $\Pr(Y = 1|x_i > 0.5) = 0.8$. The Bayes error is 0.2. A test sample of size 2000 was also generated from the same population. We fit classification trees to the training sample and to each of 200 bootstrap samples

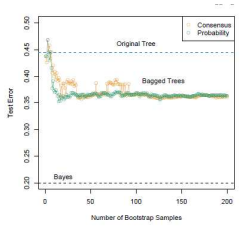


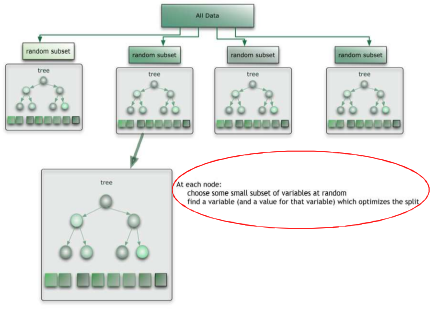
FIGURE 8.10. *Error curves for the bagging example of Figure 8.9. Shown is the test error of the original tree and bagged trees as a function of the number of bootstrap samples. The orange points correspond to the consensus vote, while the green points average the probabilities.*

The Elements of Statistical Learning, Hastie, Tibshirani, and Friedman

Random Forest

- Random Forest (RF) (Breiman, 2001)
- RF is an extension of bagging with decision trees. In RF, a random subset of features is used to select splits at each tree node. This is in addition to bootstrap sampling to create different data subsets.

Random Forest



Random Forest Performance

- Similar to SVM and Neural Networks

Test set misclassification error (%)

Data set	Forest	Single tree
Breast cancer	2.9	5.9
Ionosphere	5.5	11.2
Diabetes	24.2	25.3
Glass	22.0	30.4
Soybean	5.7	8.6
Letters	3.4	12.4
Satellite	8.6	14.8
Shuttle $\times 10^3$	7.0	62.0
DNA	3.9	6.2
Digit	6.2	17.1

Boosting

- Boosting refers to any procedure that combines many weak learners to yield a much higher performance. Unlike bagging, the base learners in boosting are sequentially generated by focusing on examples misclassified by earlier weak learners in the chain.

Key Idea of Boosting

- Each training example is assigned a weight. The weight determines the training example's probability of being selected for training by the next base learner.
- Initially, all examples have identical weight. The misclassified examples see their weight go up to increase their selection chance for training
- The outputs of the learners are combined using weights to yield the final response

Boosting: Basic Algorithm

- General Loop:

  Set all examples to have equal uniform weights.
  For t from 1 to T do:
  Learn a hypothesis, h_t, from the weighted examples
  Decrease the weights of examples h_t classifies correctly
- Base (weak) learner must focus on correctly classifying the most highly weighted examples while strongly avoiding over-fitting.
- During testing, each of the T hypotheses get a weighted vote proportional to their accuracy on the training data.

16

AdaBoost Pseudocode

TrainAdaBoost(D, BaseLearn)
  For each example d_i in D let its weight w_i=1/|D|
  Let H be an empty set of hypotheses
  For t from 1 to T do:
    Learn a hypothesis, h_t, from the weighted examples: h_t=BaseLearn(D)
    Add h_t to H
    Calculate the error, ε_t, of the hypothesis h_t as the total sum weight of the examples that it classifies incorrectly.
    If $\varepsilon_t > 0.5$ then exit loop, else continue.
    Let $\beta_t = \varepsilon_t \ / \ (1 - \varepsilon_t)$
    Multiply the weights of the examples that h_t classifies correctly by β_t
    Rescale the weights of all of the examples so the total sum weight remains 1.
  Return H

TestAdaBoost(*ex*, H)
    Let each hypothesis, h_t, in H vote for *ex*' s classification with weight log(1/ β_t)
    Return the class with the highest weighted vote total.

17

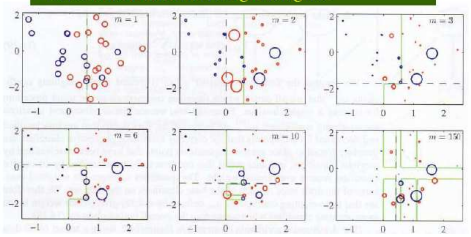
How is Example Weighting Used?

- Generic approach is to replicate examples in the training set proportional to their weights (e.g. 10 replicates of an example with a weight of 0.01 and 100 for one with weight 0.1).
- Most algorithms can be enhanced to efficiently incorporate weights directly in the learning algorithm so that the effect is the same. For example when calculating information gain with decision tree learning, simply increment the count by w_i for example i rather than by 1.

18

Illustration of AdaBoost

- Weak Learner: Threshold applied to one or other axis
- Ensemble decision boundary is Green
- Current base learner is dashed black line
- Size of circle indicates weight assigned



Combining Weak Learners

- Averaging (simple or weighted) for regression
- Voting (simple or weighted) for classification tasks
- Stacking. Here the combiner is trained to combine the output of weak learners. The training data for this is different from the data used for weak learners. The combiner in stacking is often called the *second-level learner* or *meta learner*. The weak learners are called the *first-level learners*.

Ensembles' Performance

- Ensembles have been used to improve generalization accuracy on a wide variety of problems.
- On average, Boosting provides a larger increase in accuracy than Bagging.
- Boosting on rare occasions can degrade accuracy.
- Bagging more consistently provides a modest improvement.
- Boosting is particularly subject to over-fitting when there is significant noise in the training data.

21

Issues in Ensembles

- Parallelism in Ensembles: Bagging is easily parallelized, Boosting is not.
- Variants of Boosting to handle noisy data.
- How "weak" should a base-learner for Boosting be?
- What is the theoretical explanation of boosting's ability to improve generalization?
- Exactly how does the diversity of ensembles affect their generalization performance.
- Combining Boosting and Bagging.

22