Ensembles of Classifiers

- Traditional approach: Use one classifier
- Can one do better?
- Approaches:
  - Cross-validated committees
  - Bagging
  - Boosting
  - Stacking

Voting

- Assume
  - Errors are independent (suppose 30% error)
  - Majority vote
- Probability that majority is wrong...
  - $\prob = \text{area under binomial distribution}$
- If individual area is 0.3
- Area under curve for $\geq 11$ wrong is 0.026
- Order of magnitude improvement!

Constructing Ensembles

Cross-validated committees

- Partition examples into $k$ disjoint equiv classes
- Now create $k$ training sets
  - Each set is union of all equiv classes except one
  - So each set has $(k-1)/k$ of the original training data
- Now train a classifier on each set

Ensemble Construction II

Bagging

- Generate $k$ sets of training examples
- For each set
  - Draw $m$ examples randomly (with replacement)
  - From the original set of $m$ examples
- Each training set corresponds to
  - $63.2\%$ of original (+ duplicates)
- Now train classifier on each set
- Intuition: Sampling helps algorithm become more robust to noise/outliers in the data
Ensemble Creation III
Boosting

- Create 1st weight distribution (uniform) over training ex: \( \{w_1^1\} \)
- Create M classifiers iteratively:
  - On iteration \( m \)
  - Train \( C_m \) by minimizing \( \sum w_m^i [y(x_i) \neq t] \)
  - Modify distribution: increase \( P \) of each example predicted incorrectly
  - Assign confidence to classifier \( C_m = f(\text{error}) \); prefer accurate ones

- Combine
- Create harder and harder learning problems...
- Optimized choice of examples

Boosting Decision Stumps

Bagging vs Boosting

Ensemble Creation IV
Stacking

- Train several base learners
- Next train meta-learner
  - Learns when base learners are right / wrong
  - Now meta learner arbitrates

- Train using cross validated committees
  - Meta-L inputs = base learner predictions
  - Training examples = 'test set' from cross validation

Causes of Expected Error

- Variance:
  - How much \( h(x^*) \) varies from one training set to another
- Bias:
  - Describes the \textit{average} error of \( h(x^*) \).
- Noise:
  - Describes how much \( y^* \) varies from \( f(x^*) \)

Tradeoff

- Overfitting – too much variance
- Underfitting – too much bias
Interlude: Bias

• Bias (Statistical)
  – the difference between an estimator’s expectation and the true value of the parameter being estimated.

• Inductive Bias
  – Set of assumptions that the learner uses to predict outputs given inputs that it has not encountered

• Bias (Engineering)
  – establishing predetermined voltage at a point in a circuit to set an appropriate operating point.
  
  \[ y = w_0 + \sum w_i x_i \]

Bias-Variance Analysis in Regression

• True function is \( y = f(x) + \varepsilon \)
  – where \( \varepsilon \) is normally distributed with zero mean and standard deviation \( \sigma \).

• Given a set of training examples, \( \{(x_i, y_i)\} \), we fit an hypothesis \( h(x) = w \cdot x + b \) to the data to minimize the squared error
  \[ \sum_i (y_i - h(x_i))^2 \]

Example: 20 points

\[ y = x + 2 \sin(1.5x) + N(0,0.2) \]

50 fits (20 examples each)

Bias-Variance Analysis

• Now, given a new data point \( x^* \) (with observed value \( y^* = f(x^*) + \varepsilon \)), we would like to understand the expected prediction error

\[ E[(y^* - h(x^*))^2] \]

Classical Statistical Analysis

• Imagine that our particular training sample \( S \) is drawn from some population of possible training samples according to \( P(S) \).

• Compute \( E_p [ (y^* - h(x^*))^2 ] \)

• Decompose this into “bias”, “variance”, and “noise”
Lemma

- Let $Z$ be a random variable with probability distribution $P(Z)$.
- Let $Z = E[Z]$ be the average value of $Z$.
  - $E[Z^2] - Z^2$

Bias-Variance-Noise Decomposition

$E[(h(x^*) - y^*)^2] = E[h(x^*)^2 - 2h(x^*)y^* + y^{*2}]$

$= E[h(x^*)^2] - 2E[h(x^*)]E[y^*] + E[y^{*2}]$

$= E[(h(x^*) - h(x^*))^2] + h(x^*)^2$ (lemma)

$- 2h(x^*)f(x^*)$

$+ E[(y^* - f(x^*))^2] + f(x^*)^2$ (lemma)

$= E[(h(x^*) - h(x^*))^2] + [bias^2]$

$+ E[(y^* - f(x^*))^2] + [noise]$

Bias, Variance, and Noise

- Variance: $E[(h(x^*) - h(x^*))^2]$
  Describes how much $h(x^*)$ varies from one training set $S$ to another.
- Bias: $[h(x^*) - f(x^*)]$
  Describes the average error of $h(x^*)$.
- Noise: $E[(y^* - f(x^*))^2] = E[\epsilon^2] = \sigma^2$
  Describes how much $y^*$ varies from $f(x^*)$.

50 fits (20 examples each)
Noise

50 fits (20 examples each)

Distribution of predictions at x=2.0

Distribution of predictions at x=5.0

Polynomial Curve Fitting

Hypothesis Space

\[ p(x, w) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j \]
Regularization:

\[
\tilde{E}(w) = \frac{1}{2} \sum_{n=1}^{N} (y_n(x_n, w) - t_n)^2 + \frac{\lambda}{2} \|w\|^2
\]

Penalize large coefficient values

Increasing \( \lambda \) trades bias for variance

Regularization: \( \ln \lambda = -18 \)

Regularization: \( \ln \lambda = 0 \)
Ensemble Methods

- Combining many biased learners
  - Eg decision stumps
- Keeps variance low
- Can represent more expressive hypotheses
  - Hence, also lowers error from bias