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 $\geq 0.6$).

- Can we somehow use this algorithm to do better? How?

Some General and Specialized Alg

- Bagging
- Bagging of Decision Trees
- Random Forests
- Random Trees

Improving over Decision Trees

Figure 2: Comparison of C4.5 versus bagging C4.5 and boosting C4.5 on a set of 27 benchmark problems as reported by Freund and Schapire [18]. Each point in each scatter plot shows the test error rate of the two competing algorithms on a single benchmark. The $y$-coordinate of each point gives the test error rate (in percent) of C4.5 on the given benchmark, and the $x$-coordinate gives the error rate of bagging (left plot) or boosting (right plot). All error rates have been averaged over multiple runs.

Our main observation is that both boosting and bagging tend to increase the margins associated with
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?

Confidence Rated AdaBoost [SS99]

Given: \((x_1, y_1), \ldots, (x_m, y_m)\) \(y_i \in \{-1, +1\}\)
Initial: \(D(0) = \frac{1}{m}\)
For \(t = 1, \ldots, T\):
  - Train weak learner using distribution \(D_t\)
  - Get weak hypothesis \(h_t: X \mapsto [-1, 1]\)
  - Choose \(\alpha_t \in \mathbb{R}\)
  - Update:
    \[
    D_{t+1}(x) = \frac{D_t(x) e^{-\alpha_t y_t h_t(x)}}{Z_t}
    \]
    where \(Z_t\) is a normalization factor (chosen so that \(D_{t+1}\) will be a distribution).
Output the final hypothesis:
\[
H(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t h_t(x) \right).
\]

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
  Train Error \(\leq e^{-\frac{1}{2} \sum r_t^2} \leq e^{-\frac{1}{2} T r^2}\)
  When \(r_t \geq r\) for all \(t\)

- When \(T\) is "not too large" and "not too small": CLT analysis guarantees low error
Using C4.5 as the base learning algorithm. Results are given for the letter, satimage and vehicle datasets. (See Figure 4: Error curves and margin distribution graphs for three voting methods (bagging, boosting and ECOC) caption under Figure 1 for an explanation of these curves.)

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

Margins Explanation

Visualizing Diversity

Bagging

Boosting

Sick dataset; base learner C4.5; no noise

Kappa: (1 → same hyp; 0 → independent; -1 → reverse labels)
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

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)